|  |
| --- |
| **KHOA CÔNG NGHỆ THÔNG TIN**  **----------** |
| **ĐỒ ÁN CUỐI KỲ**  **\***  **“Đề tài 5: Du lịch - Quản lý khách sạn”**   |  |  |  | | --- | --- | --- | | **STT** | **Họ Và Tên** | **Mã Số Sinh Viên** | | **1** | **Đái Triệu Phi** | **21133068** | | **2** | **Võ Công Minh** | **21133056** | | **3** | **Trần Thị Ngọc Trang** | **21133109** |   **BỘ MÔN: LẬP TRÌNH TRÊN WINDOWS** |



**MÃ MÔN HỌC:WIPR230579\_23\_2\_10**

**LỚP: Thứ 6, tiết 2-5**

**GVHD:ThS. Nguyễn Thủy An**

**Tp. Hồ Chí Minh, tháng 05 năm 2023**



**NHẬN XÉT CỦA GIẢNG VIÊN**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

**…………………………………………………………………………......**

|  |
| --- |
| ĐIỂM (CÔNG SỐ): ................. |
| CÔNG CHỮ: ........................... |
| CHỮ KÝ GV:............................ |

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Cô Nguyễn Thủy An vì đã dành thời gian và công sức hướng dẫn và hỗ trợ chúng em trong quá trình nghiên cứu và hoàn thành đồ án về đề tài "Quản lý chuỗi khách sạn".

Cô đã không chỉ cung cấp cho chúng em kiến thức chuyên môn sâu rộng về lĩnh vực quản lý khách sạn, mà còn giúp chúng em xây dựng nền tảng vững chắc về cách thức tổ chức và điều hành các hoạt động trong ngành này. Những buổi thảo luận và hướng dẫn của cô đã là nguồn động viên lớn lao cho chúng em.

Chúng em rất biết ơn sự tận tụy và sự nhiệt tình mà Cô đã dành cho mỗi sinh viên trong lớp. Cô luôn sẵn lòng giải đáp mọi thắc mắc và hỗ trợ chúng em trong quá trình thực hiện đồ án. Phong cách giảng dạy của Cô rất chi tiết, rõ ràng và dễ hiểu, giúp chúng em tiếp cận và áp dụng kiến thức vào thực tiễn một cách hiệu quả.

Một lần nữa, chúng em xin chân thành cảm ơn Cô Nguyễn Thủy An vì sự hỗ trợ và động viên của Cô trong suốt quá trình thực hiện đồ án. Kính chúc Cô luôn mạnh khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy.

**MỤC LỤC**

[I. MỞ ĐẦU 1](#_heading=h.30j0zll)

[1. Giới thiệu đồ án 1](#_heading=h.1fob9te)

[2. Mục tiêu của đồ án 1](#_heading=h.3znysh7)

[3. Phạm vi đồ án 1](#_heading=h.2et92p0)

[II. Kinh nghiệm khi thực hiện đồ án 2](#_heading=)

[2.1. Những điều làm được 2](#_heading=h.bu3nftnlfcxp)

[2.2. Những khó khăn 2](#_heading=h.ror78qwtl01u)

[III. TỔNG QUAN VỀ ĐỒ ÁN VÀ CÁCH HOẠT ĐỘNG 4](#_heading=)

[1. Form đăng nhập và đăng ký 4](#_heading=h.3dy6vkm)

[2. Form Admin (owner) 5](#_heading=h.1t3h5sf)

[2.1. Form tổng quát 5](#_heading=h.9cel77hczjfd)

[2.2. Form Search 6](#_heading=h.qokm009uwrv1)

[2.3. Form Thêm khách sạn 7](#_heading=h.whnkyix2bpl5)

[2.4. Form Chi tiết Hotel 7](#_heading=h.4m91h7s35ndq)

[2.5. Form add room 8](#_heading=h.3mng9ob3rpsm)

[2.5 Form chi tiết Room 8](#_heading=h.fvcyuu3z6mnt)

[2.6 Form lịch sử đặt phòng 9](#_heading=h.9t3ma25zetnv)

[3. Form User 9](#_heading=h.4d34og8)

[3.1. Form search 9](#_heading=h.gozznouy1fyy)

[3.2. Form chi tiết khách sạn 10](#_heading=h.y4w4cuj0zoy5)

[3.3. Form booking 11](#_heading=h.keyk7i2fi5qx)

[3.4. Form xem lại lịch sử đặt phòng 11](#_heading=h.t8josthv8l3c)

[IV. QUÁ TRÌNH THỰC HIỆN ĐỒ ÁN 13](#_heading=h.3rdcrjn)

[Tuần 1, 2 13](#_heading=)

[Tuần 3, 4 13](#_heading=)

[Tuần 5, 6 13](#_heading=)

[Tuần 7, 8 14](#_heading=)

[Tuần 9, 10 14](#_heading=)

[Tuần 11, 12 14](#_heading=)

[Tuần 13, 14 15](#_heading=)

[Tuần 15 15](#_heading=)

[V. CÁC CÔNG NGHỆ, CÔNG CỤ TÍCH HỢP VÀO ĐỒ ÁN 16](#_heading=)

[5.1 Công nghệ 16](#_heading=)

[5.1.1. ActiveX Data Objects .NET (ADO.Net) 16](#_heading=h.9yubl987kba4)

[5.1.2. Entity Framework 17](#_heading=h.em3t5wwo85vq)

[5.1.3. Guna UI Framework 19](#_heading=h.ris1ga920ktt)

[5.1.4. Bunifu UI Framework 20](#_heading=h.ve6770buggov)

[5.2. Kỹ thuật 21](#_heading=)

[5.2.1. Object-Oriented Programming (OOP) 21](#_heading=h.sx5e1pqpwjkl)

[5.2.2. Mô hình 3 lớp three layers trong c# 22](#_heading=h.tkn3l7kdoo5s)

[5.2.3. Singleton 23](#_heading=h.87n3jl3b1f2m)

[5.2.3. Store Procedure 24](#_heading=h.p8g1et6qyrx)

BÁO CÁO ĐỒ ÁN CUỐI KỲ

# I. MỞ ĐẦU

## 1. Giới thiệu đồ án

Đồ án quản lý khách sạn này là một dự án có mục tiêu tạo ra một hệ thống tự động hóa toàn diện để quản lý các hoạt động của khách sạn. Với sự phát triển không ngừng của công nghệ và hệ thống thông tin, việc cải thiện hiệu suất và trải nghiệm của cả khách hàng và chủ khách sạn là một ưu tiên hàng đầu. Bằng cách kết hợp các công nghệ tiên tiến và mô hình quản lý hiện đại, đồ án nhằm mục đích tạo ra một nền tảng linh hoạt, dễ dàng mở rộng và tối ưu hóa quy trình quản lý của các khách sạn.

Link code ( git) : <https://github.com/TrieuPhi/Hotel_Management_Project.git>

Chạy chương trình ( username/pass or …) :

* Đăng nhập với role Chủ khách sạn: username: owner1 , pass: owner1
* Đăng nhập với role Khách hàng : username: user1 , pass: user1

## 2. Mục tiêu của đồ án

Mục tiêu chính của đồ án là xây dựng một hệ thống quản lý khách sạn hiệu quả, bao gồm các chức năng quản lý cơ bản như quản lý thông tin khách hàng, quản lý đặt phòng, quản lý dịch vụ khách sạn, thanh toán và báo cáo và thống kê. Bằng cách sử dụng các công nghệ và khung công cụ hiện đại, đồ án nhằm mục đích tối ưu hóa quy trình làm việc, tăng cường trải nghiệm người dùng và nâng cao hiệu suất của khách sạn.

## 3. Phạm vi đồ án

Phạm vi của đồ án bao gồm việc sử dụng ADO.NET để kết nối và tương tác với cơ sở dữ liệu, Guna UI Framework để thiết kế giao diện người dùng đẹp mắt và dễ sử dụng, Entity Framework để quản lý dữ liệu, và áp dụng các nguyên tắc lập trình hướng đối tượng để xây dựng một hệ thống linh hoạt và dễ bảo trì. Qua đó, đồ án mong muốn cung cấp một giải pháp toàn diện cho các khách sạn trong việc quản lý và phục vụ khách hàng.

# II. Kinh nghiệm khi thực hiện đồ án

## 2.1. Những điều làm được

* **Thiết kế giao diện trực quan:** Sử dụng C# Windows Form để tạo ra giao diện người dùng thân thiện và dễ sử dụng cho ứng dụng. (sử dụng 2 framework dành cho giao diện là: Guna UI, và Bunifu)
* **Quản lý thông tin khách sạn:** Xây dựng cơ sở dữ liệu để lưu trữ thông tin về phòng, đặt phòng, khách hàng và các dịch vụ khác của khách sạn.
* **Xử lý giao dịch:** Tạo các chức năng để xử lý đặt phòng, thanh toán và quản lý lịch sử giao dịch của khách hàng.
* **Tối ưu hóa hiệu suất:** Tối ưu hóa code, xây dựng theo kiến trúc OOP, mô hình 3 lớp, và kiến trúc singleton. Để đảm bảo ứng dụng chạy mượt mà và dễ dàng bảo trì code.
* **Bảo mật thông tin khi đăng nhập:** sử dụng Store Procedure khi tài khoản người dùng đăng nhập, để hạn chế lỗi SQL injection.

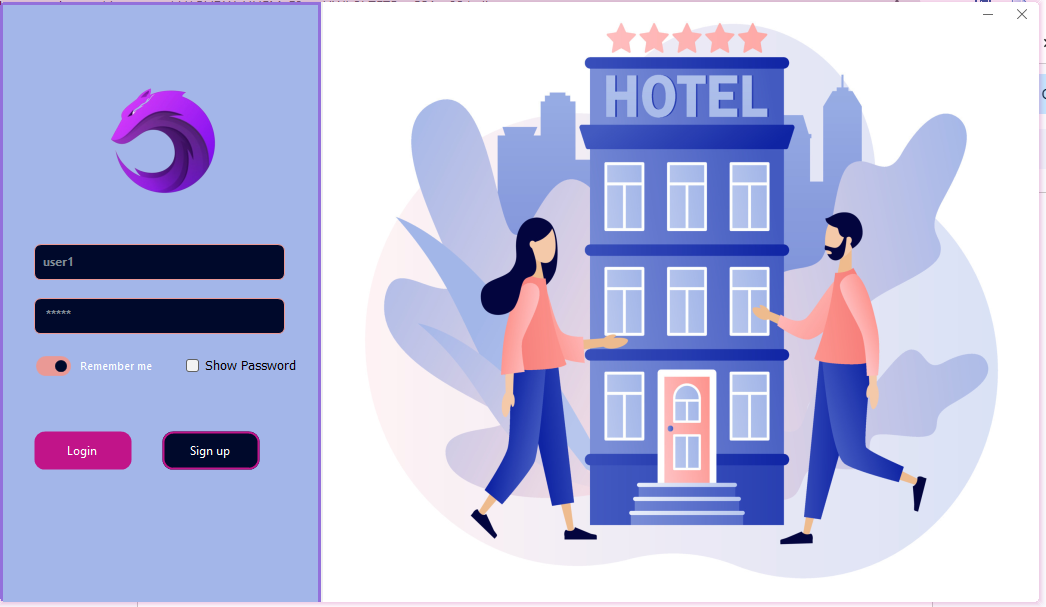
## 2.2. Những khó khăn

* Khó khăn đầu tiên ở việc thiết kế giao diện, những framework hỗ trợ thiết kế giao diện như Guna UI, và Bunifu framework, cần có bản quyền, hoặc tài khoản license để kích hoạt và sử dụng.
* Việc đồng bộ giao diện giữa các thành viên cũng sẽ gặp khó khăn, nếu như kích thước màn hình của mỗi thành viên là khác nhau( 14 inch, hoặc 15.6 inch) cho nên cần lưu ý thống như kích thước của bộ khung giao diện từ ban đầu, để giao diện được thể hiện trực quan nhất trên tất cả máy tính.
* Việc sử dụng git và github vào đồ án, cần lưu ý trong quá trình sử dụng git đối với visual studio code. Việc các log file sẽ được tạo ra mỗi khi visual studio code khởi động, vì vậy để bảo đảm quá trình đóng gói code hoàn chỉnh trước khi push hoặc pull, cần phải đóng visual studio. Hoặc có thể dụng git ignore để đưa folder (obj), vì mỗi máy tính sẽ khó folder(obj) khác nhau.
* Xử lý logic phức tạp: Quản lý các quy trình phức tạp như đặt phòng, hủy phòng, tính toán giá cả và lập hóa đơn có thể gặp phải nhiều thách thức. Cần phải thiết kế và triển khai logic phù hợp để đảm bảo tính đúng đắn và hiệu quả của hệ thống.
* Chuyển từ ADO sang Entity Framework gặp khó khăn trong việc hiểu cách ánh xạ dữ liệu và thay đổi cách thức truy cập dữ liệu từ SQL sang LINQ.

# III. TỔNG QUAN VỀ ĐỒ ÁN VÀ CÁCH HOẠT ĐỘNG

## 1. Form đăng nhập và đăng ký

* Form đăng nhập

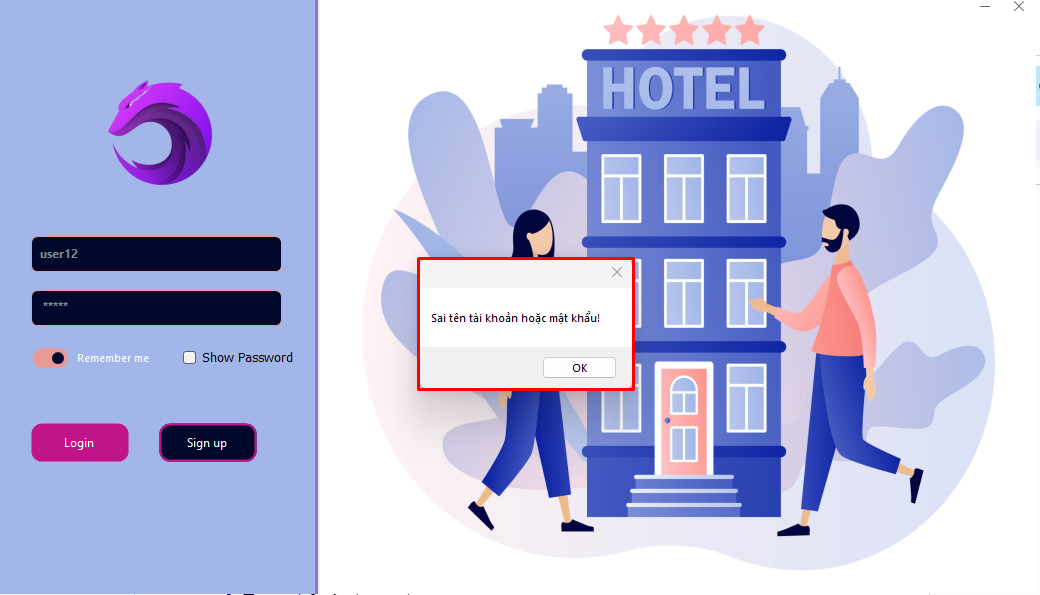


Bước 1: Nhập tài khoản và mật khẩu.

Bước 2: Nhấn chọn “Login”

Khi người dùng đăng nhập thành công sẽ được đưa đến màn hình chính theo từng loại tài khoản.

Đăng nhập thất bại:

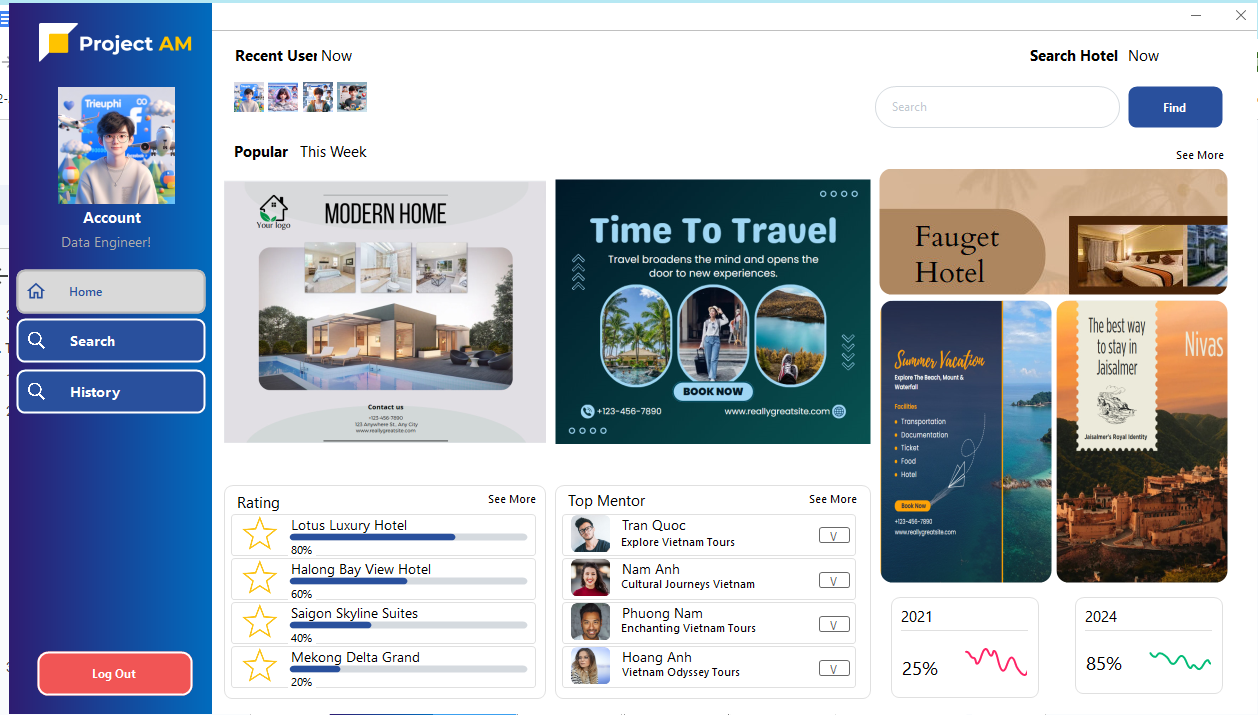


* Form đăng kí

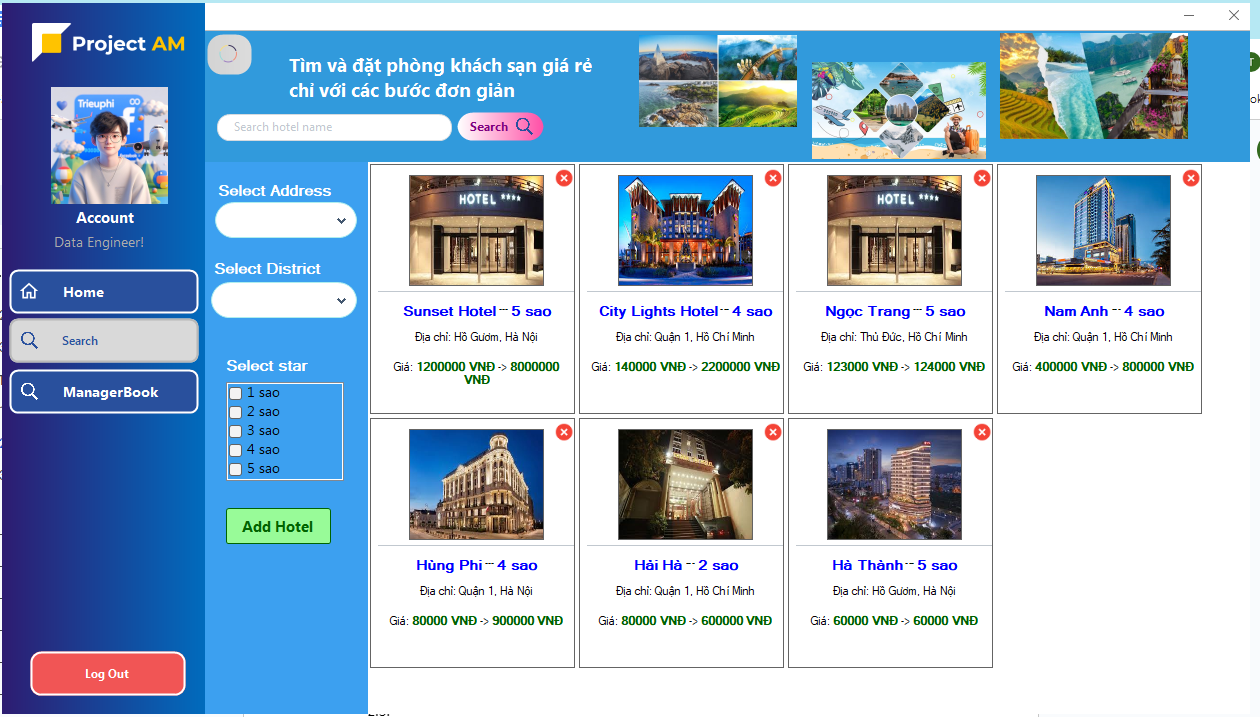


## 2. Form Admin (owner)

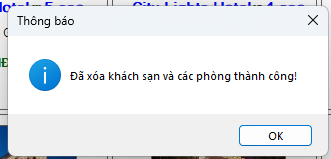
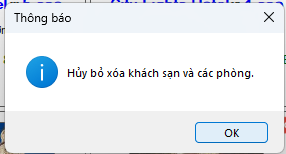
### 2.1. Form tổng quát



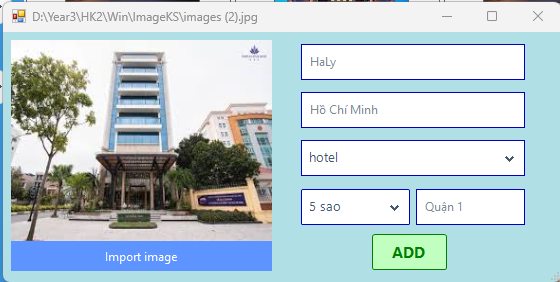
### 2.2. Form Search



Ở mục này chủ khách sạn có thể:

* Thêm khách sạn bằng cách nhấn vào nút “Add Hotel”
* Xóa khách sạn nhấn “X”, Thông báo xác nhận:
* 
* Chọn “OK” để xác nhận
* 
* Chọn Cancel để Hủy
* 

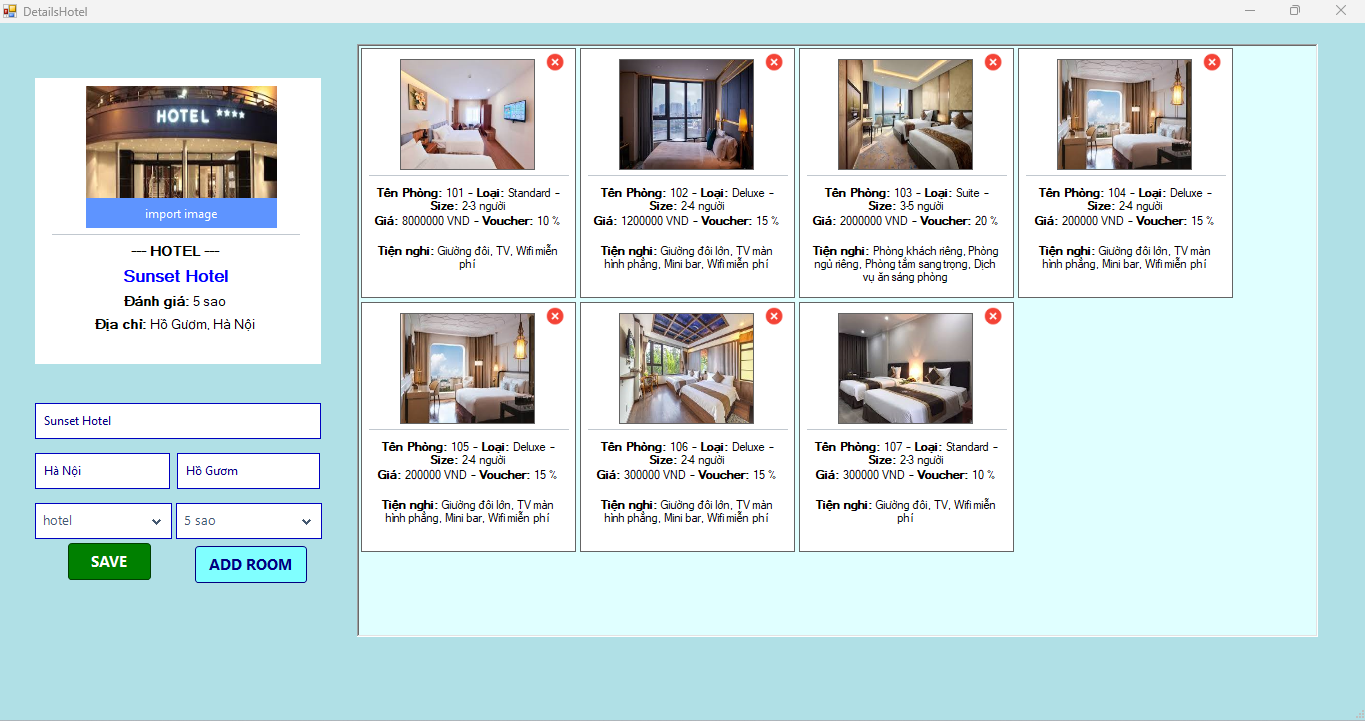
### 2.3. Form Thêm khách sạn



Thêm khách sạn (tên, địa chỉ, …) thuộc loại: homestay, hotel

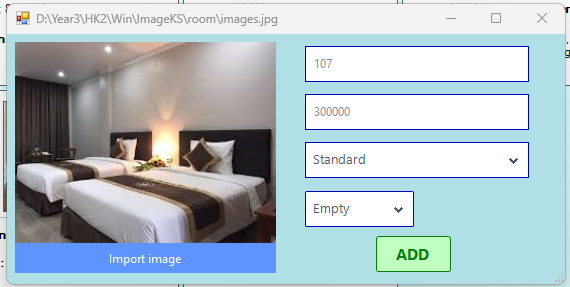
* B1: điền đầy đủ thông tin khách sạn
* B2: Nhấn “ADD”

### 2.4. Form Chi tiết Hotel



* Admin:
* Chỉnh sửa thông tin khách sạn
* Thêm, Xóa và chỉnh sửa khách sạn

### 2.5. Form add room



Thêm phòng cho khách sạn(giá, tiện ích, …)

B1: Điền đầy đủ thông tin khách sạn

B2: Nhấn “ADD”

### 2.6. Form chi tiết Room



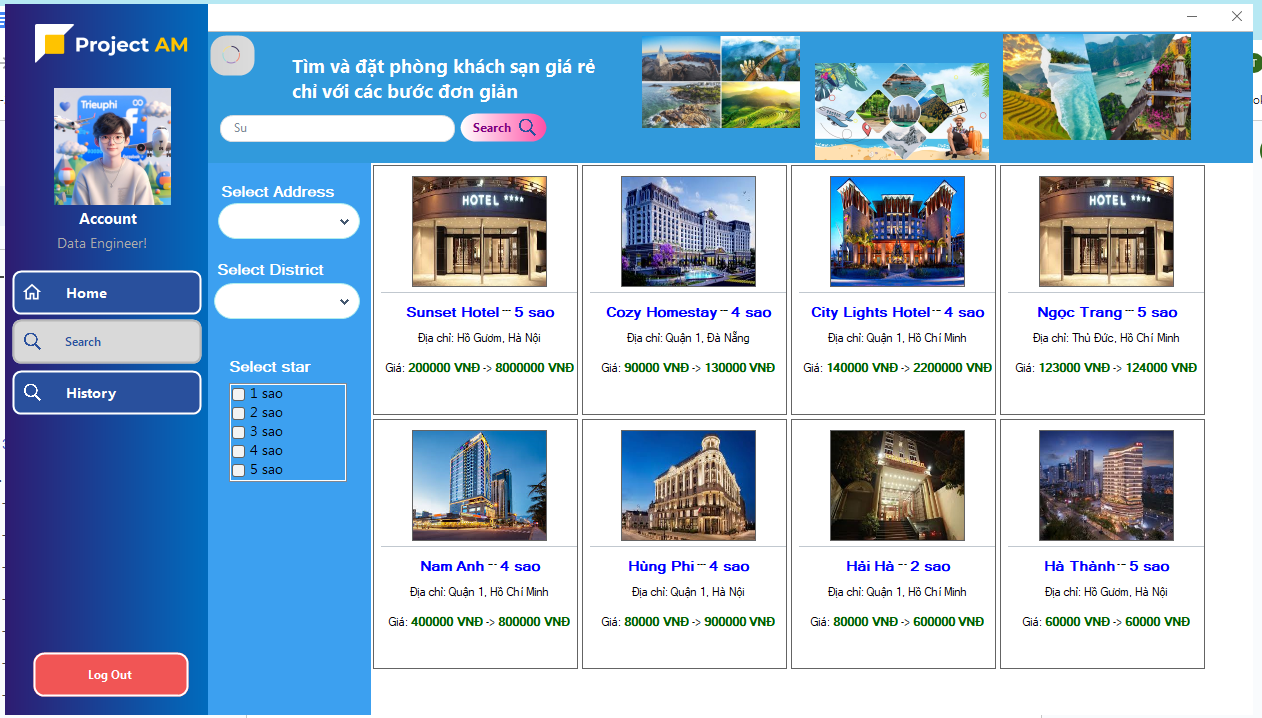
### 2.7. Form lịch sử đặt phòng



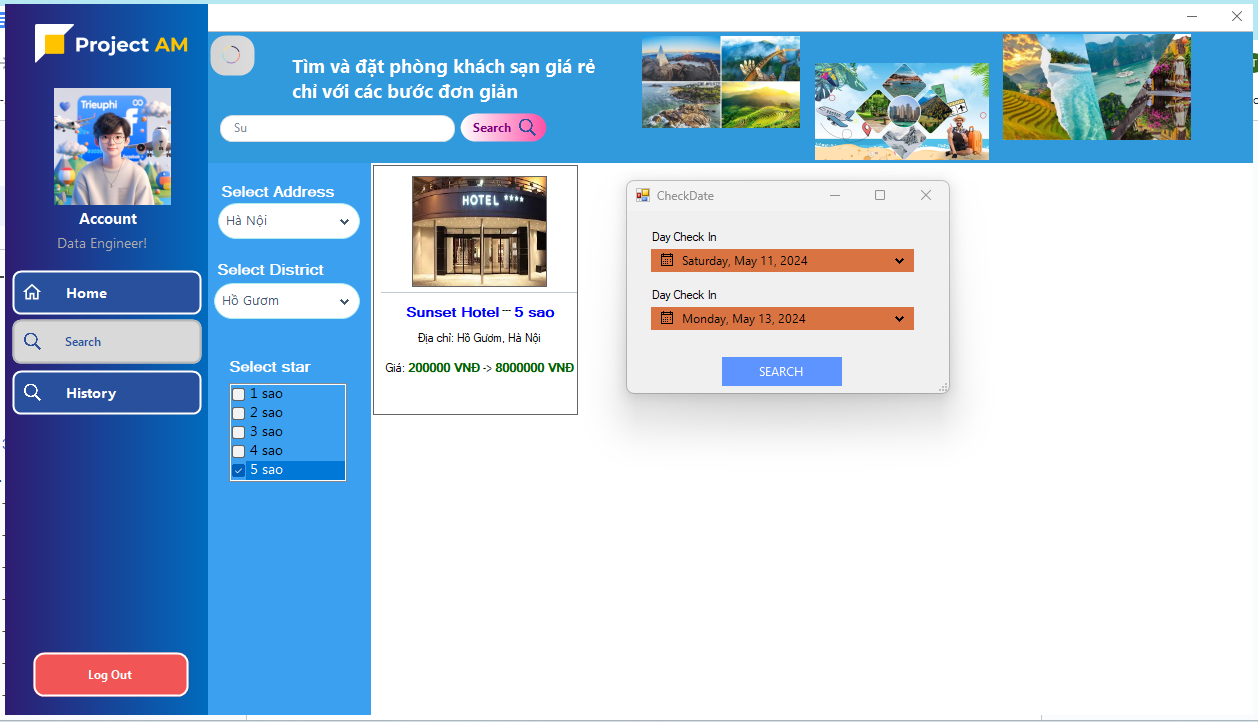
* Form này có chức năng:
* Xem lại Danh sách các phòng của các khách sạn đang được đặt
* Xem lại Danh sách các phòng của các khách sạn đang được đặt và lưu trú thành công
* Xem Doanh Thu: Theo tháng, theo quý và theo năm

## 3. Form User

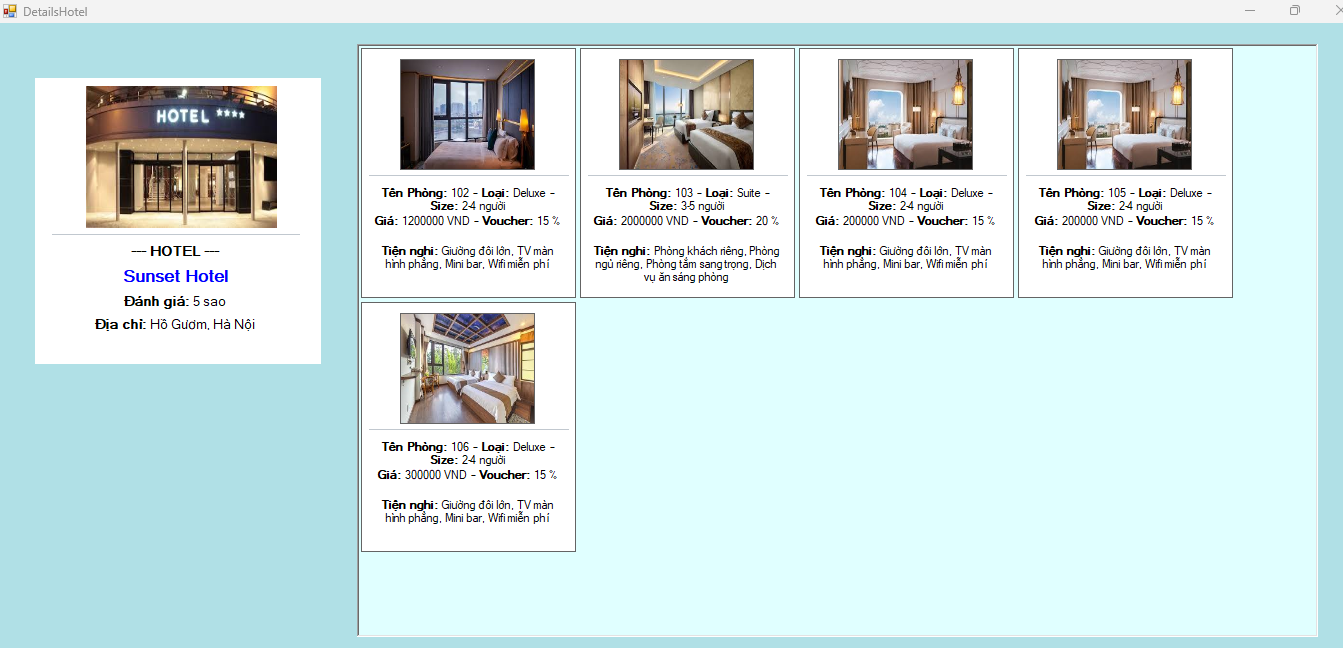
### 3.1. Form search



* Hiển Thị tên khách sạn, Địa chỉ, số sao và khoản giá phòng
* Form này thực hiện các chức năng:
* Search Khách sạn theo: Tên, Thành Phố , Quận-Huyện, Số sao và Ngày checkin-checkout

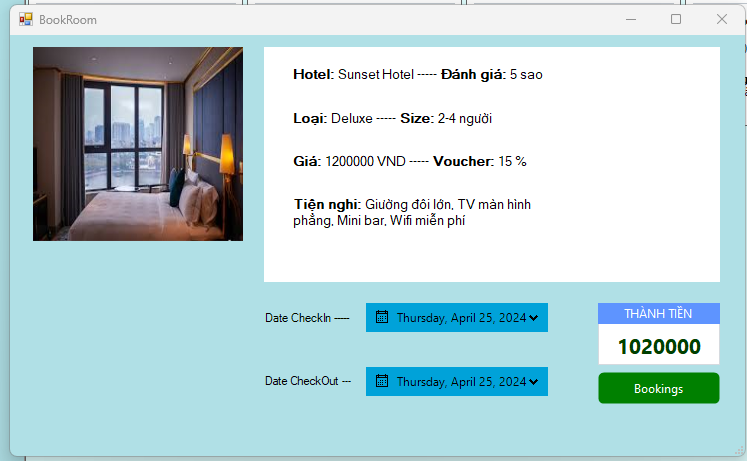


### 3.2. Form chi tiết khách sạn



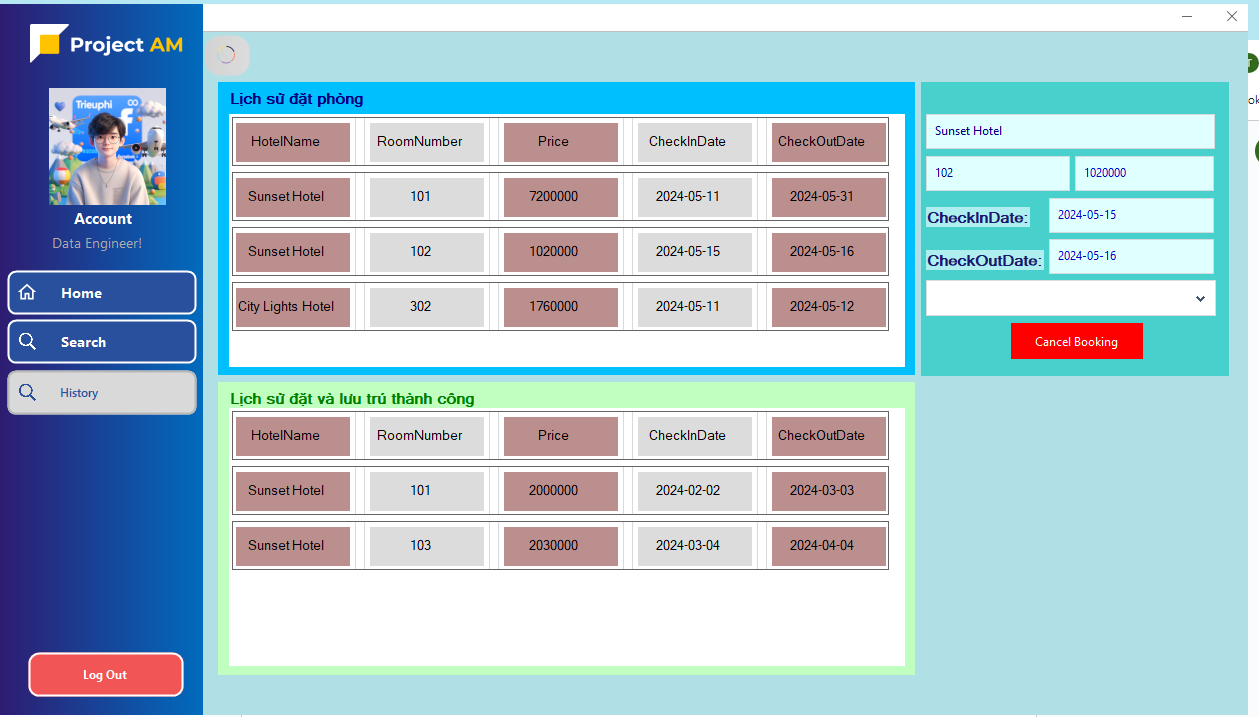
* Khách hàng có thể xem thông tin chi tiết khách sạn: Thông tin khách sạn và thông tin phòng

### 3.3. Form booking



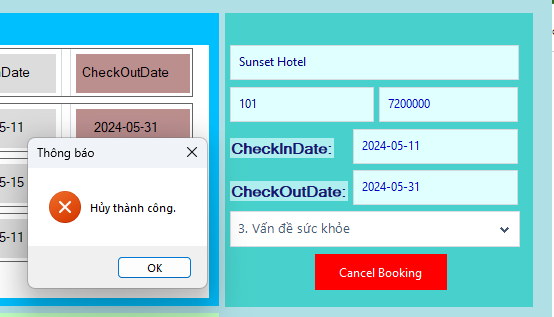
Chức năng: Chọn ngày checkin-checkout và booking

### 3.4. Form xem lại lịch sử đặt phòng

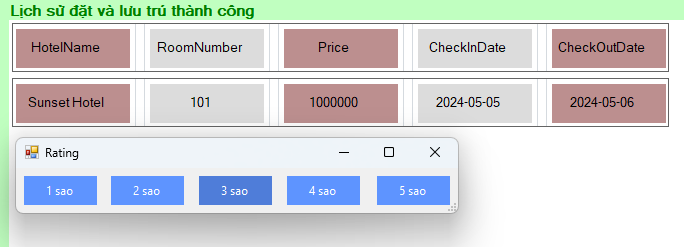


Chức năng form này:

* Xem lại lịch sử phòng đã đặt
* Xem lại lịch sử phòng đặt và lưu trú thành công
* Hủy Phòng đã đặt + Lý do hủy



* Đánh giá Khách sạn



# IV. QUÁ TRÌNH THỰC HIỆN ĐỒ ÁN

## Tuần 1, 2

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Ôn tập kiến thức OOP | 100% |
| Đái Triệu Phi | Ôn tập kiến thức OOP | 100% |
| Trần Thị Ngọc Trang | Ôn tập kiến thức OOP | 100% |

## Tuần 3, 4

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Tìm hiểu về giao diện window form | 100% |
| Đái Triệu Phi | Tìm hiểu về giao diện window form | 100% |
| Trần Thị Ngọc Trang | Tìm hiểu về giao diện window form | 100% |

## Tuần 5, 6

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Xây dựng thiết kế giao diện và cơ sở dữ liệu | 100% |
| Đái Triệu Phi | Thiết kế cơ sở dữ liệu và triển khai | 100% |
| Trần Thị Ngọc Trang | Xây dựng thiết kế giao diện và môi trường phát triển | 100% |

## Tuần 7, 8

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Quản lý khách hàng và đặt phòng |  |
| Đái Triệu Phi | Phát triển chức năng đăng nhập và đăng ký |  |
| Trần Thị Ngọc Trang | Kiểm thử và sửa lỗi |  |

## Tuần 9, 10

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Phát triển chức năng đặt phòng và tính giá |  |
| Đái Triệu Phi | Sử dụng flow panel, và kiến trúc singleton. |  |
| Trần Thị Ngọc Trang | Quản lý dịch vụ đi kèm và tính năng đánh giá |  |

## Tuần 11, 12

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Chuyển sang code Entity framework. |  |
| Đái Triệu Phi | Tiếp tục hoàn thiện chức năng của ADO. |  |
| Trần Thị Ngọc Trang | Quản lý thông tin khách sạn và phòng |  |

## Tuần 13, 14

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Hoàn thiện chức năng đặt phòng, lịch sử đặt phòng, owner được quyền thêm khách sạn và thêm phòng ở bản Entity. |  |
| Đái Triệu Phi | Hoàn thiện chức năng đặt phòng khách sạn và thêm phòng ở bản ADO. |  |
| Trần Thị Ngọc Trang | Phụ Công Minh làm các chức năng bên Entity. |  |

## Tuần 15

|  |  |  |
| --- | --- | --- |
| Sinh viên | **Nội dung thực hiện** | **Tiến độ** |
| Võ Công Minh | Bổ sung thêm chức năng đánh giá sau khi checkout, thêm bộ lọc để xem khách sạn nào còn phòng trống. Thống kê doanh thu theo tháng theo quý theo năm đối với owner. Chức năng booking được tối ưu hơn. |  |
| Đái Triệu Phi | Hoàn thiện theo các chức năng của entity. |  |
| Trần Thị Ngọc Trang | Thử nghiệm và kiểm tra tính năng của ứng dụng. |  |

# V. CÁC CÔNG NGHỆ, CÔNG CỤ TÍCH HỢP VÀO ĐỒ ÁN

## 5.1 Công nghệ

### 5.1.1. ActiveX Data Objects .NET (ADO.Net)

ActiveX Data Objects .NET (ADO.Net) là một công nghệ được sử dụng trong môi trường .NET Framework để làm việc với dữ liệu trong các ứng dụng Windows. Nó cung cấp các thành phần và lớp để kết nối, truy vấn và xử lý dữ liệu từ các nguồn dữ liệu khác nhau như cơ sở dữ liệu SQL Server, Oracle, MySQL và nhiều nguồn dữ liệu khác.

Ưu điểm khi thiết kế một phần mềm windows:

* Hiệu suất cao: ADO.NET được tối ưu hóa để đạt hiệu suất cao khi làm việc với dữ liệu, sử dụng các đối tượng như SqlConnection, SqlDataAdapter và SqlDataReader. Điều này đảm bảo ứng dụng hoạt động mượt mà và đáp ứng nhanh chóng với dữ liệu.
* Đa dạng nguồn dữ liệu: ADO.NET hỗ trợ làm việc với nhiều nguồn dữ liệu khác nhau, từ cơ sở dữ liệu quan hệ đến cơ sở dữ liệu XML và các nguồn dữ liệu khác. Điều này giúp phát triển ứng dụng Windows tương tác với các nguồn dữ liệu khác nhau một cách linh hoạt.
* Tích hợp với .NET Framework: ADO.NET tích hợp chặt chẽ với .NET Framework, giúp tối ưu hóa quy trình phát triển và tăng cường tính nhất quán và dễ dàng quản lý mã nguồn.

Nhược điểm:

* Khó hiểu và phức tạp: ADO.NET có cấu trúc phức tạp và yêu cầu kiến thức kỹ thuật cao để làm việc với nó, đặc biệt là kiến thức về SQL. Điều này có thể tăng thời gian và công sức để phát triển và bảo trì ứng dụng.
* Cần viết mã thủ công: Trong ADO.NET, bạn phải viết mã thủ công để thực hiện các tác vụ như kết nối cơ sở dữ liệu và xử lý dữ liệu, so với các công nghệ ORM như Entity Framework.
* Khó khăn trong bảo trì và mở rộng: Việc bảo trì và mở rộng có thể gặp khó khăn với ADO.NET do mã thường phải được viết thủ công.
* Thiếu tính năng cao cấp: ADO.NET tập trung vào cung cấp các đối tượng cơ bản để truy cập và xử lý dữ liệu, nhưng thiếu một số tính năng cao cấp như ánh xạ đối tượng.
* Giới hạn đa nền tảng: Ứng dụng sử dụng ADO.NET chỉ hỗ trợ hệ điều hành Windows, giới hạn khả năng đa nền tảng của ứng dụng.

### 5.1.2. Entity Framework

Entity Framework là một công nghệ giao tiếp dữ liệu trong môi trường .NET Framework, giúp phát triển ứng dụng Windows dễ dàng làm việc với cơ sở dữ liệu. Nó cung cấp mô hình đối tượng-được-mapping (ORM) cho phép truy cập và thao tác dữ liệu qua các đối tượng trong mã .NET, giảm sự phụ thuộc vào ngôn ngữ truy vấn SQL.

Ưu điểm khi thiết kế một phần mềm Windows sử dụng Entity Framework:

* Tính linh hoạt và tự động hóa: Entity Framework giúp giảm khối lượng công việc bằng cách tự động sinh mã SQL từ các truy vấn LINQ hoặc từ mô hình đối tượng. Nó cung cấp các tính năng như ánh xạ đối tượng, quản lý mối quan hệ và tự động tạo và cập nhật cơ sở dữ liệu dựa trên thay đổi trong mô hình.
* Tích hợp với .NET Framework: Entity Framework được tích hợp chặt chẽ với .NET Framework, cung cấp các lớp và API cho phép làm việc với dữ liệu trong các ứng dụng Windows. Điều này tận dụng các lợi ích của .NET Framework như tính nhất quán, quản lý bộ nhớ và khả năng triển khai trên nhiều nền tảng Windows.
* Khả năng đa nền tảng: Entity Framework hỗ trợ đa nền tảng, cho phép ứng dụng Windows có thể chạy trên nhiều hệ điều hành khác nhau, từ Windows đến macOS và Linux. Điều này giúp mở rộng khả năng đa nền tảng và đáp ứng được yêu cầu của người dùng sử dụng các nền tảng khác nhau.
* Tính bảo mật và kiểm soát dữ liệu: Entity Framework cung cấp các tính năng bảo mật như cơ chế kiểm soát truy cập và mã hóa dữ liệu. Nó cho phép xác thực và ủy quyền người dùng để kiểm soát quyền truy cập vào dữ liệu và đảm bảo tính bảo mật của hệ thống.

Nhược điểm khi thiết kế một phần mềm Windows sử dụng Entity Framework:

* Hiệu suất chậm hơn: Một nhược điểm của Entity Framework là hiệu suất có thể chậm hơn so với việc sử dụng trực tiếp SQL hoặc ADO.NET thủ công. Điều này xuất phát từ việc Entity Framework phải thực hiện các quy trình mapping và quản lý trạng thái đối tượng, gây tốn thời gian và tài nguyên.
* Độ phức tạp cao: Entity Framework có một học thuật cao và yêu cầu hiểu rõ về cấu trúc và quy tắc hoạt động của nó. Việc làm quen và sử dụng đúng các khái niệm, cấu trúc và quy ước trong Entity Framework có thể đòi hỏi một thời gian học tập và tìm hiểu đáng kể.
* Giới hạn tùy chỉnh: Trong một số trường hợp đặc biệt, có thể gặp khó khăn trong việc tùy chỉnh và điều chỉnh cách thức hoạt động của Entity Framework. Điều này có thể xảy ra khi muốn triển khai các logic phức tạp hoặc tùy chỉnh sâu hơn trong việc tạo câu truy vấn hoặc xử lý dữ liệu.
* Phụ thuộc vào phiên bản Entity Framework: Khi sử dụng Entity Framework, cần quan tâm đến phiên bản sử dụng, vì mỗi phiên bản có thể có các tính năng và cách thức hoạt động khác nhau. Việc nâng cấp hoặc di chuyển ứng dụng lên phiên bản mới có thể gây ra sự không tương thích và yêu cầu sửa đổi trong mã nguồn.

### 5.1.3. Guna UI Framework

Guna là một thư viện giao diện người dùng (UI) được phát triển cho nền tảng .NET, cung cấp các thành phần giao diện đồ họa đẹp mắt và dễ sử dụng cho các ứng dụng Windows. Thư viện này được thiết kế để giúp nhà phát triển tạo ra giao diện người dùng chuyên nghiệp và linh hoạt mà không cần phải tốn quá nhiều thời gian và công sức trong việc thiết kế từ đầu.

Ưu điểm của Guna UI Framework:

* Giao diện người dùng đẹp mắt: Guna cung cấp các thành phần giao diện người dùng (UI) thiết kế một cách chuyên nghiệp và hiện đại, giúp ứng dụng có vẻ ngoài hấp dẫn và chuyên nghiệp.
* Dễ sử dụng và tích hợp: Guna cung cấp các thành phần giao diện đồ họa dễ sử dụng và dễ tích hợp vào các ứng dụng .NET Framework, giúp nhà phát triển tiết kiệm thời gian và công sức trong việc phát triển giao diện.
* Tính linh hoạt và tùy chỉnh: Guna cho phép tùy chỉnh và điều chỉnh các thành phần giao diện để phù hợp với nhu cầu cụ thể của ứng dụng, từ việc thay đổi màu sắc cho đến định dạng và kiểu hiển thị.
* Cộng đồng và tài liệu phong phú: Có một cộng đồng lớn và hoạt động tích cực xung quanh Guna UI Framework, cung cấp hỗ trợ và tài liệu phong phú cho nhà phát triển.

Nhược điểm của Guna UI Framework:

* Giới hạn về tính linh hoạt: Mặc dù Guna cung cấp nhiều tính năng và tùy chọn tùy chỉnh, nhưng vẫn có thể có giới hạn trong việc đáp ứng một số yêu cầu cụ thể hoặc thiết kế giao diện phức tạp.
* Hiệu suất và tài nguyên: Các thành phần giao diện đẹp mắt có thể yêu cầu nhiều tài nguyên hệ thống hơn, đặc biệt là khi áp dụng hiệu ứng và đồ họa phức tạp, có thể làm giảm hiệu suất của ứng dụng.
* Phụ thuộc vào bản quyền: Một số tính năng của Guna có thể yêu cầu mua bản quyền hoặc sử dụng phiên bản cao cấp của thư viện, điều này có thể là hạn chế đối với các dự án có ngân sách hạn chế.
* Khả năng tương thích và hỗ trợ: Dù có một cộng đồng tích cực, nhưng việc tương thích với các phiên bản mới của .NET Framework và hỗ trợ kỹ thuật có thể không được cập nhật đầy đủ hoặc nhanh chóng.

### 5.1.4. Bunifu UI Framework

Bunifu UI Framework là một bộ công cụ phát triển giao diện người dùng (UI) cho ứng dụng Windows Forms và WPF trong lập trình .NET. Dưới đây là một số ưu điểm và nhược điểm của Bunifu UI Framework:

Ưu điểm:

* Giao diện đẹp mắt: Bunifu UI Framework cung cấp các thành phần giao diện người dùng được thiết kế đẹp mắt, hiện đại và chuyên nghiệp, giúp ứng dụng của bạn trở nên hấp dẫn hơn đối với người dùng.
* Dễ sử dụng: Bunifu cung cấp một bộ công cụ dễ sử dụng cho việc thiết kế giao diện người dùng, giúp lập trình viên tạo ra các ứng dụng một cách nhanh chóng và dễ dàng.
* Hỗ trợ tùy chỉnh: Bunifu UI Framework cho phép bạn tùy chỉnh các thành phần giao diện người dùng một cách linh hoạt, từ màu sắc cho đến hình dạng và kiểu dáng, giúp bạn tạo ra giao diện độc đáo và phù hợp với nhu cầu cụ thể của ứng dụng.
* Tích hợp dễ dàng: Bunifu UI Framework tích hợp tốt với các công cụ phát triển .NET như Visual Studio, giúp việc tích hợp và sử dụng các thành phần trong dự án của bạn trở nên thuận tiện.

Nhược điểm:

* Giá cả: Bunifu UI Framework có thể được xem là đắt đỏ đối với một số lập trình viên, đặc biệt là đối với các dự án nhỏ hoặc cá nhân. Việc phải mua giấy phép có thể tăng chi phí phát triển.
* Tính linh hoạt giảm: Mặc dù Bunifu cung cấp một loạt các thành phần giao diện người dùng đẹp mắt, nhưng có thể thiếu tính linh hoạt so với việc tạo ra các giao diện hoàn toàn tùy chỉnh từ đầu bằng cách sử dụng các thư viện và công cụ khác.

## 5.2. Kỹ thuật

### 5.2.1. Object-Oriented Programming (OOP)

OOP (Object-Oriented Programming) là một phương pháp lập trình mà tập trung vào việc xây dựng các đối tượng, mô hình hóa thế giới thực vào trong mã nguồn. Trong OOP, các đối tượng được tạo ra từ các lớp, có khả năng tương tác với nhau thông qua việc truyền tin nhắn và thực hiện các phương thức.

Ưu điểm của OOP (Object-Oriented Programming):

* Tính mô-đun hóa: OOP cho phép mô-đun hóa mã nguồn thành các đối tượng độc lập, giúp tạo ra mã dễ bảo trì, tái sử dụng và mở rộng. Các đối tượng có thể được tạo ra và sử dụng lại trong các phần mềm khác nhau, giúp tiết kiệm thời gian và công sức phát triển.
* Tính kế thừa và đa hình: OOP cung cấp tính năng kế thừa và đa hình, giúp giảm sự lặp lại mã nguồn và tăng tính linh hoạt trong việc thiết kế phần mềm. Lớp con có thể kế thừa các thuộc tính và phương thức từ lớp cha và tuỳ chỉnh hoặc mở rộng chúng.
* Tính trừu tượng: OOP cho phép tạo ra các lớp trừu tượng, chỉ định giao diện chung cho các đối tượng và giới hạn truy cập vào các thành phần nội bộ. Điều này giúp tách biệt phần hiện thực và phần giao diện của phần mềm, giúp việc phát triển và bảo trì dễ dàng hơn.
* Quản lý dễ dàng: OOP giúp quản lý mã nguồn dễ dàng hơn bằng cách chia thành các đối tượng, giúp tăng tính tổ chức và đơn giản hóa quá trình phát triển. Các đối tượng có thể được quản lý độc lập và tái sử dụng, giúp nâng cao hiệu suất và khả năng mở rộng của phần mềm.

Nhược điểm của OOP:

* Đòi hỏi kiến thức cao: OOP là một phương pháp lập trình phức tạp, đòi hỏi người phát triển phải có kiến thức chuyên sâu về nguyên lý và quy tắc của OOP. Việc hiểu và áp dụng các khái niệm như kế thừa, đa hình, trừu tượng và gói gọn yêu cầu một quá trình học tập và nắm vững.
* Phức tạp khi dự án lớn: Khi dự án có quy mô lớn, việc sử dụng OOP có thể làm cho cấu trúc và quản lý mã nguồn trở nên phức tạp hơn. Việc xác định và thiết kế các lớp, quan hệ giữa chúng, và quản lý sự tương tác giữa các đối tượng có thể trở thành một nhiệm vụ khó khăn và tốn thời gian.
* Khó khăn trong việc xác định các lớp và mối quan hệ: Trong OOP, việc xác định các lớp và quan hệ giữa chúng là một phần quan trọng trong thiết kế phần mềm. Việc thiếu sự chính xác hoặc đánh giá sai về các đối tượng và quan hệ có thể dẫn đến cấu trúc phần mềm không linh hoạt, khó mở rộng và khó bảo trì.
* Hiệu suất yếu trong một số trường hợp: Trong một số trường hợp, việc sử dụng OOP có thể làm giảm hiệu suất của phần mềm. Việc sử dụng nhiều đối tượng và gọi các phương thức qua lại có thể tạo ra chi phí thời gian và tài nguyên, đặc biệt quan trọng đối với các ứng dụng yêu cầu hiệu suất cao hoặc xử lý dữ liệu lớn.

### 5.2.2. Mô hình 3 lớp trong C#

Mô hình ba lớp (3-layer architecture) là một kiến trúc phần mềm phổ biến được sử dụng để phân chia ứng dụng thành ba phần chính: Presentation Layer (lớp trình bày), Business Logic Layer (lớp logic kinh doanh), và Data Access Layer (lớp truy cập dữ liệu). Mỗi lớp có một trách nhiệm cụ thể trong quá trình xử lý thông tin và tương tác với các thành phần khác trong hệ thống. Dưới đây là một giới thiệu về mỗi lớp trong mô hình ba lớp khi sử dụng trong C#:

– Presentation Layer (GUI) : Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

– Business Logic Layer (BLL) : Layer này phân ra 2 thành nhiệm vụ :

* Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
* Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

– Data Access Layer (DAL) : Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).

**Ưu điểm:**

Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ nhằm giảm sự kết dính.

Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.

Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.

Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.

Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

### 5.2.3. Singleton

Singleton là một mẫu thiết kế (design pattern) trong lập trình phần mềm, mà mục đích chính là đảm bảo rằng một lớp chỉ có duy nhất một thể hiện (instance) và cung cấp một cách truy cập toàn cục đến thể hiện đó. Điều này đảm bảo rằng chỉ có một thể hiện duy nhất của lớp được tạo ra và sử dụng trong toàn bộ ứng dụng.

Ưu điểm khi thiết kế một phần mềm windows:

* Đảm bảo duy nhất thể hiện: Singleton đảm bảo rằng chỉ có một thể hiện duy nhất của lớp được tạo ra. Điều này hữu ích khi bạn muốn chia sẻ dữ liệu hoặc tài nguyên duy nhất giữa các thành phần khác nhau trong ứng dụng.
* Truy cập toàn cục: Singleton cung cấp một cách truy cập toàn cục đến thể hiện duy nhất của lớp. Điều này giúp giảm sự phức tạp trong việc quản lý các thể hiện và cho phép các thành phần khác nhau trong ứng dụng truy cập dễ dàng vào thể hiện duy nhất.
* Tối ưu hóa tài nguyên: Bằng cách sử dụng Singleton, bạn có thể tối ưu hóa việc sử dụng tài nguyên bằng cách chỉ tạo ra một thể hiện của lớp khi cần thiết. Điều này có thể giảm bớt việc sử dụng bộ nhớ và tài nguyên hệ thống.

### 5.2.3. Store Procedure

Tối Ưu Hóa Đăng Nhập Sử Dụng Stored Procedure để Hạn Chế Lỗi SQL Injection:

* Trong ứng dụng phát triển, việc đảm bảo bảo mật thông tin người dùng là một ưu tiên hàng đầu. Trong quá trình xử lý đăng nhập, rủi ro SQL injection là một trong những vấn đề phổ biến mà các nhà phát triển phải đối mặt. Để giảm thiểu nguy cơ này, việc sử dụng Stored Procedure là một giải pháp hiệu quả.
* Tối Ưu Hiệu Suất: Stored Procedure giúp thực thi các truy vấn trực tiếp trên cơ sở dữ liệu một cách hiệu quả, giảm thiểu việc truyền dữ liệu qua lại giữa ứng dụng và cơ sở dữ liệu.
* Bảo Mật Dữ Liệu: Stored Procedure mã hóa logic xử lý trên cơ sở dữ liệu, không cho phép người dùng cuối truy cập trực tiếp vào các truy vấn SQL, từ đó giảm thiểu khả năng bị tấn công SQL injection.
* Thiết Kế Stored Procedure: Xây dựng Stored Procedure để thực hiện quá trình đăng nhập, bao gồm việc kiểm tra tên người dùng và mật khẩu trong cơ sở dữ liệu.
* Thực Thi Stored Procedure: Sử dụng ADO.NET hoặc Entity Framework để gọi và thực thi Stored Procedure từ ứng dụng.
* Xử Lý Kết Quả: Xử lý kết quả trả về từ Stored Procedure để xác nhận đăng nhập thành công hoặc thất bại.

Sử dụng Stored Procedure trong quá trình đăng nhập không chỉ tối ưu hiệu suất mà còn giúp tăng cường bảo mật cho ứng dụng bằng cách hạn chế nguy cơ SQL injection. Đây là một giải pháp hiệu quả và đáng tin cậy trong việc bảo vệ dữ liệu người dùng.