

PHỤ LỤC

A. HỆ THỐNG BÁN HÀNG ONLINE FRESHFOOD

Phần này tập trung minh họa các bước chính của hoạt động đặc tả yêu cầu, phân tích và thiết kế hệ thống Bán hàng Online Freshfood.

A.1. ĐẶC TẢ YÊU CẦU

A.1. 1. Mô tả yêu cầu

Xây dựng Website cho phép giới thiệu và đặt hàng online cho cửa hàng bán thực phẩm sạch FreshFood. Website cho phép khách hàng có thể xem và đặt hàng online, nhân viên của cửa hàng có thể quản lý thông tin của Website và quản lý hoạt động đặt và giao hàng. Các yêu cầu của hệ thống được mô tả như sau:

1. Cửa hàng bán các loại thực phẩm khác nhau, mỗi loại có các thông tin (Mã loại, tên loại, mô tả). Khi cửa hàng bán một loại thực phẩm mới nhân viên cửa hàng có nhiệm vụ cập nhật thông tin về loại thực phẩm vào trong hệ thống, nếu thông tin có sai sót nhân viên có thể thực hiện sửa hoặc xóa thông tin về loại thực phẩm.
2. Một loại thực phẩm sẽ có nhiều sản phẩm khác nhau. Mỗi sản phẩm trong cửa hàng gồm các thông tin (Mã sản phẩm, Tên sản phẩm, Đơn vị tính, mô tả, Hình ảnh sản phẩm). Khi cửa hàng bán một sản phẩm mới, nhân viên thực hiện nhập thông tin về sản phẩm vào trong hệ thống, nếu thông tin của sản phẩm có sai sót nhân viên có thể thực hiện sửa hoặc xóa thông tin của sản phẩm.
3. Giá bán của các mặt hàng được điều chỉnh theo giá bán của thị trường, khi giá của các mặt hàng thay đổi thì nhân viên có nhiệm vụ cập nhật lại giá bán, nếu giá bán có sai sót thì có thể thực hiện sửa giá bán. Giá bán của mỗi sản phẩm gồm các thông tin (Mã sản phẩm, Giá bán, Ngày hiệu lực, Ngày hết hiệu lực).
4. Cửa hàng có nhập sản phẩm của các nhà cung cấp khác nhau thông tin về nhà cung cấp gồm (Mã nhà cung cấp, Tên nhà cung cấp, Địa chỉ, Email, Số điện thoại). Khi có một nhà cung cấp mới nhân viên thực hiện nhập thông tin nhà cung cấp vào hệ thống, nếu thông tin nhà cung cấp có sai sót thì nhân viên có thể sửa hoặc xóa thông tin. Một nhà cung cấp có thể cung cấp nhiều sản phẩm khác nhau.

5. Khi nhập hàng về cửa hàng, nhân viên có nhiệm vụ nhập sản phẩm vào cửa hàng và lưu thông tin về hóa đơn nhập hàng vào hệ thống hóa đơn nhập hàng gồm thông tin (Mã hóa đơn nhập, mã nhà cung cấp, Ngày nhập, Thành tiền và các thông tin về sản phẩm nhập (Mã sản phẩm, Số lượng, Giá nhập, Hạn sử dụng)). Khi thông tin về hóa đơn nhập có sai sót, hệ thống cho phép nhân viên kho có thể sửa, xóa thông tin hóa đơn nhập.

6. Khi khách hàng muốn mua sản phẩm, khách hàng vào website xem thông tin của các sản phẩm (thông tin về các sản phẩm có thể được hiển thị theo nhóm như sản phẩm mới nhập, sản phẩm bán chạy nhất, sản phẩm khuyến mại, sản phẩm theo loại sản phẩm). Khi khách hàng quan tâm đến sản phẩm nào khách hàng có thể xem thông tin chi tiết về sản phẩm đó. Trong quá trình xem thông tin sản phẩm trên website, nếu khách hàng mua sản phẩm nào sẽ chọn sản phẩm đó đưa vào giỏ hàng. Ngoài việc xem thông tin sản phẩm khách hàng có thể đọc được các bài viết trên website.

7. Sau khi chọn mua các sản phẩm trong giỏ hàng, khách hàng có thể xem thông tin các sản phẩm trong giỏ hàng, sửa số lượng sản phẩm mua hoặc xóa sản phẩm chọn mua trong giỏ hàng. Khi khách hàng muốn đặt mua, khách hàng sẽ thực hiện đặt hàng, thông tin đơn đặt hàng được lưu vào hệ thống. Thông tin của đơn hàng gồm các thông tin (Mã đơn hàng, thông tin của khách hàng, ngày đặt hàng, thành tiền, địa chỉ giao hàng (nếu có), số điện thoại nhận hàng (nếu có), trạng thái đơn hàng và các thông tin về các sản phẩm đặt mua gồm (Mã sản phẩm, số lượng, đơn giá)).

8. Khi có một khách hàng mới đặt hàng lần đầu trong hệ thống, thông tin của khách hàng sẽ được lưu lại sau khi thực hiện xác minh được thông tin của khách. Thông tin của khách hàng gồm (Mã khách hàng, Tên khách hàng, Số điện thoại, Email, Địa chỉ). Khách hàng có thể xem lại thông tin khách hàng, nếu có sai sót khách hàng có thể thực hiện sửa thông tin.

9. Khách hàng có thể xem lại danh sách các đơn đặt hàng đã đặt và thông tin chi tiết từng đơn hàng đã đặt. Với các đơn hàng chưa được

giao, khách hàng có thể sửa số lượng các mặt hàng đã đặt mua, xóa bỏ các mặt hàng đã đặt hoặc hủy đơn đặt hàng.

10. Với các đơn hàng khách hàng đã đặt nhân viên cửa hàng có thể thực hiện xóa và xem được các đơn hàng chưa xác thực; xem danh sách các đơn hàng đã xác thực và xác nhận đã xử lý; xem được các đơn hàng đã xử lý và xác nhận đã chuyển cho nhà cung cấp; xem được các đơn hàng đang vận chuyển và xác nhận đã thanh toán hoặc đổi trả hoặc giao hàng không thành công.

11. Để quảng bá tốt thông tin các sản phẩm đến khách hàng trên Website, người quản trị có thể cập nhật các tin tức mới về sản phẩm hoặc các bài viết, nếu thông tin có sai sót người quản trị có thể thực hiện sửa và xóa thông tin về các bài viết gồm (Mã bài viết, Tiêu đề, Người đăng, Thời gian đăng, Nội dung).

12. Tất cả nhân viên thực hiện quản lý thông tin hệ thống phải đăng nhập trước khi thực hiện, thông tin nhân viên gồm (Mã nhân viên, Mật khẩu, Vai trò, Trạng thái hoạt động).

13. Nếu nhân viên cửa hàng vắng mặt/bận thì admin có thể đóng vai trò như nhân viên.

14. Khách hàng khi thực hiện xem thông tin về đơn hàng đã đặt, hoặc đặt hàng thì phải đăng nhập.

A.1.2. Từ điển dữ liệu

Khảo sát và thu thập yêu cầu từ người dùng và các chuyên gia miền, đội phân tích yêu cầu sẽ xây dựng được các từ điển dữ liệu của miền ứng dụng như trong Bảng A-1.

Bảng A-1 Từ điển dữ liệu

Tên	Kiểu dữ liệu	Độ rộng	Mô tả
LoaiSanPham			Loại sản phẩm
MaLoai	Chuỗi ký tự	20	Mã loại sản phẩm
TenLoai	Chuỗi ký tự	50	Tên loại sản phẩm

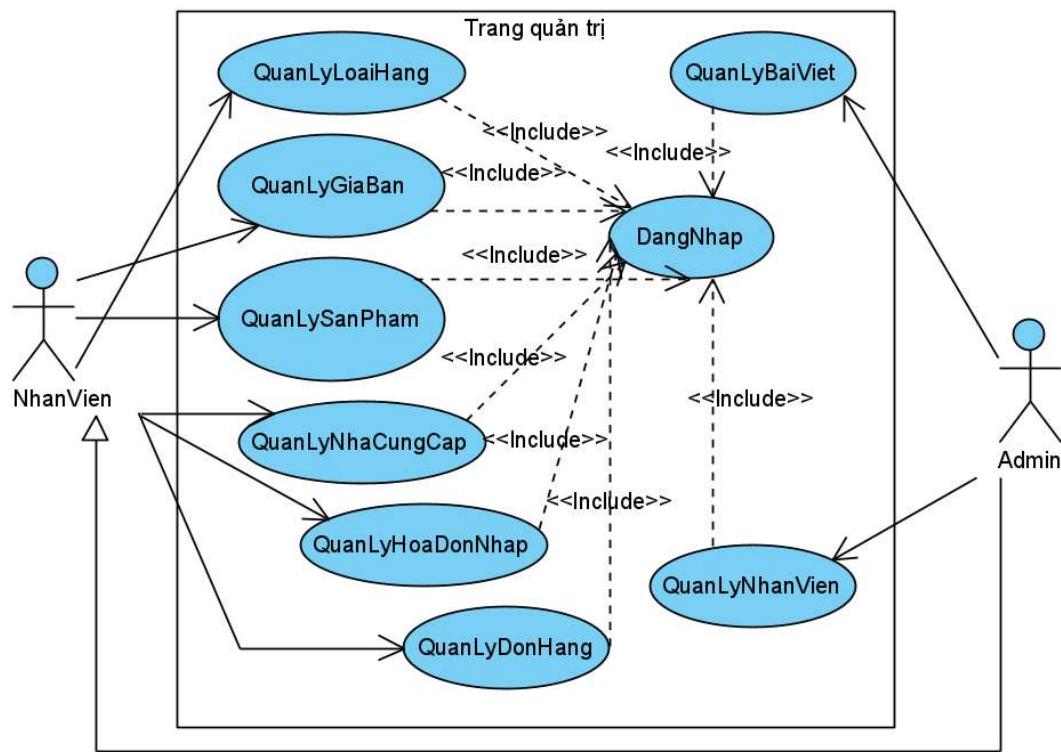
SanPham			Sản phẩm
MaSP	Chuỗi ký tự	20	Mã sản phẩm
TenSP	Chuỗi ký tự	50	Tên sản phẩm
DonVi	Chuỗi ký tự	50	Đơn vị tính
MoTa	Chuỗi ký tự	max	Mô tả sản phẩm
Anh	Chuỗi ký tự	50	Ảnh sản phẩm
NhaCungCap			Nhà cung cấp
MaNCC	Chuỗi ký tự	20	Mã nhà cung cấp
TenNCC	Chuỗi ký tự	50	Tên nhà cung cấp
DiaChi	Chuỗi ký tự	max	Địa chỉ nhà cung cấp
SDThoai	Chuỗi ký tự	10	Số điện thoại
KhachHang			Khách hàng
MaKhach	Chuỗi ký tự	20	Mã khách hàng
TenKhach	Chuỗi ký tự	50	Tên khách hàng
SDThoai	Chuỗi ký tự	10	Số điện thoại
Email	Chuỗi ký tự	50	Email khách hàng
DiaChi	Chuỗi ký tự	max	Địa chỉ khách hàng
GiaBan			Giá bán
MaSP	Chuỗi ký tự	20	Mã sản phẩm
GiaBan	Số		Giá bán sản phẩm
NgayBD	Chuỗi ký tự	20	Ngày bắt đầu
NgayKT	Chuỗi ký tự	20	Ngày kết thúc
DonHang			Đơn hàng
MaDonHang	Chuỗi ký tự	20	Mã đơn hàng

MaKhach	Chuỗi ký tự	50	Mã khách hàng
DiaChiNhan	Chuỗi ký tự	max	Địa chỉ nhận hàng
TinhTrang	Chuỗi ký tự	50	Tình trạng
GhiChu	Chuỗi ký tự	50	Ghi chú
ThanhTien	Số		Thành tiền
NgayDat	Chuỗi ký tự	20	Ngày đặt hàng
CTDonHang			Chi tiết đơn hàng
MaSP	Chuỗi ký tự	20	Mã sản phẩm
SoLuong	Số		Số lượng sản phẩm
DonGia	Số		Đơn giá
HoaDonNhap			Hóa đơn nhập
MaHDNhap	Chuỗi ký tự	20	Mã hóa đơn nhập
NgayNhap	Chuỗi ký tự	20	Ngày nhập hóa đơn
NhaCC	Chuỗi ký tự	50	Nhà cung cấp
NhanVienNhap	Chuỗi ký tự	50	Nhân viên nhập
TongTien	Số		Tổng tiền hóa đơn
CTHoaDonNhap			Chi tiết hóa đơn nhập
MaSP	Chuỗi ký tự	20	Mã sản phẩm
SoLuong	Số		Số lượng
GiaNhap	Số		Giá nhập
HanSuDung	Chuỗi ký tự	50	Hạn sử dụng
User			Người dùng
Username	Chuỗi ký tự	50	Tên người dùng
Password	Chuỗi ký tự	50	Mật khẩu

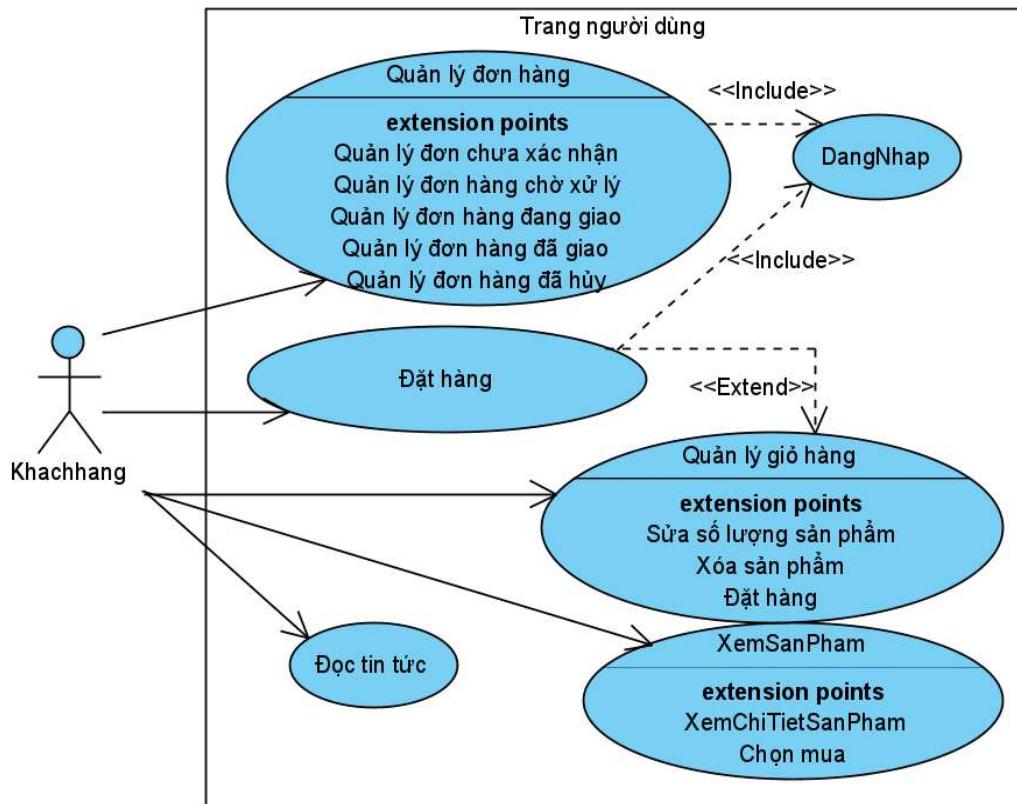
BaiViet			Sản phẩm
MaBaiViet	Chuỗi ký tự	20	Mã bài viết
TieuDe	Chuỗi ký tự	50	Tiêu đề bài viết
NoiDung	Chuỗi ký tự	max	Nội dung
ThoiGianDang	Chuỗi ký tự	20	Thời gian đăng bài
MaNV	Chuỗi ký tự	20	Mã nhân viên

A.1.3. Mô hình ca sử dụng

Hệ thống gồm hai trang chính. Một trang dành cho người mua hàng và một trang dành cho người quản trị và nhân viên của cửa hàng. Tác nhân tham gia vào trang người dùng dành cho người mua hàng là khách hàng khi thực hiện vào website xem thông tin sản phẩm và đặt hàng online. Tác nhân tham gia vào trang quản lý thông tin của Website và quản lý hoạt động đặt và giao hàng gồm hai tác nhân là người quản trị và nhân viên. Khảo sát trên các tác nhân và sử dụng các mối quan hệ để tổ chức lại biểu đồ ca sử dụng chúng ta xây dựng được biểu đồ ca sử dụng mức tổng quát và phân rã cho hai trang trên. Hình A-1 và A-2 là biểu đồ tổng quát cho trang quản trị và trang người dùng.

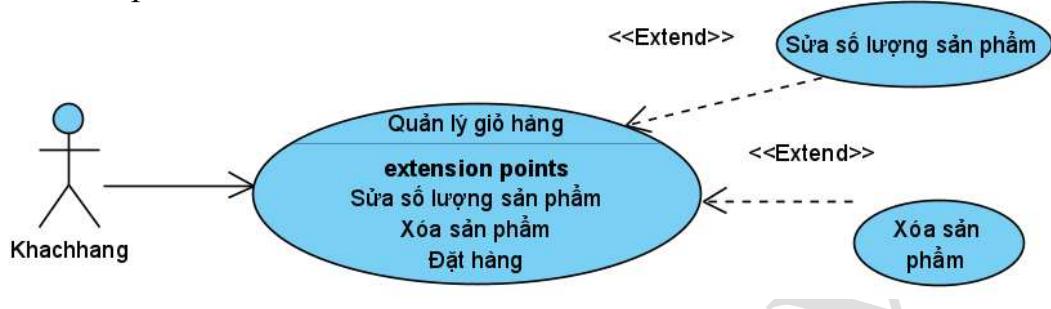


Hình 0-1 Biểu đồ tổng quát trang quản trị

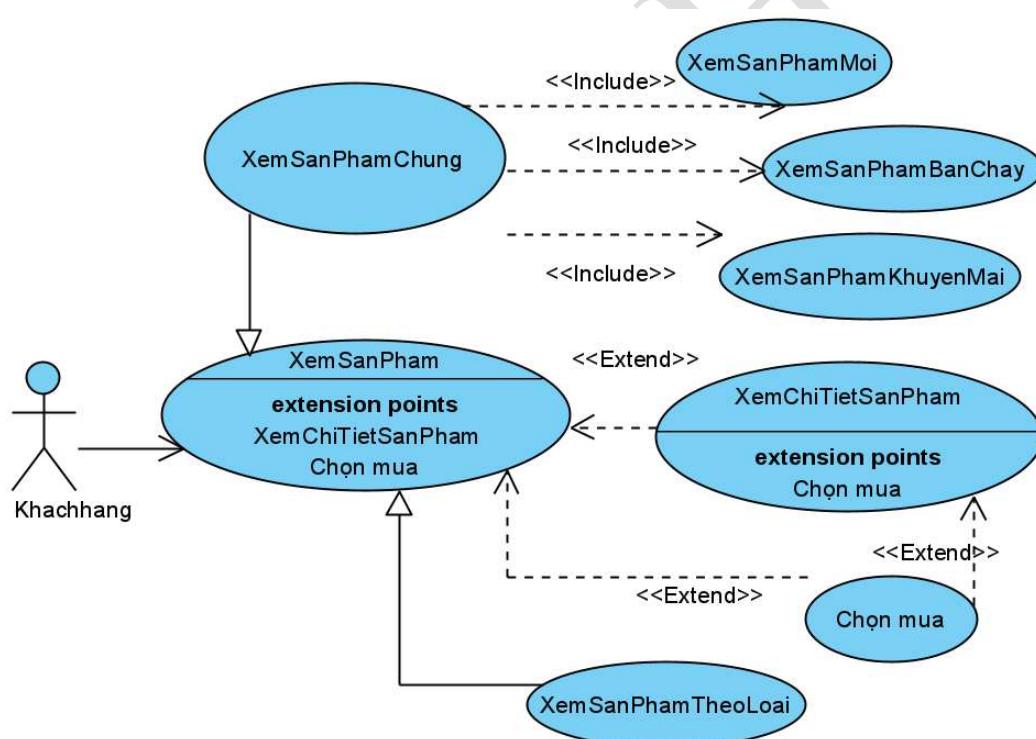


Hình A-2 Biểu đồ tổng quát trang người dùng

Sau khi xây dựng biểu đồ use case mức tổng quát, chúng ta tiếp tục xây dựng các biểu đồ use case phân rã. Hình A-3 là biểu đồ phân rã của ca sử dụng Quản lý giỏ hàng. Hình A-4 là biểu đồ phân rã của ca sử dụng *Xem sản phẩm*.



Hình A-3 Biểu đồ phân rã của use case Quản lý giỏ hàng



Hình A-4 Biểu đồ phân rã use case Xem sản phẩm

Sau khi phân rã hết tất cả các ca sử dụng trong hệ thống, bước tiếp theo đội phát triển nghiên cứu nghiệp vụ làm việc với người dùng, khách hàng, chuyên gia miền để xây dựng tài liệu đặc tả chi tiết cho từng ca sử dụng. Các thông tin mô tả chi tiết ca sử dụng gồm một số các thông tin chung như mô tả (description), tác nhân (actor), tiền điều kiện (pre

condition), hậu điều kiện (post condition), luồng sự kiện. Từ các tài liệu mô tả chi tiết từng ca sử dụng, chúng ta chuyển sang pha phân tích và thiết kế tiếp theo. Dưới đây là đặc tả chi tiết của ca sử dụng xem thông tin sản phẩm theo loại.

XemSanPhamTheoLoai

ID: UC04

Primary Actors

Khachhang

Scenarios

Flow of Events

1. Người dùng chọn loại sản phẩm cần xem
2. **SYSTEM** Hệ thống hiển thị giao diện xem sản phẩm
3. **SYSTEM** Hệ thống lấy về danh sách sản phẩm theo loại đã chọn, không có sản phẩm nào chuyển sang luồng phụ 2a
4. **SYSTEM** Hiển thị danh sách sản phẩm

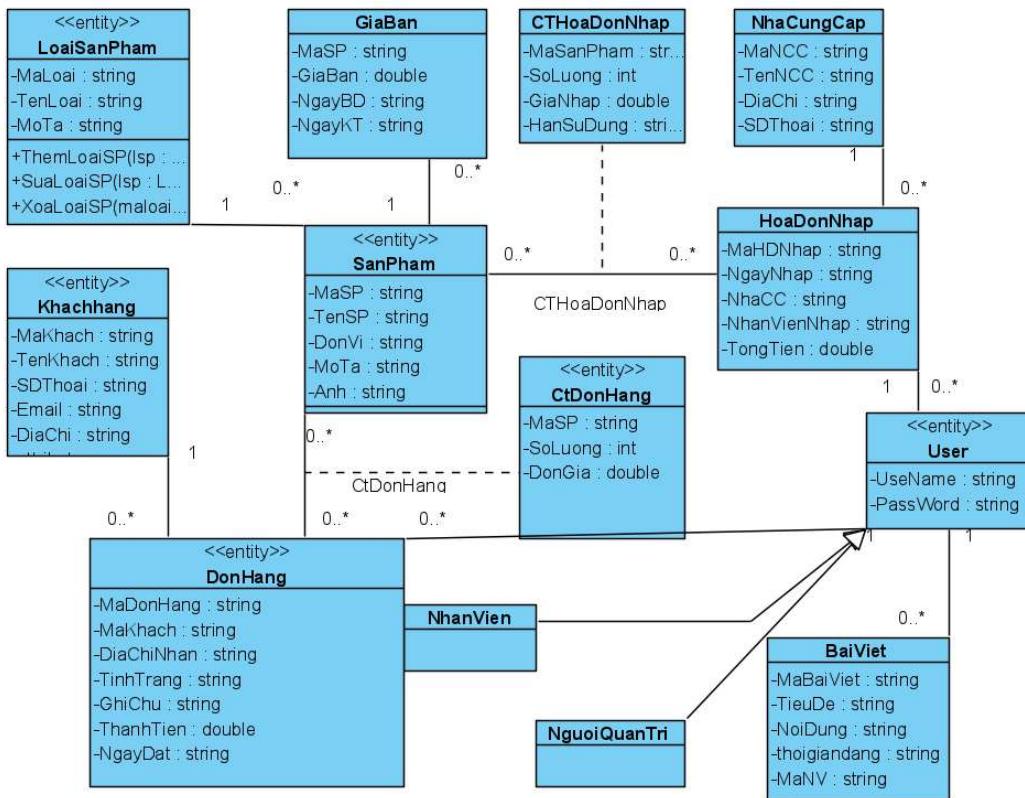
Extension:

- 3.a. Không có sản phẩm
 1. **SYSTEM** Hiển thị thông báo không có sản phẩm thuộc loại đã chọn

A.2. PHÂN TÍCH

A.2.1. Biểu đồ lớp thực thể

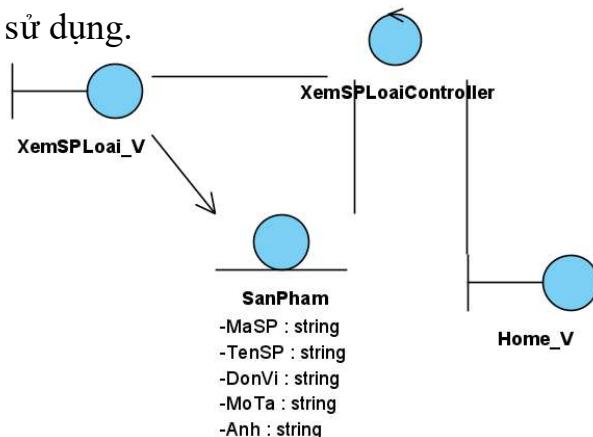
Đội phát triển tiến hành phân tích yêu cầu người dùng, bảng từ điển dữ liệu từ đó xác định được các thực thể lưu trữ dữ liệu trong hệ thống và mô hình hóa bằng biểu đồ lớp thực thể.



Hình A-5 Biểu đồ lớp thực thể hệt

A.2.2. Biểu đồ lớp phân tích use case

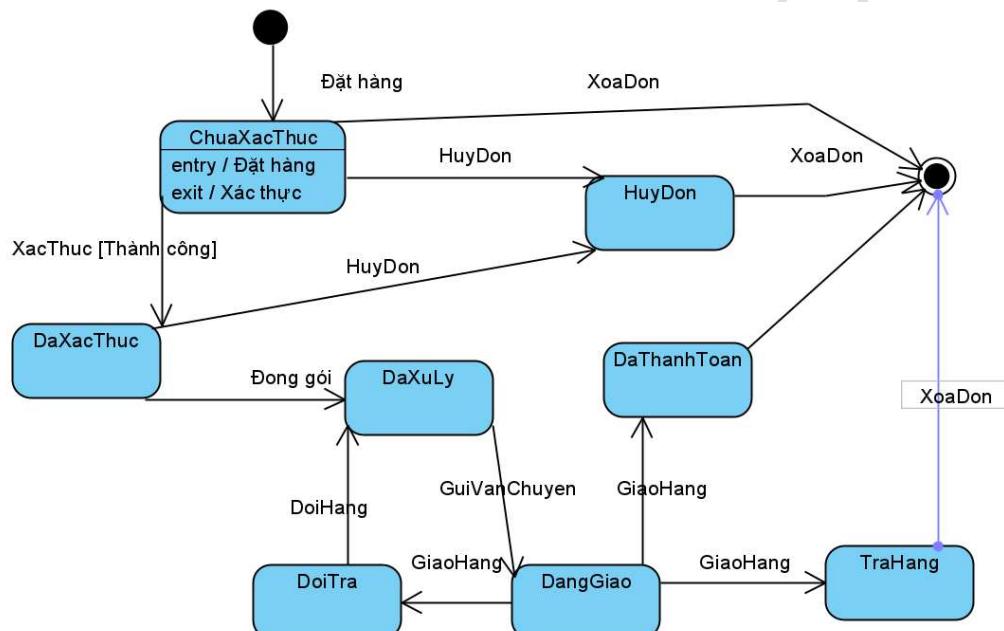
Đội phát triển phần mềm sẽ tiến hành phân tích lần lượt từng ca sử dụng, từ ca sử dụng tổng quát đến ca sử dụng phân rã. Từ kịch các luồng sự kiện trong mô tả chi tiết của từng ca sử dụng, nhà phát triển tiến hành phân tích để xác định các lớp tham gia thực hiện ca sử dụng và mô hình hóa bằng biểu đồ lớp VOPC cho từng ca sử dụng. Dựa trên phân tích kiến trúc đã lựa chọn, nhà phát triển sẽ xác định được các lớp tham gia thực hiện ca sử dụng.



Hình A-6 Biểu đồ lớp phân tích của use case Xem sản phẩm theo loại

A.2.3. Biểu đồ trạng thái

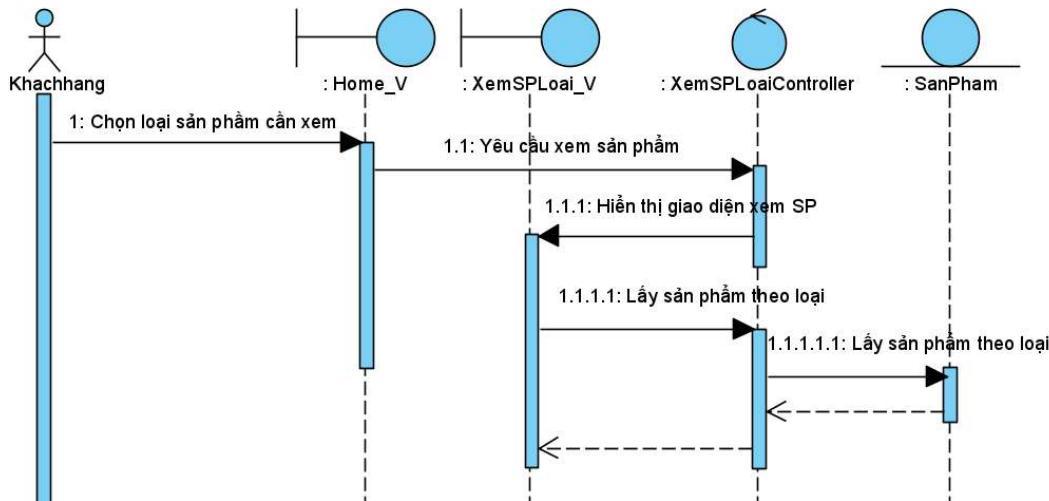
Với mỗi lớp thực thể, chúng ta phân triển hành phân tích để xác định các trạng thái và chuyển trạng thái của một đối tượng khi tham gia thực hiện một ca sử dụng hoặc trong vòng đời tồn tại của nó. Nếu cần thiết, chúng ta có thể mô hình hóa các trạng thái và chuyển trạng thái của đối tượng khi một ca sử dụng thực hiện hoặc vòng đời tồn tại của chúng trong phần mềm bằng biểu đồ trạng thái. Hình A-7 dưới đây là biểu đồ biến đổi trạng thái của một đối tượng đơn hàng trong vòng đời tồn tại của nó.



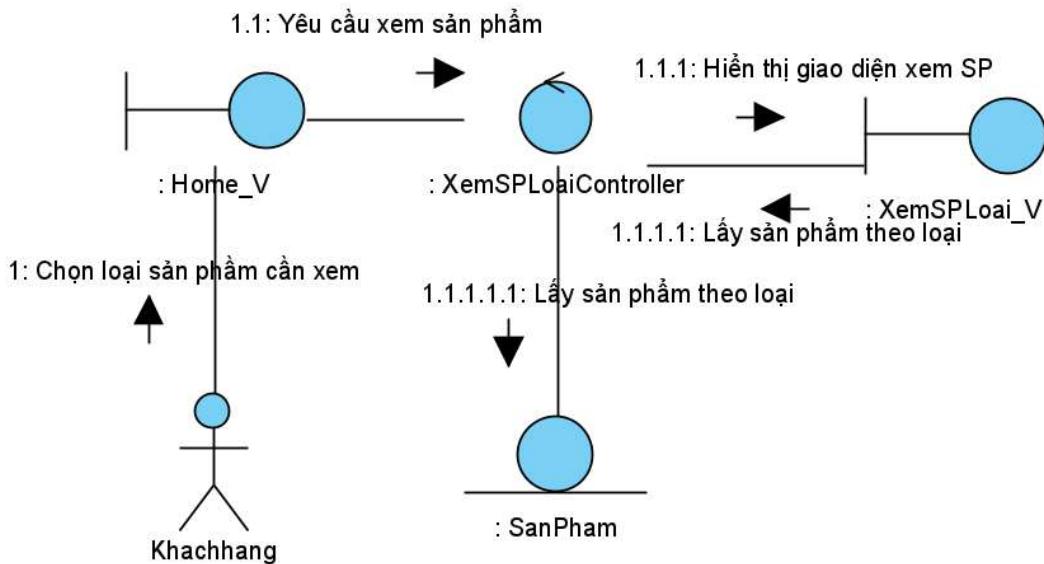
Hình A-7 Biểu đồ trạng thái của đơn hàng

A.2.4. Biểu đồ tương tác

Bước tiếp theo, biểu đồ tuần tự hoặc biểu đồ cộng tác được sử dụng để mô hình hóa các chuỗi tương tác giữa các đối tượng là thể hiện của các lớp tham gia thực hiện ca sử dụng. Hình A-8 là biểu đồ hoạt động của ca sử dụng *xem sản phẩm theo loại*. Hình A-9 là biểu đồ cộng tác của ca sử dụng *xem sản phẩm theo loại*.



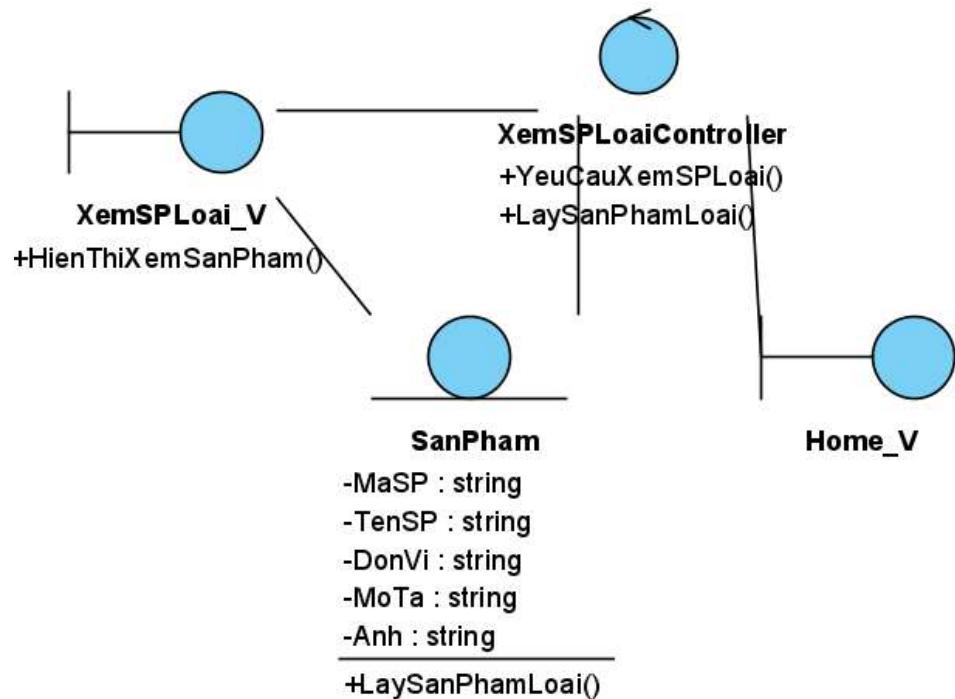
Hình A-8 Biểu đồ tuần tự ca sử dụng Xem sản phẩm theo loại



Hình A-9 Biểu đồ công tác ca sử dụng Xem sản phẩm theo loại

A.2.5. Phân bô hành vi tới các lớp

Sau khi xây dựng biểu đồ tuần tự, nhà phát triển thực hiện phân bô hành vi cho các lớp dựa vào biểu đồ tuần tự như Hình A-10 dưới đây.

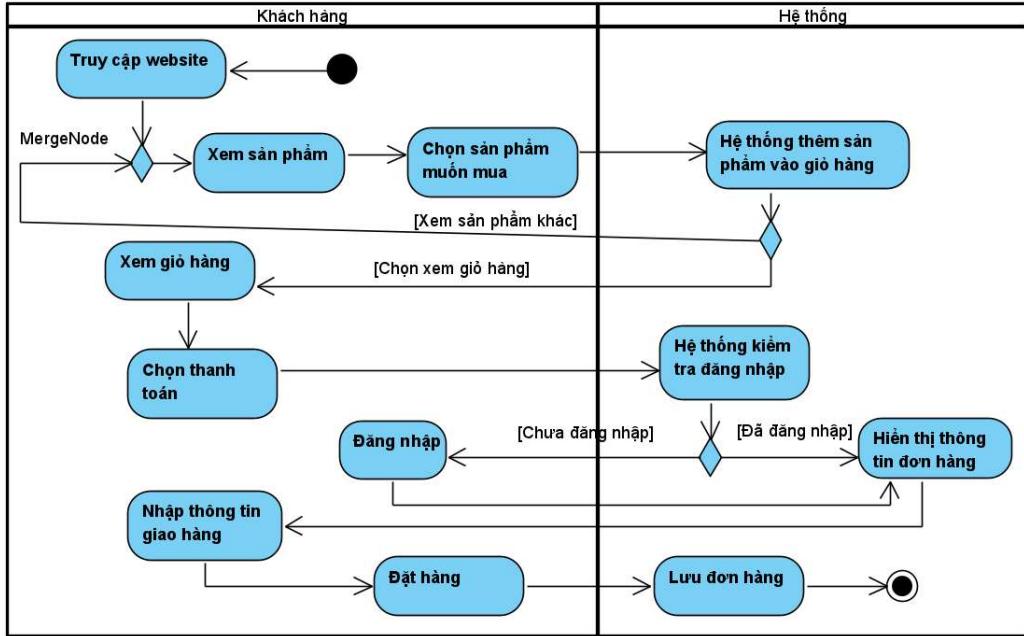


Hình A-10 Phân bổ hành vi cho các lớp

A.2.6. Biểu đồ hoạt động

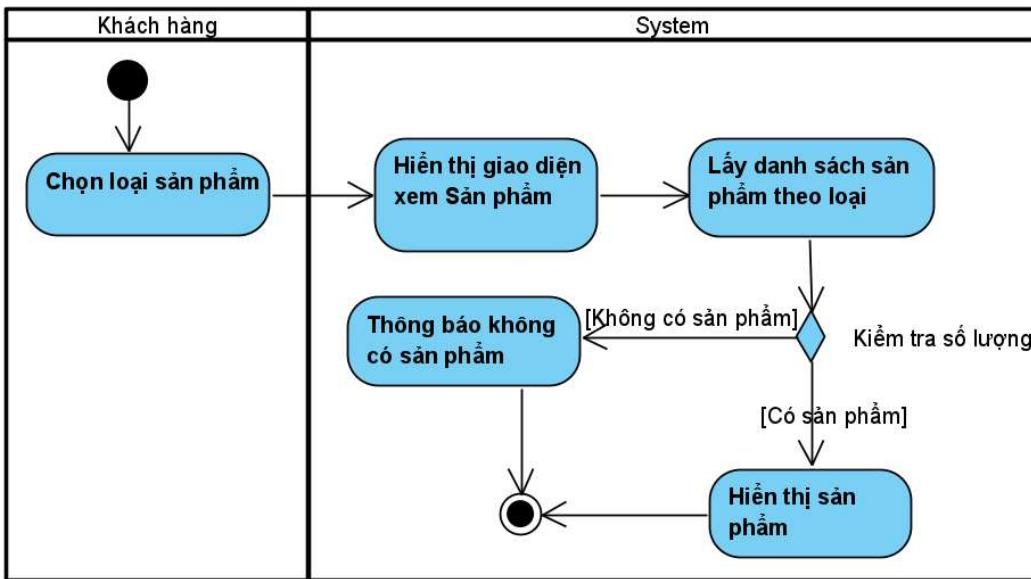
Biểu đồ hoạt động được sử dụng trong các pha khác nhau trong quy trình phát triển phần mềm. Trong giai đoạn phân tích nghiệp vụ, biểu đồ hoạt động có thể được sử dụng để mô hình hóa quy trình nghiệp vụ của hệ thống khảo sát. Trong giai phân tích use case, biểu đồ hoạt động có thể được sử dụng để mô hình hóa luồng sự kiện trong use case. Trong giai đoạn thiết kế, biểu đồ hoạt động có thể mô hình hóa sự hoạt động của một phương thức, một thuật toán.

Hình A-11 dưới đây là quy trình đặt hàng của khách hàng trên website đã được mô hình hóa bằng biểu đồ hoạt động.



Hình A-11 Quy trình đặt hàng của khách hàng

Hình A-12 là biểu đồ hoạt động mô hình hóa chuỗi tương tác thực hiện use case *Xem sản phẩm theo loại*.



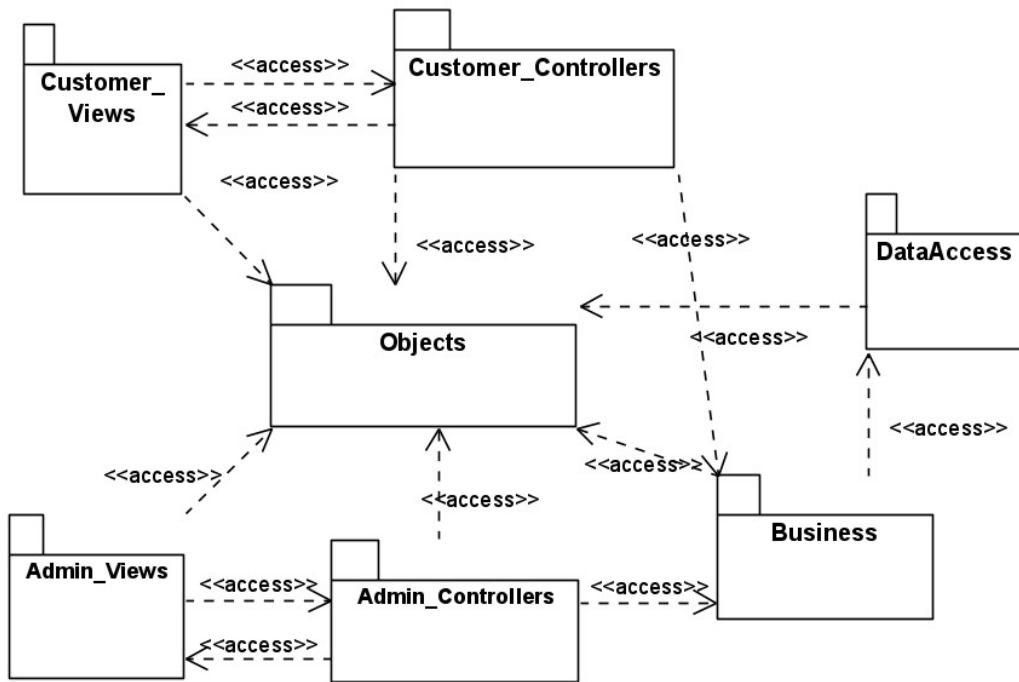
Hình A-12 Biểu đồ hoạt động của use case *Xem sản phẩm theo loại*

A.3. Thiết kế

Trong pha thiết kế, đội thiết kế sẽ tổ chức thiết kế xây dựng hệ thống trên môi trường triển khai cụ thể. Các chế tác đầu vào của pha này là các mô hình thu được từ pha phân tích ở bước trên. Các hiện thực hóa của ca sử dụng sẽ được tiếp tục làm mịn đến các lớp thiết kế chi tiết. Hoạt động đầu tiên là thiết kế kiến trúc, khi đó đội thiết kế sẽ tiến hành xác định kiến trúc vật lý của ứng dụng, tổ chức các gói, các thành phần thiết kế. Sau đó là hoạt động thiết kế mô đun. Hoạt động này sẽ thực hiện xây dựng các biểu đồ lớp thiết kế chi tiết cho từng ca sử dụng và tổng thể cả hệ thống.

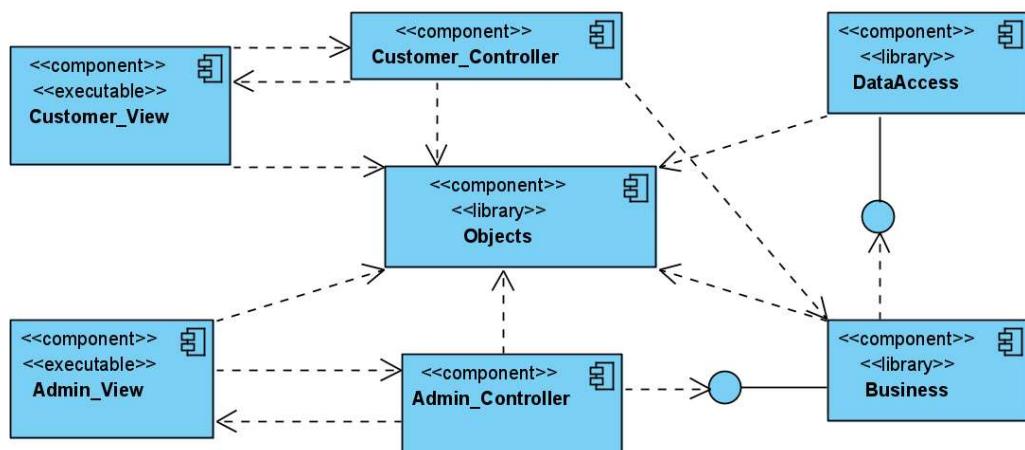
A.3.1. Biểu đồ gói và biểu đồ thành phần

Ứng dụng được triển khai dưới dạng Website bằng công nghệ ASP.NET MVC kết hợp với kiến trúc 3 tầng. Khi đó các lớp thực thể trong mô hình VOPC của mỗi use case trong giai đoạn phân tích được tách thành 2 lớp: 1 lớp chứa các thuộc tính (các lớp này được tổ chức vào gói Objects) và 1 lớp chứa các hành vi thao tác với cơ sở dữ liệu (các lớp này được tổ chức vào gói DataAccess). Mỗi lớp điều khiển được tách thành 2 lớp: 1 lớp Controller (các lớp này được tổ chức vào 2 gói Customer.Controllers và Admin.Controllers) và 1 lớp Business chứa các nghiệp vụ xử lý và tương tác với lớp cơ sở dữ liệu thông qua lớp ở tầng DataAccess (các lớp này được tổ chức vào gói Business). Các lớp giao diện và mã nguồn phía client được tổ chức trong gói Customer.Views và Admin.Views. Các lớp của dự án được tổ chức thành các gói chính như trong Hình A-13.



Hình A-13 Biểu đồ gói của hệ thống

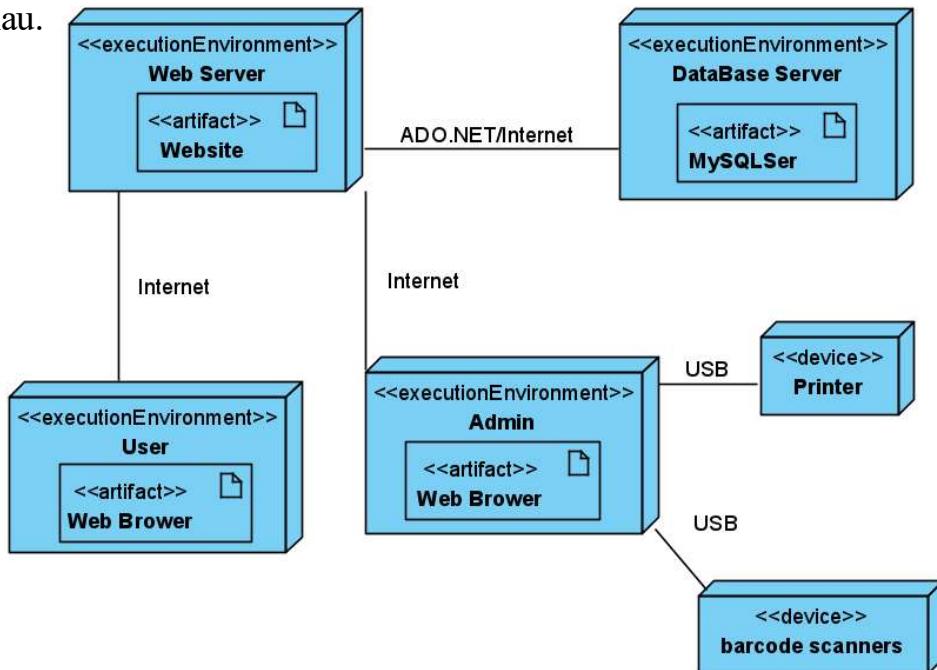
Các gói Objects, Business, DataAccess được triển khai thành các thành phần dll, các gói Views và Controllers được triển khai thành thành phần mã nguồn có thể thực thi được của hai subsystem dành cho khách hàng và người quản trị như trong Hình A-14.



Hình A-14 Biểu đồ thành phần của hệ thống

A.3.2. Biểu đồ triển khai hệ thống

Dựa trên kiến trúc và giải pháp thiết kế phần mềm xác định các node triển khai của hệ thống bao gồm có các node processor và các node device. Sau đó, chúng ta xác định liên kết giữa các Node. Liên kết giữa các Node là giao thức, cách thức mà các Node trao đổi thông tin với nhau.



Hình A-15 Biểu đồ triển khai hệ thống

Hình A-15 là biểu đồ triển khai hệ thống. Trong đó DataBase được triển khai trên một DataServer, Website được triển khai trên một Webserver. Khách hàng mua hàng dùng máy tính có kết nối mạng Internet để truy cập vào website thông qua trình duyệt web để xem hàng hóa và đặt mua hàng. Người quản trị và nhân viên mua hàng dùng máy tính kết nối mạng Internet để truy cập vào website thông qua trình quyết web để quản trị nội dung và quản lý hoạt động đặt hàng. Tại máy người quản trị hoặc nhân viên sẽ có máy in và máy quyết mã vạch kết nối với máy tính để in đơn hàng và quyết mã sản phẩm.

A.3.3. Thiết kế cho từng ca sử dụng

Với từng ca sử dụng, chúng ta tập trung thiết kế cho từng gói thiết kế đã được xác định ở trên. Ở đây chúng ta lần lượt xác định các lớp thiết kế cho từng ca sử dụng.

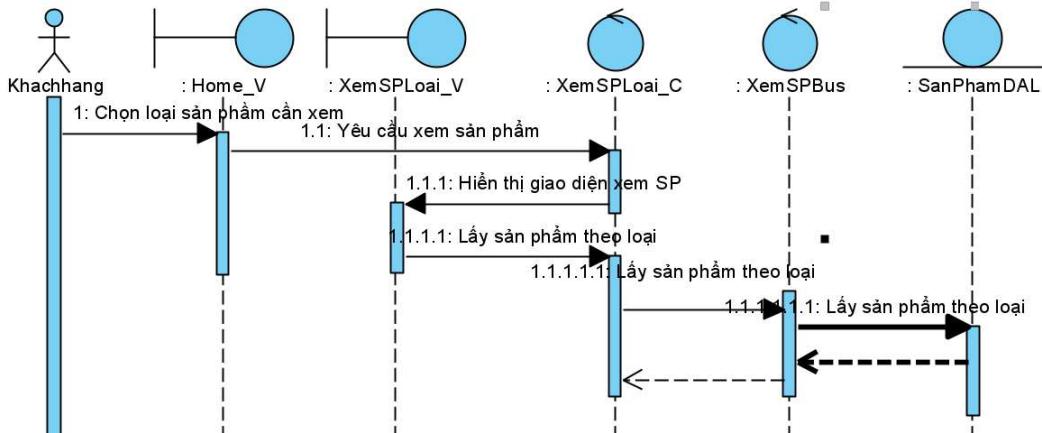
a) Xác định các phần tử thiết kế

Đầu vào để xác định các lớp thiết kế là các lớp trong pha phân tích. Mỗi lớp trong pha phân tích sẽ trở thành một khối thiết kế. Một khối thiết kế có thể gồm một hoặc nhiều lớp thiết kế. Với mỗi biểu đồ VOPC trong pha phân tích, đội thiết kế tiến hành bổ sung, tách lớp, gộp các lớp theo kiến trúc, công nghệ triển khai đã lựa chọn. Bảng A-2 là các lớp thiết kế của ca sử dụng *Xem sản phẩm theo loại*.

Bảng A-2 Các lớp thiết kế của use case Xem sản phẩm theo loại

Lớp phân tích	Lớp thiết kế	Chú thích
XemSPLoai_V	XemSPLoai_View	Trang HTML + JavaScript (Lớp biên use case)
Home_V	Home_View	Trang HTML + JavaScript (lớp biên hệ thống)
XemSPLoai_C	- XemSPLoaiController	Lớp Controller trong MVC mã C#
	- XemSPLoaiBus	Lớp xử lý nghiệp vụ mã C#
SanPham	SanPham	Lớp gồm các thuộc tính mô tả tính chất của lớp SanPham mã C#
	SanPhamDAO	Lớp chứa các hành vi thao tác với CSDL quan hệ mã C#

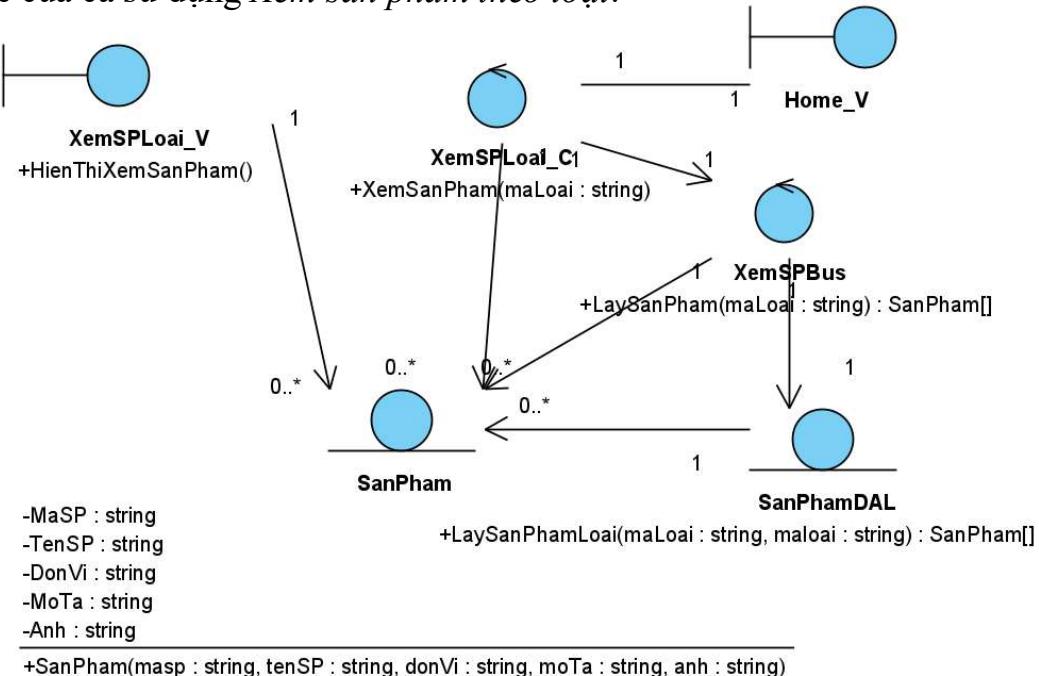
Sau khi xác định được các lớp thiết kế, nhà thiết kế tiến hành vẽ lại biểu đồ tuần tự. Ví dụ Hình A-16 là biểu đồ tuần tự được vẽ lại của ca sử dụng *Xem sản phẩm theo loại*.



Hình A-16 Biểu đồ tuần tự mức thiết kế

b) Thiết kế biểu đồ lớp chi tiết

Tiếp theo, biểu đồ lớp thiết kế được bổ sung đầy đủ các thuộc tính, phương thức, quan hệ giữa các lớp như Hình A-17 là biểu đồ lớp thiết kế của ca sử dụng *Xem sản phẩm theo loại*.



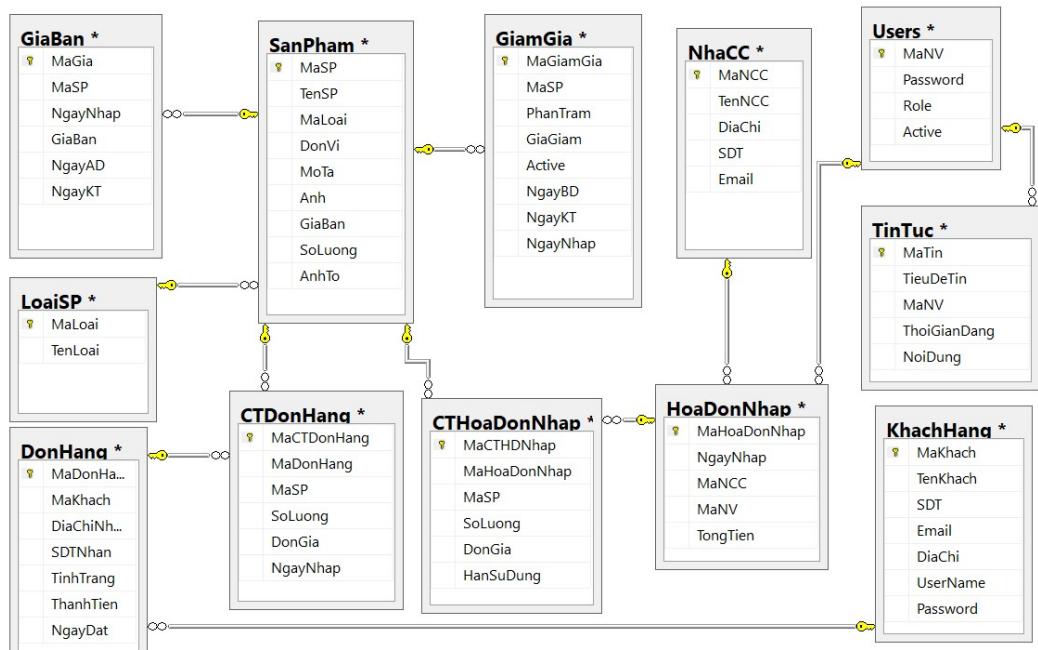
Hình A-17 Biểu đồ lớp thiết kế chi tiết của use case *Xem sản phẩm theo loại*

A.3.4. Thiết kế cơ sở dữ liệu

Đầu vào để thiết kế cơ sở dữ liệu là biểu đồ lớp thực thể và tài liệu từ điển thuật ngữ. Trong đó, các lớp được ánh xạ sang các bảng trong cơ sở dữ liệu, thuộc tính được ánh xạ thành các trường và các quan hệ được ánh xạ theo nguyên tắc:

- Quan hệ Association giữa các lớp:
 - Quan hệ 1-n: chuyển khóa chính bên 1 sang bên nhiều làm khóa ngoại, loại bỏ các thuộc tính bên nhiều suy ra được từ khóa ngoại; chuyển các thuộc tính còn lại ở hai bên thành trường dữ liệu trong bảng.
 - Quan hệ n-n: chuyển khóa chính hai bên nhiều sang lớp liên kết, loại bỏ các thuộc tính trong lớp liên kết mà suy ra được từ khóa ngoại, chuyển lớp liên kết thành bảng, chuyển hai lớp liên kết thành hai bảng, chuyển các thuộc tính còn lại thành trường dữ liệu.
- Quan hệ Generalization có ba cách chuyển
 - Cách 1: chuyển tất cả các thuộc tính từ lớp cha và các lớp con thành các trường của một bảng duy nhất
 - Cách 2: chuyển tất cả các lớp con thành bảng và gộp các thuộc tính của lớp cha với thuộc tính của lớp con để chuyển thành các trường của bảng
 - Cách 3: chuyển lớp cha và các lớp con thành bảng
- Quan hệ composition, khi đó hai lớp sẽ được ánh xạ thành hai bảng và theo quy tắc đã xác định khóa chính của lớp liên kết được chuyển sang làm khóa ngoại với lớp được liên kết.
- Quan hệ aggregation, khi đó ánh xạ hai bảng ở đầu liên kết được ánh xạ thành hai bảng. Khóa chính của bảng bộ phận được chuyển sang làm khóa ngoại của bảng toàn thể.

Hình A-18 là cơ sở dữ liệu quan hệ được thiết kế của hệ thống.



Hình A-18 Cơ sở dữ liệu quan hệ

B. ÁNH XẠ UML SANG MÃ NGUỒN

Sau khi quá trình phân tích và thiết kế hệ thống kết thúc, kết quả của quá trình này chúng ta đã xác định được cái mà hệ thống cần làm (What) và làm như thế nào (How). Chính là kết quả của một tập các biểu đồ đã được mô hình hóa để biểu diễn các khía cạnh khác nhau của phần mềm. Quá trình lập trình là quá trình hiện thực hóa các thiết kế phần mềm (implementation). Khi đó các biểu đồ sẽ được ánh xạ, dịch chuyển sang mã nguồn trong không gian giải pháp.

- Biểu đồ Use case được ánh xạ thành các chức năng mà hệ thống cần xây dựng.
- Biểu đồ thành phần, biểu đồ gói ánh xạ thành cách tổ chức các lớp trong hệ thống phần mềm.
- Biểu đồ triển khai ánh xạ thành các hệ thống con và các thiết bị phần cứng mà hệ thống cần có.
- Biểu đồ lớp thực thể là biểu đồ của các lớp lưu trữ thông tin của hệ thống sẽ được chuyển thành cơ sở dữ liệu của hệ thống.
- Để hiện thực hóa từng ca sử dụng, chúng ta sẽ chuyển biểu đồ lớp thiết kế chi tiết thành mã nguồn triển khai. Trong đó lớp sẽ được chuyển thành class, thuộc tính chuyển thành thành phần dữ liệu, phương thức chuyển thành phương thức, tùy từng mối quan hệ giữa các lớp mà được chuyển khác nhau.

B.1. Ánh xạ biểu đồ lớp sang mã nguồn

B.1.1. Ánh xạ lớp

Mỗi lớp trong biểu đồ lớp được ánh xạ tương ứng với một class được cài đặt trong ngôn ngữ lập trình như lớp Student dưới đây sẽ được ánh xạ tới class Student.

Student
- name: String
+ addSchedule (theSchedule: Schedule, forSemester: Semester)
+ hasPrerequisites(forCourseOffering: CourseOffering) : int
passed(theCourseOffering: CourseOffering) : int

```

public class Student
{
    private String name;
    public void addSchedule (Schedule theSchedule; Semester
forSemester)
    {
        //Mã thân phương thức
    }

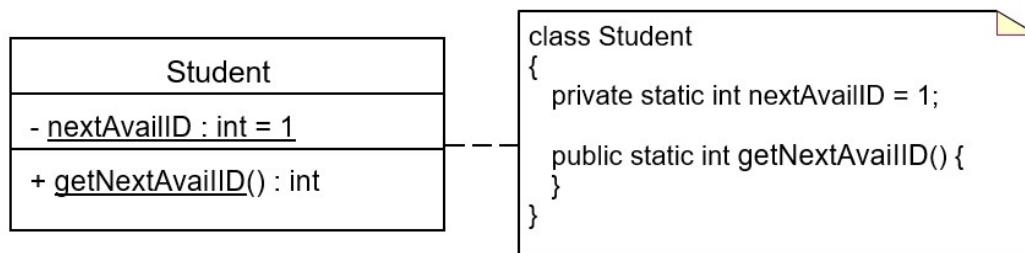
    public      boolean      hasPrerequisites(CourseOffering
forCourseOffering) {
    }

    protected boolean passed(CourseOffering theCourseOffering)
    {
    }
}

```

Phạm vi của phương thức và thuộc tính

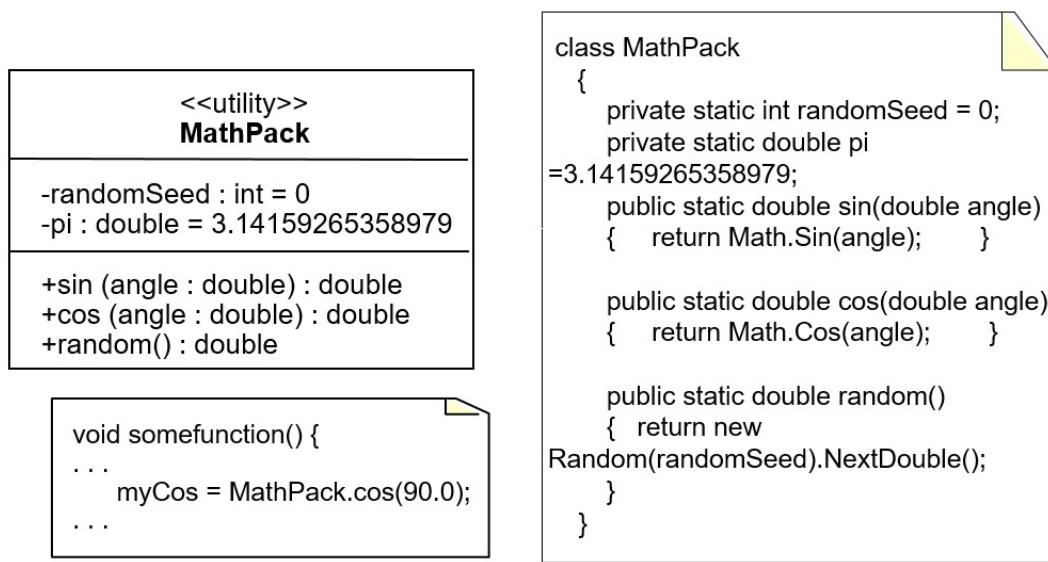
Phạm vi truy cập của phương thức và thuộc tính trong UML được ánh xạ thành phạm vi truy cập tương ứng của phương thức và thuộc tính trong mã chương trình như Hình B-1 dưới đây.



Hình B-5-33 Ví dụ ánh xạ phạm vi truy cập sang mã

Lớp tiện ích

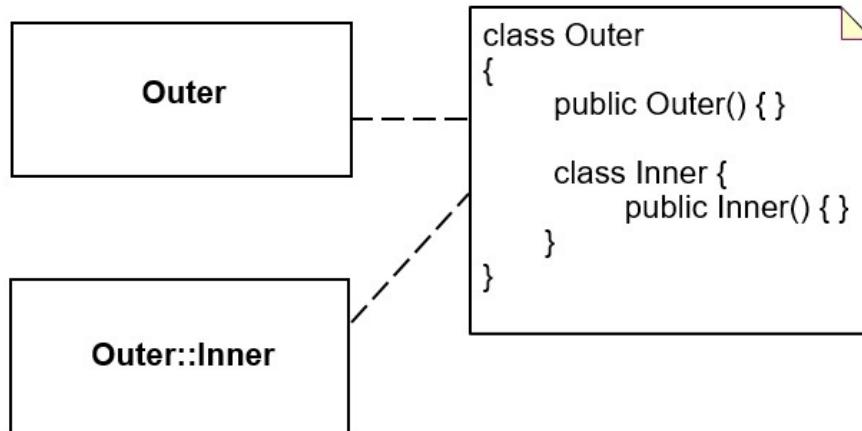
Lớp tiện ích là lớp chứa các thuộc tính và phương thức toàn cục. Các phương thức và thuộc tính được cài đặt dưới dạng tĩnh (static) như Hình B-2 dưới đây.



Hình B-5-34 Chuyển lớp tiện ích sang mã

Lớp lồng nhau

Lớp Inter là lớp ẩn trong lớp Outer mà chứa các phương thức chỉ được sử dụng bởi lớp Outer.

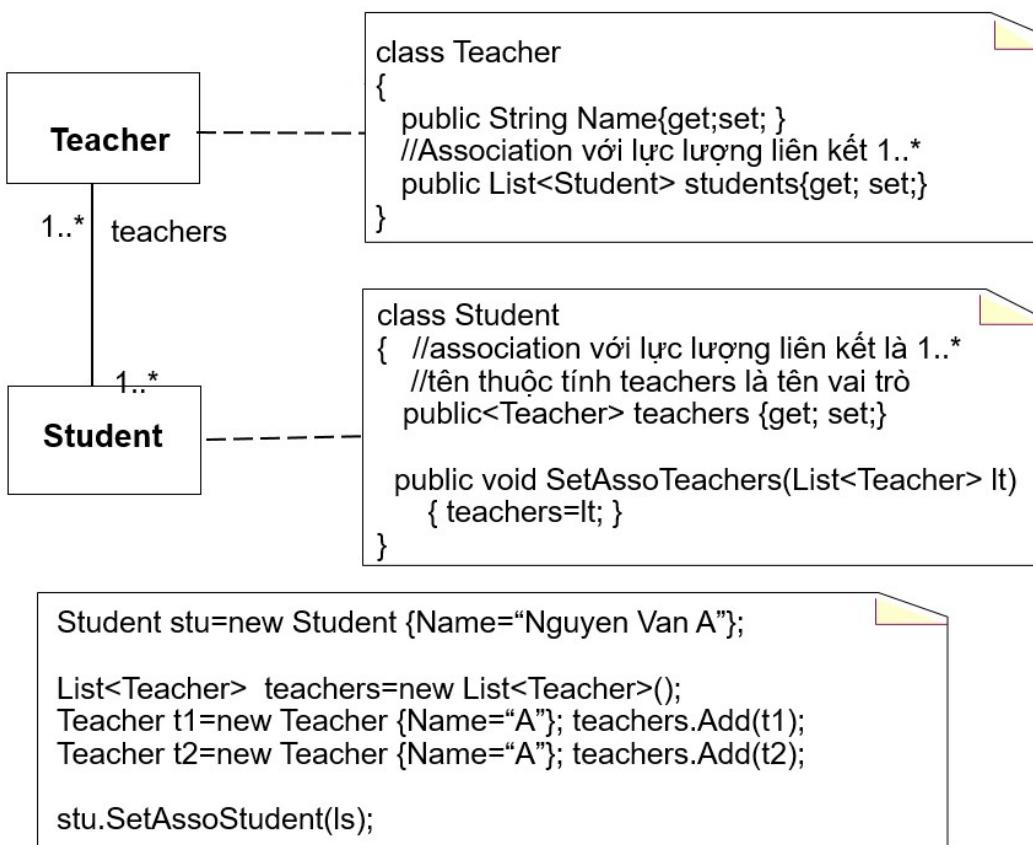


Hình B-5-35 Chuyển lớp lồng nhau sang mã

B.1.2. Ánh xạ quan hệ trong biểu đồ lớp

Quan hệ Association

Quan hệ liên kết (association) là mối quan hệ mà về mặt khái niệm có nghĩa là hai thành phần được liên kết với nhau. Loại quan hệ này được gọi là quan hệ sử dụng, trong đó có một thể hiện của lớp này sử dụng thể hiện của lớp khác. Nhưng đặc điểm chính là thời gian tồn tại của hai thể hiện độc lập với nhau và không có quyền sở hữu giữa hai lớp. Ví dụ quan hệ association giữa Student và Teacher, một Student học nhiều Teacher và một Teacher dạy nhiều Student. Hình B-4 là ánh xạ quan hệ association sang mã lệnh.



Hình B-5-36 Ánh xạ quan hệ Association hai chiều

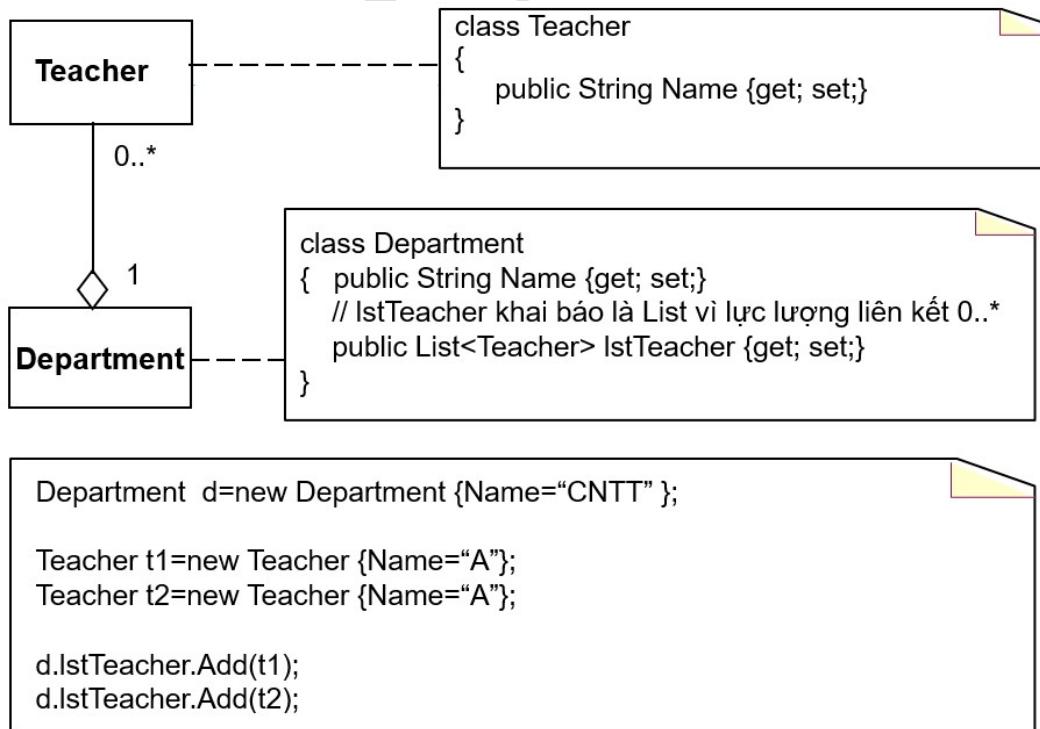
Trong ví dụ này, chúng ta có một thể hiện Student duy nhất *stu*. Student *stu* này có thể có nhiều thể hiện Teacher (*t1*, *t2*) liên kết với nó. Cả hai *t1* và *t2* được tạo bên ngoài lớp Student. Vì vậy thời gian tồn tại của *t1* và *t2* độc lập với thời gian tồn tại của thể hiện Student *stu*, khi *stu* bị

loại bỏ thì $t1$ và $t2$ vẫn tồn tại. Ngoài ra đối tượng stu không có quyền sở hữu $t1$ và $t2$ trong liên kết.

Nếu quan hệ Association có hướng từ A sang B thì chỉ có B sẽ được sử dụng khai báo trong A . Lực lượng liên kết chỉ số đối tượng được sử dụng. Ví dụ trong Hình 6-4 lực lượng liên kết giữa Student và Class là $1..*$, vì vậy trong lớp Class khai báo danh sách đối tượng $students$. Để làm rõ vai trò của liên kết, cuối liên kết tên vai trò có thể được thêm vào. Khi đó tên vai trò được sử dụng đặt tên biến khai báo như trong lớp Student khai báo tên biến là tên vai trò.

Mối quan hệ Aggregation

Quan hệ kết tập (aggregation) là dạng đặc biệt của quan hệ association giữa lớp toàn thể (sở hữu) và lớp bộ phận. Khi đó đối tượng lớp bộ phận là một phần của đối tượng lớp toàn thể và có thể tồn tại độc lập với đối tượng toàn thể. Ví dụ mối quan hệ giữa Department và Teacher là mối quan hệ aggregation. Một đối tượng Department có thể có nhiều Teacher được liên kết với nó. Nhưng một đối tượng Teacher chỉ thuộc về một Department ở một thời điểm. Hình B-5 là ánh xạ quan hệ aggregation sang mã lệnh.

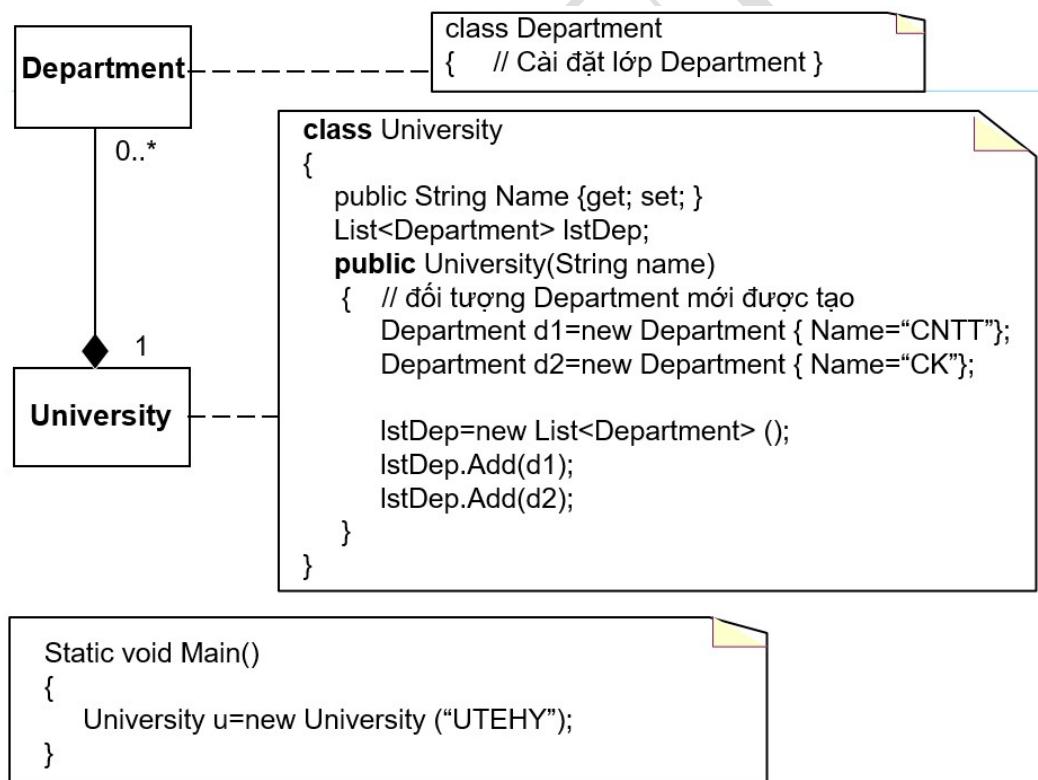


Hình B-5-37 Ánh xạ Aggregation

Đối tượng bộ phận có thể tồn tại độc lập với đối tượng toàn thể, khi đó nếu đối tượng toàn thể được tạo ra thì sẽ được gán cho một tập các đối tượng bộ phận. Ví dụ Hình B-6 là quan hệ aggregation được ánh xạ sang mã lệnh.

Trong ví dụ trên, d là một thẻ hiện của Department và t1, t2 là hai thẻ hiện của Teacher. Đối tượng t1 và t2 được tạo ra ngoài lớp Department nên tồn tại độc lập với đối tượng d. Nên khi d bị loại bỏ thì t1 và t2 vẫn tồn tại. Ngoài ra đối tượng d có thể có nhiều đối tượng Teacher được liên kết, nhưng đối tượng Teacher t1 và t2 chỉ thuộc về Department d mà không thuộc về đối tượng Department nào khác. Nên đối tượng d là chủ sở hữu đối tượng t1, t2.

Mối quan hệ Composition



Hình B-5-38 Ánh xạ quan hệ Composition

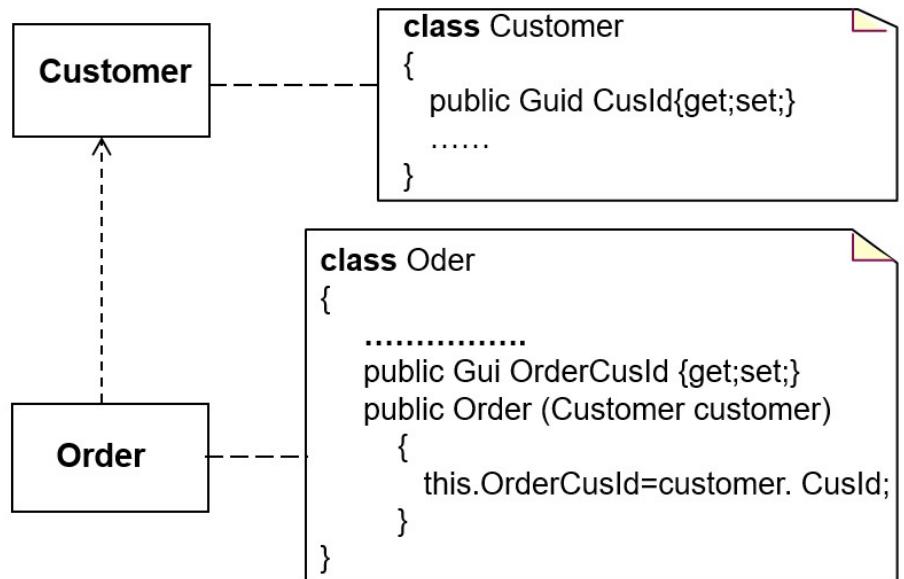
Mối quan hệ cộng hợp (composition) là dạng đặc biệt của Aggregation với đặc điểm sự tồn tại, hủy bỏ của đối tượng thành phần phụ thuộc vào đối tượng toàn thể, nghĩa là đối tượng thành phần được tạo ra khi đối

tượng toàn thể được tạo ra. Ví dụ mối quan hệ giữa Department với University là mối quan hệ composition. Một University có thể có nhiều Departments trong nó. Nhưng một Department chỉ thuộc về một University. Ví dụ Hình 6-6 là quan hệ composition được ánh xạ sang mã lệnh.

Trong ví dụ trên, hai thẻ hiện Department $d1$ và $d2$ chỉ thuộc về một University u duy nhất tại một thời điểm. Nhưng một University có thể có nhiều Departments bộ phận của nó. Ví vậy đối tượng University u là chủ sở hữu của các đối tượng $d1$ và $d2$ đây là đặc điểm của mối quan hệ aggregation. Ngoài ra, thời gian tồn tại của đối tượng $d1$ và $d2$ phụ thuộc vào đối tượng University, vì chúng được tạo ra trong phương thức khởi tạo của đối tượng University nên chỉ khi đối tượng University được tạo ra thì đối tượng Department $d1$ và $d2$ mới được tạo ra. Nên thời gian tồn tại của đối Department phụ thuộc vào thời gian tồn tại của đối tượng University, khi University được tạo ra thì $d1$ và $d2$ được tạo ra, khi đối tượng University bị xóa bỏ thì $d1$ và $d2$ cũng bị xóa bỏ, đây chính là đặc điểm của mối quan hệ composition.

Quan hệ phụ thuộc.

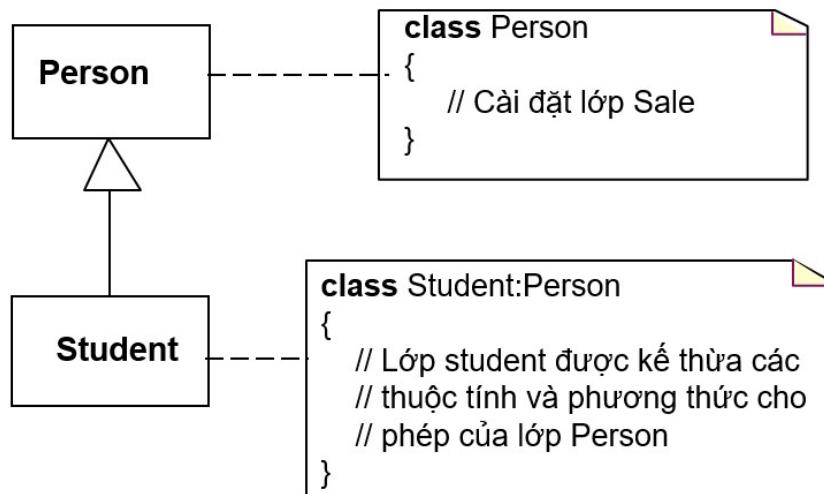
Quan hệ phụ thuộc (dependency) là mối quan hệ giữa hai lớp mà một lớp A phụ thuộc vào một lớp B , nhưng lớp B có thể phụ thuộc hoặc không phụ thuộc vào lớp A . Vì vậy, bất kỳ sự thay đổi lớp B có thể ảnh hưởng đến lớp A mà phụ thuộc vào lớp B . Ví dụ, chúng ta có lớp Custumer và lớp Order. Khi chúng ta cần lưu một Order mới, chúng ta cần lưu lại Order đó tương ứng với một khách hàng. Để làm như vậy, lớp Order cần tham chiếu đến lớp Customer và lưu dữ liệu của nó. Vì vậy, trong trường hợp này, lớp Order phụ thuộc vào lớp Customer. Sau này, nếu có bất kỳ thay đổi nào đối với khách hàng, nó có thể dẫn đến các thay đổi đối với lớp Order. Khi đó đối tượng Customer sẽ là tham số đầu vào của một phương thức nào đó trong Order. Hình B -7 là quan hệ phụ thuộc được ánh xạ sang mã lệnh.



Hình B-5-39 Ánh xạ quan hệ Dependency

Quan hệ tổng quát hóa

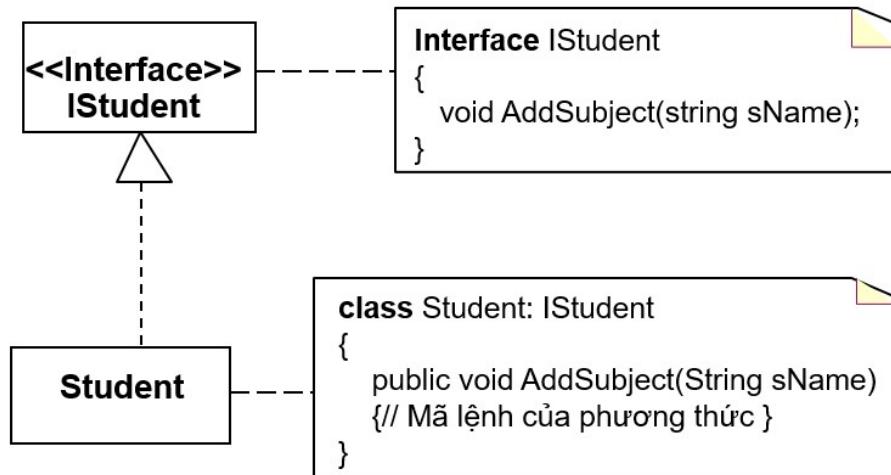
Quan hệ tổng quát hóa (*generalization*) là quan hệ giữa hai lớp cha và con. Trong đó lớp con *A* kế thừa lớp cha *B*, thì lớp con *A* được thừa kế tất cả các phương thức và thuộc tính mà cha cho phép (public và protected). Hình B-8 là quan hệ kế thừa được ánh xạ sang mã lệnh.



Hình B-5-40 Ánh xạ quan hệ kế thừa

Quan hệ hiện thực hóa

Quan hệ hiện thực hóa (*realization*) là quan hệ giữa một lớp và một giao diện (Interface). Lớp *A* hiện thực hóa giao diện *B* thì lớp *A* sẽ triển khai các phương thức trừu tượng khai báo trong giao diện *A*. Ví dụ Hình B-9 là quan hệ hiện thực hóa được ánh xạ sang mã lệnh.



Hình B-5-41 Ánh xạ hiện thực hóa

Đoạn mã lệnh dưới đây là cài đặt hai lớp DonHangDAO và DonHang có mối quan hệ association 1 – 1 có hướng từ DonHangDAO tới DonHang.

```
namespace DTO
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;
    using System.ComponentModel.DataAnnotations.Schema;

    public partial class DonHang
    {
        [Key]
        [StringLength(50)]
        public string MaDonHang { get; set; }

        [StringLength(50)]
        public string MaKhach { get; set; }

        [StringLength(200)]
        public string DiaChiNhan { get; set; }
    }
}
```

```

        [StringLength(12)]
        public string SDTNhan { get; set; }

        public int? TinhTrang { get; set; }

        public double? ThanhTien { get; set; }

        public DateTime? NgayDat { get; set; }
    }
}

namespace DAO
{
    using DTO;
    public class DonHangDAO
    {
        DataHelper dc = new DataHelper();
        public bool LuuDonHang(DonHang d)
        {
            string sql = "INSERT into DonHang values('" +
d.MaDonHang + "','" +
                d.MaKhach + "','" +
                d.DiaChiNhan + "','" +
d.SDTNhan + "','" +
                d.TinhTrang + "," +
                d.ThanhTien + "," +
d.NgayDat + "')";
            string s = dc.ExcuteNonQuery(sql);
            if (s == "")
                return true;
            else
                return false;
        }
    }
}

```

B.2. Ánh xạ các biểu đồ khác sang mã nguồn

B.2.1. Ánh xạ biểu đồ gói

Một gói (package) là một nhóm của các phần tử mô hình. Điều này có nghĩa là một gói có thể chứa các phần tử mô hình thuộc các loại khác nhau bao gồm các gói khác và các lớp để tạo cấu trúc phân cấp. Biểu đồ gói được sử dụng để đơn giản hóa các biểu đồ lớp phức tạp, chúng ta có thể nhóm các lớp vào trong một gói. Một gói khi đó sẽ định nghĩa một không gian tên (namespace) cho nội dung của nó sử dụng cho các

mục đích khác nhau. Ví dụ như gói Controllers trong biểu đồ gói của Hình 5-13 được ánh xạ trong mã lệnh như Hình dưới đây.

```

1  using BLL;
2  using BLL.ServiceInterface;
3  using tool;
4  using System;
5  using System.Web.Mvc;
6  using DTO;
7
8  namespace OnlineFreshShop.Controllers
9  {
10    public class AccountController : Controller
11    {
12
13      Class1 to = new Class1();
14      IAccount acc = new KHACH_HANG_BLL();
15      // GET: Account
16      // Xử lý các sự kiện liên quan đến tài khoản
17      public ActionResult Login()
18      {
19        return View();
20      }
21    }
22  }

```

Hình B-5-42 Ánh xạ package

Trong đó, mỗi một gói tương ứng một thư mục trong dự án và được khai báo một namespace. Các lớp đặt trong một thư mục sẽ được đặt trong khai báo namespace của gói sẽ là những lớp thuộc gói. Trong gói có thể chứa các gói nhỏ hơn nằm ở trong.

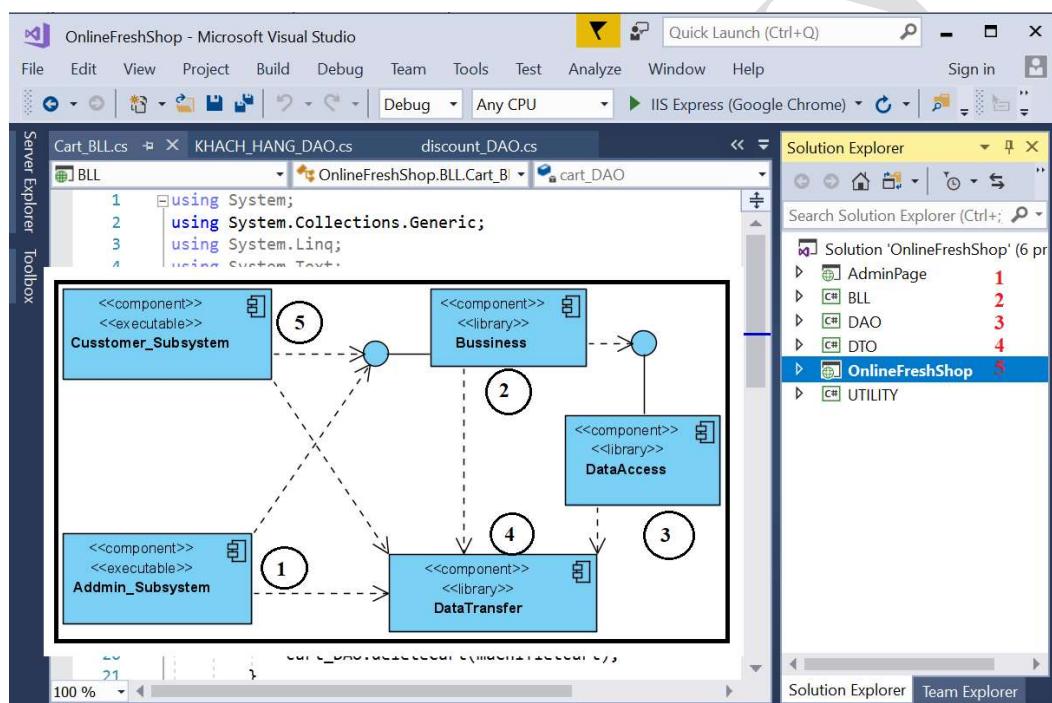
Quan hệ phụ thuộc giữa các gói được ánh xạ sang mối quan hệ khai báo sử dụng giữa các gói. Ví dụ nếu gói Controllers có quan hệ phụ thuộc gói DTO và BLL thì trong các tệp mã nguồn của gói Controllers sẽ khai báo sử dụng hai gói DTO và BLL. Như trong Hình 6-10, dòng 1 và dòng 6 trong hình là câu lệnh khai báo sử dụng gói BLL và gói DTO là hai gói được sử dụng trong gói Controllers. Đây chính là ánh xạ mối quan hệ phụ thuộc giữa gói Controllers với hai gói BLL và DTO.

B.2.2. Ánh xạ biểu đồ thành phần sang mã nguồn

Biểu đồ thành phần (component diagram) được sử dụng để chia nhỏ một hệ thống hướng đối tượng lớn thành các thành phần nhỏ hơn, để dễ dàng quản lý. Nó mô hình hóa chế độ xem vật lý của một hệ thống như tệp thực thi, tệp, thư viện, ... mà nằm bên trong các nút triển khai hệ thống.

Biểu đồ này biểu diễn các mối quan hệ cũng như tổ chức các thành phần trong hệ thống. Nó giúp hình thành một hệ thống thực thi. Thành phần là đơn vị duy nhất của hệ thống, có thể thay thế và thực thi được. Chỉ

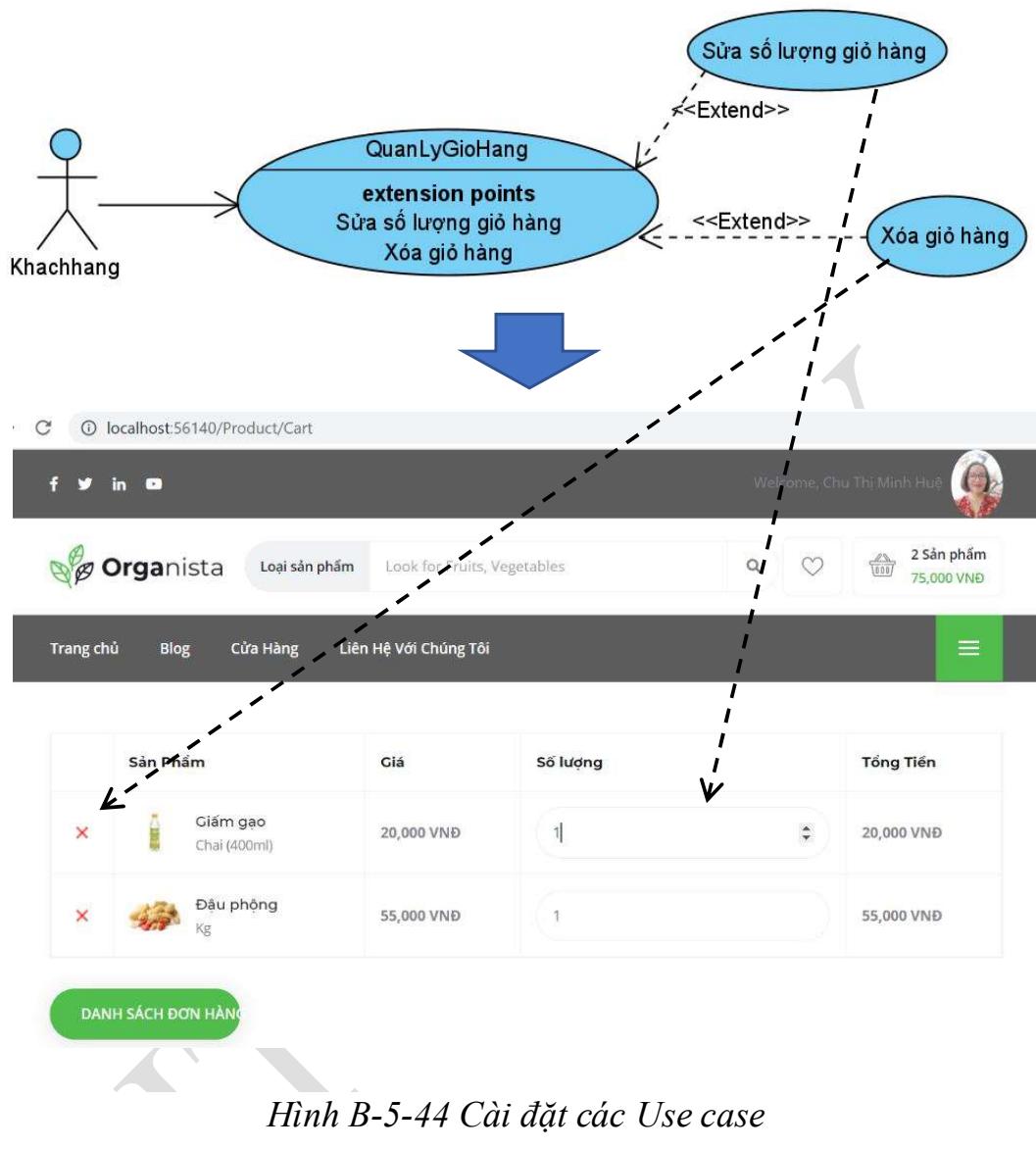
tiết triển khai của một thành phần là ẩn, và nó cần một giao diện để thực thi một chức năng. Một thành phần giống như một hộp đen mà hành vi của nó được giải thích và cung cấp bởi giao diện được yêu cầu. Ví dụ như gói Controllers trong biểu đồ gói của hệ thống Bán hàng online được cài đặt trong mã lệnh như Hình B-11 dưới đây. Trong đó các thành phần <<library>> được cài đặt tương ứng với các dự án thư viện library trong Dot.Net, hai thành phần Cusstonmer_subsystem và Admin_Subsystem được cài đặt tương ứng với hai dự án MVC trong Dot.Net. Ngoài ra trong dự án có thêm một thành phần thư viện chung UTILITY chưa được mô hình hóa trong mô hình.



Hình B-5-43 Ánh xạ biểu đồ thành phần

B.2.3. Ánh xạ ca sử dụng

Mỗi ca sử dụng sẽ được cài đặt thành một chức năng của hệ thống. Như Hình B-12 thể hiện các ca sử dụng trong biểu đồ phân rã của use case Quản lý giờ hàng được cài đặt thành các chức năng tương ứng.



TÀI LIỆU THAM KHẢO

- [1] R. P. a. B. Maxim, software engineer a practitioner's approach, McGraw-Hill Education, 2014.
- [2] N. V. H. Nguyễn Văn Vy, Giáo trình kỹ nghệ phần mềm, Hà Nội: NXB Giáo dục, 2009.
- [3] "tutorialspoint.com," Tutorialspoint, 3 2020. [Online]. Available:
https://www.tutorialspoint.com/software_engineering/software_development_life_cycle.htm.
- [4] C. L. M. David A. Marca, Structured Analysis and Design Techniques, McGraw-Hill, 1987.
- [5] G. Booch, Object-oriented Analysis and Design with Applications, Redwood City, Calif. : Benjamin/Cummings, 1994.
- [6] J. Rumbaugh, Object Oriented Modeling and Design, Prentice Hall , 1991.
- [7] I. Jacobson, Object Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley Professional, 1992.
- [8] D. Coleman, Object-oriented Development: The Fusion Method, Prentice Hall, 1994.
- [9] P. Y. E. Coad, Object-Oriented Design, Prentice-Hall, Englewood Cliffs, 1991.
- [10] A. Cockburn, Writing Effective Use Cases, Boston: Addison-Wesley Professional, October 2000.

- [11] Bộ bài giảng "Object oriented Analysis and Design" của IBM (DEV475)[online].
- [12] B. Unhelkar, Software Engineering With UML, Auerbach Publications, 2017.
- [13] I. Sommerville, Software Engineering, Courier Westford in the United States of America, 2016.
- [14] "www.tutorialspoint.com," Tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/software_engineering/index.htm.
- [15] Đ. V. Đức, Phân tích và thiết kế hướng đối tượng bằng UML, NXB Giáo dục, 2002.
- [16] "visual-paradigm," [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>.
- [17] Rational, "Rational Unified Process Best Practices for Software Development Teams," 1998.
- [18] I. Jacobson, "Object-oriented Development in an Industrial Environment," in *Object-oriented Programming Systems, Languages and Applications (OOPSLA)*, 1987.
- [19] Trương Ninh Thuận, Đặng Đức Hạnh, Giáo trình phân tích và thiết kế hướng đối tượng, Hà Nội: NXB ĐH Quốc gia Hà Nội, 2016.