**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MÊM**

**ĐỀ TÀI:**

**TÌM HỂU VỀ LẬP TRÌNH ĐA TẦNG VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ**

**Giảng viện hướng dẫn:** ThS. Trần Phong Nhã

**Sinh viên thực hiện:**

1. Phạm Thành Hậu - 5951071024 Lớp: CQ.59.CNTT

2. Phạm Trọng Trường - 5951071113 Lớp: CQ.59.CNTT

3. Trần Quang Trường - 5951071114 Lớp: CQ.59.CNTT

4. Hoàng Huy Tuấn - 5951071116 Lớp: CQ.59.CNTT

Tp. Hồ Chí Minh năm 2021

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌCGIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MÊM**

**ĐỀ TÀI:**

**TÌM HỂU VỀ LẬP TRÌNH ĐA TẦNG VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ**

**Giảng viện hướng dẫn:** ThS. Trần Phong Nhã

**Sinh viên thực hiện:**

1. Phạm Thành Hậu - 5951071024 Lớp: CQ.59.CNTT

2. Phạm Trọng Trường - 5951071113 Lớp: CQ.59.CNTT

3. Trần Quang Trường - 5951071114 Lớp: CQ.59.CNTT

4. Hoàng Huy Tuấn - 5951071116 Lớp: CQ.59.CNTT

Tp. Hồ Chí Minh năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH ĐỘC LẬP – TỰ DO – HẠNH PHÚC**

**NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Sinh viên thực hiện:** Phạm Thành Hậu -5951071024

Phạm Trọng Trường -5951071113

Trần Quang Trường -5951071114

Hoàng Huy Tuấn -5951071116

**Khóa:** 59 **Lớp:** CQ.59.CNTT

1. **Tên đề tài**

Tìm hiểu về lập trình da tầng và xây dựng ứng dụng minh hoạ.

1. **Mục đích, yêu cầu**
   1. **Mục đích:**

* Hiểu rõ quy trình, cơ cấu tổ chức và cơ chế hoạt động của nhà hàng.
* Xây dựng “Phần mềm quản lý nhà hàng triển khai bằng lập trình đa tầng” với mục đích chính là ứng dụng được vào phần mềm nhằm giải quyết các nhu cầu của người dùng và ứng dụng công nghệ thông tin trong quản lí hoạt động giúp tối ưu hóa công việc, đem lại sự tiện lợi, tiết kiệm chi phí và lưu trữ dữ liệu… Hệ thống được xây dựng trên nền tảng WindowsForm.
  1. **Yêu cầu:**
* **Yêu cầu công nghệ**
  + Sử dụng ngôn ngữ lập trình C#.
  + Sử dụng công cụ Visual Studio 2019 và .Net Framework.
* **Yêu cầu chức năng**
  + Phần mềm cho phép người dùng: Quản Lý Nhà Hàng lập trình đa tầng của mình dễ dàng hơn thay vì quản lý bằng phương pháp truyền thống .
* **Yêu cầu phi chức năng**
  + Tốc độ: Tối ưu về tốc độ, làm việc với dữ liệu vừa phải, tốc độ hiển thị nhanh.
  + Giao diện: Thân thiện với người dùng và dễ dàng thao tác.

1. **Nội dung và phạm vi đề tài**
   1. **Nội dung:**
      * Cơ sở lý thuyết.
      * Tổng quan về kiến trúc 3 tầng .
      * Chương trình demo.
      * Kết quả thu được.
   2. **Phạm vi:**
      * Nghiên cứu sử dụng công cụ Visual Studio và ngôn ngữ C# Winform.
      * Nghiên cứu và phân tích hê thống.
2. **Công nghệ, công cụ và ngôn ngữ lập trình**
   * + Công nghệ sử dụng: .Net Framework.
     + Công cụ Visual Studio 2019, Microsoft Excel.
     + Ngôn ngữ lập trình: C# Winform.
3. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**
   * + Bài báo cáo đề tài (bản Word và File Power Point)
     + Xây dựng được phần mềm quản lý nhà hàng triển khai bằng mô hình đa tầng.

**6. Giảng viên và cán bộ hướng dẫn**

Họ tên: Đơn vị công tác:

Điện thoại: Email:

**Tp. Hồ Chí Minh, ngày ….… tháng ….… năm 2021**

**Sinh viên thực hiện**

Phạm Thành Hậu

Phạm Trọng Trường

Trần Quang Trường

Hoàng Huy Tuấn

**LỜI MỞ ĐẦU**

Để hoàn thành đề tài này trước hết em xin gửi đến quý thầy, cô **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh** lời cảm ơn chân thành vì đã truyền đạt cho em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt em xin gửi đến Thầy Trần Phong Nhã-giáo viên dạy môn chuyên đề công nghệ phần mềm- lời cảm ơn sâu sắc nhất. Tuy đề tài không được lớn lao nhưng nếu không được sự hướng dẫn, giảng dạy chỉ bảo tận tình của cô.

Vì thời gian làm đề tài có hạn cũng như hiểu biết cá nhân còn hạn chế, chúng em cũng đã nỗ lực hết sức để hoàn thành đề tài một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc thầy Trần Phong Nhã cũng như Quý Thầy Cô trong Bộ môn Công nghệ thông tin hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

**Tp. Hồ Chí Minh, ngày ….… tháng ….… năm 2021**

**Sinh viên thực hiện**

Phạm Thành Hậu

Phạm Trọng Trường

Trần Quang Trường

Hoàng Huy Tuấn

# 

**NHẬN XÉT CỦA GIÁO VIÊN**

*Tp. Hồ Chí Minh, ngày … tháng … năm …*

Giảng viên hướng dẫn

**MỤC LỤC**

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1](#_Toc84099572)

[1.1 Giới thiệu về đa tầng 1](#_Toc84099573)

[1.2 Một số thuật ngữ thông dụng trong kiến trúc N-Tier 2](#_Toc84099574)

[1.2.1 Mạng phân tán 2](#_Toc84099575)

[1.2.2 Kiến trúc Máy khách – Máy chủ 2](#_Toc84099576)

[1.2.3 Nền tảng 3](#_Toc84099577)

[1.2.4 Cơ sở dữ liệu 3](#_Toc84099578)

[1.3 Các loại kiến trúc N-Tier 3](#_Toc84099579)

[1.3.1 Kiến trúc 3 tầng 4](#_Toc84099580)

[1.3.2 Kiến trúc 2 tầng 5](#_Toc84099581)

[1.3.4 Kiến trúc 1 tầng 6](#_Toc84099582)

[1.4 Ưu điểm và nhược điểm của kiến trúc đa tầng 6](#_Toc84099583)

[1.4.1 Ưu điểm và nhược điểm 6](#_Toc84099584)

[1.5 Những công nghệ hỗ trợ xây dựng các ứng dụng 3 tier 6](#_Toc84099585)

[CHƯƠNG 2: TỔNG QUAN VỀ KIẾN TRÚC 3 TẦNG 8](#_Toc84099586)

[2.1 Mô hình 3 Tier 8](#_Toc84099587)

[2.1.1 Ưu điểm và nhược điểm mô hình 3 Tier 8](#_Toc84099588)

[2.2.2 Lý do mô hình 3 tầng 9](#_Toc84099589)

[2.2 Sự khác nhau giữa 3 layer và 3 tầng 9](#_Toc84099590)

[2.2.1 Giới thiệu 3 layer 9](#_Toc84099591)

[2.2.2 Sự khác nhau 9](#_Toc84099592)

[2.3 Triển khai và xây dựng một ứng dụng 3 tầng trên c# 10](#_Toc84099593)

[2.3.1 Tầng giao diện người dùng(UIL) 10](#_Toc84099594)

[2.3.2 Tầng Logic nghiệp vụ dự án(BLL) 10](#_Toc84099595)

[2.3.3 Tầng truy cập cơ sở dữ liệu (DAL) 10](#_Toc84099596)

[CHƯƠNG 3: CHƯƠNG TRÌNH DEMO 13](#_Toc84099597)

[3.1 Giao diện chính 13](#_Toc84099598)

[3.2 Giao diện thống kê 13](#_Toc84099599)

[3.3 Giao diện bán hàng 14](#_Toc84099600)

[3.4 Giao diện kho hàng 14](#_Toc84099601)

[KẾT LUẬN VÀ KHUYẾN NGHỊ 15](#_Toc84099602)

[TÀI LIỆU THAM KHẢO 16](#_Toc84099603)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Sơ đồ kiến trúc bậc N 1](#_Toc84099560)

[Hình 2: Sơ đồ kiến trúc 3 tầng 4](#_Toc84099561)

[Hình 3: Lớp giao diện 4](#_Toc84099562)

[Hình 4: Tạo dự án cơ sở dữ liệu 10](#_Toc84099563)

[Hình 5: Thiết kế cơ sở dữ liệu 11](#_Toc84099564)

[Hình 6: Triển khai sql 11](#_Toc84099565)

[Hình 7: Kiểm tra sql 12](#_Toc84099566)

[Hình 8: Giao diện chính 13](#_Toc84099567)

[Hình 9: Giao diện thống kê 13](#_Toc84099568)

[Hình 10: Giao diện quản lý bán hàng 14](#_Toc84099569)

[Hình 11: Giao diện kho hàng 14](#_Toc84099570)

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về đa tầng

Một **N-Tier Application** chương trình là một trong đó là phân bố trong ba hoặc riêng biệt nhiều máy tính trong một mạng lưới phân phối.

Hình thức phổ biến nhất của cấp n là Ứng dụng 3 cấp, và nó được phân thành ba loại.

* Lập trình giao diện người dùng trong máy tính của người dùng
* Logic kinh doanh trong một máy tính tập trung hơn và
* Dữ liệu bắt buộc trong máy tính quản lý cơ sở dữ liệu.

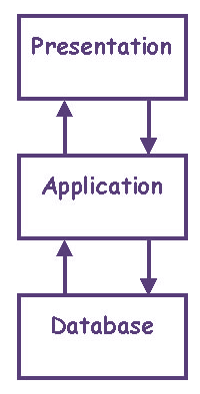
Mô hình kiến trúc này cung cấp cho các Nhà phát triển phần mềm tạo ứng dụng hệ thống có thể tái sử dụng với tính linh hoạt tối đa.

Trong **N-tier, “N”** đề cập đến một số tầng hoặc lớp đang được sử dụng như thế nào - **2 tầng, 3 tầng hoặc 4 tầng, vv** . Nó còn được gọi là “ **Kiến trúc nhiều tầng ”** .

**Kiến trúc n-tier** là một mô hình kiến trúc phần mềm phù hợp để hỗ trợ các ứng dụng máy khách-máy chủ cấp doanh nghiệp bằng cách cung cấp các giải pháp về khả năng mở rộng, bảo mật, khả năng chịu lỗi, khả năng tái sử dụng và khả năng bảo trì. Nó giúp các nhà phát triển tạo ra các ứng dụng linh hoạt và có thể tái sử dụng.

**Kiến trúc N-Tier**

Ở đây mô tả biểu diễn dạng sơ đồ của hệ thống n-tier - các lớp trình bày, ứng dụng và cơ sở dữ liệu.



Hình 1: Sơ đồ kiến trúc bậc N

Ba lớp này có thể được chia nhỏ hơn nữa thành các lớp con khác nhau tùy theo yêu cầu.

Một số trang web phổ biến đã áp dụng kiến ​​trúc này là

* MakeMyTrip.com
* Amazon.com, v.v.

## Một số thuật ngữ thông dụng trong kiến trúc N-Tier

### 1.2.1 Mạng phân tán

**Mạng phân tán:** Là một kiến ​​trúc mạng, trong đó các thành phần nằm trong các máy tính mạng phối hợp và giao tiếp các hành động của chúng chỉ bằng cách chuyển các thông điệp. Nó là một tập hợp của nhiều hệ thống nằm ở các nút khác nhau nhưng xuất hiện với người dùng như một hệ thống duy nhất.

* Nó cung cấp một mạng truyền thông dữ liệu duy nhất có thể được quản lý riêng biệt bởi các mạng khác nhau.
* Một ví dụ về Mạng phân tán - nơi các máy khách khác nhau được kết nối trong kiến ​​trúc mạng LAN ở một bên và ở phía bên kia chúng được kết nối với các bộ chuyển mạch tốc độ cao cùng với một giá đỡ các máy chủ chứa các nút dịch vụ.

### 1.2.2 Kiến trúc Máy khách – Máy chủ

**Kiến trúc Máy khách-Máy chủ:** Là một mô hình kiến ​​trúc trong đó máy khách (một chương trình) yêu cầu một dịch vụ từ một máy chủ (một chương trình khác) tức là Nó là một dịch vụ phản hồi yêu cầu được cung cấp qua internet hoặc thông qua mạng nội bộ.

Trong mô hình này, **Client** sẽ đóng vai trò là một tập hợp chương trình / mã thực thi một tập hợp các hành động qua mạng. Trong khi **Server** , ngược lại, là một tập hợp của một chương trình khác, nó sẽ gửi các tập hợp kết quả đến hệ thống máy khách theo yêu cầu.

* Trong trường hợp này, máy khách cung cấp giao diện cho người dùng cuối để yêu cầu dịch vụ hoặc tài nguyên từ máy chủ và mặt khác máy chủ xử lý yêu cầu và hiển thị kết quả cho người dùng cuối.
* Một ví dụ về Mô hình Máy khách-Máy chủ - một máy ATM. Ngân hàng là máy chủ xử lý đơn trong cơ sở dữ liệu khách hàng lớn và máy ATM là máy khách có giao diện người dùng với một số xử lý ứng dụng đơn giản.

### 1.2.3 Nền tảng

Trong khoa học máy tính hoặc ngành công nghiệp phần mềm, nền tảng là một hệ thống mà chương trình ứng dụng có thể chạy. Nó bao gồm sự kết hợp giữa phần cứng và phần mềm có hướng dẫn tích hợp cho bộ vi xử lý / bộ vi xử lý để thực hiện các hoạt động cụ thể.

* Nói một cách đơn giản hơn, nền tảng là một hệ thống hoặc một cơ sở nơi bất kỳ ứng dụng nào có thể chạy và thực thi để đạt được một tác vụ cụ thể.
* Ví dụ về Nền tảng - Một máy cá nhân được tải Windows 2000 hoặc Mac OS X làm ví dụ về 2 nền tảng khác nhau.

### 1.2.4 Cơ sở dữ liệu

Là tập hợp thông tin một cách có tổ chức để có thể dễ dàng truy cập, quản lý và cập nhật thông tin.

* Ví dụ về Cơ sở dữ liệu - MySQL, [SQL](https://www.guru99.com/sql.html) Server và Cơ sở dữ liệu Oracle là một số Db phổ biến.

## Các loại kiến trúc N-Tier

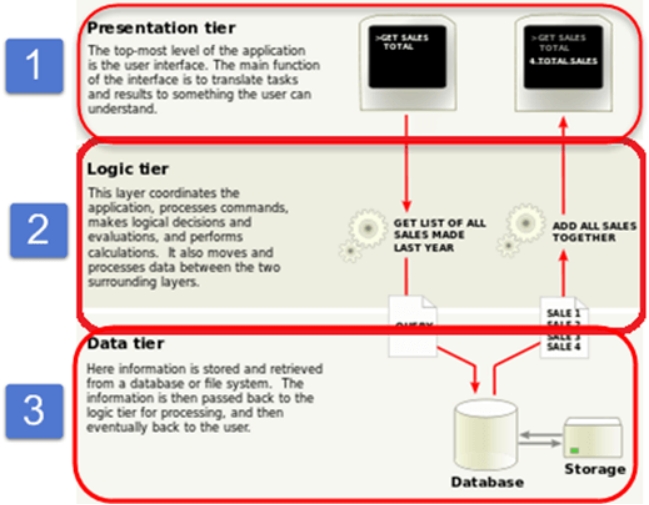
Có nhiều loại Kiến trúc N-Tier khác nhau, như Kiến trúc **3 bậc, Kiến trúc 2 bậc và Kiến trúc 1- Bậc.**

Đầu tiên, chúng ta sẽ thấy Kiến trúc 3 tầng, điều này rất quan trọng.

### 1.3.1 Kiến trúc 3 tầng

Bằng cách nhìn vào sơ đồ dưới đây, bạn có thể dễ dàng nhận ra rằng **kiến trúc 3 tầng** có ba Tầng khác nhau.

* Tầng trình bày
* Tầng nghiệp vụ
* Tầng cơ sở dữ liệu



Tầng **giao diện người dùng hoặc tầng trình bày**

Graphical user interface

Description automatically generated with low confidence

Hình 3: Lớp giao diện

Đây là cấp cao nhất của ứng dụng. Tầng trình bày hiển thị thông tin liên quan đến các dịch vụ như duyệt hàng hóa, mua hàng và nội dung giỏ hàng. Nó giao tiếp với các tầng khác mà qua đó nó đưa ra kết quả cho tầng trình duyệt / máy khách và tất cả các tầng khác trong mạng. Nói một cách dễ hiểu, nó là một lớp mà người dùng có thể truy cập trực tiếp (chẳng hạn như trang web hoặc GUI của hệ điều hành).

Tầng **nghiệp vụ**

Đây là chức năng của lớp nghiệp vụ chấp nhận dữ liệu từ lớp ứng dụng và chuyển nó đến lớp dữ liệu.

* Logic nghiệp vụ hoạt động như một giao diện giữa lớp Khách hàng và Lớp truy cập dữ liệu
* Tất cả logic nghiệp vụ - như xác thực dữ liệu, tính toán, chèn / sửa đổi dữ liệu đều được viết dưới lớp logic nghiệp vụ.
* Nó giúp giao tiếp nhanh hơn và dễ dàng hơn giữa máy khách và lớp dữ liệu
* Xác định một hoạt động quy trình công việc thích hợp cần thiết để hoàn thành một nhiệm vụ.

Tầng **truy cập dữ liệu**

Đây là chức năng của lớp dữ liệu, nhận dữ liệu từ lớp nghiệp vụ và thực hiện các thao tác cần thiết vào cơ sở dữ liệu.

### 1.3.2 Kiến trúc 2 tầng

Nó giống như kiến ​​trúc Client-Server, nơi giao tiếp diễn ra giữa máy khách và máy chủ.

Trong kiểu kiến ​​trúc phần mềm này, Tầng trình bày hoặc Tầng giao diện người dùng chạy ở phía máy khách trong khi Tầng tập dữ liệu được thực thi và lưu trữ ở phía máy chủ.

Không có Tầng logic nghiệp vụ hoặc Tầng ngay lập tức giữa máy khách và máy chủ.

### 1.3.4 Kiến trúc 1 tầng

Đây là cách đơn giản nhất vì nó tương đương với việc chạy ứng dụng trên máy tính cá nhân. Tất cả các thành phần bắt buộc để một ứng dụng chạy đều nằm trên một ứng dụng hoặc một máy chủ.

Lớp trình bày, lớp logic nghiệp vụ và lớp dữ liệu đều nằm trên một máy duy nhất.

## Ưu điểm và nhược điểm của kiến trúc đa tầng

### 1.4.1 Ưu điểm và nhược điểm

|  |  |
| --- | --- |
| **Thuận lợi** | **Nhược điểm** |
| * Khả năng mở rộng | * Tăng nỗ lực |
| * Toàn vẹn dữ liệu | * Tăng độ phức tạp |
| * Khả năng tái sử dụng |  |
| * Giảm phân phối |  |
| * Cải thiện bảo mật |  |
| * Tính khả dụng được cải thiện |  |

Do đó, nó là một phần của chương trình mã hóa các vấn đề kinh doanh trong thế giới thực và xác định cách dữ liệu có thể được cập nhật, tạo, lưu trữ hoặc thay đổi để hoàn thành nhiệm vụ hoàn chỉnh.

## Những công nghệ hỗ trợ xây dựng các ứng dụng 3 tier

Tùy thuộc vào nền tảng, bạn có thể chọn một trong các công nghệ như EJB (J2EE), COM+ (Windows), hay cũng có thể dùng các máy chủ web như nền tảng xây dựng lớp giữa (dùng webservice). Tuy nhiên, EJB và COM+ là hai tùy chọn tốt nhất vì nó có nhiều công nghệ hỗ trợ như Object Pooling, Authentication và Authority, Resource management, Remote Object Access, Transaction…Các công nghệ truyền thông điệp như JMS hay MSMQ cũng hỗ trợ nhiều trong việc tạo các lời gọi không đồng bộ.

# TỔNG QUAN VỀ KIẾN TRÚC 3 TẦNG

## Mô hình 3 Tier

3-tiers là một kiến trúc kiểu **Client/Server** mà trong đó giao diện người dùng (**UI-User Interface**), các quy tắc xử lý(**BR-business rule** hay **BL-business logic**), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế.

3-Tiers có tính vật lý (physical): là mô hình client-server (mỗi tier có thể đặt chung 1 nơi hoặc nhiều nơi, kết nối với nhau qua Web services, WCF, Remoting…).

Như vậy, ta có thể mô hình này phân tách ứng dụng ra làm 3 module riêng biệt, bao gồm:

– **Presentation tier:** bao gồm các thành phần phần xử lý giao diện Graphic User Interface (GUI). Được dùng để giao tiếp với người dùng, nhiệm vụ chính là hiển thị dữ liệu và nhận dữ liệu từ người dùng.  
– **Business tier:** gồm các thành phần Business Logic Layer (BLL), Data Access Layer (DAL) và Data Tranfer Object (DTO). Được dùng để cung cấp các chức năng của phần mềm.  
– **Data tier:** lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,…Dùng để lưu trữ dữ liệu, cho phép lớp Business Logic có thể tìm kiếm, trích xuất, cập nhật…

### 2.1.1 Ưu điểm và nhược điểm mô hình 3 Tier

Ưu điểm:

– Dễ dàng mở rộng, thay đổi quy mô của hệ thống: Khi cần tải lớn, người quản trị có thể dễ dàng thêm các máy chủ vào nhóm, hoặc lấy bớt ra trong trường hợp ngược lại.

Nhược điểm:

– Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói -> truyền đi -> mở gói trước khi có thể dùng được.

– Việc phát triển ứng dụng phức tạp hơn.

### 2.2.2 Lý do sử dụng mô hình 3 tầng

Khi dùng từ layer, chúng ta nói tới việc phân chia ứng dụng thành các thành phần một cách logic theo chức năng hoặc theo vai trò, điều này giúp phần mềm của bạn có cấu trúc sáng sủa, dễ dùng lại, từ đó giúp việc phát triển và bảo trì dễ dàng hơn. Các layer khác nhau khi được thực thi vẫn có thể nằm trong cùng một vùng bộ nhớ của một process, và hiển nhiên việc giao tiếp giữa 2 layer có thể không phải là giao tiếp giữa 2 process, đồng nghĩa với việc chúng không liên quan tới mô hình **Client/Server.**

Trái lại, tier liên quan đến cách phân chia một cách vật lý các thành phần trên các máy tính khác nhau.

Điều làm nhiều người nhầm lẫn giữa layer và tier là chúng có cùng cách phân chia (presentation, business, data), tuy nhiên trên thực tế chúng khác nhau. Vì cách phân chia như trên nên 1 tier có thể chứa nhiều hơn 1 layer.

## Sự khác nhau giữa 3 layer và 3 tầng

### 2.2.1 Giới thiệu 3 layer

Layer là sự phân tách mã theo phương pháp ogical và Tier là sự phân tách vật lý của mã .

Nói cách khác, chúng ta có thể nói Tiers là sự triển khai vật lý của các lớp. Hoặc một Lớp trở thành một Lớp nếu nó có thể được tách biệt về mặt vật lý khỏi các Lớp mà nó sử dụng.

### 2.2.2 Sự khác nhau

Trong kiến ​​trúc 3 lớp, Lớp truy cập cơ sở dữ liệu **(DAL)** , Lớp logic nghiệp vụ **(BLL)** và Lớp giao diện người dùng **(UIL)** nằm dưới dạng 3 dự án khác nhau và kết quả đầu ra của 3 dự án này (tệp .dll) phải cùng trong một máy chủ hoặc trên cùng một máy để hệ thống chạy.

Trong kiến ​​trúc 3 lớp DAL, BLL và UIL có thể hoạt động trên cùng một máy trong đó giống như trong kiến ​​trúc 3 Bậc mà một máy khách ở trên một máy, Máy chủ ứng dụng được lưu trữ trong một máy khác và máy chủ cơ sở dữ liệu nằm trong một máy khác tức là Ba tầng là một [kiến trúc máy khách-máy chủ](http://en.wikipedia.org/wiki/Client%E2%80%93server_architecture) trong đó [Giao diện Người dùng](http://en.wikipedia.org/wiki/User_interface) , [Logic Kinh doanh](http://en.wikipedia.org/wiki/Business_logic_layer) , [Lưu trữ](http://en.wikipedia.org/wiki/Computer_data_storage) [Dữ liệu](http://en.wikipedia.org/wiki/Data_access) và [Truy cập Dữ liệu](http://en.wikipedia.org/wiki/Data_access) được phát triển và duy trì như [các mô-đun](http://en.wikipedia.org/wiki/Module_(computing)) độc lập trên các [nền tảng](http://en.wikipedia.org/wiki/Platform_(computing)) riêng biệt .

Trong kiến ​​trúc 3 tầng, Tầng truy cập cơ sở dữ liệu **(DAL)** , Tầng logic nghiệp vụ **(BLL)** và Tầng giao diện người dùng **(UIL)** nằm dưới dạng 3 dự án khác nhau. Nhưng mỗi dự án có thể được triển khai tại các máy chủ khác nhau hoặc tại các máy khác nhau và chức năng phân tán được khám phá.

## Triển khai và xây dựng một ứng dụng 3 tầng trên c#

### 2.3.1 Tầng giao diện người dùng(UIL)

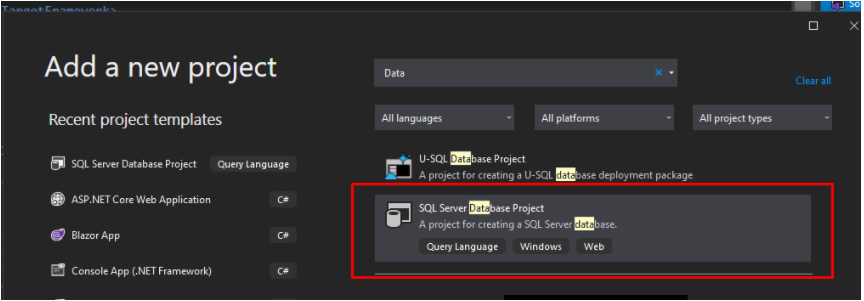
Áp dụng kiến ​​trúc 3 lớp cho tầng Ứng dụng.

### 2.3.2 Tầng Logic nghiệp vụ dự án(BLL)

Đây là trung tâm của kiến ​​trúc 3 tầng, cũng là tầng thực hiện phức tạp và khó khăn nhất, vì vậy chúng ta cần một thiết kế tốt để quản lý và tổ chức mã. Đó là lý do tại sao tôi sử dụng kiến ​​trúc 3 lớp cho tầng này.

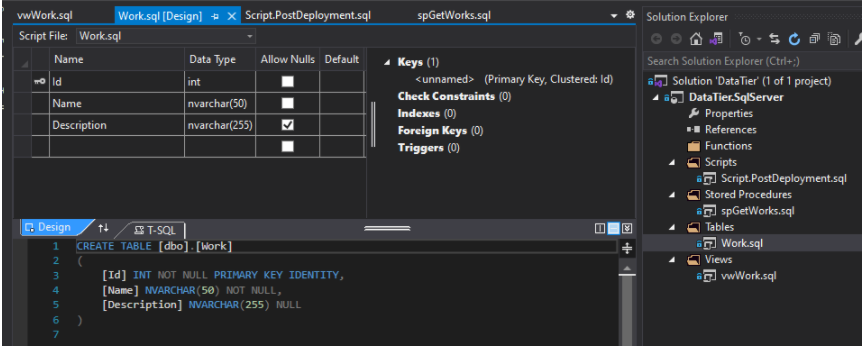
### 2.3.3 Tầng truy cập cơ sở dữ liệu (DAL)

**Tạo dự án cơ sở dữ liệu**  
Hãy sử dụng mẫu dự án cơ sở dữ liệu trong Visual Studio 2019 và tạo một dự án cơ sở dữ liệu.



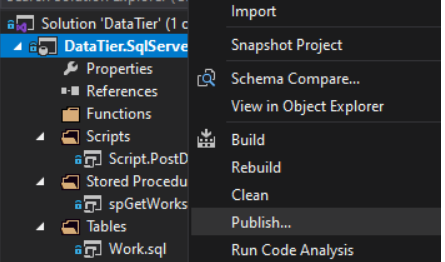
Hình 4: Tạo dự án cơ sở dữ liệu

**Thiết kế cơ sở dữ liệu**  
Sau khi dự án được tạo, hãy thiết kế các bảng, dạng xem và các thủ tục được lưu trữ của bạn cho cơ sở dữ liệu này. Ngoài ra, hãy thêm một tập lệnh để khởi tạo dữ liệu khi dự án được xuất bản lên SQL Server.



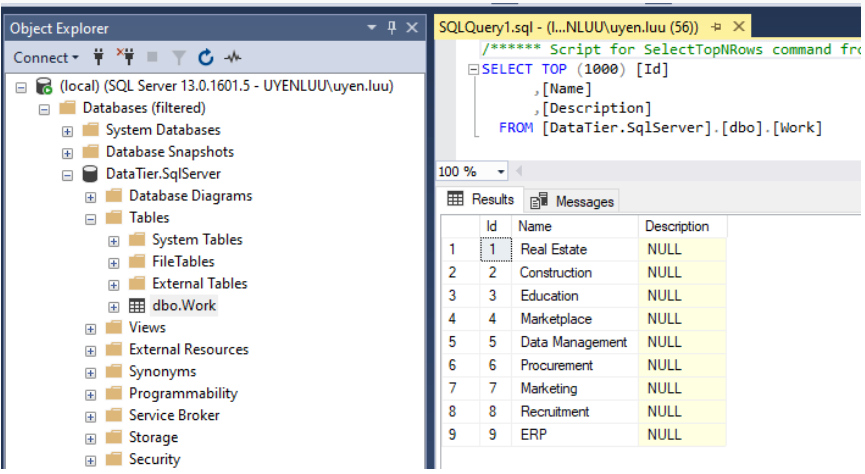
Hình 5: Thiết kế cơ sở dữ liệu

**Triển khai**  
Nhấp chuột phải vào dự án và chọn Xuất bản.



Hình 6: Triển khai sql

**Kiểm tra**  
Mở SSMS của bạn để kiểm tra kết quả.



Hình 7: Kiểm tra sql

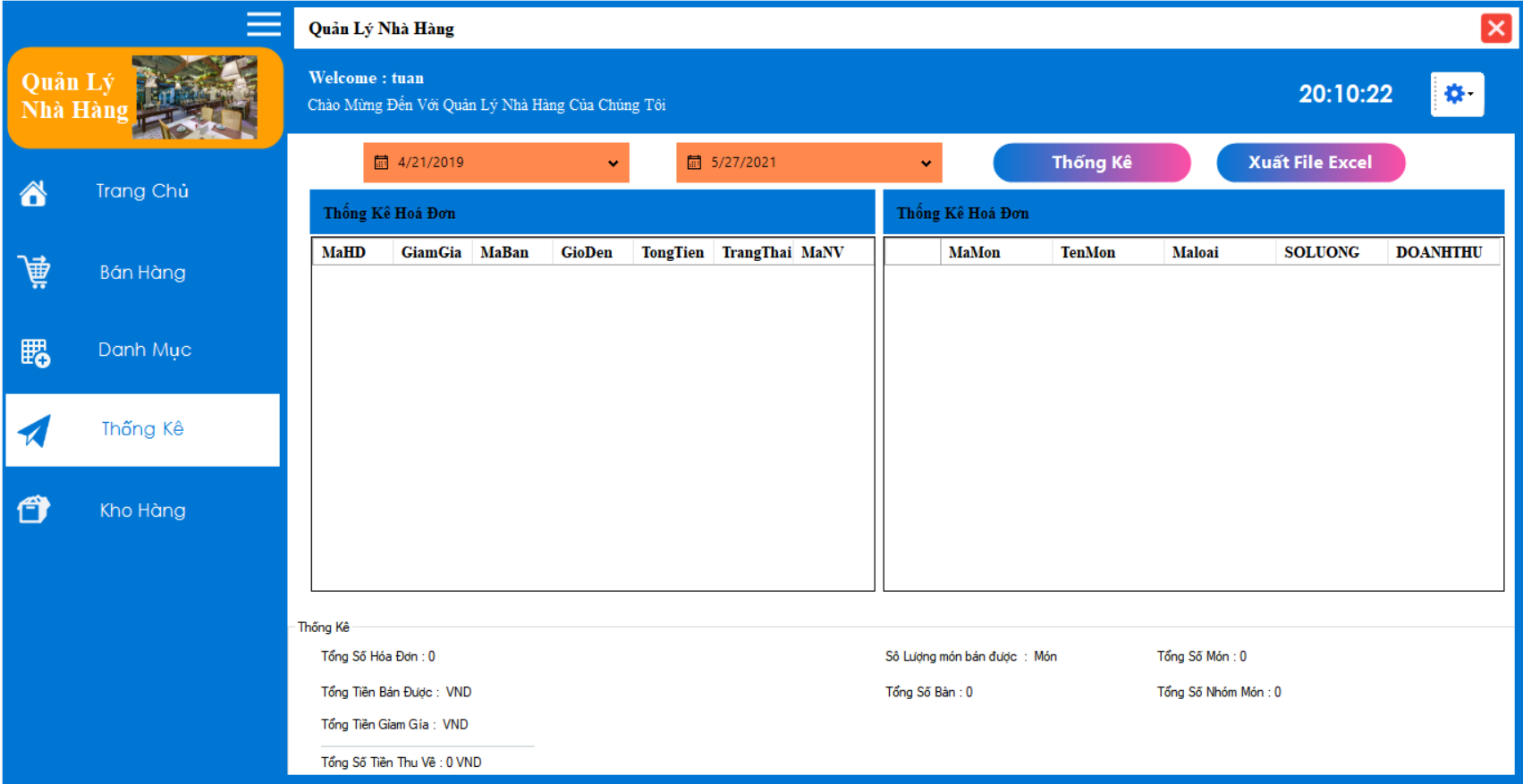
# CHƯƠNG TRÌNH DEMO

## Giao diện chính



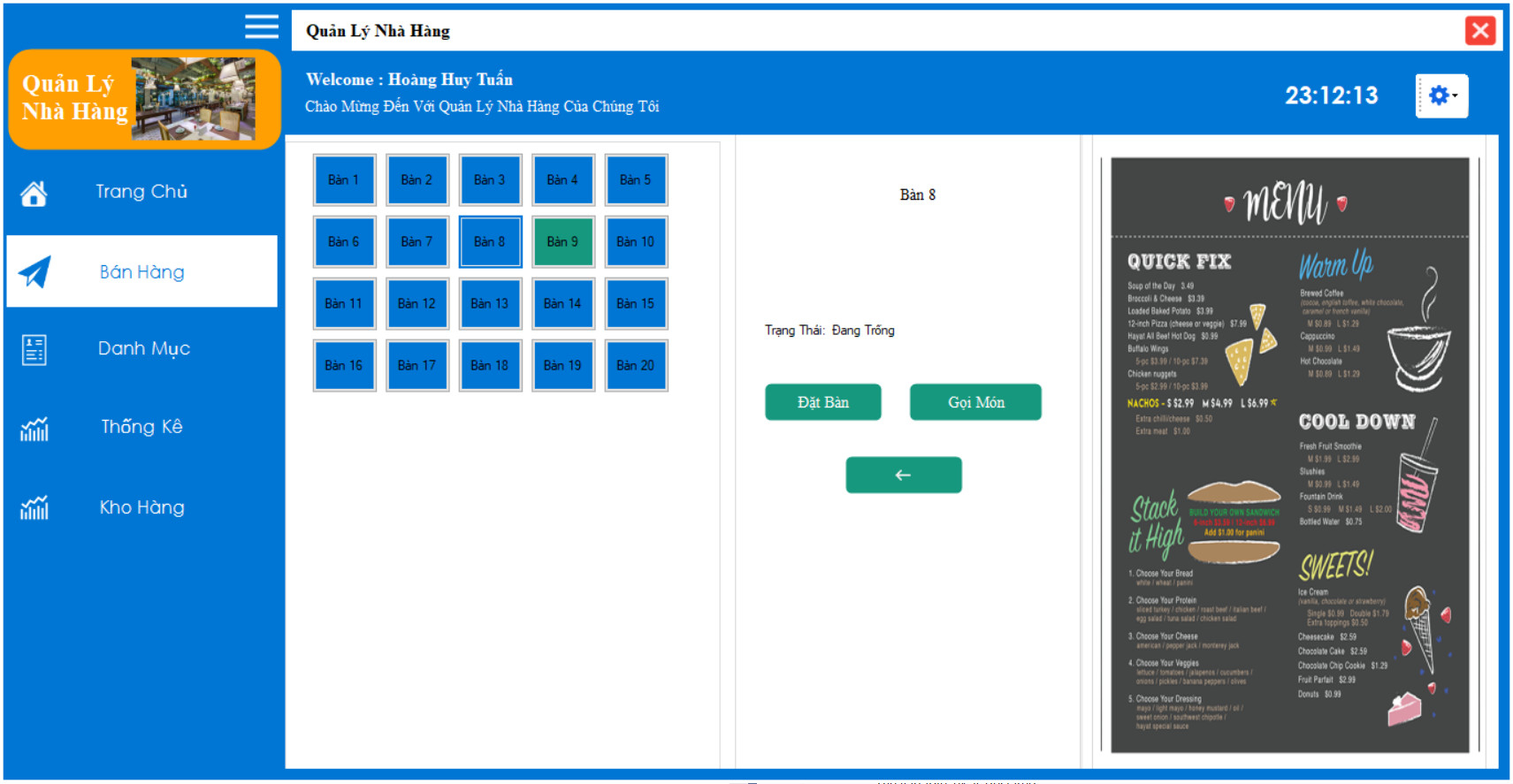
Hình 8: Giao diện chính

## Giao diện thống kê



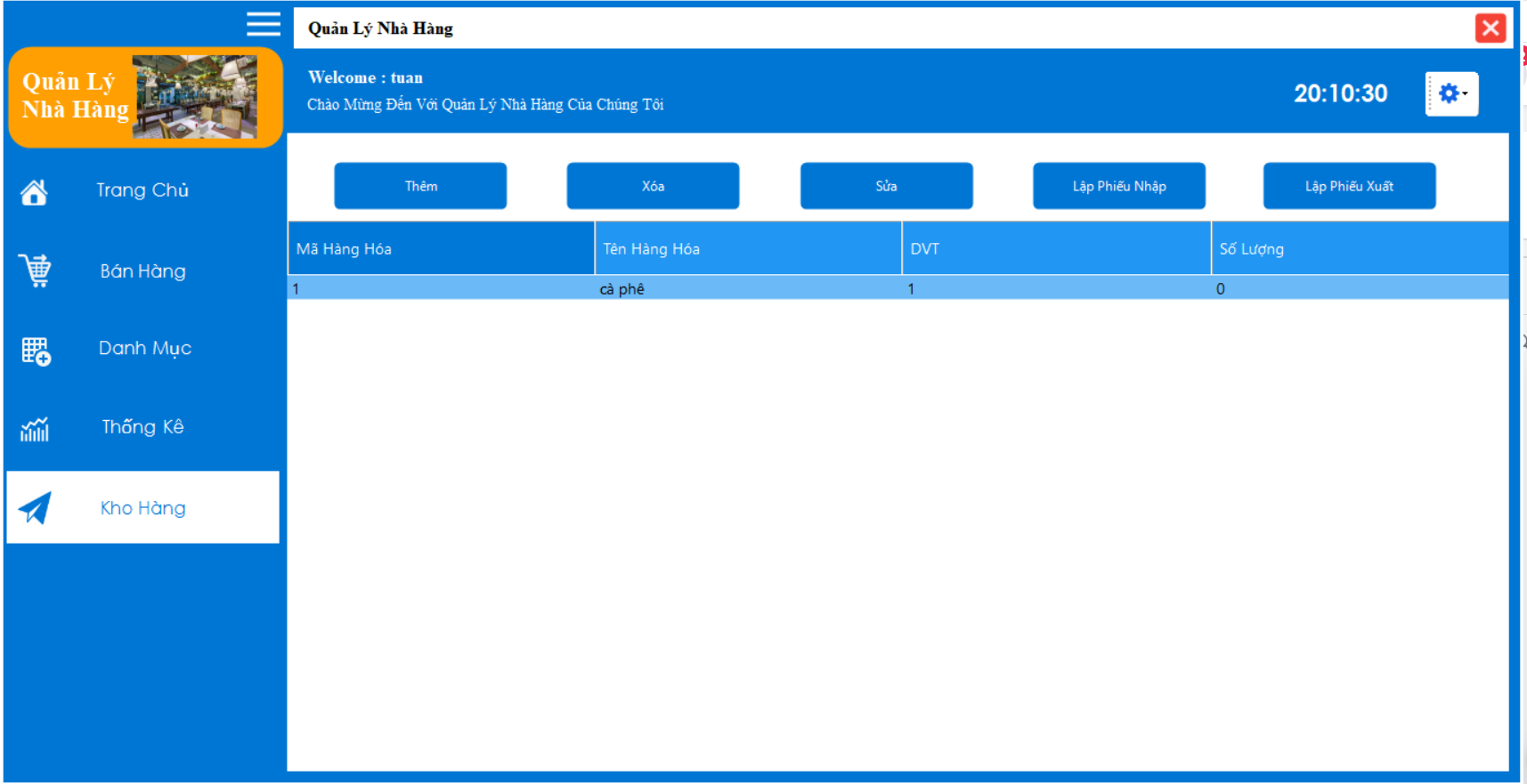
Hình 9: Giao diện thống kê

## Giao diện bán hàng



Hình 10: Giao diện quản lý bán hàng

## Giao diện kho hàng



Hình 11: Giao diện kho hàng

# KẾT LUẬN VÀ KHUYẾN NGHỊ

* **Kết quả đạt được**
* Về chuyên môn ứng dụng đã xây dựng được giao diện gần gũi, nội dung dễ hiểu, dễ sử dụng, và đảm bảo được các chức năng cơ bản .
* Hoạt động trên nhiều môi trường như: Win 7, Win 10,...
* Ứng dụng thân thiện, hài hòa, có ấn tượng đối với người xem.
* Ứng dụng giúp cho người dùng thao tác nhanh hơn trong quá trình tính toán cập nhật sản phẩm.
* **Hướng phát triển của đề tài**
* Phần mềm hướng tới việc phát triển mở rộng toàn phần như thực hiện toàn bộ online ,nâng cấp các chức năng....
* Bổ sung các hạn chế của đề tài, mở rộng các chức năng của hệ thống,đảm bảo độ tin cậy, và chính xác của hệ thống.

# TÀI LIỆU THAM KHẢO

[1] C# Programming: From Problem Analysis to Program Design 004 Edition, Kindle Edition by [Barbara Doyle](https://www.amazon.com/Barbara-Doyle/e/B003VN12TG/ref=dp_byline_cont_ebooks_1)

[2] https://en.wikipedia.org/wiki/Multitier\_architecture

[3] https://docs.microsoft.com/vi-vn/dotnet/csharp/

[4] https://enlabsoftware.com/development/how-to-build-and-deploy-a-3-layer- architecture-application-with-c-sharp-net-in-practice.html

[5] https://thongtintonghop.org/kien-truc-n-tier-architecture/

[6] https://stackify.com/n-tier-architecture/

[7] https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n- tier