

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN



BÀI TẬP LỚN LẬP TRÌNH WEB VỚI MVC
XÂY DỰNG TRANG WEB BÁN ĐỒ CÔNG NGHỆ

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

SINH VIÊN: ĐỖ THÀNH TÔN
MÃ LỚP: 101195
GIẢNG VIÊN: TRẦN ĐỖ THU HÀ

HƯNG YÊN – 2021

Nhận xét của giảng viên hướng dẫn:

[illegible]

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan Bài tập lớn “Xây dựng trang web bán đồ công nghệ” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của cô Trần Đỗ Thu Hà.

Những phần sử dụng tài liệu tham khảo trong Bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong Bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

Hưng Yên, ngày ... tháng ... năm.....

Sinh viên

.....

LỜI CẢM ƠN

Để có thể hoàn thành Bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện Bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn cô Trần Đỗ Thu Hà đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện Bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được Bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong Bài tập lớn.

Em xin trân trọng cảm ơn!

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ	7
DANH MỤC CÁC BẢNG	8
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ	9
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	10
1.1. Lý do chọn đề tài.....	10
1.2. Mục tiêu của đề tài	10
1.2.1 Mục tiêu tổng quát.....	10
1.2.2 Mục tiêu cụ thể	10
1.3. Giới hạn và phạm vi của đề tài.....	10
1.3.1 Đối tượng nghiên cứu.....	10
1.3.2 Phạm vi nghiên cứu	10
1.4. Nội dung thực hiện.....	10
1.5. Phương pháp tiếp cận.....	11
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	12
2.1. Quy trình phát triển phần mềm	12
2.2. Phương pháp phân tích thiết kế hướng đối tượng.....	15
2.3. Tổng quan về ASP.NET MVC	20
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	24
3.1 Đặc tả yêu cầu phần mềm	24
3.1.1 Các yêu cầu chức năng.....	24
3.1.2 Các yêu cầu phi chức năng.....	28
3.2 Thiết kế hệ thống.....	28
3.2.1 Thiết kế lớp đối tượng	28
3.2.2 Thiết kế cơ sở dữ liệu	36

3.2.3 Thiết kế giao diện	38
CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG.....	43
4.1 Triển khai các chức năng của hệ thống	43
4.1.1 Chức năng hiển thị danh sách sản phẩm	43
4.1.2 Chức năng Xem chi tiết sản phẩm.....	45
4.2 Kiểm thử và triển khai ứng dụng	50
4.2.1 Kiểm thử.....	50
4.2.2 Đóng gói ứng dụng.....	51
4.2.3 Triển khai ứng dụng	51
KẾT LUẬN	53
TÀI LIỆU THAM KHẢO	54

DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng anh	Diễn giải
1	CSDL	Data base	Cơ sở dữ liệu
2	SQL	Struct query language	Ngôn ngữ truy vấn có cấu trúc
3	MVC	Model View Controller	Mẫu kiến trúc phần mềm để tạo giao diện người dùng trên máy tính

DANH MỤC CÁC BẢNG

Bảng 3.1. Dòng sự kiện chính đăng nhập.....	24
Bảng 3.2. Dòng sự kiện quản lý hàng hóa.....	25
Bảng 3.3. Dòng sự kiện quản lý loại hàng	27
Bảng 3.4. Khảo sát bài toán.....	27
Bảng 3.5. Phân tích yêu cầu chức năng.....	27
Bảng 3.6. Các yêu cầu phi chức năng.....	28
Bảng 3.7. Danh sách thuộc tính Account	33
Bảng 3.8. Danh sách phương thức Account.....	33
Bảng 3.9. Danh sách thuộc tính News.....	33
Bảng 3.10. Danh sách phương thức News	33
Bảng 3.11. Danh sách thuộc tính Category	34
Bảng 3.12. Danh sách phương thức ContentChat	34
Bảng 3.13. Danh sách thuộc tính Brand	34
Bảng 3.14. Danh sách phương thức Brand.....	34
Bảng 3.15. Danh sách thuộc tính Product	35
Bảng 3.16. Danh sách phương thức Product	35
Bảng 3.17. Danh sách thuộc tính Cart.....	36
Bảng 3.18. Danh sách phương thức WatchVideo	36
Bảng 4.1. Kiểm thử cho chức năng đăng ký tài khoản.....	50
Bảng 4.2. Kiểm thử cho chức năng đăng nhập.....	50
Bảng 4.3. Kiểm thử cho chức năng chính	51

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 2.1. Giai đoạn thiết kế	13
Hình 2.2. Các View trong OOAD sử dụng UML	17
Hình 2.3. Các bản vẽ trong OOAD sử dụng UML	18
Hình 2.4. Ký hiệu về Use Case	19
Hình 2.5. Ký hiệu về Class	19
Hình 2.6. Ký hiệu về Actor	19
Hình 2.7: Cấu trúc entity framework	21
Hình 3.1. Biểu đồ usecase Tổng quát	24
Hình 3.2 Biểu đồ usecase Quản lý hàng hóa	25
Hình 3.3 Biểu đồ usecase Quản lý nhà sản xuất	26
Hình 3.4. Biểu đồ usecase Quản lý loại hàng	26
Hình 3.5. Biểu đồ tuần tự thêm sản phẩm	28
Hình 3.6. Biểu đồ tuần tự sửa sản phẩm	29
Hình 3.7. Biểu đồ tuần tự xóa sản phẩm	29
Hình 3.8. Biểu đồ lớp	30
Hình 3.9. Biểu đồ chi tiết quản lý bán hàng	31
Hình 3.10. Biểu đồ chi tiết quản lý loại sản phẩm	31
Hình 3.11. Biểu đồ chi tiết quản lý Sản phẩm	32
Hình 3.12. Biểu đồ Quản lý NSX	32
Hình 3.13. Biểu đồ Cơ sở dữ liệu	37
Hình 3.16. Giao diện trang chủ	39
Hình 3.17. Giao diện sản phẩm theo danh mục	39
Hình 3.14. Giao diện chi tiết sản phẩm	40
Hình 3.15. Giao diện giỏ hàng	40
Hình 3.18. Giao diện đặt hàng	41
Hình 3.19. Giao diện tình trạng đơn hàng	41
Hình 3.20. Giao diện trang quản trị	42
Hình 4.1 Đóng gói ứng dụng	51

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

- Tầm quan trọng, vai trò của đề tài

Ngày nay công nghệ thông tin ngày phát triển, nên việc bán hàng online trở thành một xu thế mới. Vì vậy em quyết định xây dựng trang web bán hàng để giúp mọi người có thể sở hữu những món đồ công nghệ mà không phải tới tận cửa hàng.

- Tính cấp thiết của đề tài
 - o Giúp việc mua bán hàng trở lên dễ dàng hơn
 - o Thời kì giãn cách giúp mọi người không phải đến những nơi mà chỉ cần đặt online

1.2. Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

Xây dựng trang web bán đồ công nghệ

1.2.2 Mục tiêu cụ thể

Phát triển trang web gồm các chức năng:

- Đăng nhập
- Quản lý thông tin sản phẩm
- Hiển thị danh sách sản phẩm
- Đặt hàng
- Tìm kiếm sản phẩm

1.3. Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

- Đối tượng nghiên cứu: Mọi người
- Các trang web bán hàng hiện nay

1.3.2 Phạm vi nghiên cứu

- Phạm vi không gian: trên mạng xã hội
- Phạm vi thời gian: Học kì 1 năm 3

1.4. Nội dung thực hiện

- Nghiên cứu về đồ công nghệ
- Tìm hiểu về mọi người xung quanh

- Phân tích thiết kế phần mềm
- Lập trình ứng dụng
- Kiểm thử ứng dụng

1.5. Phương pháp tiếp cận

- Tham khảo một số hệ thống phần mềm liên quan đến bán hàng
- Khảo sát yêu cầu của người dùng về việc mua hàng
- Tìm hiểu thông tin khác liên quan tới phần mềm

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Quy trình phát triển phần mềm

Giai đoạn 1: Khảo sát dự án

Khảo sát hiện trạng là giai đoạn đầu tiên trong quá trình phát triển một hệ thống thông tin. Nhiệm vụ chính trong giai đoạn này là tìm hiểu, thu thập thông tin cần thiết để chuẩn bị cho việc giải quyết các yêu cầu được đặt ra của dự án. Giai đoạn khảo sát được chia làm hai bước:

Bước 1:

- Khảo sát sơ bộ: tìm hiểu các yếu tố cơ bản (tổ chức, văn hóa, đặc trưng, con người,...) tạo tiền đề để phát triển PMMT phù hợp với dự án và doanh nghiệp.
- Khảo sát chi tiết: thu thập thông tin chi tiết của hệ thống (chức năng xử lý, thông tin được phép nhập và xuất khỏi hệ thống, ràng buộc, giao diện cơ bản, nghiệp vụ) phục vụ cho việc phân tích và thiết kế.

Bước 2: Đặt ra các vấn đề trọng tâm cần phải giải quyết, như:

- Thông tin đưa vào hệ thống phải như thế nào?
- Dữ liệu hiển thị và xuất ra khác nhau ở những điểm nào?
- Ràng buộc giữa các đối tượng trong hệ thống cần xây dựng ra sao?
- Chức năng và quy trình xử lý của hệ thống phải đảm bảo những yêu cầu nào?
- Cần sử dụng những giải pháp nào? Tính khả thi của từng giải pháp ra sao?

Từ những thông tin thu thập được và vấn đề đã đặt ra trong giai đoạn khảo sát, nhà quản trị và các chuyên gia sẽ chọn lọc những yếu tố cần thiết để cấu thành hệ thống thông tin riêng cho doanh nghiệp.

Giai đoạn 2: Phân tích hệ thống

Mục tiêu của giai đoạn là xác định các thông tin và chức năng xử lý của hệ thống, cụ thể như sau:

- Xác định *yêu cầu của PMMT* gồm: các chức năng chính – phụ; nghiệp vụ cần phải xử lý đảm bảo tính chính xác, tuân thủ đúng các văn bản luật và quy định hiện hành; đảm bảo tốc độ xử lý và khả năng nâng cấp trong tương lai.
- Phân tích và đặc tả *mô hình phân cấp chức năng tổng thể* thông qua sơ đồ BFD (Business Flow Diagram), từ mô hình BFD sẽ tiếp tục được xây dựng thành mô hình luồng dữ liệu DFD (Data Flow Diagram) thông qua quá trình phân rã chức năng theo các mức 0, 1, 2 ở từng ô xử lý.

- Phân tích *bảng dữ liệu*. Cần đưa vào hệ thống những bảng dữ liệu (data table) gồm các trường dữ liệu (data field) nào? Xác định khóa chính (primary key), khóa ngoại (foreign key) cũng như mối quan hệ giữa các bảng dữ liệu (relationship) và ràng buộc (constraint) dữ liệu cần thiết.

Ở giai đoạn này, các chuyên gia sẽ đặc tả sơ bộ các bảng dữ liệu trên giấy để có cái nhìn khách quan. Qua đó, xác định các giải pháp tốt nhất cho hệ thống đảm bảo đúng các yêu cầu đã khảo sát trước khi thực hiện trên các phần mềm chuyên dụng.

Giai đoạn 3: Thiết kế

Thông qua thông tin được thu thập từ quá trình khảo sát và phân tích, các chuyên gia sẽ chuyên hóa vào phần mềm, công cụ chuyên dụng để đặc tả thiết kế hệ thống chi tiết. Giai đoạn này được chia làm hai bước sau:



Hình 2.1. Giai đoạn thiết kế

Trên cơ sở các bảng dữ liệu đã phân tích và đặc tả trên giấy sẽ được thiết kế dưới dạng mô hình mức ý niệm bằng phần mềm chuyên dụng như Sybase PowerDesigner, CA ERwin Data Modeler. Bằng mô hình mức ý niệm sẽ cho các chuyên gia có **cái nhìn tổng quát nhất về mối quan hệ giữa các đối tượng** trước khi chuyển đổi thành mô hình mức vật lý.

Bước 2: Thiết kế chi tiết

- Thiết kế cơ sở dữ liệu (Database): Với mô hình mức vật lý hoàn chỉnh ở giai đoạn thiết kế đại thể sẽ được kết sinh mã thành file sql.
- Thiết kế truy vấn, thủ tục, hàm: thu thập, xử lý thông tin nhập và đưa ra thông tin chuẩn xác theo đúng nghiệp vụ.

- Thiết kế giao diện chương trình đảm bảo phù hợp với môi trường, văn hóa và yêu cầu của doanh nghiệp thực hiện dự án.
- Thiết kế chức năng chương trình đảm bảo tính logic trong quá trình nhập liệu và xử lý cho người dùng.
- Thiết kế báo cáo. Dựa trên các yêu cầu của mỗi doanh nghiệp và quy định hiện hành sẽ thiết kế các mẫu báo cáo phù hợp hoặc cho phép doanh nghiệp tự tạo mẫu báo cáo ngay trên hệ thống.
- Thiết kế các kiểm soát bằng hình thức đưa ra các thông báo, cảnh báo hoặc lỗi cụ thể tạo tiện lợi và kiểm soát chặt chẽ quá trình nhập liệu với mục tiêu tăng độ chính xác cho dữ liệu.

Tóm lại, *thiết kế là việc áp dụng các công cụ, phương pháp, thủ tục để tạo ra mô hình hệ thống cần sử dụng*. Sản phẩm cuối cùng của giai đoạn thiết kế là đặc tả hệ thống ở dạng nó tồn tại thực tế, sao cho nhà lập trình và kỹ sư phần cứng có thể dễ dàng chuyển thành chương trình và cấu trúc hệ thống.

Giai đoạn 4: Thực hiện

Đây là giai đoạn nhằm xây dựng hệ thống theo các thiết kế đã xác định. Giai đoạn này bao gồm các công việc sau:

- Lựa chọn hệ quản trị cơ sở dữ liệu (SQL Server, Oracle, MySQL, ...) và cài đặt cơ sở dữ liệu cho hệ thống.
- Lựa chọn công cụ lập trình để xây dựng các modules chương trình của hệ thống (Microsoft Visual Studio, PHP Designer,...).
- Lựa chọn công cụ để xây dựng giao diện hệ thống (DevExpress, Dot Net Bar,...).

Viết tài liệu hướng dẫn sử dụng, tài liệu kỹ thuật hoặc clip hướng dẫn.

Giai đoạn 5: Kiểm thử

- Trước hết phải lựa chọn công cụ kiểm thử.
- Kiểm chứng các modules chức năng của hệ thống thông tin, chuyển các thiết kế thành các chương trình (phần mềm).
- Thử nghiệm hệ thống thông tin.
- Cuối cùng là khắc phục các lỗi (nếu có).
- Viết test case theo yêu cầu.

Kết quả cuối cùng là một hệ thống thông tin đạt yêu cầu đặt ra.

Giai đoạn 6: Triển khai và bảo trì

- Lắp đặt phần cứng để làm cơ sở cho hệ thống.
- Cài đặt phần mềm.
- Chuyển đổi hoạt động của hệ thống cũ sang hệ thống mới, gồm có: chuyển đổi dữ liệu; bố trí, sắp xếp người làm việc trong hệ thống; tổ chức hệ thống quản lý và bảo trì.
- Phát hiện các sai sót, khuyết điểm của hệ thống thông tin.
- Đào tạo và hướng dẫn sử dụng.
- Cải tiến và chỉnh sửa hệ thống thông tin.
- Bảo hành.
- Nâng cấp chương trình khi có phiên bản mới.

2.2. Phương pháp phân tích thiết kế hướng đối tượng

1. Khái niệm về Phân tích và thiết kế hướng đối tượng (Object Oriented Analysis and Design: OOAD)

Trong kỹ nghệ phần mềm để sản xuất được một sản phẩm phần mềm người ta chia quá trình phát triển sản phẩm ra nhiều giai đoạn như thu thập và phân tích yêu cầu, phân tích và thiết kế hệ thống, phát triển (coding), kiểm thử, triển khai và bảo trì. Trong đó, giai đoạn phân tích, thiết kế bao giờ cũng là giai đoạn khó khăn và phức tạp nhất. Giai đoạn này giúp chúng ta hiểu rõ yêu cầu đặt ra, xác định giải pháp, mô tả chi tiết giải pháp. Nó trả lời 2 câu hỏi What (phần mềm này làm cái gì?) và How (làm nó như thế nào?).

Để phân tích và thiết kế một phần mềm thì có nhiều cách làm, một trong những cách làm đó là xem hệ thống gồm những đối tượng sống trong đó và tương tác với nhau. Việc mô tả được tất cả các đối tượng và sự tương tác của chúng sẽ giúp chúng ta hiểu rõ hệ thống và cài đặt được nó. Phương thức này gọi là Phân tích thiết kế hướng đối tượng (OOAD)

2. Khái niệm về UML (Unified Modeling Language)

UML là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau cũng như dùng để thi công hệ thống (phát triển), thuyết phục khách hàng, các nhà đầu tư v.v.. (Giống như trong xây dựng người ta dùng các bản vẽ thiết kế để hướng dẫn và kiểm soát thi công, bán hàng, căn hộ v.v..)

3. Tại sao lại là OOAD và UML?

OOAD cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung thể hiện. Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó nên chúng thường đi đôi với nhau.

4. OOAD sử dụng UML

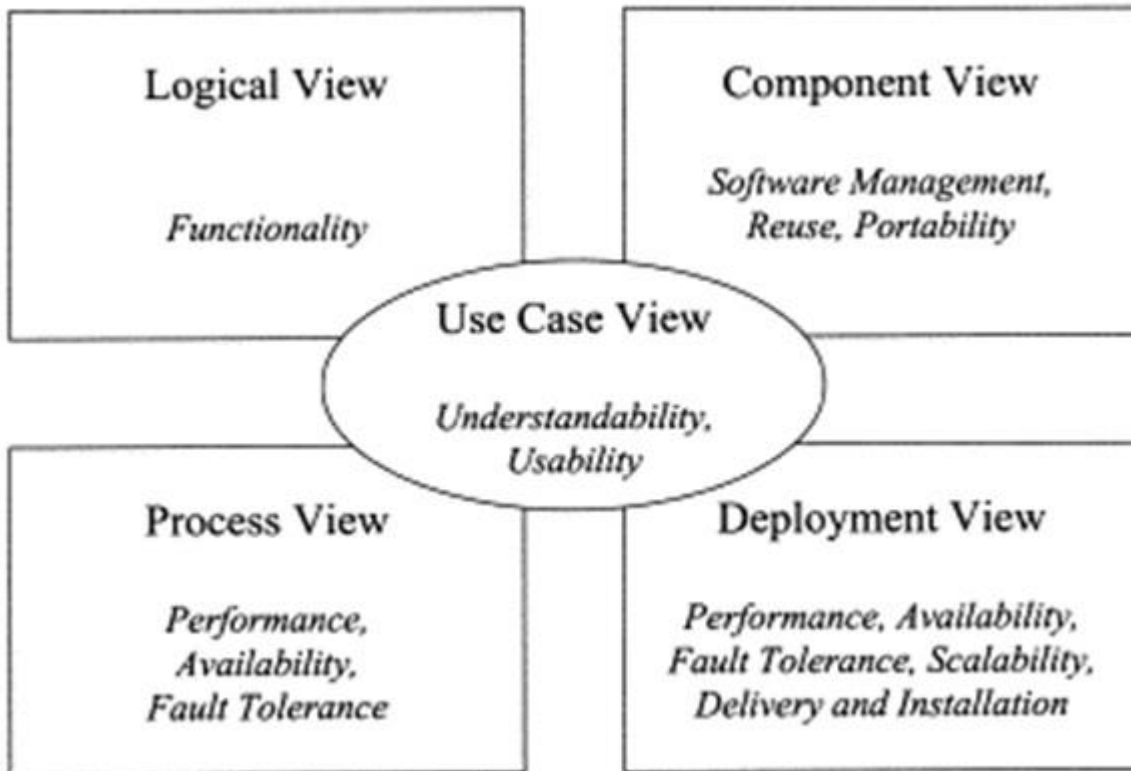
UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng v... trong phạm vi các bài viết này chúng ta chỉ nghiên cứu cách sử dụng UML cho phân tích và thiết kế hướng đối tượng trong ngành phần mềm. OOAD sử dụng UML bao gồm các thành phần sau:

- View (góc nhìn)
- Diagram (bản vẽ)
- Notations (ký hiệu)
- Mechanisms (qui tắc, cơ chế)

Chúng ta sẽ tìm hiểu kỹ hơn các thành phần trên.

4.1 View (góc nhìn)

Mỗi góc nhìn như thầy bói xem voi, nó không thể hiện hết hệ thống nhưng thể hiện rõ hệ thống ở một khía cạnh. Chính vì thế trong xây dựng có bản vẽ kiến trúc (nhìn về mặt kiến trúc), bản vẽ kết cấu (nhìn về mặt kết cấu), bản vẽ thi công (nhìn về mặt thi công). Trong phần mềm cũng như vậy, OOAD sử dụng UML có các góc nhìn sau:



Hình 2.2. Các View trong OOAD sử dụng UML

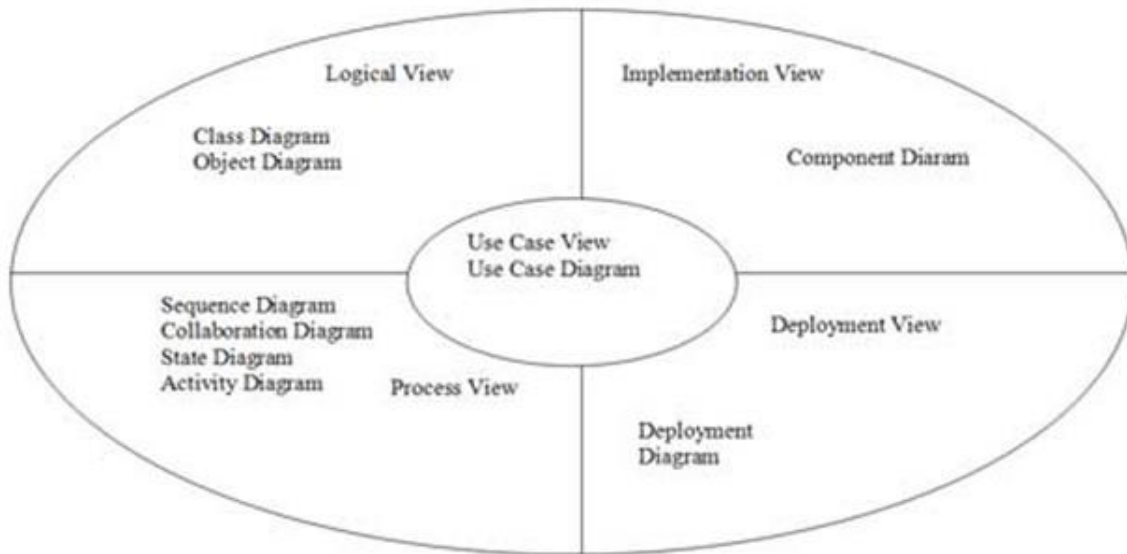
Trong đó,

- **Use Case View:** cung cấp góc nhìn về các ca sử dụng giúp chúng ta hiểu hệ thống có gì? ai dùng và dùng nó như thế nào.
- **Logical View:** cung cấp góc nhìn về cấu trúc hệ thống, xem nó được tổ chức như thế nào. Bên trong nó có gì.
- **Process View:** cung cấp góc nhìn động về hệ thống, xem các thành phần trong hệ thống tương tác với nhau như thế nào.
- **Component View:** Cũng là một góc nhìn về cấu trúc giúp chúng ta hiểu cách phân bổ và sử dụng lại các thành phần trong hệ thống ra sao.
- **Deployment View:** cung cấp góc nhìn về triển khai hệ thống, nó cũng ảnh hưởng lớn đến kiến trúc hệ thống.

Tập hợp các góc nhìn này sẽ giúp chúng ta hiểu rõ hệ thống cần phân tích, thiết kế. Trong hình 1 chúng ta thấy góc nhìn Use Case View nằm ở giữa và chi phối tất cả các góc nhìn còn lại. Chính vì thế chúng ta thường thấy các tài liệu nói về 4 view + 1 chứ không phải 5 view nhằm nhấn mạnh vai trò của Use Case View.

4.2 Diagram (Bản vẽ)

Diagram các bạn có thể dịch là sơ đồ. Tuy nhiên ở đây chúng ta sử dụng từ bản vẽ cho dễ hình dung. Các bản vẽ được dùng để thể hiện các góc nhìn của hệ thống.



Hình 2.3. Các bản vẽ trong OOAD sử dụng UML

Trong đó,

- **Use Case Diagram:** bản vẽ mô tả về ca sử dụng của hệ thống. Bản vẽ này sẽ giúp chúng ta biết được ai sử dụng hệ thống, hệ thống có những chức năng gì. Lập được bản vẽ này bạn sẽ hiểu được yêu cầu của hệ thống cần xây dựng.
- **Class Diagram:** bản vẽ này mô tả cấu trúc của hệ thống, tức hệ thống được cấu tạo từ những thành phần nào. Nó mô tả khía cạnh tĩnh của hệ thống.
- **Object Diagram:** Tương tự như Class Diagram nhưng nó mô tả đến đối tượng thay vì lớp (Class).
- **Sequence Diagram:** là bản vẽ mô tả sự tương tác của các đối tượng trong hệ thống với nhau được mô tả tuần tự các bước tương tác theo thời gian.
- **Collaboration Diagram:** tương tự như sequence Diagram nhưng nhấn mạnh về sự tương tác thay vì tuần tự theo thời gian.
- **State Diagram:** bản vẽ mô tả sự thay đổi trạng thái của một đối tượng. Nó được dùng để theo dõi các đối tượng có trạng thái thay đổi nhiều trong hệ thống.
- **Activity Diagram:** bản vẽ mô tả các hoạt động của đối tượng, thường được sử dụng để hiểu về nghiệp vụ của hệ thống.
- **Component Diagram:** bản vẽ mô tả về việc bố trí các thành phần của hệ thống cũng như việc sử dụng các thành phần đó.
- **Deployment Diagram:** bản vẽ mô tả việc triển khai của hệ thống như việc kết nối, cài đặt, hiệu năng của hệ thống v.v...

Chúng ta sẽ bàn kỹ các bản vẽ này trong các bài tiếp theo. Vì nó chính là hạt nhân của loạt bài này.

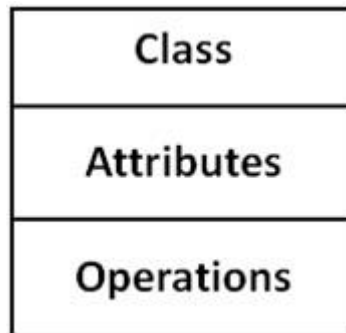
Lưu ý: Ở đây chúng ta sử dụng từ hệ thống tương đương với sản phẩm phần mềm.

4.3 Notations (các ký hiệu)

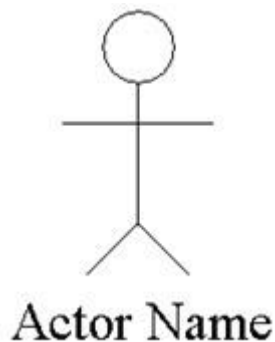
Notations là các ký hiệu để vẽ, nó như từ vựng trong ngôn ngữ tự nhiên. Bạn phải biết từ vựng thì mới ghép thành câu, thành bài được. Chúng ta sẽ tìm hiểu kỹ các notations trong từng bản vẽ sau này. Dưới đây là vài ví dụ về notation.



Hình 2.4. Ký hiệu về Use Case



Hình 2.5. Ký hiệu về Class



Hình 2.6. Ký hiệu về Actor

Và còn rất nhiều ký hiệu nữa.

4.4 Mechanisms (Rules)

Mechanisms là các qui tắc để lập nên bản vẽ, mỗi bản vẽ có qui tắc riêng và bạn phải nắm được để tạo nên các bản vẽ thiết kế đúng. Các qui tắc này chúng ta sẽ bàn kỹ trong các bài về các bản vẽ.

2.1. Tổng quan về ASP.NET MVC

ASP.NET MVC là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.

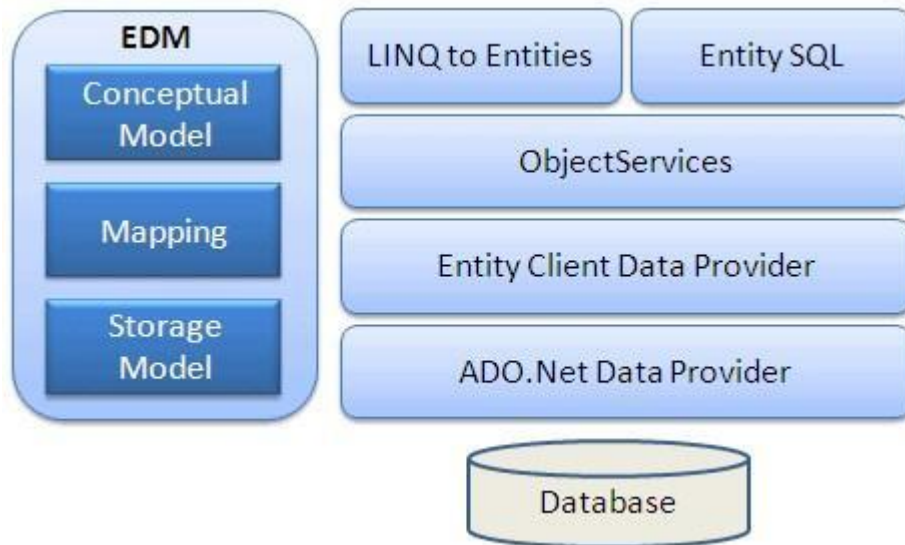
- Model: giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
- View: được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.
- Controller: chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Framework này khá là nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.

2.1. Công nghệ thao tác dữ liệu Entity framework (hoặc ADO.NET)

❖ Cấu trúc của Entity framework

Cấu trúc được mô tả đơn giản qua hình sau:



Hình 2.7: Cấu trúc entity framework

❖ EDM (Entity Data Model):

EDM bao gồm 3 phần :

- **Conceptual Model:** Chứa các model class và các quan hệ của nó. Nó độc lập với thiết kế bảng CSDL của bạn.
- **Mapping:** gồm có thông tin về cách **Conceptual model** kết nối với **Storage model**.
- **Storage Model:** Đây là database design model bao gồm các bảng, views, stored procedures, và những quan hệ và các khóa.

❖ LINQ to Entities:

Là ngôn ngữ truy vấn sử dụng để viết các truy vấn tới object model. Trả về các thực thể được định nghĩa bên trong **Conceptual model**.

❖ Entity SQL:

Đây là một ngôn ngữ truy vấn khác, nó giống LINQ to Entities.

❖ Object Service:

Object service sẽ có trách nhiệm trong việc cụ thể hóa quá trình chuyển đổi dữ liệu trả về từ một entity client data provider tới một entity object structure.

❖ Entity Client Data Provider:

Nhiệm vụ chính của nó là chuyển đổi L2E hoặc những truy vấn Entity SQL vào một truy vấn SQL. Nó được hiểu bởi CSDL cơ bản. Giao tiếp với ADO.Net data provider lần lượt gửi và nhận dữ liệu từ CSDL.

❖ ADO.Net Data Provider:

Thực hiện giao tiếp với CSDL bằng sử dụng chuẩn ADO.Net.

2.2. Mô hình thao tác dữ liệu 3 layer

Mô hình 3 lớp hay còn được gọi là mô hình Three Layer (3-Layer), mô hình này ra đời nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và chạy lộn xộn. Mô hình này phát huy hiệu quả nhất khi xây dựng một dự án(project) lớn, việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn.

Mô hình 3-layer gồm có 3 phần chính:

Presentation Layer (GUI)

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

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

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

Data Access Layer (DAL)

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

❖ **Các thành phần của từng lớp**

Presentation Layer (GUI)

Có hai thành phần chính sau đây với những tác vụ cụ thể :

UI Components: gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng... Ví dụ : textbox, button, combobox, ... **UI Process Components:** là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI.

Business Layer (BLL)

Lớp này gồm 4 thành phần:

Service Interface: là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.

Business Workflows: chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

Business Components: chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc . Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

Business Entities: thường được sử dụng như Data Transfer Objects (DTO) . Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu (DataSets, XML,...) hay các lớp đối tượng đã được tùy chỉnh.

Data Layer (DAL)

Data Access Logic Components: chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu (Data Sources) như XML, file system,... Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì. **Service Agents :** giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

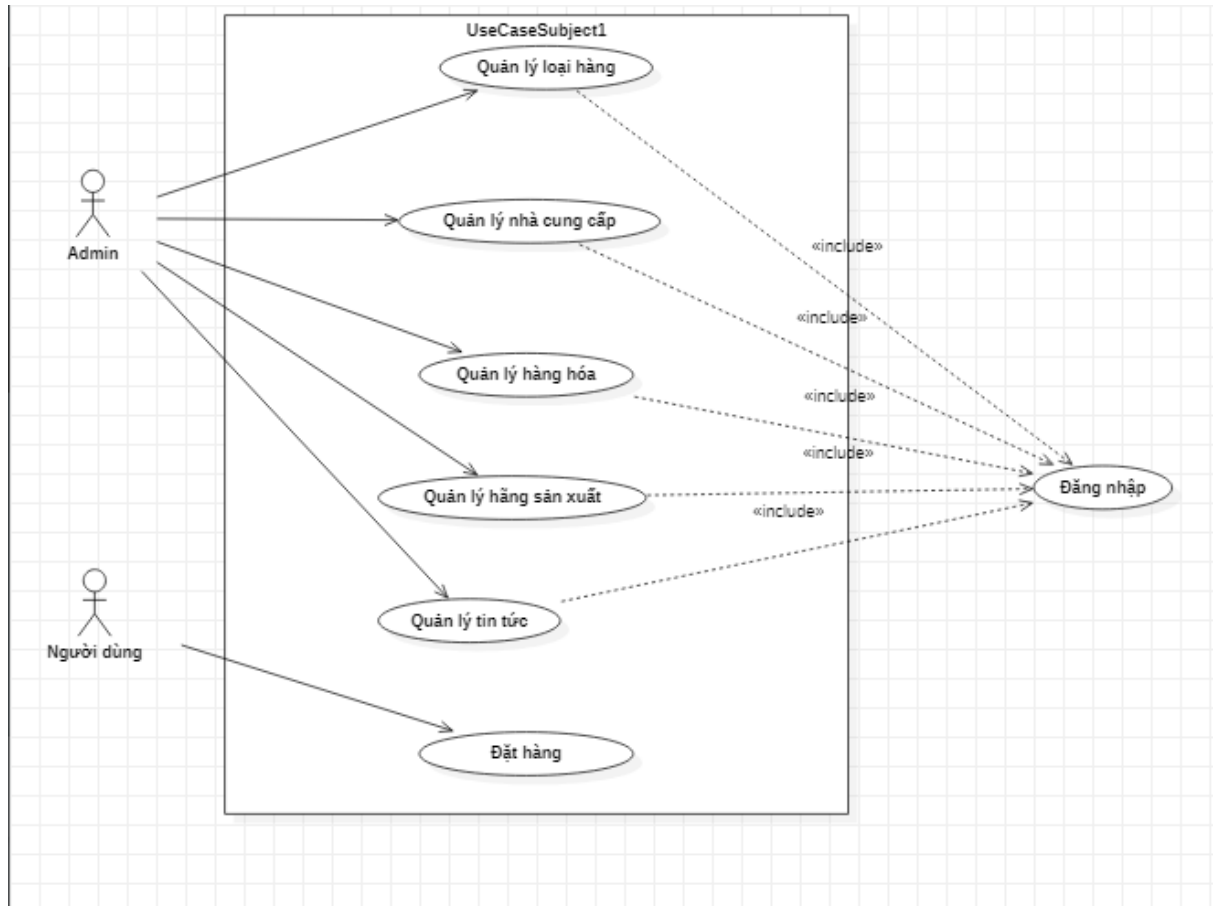
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Đặc tả yêu cầu phần mềm

3.1.1 Các yêu cầu chức năng

a) Chức năng của phân hệ quản trị nội dung (nếu có)

- Biểu đồ use case Tổng quát



Hình 3.1. Biểu đồ usecase Tổng quát

- Biểu đồ use case Quản lý thông tin cá nhân

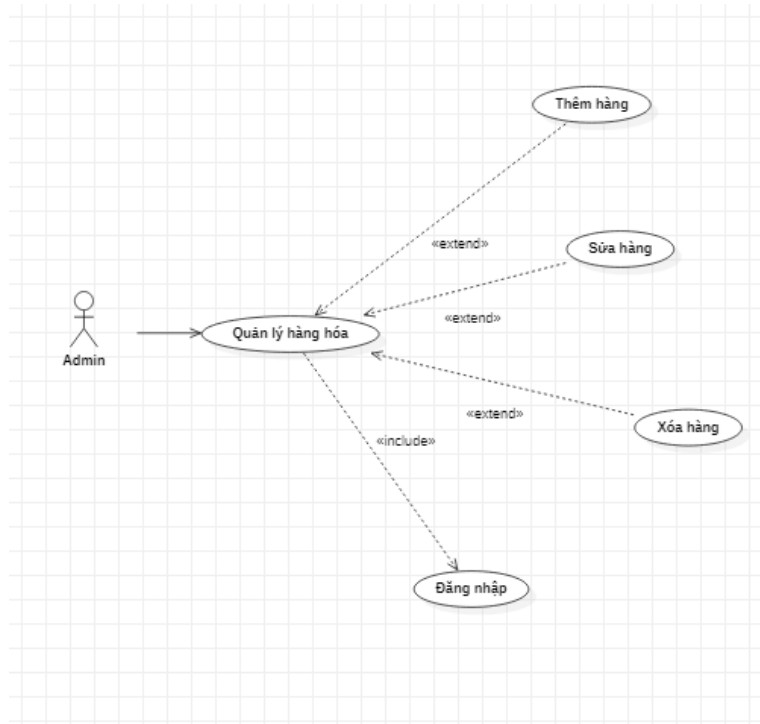
Dòng sự kiện chính

Bảng 3.1 Dòng sự kiện chính đăng nhập

Hành động của tác nhân	Phản ứng của hệ thống
Đăng nhập hệ thống	Hiển thị giao diện đăng nhập
Sau khi đăng nhập thành công	Hiển thị giao diện phần mềm
Chọn chức năng quản lý người dùng từ menu	Hiển thị giao diện quản lý người dùng

Tùy chọn các chức năng của quản lý người dùng

- Biểu đồ use case Quản lý hàng hóa



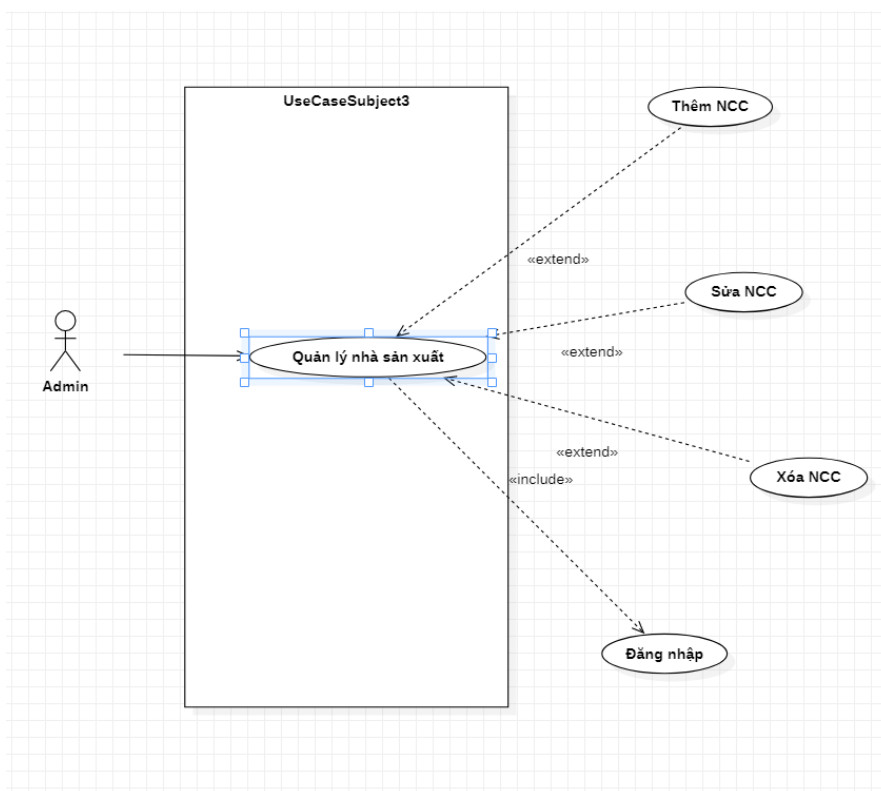
Hình 3.2 Biểu đồ usecase Quản lý hàng hóa

Dòng sự kiện chính

Bảng 3.2. Dòng sự kiện quản lý hàng hóa

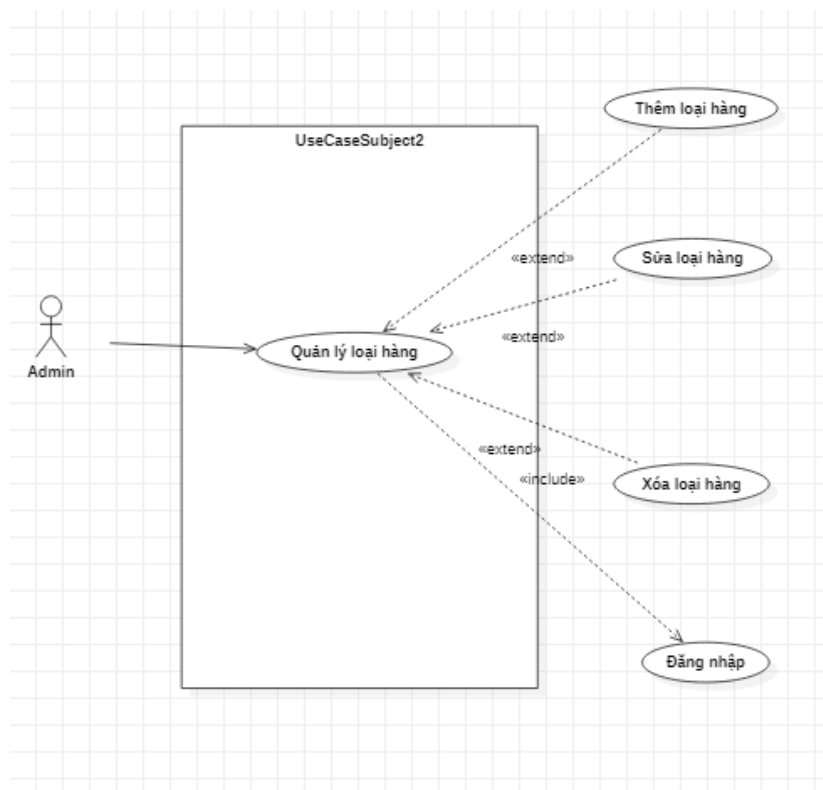
Hành động của tác nhân	Phản ứng của hệ thống
Đăng nhập hệ thống	Hiển thị giao diện đăng nhập
Sau khi đăng nhập thành công	Hiển thị giao diện phần mềm
Chọn chức năng từ màn hình	Hiển thị giao diện quản lý hàng hóa
Tùy chọn các chức năng	

- Biểu đồ use case Quản lý nhà sản xuất



Hình 3.3 Biểu đồ usecase Quản lý nhà sản xuất

Biểu đồ usecase quản lý loại hàng



Hình 3.4. Biểu đồ usecase Quản lý loại hàng

Dòng sự kiện chính

Bảng 3.3. Dòng sự kiện quản lý loại hàng

Hành động của tác nhân	Phản ứng của hệ thống
Đăng nhập hệ thống	Hiển thị giao diện đăng nhập
Sau khi đăng nhập thành công	Hiển thị giao diện phần mềm
Chọn chức năng nhấn tin	Hiển thị giao diện quản lý loại hàng

b) Chức năng của phân hệ người dùng

- Khảo sát bài toán

Bảng 3.4. Khảo sát bài toán

STT	Tên yêu cầu
1	Đăng nhập vào phần mềm
2	Hiện thị chức năng quản lý
3	Cập nhật thông tin sản phẩm

- Phân tích yêu cầu chức năng

Bảng 3.5. Phân tích yêu cầu chức năng

STT	Tên yêu cầu	Mô tả yêu cầu
I	Các yêu cầu chức năng nghiệp vụ	
1	Chức năng đăng nhập	Đăng nhập vào trang quản trị
2	Chức năng quản lý thông tin	Cập nhật thông tin sản phẩm
3	Chức năng tìm kiếm	Cho phép người dùng tìm kiếm sản phẩm
4	Chức năng đặt hàng	Người dùng có thể thêm sản phẩm vào giỏ hàng, tiến hành đặt hàng

II Yêu cầu chức năng hệ thống		
1	Quản lý hệ thống	<ul style="list-style-type: none"> - Quản lý sản phẩm - Quản lý tài khoản

3.1.2 Các yêu cầu phi chức năng

Bảng 3.6. Các yêu cầu phi chức năng

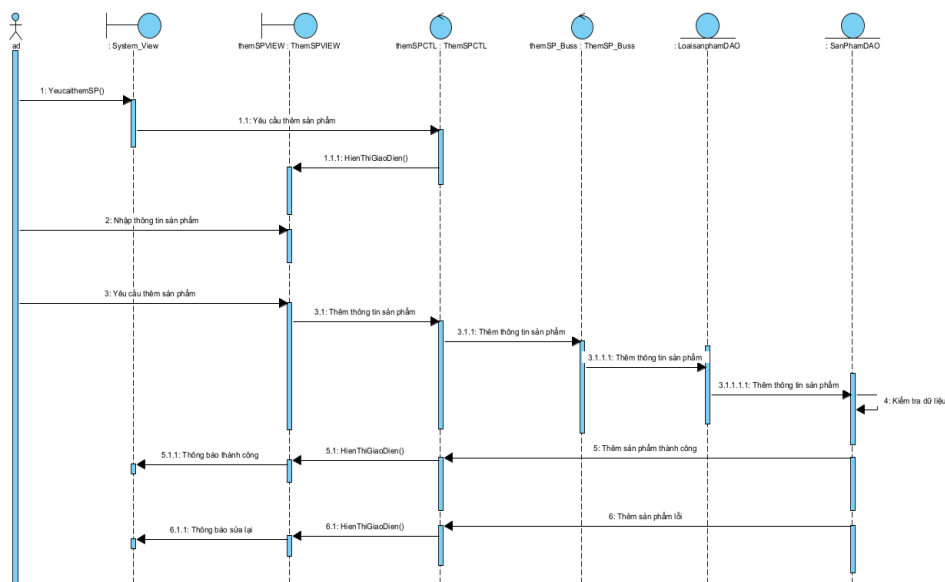
1	Giao diện	Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với người dung
2	Tốc độ xử lý	Tối ưu chương trình nhanh chóng, chính xác
3	Bảo mật dữ liệu	Tính bảo mật thông tin cao

3.2 Thiết kế hệ thống

3.2.1 Thiết kế lớp đối tượng

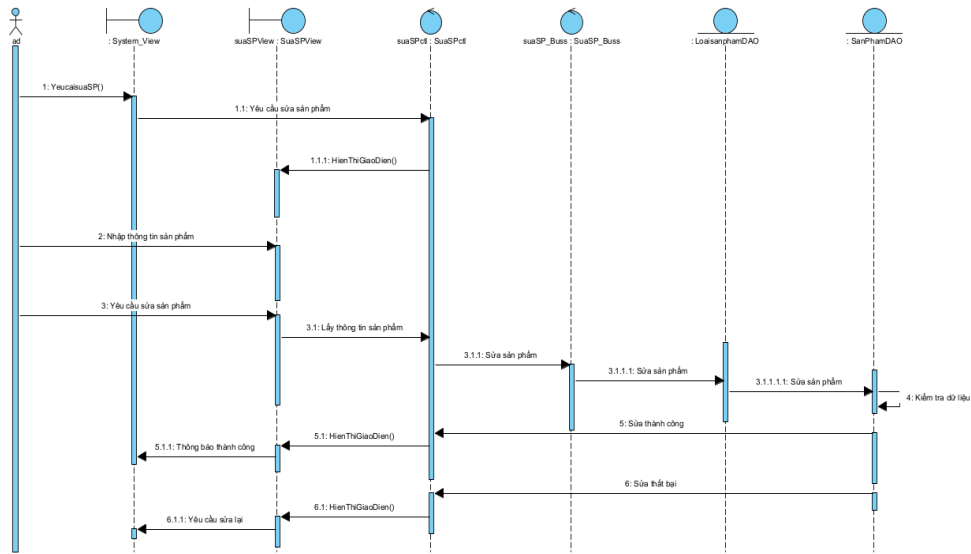
a) Biểu đồ tuần tự

Biểu đồ tuần tự thêm sản phẩm :



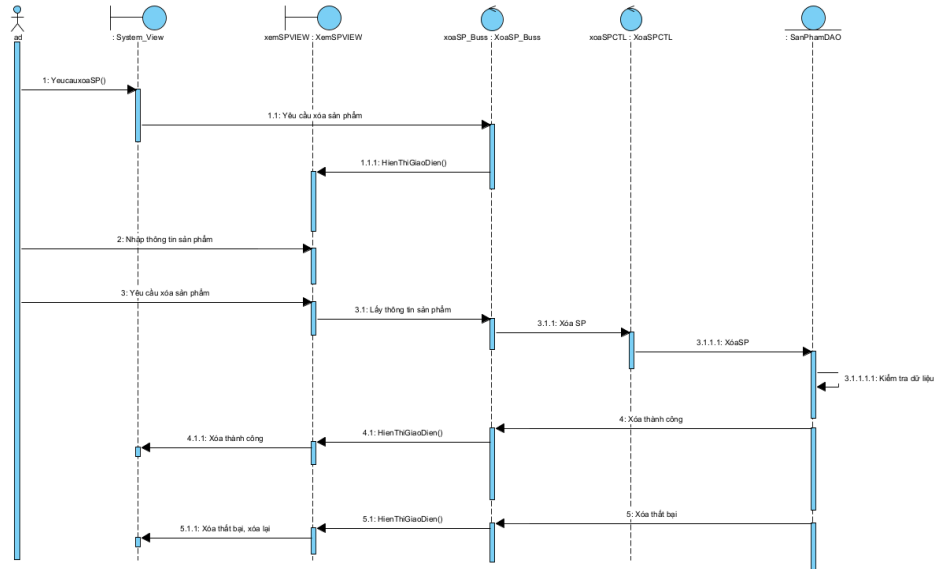
Hình 3.5. Biểu đồ tuần tự thêm sản phẩm

Biểu đồ tuần tự sửa sản phẩm :



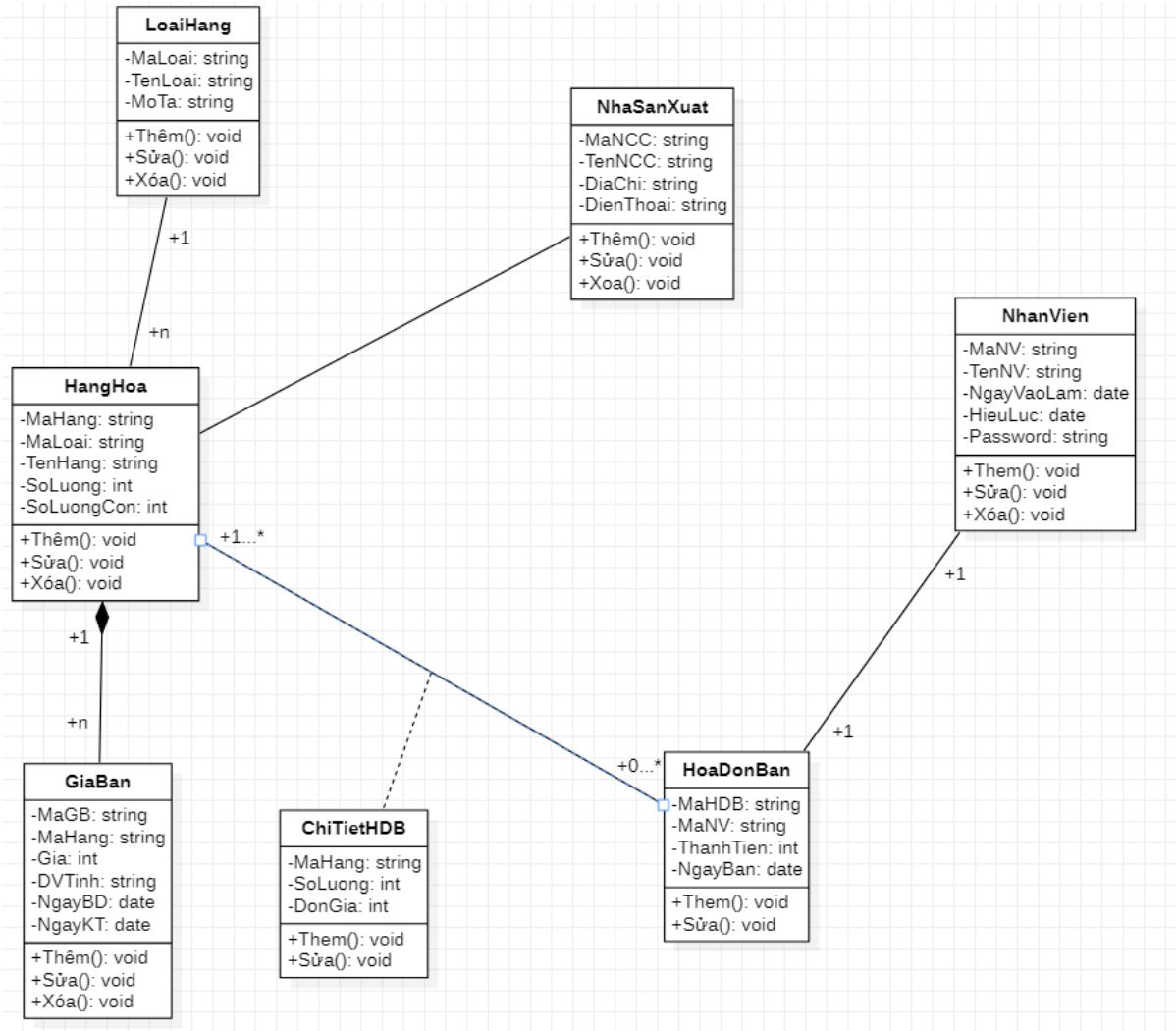
Hình 3.6. Biểu đồ tuần tự sửa sản phẩm

Biểu đồ tuần tự xóa sản phẩm :



Hình 3.7. Biểu đồ tuần tự xóa sản phẩm

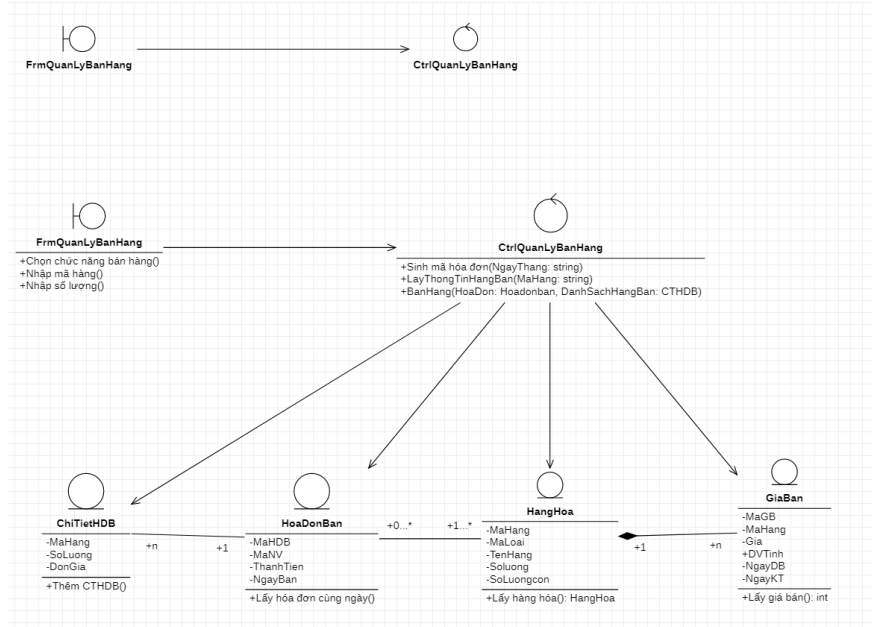
b) Biểu đồ lớp chi tiết



Hình 3.8. Biểu đồ lớp

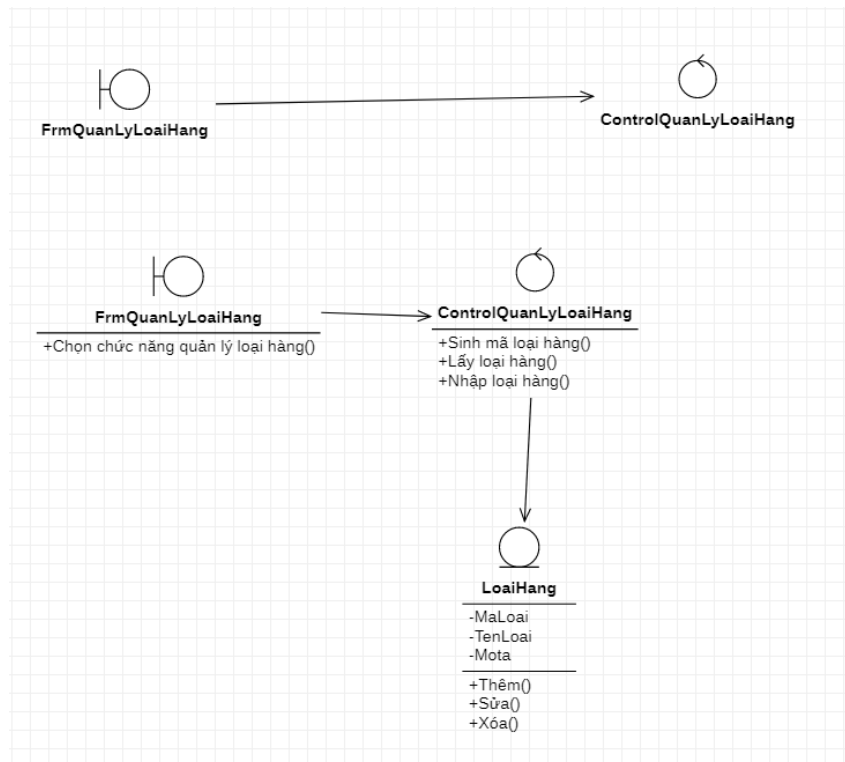
Biểu đồ lớp chi tiết VOPC

Biểu đồ VOPC Quản lý bán hàng



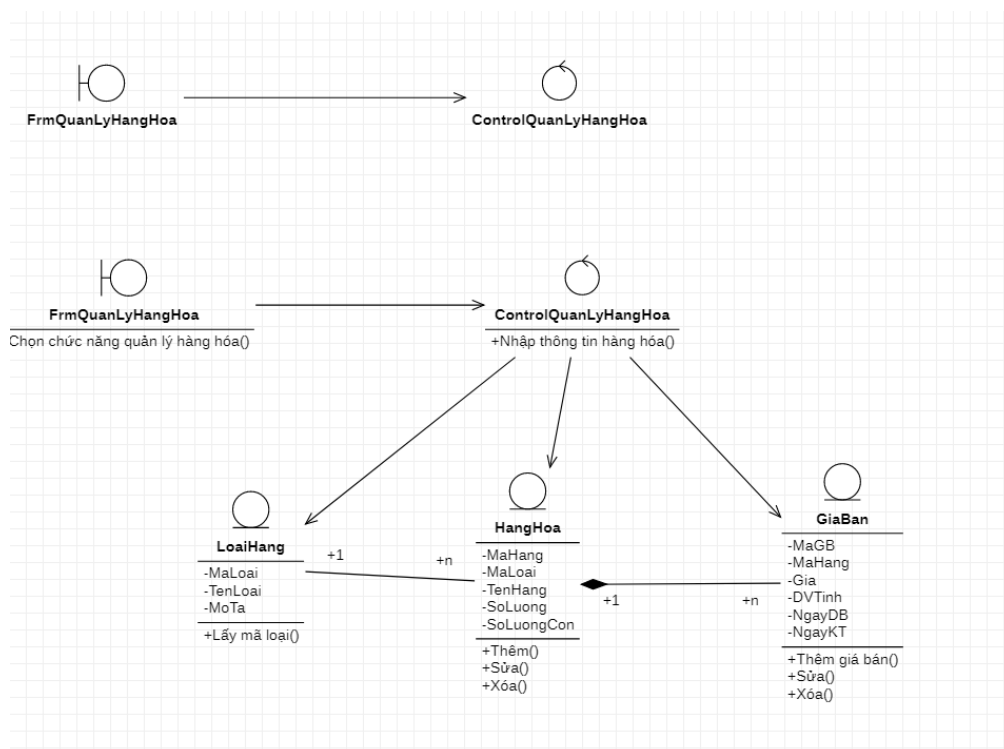
Hình 3.9. Biểu đồ chi tiết quản lý bán hàng

Biểu đồ VOPC Quản lý loại sản phẩm



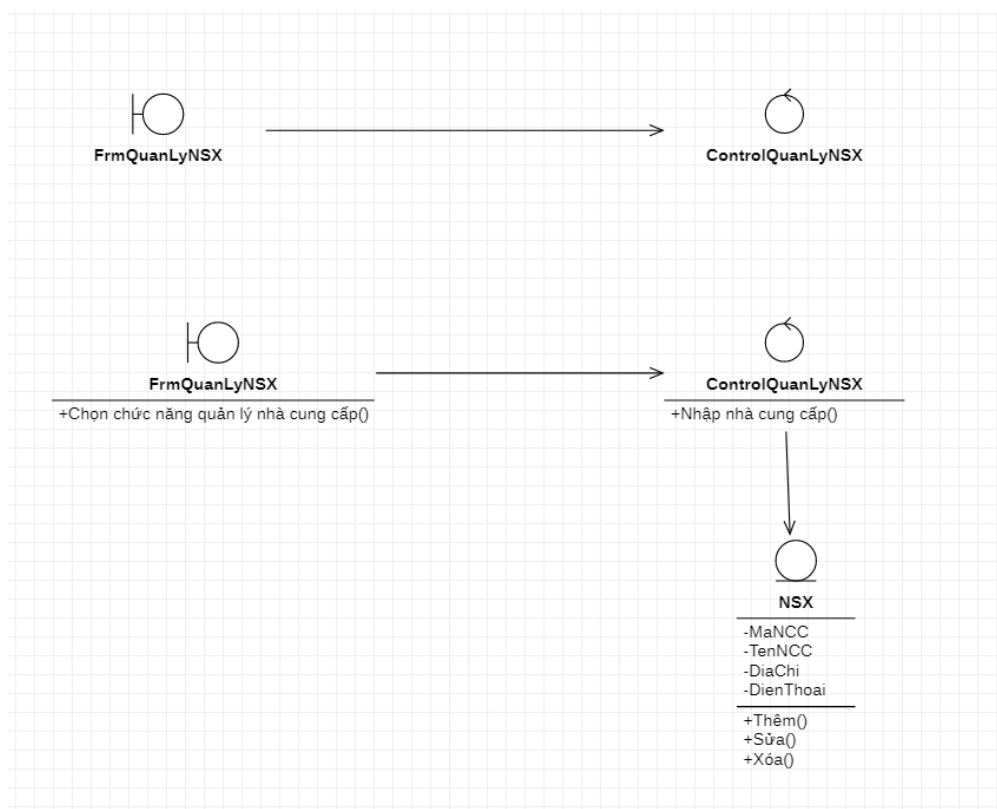
Hình 3.10. Biểu đồ chi tiết quản lý loại sản phẩm

Biểu đồ VOPC quản lý sản phẩm



Hình 3.11. Biểu đồ chi tiết quản lý Sản phẩm

Biểu đồ VOPC Quản lý NSX



Hình 3.12. Biểu đồ Quản lý NSX

- Lớp Account
 - Danh sách thuộc tính

Bảng 3.7. Danh sách thuộc tính Account

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	Username	String	Tên tài khoản
2	Password	String	Mật khẩu

- Danh sách phương thức

Bảng 3.8. Danh sách phương thức Account

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Add	Void	Public	Tạo tài khoản
2	Login	Void	Public	Đăng nhập vào hệ thống

- Lớp News
 - Danh sách thuộc tính

Bảng 3.9. Danh sách thuộc tính News

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	Int	Mã tin tức
2	Title	String	Tiêu đề
3	Image	String	Hình ảnh
4	Content	String	Nội dung
5	Comment	List	Danh sách Bình luận

- Danh sách phương thức

Bảng 3.10. Danh sách phương thức News

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Create	Void	Public	Thêm tin tức
2	Delete	Void	Public	Xóa tin tức

3	Update	Void	Public	Cập nhật
4	Commnt	Void	Public	Bình luận tin tức

- Lớp Category
 - Danh sách thuộc tính

Bảng 3.11. Danh sách thuộc tính Category

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	Int	Mã danh mục
2	Name	String	Tên danh mục
3	MetaName	String	Thẻ danh mục

- Danh sách phương thức

Bảng 3.12. Danh sách phương thức ContentChat

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Create	Void	Public	Thêm danh mục
2	Update	Void	Public	Cập nhật danh mục
3	Delete	Void	Public	Xóa danh mục

- Lớp Brand
 - Danh sách thuộc tính

Bảng 3.13. Danh sách thuộc tính Brand

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	String	Mã hãng
2	Name	String	Tên hãng
3	MetaName	String	Tên thẻ

- Danh sách phương thức

Bảng 3.14. Danh sách phương thức Brand

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Add	Void	Public	Thêm

2	Update	Void	Public	Cập nhật
3	Delete	Void	Public	Xóa

- Lớp Product
 - Danh sách thuộc tính

Bảng 3.15. Danh sách thuộc tính Product

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	Int	
2	Name	String	Tên sản phẩm
3	MetaName	String	Tên thẻ
4	Cost	Int	Giá bán
5	Image	String	Hình ảnh sản phẩm
6	ImageProduct	List	Danh sách hình ảnh
7	Description	String	Mô tả sản phẩm
8	Detail	String	Chi tiết sản phẩm
9	HotDeal	Boolean	Giảm giá
10	TopSeller	Boolean	Bán chạy
11	OnTop	Boolean	Đầu trang
12	IsNew	Boolean	Sản phẩm mới
13	IsStatus	Boolean	Tình trạng
14	Category	String	Loại sản phẩm
15	Brand	String	Hãng sản xuất
16	Review	List	Các đánh giá sản phẩm

- Danh sách phương thức

Bảng 3.16. Danh sách phương thức Product

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Add	Void	Public	Thêm
2	Update	Void	Public	Cập nhật
3	Delete	Void	Public	Xóa
4	Search	Void	Public	Tìm kiếm

5	Comment	Void	Public	Bình luận sản phẩm
---	---------	------	--------	--------------------

- Lớp Cart
 - Danh sách thuộc tính

Bảng 3.17. Danh sách thuộc tính Cart

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	ID	String	Mã giỏ hàng
2	Product	Int	Mã sản phẩm
3	Quantity	Int	Số lượng

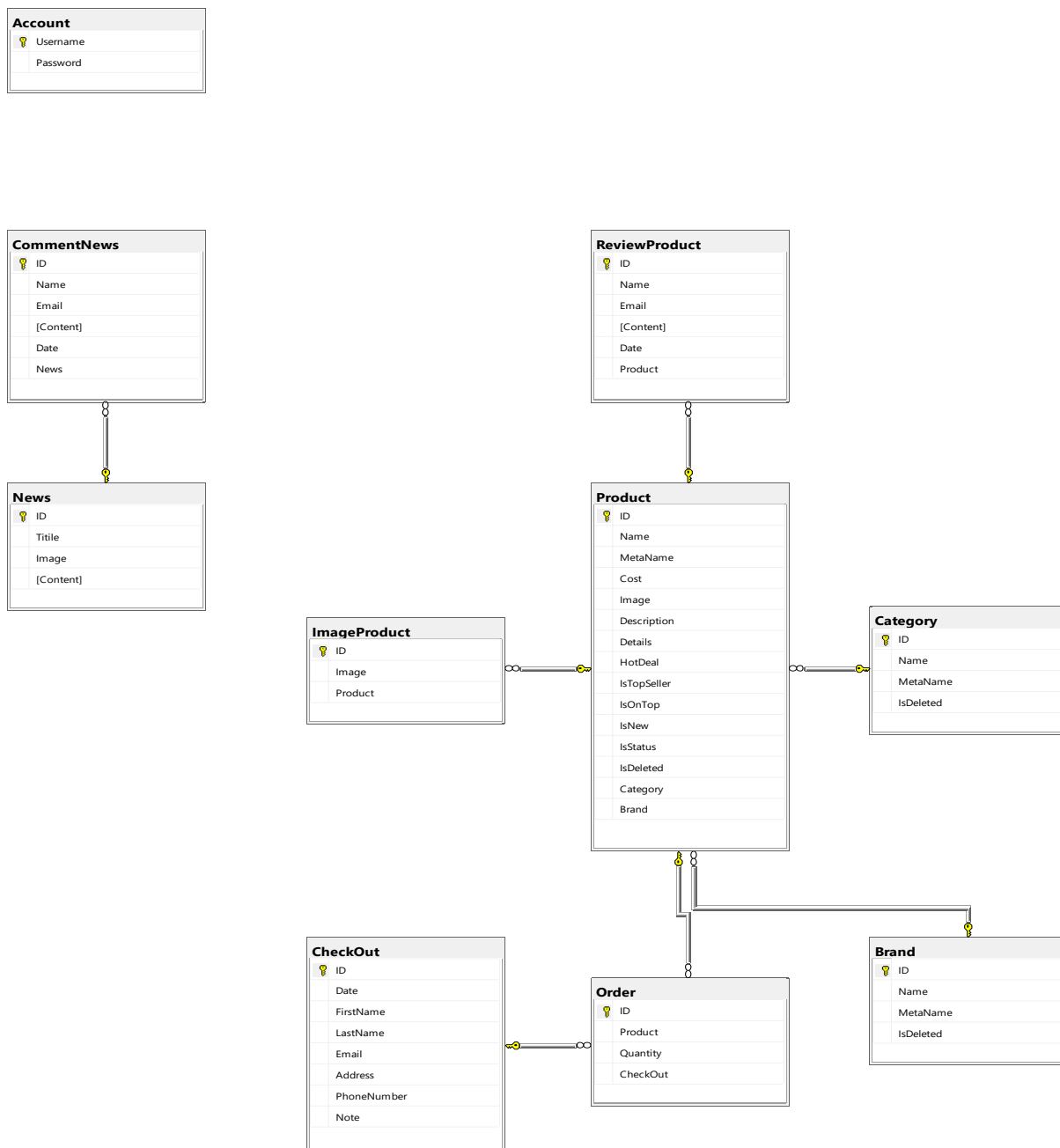
- Danh sách phương thức

Bảng 3.18. Danh sách phương thức WatchVideo

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	Add	Void	Public	Thêm giỏ hàng
2	Delete	Void	Public	Xóa khỏi giỏ hàng
3	Update	Void	Public	Cập nhật số lượng

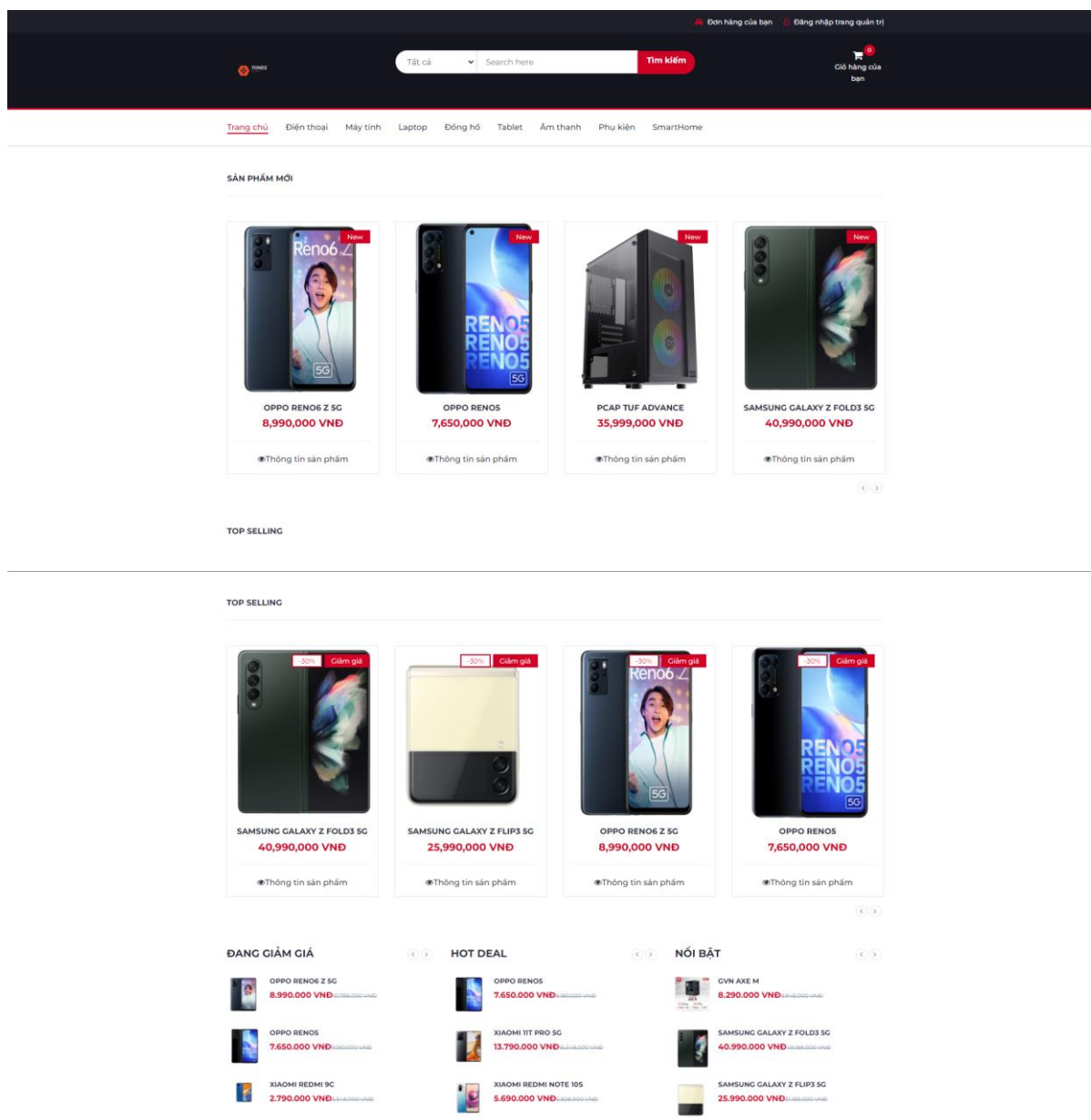
3.2.2 Thiết kế cơ sở dữ liệu

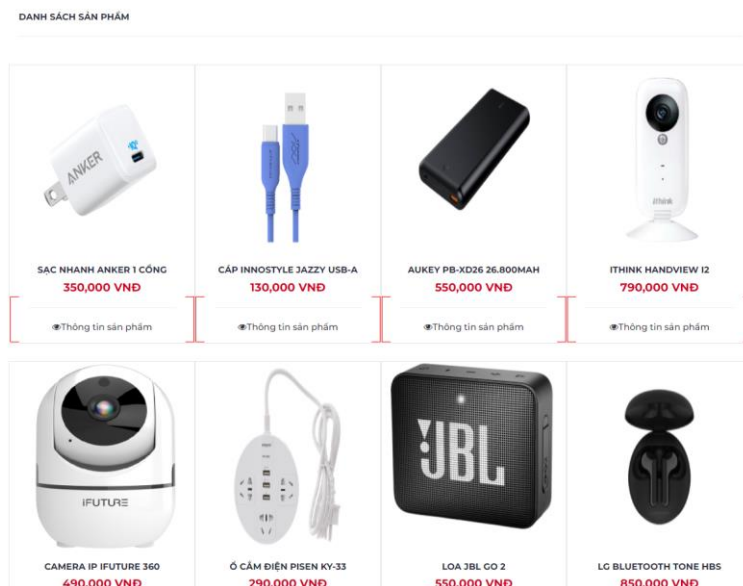
Biểu đồ Cơ sở dữ liệu



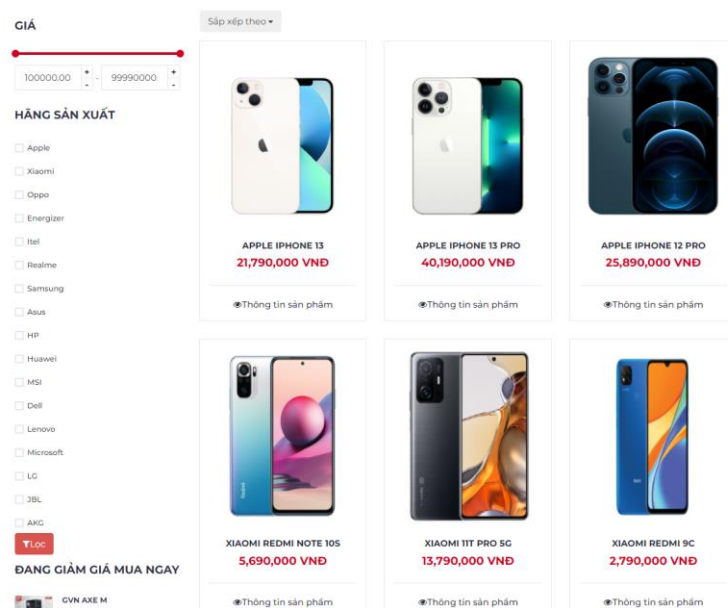
Hình 3.13. Biểu đồ Cơ sở dữ liệu

3.2.3 Thiết kế giao diện

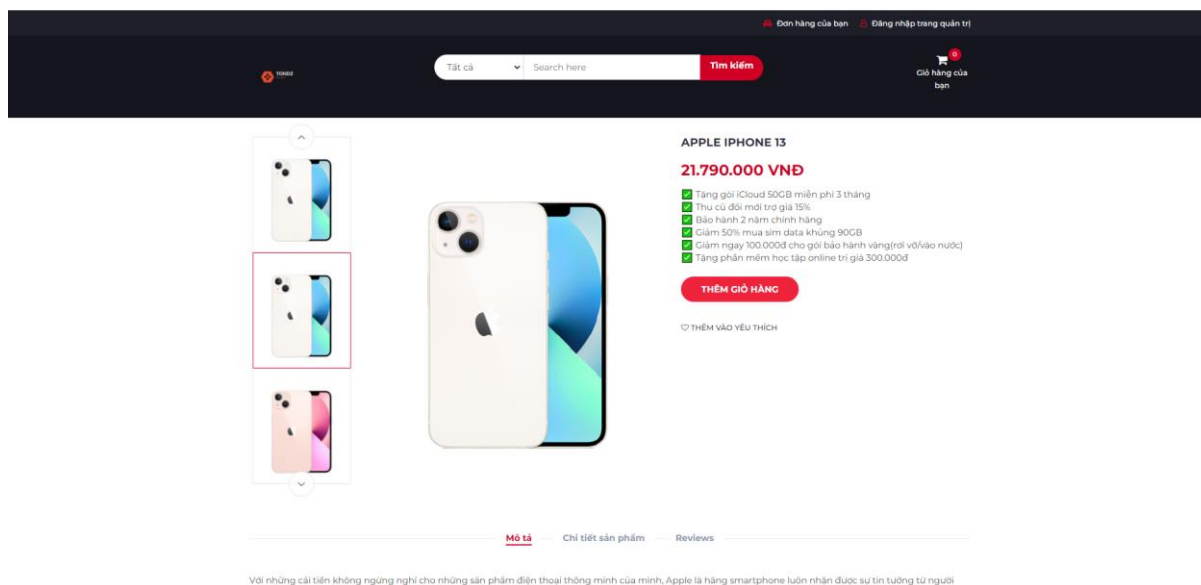




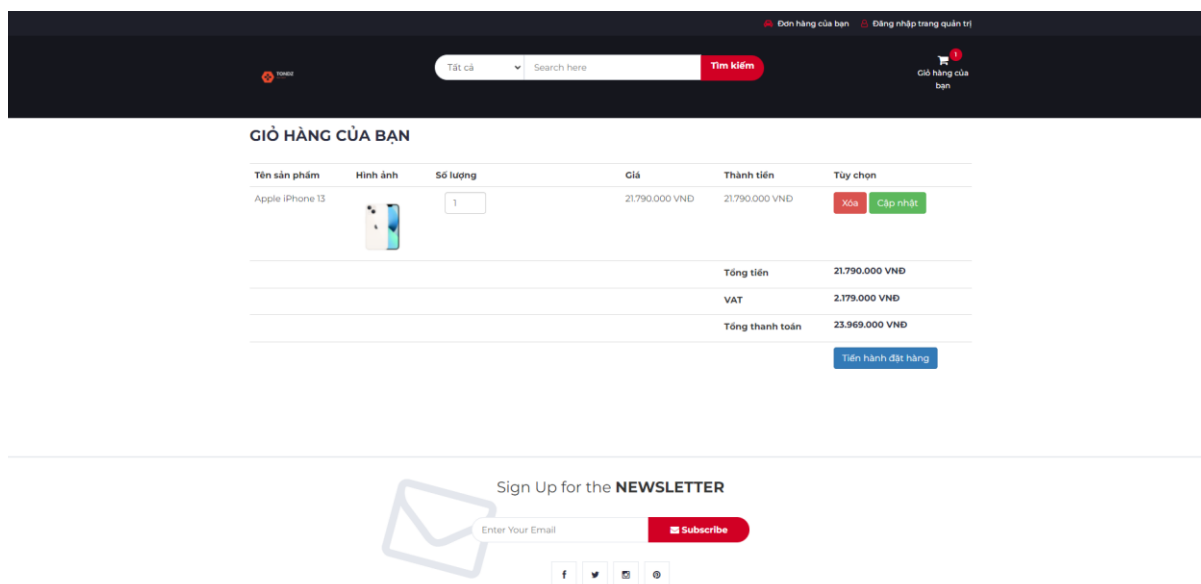
Hình 3.16. Giao diện trang chủ



Hình 3.17. Giao diện sản phẩm theo danh mục



Hình 3.14. Giao diện chi tiết sản phẩm



Hình 3.15. Giao diện giỏ hàng

Đơn hàng của bạn

Đăng nhập trang quản trị

Tất cảSearch hereTìm kiếmGiỏ hàng của bạn

ĐẶT HÀNGHOME / CHECKOUT

ĐỊA CHỈ THANH TOÁN

Họ

Tên

Email

Địa chỉ giao hàng

Số điện thoại

Ghi chú

ĐƠN HÀNG CỦA BẠN

Sản phẩm	Thành tiền
1x Apple iPhone 13	21.790.000 VND
VAT(10%)	2.179.000 VND
Shipping	FREE
Tổng	23.969.000VND

ĐẶT HÀNG

Sign Up for the **NEWSLETTER**

Hình 3.18. Giao diện đặt hàng


Trạng thái đơn hàng

Xác nhận đơn hàng

Chuẩn bị đơn hàng

Đang giao hàng

Nhận hàng

 Apple iPhone 13
x2
43.580.000

THÔNG TIN

Mã đơn hàng:

1000

Họ:

Đỗ

Tên:

Tôn

Email liên hệ:

thanhtonvk@gmail.com

Địa chỉ giao hàng

Hưng Yên

SĐT liên hệ:

094542134

Ghi chú:

giao trước 8h

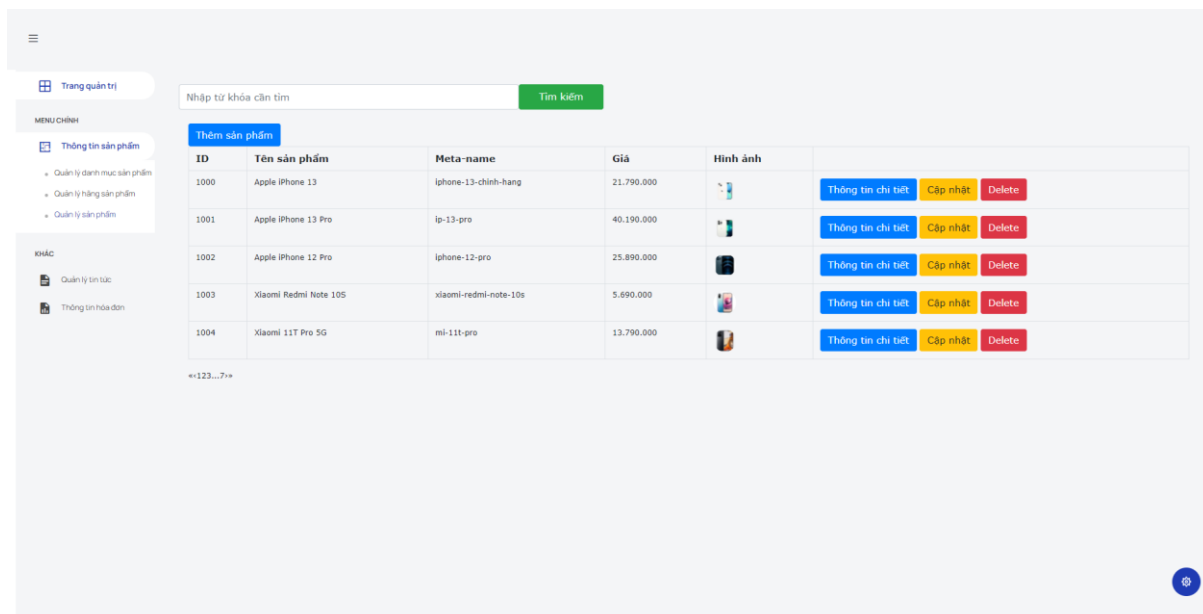
ĐƠN HÀNG CỦA BẠN

Sản phẩm	Thành tiền
2x Apple iPhone 13	21.790.000 VND
VAT(10%)	4.358.000 VND
Shipping	FREE
Tổng	47.938.000VND

TIẾP TỤC MUA SẮM

Hình 3.19. Giao diện tình trạng đơn hàng

Bài tập lớn MVC: Xây dựng trang web bán đồ công nghệ



Hình 3.20. Giao diện trang quản trị

CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG

4.1 Triển khai các chức năng của hệ thống

4.1.1 Chức năng hiển thị danh sách sản phẩm

a) Thiết kế layout

- Xây dựng layout bằng HTML

```
<!DOCTYPE html>

<html>
<head>
  <title>title</title>
  <!-- Google font -->
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,500,700"
rel="stylesheet">

  <!-- Bootstrap -->
  <link type="text/css" rel="stylesheet" href="~/Content/Home/css/bootstrap.min.css"/>

  <!-- Slick -->
  <link type="text/css" rel="stylesheet" href="~/Content/Home/css/slick.css"/>
  <link type="text/css" rel="stylesheet" href="~/Content/Home/css/slick-theme.css"/>

  <!-- nouislider -->
  <link type="text/css" rel="stylesheet" href="~/Content/Home/css/nouislider.min.css"/>

  <!-- Font Awesome Icon -->
  <link rel="stylesheet" href="~/Content/Home/css/font-awesome.min.css">

  <!-- Custom stylesheet -->
  <link type="text/css" rel="stylesheet" href="~/Content/Home/css/style.css"/>

  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <script src="~/Scripts/angular.js"></script>
  <script src="~/Scripts/dirPagination.js"></script>
  <script src="~/App/Home/app.js"></script>
  <script src="~/Scripts/jquery-3.3.1.js"></script>
</head>
<body ng-app="Home">
<div ng-controller="HomeController" ng-init="GetAllProduct()">
  <!-- SECTION -->
  <div class="section">
    <!-- container -->
    <div class="container">
      <!-- row -->
      <div class="row">

        <!-- section title -->
        <div class="col-md-12">
          <div class="section-title">
            <h5 class="title">Danh sách sản phẩm</h5>
            <hr/>
          </div>
        </div>
      </div>
      <!-- /section title -->

      <!-- Products tab & slick -->
      <div class="col-md-12">
```

```

<div class="row">
  <!-- product -->
  <div class="product col-md-3 col-lg-3 col-sm-3" dir-paginate="item
in ProductList|itemsPerPage:8">
    <div class="product-img">
      
    </div>
    <div class="product-body">
      <h3 class="product-name">
        <a href="#">{{item.Name}}</a>
      </h3>
      <h4 class="product-price">{{item.Cost.toLocaleString(vi-
VN)}} VNĐ</h4>
      <hr/>
      <div class="product-btns" >
        <a class="quick-view" ng-click="GetProduct(item.ID)"
href="@Url.Action("DetailsProduct", "Home")">
          <i class="fa fa-eye"></i><span
class="tooltip">Thông tin sản phẩm</span>
        </a>
      </div>
      </div>
      <div class="add-to-cart" style="background-color: white;border-
style: ridge;border-color: red;border-width: 1px">
        <button class="add-to-cart-btn" ng-
click="AddCart(item.ID)"><i class="fa fa-shopping-cart"></i> Thêm vào giỏ hàng</button>
      </div>
    </div>
  <!-- /product -->
</div>
<ul>
  <dir-pagination-controls
direction-links="true"
boundary-links="true"
on-page-change="pageChanged(newPageNumber)">
  </dir-pagination-controls>
</ul>
</div>
<!-- /Products tab & slick -->
</div>
<!-- /row -->
</div>
<!-- /container -->
</div>
<!-- /SECTION -->
</div>
</body>
</html>

```

b) Code JavaScript

```

$scope.GetAllProduct = function () {
  $http({
    method: "get",
    url: "/Home/GetAllProduct/",
  }).then(function (response) {
    $scope.ProductList = response.data;
  })
}

```

c) Code C#

```
public JsonResult GetAllProduct()
{
    db.Configuration.ProxyCreationEnabled = false;
    var model = db.Products.Where(x => x.IsDeleted == false && x.IsStatus ==
true).ToList();
    model.Reverse();
    return Json(model, JsonRequestBehavior.AllowGet);
}
```

4.1.2 Chức năng Xem chi tiết sản phẩm

a) Thiết kế layout

```
<div class="section">
<!-- container -->
<div class="container">
<!-- row -->
<div class="row">
<!-- Product main img -->
<div class="col-md-5 col-md-push-2">
    <div id="product-main-img">
        <div class="product-preview">
            
        </div>

        <div class="product-preview">
            
        </div>
        <div class="product-preview">
            
        </div>
        <div class="product-preview">
            
        </div>
    </div>
</div>
<!-- /Product main img -->

<!-- Product thumb imgs -->
<div class="col-md-2 col-md-pull-5">
    <div id="product-imgs">
        <div class="product-preview">
            
        </div>

        <div class="product-preview">
            
        </div>
        <div class="product-preview">
            
        </div>
        <div class="product-preview">
            
        </div>
    </div>
</div>
<!-- /Product thumb imgs -->

<!-- Product details -->
<div class="col-md-5">
    <div class="product-details">
```

```
<h2 class="product-name">{{_Name}}</h2>
<div>
  <h3 class="product-price">{{_Cost.toLocaleString('vi-VN')}} VNĐ</h3>
</div>
<ul>
  <li>
    <div>
      <span>✔ Tặng gói iCloud 50GB miễn phí 3 tháng </span>
    </div>
  </li>
  <li>
    <div>
      <span>✔ Thu cũ đổi mới trợ giá 15% </span>
    </div>
  </li>
  <li>
    <div>
      <span>✔ Bảo hành 2 năm chính hãng </span>
    </div>
  </li>
  <li>
    <div>
      <span>✔ Giảm 50% mua sim data khủng 90GB </span>
    </div>
  </li>
  <li>
    <div>
      <span>✔ Giảm ngay 100.000đ cho gói bảo hành vàng(rơi vỡ/vào nước)
    </span>
    </div>
  </li>
  <li>
    <div>
      <span>✔ Tặng phần mềm học tập online trị giá 300.000đ </span>
    </div>
  </li>
</ul>
<br/>
<div class="add-to-cart">
  <button ng-click="AddCart(_ID)" class="add-to-cart-btn"><i class="fa fa-
shopping-cart"></i> Thêm giỏ hàng</button>
</div>

<ul class="product-btns">
  <li>
    <a href="#"><i class="fa fa-heart-o"></i> Thêm vào yêu thích</a>
  </li>
</ul>

</div>
</div>
<!-- /Product details -->

<!-- Product tab -->
<div class="col-md-12">
  <div id="product-tab">
    <!-- product tab nav -->
    <ul class="tab-nav">
```

```
<li class="active">
  <a data-toggle="tab" href="#tab1">Mô tả</a>
</li>
<li>
  <a data-toggle="tab" href="#tab2">Chi tiết sản phẩm</a>
</li>
<li>
  <a data-toggle="tab" href="#tab3">Reviews</a>
</li>
</ul>
<!-- /product tab nav -->

<!-- product tab content -->
<div class="tab-content">
  <!-- tab1 -->
  <div id="tab1" class="tab-pane fade in active">
    <div class="row">
      <div class="col-md-12">
        <p id="txtDescription"></p>
      </div>
    </div>
  </div>
  <!-- /tab1 -->

  <!-- tab2 -->
  <div id="tab2" class="tab-pane fade in">
    <div class="row">
      <div class="col-md-12">
        <p id="txtDetails"></p>
      </div>
    </div>
  </div>
  <!-- /tab2 -->

  <!-- tab3 -->
  <div id="tab3" class="tab-pane fade in">
    <div class="row">
      <div class="col-md-3">
        <div id="rating">
          <div class="rating-avg">
            <span>4.5</span>
            <div class="rating-stars">
              <i class="fa fa-star"></i>
              <i class="fa fa-star"></i>
              <i class="fa fa-star"></i>
              <i class="fa fa-star"></i>
              <i class="fa fa-star-o"></i>
            </div>
          </div>
          <ul class="rating">
            <li>
              <div class="rating-stars">
                <i class="fa fa-star"></i>
                <i class="fa fa-star"></i>
                <i class="fa fa-star"></i>
                <i class="fa fa-star"></i>
                <i class="fa fa-star"></i>
              </div>
              <div class="rating-progress">
                <div style="width: 80%;"></div>
              </div>
              <span class="sum">3</span>
            </li>
            <li>
              <div class="rating-stars">
```

```

        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star-o"></i>
    </div>
    <div class="rating-progress">
        <div style="width: 60%;"></div>
    </div>
    <span class="sum">2</span>
</li>
<li>
    <div class="rating-stars">
        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
    </div>
    <div class="rating-progress">
        <div></div>
    </div>
    <span class="sum">0</span>
</li>
<li>
    <div class="rating-stars">
        <i class="fa fa-star"></i>
        <i class="fa fa-star"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
    </div>
    <div class="rating-progress">
        <div></div>
    </div>
    <span class="sum">0</span>
</li>
<li>
    <div class="rating-stars">
        <i class="fa fa-star"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
        <i class="fa fa-star-o"></i>
    </div>
    <div class="rating-progress">
        <div></div>
    </div>
    <span class="sum">0</span>
</li>
</ul>
</div>
</div>
<!-- /Rating -->
<!-- Reviews -->
<div class="col-md-6" ng-init="GetReviewProductList()">
    <div id="reviews">
        <ul class="reviews">
            <li dir-paginate="item in
ReviewProductList|itemsPerPage:3">
                <div class="review-heading">
                    <h5 class="name">{{item.Name}}</h5>
                    <p class="date">{{item.Date}}</p>
                </div>
                <div class="review-body">

```



```

        <p>{{item.Content}}</p>
    </div>
</li>
</ul>
<ul>
    <dir-pagination-controls
        direction-links="true"
        boundary-links="true"
        on-page-change="pageChanged(newPageNumber)">
    </dir-pagination-controls>
</ul>
</div>
</div>
<!-- /Reviews -->
<!-- Review Form -->
<div class="col-md-3">
    <div id="review-form">
        <form class="review-form">
            <input class="input" type="text" placeholder="Tên của bạn"
ng-model="_NamePerson">
            <input class="input" type="text" placeholder="Email" ng-
model="_EmailPerson">
            <textarea class="input" placeholder="Nội dung" ng-
model="_ContentPerson"></textarea>
            <input type="button" id="btnSave" class="btn btn-danger"
value="Bình luận" ng-click="InsertReviewProduct()"/>
        </form>
    </div>
</div>
<!-- /Review Form -->
</div>
<!-- /tab3 -->
</div>
<!-- /product tab content -->
</div>
</div>
```

b) Code JavaScript

```
$scope.LoadProduct = function () {
    var id = localStorage.getItem("ID");
    $http({
        method: "post",
        url: "/Home/GetProduct/?id=" + id,
    }).then(function (response) {
        $scope.Product = response.data;
        document.getElementById("txtDetails").innerHTML = $scope.Product.Details;
        document.getElementById("txtDescription").innerHTML = $scope.Product.Description;
        localStorage.setItem("Category", $scope.Product.Category);
    })
    $http({
        method: "post",
        url: "/Home/GetImageProduct/?id=" + id,
    }).then(function (response) {
        $scope.ImageProductList = response.data;
    })
    $http({
        method: "post",
        url: "/Home/GetRelatedProduct/?category=" + localStorage.getItem("Category"),
    }).then(function (response) {
        $scope.relatedProduct = response.data;
    })
})
```

}

c) Code C#

```
//Lấy sản phẩm bằng ID
public JsonResult GetProduct(int? id)
{
    db.Configuration.ProxyCreationEnabled = false;
    Product Product = db.Products.Find(id);
    return Json(Product, JsonRequestBehavior.AllowGet);
}

//Lấy ds hình ảnh sản phẩm
public JsonResult GetImageProduct(int? id)
{
    db.Configuration.ProxyCreationEnabled = false;
    var model = db.ImageProducts.Where(x => x.Product == id).ToList();
    return Json(model, JsonRequestBehavior.AllowGet);
}
```

4.2 Kiểm thử và triển khai ứng dụng

4.2.1 Kiểm thử

Bảng 4.1. Kiểm thử cho chức năng đăng ký tài khoản

STT	Tên testcase	Mục đích	Thực hiện	Kết quả
1	Đăng ký tài khoản	Kiểm tra chức năng tạo tài khoản	Nhập thông tin tài khoản: username, fullname,... Ấn nút Create new account	Thông báo tạo tài khoản thành công
2	Nhập tài khoản đã tồn tại	Kiểm tra tài khoản đã tồn tại	Nhập thông tin tài khoản đã tồn tại	Thông báo tài khoản đã tồn tại
3	Để trống các textbox	Kiểm tra khi textbox rỗng	Không nhập gì	Thông báo không được để trống

Bảng 4.2. Kiểm thử cho chức năng đăng nhập

STT	Tên testcase	Mục đích	Thực hiện	Kết quả
1	Đăng nhập tài khoản chính xác	Kiểm tra chức năng đăng nhập	Nhập thông tin tài khoản và mật khẩu	Truy cập trang quản trị
2	Đăng nhập tài khoản không chính xác	Kiểm tra chức năng đăng nhập	Nhập sai thông tin email hoặc mật khẩu	Thông báo không thành công

Bảng 4.3. Kiểm thử cho chức năng chính

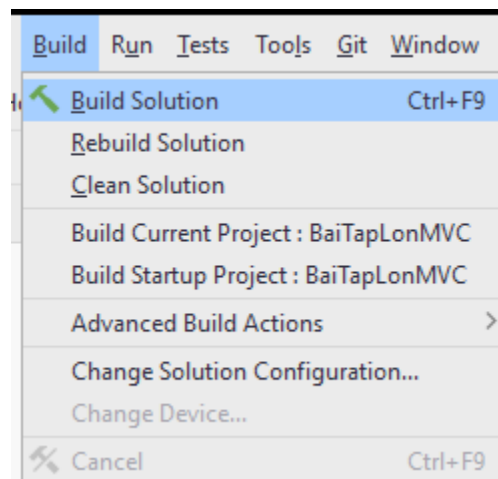
STT	Tên testcase	Mục đích	Thực hiện	Kết quả
1	Hiện danh sách sản phẩm	Kiểm tra hiển thị sản phẩm	Truy cập trang chủ	Hiện thị đầy đủ danh sách sản phẩm
2	Chi tiết sản phẩm	Kiểm tra chức năng hiển thị thông tin chi tiết sản phẩm	Chọn xem thông tin 1 sản phẩm bất kì	Hiện thị đầy đủ hình ảnh cũng như thông tin của sản phẩm
3	Tìm kiếm sản phẩm	Kiểm tra chức năng tìm kiếm sản phẩm	Nhập một sản phẩm bất kì	Hiện thông tin sản phẩm đúng với tìm kiếm
4	Thêm giỏ hàng	Kiểm tra chức năng giỏ hàng	Chọn thêm giỏ hàng trên một sản phẩm	Sản phẩm được thêm vào giỏ hàng

4.2.2 Đóng gói ứng dụng

Đóng gói ứng dụng thành file thực thi:

Bước 1: Tại Giao diện JetBrains Raider, chọn chức năng build trên thanh công cụ

Bước 2: Chọn Build Solution



Hình 4.1 Đóng gói ứng dụng

4.2.3 Triển khai ứng dụng

- Môi trường chạy: Window, Linux, MacOS

- Yêu cầu
 - .NET Framework 4.7.2 trở lên
 - CPU 2 nhân trở lên
 - Ram 2GB trở lên
 - Bộ nhớ trong trống: 500mb

KẾT LUẬN

Kết quả đạt được

Sau khi hoàn thành sản phẩm “Trang web bán đồ công nghệ” bản thân em đã đạt được những điều sau:

- Thực hiện được những công việc cơ bản trong lập trình web MVC như: sử dụng thành thạo các điều khiển, Entity Framework.
- Thực hiện tương đối nghiệp vụ bài toán:

- + Quản lý Sản phẩm(thêm, sửa, xóa, tìm kiếm)

- + Hiển thị danh sách sản phẩm trên trang web, thêm giỏ hàng, đặt hàng

Hạn chế của đề tài

- Giao diện vẫn còn chưa được đẹp mắt
- Đôi khi vẫn còn bị lỗi

Hướng phát triển của đề tài

- Xây dựng ứng dụng với các tính năng phong phú hơn
- Triển khai lên Hosting

TÀI LIỆU THAM KHẢO

- [1]. *Giáo trình môn Lập trình web với MVC*, Bộ môn CNPM, Trường ĐHSPKTHY.
- [2]. *Giáo trình môn cơ sở kỹ thuật lập trình*, Bộ môn CNPM, Trường ĐHSPKTHY.
- [3]. *Giáo trình môn cấu trúc dữ liệu giải thuật*, Bộ môn CNPM, Trường ĐHSPKTHY.
- [4]. *Giáo trình môn lập trình hướng đối tượng với C#*, Bộ môn CNPM, Trường ĐHSPKTHY.
- [5]. *Giáo trình môn Cơ sở dữ liệu*, Bộ môn CNPM, Trường ĐHSPKTHY.
- [6]. stackoverflow.com, <https://stackoverflow.com>