

ĐỒ ÁN HỌC PHẦN
LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

APP BÁN CÁ CẢNH

Ngành: **Công Nghệ Thông Tin**

Chuyên ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn: ThS. Nguyễn Chí Toàn

Sinh viên thực hiện:

Dương Vĩnh Khiêm MSSV: 2180603833

Lớp: 21DTHC4

Dương Gia Linh MSSV: 2180602513

Lớp: 21DTHC4

Nguyễn Hoàng Duy MSSV: 2180607029

Lớp: 21DTHC4

Phan Bảo Long MSSV: 2180606972

Lớp: 21DTHC4

TP.Hồ Chí Minh, 2024

LỜI MỞ ĐẦU

Trong kỷ nguyên hiện đại, công nghệ thông tin đã phát triển nhanh chóng và được ứng dụng rộng rãi trong mọi lĩnh vực trên toàn thế giới. Việt Nam cũng đã đạt được nhiều thành tựu trong việc ứng dụng công nghệ vào cuộc sống và kinh doanh, góp phần quan trọng vào sự phát triển kinh tế xã hội và nâng cao chất lượng cuộc sống của người dân. Trong bối cảnh này, công nghệ thông tin đã trở thành một yếu tố không thể thiếu đối với bất kỳ tổ chức hay doanh nghiệp nào, đặc biệt trong việc quản lý thông tin và tối ưu hóa các quy trình kinh doanh.

Ngành nuôi cá cảnh tại Việt Nam đang phát triển vượt bậc, không chỉ là một sở thích mà còn là một lĩnh vực kinh doanh tiềm năng. Tuy nhiên, việc quản lý và kinh doanh cá cảnh truyền thống gặp nhiều thách thức như tiếp cận khách hàng, quản lý sản phẩm và dịch vụ hậu mãi. Để giải quyết những khó khăn này, việc áp dụng công nghệ thông tin trong lĩnh vực kinh doanh cá cảnh trở nên cần thiết và cấp bách.

Hiện nay, các thiết bị di động như điện thoại thông minh và máy tính bảng đã trở nên phổ biến và không thể thiếu trong cuộc sống hàng ngày. Với sự phát triển nhanh chóng của công nghệ, nhu cầu giải trí và dịch vụ trên các thiết bị số cũng tăng cao. Điều này đã mở ra những cơ hội lớn cho ngành kinh doanh cá cảnh, khi mà việc mua sắm và quản lý sản phẩm có thể thực hiện dễ dàng và tiện lợi hơn qua các nền tảng trực tuyến.

Nhận thấy tiềm năng và nhu cầu của thị trường, nhóm chúng em đã nghiên cứu và quyết định triển khai dự án “App bán cá cảnh”. Dưới sự hướng dẫn của ThS. Nguyễn Chí Toàn, chúng em đã áp dụng những kiến thức học được để xây dựng một hệ thống quản lý và bán hàng trực tuyến chuyên nghiệp cho ngành cá cảnh. Hệ thống này không chỉ giúp các chủ cửa hàng dễ dàng quản lý sản phẩm, đơn hàng mà còn tạo điều kiện thuận lợi cho khách hàng trong việc tìm kiếm, mua sắm và nhận được sự hỗ trợ tốt nhất.

Chúng em hy vọng rằng dự án này sẽ góp phần nâng cao hiệu quả kinh doanh, tiết kiệm thời gian và công sức cho các doanh nghiệp, đồng thời mang đến trải nghiệm mua sắm tiện lợi và thú vị cho khách hàng. Do thời gian và kiến thức của nhóm còn hạn

chế, dự án không tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý của các thầy cô và các bạn để dự án được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

Dương Vĩnh Khiêm

Dương Gia Linh

Nguyễn Hoàng Duy

Phan Bảo Long

LỜI CẢM ƠN

Trước hết, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô Khoa Công Nghệ Thông Tin HUTECH đã tận tình hỗ trợ và truyền đạt những kiến thức quý báu về Công Nghệ Thông Tin, giúp chúng em có nền tảng vững chắc để thực hiện đồ án này.

Chúng em cũng xin bày tỏ lòng biết ơn sâu sắc đến thầy hướng dẫn, người đã trực tiếp giúp đỡ và tạo điều kiện thuận lợi, cung cấp cho chúng em những kiến thức và công nghệ cần thiết để hoàn thành đồ án. Sự hướng dẫn nhiệt tình và tận tâm của thầy đã góp phần quan trọng vào sự thành công của dự án.

Cuối cùng, chúng em xin cảm ơn gia đình, các anh chị và các bạn đã luôn động viên, hỗ trợ chúng em trong suốt quá trình học tập, nghiên cứu và thực hiện đề tài. Sự ủng hộ và khích lệ từ mọi người là nguồn động lực lớn lao giúp chúng em vượt qua mọi khó khăn.

Chúng em xin kính chúc Ban Giám Hiệu nhà trường cùng quý thầy cô sức khỏe dồi dào, luôn vui vẻ và đạt nhiều thành công trong công việc.

TP. Hồ Chí Minh, ngày 04 tháng 01 năm 2025

LỜI CAM ĐOAN

Chúng em xin cam kết rằng toàn bộ nội dung của đồ án chuyên ngành “App bán cá cảnh” là kết quả của sự nỗ lực và làm việc của nhóm chúng em. Những vấn đề được trình bày trong báo cáo này phản ánh kết quả của quá trình học tập, nghiên cứu và làm việc chăm chỉ của nhóm. Tất cả các tài liệu tham khảo đều được trích dẫn rõ ràng và tuân thủ đúng quy định.

Chúng em xin chịu hoàn toàn trách nhiệm về tính trung thực và chính xác của các nội dung được trình bày trong báo cáo này.

TP. Hồ Chí Minh, ngày 04 tháng 01 năm 2025

Nhóm cam đoan

MỤC LỤC

LỜI MỞ ĐẦU	2
LỜI CẢM ƠN.....	4
LỜI CAM ĐOAN	5
MỤC LỤC	6
CHƯƠNG 1 TỔNG QUAN	9
1.1. KHẢO SÁT THỰC TRẠNG.....	9
1.2. TÍNH KHẢ THI CỦA BÀI TOÁN.....	10
1.2.1. Quản lý thông tin khách hàng.....	10
1.2.2. Quản lý sản phẩm và đơn hàng.....	10
1.2.3. Quản lý dịch vụ bổ sung	11
1.2.4. Triển khai các ứng dụng	11
1.3. ĐỐI TƯỢNG NGHIÊN CỨU SỬ DỤNG	11
1.3.1. Khách hàng tiềm năng và khách hàng hiện tại	11
1.3.2. Nhân viên bán hàng và quản lý sản phẩm	11
1.3.3. Quản lý cửa hàng	12
1.4. MỤC TIÊU NGHIÊN CỨU	12
1.4.1. BackEnd.....	12
1.4.2. Cơ Sở Dữ Liệu.....	12
1.4.3. Quản Lý Dự Án	12
1.5. PHẠM VI GIỚI HẠN.....	12
1.6. Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN CỦA ĐỀ TÀI	13
1.7. CẤU TRÚC CỦA ĐỒ ÁN.....	13
1.7.1. Phát triển cơ sở dữ liệu	13
1.7.2. Phát triển hệ thống quản lý bán cá cảnh	14
1.7.3. Sử dụng ASP.NET CORE và Identity Framework	14
1.7.4. Quản lý dự án.....	14
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	15
2.1. GIỚI THIỆU VỀ NGÔN NGỮ SỬ DỤNG.....	15
2.1.1. Giới thiệu ngôn ngữ lập trình C#.....	15
2.1.2. Giới thiệu về DART.....	17
2.2. GIỚI THIỆU VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER	18

2.3. CÔNG CỤ SỬ DỤNG.....	19
2.3.1. Giới thiệu về IDE – Visual Studio 2022	19
2.3.2. Giới thiệu về GitHub	21
2.3.3. Giới thiệu về Draw.io.....	23
2.4. MÔ HÌNH VÀ KỸ THUẬT	25
2.4.1. Giới thiệu về EntityFramework	25
2.4.2. Giới thiệu về ASP.NET Identity.....	26
Các khái niệm cơ bản trong GET	35
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	36
3.1. MÔ TẢ NGHIỆP VỤ.....	36
3.1.1. Mô tả chung	36
3.1.2. Quản lý tài khoản người dùng	36
3.1.3. Quản lý sản phẩm	36
3.1.4. Mua hàng	37
3.1.5. Quản lý đơn hàng.....	37
3.2. CÁC LOẠI THỰC THỂ	37
3.2.1. Cơ bản.....	37
3.2.2. Đối tượng ngoài	38
3.2.3. Nghiệp vụ.....	38
3.3. MÔ HÌNH ERD (ENTITY -RELATIONSHIP DIAGRAM)	38
3.3.1. Phân tích mô hình ERD	38
3.4. MÔ HÌNH QUAN HỆ	38
3.4.1. Mô tả các quan hệ	38
3.4.2. Danh sách thông tin các bảng	40
3.5. SƠ ĐỒ USE CASE.....	44
Các Use Case được thiết kế cho hệ thống	44
CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM	45
4.1. GIAO DIỆN API.....	45
CHƯƠNG 5 KẾT LUẬN VỀ KIẾN NGHỊ	60
5.1. KẾT LUẬN CHUNG.....	60
5.1. Kết luận chung	60
5.2. Kết quả đạt được	60
5.3. Kết quả chưa đạt được.....	60
5.4. Hướng phát triển	60

TÀI LIỆU THAM KHẢO	62
--------------------------	----

CHƯƠNG 1 TỔNG QUAN

1.1. KHẢO SÁT THỰC TRẠNG

Trong thời đại công nghệ phát triển nhanh chóng, việc áp dụng công nghệ thông tin (CNTT) vào các lĩnh vực kinh doanh đang trở nên phổ biến và mang lại nhiều lợi ích đáng kể. Tại Việt Nam, CNTT đang được đầu tư mạnh mẽ, giúp nâng cao chất lượng dịch vụ và mở ra nhiều cơ hội kinh doanh mới. Ngành nuôi và kinh doanh cá cảnh cũng không ngoại lệ, khi mà việc áp dụng công nghệ thông tin có thể giải quyết nhiều thách thức trong quản lý và vận hành. Hiện nay, nhiều cửa hàng kinh doanh cá cảnh tại Việt Nam vẫn quản lý thông tin khách hàng và sản phẩm một cách thủ công hoặc bán tự động. Điều này dẫn đến nhiều khó khăn như quản lý thông tin khách hàng. Việc lưu trữ thông tin khách hàng bằng sổ sách hoặc file đơn lẻ dễ dẫn đến sai sót và mất mát dữ liệu, gây khó khăn trong việc chăm sóc khách hàng và theo dõi lịch sử mua hàng. Thêm vào đó, quản lý sản phẩm và dịch vụ không hiệu quả do thông tin về các loại cá, tình trạng hàng tồn kho không được quản lý chặt chẽ. Điều này dẫn đến lãng phí và không hiệu quả. Cuối cùng, tính bảo mật và kết nối cũng là một thách thức lớn, khi dữ liệu lưu trữ trên giấy tờ hoặc các file đơn giản thiếu tính bảo mật và dễ bị rò rỉ. Các bộ phận như bán hàng, chăm sóc khách hàng và quản lý kho không có sự liên kết chặt chẽ, gây ra sự thiếu đồng bộ và hiệu quả trong quản lý.

Trước những hạn chế của phương pháp quản lý truyền thống, nhu cầu hiện đại hóa và số hóa trong quản lý kinh doanh cá cảnh trở nên cấp thiết. Hệ thống cần có khả năng quản lý thông tin khách hàng chi tiết và chính xác, từ thông tin cá nhân, lịch sử mua hàng đến các yêu cầu đặc biệt. Hệ thống cũng cần cung cấp giao diện thân thiện, cho phép khách hàng dễ dàng đặt hàng trực tuyến, kiểm tra tình trạng sản phẩm.. Một số cửa hàng cá cảnh lớn tại Việt Nam đã bắt đầu áp dụng các hệ thống quản lý hiện đại để cải thiện dịch vụ và quản lý hiệu quả hơn. Tuy nhiên, nhiều cửa hàng vừa và nhỏ vẫn chưa có điều kiện hoặc chưa nhận thức đầy đủ về lợi ích của việc số hóa quản lý. Việc triển khai một hệ thống quản lý kinh doanh cá cảnh thông minh và hiệu quả có thể mang lại nhiều lợi ích, bao gồm tăng cường hiệu quả quản lý, nâng cao chất lượng dịch vụ, tăng cường khả năng cạnh tranh và giúp cửa hàng tạo ra sự khác biệt và thu hút khách hàng hơn so với các đối thủ cạnh tranh.

Hiện nay, có nhiều giải pháp công nghệ hiện đại được áp dụng trong quản lý kinh doanh, bao gồm hệ thống quản lý tích hợp nhiều chức năng từ đặt hàng, quản lý dịch vụ, quản lý nhân sự đến quản lý tài chính; hệ thống quản lý quan hệ khách hàng (CRM) giúp quản lý thông tin khách hàng, nâng cao trải nghiệm khách hàng và tăng cường khả năng giữ chân khách hàng; các ứng dụng di động cho phép khách hàng đặt hàng, kiểm tra tình trạng sản phẩm và sử dụng các dịch vụ của cửa hàng một cách tiện lợi; công nghệ AI và IoT giúp tối ưu hóa hoạt động quản lý, từ việc dự đoán nhu cầu khách hàng đến quản lý tiết kiệm năng lượng và bảo trì thiết bị. Việc áp dụng hệ thống quản lý kinh doanh cá cảnh hiện đại mang lại nhiều lợi ích, bao gồm tăng cường hiệu quả quản lý, nâng cao chất lượng dịch vụ, bảo mật thông tin và tăng cường khả năng cạnh tranh. Trước những thách thức và nhu cầu cấp thiết trong quản lý kinh doanh, việc áp dụng công nghệ thông tin và triển khai các hệ thống quản lý hiện đại là một bước đi cần thiết. Điều này không chỉ giúp nâng cao hiệu quả quản lý mà còn mang lại nhiều lợi ích cho khách hàng và tạo nên sự khác biệt trong môi trường cạnh tranh ngày càng khốc liệt. Khảo sát thực trạng hiện tại cho thấy việc số hóa và hiện đại hóa quản lý kinh doanh cá cảnh sẽ là một xu hướng tất yếu và mang lại nhiều tiềm năng phát triển trong tương lai.

1.2. TÍNH KHẢ THI CỦA BÀI TOÁN

Qua khảo sát thực tế, chúng em nhận thấy rằng việc thiết lập hệ thống bán cá cảnh có thể giải quyết các vấn đề sau:

1.2.1. Quản lý thông tin khách hàng

Hệ thống sẽ quản lý được thông tin chi tiết của khách hàng từ lúc mua hàng đến khi hoàn tất giao dịch. Các thông tin bao gồm: thông tin cá nhân, lịch sử mua hàng, các yêu cầu đặc biệt.

Việc cập nhật thông tin khách hàng sẽ được thực hiện định kỳ để đảm bảo dữ liệu luôn được cập nhật và chính xác, giúp cải thiện dịch vụ chăm sóc khách hàng và tăng cường khả năng giữ chân khách hàng.

1.2.2. Quản lý sản phẩm và đơn hàng

Hệ thống sẽ quản lý thông tin về các loại cá cảnh, tình trạng hàng tồn kho, giá cả và các dịch vụ đi kèm. Điều này giúp tối ưu hóa việc sử dụng kho và giảm thiểu tình trạng hàng tồn kho không cần thiết.

Quản lý đơn hàng sẽ được tự động hóa, cho phép khách hàng tự đặt hàng trực tuyến, kiểm tra tình trạng hàng hóa. Điều này giúp tiết kiệm thời gian và giảm thiểu sai sót trong quá trình đặt hàng.

1.2.3. Quản lý dịch vụ bổ sung

Thông tin về dịch vụ sẽ được cập nhật đầy đủ và chính xác, giúp khách hàng dễ dàng lựa chọn và sử dụng các dịch vụ.

1.2.4. Triển khai các ứng dụng

Hệ thống sẽ được xây dựng dựa trên các mô hình đã được phân tích kỹ lưỡng và tối ưu hóa để phát triển thông qua ASP.NET CORE và MySQL. Điều này giúp triển khai dễ dàng trên các nền tảng Website và ứng dụng di động.

ASP.NET CORE cung cấp khả năng xây dựng các ứng dụng web mạnh mẽ và linh hoạt, đồng thời dễ dàng tích hợp với MySQL để quản lý cơ sở dữ liệu hiệu quả.

1.3. ĐỐI TƯỢNG NGHIÊN CỨU SỬ DỤNG

Hệ thống bán cá cảnh, đối tượng nghiên cứu và sử dụng bao gồm các thành phần sau:

1.3.1. Khách hàng tiềm năng và khách hàng hiện tại

Khách hàng tiềm năng: Những người tìm kiếm thông tin về cá cảnh và các dịch vụ liên quan để mua sắm. Họ cần một giao diện trực quan và dễ sử dụng để tìm kiếm, xem sản phẩm.

Khách hàng hiện tại: Những người đã mua hàng và sử dụng dịch vụ tại cửa hàng. Họ cần khả năng đăng nhập với vai trò là khách hàng, yêu cầu dịch vụ bổ sung và liên lạc với cửa hàng qua hệ thống.

1.3.2. Nhân viên bán hàng và quản lý sản phẩm

Nhân viên bán hàng: Quản lý thông tin đơn hàng, kiểm tra tình trạng hàng hóa và hỗ trợ khách hàng trong quá trình mua sắm và sử dụng dịch vụ.

Quản lý sản phẩm: Cập nhật tình trạng hàng tồn kho và xử lý các yêu cầu đặc biệt từ khách hàng.

1.3.3. *Quản lý cửa hàng*

Theo dõi hoạt động hàng ngày: Quản lý doanh thu và phân tích dữ liệu khách hàng. Cung cấp báo cáo chi tiết về tình trạng hàng tồn kho, doanh thu và các chỉ số hiệu suất khác.

1.4. MỤC TIÊU NGHIÊN CỨU

Để triển khai hệ thống quản lý và bán cá cảnh một cách hiệu quả, nhóm chúng em đã sử dụng các công nghệ và công cụ sau:

1.4.1. *BackEnd*

ASP.NET CORE và C#: Chúng em đã nghiên cứu và sử dụng kiến thức cơ bản về ASP.NET CORE kết hợp với ngôn ngữ lập trình C# để xây dựng các thành phần của ứng dụng app. ASP.NET CORE cung cấp một kiến trúc mạnh mẽ và linh hoạt giúp chúng em dễ dàng quản lý và phát triển hệ thống.

ASP.NET Identity Framework: Hệ thống sử dụng ASP.NET Identity Framework để quản lý và phân quyền người dùng. Identity Framework cung cấp các tính năng mạnh mẽ như đăng nhập, đăng ký, quản lý vai trò và quyền hạn, giúp chúng em dễ dàng kiểm soát quyền truy cập của người dùng trong hệ thống.

1.4.2. *Cơ Sở Dữ Liệu*

MySQL: Chúng em sử dụng MySQL, hệ quản trị cơ sở dữ liệu của Oracle Corporation, để quản lý dữ liệu cho hệ thống. MySQL cung cấp các tính năng mạnh mẽ và linh hoạt giúp chúng em dễ dàng thao tác và quản lý dữ liệu một cách hiệu quả.

1.4.3. *Quản Lý Dự Án*

GitHub: Để quản lý các công việc cho từng thành viên trong dự án và quản lý source code, chúng em đã tìm hiểu và sử dụng GitHub. GitHub giúp chúng em theo dõi tiến độ dự án, quản lý version code và dễ dàng hợp tác làm việc nhóm.

1.5. PHẠM VI GIỚI HẠN

Hệ thống sẽ không bao gồm các tính năng phức tạp như tích hợp hệ thống thanh toán quốc tế hoặc các công cụ quản lý tài chính phức tạp.

Hệ thống sẽ không tập trung vào các tính năng marketing nâng cao hoặc các công cụ phân tích dữ liệu phức tạp.

Phạm vi nghiên cứu sẽ tập trung chính vào việc phát triển website để cung cấp một nền tảng bán cá cảnh trực tuyến, hỗ trợ quản lý và mua bán các sản phẩm và dịch vụ liên quan.

1.6. Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN CỦA ĐỀ TÀI

Ý nghĩa khoa học của đề tài là áp dụng công nghệ và phương pháp xây dựng phát triển hệ thống để cải thiện quá trình quản lý và kinh doanh cá cảnh. Hệ thống quản lý bán cá cảnh giúp các cửa hàng dễ dàng quản lý thông tin khách hàng, sản phẩm, và đơn hàng. Việc áp dụng công nghệ ASP.NET CORE, Identity Framework, và MySQL vào hệ thống này không chỉ nâng cao hiệu quả quản lý mà còn tạo ra các giải pháp kỹ thuật tiên tiến trong việc xử lý và quản lý dữ liệu.

Thực tiễn của đề tài được thể hiện qua việc cung cấp một nền tảng trực tuyến tiện lợi cho quá trình mua bán và quản lý cá cảnh. Hệ thống bán cá cảnh trực tuyến tạo điều kiện thuận lợi cho việc quản lý sản phẩm, đơn hàng, và dịch vụ. Điều này giúp các cửa hàng tối ưu hóa quy trình kinh doanh, giảm thiểu sai sót, và tiết kiệm thời gian. Đồng thời, khách hàng có thể dễ dàng truy cập thông tin sản phẩm, đặt hàng trực tuyến, và sử dụng các dịch vụ bổ sung một cách thuận tiện, nâng cao trải nghiệm mua sắm và tăng cường sự hài lòng của khách hàng.

Tổng quan, đề tài hệ thống quản lý bán cá cảnh mang ý nghĩa quan trọng trong việc nâng cao quá trình quản lý và tạo môi trường dịch vụ trực tuyến hiệu quả cho các cửa hàng cá cảnh, đồng thời góp phần thúc đẩy sự phát triển của công nghệ và ứng dụng web trong lĩnh vực kinh doanh cá cảnh.

1.7. CẤU TRÚC CỦA ĐỒ ÁN

Đồ án tập trung vào các quá trình nghiên cứu và phát triển như sau:

1.7.1. Phát triển cơ sở dữ liệu

Tối ưu hóa các thành phần trong hệ thống: Đảm bảo rằng dữ liệu phản ánh thực tế một cách chính xác nhất. Điều này bao gồm việc tiến hành các khảo sát và phân tích

dữ liệu để thu thập thông tin chi tiết cho mỗi thực thể như khách hàng, sản phẩm (cá cảnh), dịch vụ bổ sung và đơn hàng.

1.7.2. Phát triển hệ thống quản lý bán cá cảnh

– Quản lý thông tin khách hàng

Hệ thống sẽ quản lý thông tin cá nhân, lịch sử mua hàng, yêu cầu đặc biệt và phản hồi của khách hàng. Điều này giúp cải thiện dịch vụ chăm sóc khách hàng và tăng cường khả năng giữ chân khách hàng.

– Quản lý sản phẩm và đơn hàng

Hệ thống sẽ quản lý thông tin về các loại cá cảnh, tình trạng hàng tồn kho (còn hàng, đã bán, đang bảo trì), giá cả và các dịch vụ đi kèm. Hệ thống cho phép khách hàng tự đặt hàng trực tuyến, kiểm tra tình trạng hàng hóa.

– Quản lý dịch vụ bổ sung

Thông tin về dịch vụ sẽ được cập nhật đầy đủ và chính xác, giúp khách hàng dễ dàng lựa chọn và sử dụng các dịch vụ. Hệ thống cũng sẽ hỗ trợ tính toán chi phí và thanh toán dịch vụ bổ sung một cách chính xác và minh bạch.

1.7.3. Sử dụng ASP.NET CORE và Identity Framework

ASP.NET CORE: Sử dụng ASP.NET CORE để phát triển hệ thống, cung cấp một kiến trúc mạnh mẽ và linh hoạt giúp quản lý và phát triển ứng dụng một cách hiệu quả.

Identity Framework: Sử dụng Identity Framework để quản lý và phân quyền người dùng. Hệ thống sẽ hỗ trợ các tính năng như đăng nhập, đăng ký, quản lý vai trò và quyền hạn, giúp dễ dàng kiểm soát quyền truy cập của người dùng.

1.7.4. Quản lý dự án

GitHub: Sử dụng GitHub để quản lý các công việc cho từng thành viên trong dự án và quản lý source code. GitHub giúp theo dõi tiến độ dự án, quản lý version code và dễ dàng hợp tác làm việc nhóm.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1. GIỚI THIỆU VỀ NGÔN NGỮ SỬ DỤNG

2.1.1. Giới thiệu ngôn ngữ lập trình C#

- C# là gì?

C# (phát âm là "C-sharp") là một ngôn ngữ lập trình hướng đối tượng hiện đại, mạnh mẽ, được phát triển bởi Microsoft trong khuôn khổ của nền tảng .NET. Ngôn ngữ này được tạo ra bởi Anders Hejlsberg và nhóm của ông, và lần đầu tiên xuất hiện vào năm 2000 với phiên bản .NET Framework 1.0. C# kết hợp các ưu điểm của nhiều ngôn ngữ lập trình khác như C++, Java và Visual Basic, nhằm tạo ra một ngôn ngữ dễ học, dễ sử dụng nhưng vẫn mạnh mẽ và linh hoạt.

- Đặc điểm chính của C#

Hướng đối tượng (Object-Oriented): C# hỗ trợ hoàn toàn lập trình hướng đối tượng với các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), đóng gói (encapsulation), và trừu tượng (abstraction).

An toàn kiểu (Type-Safe): C# kiểm tra chặt chẽ kiểu dữ liệu trong thời gian biên dịch, giúp giảm thiểu lỗi do không tương thích kiểu dữ liệu.

Quản lý bộ nhớ tự động (Automatic Memory Management): Với cơ chế garbage collection, C# tự động quản lý việc cấp phát và giải phóng bộ nhớ, giúp lập trình viên không cần quan tâm nhiều đến vấn đề quản lý bộ nhớ thủ công.

Tích hợp mạnh mẽ với .NET: C# được thiết kế để làm việc chặt chẽ với .NET Framework, cung cấp một thư viện phong phú và các dịch vụ như ASP.NET cho phát triển web, ADO.NET cho truy cập dữ liệu, và Windows Forms cho ứng dụng giao diện người dùng.

Ngôn ngữ linh hoạt: C# hỗ trợ nhiều phong cách lập trình khác nhau như lập trình hàm (functional programming), lập trình đa nhiệm (multithreading), và lập trình sự kiện (event-driven programming).

- Ứng dụng của C#

C# được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ phát triển ứng dụng doanh nghiệp đến các dự án cá nhân. Dưới đây là một số ứng dụng phổ biến của C#:

Phát triển ứng dụng Windows: C# là một trong những ngôn ngữ chính để phát triển các ứng dụng trên nền tảng Windows với sự hỗ trợ của Windows Forms, WPF (Windows Presentation Foundation), và UWP (Universal Windows Platform).

Phát triển web: ASP.NET, một phần của .NET Framework, cho phép lập trình viên sử dụng C# để xây dựng các ứng dụng web mạnh mẽ, an toàn và hiệu quả.

Phát triển ứng dụng di động: Với Xamarin, một nền tảng thuộc Microsoft, lập trình viên có thể sử dụng C# để phát triển các ứng dụng di động cho iOS, Android, và Windows.

Phát triển trò chơi: C# là ngôn ngữ chính được sử dụng trong Unity, một trong những nền tảng phát triển trò chơi phổ biến nhất hiện nay. Unity hỗ trợ phát triển trò chơi 2D và 3D trên nhiều nền tảng khác nhau như PC, console, và di động.

Phát triển ứng dụng đám mây: Với Azure, nền tảng đám mây của Microsoft, C# được sử dụng để xây dựng các dịch vụ đám mây, microservices và các giải pháp đám mây khác.

Phát triển hệ thống nhúng: C# cũng được sử dụng trong một số hệ thống nhúng và IoT (Internet of Things), nơi yêu cầu các giải pháp phần mềm mạnh mẽ và hiệu quả.

- **Ưu điểm của C#**

Hỗ trợ mạnh mẽ từ Microsoft: C# và .NET được phát triển và duy trì bởi Microsoft, đảm bảo có tài liệu phong phú, các công cụ hỗ trợ chất lượng cao và sự cập nhật thường xuyên.

Hiệu suất cao: C# cung cấp hiệu suất tốt nhờ khả năng biên dịch mã nguồn sang mã máy trung gian (IL) và sau đó biên dịch Just-In-Time (JIT) để thực thi.

Tính an toàn và ổn định: Các tính năng như an toàn kiểu (type-safety), quản lý bộ nhớ tự động và xử lý lỗi (exception handling) giúp giảm thiểu các lỗi phần mềm và tăng độ ổn định.

Thư viện phong phú: .NET Framework và .NET Core cung cấp nhiều thư viện và API phong phú giúp tăng tốc quá trình phát triển phần mềm.

Đa nền tảng: Với .NET Core và Xamarin, C# có thể được sử dụng để phát triển ứng dụng trên nhiều nền tảng khác nhau như Windows, Linux, macOS, iOS, và Android.

Tích hợp tốt với các công nghệ khác của Microsoft: C# tích hợp tốt với các công nghệ như Azure, MySQL, và các công cụ phát triển như Visual Studio.

Cộng đồng lớn và tài liệu phong phú: C# có một cộng đồng lập trình viên lớn và nhiều tài liệu học tập, giúp việc học và giải quyết vấn đề trở nên dễ dàng hơn.

Nhược điểm của C#

Phụ thuộc vào môi trường Microsoft: Mặc dù C# đã trở nên đa nền tảng hơn với .NET Core, nhưng nó vẫn phụ thuộc nhiều vào các công nghệ và dịch vụ của Microsoft.

Chi phí cao cho một số công cụ: Một số công cụ và dịch vụ liên quan đến C# và .NET, như Visual Studio Professional hay Enterprise, có chi phí khá cao.

Độ phức tạp: Mặc dù C# được thiết kế để dễ học, nhưng việc khai thác toàn bộ sức mạnh của ngôn ngữ và nền tảng .NET đòi hỏi kiến thức sâu rộng và kinh nghiệm.

Hiệu suất so với ngôn ngữ cấp thấp: Mặc dù C# có hiệu suất cao, nhưng vẫn có thể thua kém so với các ngôn ngữ lập trình cấp thấp như C hoặc C++ trong một số ứng dụng yêu cầu tối ưu hóa cao.

Khả năng tương thích ngược: Một số thay đổi lớn trong các phiên bản mới của .NET có thể không tương thích hoàn toàn với mã cũ, đòi hỏi lập trình viên phải cập nhật mã nguồn và tái biên dịch.

2.1.2. Giới thiệu về DART

- Giới thiệu chung

Dart là một ngôn ngữ lập trình mã nguồn mở được Google phát triển, được biết đến rộng rãi nhờ vai trò cốt lõi trong việc xây dựng các ứng dụng đa nền tảng với Flutter. Với cú pháp rõ ràng, dễ học và hiệu năng cao, Dart đã nhanh chóng trở thành lựa chọn hàng đầu cho nhiều nhà phát triển trên toàn thế giới.

Các tính năng chính của Dart bao gồm:

- **Hướng đối tượng thuần túy:** Mọi thứ trong Dart đều là đối tượng, giúp bạn tổ chức mã một cách hiệu quả.
- **Kiểu dữ liệu mạnh:** Giúp giảm thiểu lỗi và tăng tính ổn định của ứng dụng.
- **Không đồng bộ:** Xử lý các tác vụ một cách hiệu quả mà không làm chậm ứng dụng.
- **Biên dịch AOT và JIT:** Đảm bảo hiệu năng cao và khả năng phát triển nhanh.
- **Cú pháp rõ ràng, dễ học:** Giúp bạn nhanh chóng làm quen với ngôn ngữ.
- **Thư viện phong phú:** Cung cấp nhiều công cụ để phát triển ứng dụng.
- **Phát triển đa nền tảng:** Với Flutter, bạn có thể xây dựng ứng dụng cho nhiều nền tảng khác nhau.

2.2. GIỚI THIỆU VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU SQL SERVER

- Giới thiệu chung

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở (open-source) phổ biến, được sử dụng rộng rãi để lưu trữ và quản lý dữ liệu. Dưới đây là một số điểm nổi bật về MySQL:

- Lịch sử và Phát triển

MySQL được phát triển lần đầu vào năm 1995 bởi công ty MySQL AB, sau này được Oracle Corporation mua lại vào năm 2010.

MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, đặc biệt trong các ứng dụng web.

- Tính Năng Chính

Mã nguồn mở: MySQL là phần mềm mã nguồn mở, cho phép người dùng tự do sử dụng, sửa đổi và phân phối.

Hiệu suất cao: MySQL được tối ưu hóa cho hiệu suất, cho phép xử lý các truy vấn nhanh chóng và hiệu quả.

Hỗ trợ SQL: MySQL sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để thực hiện các thao tác trên cơ sở dữ liệu.

Khả năng mở rộng: MySQL có thể xử lý từ các ứng dụng nhỏ đến các hệ thống lớn với hàng triệu bản ghi.

Bảo mật: MySQL cung cấp nhiều tính năng bảo mật, bao gồm xác thực người dùng và mã hóa dữ liệu.

- **Kiến Trúc**

MySQL sử dụng kiến trúc client-server, trong đó máy chủ MySQL quản lý cơ sở dữ liệu và các ứng dụng (client) gửi yêu cầu đến máy chủ để thực hiện các thao tác.

- **Sử Dụng**

MySQL thường được sử dụng trong các ứng dụng web, hệ thống quản lý nội dung (CMS), và các ứng dụng doanh nghiệp.

Nó là một phần quan trọng trong bộ công cụ LAMP (Linux, Apache, MySQL, PHP/Python/Perl), một trong những nền tảng phát triển web phổ biến.

- **Cộng Đồng và Tài Nguyên**

MySQL có một cộng đồng lớn và nhiều tài liệu hướng dẫn, diễn đàn hỗ trợ, và các khóa học trực tuyến giúp người dùng học hỏi và giải quyết vấn đề.

- **Phiên Bản**

MySQL có nhiều phiên bản khác nhau, bao gồm MySQL Community Edition (miễn phí) và MySQL Enterprise Edition (có phí, với các tính năng bổ sung và hỗ trợ kỹ thuật).

2.3. CÔNG CỤ SỬ DỤNG

2.3.1. Giới thiệu về IDE – Visual Studio 2022

- **Giới thiệu tổng quan:**

Visual Studio 2022 là môi trường phát triển tích hợp (IDE) tiên tiến và mạnh mẽ do Microsoft phát triển, nhằm hỗ trợ lập trình viên trong việc viết mã, kiểm thử và gỡ lỗi ứng dụng trên nhiều nền tảng khác nhau. Phiên bản mới nhất này của Visual Studio mang đến nhiều cải tiến về hiệu suất và tính năng, giúp tối ưu hóa quá trình phát triển phần mềm và đáp ứng các nhu cầu ngày càng cao của lập trình viên hiện đại.

- Lịch sử phát triển:

Visual Studio đã trải qua một chặng đường phát triển dài từ khi phiên bản đầu tiên được ra mắt vào năm 1997. Qua từng phiên bản, Visual Studio không ngừng được cải tiến và bổ sung nhiều tính năng mới, trở thành một trong những công cụ phát triển phần mềm được ưa chuộng nhất. Visual Studio 2022 tiếp tục kế thừa và phát triển những thành tựu này, mang đến một công cụ phát triển mạnh mẽ và toàn diện hơn cho các nhà lập trình.

- Đặc trưng cơ bản của Visual Studio 2022:

Hiệu suất vượt trội: Visual Studio 2022 được thiết kế để chạy mượt mà hơn trên các hệ thống cấu hình cao, với khả năng xử lý nhanh hơn và quản lý tài nguyên hiệu quả.

Hỗ trợ đa nền tảng: IDE này hỗ trợ phát triển các ứng dụng trên nhiều hệ điều hành và nền tảng khác nhau như Windows, macOS, Linux, Android, iOS và web.

Giao diện người dùng thân thiện: Giao diện của Visual Studio 2022 được cải tiến hiện đại, trực quan, giúp lập trình viên dễ dàng sử dụng và truy cập các công cụ cần thiết.

Đa dạng ngôn ngữ lập trình: Visual Studio 2022 hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C#, C++, JavaScript, Python, TypeScript, HTML, CSS và nhiều ngôn ngữ khác, giúp lập trình viên linh hoạt trong việc phát triển ứng dụng.

Công cụ phát triển tích hợp: Cung cấp các công cụ tích hợp để viết mã, kiểm thử, gỡ lỗi, quản lý phiên bản và triển khai ứng dụng, tăng cường năng suất và hiệu quả làm việc của lập trình viên.

AI IntelliCode: Tính năng IntelliCode sử dụng trí tuệ nhân tạo để gợi ý và tự động hoàn thành mã, dựa trên các mẫu mã phổ biến và phong cách mã của dự án.

Tích hợp Git và GitHub: Visual Studio 2022 tích hợp sẵn Git và GitHub, giúp quản lý mã nguồn, theo dõi thay đổi và cộng tác với các thành viên trong nhóm dễ dàng hơn.

- Các tính năng nổi bật của Visual Studio 2022:

Live Share: Cho phép các lập trình viên cộng tác trực tiếp trên cùng một mã nguồn trong thời gian thực, tăng cường khả năng làm việc nhóm và giải quyết vấn đề nhanh chóng.

Debugger mạnh mẽ: Cung cấp các công cụ gỡ lỗi mạnh mẽ, dễ sử dụng, giúp kiểm tra và sửa lỗi mã nguồn hiệu quả.

CodeLens: Hiển thị thông tin liên quan đến mã nguồn ngay trong trình soạn thảo, bao gồm thông tin về tác giả, thay đổi gần đây, các bài kiểm thử và nhiều thông tin khác.

Công cụ kiểm thử tích hợp: Hỗ trợ nhiều công cụ kiểm thử tự động và thủ công, giúp đảm bảo chất lượng mã nguồn và phát hiện lỗi sớm trong quá trình phát triển.

Tích hợp Azure: Visual Studio 2022 tích hợp sâu với các dịch vụ đám mây Azure, giúp triển khai và quản lý các ứng dụng đám mây dễ dàng hơn.

- Ứng dụng của Visual Studio 2022:

Phát triển ứng dụng web: Sử dụng ASP.NET, HTML, CSS và JavaScript để phát triển các ứng dụng web động và tương tác.

Phát triển ứng dụng di động: Sử dụng Xamarin hoặc các công cụ khác để phát triển ứng dụng di động cho Android và iOS.

Phát triển ứng dụng máy tính để bàn: Sử dụng C#, C++ hoặc các ngôn ngữ khác để phát triển các ứng dụng máy tính để bàn cho Windows và macOS.

Phát triển game: Sử dụng Unity hoặc Unreal Engine cùng với Visual Studio để phát triển các trò chơi điện tử.

2.3.2. *Giới thiệu về GitHub*

- Giới thiệu tổng quan:

GitHub là một nền tảng lưu trữ mã nguồn và quản lý dự án trực tuyến sử dụng hệ thống kiểm soát phiên bản Git. Được phát triển bởi Tom Preston-Werner, Chris Wanstrath, PJ Hyett và Scott Chacon, GitHub ra mắt lần đầu vào năm 2008. Đây là nơi các lập trình viên và nhóm phát triển có thể lưu trữ mã nguồn, theo dõi các thay đổi, quản lý dự án và cộng tác với nhau một cách hiệu quả. GitHub hiện nay đã trở thành một công cụ quan trọng và phổ biến trong cộng đồng phát triển phần mềm.

- Lịch sử phát triển:

GitHub được ra mắt vào tháng 4 năm 2008 và nhanh chóng trở thành một trong những nền tảng lưu trữ mã nguồn lớn nhất thế giới. Năm 2018, Microsoft đã mua lại GitHub với giá 7,5 tỷ USD, đánh dấu một bước tiến lớn trong việc mở rộng và phát triển các công cụ hỗ trợ lập trình. Từ đó, GitHub đã tiếp tục phát triển và giới thiệu nhiều tính năng mới, giúp cải thiện khả năng cộng tác và quản lý dự án của các lập trình viên.

- Đặc trưng cơ bản của GitHub:

Quản lý mã nguồn: GitHub cho phép lưu trữ, quản lý và theo dõi các thay đổi trong mã nguồn một cách hiệu quả thông qua hệ thống kiểm soát phiên bản Git.

Hỗ trợ cộng tác: GitHub cung cấp các công cụ và tính năng hỗ trợ làm việc nhóm, như pull requests, code review, và issues, giúp các lập trình viên dễ dàng hợp tác và quản lý công việc.

Hosting repository: Cho phép người dùng tạo và lưu trữ các repository công khai hoặc riêng tư, hỗ trợ các dự án từ nhỏ đến lớn.

Tích hợp với nhiều công cụ khác: GitHub tích hợp tốt với nhiều công cụ và dịch vụ khác như CI/CD, DevOps, và các IDE phổ biến, giúp tăng cường hiệu quả làm việc.

GitHub Pages: Cho phép người dùng tạo các trang web tĩnh miễn phí để giới thiệu dự án hoặc tài liệu hướng dẫn, trực tiếp từ repository của họ.

Bảo mật và quyền truy cập: Cung cấp các công cụ quản lý quyền truy cập, giúp kiểm soát ai có thể xem và thay đổi mã nguồn của dự án.

- Các tính năng nổi bật của GitHub:

Pull Requests: Tính năng này cho phép người dùng đề xuất các thay đổi trong mã nguồn và tiến hành review trước khi tích hợp vào nhánh chính.

Issues: Giúp theo dõi các lỗi, yêu cầu tính năng và các nhiệm vụ khác liên quan đến dự án, giúp quản lý công việc hiệu quả hơn.

Actions: GitHub Actions cho phép tự động hóa quy trình làm việc bằng cách tạo ra các workflow để build, test và triển khai mã nguồn.

Code Review: Công cụ review mã nguồn mạnh mẽ, giúp các lập trình viên nhận phản hồi, thảo luận và cải thiện chất lượng mã nguồn.

GitHub Marketplace: Nơi cung cấp các công cụ và ứng dụng bổ trợ cho việc phát triển và quản lý dự án, giúp mở rộng chức năng của GitHub.

- Ứng dụng của GitHub:

Lưu trữ và quản lý mã nguồn: GitHub là nơi lý tưởng để lưu trữ và quản lý mã nguồn của các dự án phần mềm, từ các dự án cá nhân đến các dự án lớn với sự tham gia của nhiều người.

Cộng tác và làm việc nhóm: GitHub hỗ trợ cộng tác hiệu quả giữa các lập trình viên, giúp theo dõi tiến độ công việc, quản lý các yêu cầu và phản hồi, và giữ cho dự án luôn được cập nhật.

Phát triển mã nguồn mở: GitHub là nền tảng phổ biến cho các dự án mã nguồn mở, nơi các nhà phát triển có thể chia sẻ mã nguồn, đóng góp và hợp tác với cộng đồng.

Tài liệu và hướng dẫn: GitHub Pages và README files cho phép các nhà phát triển tạo tài liệu và hướng dẫn chi tiết cho dự án của mình.

Triển khai và tích hợp liên tục: Với GitHub Actions và tích hợp với các công cụ CI/CD khác, GitHub hỗ trợ triển khai và kiểm thử tự động, giúp đảm bảo chất lượng và ổn định của mã nguồn.

2.3.3. *Giới thiệu về Draw.io*

- Giới thiệu tổng quan:

Draw.io, hiện được biết đến với tên gọi diagrams.net, là một công cụ vẽ sơ đồ trực tuyến miễn phí. Nó được thiết kế để giúp người dùng tạo ra các loại sơ đồ đa dạng, từ sơ đồ tổ chức, sơ đồ quy trình, sơ đồ mạng, đến biểu đồ UML và các bản thiết kế kỹ thuật. Với giao diện thân thiện và nhiều tính năng phong phú, Draw.io là lựa chọn ưa thích của cá nhân và doanh nghiệp khi cần tạo sơ đồ và biểu đồ trực quan.

- Lịch sử phát triển:

Được phát triển bởi công ty JGraph Ltd., Draw.io ra mắt lần đầu vào năm 2011. Sau đó, công cụ này được đổi tên thành diagrams.net để phản ánh chính xác hơn về chức năng và mục tiêu của nó. Dù có sự thay đổi tên gọi, Draw.io vẫn duy trì được sự tin cậy và ưa chuộng của người dùng toàn cầu nhờ tính linh hoạt và hiệu quả cao.

- Đặc trưng cơ bản của Draw.io:

Dễ sử dụng: Giao diện trực quan của Draw.io cho phép người dùng dễ dàng tạo sơ đồ bằng các công cụ kéo thả, không cần kỹ năng thiết kế chuyên nghiệp.

Tích hợp đám mây: Hỗ trợ lưu trữ và truy cập trực tiếp từ các dịch vụ đám mây như Google Drive, OneDrive, Dropbox, GitHub, và GitLab, giúp dễ dàng truy cập và chia sẻ sơ đồ ở bất kỳ đâu.

Đa dạng mẫu sơ đồ: Cung cấp hàng ngàn mẫu sơ đồ và biểu đồ có sẵn, giúp người dùng nhanh chóng tạo và tùy chỉnh sơ đồ theo nhu cầu.

Hỗ trợ nhiều định dạng: Cho phép xuất sơ đồ ra nhiều định dạng khác nhau như PNG, JPEG, SVG, PDF, và XML, phù hợp cho trình chiếu, in ấn và tích hợp vào các tài liệu và báo cáo.

Khả năng cộng tác: Hỗ trợ làm việc nhóm và cộng tác trực tuyến, cho phép nhiều người cùng chỉnh sửa một sơ đồ đồng thời, nâng cao hiệu quả công việc nhóm.

Bảo mật và quyền riêng tư: Cam kết bảo mật dữ liệu người dùng, cho phép tùy chỉnh quyền truy cập và bảo mật cho các sơ đồ được lưu trữ trực tuyến.

- Các tính năng nổi bật của Draw.io:

Công cụ vẽ đa dạng: Cung cấp nhiều công cụ vẽ như hình học cơ bản, mũi tên, đường nối, biểu đồ, và các hình ảnh minh họa khác để tạo sơ đồ phức tạp dễ dàng.

Thư viện biểu tượng phong phú: Hàng ngàn biểu tượng có sẵn trong thư viện giúp người dùng thêm vào sơ đồ mà không cần tìm kiếm từ nguồn bên ngoài.

Khả năng tùy biến cao: Cho phép tùy chỉnh các yếu tố của sơ đồ từ màu sắc, kiểu dáng đến bố cục, giúp tạo ra những sơ đồ phù hợp với yêu cầu và phong cách riêng.

Tính năng undo/redo không giới hạn: Người dùng có thể hoàn tác hoặc làm lại các thay đổi mà không bị giới hạn, giúp việc chỉnh sửa sơ đồ linh hoạt và thuận tiện hơn.

Tích hợp với các nền tảng khác: Tích hợp tốt với các nền tảng như Confluence, Jira, và các công cụ quản lý dự án, giúp người dùng dễ dàng nhúng sơ đồ vào tài liệu và quy trình công việc.

- Ứng dụng của Draw.io:

Giáo dục: Giáo viên và sinh viên sử dụng Draw.io để tạo sơ đồ tư duy, sơ đồ quy trình học tập và tài liệu giảng dạy trực quan.

Kinh doanh: Các doanh nghiệp sử dụng Draw.io để lập sơ đồ tổ chức, quy trình làm việc và các bản thiết kế kinh doanh, giúp cải thiện hiệu quả và quản lý công việc.

Kỹ thuật và CNTT: Hỗ trợ kỹ sư và chuyên gia CNTT tạo các sơ đồ mạng, sơ đồ hệ thống và các bản thiết kế kỹ thuật chi tiết.

Quản lý dự án: Quản lý dự án sử dụng Draw.io để lập kế hoạch, theo dõi tiến độ và quản lý các nhiệm vụ trong dự án thông qua các sơ đồ Gantt, Kanban và các biểu đồ khác.

2.4. MÔ HÌNH VÀ KỸ THUẬT

2.4.1. Giới thiệu về EntityFramework

- Giới thiệu tổng quan:

Entity Framework (EF) là một Object-Relational Mapper (ORM) mã nguồn mở cho .NET, do Microsoft phát triển. EF giúp lập trình viên làm việc với cơ sở dữ liệu dễ dàng hơn bằng cách sử dụng các đối tượng .NET thay vì các câu lệnh SQL thuần túy. Entity Framework tự động ánh xạ các đối tượng trong code với các bảng trong cơ sở dữ liệu, giúp giảm thiểu mã lệnh phải viết và tăng hiệu quả phát triển phần mềm.

- Lịch sử phát triển:

Entity Framework được giới thiệu lần đầu vào năm 2008 như một phần của .NET Framework 3.5 SP1 và Visual Studio 2008 SP1. Từ đó, EF đã trải qua nhiều phiên bản và cải tiến, bao gồm việc chuyển sang mã nguồn mở và hỗ trợ cho .NET Core. Entity Framework Core (EF Core) là phiên bản hiện đại và nhẹ hơn của EF, được phát triển để tương thích với .NET Core và các nền tảng đa hệ điều hành.

- Đặc trưng cơ bản của Entity Framework:

Model-Based Development: Entity Framework hỗ trợ ba phương pháp phát triển mô hình: Database-First, Model-First và Code-First, cho phép lập trình viên lựa chọn phương pháp phù hợp nhất với nhu cầu của dự án.

Làm việc với dữ liệu như các đối tượng: EF cho phép lập trình viên làm việc với dữ liệu dưới dạng các đối tượng .NET, giúp tăng tính trực quan và dễ dàng quản lý.

Lazy Loading và Eager Loading: Entity Framework hỗ trợ cả hai kỹ thuật tải dữ liệu Lazy Loading và Eager Loading, giúp tối ưu hóa hiệu suất truy vấn cơ sở dữ liệu.

Migration: EF cung cấp các công cụ để quản lý và áp dụng các thay đổi trong cấu trúc cơ sở dữ liệu một cách an toàn và dễ dàng thông qua tính năng Migrations.

Query Translation: EF tự động chuyển đổi các truy vấn LINQ thành các câu lệnh SQL tối ưu, giúp tiết kiệm thời gian và công sức khi viết các truy vấn phức tạp.

- Các tính năng nổi bật của Entity Framework:

Khả năng mở rộng và tùy chỉnh: EF cho phép lập trình viên tùy chỉnh ánh xạ giữa các đối tượng và cơ sở dữ liệu thông qua các Data Annotations hoặc Fluent API.

Hỗ trợ đa nền tảng: Entity Framework Core hỗ trợ nhiều hệ điều hành và cơ sở dữ liệu khác nhau như SQL Server, MySQL, PostgreSQL, SQLite, và Oracle, giúp mở rộng phạm vi sử dụng.

Đơn giản hóa việc xử lý dữ liệu phức tạp: EF giúp đơn giản hóa việc xử lý các mối quan hệ phức tạp giữa các bảng trong cơ sở dữ liệu thông qua các khái niệm như Foreign Keys, Navigational Properties và các loại quan hệ (một-nhiều, nhiều-nhiều).

Tích hợp với ASP.NET: EF tích hợp chặt chẽ với ASP.NET, giúp phát triển các ứng dụng web và dịch vụ RESTful dễ dàng hơn.

Tự động theo dõi thay đổi: EF tự động theo dõi các thay đổi trong các đối tượng và áp dụng các thay đổi đó vào cơ sở dữ liệu khi lưu, giúp đảm bảo tính nhất quán và toàn vẹn dữ liệu.

- Ứng dụng của Entity Framework:

Phát triển ứng dụng doanh nghiệp: EF được sử dụng rộng rãi trong các ứng dụng doanh nghiệp để quản lý dữ liệu phức tạp và các nghiệp vụ quan trọng.

Ứng dụng web: Với sự tích hợp tốt với ASP.NET và khả năng quản lý dữ liệu hiệu quả, EF là công cụ lý tưởng cho phát triển các ứng dụng web động và dịch vụ RESTful.

Ứng dụng di động: EF Core hỗ trợ phát triển ứng dụng di động với Xamarin và các nền tảng khác, giúp đồng bộ hóa và quản lý dữ liệu trên các thiết bị di động.

Ứng dụng phân tích và báo cáo: EF giúp xử lý và truy vấn các tập dữ liệu lớn, phù hợp cho các ứng dụng phân tích dữ liệu và tạo báo cáo.

2.4.2. Giới thiệu về ASP.NET Identity

- Giới thiệu tổng quan:

ASP.NET Identity là một hệ thống quản lý danh tính linh hoạt và mở rộng được phát triển bởi Microsoft, nhằm cung cấp các giải pháp quản lý người dùng và xác thực cho các ứng dụng web ASP.NET. ASP.NET Identity cho phép lập trình viên dễ dàng quản lý thông tin người dùng, bao gồm đăng ký, đăng nhập, phân quyền và quản lý vai trò, đồng thời hỗ trợ nhiều phương thức xác thực khác nhau như mật khẩu, xác thực hai yếu tố, và xác thực bằng mạng xã hội.

- Lịch sử phát triển:

ASP.NET Identity ra mắt lần đầu vào năm 2013 như một phần của ASP.NET CORE. Trước đó, các ứng dụng ASP.NET thường sử dụng Membership Provider hoặc Simple Membership để quản lý người dùng. ASP.NET Identity được phát triển để khắc phục những hạn chế của các hệ thống trước đây, cung cấp một hệ thống quản lý danh tính linh hoạt, dễ mở rộng và tích hợp tốt với các công nghệ hiện đại.

- Đặc trưng cơ bản của ASP.NET Identity:

Quản lý người dùng và vai trò: ASP.NET Identity cho phép quản lý thông tin người dùng, bao gồm đăng ký, đăng nhập, và chỉnh sửa thông tin. Hệ thống cũng hỗ trợ quản lý vai trò và phân quyền, giúp kiểm soát truy cập dễ dàng.

Xác thực đa dạng: Hỗ trợ nhiều phương thức xác thực, bao gồm mật khẩu, xác thực hai yếu tố (2FA), xác thực bằng mạng xã hội (Google, Facebook, Twitter), và xác thực bằng các dịch vụ bên ngoài như Microsoft Account.

Tích hợp Entity Framework: ASP.NET Identity tích hợp chặt chẽ với Entity Framework, cho phép lưu trữ và quản lý dữ liệu người dùng trong cơ sở dữ liệu một cách dễ dàng và linh hoạt.

Hỗ trợ xác thực token: ASP.NET Identity hỗ trợ xác thực dựa trên token, cho phép xây dựng các API RESTful và các dịch vụ xác thực không trạng thái (stateless authentication).

- Các tính năng nổi bật của ASP.NET Identity:

Xác thực hai yếu tố (2FA): Hỗ trợ xác thực hai yếu tố để tăng cường bảo mật cho người dùng, bao gồm xác thực qua SMS, email, hoặc ứng dụng xác thực.

Khả năng lưu trữ linh hoạt: ASP.NET Identity cho phép lưu trữ dữ liệu người dùng trong nhiều loại cơ sở dữ liệu khác nhau, không chỉ giới hạn ở SQL Server mà còn bao gồm MySQL, PostgreSQL, và MongoDB.

Reset và khôi phục mật khẩu: Cung cấp các tính năng để người dùng có thể đặt lại mật khẩu khi quên hoặc khôi phục tài khoản một cách dễ dàng.

Quản lý khóa bảo mật: Hỗ trợ quản lý và xoay vòng các khóa bảo mật để bảo vệ các thông tin nhạy cảm của người dùng.

Tích hợp với các hệ thống xác thực bên ngoài: Dễ dàng tích hợp với các hệ thống xác thực bên ngoài và các nhà cung cấp OpenID Connect/OAuth 2.0.

Ứng dụng của ASP.NET Identity:

Quản lý người dùng trong ứng dụng web: ASP.NET Identity là giải pháp hoàn hảo cho việc quản lý người dùng trong các ứng dụng web ASP.NET, từ các trang web thương mại điện tử đến các hệ thống quản lý nội dung.

API RESTful và dịch vụ web: Hỗ trợ xây dựng các API RESTful bảo mật và các dịch vụ web với xác thực dựa trên token.

Ứng dụng di động: ASP.NET Identity có thể được sử dụng trong các backend của ứng dụng di động để quản lý người dùng và xác thực.

Các hệ thống doanh nghiệp: Được sử dụng rộng rãi trong các hệ thống quản lý doanh nghiệp, nơi yêu cầu quản lý người dùng và phân quyền phức tạp.

2.4.3. Giới thiệu về API

*Khái niệm: API (Application Programming Interface) là một giao diện lập trình ứng dụng cho phép các ứng dụng khác nhau tương tác với nhau. Trong Flutter, API đóng vai trò vô cùng quan trọng, cho phép ứng dụng của bạn truy xuất dữ liệu từ các nguồn bên ngoài như cơ sở dữ liệu, dịch vụ web, hoặc các thiết bị khác.

*Đặc điểm nổi bật

-Tính trừu tượng (Abstraction): API cung cấp một lớp trừu tượng, che giấu các chi tiết thực hiện bên trong của một hệ thống.

- Tính mô đun (Modularity): API chia nhỏ các chức năng của một hệ thống thành các đơn vị nhỏ hơn, gọi là các endpoint. Mỗi endpoint thực hiện một nhiệm vụ cụ thể, giúp cho việc phát triển, bảo trì và mở rộng hệ thống trở nên dễ dàng hơn.

-Tính độc lập (Independence): API cho phép các ứng dụng khác nhau tương tác với nhau một cách độc lập, không phụ thuộc vào ngôn ngữ lập trình hay nền tảng phần cứng. Điều này tạo điều kiện cho việc tích hợp các hệ thống khác nhau.

-Tính mở rộng (Extensibility): API có thể được mở rộng để đáp ứng các yêu cầu mới và thay đổi của hệ thống. Các nhà phát triển có thể thêm các endpoint mới hoặc sửa đổi các endpoint hiện có để cung cấp các chức năng bổ sung.

-Tính tiêu chuẩn hóa (Standardization): API thường tuân theo các tiêu chuẩn nhất định, giúp đảm bảo tính tương thích giữa các hệ thống khác nhau. Các tiêu chuẩn phổ biến bao gồm REST (Representational State Transfer), SOAP (Simple Object Access Protocol), GraphQL.

-Tính bảo mật (Security): API thường được bảo vệ bằng các cơ chế xác thực và ủy quyền để đảm bảo rằng chỉ những người dùng được phép mới có thể truy cập và sử dụng dữ liệu.

Các loại API phổ biến

- REST API: Là loại API phổ biến nhất, sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) để thực hiện các thao tác trên tài nguyên.
- GraphQL API: Một loại API mới hơn, cho phép khách hàng yêu cầu chính xác dữ liệu mà họ cần, giúp giảm thiểu lượng dữ liệu truyền đi.
- SOAP API: Một giao thức truyền thông dựa trên XML, thường được sử dụng trong các hệ thống doanh nghiệp.

Ứng dụng của API trong đời sống

- Ứng dụng di động: Hầu hết các ứng dụng di động đều sử dụng API để truy cập dữ liệu từ các nguồn bên ngoài như mạng xã hội, dịch vụ bản đồ, dịch vụ thanh toán,...
- Website: Các website sử dụng API để tích hợp các tính năng như đăng nhập bằng tài khoản mạng xã hội, chia sẻ nội dung lên mạng xã hội, nhận xét sản phẩm,...
- IoT: Các thiết bị IoT sử dụng API để kết nối với nhau và với các hệ thống khác.

*Ưu điểm của API:

- Tăng tốc độ phát triển: API cung cấp các khối xây dựng sẵn, giúp giảm thiểu thời gian và công sức phát triển ứng dụng.
- Nâng cao khả năng mở rộng: API cho phép các ứng dụng dễ dàng tích hợp với các hệ thống khác.
- Cải thiện khả năng bảo trì: API giúp tách biệt các thành phần của hệ thống, giúp dễ dàng bảo trì và nâng cấp.
- Tăng cường tính bảo mật: API cung cấp các cơ chế bảo mật để bảo vệ dữ liệu.
- Tiêu chuẩn hóa: API thường tuân theo các tiêu chuẩn nhất định, giúp đảm bảo tính tương thích giữa các hệ thống khác nhau.
- Tái sử dụng mã: Các nhà phát triển có thể tái sử dụng các API đã được xây dựng sẵn, tiết kiệm thời gian và công sức.

*Nhược điểm của API:

- Độ phức tạp: Việc thiết kế và quản lý API có thể phức tạp, đặc biệt là đối với các hệ thống lớn.
- Phụ thuộc vào bên thứ ba: Khi sử dụng API của bên thứ ba, bạn sẽ phụ thuộc vào sự ổn định và khả năng hỗ trợ của họ.
- Chi phí: Một số API có thể yêu cầu trả phí để sử dụng.
- Rủi ro bảo mật: Nếu API không được bảo mật tốt, có thể dẫn đến rò rỉ dữ liệu.
- Giới hạn sử dụng: Một số API có thể đặt ra các giới hạn về số lượng yêu cầu hoặc lượng dữ liệu truy xuất.

2.5. Giới thiệu về FLUTTER

* Giới thiệu chung:

Flutter là một bộ công cụ SDK (Software Development Kit) mã nguồn mở được Google phát triển, cho phép các nhà phát triển xây dựng các ứng dụng di động, web và desktop từ một cơ sở mã duy nhất. Điều này có nghĩa là bạn chỉ cần viết một lần code và có thể triển khai ứng dụng của mình trên nhiều nền tảng khác nhau như iOS, Android, web và thậm chí cả Windows, macOS và Linux.

*Đặc điểm chính:

- Hiệu năng cao: Flutter sử dụng ngôn ngữ lập trình Dart, được biên dịch thành mã máy gốc (native code) trên từng nền tảng, giúp ứng dụng chạy mượt mà và nhanh chóng.
- Giao diện người dùng đẹp mắt: Flutter cung cấp một bộ widget phong phú, tùy biến cao, giúp bạn tạo ra các ứng dụng với giao diện đẹp mắt và trải nghiệm người dùng tuyệt vời.
- Phát triển nhanh: Hot Reload cho phép bạn thấy ngay các thay đổi trên ứng dụng mà không cần khởi động lại ứng dụng, giúp tăng tốc quá trình phát triển.
- Một codebase, nhiều nền tảng: Viết một lần code và triển khai trên nhiều nền tảng, tiết kiệm thời gian và chi phí phát triển.
- Cộng đồng lớn mạnh: Flutter có một cộng đồng người dùng lớn và đang phát triển nhanh chóng, giúp bạn dễ dàng tìm kiếm tài liệu, thư viện và hỗ trợ.
- Được Google hỗ trợ: Là một sản phẩm của Google, Flutter được hỗ trợ và phát triển liên tục bởi một đội ngũ kỹ sư tài năng.

*Ứng dụng của flutter: Flutter là công cụ tuyệt vời để xây dựng các ứng dụng đa nền tảng, chất lượng cao và tiết kiệm chi phí như :

Ứng dụng thương mại điện tử: Các ứng dụng như Lazada, Alibaba... đã sử dụng Flutter để xây dựng các ứng dụng mua sắm trực tuyến với giao diện đẹp mắt và trải nghiệm người dùng mượt mà.

- Ứng dụng mạng xã hội: Các ứng dụng mạng xã hội như Instagram, Facebook cũng đang dần chuyển sang sử dụng Flutter để xây dựng các tính năng mới.
- Ứng dụng ngân hàng: Nhiều ngân hàng đã sử dụng Flutter để xây dựng các ứng dụng di động với giao diện hiện đại và bảo mật cao.
- Ứng dụng du lịch: Các ứng dụng đặt vé máy bay, khách sạn, du lịch... cũng được xây dựng bằng Flutter để cung cấp trải nghiệm đặt phòng thuận tiện.
- Ứng dụng giáo dục: Các ứng dụng học tập trực tuyến, ứng dụng luyện thi... cũng được xây dựng bằng Flutter để giúp người dùng học tập hiệu quả hơn.

Ứng dụng web

- Web app: Flutter for web cho phép bạn xây dựng các ứng dụng web với hiệu năng cao và giao diện đẹp mắt.

- Progressive Web App (PWA): Bạn có thể xây dựng các PWA để cung cấp trải nghiệm ứng dụng native cho người dùng web.

Ứng dụng desktop

- Ứng dụng desktop: Flutter for desktop cho phép bạn xây dựng các ứng dụng desktop chạy trên Windows, macOS và Linux.

*Ưu điểm:

Flutter chạy nhanh giúp tiết kiệm thời gian, công sức và tiền bạc. Giống như bất kỳ công nghệ đa nền tảng nào khác, Flutter cho phép bạn sử dụng cùng một cơ sở mã để xây dựng các ứng dụng iOS và Android riêng biệt. Điều này sẽ đẩy nhanh toàn bộ quá trình phát triển mà không cần phân tích hai cơ sở mã khác nhau cho cùng một nền tảng.

"Hot reload" (tải nóng/ nhanh) của Flutter giúp bạn thực hiện các thay đổi đối với mã code và xem được kết quả ngay lập tức trong bản xem trước ứng dụng mà không cần đọc lại mã. Bằng cách này, bạn có thể dễ dàng sửa lỗi và thử nghiệm với các phần tử và tính năng UI khác nhau.

Tùy chỉnh toàn bộ & kết xuất nhanh nhờ cấu trúc phân lớp của Flutter. Ứng dụng này cung cấp quyền kiểm soát mọi pixel trên màn hình cũng như không giới hạn người dùng thêm và tạo hoạt ảnh trong thiết kế đồ họa, video, văn bản và điều khiển.

Flutter cũng áp dụng cho web và cung cấp tài liệu thích hợp cho phép bạn kiểm tra cách các điều khiển gốc hoạt động.

Flutter tách UI khỏi các điều khiển gốc giúp loại bỏ lỗi không tương thích (dù ít xảy ra) từ phía nhà sản xuất. UI riêng biệt cũng tự động đem đến một sự đồng nhất trên tất cả các phiên bản hệ thống.

*Nhược điểm:

Flutter vẫn chưa thật hoàn thiện. Vì là một ứng dụng mới nên Flutter vẫn chưa đạt mức hoàn hảo. Thực tế, nhiều tính năng nâng cao của Flutter vẫn chưa được hỗ trợ; nhiều thư viện chưa được thử nghiệm chính thức còn tồn tại hạn chế khi so sánh với các bản sao gốc (như là Google Maps).

Dart khá "non nớt". Về cơ bản Dart khá giống với Swift và Kotlin, nhưng có ít tính năng hơn, hoặc những tính năng hiện có chưa được toàn diện.

Các ứng dụng Flutter khá "nặng". Chúng chiếm nhiều dung lượng và mất nhiều thời gian để tải xuống hoặc cập nhật.

Giao diện không giống 100% so với phiên bản gốc. Về cơ bản, Flutter không tạo ra các thành phần gốc mà sao chép không hoàn toàn các thiết kế Material Design của Android và các thành phần riêng của iOS bằng thư viện Cupertino. Thư viện này sẽ hiển thị, đặc biệt với các phiên bản hệ thống chứa các trường văn bản hoặc các nút - những thành phần biến đổi bên ngoài nhưng không thay đổi bên trong Flutter.

Hướng dẫn phát triển ứng dụng Flutter chưa được đồng nhất, điều này có thể gây nhiều khó khăn khi xây dựng các phần mềm mang tính phức tạp.

Framework thay đổi nhanh chóng gây khó khăn cho việc duy trì mã. Thêm vào đó, Flutter chưa chắc sẽ được ứng dụng trong tương lai khi Google liên tục loại bỏ các dự án của ứng dụng này.

2.6. GIỚI THIỆU CÁC THƯ VIỆN HỖ TRỢ

2.6.1. Dio

Chức năng: Dio là một HTTP client mạnh mẽ cho Dart, được sử dụng rộng rãi trong các ứng dụng Flutter để thực hiện các yêu cầu mạng như GET, POST, PUT, DELETE, v.v. Nó cung cấp một giao diện dễ sử dụng và các tính năng tiên tiến để xử lý các yêu cầu HTTP.

Các tính năng nổi bật:

- **Interceptor:** Cho phép tùy chỉnh các yêu cầu và phản hồi trước khi chúng được gửi hoặc nhận.
- **FormData:** Hỗ trợ gửi dữ liệu dạng form.
- **Cancellation:** Hủy bỏ các yêu cầu đang thực hiện.
- **Retry:** Tự động thực hiện lại các yêu cầu thất bại.
- **Timeout:** Thiết lập thời gian chờ cho các yêu cầu.
- **HTTPS:** Hỗ trợ giao thức HTTPS.

Thuận lợi :

Giàu tính năng : Dio là một HTTP client mạnh mẽ cho Dart. Nó cung cấp nhiều tính năng vượt xa mạng cơ bản.

Interceptor : Bạn có thể sử dụng interceptor cho các tác vụ như ghi nhật ký, lưu vào bộ nhớ đệm và sửa đổi yêu cầu và phản hồi.

Cấu hình toàn cầu : Tùy chỉnh cài đặt toàn cầu cho tất cả các yêu cầu. **FormData** và **Tải xuống tệp :** Hỗ trợ tải lên và tải xuống tệp.

Thời gian chờ : Đặt thời gian chờ cho các yêu cầu.

Nhanh chóng và hiệu quả : Sử dụng các yêu cầu không đồng bộ và song song.

Nhược điểm :

Độ phức tạp : Do có nhiều chức năng nên khả năng gặp phải lỗi cao hơn.

- Gói Dio là một trình khách HTTP mạnh mẽ và giàu tính năng cho Dart và Flutter, cung cấp các khả năng nâng cao như cấu hình toàn cục, bộ chặn, yêu cầu đồng thời và xử lý tệp. Trong khi gói http phù hợp với các dự án nhỏ hơn do tính đơn giản và nhẹ của nó, Dio lại vượt trội trong các ứng dụng lớn hơn, phức tạp hơn đòi hỏi các tính năng mạng mở rộng và hỗ trợ dài hạn.

2.6.2. GET

GET là một giải pháp quản lý trạng thái và điều hướng cực kỳ mạnh mẽ và linh hoạt trong Flutter. Nó được thiết kế để giúp các nhà phát triển xây dựng các ứng dụng Flutter một cách nhanh chóng, hiệu quả và dễ bảo trì hơn.

GET được thiết kế với khả năng mở rộng cao, cho phép bạn tùy chỉnh và mở rộng thư viện để phù hợp với các yêu cầu cụ thể của dự án. Bạn có thể tạo các lớp controller tùy chỉnh, các middleware riêng biệt, và thậm chí là thay đổi cách thức hoạt động của các tính năng cốt lõi. Điều này giúp GET trở thành một công cụ linh hoạt và phù hợp với nhiều loại dự án khác nhau.

Cộng đồng người dùng GET rất lớn mạnh và sôi động, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức. Có rất nhiều tài liệu, hướng dẫn, và ví dụ mã nguồn được chia sẻ trên các diễn đàn, blog và kho lưu trữ mã nguồn mở. Điều này giúp bạn dễ dàng tìm kiếm thông tin và giải quyết các vấn đề khi gặp phải.

So với các thư viện quản lý trạng thái khác như Provider, Riverpod hay BLoC, GET nổi bật với sự đơn giản và dễ sử dụng. Tuy nhiên, các thư viện khác cũng có những ưu điểm riêng của mình, phù hợp với những yêu cầu và phong cách làm việc khác nhau. Việc lựa chọn thư viện nào phụ thuộc vào quy mô dự án, kinh nghiệm của đội ngũ phát triển và các yêu cầu cụ thể của ứng dụng.

GET đang không ngừng phát triển và cải tiến, với sự đóng góp của cộng đồng và đội ngũ phát triển. Trong tương lai, chúng ta có thể mong đợi sẽ thấy thêm nhiều tính năng mới và cải tiến hiệu năng của GET. Điều này hứa hẹn sẽ giúp

GET trở thành một trong những thư viện quản lý trạng thái phổ biến nhất trong cộng đồng Flutter.

Các khái niệm cơ bản trong GET

- **GetxController:** Là một lớp cơ bản để quản lý trạng thái.
- **GetBuilder:** Widget để xây dựng UI dựa trên trạng thái của GetxController.
- **Get.to:** Hàm để điều hướng đến một màn hình khác.
- **Get.put:** Hàm để tiêm phụ thuộc.
- **Obx:** Widget để lắng nghe sự thay đổi của các biến reactive.

GET là một công cụ vô cùng hữu ích cho các nhà phát triển Flutter. Nó giúp bạn viết mã sạch hơn, dễ bảo trì hơn và tăng tốc độ phát triển ứng dụng. Nếu bạn đang tìm kiếm một giải pháp quản lý trạng thái và điều hướng hiệu quả cho ứng dụng Flutter của mình, GET là một lựa chọn tuyệt vời.

2.6.3. CachedNetworkImage

CachedNetworkImage là một widget được thiết kế để tối ưu hóa việc tải và hiển thị hình ảnh từ internet. Nó có khả năng:

- **Tự động cache hình ảnh:** Khi một hình ảnh được tải lần đầu, nó sẽ được lưu vào bộ nhớ cache của thiết bị. Những lần tải tiếp theo, widget sẽ lấy hình ảnh từ bộ nhớ cache thay vì tải lại từ mạng, giúp tiết kiệm băng thông và tăng tốc độ hiển thị.
- **Xử lý lỗi tải hình ảnh:** Nếu quá trình tải hình ảnh gặp lỗi, widget có thể hiển thị một placeholder hoặc một thông báo lỗi tùy chỉnh.
- **Cung cấp các tùy chọn tùy biến:** Bạn có thể tùy chỉnh cách thức hiển thị hình ảnh, ví dụ như điều chỉnh kích thước, độ bo tròn, hiệu ứng mờ, ...
- **Tương thích với nhiều loại định dạng hình ảnh:** Hỗ trợ các định dạng hình ảnh phổ biến như JPEG, PNG, GIF, ...
- **Ngoài các tính năng cơ bản như cache hình ảnh, xử lý lỗi và tùy biến hiển thị, CachedNetworkImage còn cung cấp nhiều tính năng nâng cao khác. Bạn có thể tùy chỉnh thời gian sống của hình ảnh trong bộ nhớ cache, kích thước bộ nhớ cache tối đa, hoặc thậm chí là tự định nghĩa các chiến lược cache tùy theo nhu cầu của ứng dụng. Ngoài ra, widget này còn hỗ trợ tải**

hình ảnh từ nhiều nguồn khác nhau như asset, file hệ thống, hay thậm chí là từ một hàm cung cấp dữ liệu.

- So với việc tự mình triển khai cơ chế tải và cache hình ảnh, việc sử dụng `CachedNetworkImage` giúp bạn tiết kiệm thời gian và công sức. Thư viện này đã được tối ưu hóa và kiểm thử kỹ lưỡng, đảm bảo hiệu suất và độ ổn định cao. Ngoài ra, so với các thư viện tương tự khác, `CachedNetworkImage` thường được đánh giá cao về tính dễ sử dụng và độ linh hoạt.

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. MÔ TẢ NGHIỆP VỤ

3.1.1. *Mô tả chung*

App bán cá cảnh được thiết kế nhằm cung cấp một nền tảng tiện lợi và hiệu quả cho việc quản lý và mua bán cá cảnh trực tuyến. Hệ thống bao gồm hai loại người dùng chính: người dùng thông thường và quản trị viên.

3.1.2. *Quản lý tài khoản người dùng*

Người dùng đã đăng nhập có thể cập nhật thông tin cá nhân của mình bao gồm tên, email và mật khẩu. Quản trị viên có quyền quản lý tất cả tài khoản người dùng, bao gồm xem danh sách người dùng, chỉnh sửa thông tin và xóa tài khoản nếu cần thiết. Đối với các tài khoản nhân viên, quản trị viên có thể tạo tài khoản và gán quyền hạn phù hợp.

3.1.3. *Quản lý sản phẩm*

Quản trị viên có thể quản lý các danh mục sản phẩm và các sản phẩm cá cảnh thông qua giao diện quản trị. Quản trị viên có thể tạo mới, chỉnh sửa và xóa danh mục sản phẩm. Khi tạo danh mục, quản trị viên cần nhập các thông tin như tên danh mục và mô tả chi tiết. Tương tự, quản trị viên có thể tạo mới, chỉnh sửa và xóa sản phẩm cá cảnh. Thông tin sản phẩm bao gồm tên sản phẩm, giá cả, loại sản phẩm, hình ảnh và tình trạng sẵn sàng.

3.1.4. *Mua hàng*

Người dùng có thể duyệt qua danh sách các sản phẩm cá cảnh có sẵn và xem chi tiết từng sản phẩm bao gồm mô tả, hình ảnh, giá cả và khuyến mãi đi kèm. Khi tìm được sản phẩm phù hợp, người dùng có thể đặt hàng bằng cách chọn số lượng sản phẩm, sau đó điền thông tin liên hệ và thanh toán. Hệ thống sẽ kiểm tra tính sẵn sàng của sản phẩm và xác nhận đơn hàng cho người dùng.

3.1.5. *Quản lý đơn hàng*

Quản trị viên có thể xem và quản lý tất cả các đơn hàng thông qua giao diện quản trị. Điều này bao gồm việc xem danh sách đơn hàng, chi tiết từng đơn hàng, cập nhật trạng thái đơn hàng (đã xác nhận, đã hủy, đang chờ) và quản lý thông tin thanh toán. Người dùng có thể xem lịch sử mua hàng của mình và trạng thái hiện tại của từng đơn hàng.

3.2. CÁC LOẠI THỰC THỂ

3.2.1. *Cơ bản*

- Danh mục sản phẩm (Category):
 - Thuộc tính: MaDanhMuc, TenDanhMuc, MoTa.
 - Mô tả: Đại diện cho các danh mục sản phẩm như cá cảnh nước ngọt, cá cảnh nước mặn, phụ kiện hồ cá, thức ăn cho cá.
- Sản phẩm (Product):
 - Thuộc tính: MaSanPham, TenSanPham, Gia, HinhAnh, TrangThai.
 - Mô tả: Đại diện cho các sản phẩm cụ thể với thông tin về tên, giá cả, hình ảnh, tình trạng.
- Hình ảnh sản phẩm (ProductImage):
 - Thuộc tính: MaHinhAnh, DuongDan, MaSanPham.
 - Mô tả: Đại diện cho các hình ảnh liên quan đến sản phẩm.
- Bình luận(Comment):MaBinhLuan, NoiDung, NgayTao, MaSanPham, MaNguoiDung
 - Mô tả : Đại diện cho các bình luận liên quan đến sản phẩm
- Địa chỉ(address):
 - Thuộc tính: MaDiaChi, Ten, ThanhPho, Quan, Huyen, Duong, SoDienThoai, MaNguoiDung, Hien.
 - Mô tả: Đại diện cho các thông tin của người dùng

3.2.2. Đối tượng ngoài

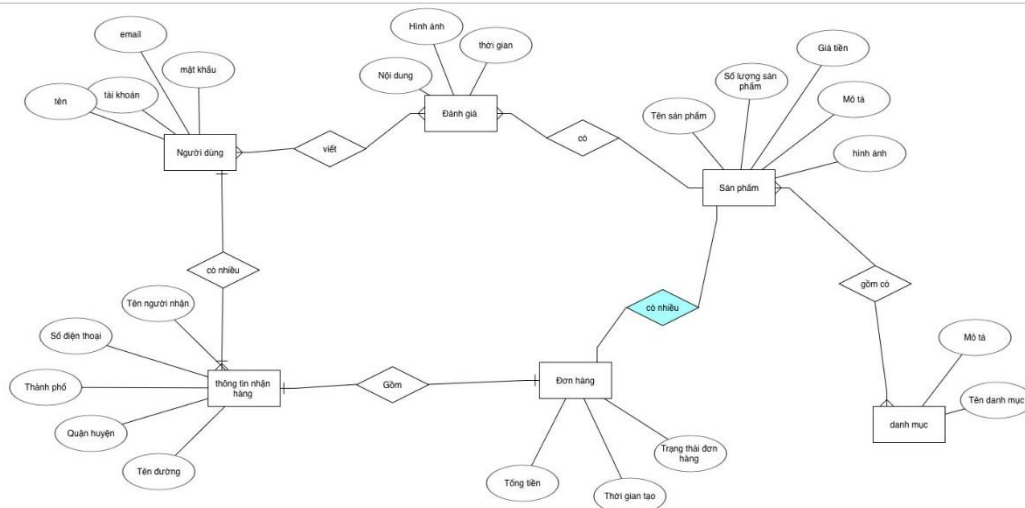
- Người Dùng (User):
 - Thuộc tính: TenDangNhap, Email, HoTen, MatKhai, VaiTro.
 - Mô tả: Đại diện cho tài khoản người dùng, bao gồm khách hàng và quản trị viên.

3.2.3. Nghiệp vụ

- Đơn hàng (Order):
 - Thuộc tính: MaDonHang, TenKhachHang, Email, SoDienThoai, NgayDatHang,
 - TongSoTien, TrangThai
 - Mô tả: Đại diện cho đơn hàng của khách hàng, bao gồm thông tin liên hệ và tổng số tiền.

3.3. MÔ HÌNH ERD (ENTITY-RELATIONSHIP DIAGRAM)

3.3.1. Phân tích mô hình ERD



Hình 3. 1. Mô hình ERD trên draw.io

3.4. MÔ HÌNH QUAN HỆ

3.4.1. Mô tả các quan hệ

- LoaiSanPham(MaLoai, TenLoaiPhong)

Tên từ: Mỗi loại sản phẩm được phân biệt với nhau thông qua mã loại (MaLoai). Thông tin của loại sản phẩm bao gồm tên loại (TenLoai).

Ràng buộc: Mã loại (MaLoai) là một số nguyên tự động tăng. Tên loại (TenLoai) là thuộc tính không NULL và được ghi nhận với tối đa 50 ký tự.

- Chitietloaisanpham(MaSanPham, MaLoai)

Ràng buộc: mỗi cặp mã loại và mã sản phẩm là duy nhất.

- SanPham(MaSanPham, TenSanPham, Gia, MoTa, SoLuongTonKho, TrangThaiHienThi, MaLoai)

Tên từ: Mỗi sản phẩm được phân biệt với nhau thông qua mã sản phẩm (MaSanPham). Thông tin của sản phẩm bao gồm tên sản phẩm (TenSanPham), giá (Gia), mô tả (MoTa), số lượng tồn kho (SoLuongTonKho), trạng thái hiển thị (TrangThaiHienThi) và mã loại (MaLoai).

Ràng buộc: Mã sản phẩm (MaSanPham) là một số nguyên tự động tăng. Tên sản phẩm (TenSanPham) là thuộc tính không NULL và được ghi nhận với tối đa 100 ký tự. Giá (Gia) và số lượng tồn kho (SoLuongTonKho) là các thuộc tính không NULL. Mã loại (MaLoai) là khóa ngoại tham chiếu đến bảng LoaiSanPham.

- HinhAnhSanPham(MaHinhAnh, DuongDanHinhAnh, MaSanPham)

Tên từ: Mỗi hình ảnh sản phẩm được phân biệt với nhau thông qua mã hình ảnh (MaHinhAnh). Thông tin của hình ảnh bao gồm đường dẫn hình ảnh (DuongDanHinhAnh) và mã sản phẩm (MaSanPham).

Ràng buộc: Mã hình ảnh (MaHinhAnh) là một số nguyên tự động tăng. Đường dẫn hình ảnh (DuongDanHinhAnh) là thuộc tính không NULL. Mã sản phẩm (MaSanPham) là khóa ngoại tham chiếu đến bảng SanPham.

- DonHang(MaDonHang, NgayDatHang, TrangThaiDonHang, TenKhachHang, SoDienThoai, ThanhPho, QuanHuyen, DiaChi, GhiChu, MaNguoiDung)

Tên từ: Mỗi đơn hàng được phân biệt với nhau thông qua mã đơn hàng (MaDonHang). Thông tin của đơn hàng bao gồm ngày đặt hàng (NgayDatHang), trạng thái đơn hàng (TrangThaiDonHang), tên khách hàng (TenKhachHang), số điện thoại

(SoDienThoai), thành phố (ThanhPho), quận/huyện (QuanHuyen), địa chỉ (DiaChi), ghi chú (GhiChu) và mã người dùng (MaNguoiDung).

Ràng buộc: Mã đơn hàng (MaDonHang) là một số nguyên tự động tăng. Ngày đặt hàng (NgayDatHang) và trạng thái đơn hàng (TrangThaiDonHang) là các thuộc tính không NULL. Tên khách hàng (TenKhachHang) và số điện thoại (SoDienThoai) là các thuộc tính không NULL, tên khách hàng được ghi nhận với tối đa 100 ký tự và số điện thoại với tối đa 20 ký tự. Mã người dùng (MaNguoiDung) là khóa ngoại tham chiếu đến bảng ApplicationUser.

- ChiTietDonHang(MaChiTietDonHang, SoLuong, DonGia, MaDonHang, MaSanPham)

Tên từ: Mỗi chi tiết đơn hàng được phân biệt với nhau thông qua mã chi tiết đơn hàng (MaChiTietDonHang). Thông tin của chi tiết đơn hàng bao gồm số lượng (SoLuong), đơn giá (DonGia), mã đơn hàng (MaDonHang) và mã sản phẩm (MaSanPham).

Ràng buộc: Mã chi tiết đơn hàng (MaChiTietDonHang) là một số nguyên tự động tăng. Số lượng (SoLuong) và đơn giá (DonGia) là các thuộc tính không NULL. Mã đơn hàng (MaDonHang) và mã sản phẩm (MaSanPham) là các khóa ngoại tham chiếu đến bảng DonHang và SanPham.

Địa chỉ

Bình luận

3.4.2. Danh sách thông tin các bảng

Bảng 3. 1. Bảng loại sản phẩm

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaLoai	Mã loại	INT		PK
TenLoai	Tên loại	NVARCHAR	50	NOT NULL

Bảng 3. 2. Bảng sản phẩm

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaSanPham	Mã sản phẩm	INT		PK
TenSanPham	Tên sản phẩm	NVARCHAR	100	NOT NULL
Gia	Giá	DECIMAL(18, 2)		NOT NULL
MoTa	Mô tả	NVARCHAR	MAX	
SoLuongTonKho	Số lượng tồn kho	INT		NOT NULL
TrangThaiHienThi	Trạng thái hiển thị	BIT		NOT NULL
MaLoai	Mã loại	INT		FK
Chitietloaisanpham		INT		PFK

Bảng 3. 3. Bảng hình ảnh sản phẩm

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaHinhAnh	Mã hình ảnh	INT		PK
DuongDanHinhAnh	Đường dẫn hình ảnh	NVARCHAR	MAX	NOT NULL
MaSanPham	Mã sản phẩm	INT		FK

Bảng 3. 4. Bảng tin tức

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaBaiViet	Mã bài viết	INT		PK
TieuDe	Tiêu đề	NVARCHAR	200	NOT NULL

NoiDung	Nội dung	NVARCHAR	MAX	NOT NULL
TrangThaiHienThi	Trạng thái hiển thị	BIT		
DuongDanAnh	Đường dẫn ảnh	NVARCHAR	MAX	
NgayTao	Ngày tạo	DATETIME		NOT NULL
MaNguoiTao	Mã người tạo	INT		FK

Bảng 3. 4. Bảng đơn hàng

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaDonHang	Mã đơn hàng	INT		PK
NgayDatHang	Ngày đặt hàng	DATETIME		NOT NULL
TrangThaiDonHang	Trạng thái đơn hàng	NVARCHAR	50	NOT NULL
TenKhachHang	Tên khách hàng	NVARCHAR	100	NOT NULL
SoDienThoai	Số điện thoại	NVARCHAR	20	NOT NULL
ThanhPho	Thành phố	NVARCHAR	100	NOT NULL
QuanHuyen	Quận/Huyện	NVARCHAR	100	NOT NULL
DiaChi	Địa chỉ	NVARCHAR	200	NOT NULL
GhiChu	Ghi chú	NVARCHAR	MAX	
MaNguoiDung	Mã người dùng	INT		FK

Bảng 3. 5. Bảng chi tiết đơn hàng

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
MaChiTietDonHang	Mã chi tiết đơn hàng	INT		PK

SoLuong	Số lượng	INT		NOT NULL
DonGia	Đơn giá	DECIMAL(18, 2)		NOT NULL
MaDonHang	Mã đơn hàng	INT		FK
MaSanPham	Mã sản phẩm	INT		FK

Bảng 3. 6. Bảng người dùng

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
Id	Mã người dùng	INT		PK
UserName	Tên đăng nhập	NVARCHAR	256	NOT NULL
Email	Email	NVARCHAR	256	
PasswordHash	Mã hash mật khẩu	NVARCHAR	MAX	

Bảng 3. 7. Bảng quyền

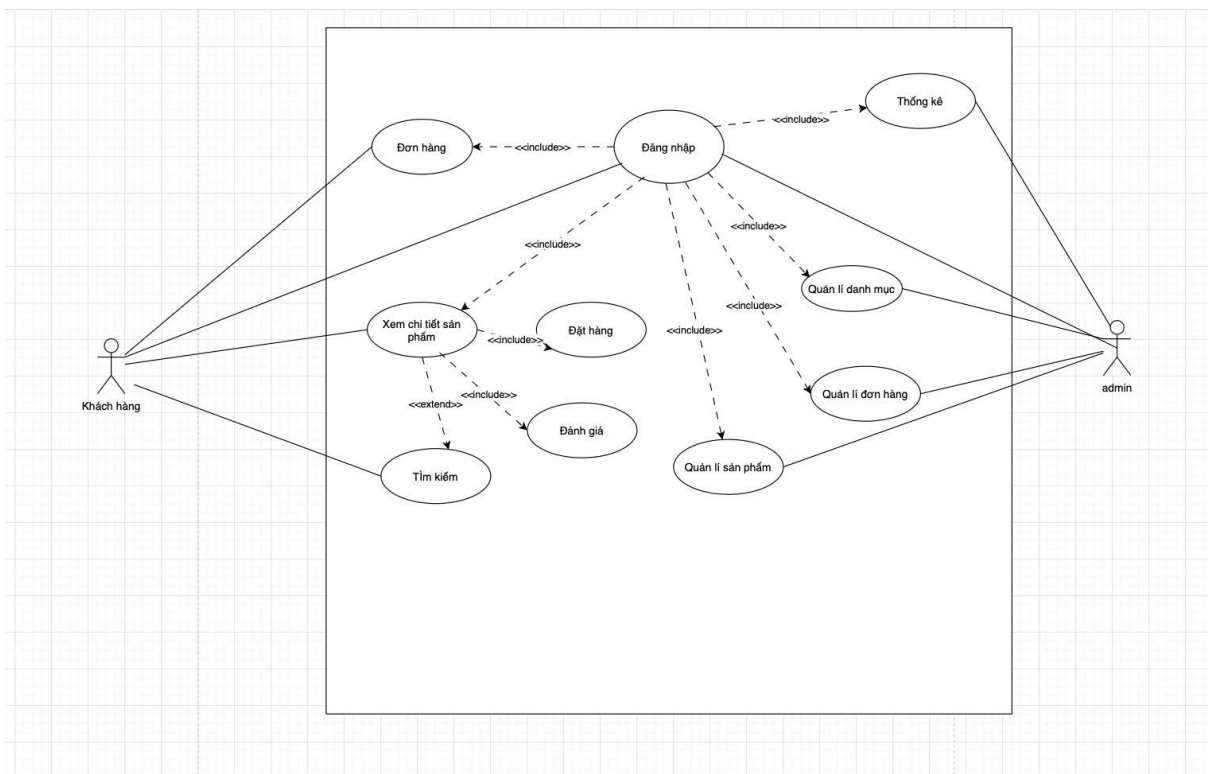
Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
Id	Mã quyền	UNIQUEIDENTIFIER	36	PK
Name	Tên quyền	NVARCHAR	256	NOT NULL

Bảng 3. 8. Bảng quyền người dùng

Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Độ dài	Ràng buộc
UserId	Mã người dùng	INT		FK
RoleId	Mã quyền	UNIQUEIDENTIFIER	36	FK

3.5. SƠ ĐỒ USE CASE

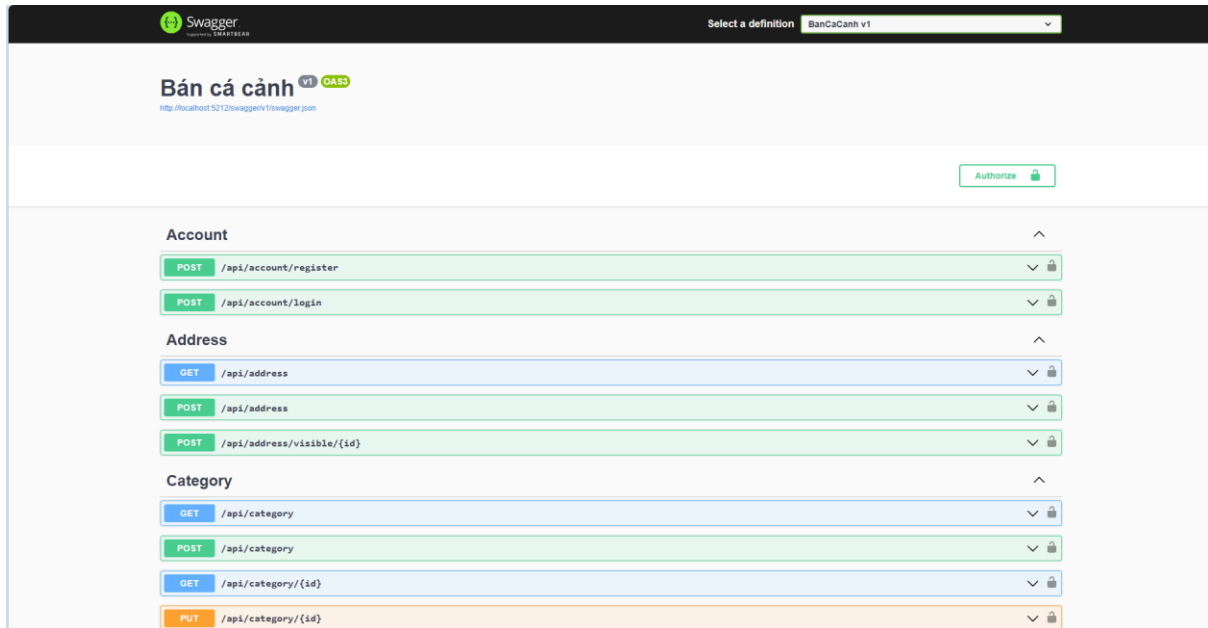
Các Use Case được thiết kế cho hệ thống



Hình 3. 2. Sơ đồ use case

CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM

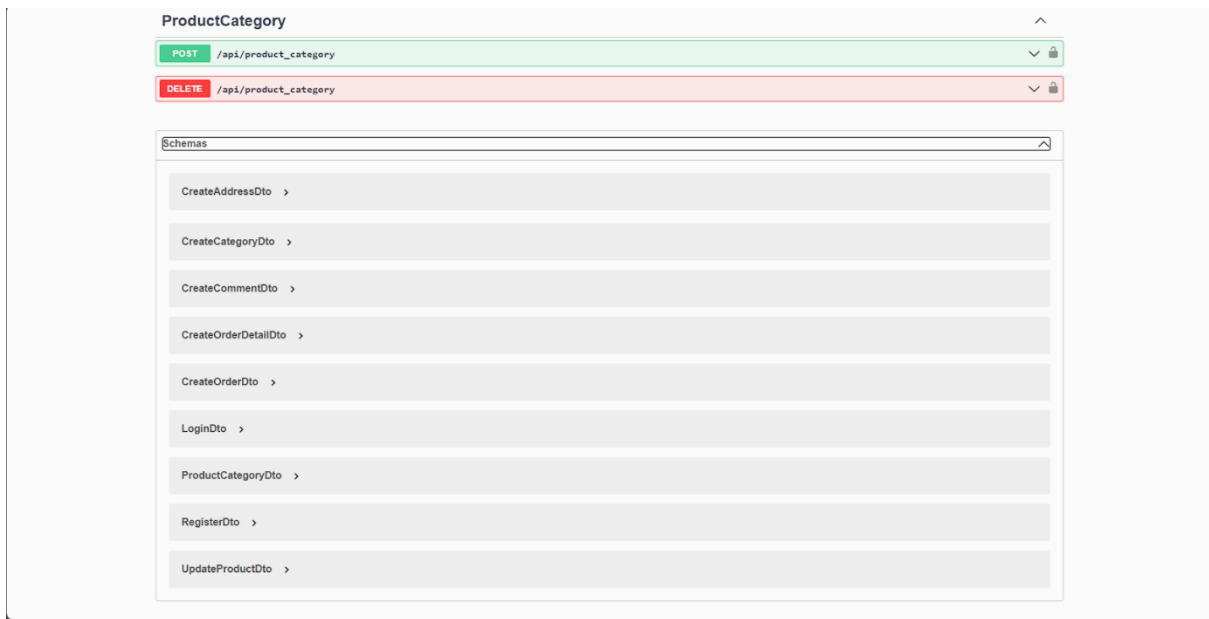
4.1. GIAO DIỆN API



Hình 4. 1. Giao diện API phần 1

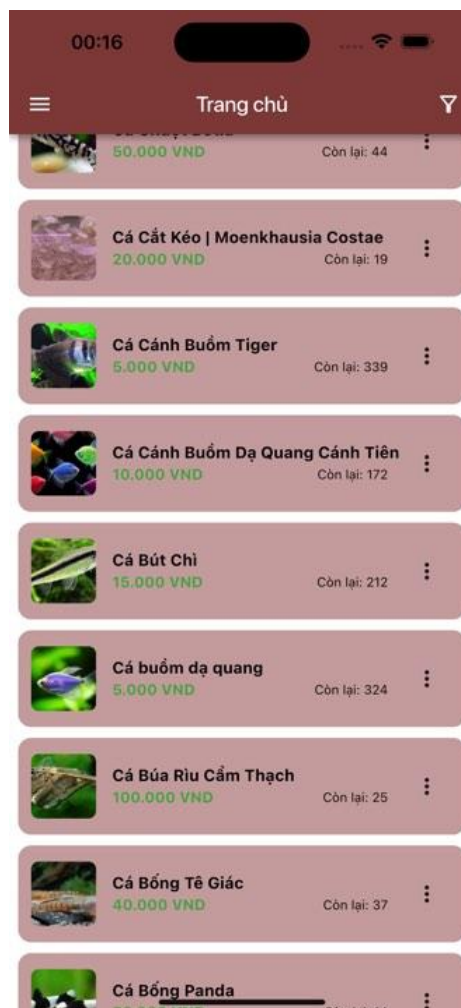


Hình 4. 2. Giao diện API phần 2

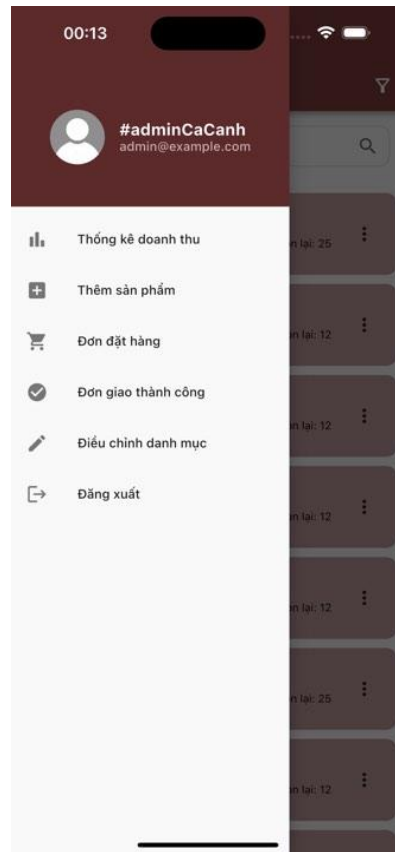


Hình 4. 3. Giao diện API phần 3

4.2. GIAO DIỆN ADMIN



Hình 4. 4. Giao diện trang chủ admin



Hình 4. 5. Giao diện các tính năng của admin

00:14

< Thêm mới sản phẩm

Tên sản phẩm

Số lượng sản phẩm

Mô tả sản phẩm

Giá sản phẩm

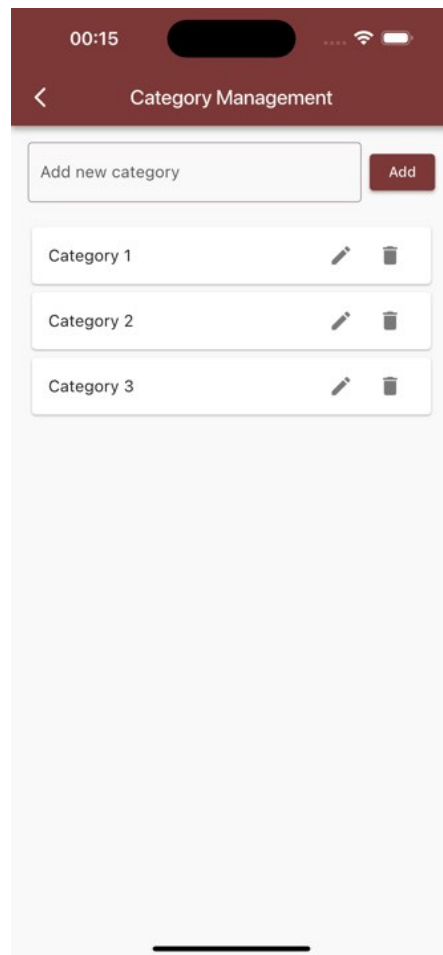
Hàng thuộc loại ▼

Hình ảnh (được chọn tối đa 4):

Chọn hình 0 / 4 đã chọn

Thêm sản phẩm

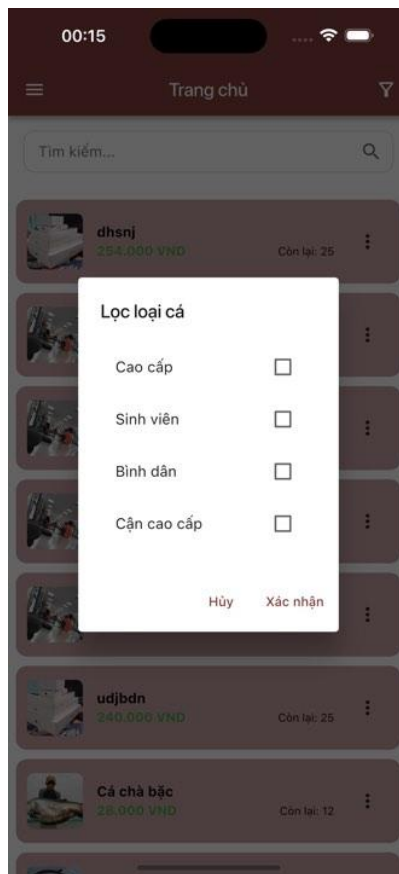
Hình 4. 6. Giao diện thêm sản phẩm



Hình 4. 7. Giao diện điều chỉnh danh mục



Hình 4. 8. Giao diện tìm kiếm sản phẩm

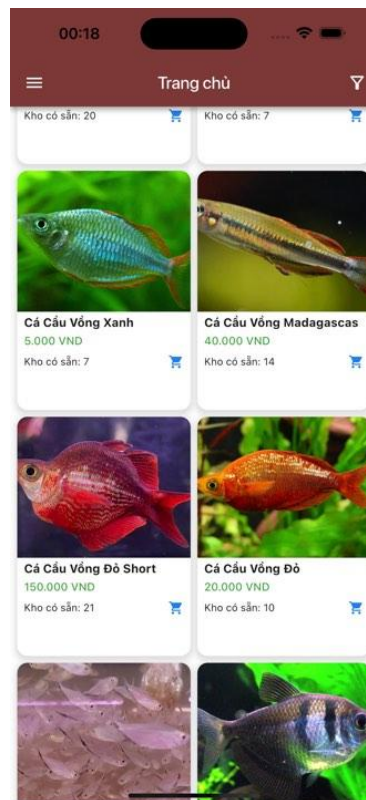


Hình 4. 9. Giao diện lọc sản phẩm

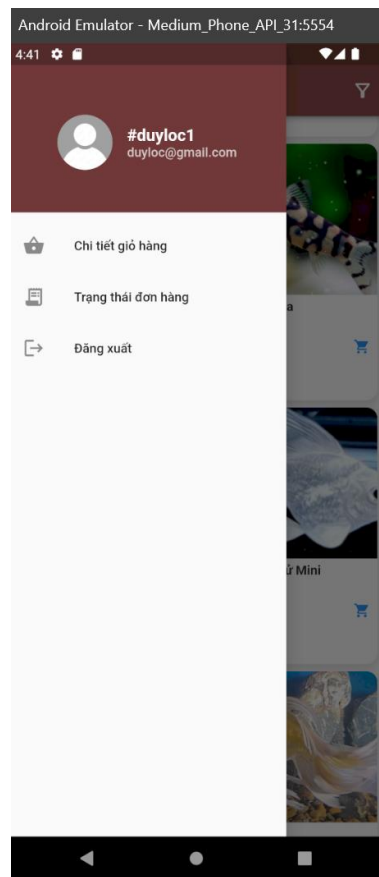


Hình 4. 10. Giao diện thêm, xóa, sửa

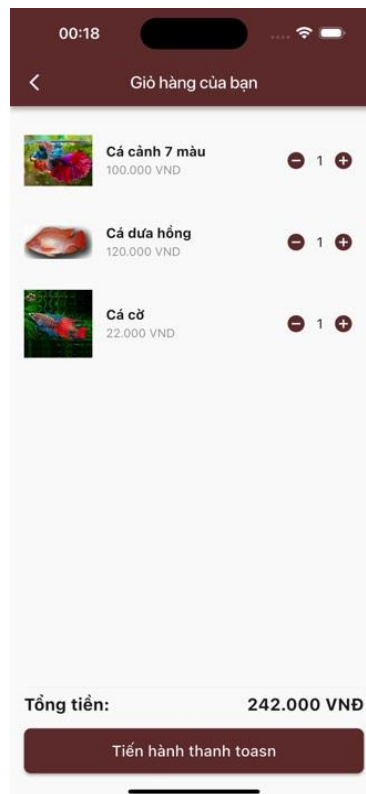
4.3. GIAO DIỆN KHÁCH HÀNG



Hình 4. 11. Giao diện trang chủ



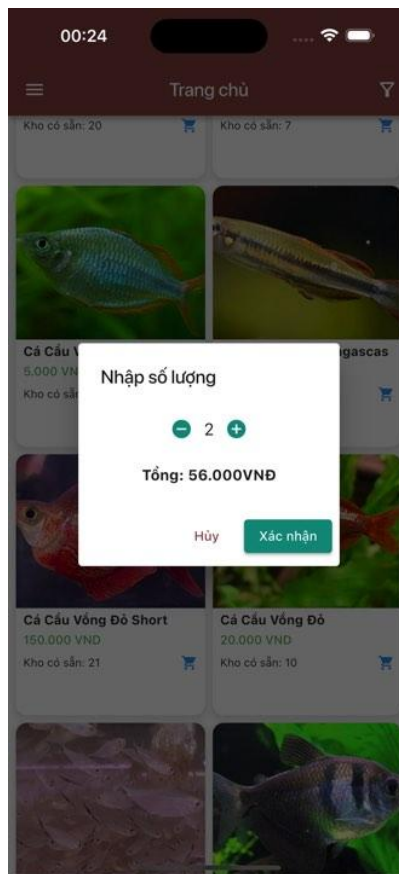
Hình 4. 12. Giao diện chức năng



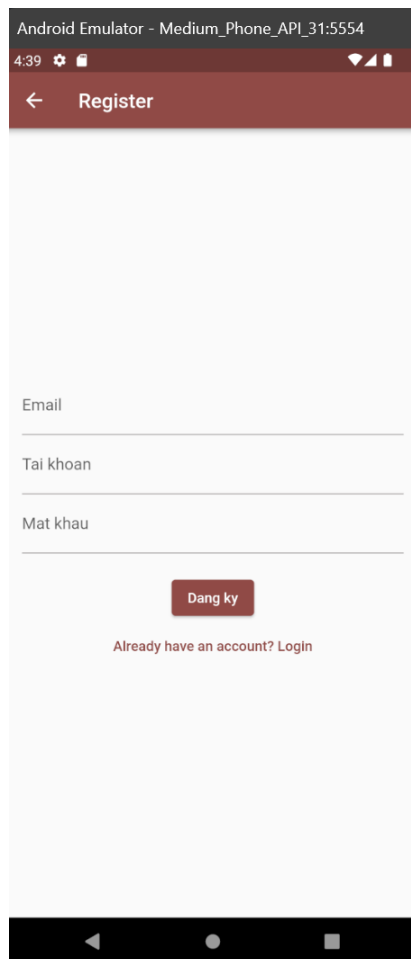
Hình 4. 13. Giao diện giỏ hàng



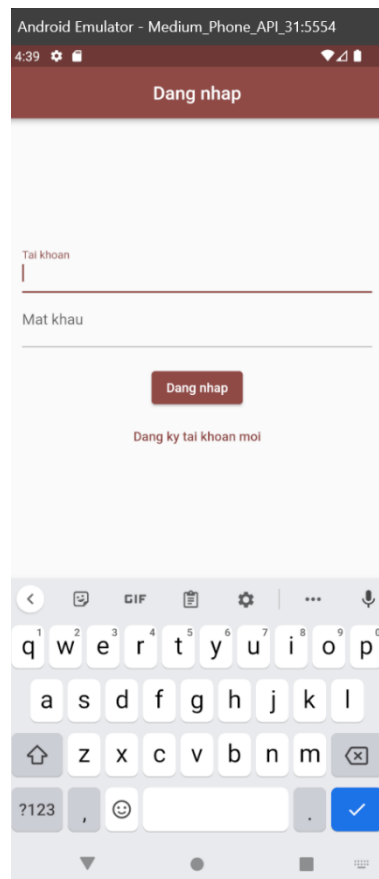
Hình 4. 14. Giao diện trạng thái đơn hàng



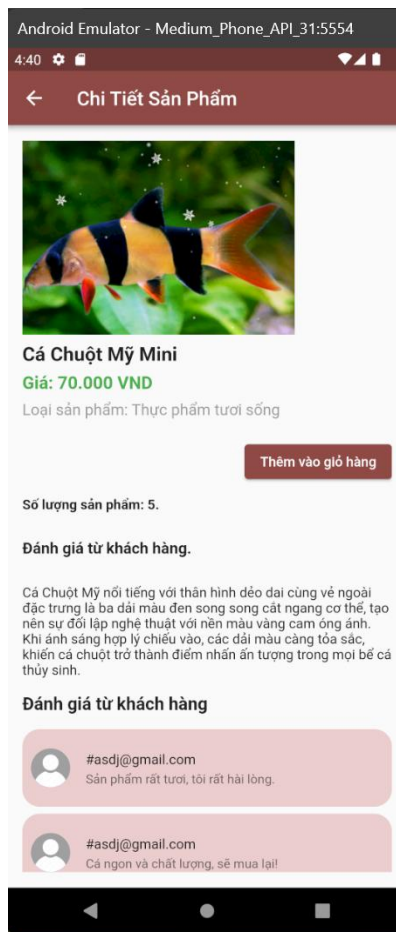
Hình 4. 15. Giao diện nhập số lượng để thêm vào giỏ hàng



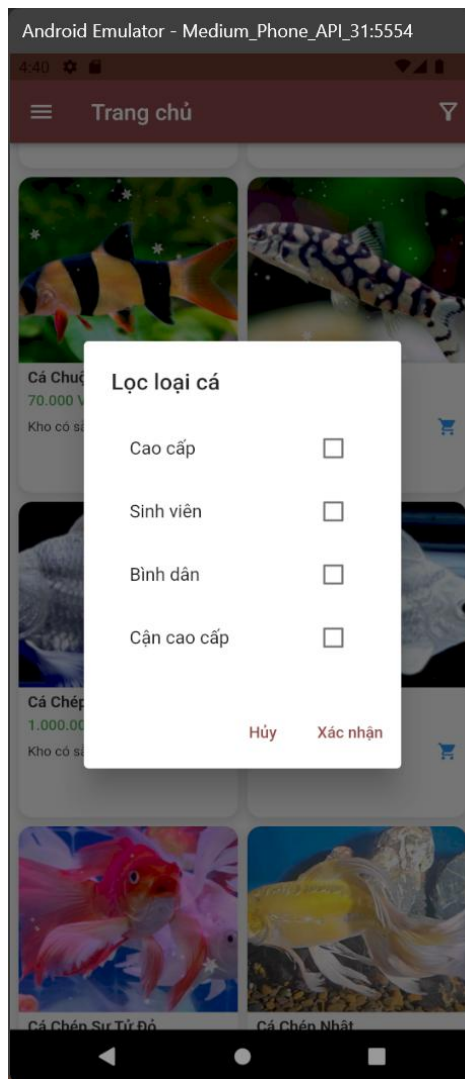
Hình 4. 16. Giao diện đăng kí



Hình 4. 17. Giao diện đăng nhập tài khoản



Hình 4. 18. Giao diện chi tiết sản phẩm



Hình 4. 19. Giao diện lọc loại cá

CHƯƠNG 5 KẾT LUẬN VỀ KIẾN NGHỊ

5.1. KẾT LUẬN CHUNG

5.1. *Kết luận chung*

Hiểu được cách vận hành của hệ thống quản lý bán cá cảnh cơ bản. Có khả năng sử dụng tốt các ngôn ngữ và công nghệ liên quan như dart, C#, và các mô hình kiến trúc phổ biến. Nắm bắt được nhu cầu thực tế của việc quản lý bán cá cảnh và xây dựng hệ thống phù hợp với yêu cầu của người dùng. Qua quá trình thực hiện dự án, nhóm đã nhận thức rõ hơn về vai trò quan trọng của công nghệ thông tin trong việc cung cấp giải pháp hiện đại cho quản lý kinh doanh.

5.2. *Kết quả đạt được*

- Thiết kế giao diện người dùng thân thiện và dễ sử dụng.
- Xây dựng hệ thống quản lý sản phẩm, đơn hàng và khách hàng một cách hiệu quả.
- Triển khai thành công ứng dụng app mobile trên các nền tảng hosting phổ biến.
- Quản lý cơ sở dữ liệu ổn định và an toàn.

5.3. *Kết quả chưa đạt được*

Tuy nhiên, còn một số hạn chế và chức năng chưa được hoàn thiện, bao gồm:

- Chưa thực hiện được một số chức năng như tính năng quản lý vận chuyển và thanh toán trực tuyến.
- Dữ liệu trên hệ thống chưa được đầy đủ và đúng đắn.
- Phần giao diện người dùng còn cần được cải thiện để tối ưu trải nghiệm người dùng.
- Hiệu suất của hệ thống cần được cải thiện để đảm bảo tính linh hoạt và nhanh chóng trong quá trình sử dụng.

5.4. *Hướng phát triển*

Để cải thiện và phát triển hệ thống, nhóm có thể xem xét các hướng phát triển sau:

- Triển khai tính năng quản lý vận chuyển và thanh toán trực tuyến để cải thiện quy trình mua bán.
- Tối ưu hóa giao diện người dùng để tăng tính thẩm mỹ và sự thoải mái khi sử dụng.
- Mở rộng chức năng quản lý kho để tối ưu hóa quy trình nhập/xuất hàng.

TÀI LIỆU THAM KHẢO

- [1]. <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>
- [2]. <https://learn.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity>
- [3]. <https://www.entityframeworktutorial.net/>
- [4]. <https://getbootstrap.com/docs/5.2/getting-started/introduction/>