**MỤC LỤC**

[LỜI NÓI ĐẦU 7](#_Toc106890752)

[CHƯƠNG I: GIỚI THIỆU TỔNG QUAN 8](#_Toc106890753)

[1.1. Giới thiệu đề tài 8](#_Toc106890754)

[1.2. Chức năng đề tài 8](#_Toc106890755)

[1.3. Nhiệm vụ nghiên cứu 8](#_Toc106890756)

[1.4. Mục tiêu nghiên cứu 8](#_Toc106890757)

[1.5. Môi trường thực hiện đề tài 9](#_Toc106890758)

[1.6. Phương tiện nghiên cứu 9](#_Toc106890759)

[1.7. Lộ trình thực hiện đề tài 9](#_Toc106890760)

[1.8. Mô tả các yêu cầu 9](#_Toc106890761)

[CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 10](#_Toc106890762)

[2.1. Đối tượng sử dụng hệ thống 10](#_Toc106890763)

[2.2 Xây dựng biểu đồ phân cấp chức năng của hệ thống 10](#_Toc106890764)

[2.2.1 Xác định các chức năng chi tiết và gom nhóm các chức năng 10](#_Toc106890765)

[2.2.2. Quy trình hoạt động 11](#_Toc106890766)

[2.2.3 Vẽ biểu đồ phân cấp chức năng 11](#_Toc106890767)

[2.3 Xây dựng các mô hình luồng dữ liệu 12](#_Toc106890768)

[2.3.1 Sơ đồ quản lí nhân viên 13](#_Toc106890769)

[2.3.2. Biểu đồ luồng dữ liệu 13](#_Toc106890770)

[2.3.3. Sơ đồ tuần tự chức năng đăng nhập 14](#_Toc106890771)

[2.3.4. Sơ đồ tuần tự chức năng đăng xuất 15](#_Toc106890772)

[2.3.5. Chức năng nhập hàng 15](#_Toc106890773)

[2.3.6. Người quản lí 16](#_Toc106890774)

[2.3.7. Nhân viên 16](#_Toc106890775)

[2.3.8. Sơ đồ hoạt động (Đăng nhập) 17](#_Toc106890776)

[2.4. Đặc tả chi tiết use case 17](#_Toc106890777)

[2.4.1. Use case Đăng nhập 17](#_Toc106890778)

[2.4.2. Use case Đăng xuất 18](#_Toc106890779)

[2.5 Đối tượng 19](#_Toc106890780)

[2.5.1 Bảng dữ liệu 19](#_Toc106890781)

[2.5.2. Sơ đồ liên kết 22](#_Toc106890782)

[CHƯƠNG III: MÔI TRƯỜNG PHÁT TRIỂN 23](#_Toc106890783)

[3.1. Visual Studio .NET 2019 23](#_Toc106890784)

[3.1.1. Visual Studio là gì? 23](#_Toc106890785)

[3.1.2. Lịch sử phát triển Visual Studio 24](#_Toc106890786)

[3.1.3. Tại sao nên sử dụng Visual Studio? 25](#_Toc106890787)

[3.1.4. Một số tính năng quan trọng của Visual Studio 25](#_Toc106890788)

[3.1.5. Thiết kế 27](#_Toc106890789)

[3.1.6. Sự khác nhau giữa các phiên bản Visual Studio 28](#_Toc106890790)

[3.1.7. Tải Visual Studio ở đâu? 31](#_Toc106890791)

[3.1.8. Những ai nên dùng Visual Studio 31](#_Toc106890792)

[3.2. Tìm Hiểu Ngôn Ngữ C# 31](#_Toc106890793)

[3.3. Giới Thiệu winform c# 35](#_Toc106890794)

[3.4. Hệ Quản trị SQL Server 36](#_Toc106890795)

[3.4.1 Khái niệm SQL 36](#_Toc106890796)

[3.4.2 Lịch sử ra đời và các ấn bản của SQL server 37](#_Toc106890797)

[3.4.3. Các thành phần cơ bản trong SQL Server 37](#_Toc106890798)

[3.5. Tại sao lại sử dụng SQL trong thiết kế web? 39](#_Toc106890799)

[CHƯƠNG 4: XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG 40](#_Toc106890800)

[4.1 Thiết kế giao diện 40](#_Toc106890801)

[4.2 Thiết kế database 42](#_Toc106890802)

[4.3 Thiết kế giao diện ứng dụng 45](#_Toc106890803)

[4.3.1 Chức năng đăng nhập 45](#_Toc106890804)

[4.3.2. Giao diện chính của chương trình 46](#_Toc106890805)

[4.3.3. Chức năng quản lí nhân viên 47](#_Toc106890806)

[4.3.4. Chức năng quản lí sản phẩm 47](#_Toc106890807)

[4.3.5. Chức năng quản lí bàn 48](#_Toc106890808)

[4.3.6. Chức năng quản lí, thêm bớt bàn ăn 48](#_Toc106890809)

[4.3.7: Chức năng quản lí ca trực nhân viên 49](#_Toc106890810)

[4.3.8. Chức năng đăng kí 49](#_Toc106890811)

[4.3.9. Chức năng thống kê 50](#_Toc106890812)

[4.3.10. Chức năng lập hóa đơn 51](#_Toc106890813)

[4.3.11. Chức năng phân quyền 52](#_Toc106890814)

[4.4. Xây dựng các chức năng 53](#_Toc106890815)

[4.5 Kết quả chạy thử nghiệm 55](#_Toc106890816)

[CHƯƠNG 5: ĐÁNH GIÁ - KẾT LUẬN 56](#_Toc106890817)

[ Ưu điểm : 56](#_Toc106890818)

[ Khuyết điểm : 56](#_Toc106890819)

[ Hướng phát triển của phần mềm : 56](#_Toc106890820)

[TÀI LIỆU THAM KHẢO 57](#_Toc106890821)

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Sơ đồ phân cấp chức năng 13](#_Toc106888607)

[Hình 2. 2 Mô hình 3 lớp 13](#_Toc106888608)

[Hình 2. 3 Sơ đồ quản lí nhân viên 14](#_Toc106888609)

[Hình 2. 4 Biểu đồ luồng đăng nhập, cấp quyền 14](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888610)

[Hình 2. 5 Sơ đồ tuần tự chức năng đăng nhập 15](#_Toc106888611)

[Hình 2. 6 Sơ đồ tuần tự chức năng đăng xuất 16](#_Toc106888612)

[Hình 2. 7 Sơ đồ chức năng nhập hàng 16](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888613)

[Hình 2. 8 Sơ đồ chức năng quản lí 17](#_Toc106888614)

[Hình 2. 9 Sơ đồ chức năng nhân viên 17](#_Toc106888615)

[Hình 2. 10 Sơ đồ hoạt động (Đăng nhập) 18](#_Toc106888616)

[Hình 2. 11 Bảng Bàn 20](#_Toc106888617)

[Hình 2. 12 Bảng quản lí ca trực 20](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888618)

[Hình 2. 13 Bảng chi tiết hóa đơn 20](#_Toc106888619)

[Hình 2. 14 Bảng thông sản phẩm 21](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888620)

[Hình 2. 15 Bảng thông tin nhân viên 21](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888621)

[Hình 2. 16 Bảng hóa đơn 21](#_Toc106888622)

[Hình 2. 17 Bảng tài khoản 21](#_Toc106888623)

[Hình 2. 18 Bảng thống kê 22](file:///C:\Users\DELL\Desktop\SodaPDF-converted-DATN_VuTienMinh%20-%20Copy%20-%20Copy%20-%20Copy.docx#_Toc106888624)

[Hình 2. 19 Sơ đồ liên kết các bảng 22](#_Toc106888625)

[Hình 2. 20 Sơ đồ liên kết 23](#_Toc106888626)

[Hình 4. 1 Tạo dự án với Winform 40](#_Toc106890375)

[Hình 4. 2 Một số công cụ của Winform 41](#_Toc106890376)

[Hình 4. 3 Database 42](#_Toc106890377)

[Hình 4. 4 Lấy dữ liệu từ database (LINQ) 42](#_Toc106890378)

[Hình 4. 5 Tải dữ liệu từ bảng vào DataGridView 43](#_Toc106890379)

[Hình 4. 6 Thêm dữ liệu vào SQL 43](#_Toc106890380)

[Hình 4. 7 Sửa dữ liệu 44](#_Toc106890381)

[Hình 4. 8 Xóa dữ liệu 44](#_Toc106890382)

[Hình 4. 9 Phương thức Login 53](#_Toc106890383)

[Hình 4. 10 Phương thức hiển thị bàn 54](#_Toc106890384)

[Hình 4. 11 Phương thức hiển thị hóa đơn theo bàn 55](#_Toc106890385)

# LỜI NÓI ĐẦU

Mức sống của người dân ngày một nâng cao, nhu cầu ăn uống, giải trí cũng ngày càng cao. Nhiều cửa hàng, quán cà phê đã được xây dựng để đáp ứng nhu cầu ngày càng đa dạng và ngày càng tăng này. Ngoài nhu cầu giải khát, thưởng thức những món ăn lạ miệng, tán gẫu cùng người thân, bạn bè thì việc học nhóm ở các quán cà phê cũng rất quen thuộc với giới trẻ. Để đáp ứng nhu cầu này của mọi người, việc kinh doanh quán cà phê ngày nay không còn xa lạ~~.~~

Quán cà phê cần nhiều nhân viên hơn và nhu cầu phục vụ mạnh mẽ hơn, làm sao có thể phục vụ được nhiều khách hàng với nhu cầu rất đa dạng và có catalogue tốt nhất. Do đó, việc tổ chức và quản lý đòi hỏi phải nhanh chóng, chính xác, thuận tiện mà không mất nhiều nhân công.

Đây cũng là lý do chính khiến em chọn đề tài “***Xây dựng phần mềm quản lý cho quán cà phê Vô Thường***”.

# CHƯƠNG I: GIỚI THIỆU TỔNG QUAN

## Lý do chọn đề tài

Các quán cà phê truyền thống hiện nay tại Việt Nam đa số mô hình kinh doanh có thể đặt đến 80-90% là vẫn còn quản lí sản phẩm, tính tiền, thanh toán hóa đơn hoặc thống kê doanh thu, sản phẩm tất cả đều làm bằng tay hoặc sổ sách.

Do đó việc xảy ra sai sót trong quá trình làm việc là điều hết sức hiển nhiên và việc đó nó diễn ra hằng ngày nếu ta tính ra được con số thất thoát trong quá trình tính toán về kinh doanh của mỗi quán thì ta sẽ ra đường con số khổng lồ nhưng hầu hết mọi người chủ kinh doanh mô hình này thường sẽ không để ý về điều đó.

Vì vậy chúng em dựa trên sự phổ biến mạnh mẽ của máy tính và sự phổ biến của nó chúng em cho ra đời mộ mô hình quản lí quán cà phê nói chung và quán cà phê Vô Thường nói riêng để giải quyết bài toán này.

Hỗ trợ phần mềm quản lý như tính tiền, in hóa đơn, quản lý nhân viên, ... vv. Phần mềm quản lý bán hàng quán cà phê giúp người dùng tiết kiệm thời gian, nhanh chóng, thao tác đơn giản, dễ sử dụng và quản lý.

Giúp tối ưu hóa việc quản lý hệ thống nhà hàng, giúp việc quản lý trở nên dễ dàng và thuận tiện hơn.

Giúp em cải thiện khả năng sáng tạo và tư duy của mình bằng cách thiết kế giao diện người dùng đưa kiến ​​thức vào thực tế.

## Mục tiêu

*Mục tiêu đề tài này là xây dựng lên một ứng dụng chạy trên các hệ điều hành máy tính trên nền tảng Windown. Ứng dụng sẽ giúp người sử dụng có thể giám sát được tính hình tổng quát của quán thông qua dữ liệu của các hòa đơn theo bàn đồng thời hỗ trợ người sử dụng thêm sửa xóa các thông tin một cách đơn giản về dễ dàng nhất có thể.*

*Cụ thể ứng dụng cho phép người dùng thực hiện các chức năng sau:*

* *Thêm sửa xóa thông tin nhân viên trong quán*
* *Hiển thị số bàn hiện có trong quán và lên hóa đơn cho từng bàn (trạng thái trống hoặc có người)*
* *Thêm sửa xóa sản phẩm đồ uống trong kho*
* *Thêm sửa xóa bàn hiện có trong quán*
* *Lên lịch ca trực*
* *Thanh toán hóa đơn*
* *Thống kê doanh thu*
* *Đăng kí tài khoản*
* *Đăng xuất*
* *Đối với quản trị viên sẽ có đầy đủ tất cả các quyền còn đối với nhân viên chỉ có 2 quyền cơ bản là lên hóa đơn và thanh toán*

*Bên cạnh việc xây dựng và hoàn thiện ứng dụng phần mềm, mục tiêu đề tài còn hoàn thiện được báo cáo đồ án tốt nghiệp với đầy đủ nội dung, trình bày theo đúng quy định cùng với việc học hỏi và hoàn thiện hơn về kỹ năng phục vụ cho công việc sau này.*

## Nội dung nghiên cứu

## *Trên cơ sở mục tiêu của đề tài đặt ra, để có thể hoàn thiện được những mục tiêu đó, em sẽ tập trung vào các nội dung nghiên cứu sau:*

## • *Nghiên cứu quy trình xây dựng, phát triển và triển khai một ứng dụng trên thiết bị di động nói chung. Đánh giá đúng quy mô triển khai của ứng dụng để thiết kế hệ thống phù hợp với nhu cầu đặt ra.*

## • *Nghiên cứu ngôn ngữ winform c# sử dụng kĩ thuật LINQ*

## • *Nghiên cứu về quy trình và yêu cầu nghiệp vụ cho bài toán.*

## • *Nghiên cứu sử dụng một số thư viện phục vụ cho ứng dụng.*

## Phạm vi của đề tài

Phần mềm quản lí quán café do chúng em thiết kế thích hợp cho quán café vừa và nhỏ đặc biệt phù hợp với các quán café truyền thống của nước ta như hiện nay.

## Phương tiện nghiên cứu

* Nghiên cứu cơ sở lý thuyết về phân tích và thiết kế hệ thống thông tin.
* Hệ quản trị CSDL và ngôn ngữ lập trình C Sharp (C#).
* Sử dụng các công cụ như Visual Studio để lập trình
* Sử dụng hệ quản trị cơ sở dữ liệu SQL Server 2014 để quản trị cơ sở dữ liệu cho đề tài

## Lộ trình thực hiện đề tài

* Tìm hiểu, khảo sát quy trình nghiệp vụ quản lý quán cafe.
* Phân tích thiết kế, phân rã chức năng hệ thống.
* Xây dựng phần mềm.
* Viết báo cáo.

## Mô tả các yêu cầu

Yêu cầu chức năng:

* + Cho phép người quản trị thêm, xóa và sửa đổi nhân viên.
  + Cho phép quản trị viên thêm, xóa và chỉnh sửa sản phẩm.
  + Cho phép quản trị viên thêm, xóa và sửa đổi bàn.
  + Cho phép quản trị viên lên lịch cho ca làm việc.
  + Cho phép người quản trị thêm, xóa, sửa đổi hóa đơn.
  + Cho phép cập nhật và xử lý tất cả các cơ sở dữ liệu.
  + Hệ thống phải cung cấp các giao diện thích hợp để người dùng triển khai phần mềm.
  + Phần mềm phải đáp ứng nhu cầu của người dùng, hoạt động nhanh chóng và chính xác.

# CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1. Đối tượng sử dụng hệ thống

Ứng dụng được phát triển hướng tới tất cả các đối tượng là cá nhân tổ chức có nhu cầu quản lý một quần cafe, trà sữa loại vừa và nhỏ.

Bao gồm 2 đối tượng chính:

Quản lý: Người quản lí có thể quản lí toàn bộ hệ thống, như thêm - sửa – xóa danh mục thực đơn, cập nhật nhân viên và tài khoản

Nhân viên: Xem danh sách bàn trống; Thực hiện kiểm soát hóa đơn theo bàn, thanh toán và thêm món.

Người sử dụng là những nhân viên được quản lý cấp tài khoản. Nhân viên có thể thay đổi mật khẩu và quản lí bàn, hóa đơn, …

## 2.2 Xây dựng biểu đồ phân cấp chức năng của hệ thống

## 2.2.1 Xác định các chức năng chi tiết và gom nhóm các chức năng

Trên cơ sở khảo sát bài toàn thực tế, ứng dụng sẽ có các chức năng chính sau: Nhóm chức năng “Hóa Đơn” bao gồm các chức năng:

* Chức năng hiển thị: Cho phép người dùng xem trạng thái của bàn “Trống” hay “Có người”
* Chức năng xem hóa đơn theo bàn: Cho phép người dung (Nhân viên) xem danh sách hóa đơn theo bàn.
* Chức năng đăng nhập: Tại phần mềm quản lí sẽ phân ra thành hai cấp quyền người sử dụng (admin) và user. Hệ thống sẽ tự kiểm tra xem user đăng nhập là admin hay nhân viên và từ đó phân quyền cho hợp lí.
* Chức năng đăng xuất: Để đăng xuất tài khoản ra khỏi ứng dụng.
* Chức năng thêm dữ liệu vào menu: Cho phép admin thêm, xóa, sửa thông tin menu cũng như cập nhật giá và số lượng tồn trong kho.
* Chức năng quản lí nhân viên: Hiển thị danh sách sinh viên và thêm sửa xóa dữ liệu nhân viên.
* Chức năng quản lí ca trực: Cho phép sắp xếp ca trực sao cho phù hợp và book ca trực cho nhân viên.
* Chức năng quản lí bàn: Chỉnh sửa thông tin của bàn ( thêm,sửa,xóa)
* Chức năng đăng kí tài khoản: Thêm tài khoản cho nhân viên hoặc quản lí mới

## 2.2.2. Quy trình hoạt động

**Hoạt động lập hóa đơn:**

Khi có nhu cầu gọi đồ uống tại quán, khách hàng sẽ liên lạc với nhân viên trực tiếp tại quán, thông báo cụ thể về nhu cầu của mình (đồ uống, số lượng người…). Sau đó, nhân sẽ kiểm tra danh sách bàn, đối chiếu với đối với nhu cầu của khách và trả lời. Nếu có thể đáp ứng được nhu cầu của khách, nhân viên sẽ tiến hành đặt bàn.

**Hoạt động thanh toán:**

Khi khách hàng đã sử dụng xong dịch vụ của quán và có nhu cầu thanh toán, thì nhân viên thu ngân sẽ tiến hành hỏi khách về số bàn, và tiến hành xuất hóa đơn cho khách. Hóa đơn bao gồm:

+ Tên đồ uống

+ Số lượng

+ Giá tiền

+ Giảm giá

+ Thành tiền

**Đăng ký*:***

Trong quá trình hoạt động quán sẽ có nhân viên mới vào thêm, quản lí sẽ có nhiệm vụ thêm mới 1 tài khoản cho nhân viên đó.

**Thống kê doanh thu:**

Thống kê doanh thu sẽ hỗ trợ cho công việc thống kê doanh thu của quán và số lượng sản phẩm bán ra tại ngày hôm đó hoặc vào tháng nào đó và báo cáo có thể chia làm nhiều loại khác nhau: báo cáo hằng ngày hay báo cáo theo khoảng thời gian (từ ngày – đến ngày )

Đây là một chức năng quan trọng của hệ thống, thông qua chức năng này, quản lí sẽ có một cái nhìn bao quát, toàn diện về tình trạng của quán và thông qua đó đưa ra các chính sách điều chỉnh phù hợp.

## 2.2.3 Vẽ biểu đồ phân cấp chức năng

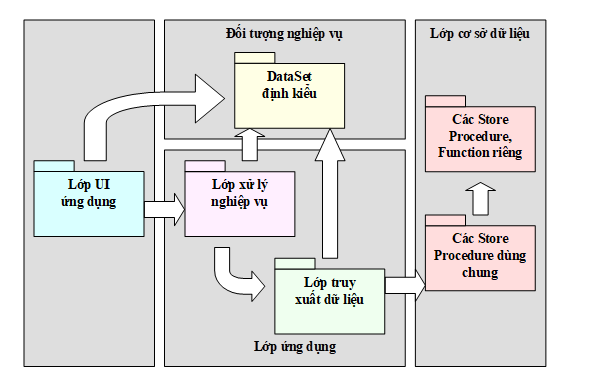
Trên cơ sở chức năng chi tiết đã xác định được trong phần 2.2.1 ở trên em tiến hành xây dựng biểu đồ phân cấp chức năng của hệ thống như hình dưới.

Diagram

Description automatically generated

Hình 2. 1 Sơ đồ phân cấp chức năng

## 2.3 Xây dựng các mô hình luồng dữ liệu



Hình 2. 2 Mô hình 3 lớp

## 2.3.1 Sơ đồ quản lí nhân viên

Diagram

Description automatically generated

Hình 2. 3 Sơ đồ quản lí nhân viên

## Không có mô tả.2.3.2. Biểu đồ luồng dữ liệu

**Hình 2. 4 Biểu đồ luồng đăng nhập, cấp quyền**

## Diagram Description automatically generated2.3.3. Sơ đồ tuần tự chức năng đăng nhập

Hình 2. 5 Sơ đồ tuần tự chức năng đăng nhập

## 2.3.4. Sơ đồ tuần tự chức năng đăng xuấtTable Description automatically generated with medium confidence

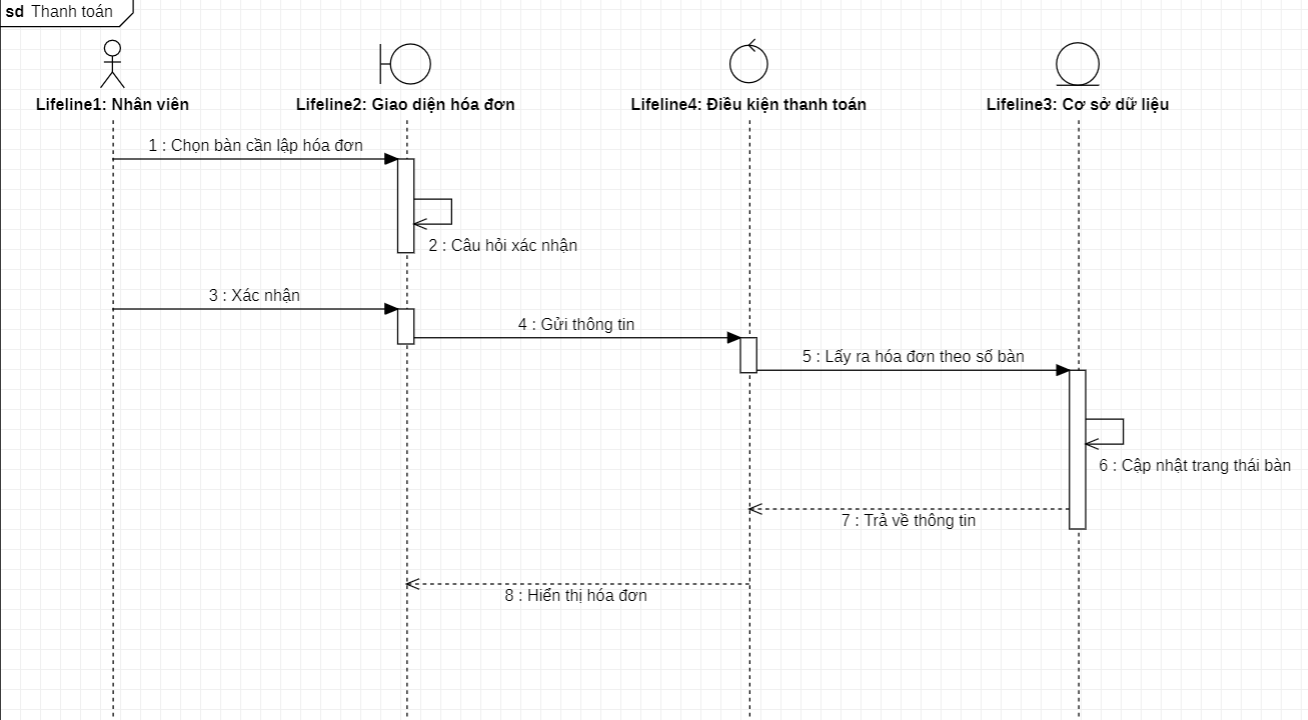
Hình 2. 6 Sơ đồ tuần tự chức năng đăng xuất

## 2.3.5. Chức năng lên hóa đơn theo bàn

## 

## 2.3.6. Sơ đồ tuần tự chức năng thanh toán

Hình 2. 7 Sơ đồ chức năng lên hóa đơn theo bàn

**

Hình 2. 8 Sơ đồ chức năng thanh toán

## 2.3.7. Người quản lí



Hình 2. 8 Sơ đồ chức năng quản lí

## 2.3.8. Nhân viên



Hình 2. 9 Sơ đồ chức năng nhân viên

## 2.3.9. Sơ đồ hoạt động (Đăng nhập)

Diagram

Description automatically generated

Hình 2. 10 Sơ đồ hoạt động (Đăng nhập)

## 2.4. Đặc tả chi tiết use case

## 2.4.1. Use case Đăng nhập

*Bảng 2.1 Use Case cho chức năng đăng nhập*

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Actor** | Nhân viên, Người quản lí |
| **Trigger** | Khi actor yêu cầu đăng nhập |
| **Brief Description** | Use case này mô tả các bước đăng nhập của actor vào hệ  thống. |
| **Pre-conditions** | Không có |
| **Basic Flows** | 1. Hệ thống yêu cầu actor nhập tên đăng nhập, mật khẩu; 2. Actor nhập tên đăng nhập, mật khẩu của mình và nhấn nút đăng nhập; 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống thông báo thành công và cho actor đăng nhập vào hệ thống, đồng thời phân quyền tùy theo loại nhân viên. |
| **Alternative Flows** | 5. Tên đăng nhập hoặc mật khẩu không đúng, hệ thống hiện  thông báo cho người dung và yêu cầu đăng nhập lại. |
| **Post-conditions** | Cho phép actor đăng nhập vào hệ thống nếu đăng nhập đúng |
| **Special Requirements** | Không có |

*Bảng 2.2 Use case chức năng đăng xuất*

|  |  |
| --- | --- |
| **Use case** | Đăng xuất |
| **Actor** | Nhân viên, Người quản lí |
| **Brief Description** | Use case này môt tả việc đăng xuất khỏi hệ thống |
| **Pre-conditions** | Actor phải đăng nhập thành công |
| **Basic Flows** | 1. Actor chọn chức năng đăng xuất khỏi hệ thống 2. Actor nhập tên đăng nhập, mật khẩu của mình và nhấn nút đăng nhập; 3. Hệ thống hiển thị yêu cầu xác nhận từ actor 4. Actor dùng xác nhận đăng xuất 5. Hệ thống đăng actor khỏi hệ thống. |
| **Alternative Flows** | 3.1. Actor không xác nhận đăng xuất thì hệ thống sẽ giữ  nguyên hiện trạng. |
| **Post-conditions** | Đăng xuất actor khỏi hệ thống và bỏ quyển sử dụng hệ  thống của actor. |
| **Special Requirements** | Không có |

Bảng 2.3. Use case chức năng thông kê doanh thu

|  |  |
| --- | --- |
| **Use case** | Thống kê doanh thu |
| **Actor** | Người quản lý |
| **Brief Description** | Use case này cho phép Người quản lý thống kê doanh  thu của quán theo yêu nhu cầu |
| **Pre-conditions** | Người quản lí phải đăng nhập thành công vào hệ thống với quyền admin |
| **Basic Flows** | 1. Người quản lí chọn nút ”Thống kê”.  2.Hệ thống hiển thị menu thống kê:  + Theo ngày  + Theo mặc định   1. Người quản lí chọn một trong các mục.   3. Hệ thống sẽ thống kê và xuất ra kết quả |
| **Alternative Flows** | Không có |

*Bảng 2.4 Chức năng thêm hóa đơn*

|  |  |
| --- | --- |
| **Use case** | Thêm hóa đơn theo bàn |
| **Actor** | Nhân viên |
| **Brief Description** | Use case này cho phép nhân viên thêm hóa đơn theo bàn |
| **Pre-conditions** | Nhân viên phải đăng nhập thành công vào hệ thống |
| **Basic Flows** | 1. Nhân viên kinh doanh chọn nút ”thêm”. 2. Hệ thống sẽ bắt nhân viên chọn bàn trước khi thêm món vào hóa đơn 3. Nhân viên chọn bàn và thêm lại món 4. Hệ thống sẽ thay đổi trạng thái bàn thành trạng thái có người và đồng thời lập hóa đơn cho bàn |
| **Alternative Flows** | Không có |
| **Exception Flows** | * 1. Nhân viên chọn ”hủy”      1. Hệ thống trở về hiện trạng ban đầu |
| **Post-conditions** | Thông tin bàn được cập nhật lại |
| **Special Requirements** | Không có |

## 2.5 Thiết kế và xây dựng cơ sở dữ liệu

## 2.5.1. Hệ Quản trị SQL Server

## ****2.5.1.1. Khái niệm SQL****

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

## ****2.5.1.2. Lịch sử ra đời và các ấn bản của SQL server****

**Lịch sử ra đời SQL server**

Năm 1989, phiên bản đầu tiên của SQL Server 1.0 ra đời được dùng cho các hệ điều hành 16bit và được phát triển cho tới ngày nay.

Cho tới khi SQL Server ra phiên bản 6.5 thì được thị trường chấp nhận rộng rãi. Một đột phá cải tiến cho SQL Server 7.0 khi được Microsoft viết lại một engine hoàn toàn mới. Đến khi SQL Server từ phiên bản 7.0 cải tiến lên 8.0 chủ yếu phát triển về tính năng **[thiết kế website](https://monamedia.co/dich-vu/thiet-ke-website/)**.

**Một vài ấn bản SQL Server**

**Enterprise:** là một ấn bản chứa tất cả các đặc điểm nổ bật của SQL Server như: các công cụ cho tạo và quản lý phân cụm SQL Server, nhân bộ máy cơ sở dữ liệu và một số dịch vụ đi kèm. Nó có thể đánh địa chỉ 12 terabytes và quản lý cơ sở dữ liệu lên tới 524 petabytes.

**Standard:** Ấn bản này có thể chạy tốt trên hệ thống lên tới 4 CPU và 2 GB RAM rất thích hợp cho các dịch vụ thiết kế web vừa và nhỏ.

**Developer:** Ấn bản này giới hạn số lượng người kết nối với server nhưng có đầy đủ các tính năng của Enterprise Edition. Đây là phiên bản được sử dụng cho kiểm tra và phát triển ứng dụng phù hợp cho các cá nhân trong lĩnh vực web như: [freelancer Việt Nam](http://freelancervietnam.vn/), [website spa](http://websitespa.vn/),…

**Workgroup:** ấn bản SQL Server này có các chức năng lõi cơ sở dữ liệu nhưng không đi kèm các dịch vụ. Ở phiên bản 2012 không có ấn bản này.

Express: Ấn bản này dễ dàng sử dụng và quản trị cơ sở dữ liệu đơn giản.

## ****2.5.1.3. Các thành phần cơ bản trong SQL Server****

Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service, … Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

**Database Engine**: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

**Integration Services**: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu.  Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access, … và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

**Analysis Services**: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.

**Notification Services**: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người dăng ký sử dụng trên nhiều loại thiết bị khác nhau.

**Reporting Services**: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

**Full Text Search Service**: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

**Service Broker**: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

## ****2.5.2. Tại sao lại sử dụng SQL trong thiết kế web?****

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập mà nó chỉ là một thành phần với vai trò ngôn ngữ là công cụ giao tiếp giữa hệ cơ sở dữ liệu và người dùng. Chính vì thế nó được sử dụng trong các dịch vụ [thiết kế web đẹp](https://mona.media/thiet-ke-website-tai-hcm/) với chức năng giao tiếp với người dùng với các vai trò sau:

SQL là một ngôn ngữ đòi hỏi có tính tương tác cao: Người dùng có thể dễ dàng trao đổi với các tiện ích thông qua các câu lệnh của SQL đến cơ sở dữ liệu và nhận kết quả từ cơ sở dữ liệu.

SQL là một ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể xây dựng các chương trình ứng dụng giao tiếp với cơ sở dữ liệu bằng cách nhúng các câu lệnh SQL vào trong ngôn ngữ lập trình.

SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu: Người quản trị cơ sở dữ liệu có thề quản lý, định nghĩa và điều khiển truy cập cơ sở dữ liệu thông qua SQL.

SQL là một ngôn ngữ lập trình cho các hệ thống chủ khách: SQL được sử dụng như là một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ.

SQL là ngôn ngữ truy cập dữ liệu trên Internet: SQL được sử dụng với vai trò tương tác với dữ liệu trong hầu hết các máy chủ web và máy chủ Internet.

SQL là ngôn ngữ cơ sở dữ liệu phân tán: Với vai trò giao tiếp với các hệ thống trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

Nhìn chung SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được sử dụng trong các chức năng tương tác giữa người dùng và dữ liệu. Chính vì thế, nó được sử dụng trong các dịch vụ thiết kế web rẻ và chuyên nghiệp.

## 2.5.1 Thiết kế CSDL

## Bảng tbl\_quanlycatruc: lưu trữ thông tin nhân viên tương ứng với ca trực của mình

## *Bảng 2.5. Thiết kế bảng Quản lý ca trực*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| ID | **X** |  | int |
| ca |  |  | nvarchar(20) |
| ngaytruc |  |  | datetime |
| IDnhanvien |  | **X** | int |

## Bảng tbl\_quanlyban: lưu trữ thông tin bàn của quán

*Bảng 2.6: Thiết kế bảng Quản lý bàn*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| Soban | **X** |  | int |
| Tinhtrang |  |  | nvarchar(30) |

## Bảng tbl\_quanlychitiethoadon: lưu trữ thông tin của hóa đơn theo bàn

## *Bảng 2.5. Thiết kế bảng chi tiết hóa đơn*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| MaHoaDon | **X** |  | int |
| MaSP |  | **X** | int |
| NgayBan |  |  | date |
| SoBan |  | **X** | int |
| SoLuong |  |  | int |
| DonGia |  |  | float |
| ThanhTien |  |  | float |

## Bảng tbl\_quanlysanpham: lưu trữ thông tin sản phẩm hiện có

## *Bảng 2.5. Thiết kế bảng sản phẩm*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| ID | **X** |  | int |
| tendouong |  |  | nvarchar(30) |
| ghichu |  |  | nvarchar(30) |
| soluong |  |  | int |
| giatien |  |  | float |

## Bảng tbl\_quanlynhanvien: lưu trữ thông tin của nhân viên

## *Bảng 2.5. Thiết kế bảng thông tin nhân viên*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| MANV | **X** |  | int |
| TENNHANVIEN |  |  | nvarchar(50) |
| GIOITINH |  |  | nvarchar(10) |
| DIENTHOAI |  |  | nvarchar(12) |

## Bảng tbl\_quanlyhoadon: lưu trữ thông tin của hóa đơn theo từng bàn

## *Bảng 2.5. Thiết kế bảng thông tin hóa đơn*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| ID | **X** |  | int |
| soban |  |  | int |

## Bảng tbl\_quanlytaikhoam: lưu trữ thông tin tài khoản

## *Bảng 2.5. Thiết kế bảng tài khoản*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| Name |  |  | varchar(20) |
| UserName | **X** |  | varchar(20) |
| Password |  |  | varchar(20) |
| priority |  |  | int |

## Bảng tbl\_quanlythongke: lưu trữ thông tin của các hóa đơn đã lập

## *Bảng 2.5. Thiết kế bảng thống kê*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Khóa chính** | **Khóa ngoại** | **Kiểu dữ liệu** |
| id | **X** |  | int |
| TenSanPham |  |  | nvarchar(30) |
| Soluong |  |  | int |
| ngaylap |  |  | date |
| doanhthu |  |  | float |

2.5.2. Sơ đồ liên kếtDiagram

Description automatically generatedDiagram

Description automatically generated

Hình 2. 20 Sơ đồ liên kết

# CHƯƠNG III: MÔI TRƯỜNG PHÁT TRIỂN

## 3.1. Visual Studio .NET 2019

## 3.1.1. Visual Studio là gì?

Là con đẻ của Microsoft, Visual Studio được biết đến như một phần mềm lập trình website nổi tiếng có một không hai và đến nay chưa có công cụ nào có thể thay thế được chúng.

A picture containing text, electronics

Description automatically generatedVisual Studio được lập trình bằng ngôn ngữ VB+ và C# - vốn là các ngôn ngữ lập trình cho phép người dùng xây dựng hệ thống website dễ dàng và nhanh chóng nhất.

Visual Studio là gì?

Với lợi thế được xây dựng và phát triển trực tiếp từ ông lớn Microsoft, Visual Studio từ khai khai sinh cho đến nay đã được cập nhật rất nhiều phiên bản sử dụng khác nhau. Điều này cho phép người dùng thoải mái lựa chọn phiên bản tốt nhất, có cấu hình phù hợp nhất với đa dạng dòng máy của mình.

Ngoài ra, tùy theo nhu cầu của người dùng, Visual Studio cũng cho phép bạn tự do chọn lựa giao diện chính cho máy của mình.

## 3.1.2. Lịch sử phát triển Visual Studio

Visual Studio được sáng tạo bởi Microsoft. Vào năm 1997, ông lớn này đã cho ra mắt hai phiên bản của Visual Studio là Professional và Enterprise. Trải qua nhiều lần phát triển, đến nay, Visual Studio sở hữu nhiều phiên bản khác nhau như VS 2005, VS 2008, VS 2010, VS 2012, VS 2013, VS 2015.

Trong đó, phiên bản mới nhất là Visual Studio 2019 được chính thức phát hành vào tháng 4 năm 2019. Theo đó, với mỗi phiên bản phát hành của mình, Visual Studio lại tích hợp thêm nhiều tính năng, công nghệ mới mẻ và hấp dẫn hơn.

A screenshot of a computer

Description automatically generated with medium confidence

Lịch sử phát triển Visual Studio

Trong trường hợp muốn tiết kiệm chi phí, người dùng có thể tải Visual Studio phiên bản Comunity vì chúng cho phép miễn phí hoàn toàn và cung cấp đầy đủ các tính năng cơ bản.

Trường hợp bạn có sẵn tài khoản DreamSpark liên kết giữa Microsoft với doanh nghiệp/nhà trường thì người dùng có thể thoải mái tải bản Visual Studio Professional và sử dụng miễn phí.

## 3.1.3. Tại sao nên sử dụng Visual Studio?

Với những người mới làm quen với lập trình, thiết kế website thì ngoài việc thắc mắc Visual Studio là gì còn rất quan tâm đến lý do tại sao nên sử dụng chúng mà không phải là các công cụ khác để lập trình như DevC, Code Block, CFree.

Sở dĩ Visual Studio được giới lập trình ưa chuộng như vậy là bởi những ưu điểm vượt trội sau đây:

* Visual Studio hỗ trợ lập trình trên nhiều nền tảng ngôn ngữ khác nhau từ C/C++, C#, cho đến F#, Visual Basic, HTML, CSS, JavaScript. Thậm chí, phiên bản VS 2019 có hỗ trợ code trên ngôn ngữ Python.
* Visual Studio giúp hỗ trợ khả năng gỡ rối (Debug) hiệu quả và dễ dàng  thông qua việc chạy từng câu lệnh và theo dõi sự thay đổi trạng thái của chương trình qua giá trị các biến, cách vận hành của mã.
* Visual Studio sở hữu giao diện thân thiện, dễ dàng sử dụng cho người mới bắt đầu.
* Visual Studio cho phép xây dựng ứng dụng chuyên nghiệp thông qua kéo thả, hỗ trợ người mới bắt đầu tiếp cận nhanh hơn.
* Visual Studio cho tích hợp nhiều ứng dụng khác như ứng dụng hỗ trợ quản lý và viết mã nhanh cho các ngôn ngữ thuộc.Net- Resharper, hay cho phép cài đặt thư viện dễ dàng nhờ Nuget.

Ngoài ra, công cụ này còn hỗ trợ phát triển ứng dụng desktop trên MFC, Windows Form hay Universal App. Đồng thời phát triển ứng dụng mobile trên Windows Phone 8/8.1, Windows 10, iOS và cả Android Xamarin. Visual Studio còn hỗ trợ phát triển website trên nền tảng Web Form, ASP.NET MVC và tất nhiên, phát triển Microsoft Office.

Với tất cả những lợi thế khác biệt đó, Visual Studio là lựa chọn ưu tiên của đông đảo lập trình viên trên toàn thế giới.

## 3.1.4. Một số tính năng quan trọng của Visual Studio

## 

**Lưu trữ dữ liệu ở dạng phân cấp**

Tại Visual Studio hầu hết các tệp lưu trữ đoạn mã code đều được đặt trong các thư mục tương đương nhau. Ngoài ra, với những tệp đặc biệt quan trọng, Visual Studio cũng cung cấp các thư mục phân cấp nhằm giúp người dùng dễ dàng lưu trữ và tìm kiếm dữ liệu.

**Hỗ trợ Git**

Sử dụng Visual Studio, người dùng có thể kéo hoặc copy mã trực tiếp từ GitHub. Sau đó, có thể thay đổi các mã này và lưu lại trên phần mềm.

**Hỗ trợ đa nền tảng**

Nếu như với các công cụ lập trình khác thường chỉ được sử dụng cho một trong 3 hệ điều hành sau: Windows, Linux, Mac Systems thì trái lại, Visual Studio cho phép bạn lập trình dễ dàng trên cả ba nền tảng trên mà không gặp bất cứ trở ngại nào.

Table

Description automatically generated

Visual Studio có thể hỗ trợ đa nền tảng

**Biên tập mã**

Như mọi IDE khác, Visual Studio cũng cung cấp trình soạn thảo mã hỗ trợ hoàn thiện mã và làm rõ cú pháp bằng cách sử dụng IntelliSense.Công cụ này không chỉ sử dụng cho các hàm, biến và các phương pháp mà còn áp dụng cho các cấu trúc ngôn ngữ bao gồm truy vấn hoặc vòng điều khiển.

Ngoài ra, những trình biên tập mã của Visual Studio còn hỗ trợ cài đặt dấu trang trong mã giúp người dùng dễ dàng thực hiện các điều hướng thu hẹp khối mã lệnh, tìm kiếm gia tăng…

Đặc biệt, Visual Studio còn có chức năng biên dịch mã nền giúp cung cấp thông tin phản hồi về cú pháp hoặc biên dịch lỗi và được đánh dấu bằng những gạch sóng đỏ.

**Cung cấp kho tiện ích mở rộng**

Bạn muốn sử dụng một ngôn ngữ lập trình mà Visual Studio không có sẵn? Rất đơn giản, bạn chỉ cần tải xuống kho tiện ích mở rộng của công cụ này là xong. Điều đặc biệt là việc tải xuống những chức năng này không làm ảnh hưởng đến hiệu năng của Visual Studio bởi chúng hoạt động hoàn toàn độc lập với nhau.

**Bảo mật kho lưu trữ hiệu quả**

Nhu cầu lưu trữ an toàn trong lập trình là rất lớn. Với Visual Studio, người dùng có thể yên tâm về vấn đề này. Bởi công cụ này dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ nào hiện có và đảm bảo an toàn cao cho chúng.

## 3.1.5. Thiết kế

Graphical user interface, application

Description automatically generated

Tính năng thiết kế của visual studio

Windows Forms Designer: Thiết kế này của Visual Studio  được dùng với mục tiêu xây dựng GUI cho Windows Forms, nhằm xây dựng các nút điều khiển bên trong hoặc khóa chúng vào bên cạnh mẫu. Việc điều khiển trình bày dữ liệu có thể sẽ được kết nối với các nguồn cơ sở dữ liệu hoặc truy vấn...

WPF Designer: Cũng giống như trên tính năng thiết kế WDP Designer cho phép người dùng kéo, thả ẩn dụ. Chức năng này sử dụng để tương tác giữa người và máy tính nhắm mục tiêu vào Windows Presentation Foundation.

Web designer/development: Visual Studio  cũng cung cấp trình soạn thảo và thiết kế website cho phép xây dựng các trang web qua chức năng kéo thả. Điều này giúp người dùng xây dựng website dễ dàng hơn. Nhất là với những doanh nghiệp cần xây dựng những website đơn giản như: thiết kế website du lịch, giáo dục… có thể sử dụng cách thức này mà vẫn cho phép bạn sở hữu một website với các chức năng hoàn chỉnh.

**Hỗ trợ nhiều ngôn ngữ lập trình**

Cũng vậy, Visual Studio ghi điểm trong mắt người dùng nhờ lợi thế hỗ trợ nhiều ngôn ngữ lập trình cùng lúc từ C/C++, C# cho đến F#, Visual Basic, HTML, CSS, JavaScript… Do đó, nó có thể phát hiện và nhanh chóng đưa ra thông báo ngay khi chương trình có lỗi.

## 3.1.6. Sự khác nhau giữa các phiên bản Visual Studio

Để hiểu rõ hơn Visual Studio là gì, bạn cần phân biệt được sự khác nhau giữa các phiên bản của chúng. Cụ thể như sau:

**Graphical user interface, website

Description automatically generatedVisual Studio Community**

Sự khác nhau giữa các phiên bản Visual Studio

Visual Studio Community được xây dựng nhằm phục vụ các nhà phát triển cá nhân, học viên lập trình hoặc những người nghiên cứu học thuật. Công cụ này cũng có thể phục vụ cho các dự án nguồn mở, các tổ chức phi doanh nghiệp với số lượng tối đa là 5 người dùng.

Visual Studio Community luôn luôn hỗ trợ nền tảng phát triển

Đối với môi trường phát triển tích hợp, Visual Studio Community sẽ hỗ trợ tích hợp một số tính năng như: Hỗ trợ định nghĩa Peek, tái cấu trúc, trình xem tài nguyên mô hình, triển khai web bằng một lần nhấp, giải pháp trực quan hóa với đồ thị phụ thuộc cùng bản đồ mã và đa mục tiêu.

Với phần gỡ lỗi và chẩn đoán nâng cao, Visual Studio Community hỗ trợ số liệu mã, gỡ lỗi đồ họa, phân tích các mã tính và trung tâm chẩn đoán và hiệu suất.

Với vai trò là phiên bản miễn phí, Visual Studio Community chỉ cung cấp công cụ Kiểm thử đơn vị cho người dùng để kiểm tra mã của họ.

Về tích hợp phát triển đa nền tảng: Visual Studio Community cung cấp các tính năng đa nền tảng với tên gọi Xamarin, bao gồm mã chia sẻ giữa các đối tác ứng dụng iOS và Android, thiết kế giao diện người dùng iOS và Android gốc, Xamarin Form và Xamarin Instant Player.

Visual Studio Community hỗ trợ phát triển hợp tác với các nhà phát triển khác ngoài Visual Studio tùy theo nhu cầu của bạn như: Powerpoint Storyboarding, đánh giá mã, chức năng tạm dừng hoặc tiếp tục tác vụ, Team Explorer,  hỗ trợ đánh giá mã, khả năng tạm dừng hoặc tiếp tục tác vụ với sự trợ giúp của các bên thứ ba.

**Visual Studio Professional**

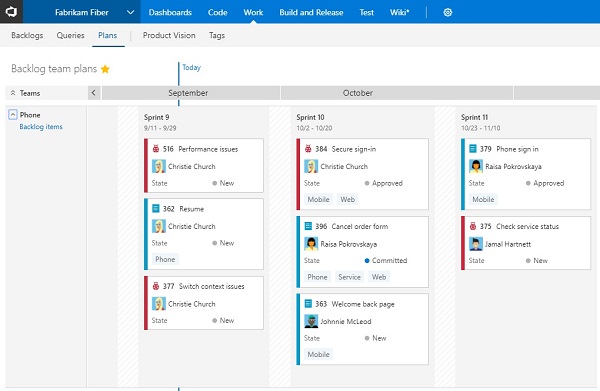
Ngoài cộng đồng Visual Studio, Visual Studio Professional thường được sử dụng phục vụ cho các doanh nghiệp là chủ yếu

Tương tự như bản Visual Studio Community,  Visual Studio Professional cũng hỗ trợ nền tảng phát triển và hỗ trợ tích hợp mọi thứ mà phiên bản anh em Community hỗ trợ, bao gồm cả CodeLens. Ngoài ra, phiên bản chuyên nghiệp này cũng hỗ trợ gỡ lối và chẩn đoán nâng cao.

Khác với phiên bản trên, Visual Studio Professional chỉ cho phép  hỗ trợ kiểm thử đơn vị cho người dùng để họ kiểm tra mã của mình.

Visual Studio Professional cũng cung cấp các tính năng phát triển đa nền tảng tương tự như Visual Studio Community. Đồng thời cho phép bạn kết hợp với các nhà phát triển khác, giống như phiên bản trên.

**Visual Studio Enterprise**

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/vis16354692484800.jpg)

Visual Studio Enterprise

Giống như phiên bản chuyên nghiệp, Visual Studio Enterprise dành cho các nhà phát triển cá nhân,  học viên, người nghiên cứu học thuật, đóng góp cho các dự án mở, các tổ chức phi doanh nghiệp với số lượng không quá 5 người dùng và các doanh nghiệp.

Công cụ này cũng hỗ trợ nền tảng phát triển và hỗ trợ tích hợp mọi thứ mà phiên bản chuyên nghiệp hỗ trợ, bao gồm xác thực phụ thuộc trực tiếp, sơ đồ lớp kiến trúc, xác thực kiến trúc và bản sao mã.

Đối với  tính năng gỡ lỗi và chẩn đoán nâng cao, bên cạnh những tính năng mà bản chuyên nghiệp hỗ trợ, Visual Studio Enterprise còn hỗ trợ thêm tích hợp trình gỡ lỗi bản đồ mã, phân tích kết xuất bộ nhớ .NET và hỗ trợ cả IntelliTrace.

Về tính năng hỗ trợ các công cụ kiểm tra: ngoài khả năng kiểm thử đơn vị, phiên bản Visual Studio Enterprise còn giúp người dùng kiểm tra nhiều mảng khác như: đơn vị trực tiếp, quản lý trường hợp kiểm tra, hiệu năng và tải web IntelliTest, kiểm tra đơn vị, bảo hiểm mã, quản lý phòng thí nghiệm, kiểm tra giao diện người dùng… Ngoài ra, công cụ này còn có tính năng kiểm tra thủ công đối với Microsoft trình quản lý kiểm tra, kiểm tra thăm dò và chuyển tiếp nhanh để kiểm tra thủ công đối  với Microsoft Test Manager.

Visual Studio Enterprise cung cấp nhiều tính năng Xamarin hơn so với bản Community và bản Enterprise hỗ trợ. Cụ thể chúng bao gồm các hội đồng nhúng, trình kiểm tra, trình biên dịch Xamarin và trình mô phỏng iOS từ xa cho Windows.

Visual Studio Enterprise cũng cho phép người dùng hợp tác với các nhà phát triển khác, bao gồm PowerPoint Storyboarding, cho phép đánh giá mã, khả năng tạm dừng hoặc tiếp tục tác vụ, Team Explorer với sự trợ giúp từ các công cụ của bên thứ ba.

## 3.1.7. Tải Visual Studio ở đâu?

Ngoài việc tìm hiểu Visual Studio là gì? Người học lập trình cũng cần biết địa chỉ tải Visual Studio.

Cụ thể, nếu muốn trải nghiệm ứng dụng này, có thể dễ dàng tìm hiểu và tải xuống trực tiếp từ VisualStudio.com.

## 3.1.8. Những ai nên dùng Visual Studio

Visual Studio dành cho bất cứ lập trình viên nào. Ngoài ra, bất cứ ai dù là testers, data administrators hay chỉ cần quan tâm đến vấn đề lập trình đều có thể sử dụng công cụ này

Là một trong những trình biên tập phổ biến và tốt nhất hiện nay, Visual Studio là lựa chọn hoàn hảo của rất nhiều lập trình viên trên toàn thế giới. Hy vọng bài viết trên đã giúp bạn hiểu rõ hơn về Visual Studio là gì, những lợi ích và tính năng mà chúng sở hữu.

## Tìm Hiểu Ngôn Ngữ C#

Ngôn ngữ C# hiện nay được đa số các công ty, lập trình viên sử dụng tại Việt Nam, ngôn ngữ dễ dàng sử dụng hơn so với các ngôn ngữ khác và được hỗ trợ, đi kèm với các công nghệ không ngừng phát triển, ở chương này sẽ giới thiệu ngắn gọn về ngôn ngữ C#.

**Khái niệm về lập trình.**

Lập trình có thể được hiểu là: Đưa những gì ở ngoài thực tế vào bên trong máy tính, được máy tính xử lý và làm việc thay thế con người. Đề làm được đều đó phải thông qua các ngôn ngữ để cho máy tính có thể hiểu được, đó là ngôn ngữ lập trình.

**Hệ thống kiểu dữ liệu.**

Để có thể lưu trữ các thông tin cơ sở (*Primitive data*) cho quá trình xử lý của chương trình, C# xây dựng các kiểu dữ liệu cơ sở cho mục đích lưu trữ thông tin ở dạng số, ký tự và luận lý. Các kiểu dữ liệu cơ sở thường được xác định trước bởi kích thước mà nó sẽ được cấp phát trong bộ nhớ cũng như khả năng chứa dữ liệu của nó. Bạn hãy quan sát bảng mô tả sau thể hiện các kiểu dữ liệu cơ bản được cung cấp trong C#.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **STC Type** | **Size** | **Range** |
| Sbyte | System.SByte | 8 | -128..127 |
| Short | System.Int16 | 16 | (-32768 .. 32767) |
| Int | Sytem. Int32 | 32 | -231..231-1 |
| Long | Sytem. Int64 | 64 | -263..263-1 |
| Byte | System.SByte | 8 | 0..255 |
| Ushort | System.UInt16 | 16 | (0 .. 65535) |
| Uint | System.UInt32 | 32 | 0..232-1 |
| Ulong | System.UInt64 | 64 | 0..264-1 |
| Float | System.Single | 32 | xấp xỉ từ 3, 4E - 38 đến 3, 4E+38 |
| Double | System.Double | 64 | 1, 7E-308 đến 1, 7E+308 |
| decimal | System.Decimal | 128 | Có độ chính xác đến 28 con số |
| Bool | System.Boolean |  | Kiểu true/false |
| Char | System.Char | 16 | Ký tự unicode |

Một số điểm cần lưu ý:

- Đối với các ký tự thuộc dạng Unicode thông thường được mô tả bởi 1 giá trị 16 bits (2 bytes) để biểu thị cho thông tin dạng ngôn ngữ của các dân tộc khác nhau trên thế giới, khi sử dụng trong chương trình, ta phải sử dụng tiền tố “U” để trình biên dịch biết và sẽ dành cho các biến thuộc dạng này với kích thước 2 bytes trong bộ nhớ.

- Đối với dữ liệu thuộc kiểu số nguyên có dấu (Signed) thường dùng để lưu trữ các giá trị số ở dạng đại số là số dương (Positive) và số âm (Negative).

- Đối với dữ liệu thuộc kiểu số thực, các giá trị của nó luôn phải kết thúc bởi ký tự F hoặc f.

-Đối với dữ liệu dạng ký tự (char) luôn phải đặt trong cặp dấu nháy đơn.

Phân loại đối với dữ liệu trong C#:

- Object: Thường được xây dựng dựa trên nền tảng là các dữ liệu cơ sở ban đầu của C#, dữ liệu Object thường dùng để lưu trữ các thông tin phức tạp với các thành phần của nó là các kiểu dữ liệu khác trong chương trình.

- String: Cũng là một dạng dữ liệu được triển khai dựa trên dữ liệu cơ sở của C# dùng để lưu trữ thông tin ở dạng ngôn ngữ, chữ viết. Dữ liệu String thông thường là tập hợp các ký tự Unicode được trình bày 1 cách tuần tự, liên tục.

- Class: Đây là 1 dạng dữ liệu phức hợp, bên trong nó thông thường bao gồm các đối tượng thuộc các kiểu dữ liệu đã được định nghĩa trước đó (Có thể là dữ liệu cơ bản, có thể là dữ liệu String hoặc Object, …) cùng với các phương thức cho phép thao tác trên dữ liệu của class cho một mục tiêu cụ thể.

- Delegate: Đây cũng là một dạng dữ liệu phức hợp do người dùng tự định nghĩa trong chương trình của mình, dữ liệu thuộc dạng này thông thường lưu trữ sự tham chiếu đến các phương thức khác nhau trong chương trình.

- Interface: Trong chương trình của mình, đôi khi bạn cần phải đặt ra các nguyên tắc cho những thành phần cụ thể trong chương trình và buộc chính mình phải lập trình theo những nguyên tắc như vậy. Đó chính là lúc bạn tạo ra các interface cần thiết, giống như các kiểu dữ liệu phức hợp khác, Interface cũng do lập trình viên tự tạo ra để sử dụng trong chương trình của mình.

- Array: Thông thường đối với những dữ liệu cùng loại, lưu trữ cho cùng mục đích và bản thân chúng có những sự tương quan nhất định với nhau, bạn sẽ tạo ra dữ liệu dạng mảng để tiện truy xuất và xử lý theo một thuật toán nào đó. Ví dụ tạo mảng để lưu trữ thông tin các sinh viên, sau này có thể in ra danh sách, sắp xếp hoặc xác định điểm bình quân của cả lớp cho 1 môn theo yêu cầu.

**Đối tượng (Project)**

Đối tượng là một một gói phần mềm bao gồm các trạng thái và hành vi có liên quan. Đối tượng phần mềm thường được sử dụng để mô phỏng các đối tượng trong thế giới thực mà bạn thấy trong cuộc sống hàng ngày. Bài học này sẽ gải thích việc các trạng thái và hành vi được trình bày như thế nào trong một đối tượng, đồng thời giới thiệu khái niệm về bao gói dữ liệu cũng như giải thích lợi ích của việc thiết kế phần mềm theo cách này.

**Lớp (Class) trong C#**

Một lớp là một bản thiết kế hoặc bản mẫu mà từ đó chúng ta sẽ tạo ra các đối tượng. Trong phần này chúng ta sẽ định nghĩa một lớp nhằm mô phỏng các trạng thái và hành vi của một đối tượng trong thế giới thực. Chúng ta sẽ cố gắng tập trung vào những vấn đề cơ bản, giúp ta thấy được việc một lớp đơn giản sẽ mô phỏng các trạng thái và hành vi như thế nào.

**Kế thừa (Inheritance)**

Kế thừa cung cấp một cơ chế mạnh mẽ và tự nhiên để sắp xếp và tổ chức một phần mềm. Phần này sẽ giải thích việc làm thế nào mà các lớp có thể kế thừa các trạng thái và hành vi từ lớp cha của nó và đồng thời cũng trình bày cách để cho một lớp kế thừa từ một lớp khác bằng cách sữ dụng cú pháp đơn giản của ngôn ngữ lập trình Java.

**Đa hình (Polvmorphism)**

Tính đa hình được thể hiện trong lập trình hướng đối tượng rất đặc biệt. Người lập trình có thể định nghĩa một thuộc tính (chẳng hạn thông qua tên của các phương thức) cho một loạt các đối tượng gần nhau nhưng khi thi hành thì dùng cùng một tên gọi mà sự thi hành của mỗi đối tượng sẽ tự động xảy ra tương ứng theo từng đối tượng không bị nhầm lẫn.

Thí dụ: khi định nghĩa hai đối tượng "hinh\_vuong" và "hinh\_tron" thì có một phương thức chung là "chu\_vi". Khi gọi phương thức này thì nếu đối tượng là "hinh\_vuong" nó sẽ tính theo công thức khác với khi đối tượng là "hinh\_tron".

**Giao diện (Interfaces)**

Một giao diện là một bản giao ước giữa một lớp với thế giới bên ngoài. Khi một lớp triển khai một giao diện thì nó sẽ đảm bảo việc cung cấp các hành vi đã được trình bày bởi giao diện đó. Phần này sẽ định nghĩa một giao diện đơn giản và giải thích về những thay đổi mà một lớp bất kỳ cần phải tiến hành khi triển khai giao diện này.

**Mô hình chuyển giao (Delegates)**

Theo như định nghĩa được đưa ra trên MSDN thì delegate là một kiểu chứa tham chiếu đến một hoặc nhiều phương thức (method), có thể là phương thức của lớp (class’s method) hoặc là phương thức của đối tượng (object’s method). Đối với những ai đã làm việc với ngôn ngữ C hoặc C++ thì có thể thấy delegate gần giống với khái niệm **Con trỏ hàm (**funtion’s pointer). Tuy nhiên, hiện nay trong nhiều tài liệu vẫn hay dùng chung từ delegate để chỉ đến hai khái niệm khác nhau, gây ra sự mập mờ, đó là **Kiểu delegate** (delegate type) và **Thực thể delegate** (delegate instance, hay còn có thể gọi là **Đối tượng delegate**); Do đó, trong bài viết này chúng ta sẽ không sử dụng từ delegate với nghĩa chung như vậy mà sẽ phân biệt rạch ròi giữa kiểu delegate và đối tượng delegate.

Một kiểu delegate được khai báo bằng cách sử dụng từ khóa *delegate* đi kèm theo với kiểu dữ liệu trả về, tên và danh sách tham số của nó. Để một delegate có thể chứa tham chiếu đến một phương thức khác (hoặc đôi khi ta có thể nói là: delegate đại diện cho một phương thức) thì cả delegate và phương thức mà nó tham chiếu đến phải có cùng nguyên mẫu(signature), có nghĩa là cả hai đều phải có chung kiểu dữ liệu trả về và danh sách tham số.

Ta sẽ xét một ví dụ mà ở đó có một delegate với tên là Calculation được khai báo, kiểu delegate này có kiểu dữ liệu trả về là int, nhận vào hai tham số thuộc kiểu int; Một đối tượng của delegate Calculation được tạo ra và tham chiếu đến phương thức Add() của lớp Program, chúng ta có thể để ý rằng cả kiểu delegate Calculation và phương thức Add() có kiểu dữ liệu trả về và danh sách tham số giống nhau.

Kết luận: C# được thiết kế theo hướng này với phong cách phát triển đơn giản và nhất quán. Đặc biệt với những bổ sung không ngừng của C# cũng cho thấy đước sự mạnh mẽ của ngôn ngữ này.

## 3.3. Giới Thiệu winform c#

**Winform là gì?**

Winform là thuật ngữ mô tả một ứng dụng được viết dùng .NET FrameWorrk và có giao diện người dùng Windows Forms.

Mỗi màn hình windows cung cấp một giao diện giúp người dùng giao tiếp với ứng dụng. Giao diện này được gọi là giao diện đồ họa (GUI) của ứng dụng.

Là các ứng dụng windows chạy trên máy tính – mã lệnh thực thi ngay trên máy tính: Microsoft, Word, Excel, Access, Calculator, yahoo, Mail…  là các ứng dụng **Windows Forms**.

**Ưu điểm:**

Winform có tuổi thọ khá lâu đời, chính vì thế nên chúng bị khai tử. Nhưng ở Việt Nam thì chúng vẫn được tồn tại và phát triển. Lý do là vì những ưu điểm mà chúng mang lại.

Những lập trình viên C# ở đời đầu cũng đều được học và sử dụng về winform. Bởi vì:

Giao diện có thể kéo thả dễ học và sử dụng.

Gắn các event cho các button chỉ cần double click. Hỗ trợ nhiều event như click, hover,…

Viết code trực quan hơn: Có thể lấy text từ textbox và show dữ liệu bằng messagebox, kết nối grid bằng SQL.

Chính vì sự dễ code khi chỉ cần kéo thả nên có nhiều component có sẵn. Phù hợp với nhiều phần mềm thống kê, quản lý, tính tiền, … Thích hợp cho nhiều doanh nghiệp từ lớn cho đến nhỏ.

Những ưu điểm chung mà winform mang lại cho người dùng:

* Dễ học và dễ dạy
* Tốc độ xử lý dữ liệu nhanh và hiệu quả.
* Chạy được trên nhiều phiên bản của Windows khác nhau.
* Có thể thao tác với nhiều giao diện.

**Nhược điểm:**

Bên cạnh những ưu điểm của winform thì chúng cũng có nhiều nhược điểm chưa được khắc phục sau đây:

Phần mềm chỉ có thể chạy trên nền tảng windows nên khi muốn sử dụng bạn bắt buộc phải dùng phải cài đặt phần mềm cho máy tính. Khá bất tiện cho công việc, nên để đảm công việc được trơn thu bạn phải mang theo máy tính cá nhân để dự phòng.

Winform chỉ phù hợp cho ứng dụng trên desktop ( ứng dụng quản lý thông tin và tương tác trực tiếp với người dùng).

Đồ họa không cao nên giao diện sẽ thiếu đi tính trực quan, không thân thiện đối với người dùng.

# CHƯƠNG 4: XÂY DỰNG VÀ PHÁT TRIỂN ỨNG DỤNG

## 4.1 Thiết kế giao diện

**Tạo dự án với Winform App**

Sau khi cài đặt xong, hãy khởi động Visual Studio.

A screenshot of a computer

Description automatically generated with medium confidence

Hình 4. 1 Tạo dự án với Winform

**Thiết kế giao diện winform**

Trong hộp Toolbox, các công cụ thường có màu khá nổi bật, trong khi các biểu mẫu Windows bình thường chỉ có màu xám. Các công cụ sử dụng các công cụ tương tự như các công cụ hiện có thường có chữ Chỉnh sửa phía sau, ví dụ: chỉnh sửa nút đơn giản.

Graphical user interface, application

Description automatically generated

Hình 4. 2 Một số công cụ của Winform

Tùy theo yêu cầu và bố cục phác thảo sẽ thiết kế cho phù hợp.

## 4.2 Thiết kế database

Diagram

Description automatically generated

Hình 4. 3 Database

Diagram

Description automatically generated

Hình 4. 4 Lấy dữ liệu từ database (LINQ)

A picture containing graphical user interface

Description automatically generated

Hình 4. 5 Tải dữ liệu từ bảng vào DataGridView

Graphical user interface, text, application

Description automatically generated

Hình 4. 6 Thêm dữ liệu vào SQL

Graphical user interface, text, application

Description automatically generated

Hình 4. 7 Sửa dữ liệu

Graphical user interface, text, application

Description automatically generated

Hình 4. 8 Xóa dữ liệu

## 4.3 Thiết kế giao diện ứng dụng

## 4.3.1 Chức năng đăng nhập

Đăng nhập chương trình sử dụng tài khoản và mật khẩu do quản trị viên cấp:

Graphical user interface, application

Description automatically generated

*Hình 4.1 Đăng nhập*

Khi đăng nhập sai sẽ hiển thị thông báo:

Graphical user interface, text, application

Description automatically generated

*Hình 4.1.1 Khi đăng nhập sai*

## 4.3.2. Giao diện chính của chương trình

Đây là giao diện quản lí chính của chương trình bao gồm:

+ Nhân viên

+ Đồ uống

+ Hóa đơn

+ Thông tin bàn

+ Ca trực

+ Thống kê doanh thu

+ Đăng kí

+ Đăng xuất

A picture containing application

Description automatically generatedHình 4.2 Giao diện chinh của chương trình

## 4.3.3. Chức năng quản lí nhân viên

Chức năng thêm sửa xóa nhân viên hiện có trong quán

Graphical user interface, application

Description automatically generated

Hình 4.5 Giao diện quản lí nhân viên

## 4.3.4. Chức năng quản lí sản phẩm

Quản lí thêm, sửa, xóa sản phẩm hiện có hoặc nhập mới sản phẩm

Table

Description automatically generated

Hình 4.6 giao diện quản lí sản phẩm

## 4.3.5. Chức năng quản lí bàn

Chức năng này sẽ hiển thị bố cục bàn hiện trong quán, cũng như trạng thái bàn đang trống hay đã có người từ đó lên được hóa đơn cho từng bàn và tính tổng tiền cũng như thanh toán hóa đơn

Graphical user interface

Description automatically generated

Hình 4.7 Giao diện quản lí bàn ăn

## 4.3.6. Chức năng quản lí, thêm bớt bàn ăn

Quản lí bàn ăn (thêm, xóa, sửa) thông tin bàn

Graphical user interface

Description automatically generated

Hình 4.8 Giao diện quản lí số lượng bàn

## 4.3.7: Chức năng quản lí ca trực nhân viên

Quản lí ca trực của nhân viên, có 2 chức năng chính thêm và chọn ca trực

Khi ca trực được chọn thì ca trực trống hiện có sẽ bị xóa đi.

Graphical user interface

Description automatically generated

Hình 4.9 Giao diện quản lí ca trực

## 4.3.8. Chức năng đăng kí

Chức năng thêm mới tài khoản cho nhân viên trong quán, gồm 2 quyền chính là nhân viên và admin

Graphical user interface, application

Description automatically generated

Hình 4.10: Giao diện đăng kí

## 4.3.9. Chức năng thống kê

Chức năng thống kê doanh thu sản phẩm đã bán được trong từng tháng, hoặc có thể thống kê theo ngày

Table

Description automatically generatedHình 4.11 Giao diện thống kê doanh thu

## 4.3.10. Chức năng lập hóa đơn

Chức năng lên hóa đơn thanh toán cho bàn đã sử dụng dịch dụ của quán, gồm các thông tin cơ bản như:

* Sản phẩm đã sử dụng
* Đơn giá
* Số lượng
* Giảm giá
* Tổng tiền

Graphical user interface, text, application, email

Description automatically generated

Hình 4.12 Giao diện hóa đơn

## 4.3.11. Chức năng phân quyền

Khi đăng nhập vào với tài khoản admin thì sẽ có thể truy cập vào đầy đủ chức năng

A picture containing graphical user interface

Description automatically generated

Hình 4.13 Khi đăng nhập bằng tài khoản admin

Ngược lại khi đăng nhập với tài khoản nhân viên chỉ có thể truy cập vào 1 chức năng duy nhất là lập hóa đơn

A picture containing graphical user interface

Description automatically generated

Hình 4.14 Khi đăng nhập tài khoản nhân viên

## 4.4. Xây dựng các chức năng

Text

Description automatically generated

Hình 4. 9 Phương thức Login

Người dùng sẽ đăng nhập vào hệ thống bằng tên tài khoản và mật khẩu. Nếu đúng thì thực hiện đăng nhập. Nếu sai thì sẽ có thông báo lỗi.

Graphical user interface, text, application

Description automatically generated

Hình 4. 10 Phương thức hiển thị bàn

Chức năng hiển thị bàn theo database

Graphical user interface, text

Description automatically generated

Hình 4. 11 Phương thức hiển thị hóa đơn theo bàn

Chức năng hiển thị hóa đơn theo từng bàn, dựa theo số bàn

## 4.5 Kết quả chạy thử nghiệm

Ứng dụng đã được chạy thử nghiệm trên máy thật.

Các chức năng chính của kiểm thử:

* + Kiểm tra đăng nhập, đăng xuất.
  + Quản lí nhân viên, bàn ăn, sản phẩm.
  + Thêm, sửa, xóa dữ liệu.
  + Thanh toán hóa đơn.
  + Kiểm tra cập nhật trạng thái. Qua các thông hóa đơn theo bàn (Bàn trống, hay có người)

*Qua các bài kiểm tra thì theo em thấy ứng dụng chạy tương đối ổn định nhưng song vào đó vì ứng dụng được xây dựng bằng phiên biển .net dưới 5. Nên đôi lúc sẽ xảy ra hiện tượng lỗi khi nhập vào vùng trống, không có dữ liệu trên datagdview*

# ĐÁNH GIÁ - KẾT LUẬN

# *Nội dung này có 1 trang nên ko để thành chương, em cần viết lại phần này, phải nêu được Đồ án hoàn thành đúng thời gian, tiến độ và đạt được những mục tiêu đã đặt ra. Nêu cụ thể các kết quả đã đạt được, những hạn chế và hướng phát triển…xem các đồ án mẫu xem người ta viết kết luận như thế nào?*

## Kết luận

# Sau thời gian thực hiện đồ án tốt nghiệp, cùng với sự cố gắng và nỗ lực của bản thân đến nay đồ án của em đã hoàn thành đáp ứng được đầy đủ các chức năng của đề tài. Trong quá trình thực hiện đồ án, bản thân em đã đạt được nhiều tiến bộ cả về việc tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình, kỹ năng viết và trình bày báo cáo

* Các kết quả đã đạt được bao gồm :
* Chương trình có giao diện thân thiện, thuận tiện cho người sử dụng.
* Hỗ trợ nhiều chức năng phù hợp với thực tế của việc tìm kiếm thông tin, trao đổi và quản lý quán cafe.
* Có nhiều chức năng xử lý linh hoạt các yêu cầu thực tế. Ví dụ như:

+ Cung cấp khả năng thống kê, báo cáo linh hoạt có tính chất tuỳ chọn theo những tiêu chí cụ thể.

+ Chức năng quản lý đăng nhập theo nhóm người dùng hỗ trợ tương đối tốt cho việc phân phối công việc theo chức năng và nhiệm vụ của các bộ phận chuyên trách trong hệ thống quản lý quán cafe.

+ Chức năng nhập và quản lý nhân viên được thực hiện logic đảm bảo cho dữ liệu được hợp lệ hoá và cập nhật chính xác.

* Dễ dàng sử dụng và quản lý.
* Cơ sở dữ liệu tương đối lớn và tính bảo mật cao.
* Một số kĩ chức năng vẫn chưa thực hiện được :
* Do thời gian nghiên cứu có hạn và khả năng còn hạn chế nên một số vẫn chỉ còn trên ý tưởng, chưa được thực thi.
* Một số chức năng còn chưa hoạt động ổn định.
* Chưa giải quyết chọn vẹn các vấn đề nảy sinh trong quá trình quản lý.
* Chương trình chưa có tính thẩm mỹ chưa cao
* **Hướng phát triển của phần mềm :**

# Để tiếp tục phát triển đề tài này trong thực tế em nhận thấy cần phải tiếp thực thực hiện các công việc sau:

# • Xử lý các lỗi chặt chẽ hơn.

# • Xử lý thêm phần đăng nhập và đăng xuất như: kiểm tra đăng nhập khi có người dùng khác đăng nhập vào tài khoản, cho phép lưu tài khoản để tự động đăng nhập lần tiếp theo

# • Nâng cấp giao diện.

# TÀI LIỆU THAM KHẢO

[ 1] [(PDF) The C# Player's Guide, 3rd Edition (thefreestudy.com)](https://www.thefreestudy.com/pdf-the-c-players-guide-3rd-edition/).

[ 2] [CLR via C# - Jeffrey Richter - Google Sách](https://books.google.com.vn/books/about/CLR_via_C.html?id=QMdCf_mm55cC&redir_esc=y&hl=vi)

[ 3] [Các Giải Pháp Lập Trình C# (NXB Giao Thông Vận Tải 2006) - Nguyễn Ngọc Bình Phương, 706 Trang - Ebook VNDOC (vndoc-ebook.com)](https://vndoc-ebook.com/cac-giai-phap-lap-trinh-c-nxb-giao-thong-van-tai-2006-nguyen-ngoc-binh-phuong-706-trang/)

[ 4] [Beginning Visual C# 2010 - Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon D. Reid, Morgan Skinner - Google Sách](https://books.google.com.vn/books/about/Beginning_Visual_C_2010.html?id=xFhZME_NbxsC&redir_esc=y)

[ 5] [Winform with C# - giáo trình của FPT-Aptech - phamphattien (google.com)](https://sites.google.com/a/ctu.edu.vn/phamphattien/departments/sample-dept-1/winformwithc-giaotrinhcuafpt-aptech)

[ 6] [Beginning SQL Server for Developers | SpringerLink](https://link.springer.com/book/10.1007/978-1-4842-0280-7?page=1)

[ 7] [Lập trình viên .NET 2021 - Trung Tâm Tin Học Trường Đại học Khoa học Tự nhiên (csc.edu.vn)](https://csc.edu.vn/lap-trinh-va-csdl/Lap-trinh-vien-NET-2021_214)

[ 8] [Giáo Trình Lập Trình Windows Form Với C# .NET Tập 1 (NXB Thông Tin 2012) - Lê Trung Hiếu, 180 Trang | Sách Việt Nam (vietbooks.info)](https://vietbooks.info/threads/giao-trinh-lap-trinh-windows-form-voi-c-net-tap-1-nxb-thong-tin-2012-le-trung-hieu-180-trang.43807/)

[ 10] <https://freetuts.net/c-sharp-la-gi-tong-quan-ve-c-sharp-1045.html>