

BÀI SỐ 1

TỔNG QUAN VỀ THIẾT KẾ HỆ THỐNG HƯỚNG ĐỐI TƯỢNG

MỤC TIÊU

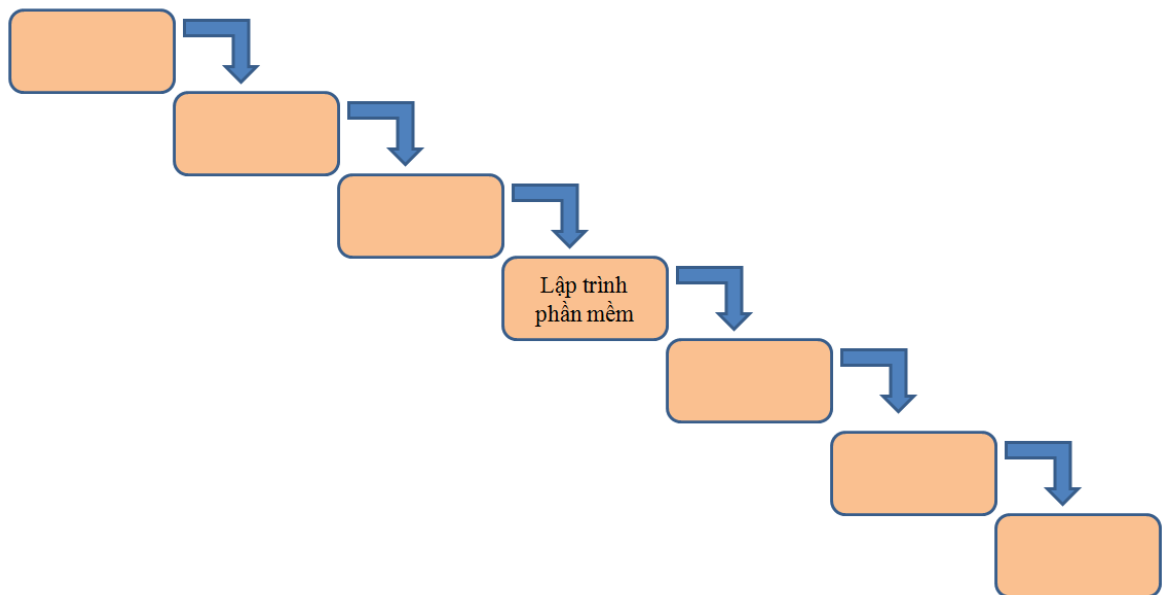
Sau khi học xong bài này, sinh viên có khả năng:

- Trình bày được các khái niệm cơ bản, ý nghĩa và tầm quan trọng của môn học trong thiết kế phần mềm;
- So sánh được ưu điểm, nhược điểm của phương pháp thiết kế hướng chức năng và phương pháp thiết kế hướng đối tượng;
- Mô tả được các bước phát triển hệ thống trong mối quan hệ với hoạt động lập trình phần mềm. Phân biệt thuộc tính và phương thức, đồng thời hình thành thói quen tư duy xây dựng thuộc tính và phương thức cho một đối tượng cụ thể
- Xác định được vai trò và vị trí của người phân tích thiết kế hệ thống và quan hệ hợp tác trong một dự án ngoài đời thực.

YÊU CẦU

Lý thuyết (Trả lời các câu hỏi)

1. Tại sao cần phải thiết kế và phân tích hệ thống trước khi bắt tay vào xây dựng?
2. Lớp và đối tượng khác nhau như thế nào?
3. Trình bày ưu điểm của phân tích thiết kế hướng đối tượng so với thiết kế hướng chức năng?
4. Mô tả 3 mô hình cơ bản được ứng dụng để phát triển hệ thống hiện nay và ưu điểm của chúng.
5. Điền các bước phát triển hệ thống phần mềm còn thiếu



6. Khi phân tích hệ thống quản lý trường học theo hướng đối tượng, hãy liệt kê 5 lớp đối tượng quan trọng có thể xuất hiện trong hệ thống?

7. Khi phân tích hệ thống quản lý trường học, với lớp đối tượng sinh viên, hãy liệt kê 5 thuộc tính và 5 phương thức quan trọng có thể có?
8. Để xây dựng một hệ thống phần mềm, ví dụ một website bán hàng, theo em cần có những nhân lực nào tham gia và nhiệm vụ của họ là gì?

BÀI SỐ 2

LÀM QUEN VỚI NGÔN NGỮ UML

MỤC TIÊU.

- Trình bày được những khái niệm cơ bản, lịch sử hình thành và vai trò của UML trong thiết kế hướng đối tượng;
- Mô tả được các khung nhìn và các loại biểu đồ trong các khung nhìn đó;
- Thực hiện được các thao tác cơ bản trên phần mềm Rational Rose để thiết kế một số biểu đồ theo mẫu.

YÊU CẦU.

Phần 1. Lý thuyết (Trả lời các câu hỏi)

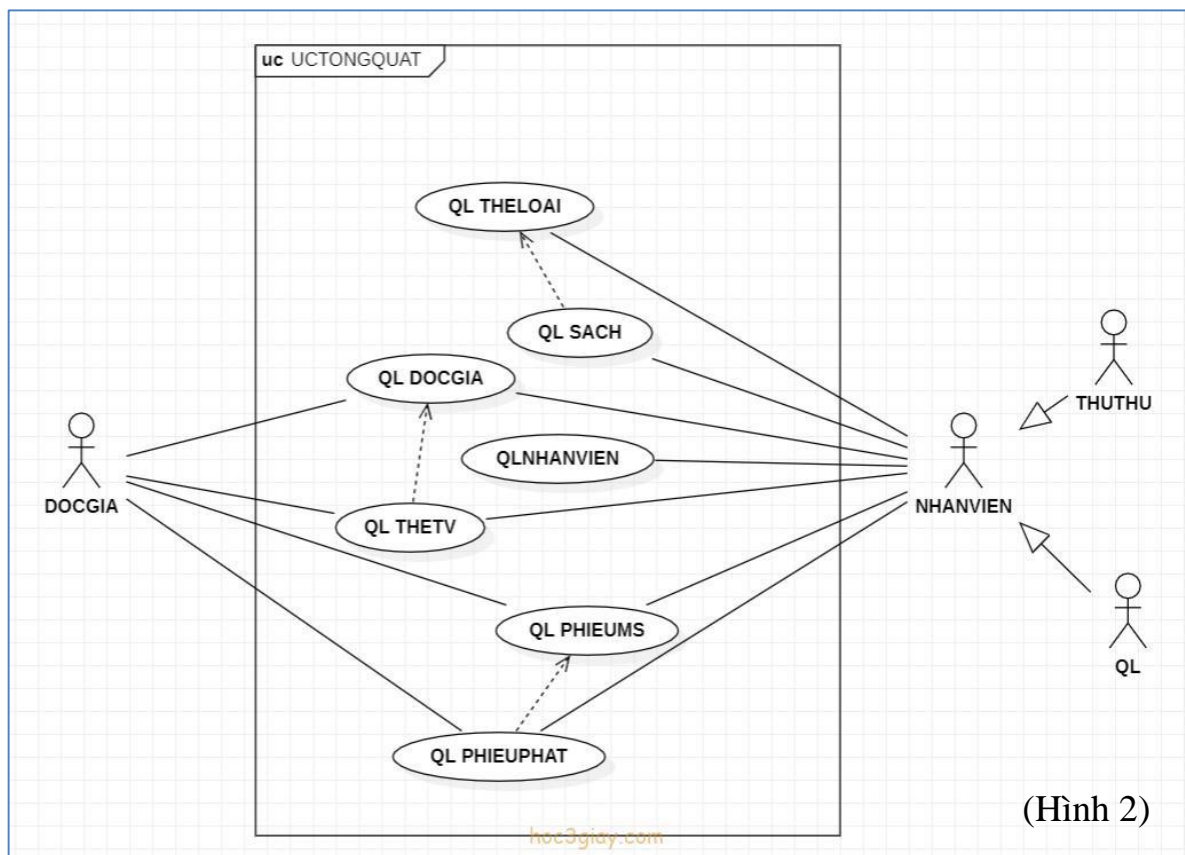
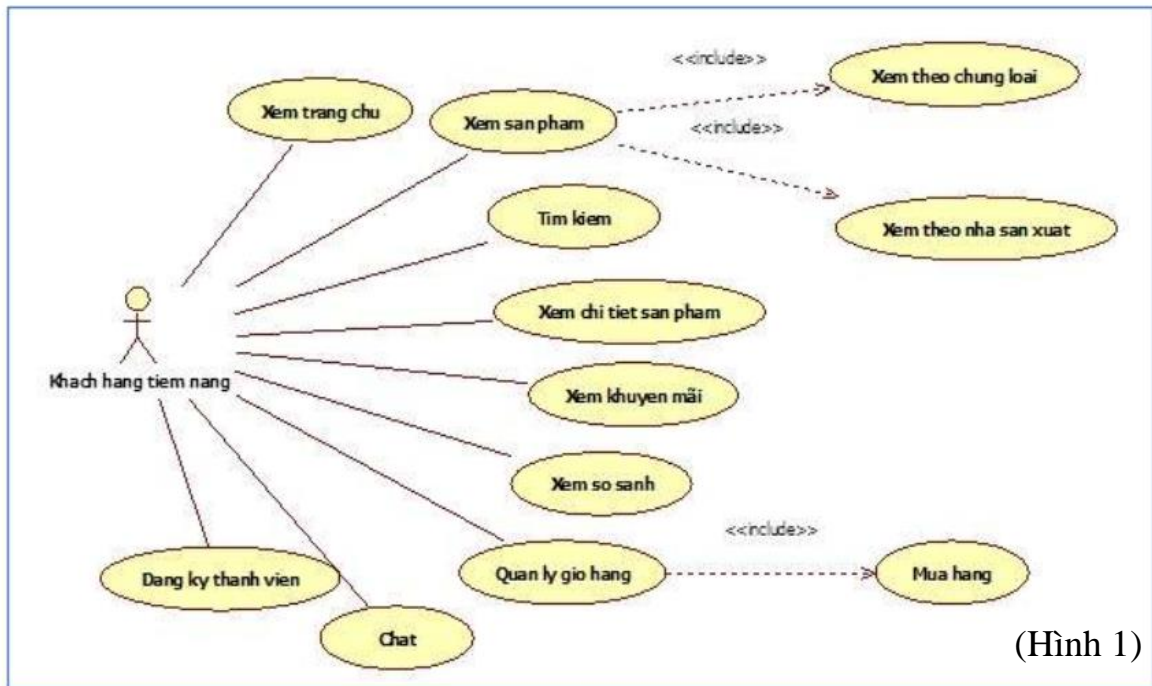
1. Trình bày lịch sử ra đời của UML và vai trò của UML trong mô hình hóa hệ thống? UML ra đời từ các ngôn ngữ và phương pháp mô hình hóa nào?
2. Khung nhìn (View) là gì? UML bao gồm các khung nhìn nào? Nêu các loại biểu đồ trong từng khung nhìn. Nêu tóm tắt nhiệm vụ của các biểu đồ đó?
3. Trình bày vai trò của Rational Rose trong quá trình phát triển phần mềm thống nhất. Hãy liệt kê các thành phần chủ yếu của ngôn ngữ UML
4. Chọn_ từ danh sách dưới đây những thuật ngữ thích hợp để điền vào chỗ trống.

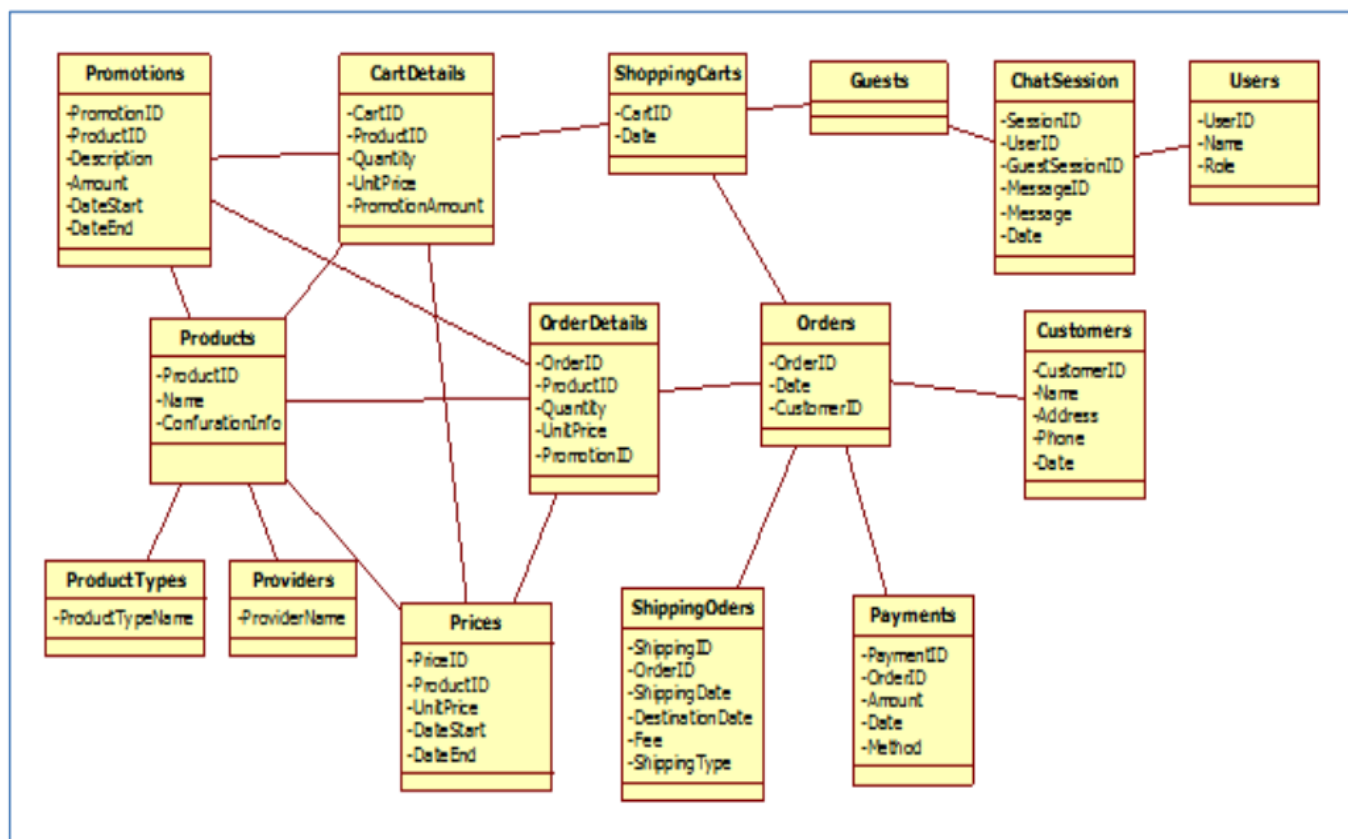
*UML là ngôn ngữ mô hình hóa, trước hết nó mô tả ...**(1)**..., ngữ nghĩa các định nghĩa trực quan tất cả các thành phần của**(2)**..... UML được sử dụng để hiển thị, đặc tả, tổ chức, xây dựng và ...**(3)**..... của....**(4)**..., đặc biệt là phân tích, thiết kế dưới dạng báo cáo, biểu đồ, bản mẫu hay các trang web,... UML là ngôn ngữ ...**(5)**...hóa độc lập với công nghệ phát triển ...**(6)**...*

- a.* quá trình phát triển phần mềm hướng đối tượng.
 - b.* làm tài liệu
 - c.* của các hệ thống phần mềm
 - d.* quá trình xử lý
 - e.* mô hình
 - f.* phần mềm
5. UML có công cụ nào phục vụ cho giai đoạn thử nghiệm đơn vị (Unit Testing)? Và có công cụ nào phục vụ cho giai đoạn thử nghiệm hệ thống (System Testing)?

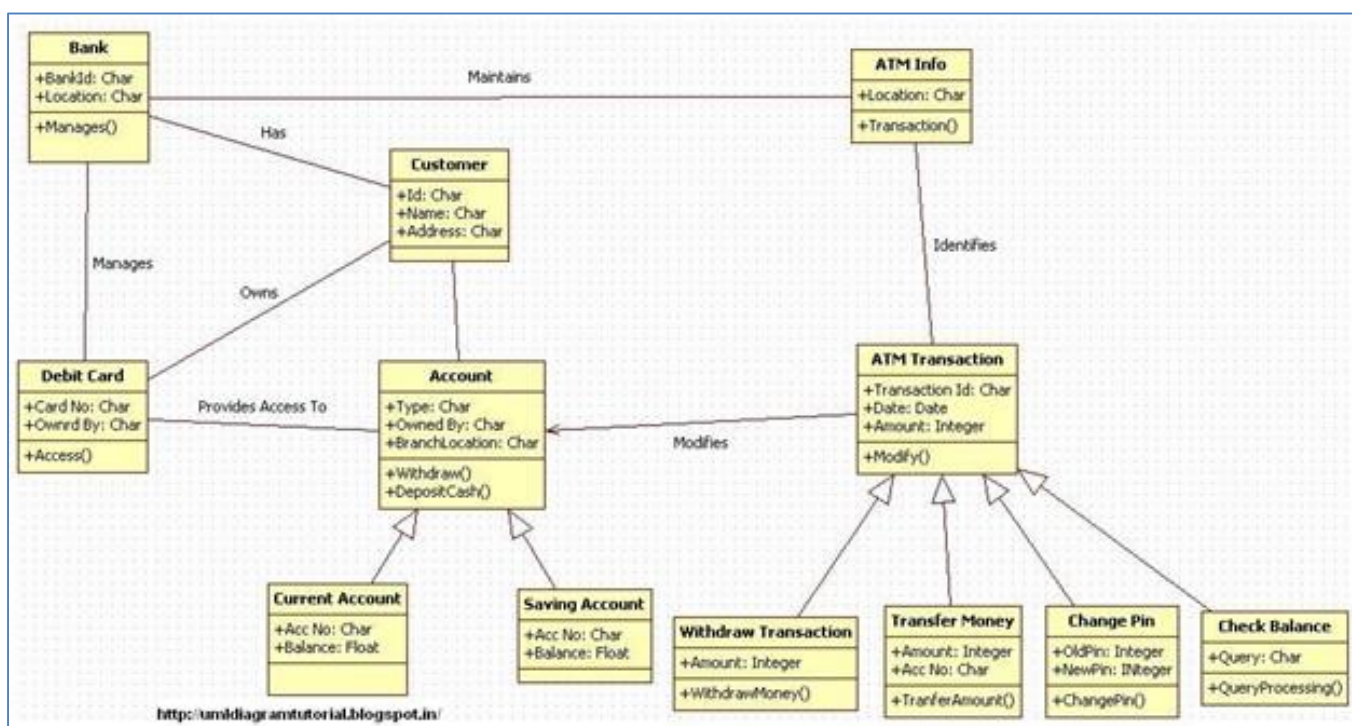
Phần 2: Thực hành (Thực hiện trên Rational Rose)

1. Khởi động Rational Rose
2. Tạo tệp mới
3. Thiết kế các biểu đồ mẫu sau:





(Hình 3)



(Hình 4)

BÀI SỐ 3

XÂY DỰNG BIỂU ĐỒ CA SỬ DỤNG (Use Case Diagram)

MỤC TIÊU

- Trình bày được các thành phần trong biểu đồ Use case
- Xác định được các Actor trong hệ thống
- Xác định được các Use case trong hệ thống
- Xác định được các mối quan hệ giữa các Use case
- Sử dụng được phần mềm Rational Rose để biểu diễn biểu đồ Use case
- Rèn luyện tính cẩn thận, chính xác, tư duy logic, sáng tạo của người học.

YÊU CẦU.

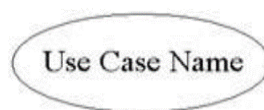
Phần 1. Lý thuyết

1. Tập ký hiệu cho biểu đồ Use case

- *Actor* (tác nhân): Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống. Actor được biểu diễn như sau:



- *Use Case* (ca sử dụng): là chức năng mà các Actor sẽ sử dụng. Nó được ký hiệu như sau:



- *Relationship* (mối quan hệ): được sử dụng để kết nối giữa các đối tượng với. Có các kiểu quan hệ:

- + Quan hệ *Association*: thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau.



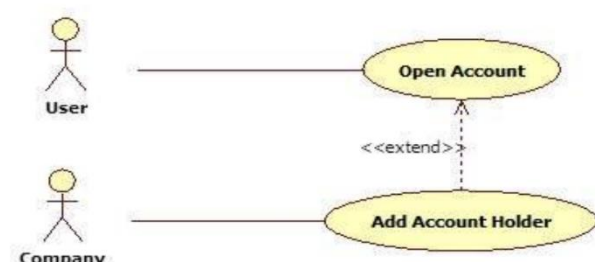
- + Quan hệ *Generalization*: được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau



- + Quan hệ *Include*: là quan hệ giữa các Use Case với nhau, nó mô tả việc một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa) hoặc thể hiện sự dùng lại.



- + Quan hệ *Extend*: dùng để mô tả quan hệ giữa 2 Use Case, được sử dụng khi có một



Use Case được tạo ra để bổ sung chức năng cho một Use Case có sẵn và được sử dụng trong một điều kiện nhất định nào đó.

2. Các bước xây dựng Use Case Diagram

Bước 1: Tìm các Actor, Trả lời các câu hỏi sau để xác định Actor cho hệ thống:

- Ai sử dụng hệ thống này?
- Hệ thống nào tương tác với hệ thống này?

Bước 2: Tìm các Use case, Trả lời câu hỏi: các Actor sử dụng chức năng gì trong hệ thống?

Bước 3: Xác định các quan hệ Phân tích và các loại quan hệ giữa các Actor và Use Case, giữa các Actor với nhau, giữa các Use Case với nhau sau đó nối chúng lại chúng ta sẽ được bản vẽ Use Case.

Phần 2. Thực hành

Bài tập 1.

Xây dựng sơ đồ Use case cho một hệ thống thương mại điện tử (E- Commerce) như sau: “Một công ty chuyên kinh doanh về thực phẩm chế biến từ Gà và đã có một lượng khách hàng nhất định. Để mở rộng hoạt động kinh doanh của mình, công ty mong muốn xây dựng một hệ thống thương mại điện tử nhằm mở rộng phạm vi kinh doanh trên mạng Internet.

Hệ thống mới phải đảm bảo cho khách hàng viếng thăm Website để dàng lựa chọn các sản phẩm, xem các khuyến mãi cũng như mua hàng. Việc thanh toán có thể được thực hiện qua mạng hoặc thanh toán trực tiếp tại cửa hàng.

Khách hàng có thể nhận hàng tại cửa hàng hoặc sử dụng dịch vụ chuyển hàng có phí của công ty.

Ngoài ra, hệ thống cũng cần có phân hệ để đảm bảo cho công ty quản lý các hoạt động kinh doanh như quản lý đơn đặt hàng, tình trạng giao hàng, thanh toán ...”

Thông tin chi tiết các chức năng có thể tham khảo thêm tại các Website bán hàng. Hãy xây dựng hệ thống use case trên.

Hướng dẫn thực hiện

Bước 1: Thu thập kiến thức liên quan đến hệ thống sẽ xây dựng Trước hết, để phân tích hệ thống trên ta phải có kiến thức về hệ thống thương mại điện tử, chúng ta có thể tìm hiểu thông qua các nguồn sau:

- Xem các trang Web bán hàng qua mạng như amazon, lazada.vn, bkc.vn v.v..
- Xem các hệ thống mẫu về thương mại điện tử nguồn mở như Magento, OpenCart, Spree Commerce v.v...
- Đọc sách, báo về eCommerce
- Hỏi những người chuyên về lĩnh vực này (hỏi chuyên gia)

Bước 2: Xác định các Actor, Hãy trả lời cho câu hỏi “Ai sử dụng hệ thống này?” Xem xét Website chúng ta nhận thấy:

- Những người muốn mua hàng vào website để xem thông tin. Những người này là **Khách hàng tiềm năng** (Guest).
- Những người đã đặt hàng vào kiểm tra đơn hàng, thanh toán v.v.. gọi là **Khách hàng** (Customer).
- Về phía đơn vị bán hàng, có những người sau đây tham gia vào hệ thống:
 - + **Người quản lý bán hàng**: quyết định nhập hàng, giá bán, quản lý tồn kho, doanh thu, chính sách khuyến mãi.
 - + **Người bán hàng**: Tư vấn cho khách hàng, theo dõi đơn hàng, thu tiền, theo dõi chuyển hàng cho khách.
 - + **Quản lý kho**: xuất, nhập hàng, quản lý tồn kho o Quản trị hệ thống: Tạo người dùng, Phân quyền, Tạo cửa hàng

Tiếp theo chúng ta trả lời câu hỏi “*Hệ thống nào tương tác với hệ thống này?*” Giả sử ở đây, chúng ta sử dụng dịch vụ của Ngân Lượng để thanh toán trực tuyến và gọi nó là “**Cổng thanh toán**” thì ta có thêm một Actor tương tác với hệ thống.

Như vậy, chúng ta đã có các Actor của hệ thống gồm: Khách hàng tiềm năng, khách hàng, Người bán hàng, Quản lý Kho, Quản trị hệ thống, Cổng thanh toán.

Chúng ta cần khảo sát và phân tích thêm cũng như hỏi trực tiếp khách hàng để xác định đầy đủ các Actor cho hệ thống.

Bước 3: Xác định Use Case, Ta cần trả lời câu hỏi “*Actor sử dụng chức năng gì trên hệ thống?*”. Trước tiên, xem xét với Actor “**Khách hàng tiềm năng**” trên trang bkc.vn để xem họ sử dụng chức năng nào?

- Xem trang chủ
- Xem các sản phẩm theo:
 - + Theo chủng loại
 - + Nhà sản xuất
 - + Tìm kiếm theo văn bản gõ vào
- Xem chi tiết sản phẩm được chọn
- Xem khuyến mãi
- Xem so sánh
- Mua hàng
- Quản lý giỏ hàng
- Chat với người bán hàng
- Đăng ký tài khoản để trở thành khách hàng

Tiếp theo, xem xét Actor “**Khách hàng**” và nhận thấy họ sử dụng chức năng:

- Đăng nhập
- Xem đơn hàng
- Thanh toán

Tiếp theo, xem xét Actor “**Người bán hàng**” và họ có thể sử dụng các chức năng:

- Đăng nhập

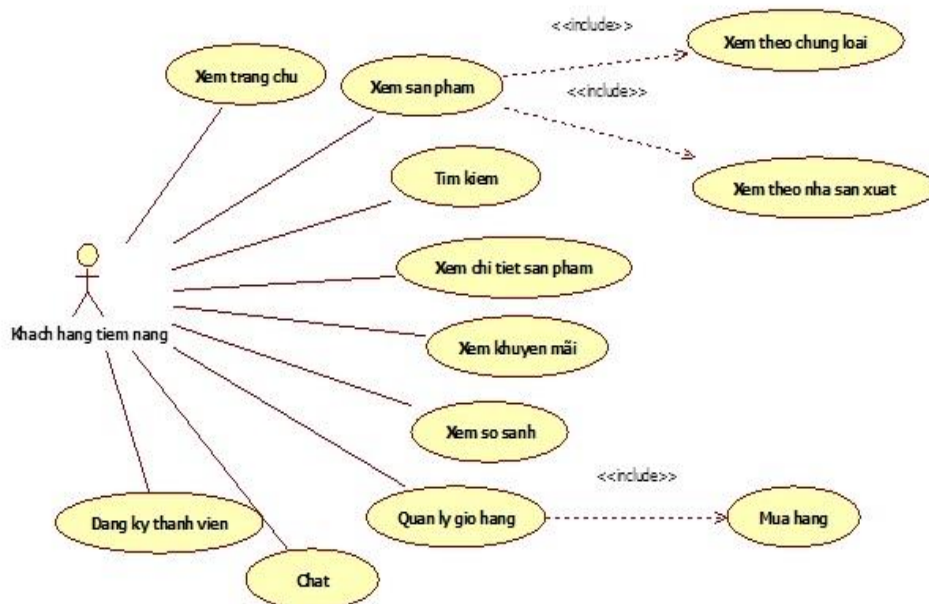
- Chat với khách hàng
- Theo dõi đơn hàng
- Thu tiền

Theo dõi chuyển hàng Tương tự như vậy ta xác định chức năng cho các Actor còn lại.

Bước 4: Vẽ bản vẽ Use Case

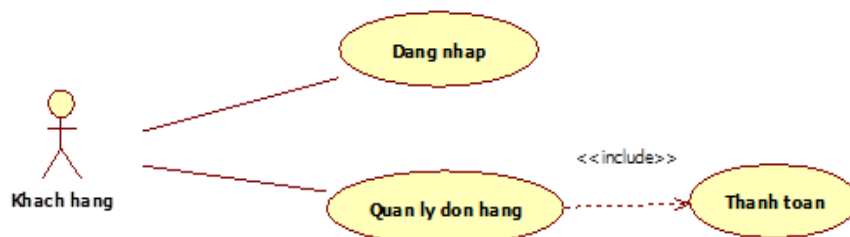
Trước hết chúng ta xem xét và phân tích các chức năng của “**Khách hàng tiềm năng**” chúng ta nhận thấy.

- Chức năng xem sản phẩm có 2 cách là chọn loại sản phẩm, nhà sản xuất để xem và gõ vào ô tìm kiếm. Nên chúng ta tách ra làm 2 là Xem sản phẩm và Tìm kiếm.
- Chức năng mua hàng, thực chất là thêm vào giỏ hàng nên có thể xem là chức năng con của quản lý giỏ hàng.
- Đặt lại tên cho gọn và xác định các mối quan hệ của chúng, chúng ta có thể vẽ Use Case Diagram cho Actor này như sau:



(Bản vẽ Use Case cho Actor “Khách hàng tiềm năng”)

Tiếp theo, chúng ta xem xét các chức năng cho Actor “**Khách hàng**” và nhận thấy chức năng “**Thanh toán**” thường thực hiện cho từng đơn hàng cụ thể nên có thể nó là chức năng con của “**Quản lý đơn hàng**”. Ngoài ra, các chức năng Actor này sử dụng không giao với Actor “Khách hàng tiềm năng” nên nó được biểu diễn như sau:



Tiếp tục xem xét Actor “**Người bán hàng**” chúng ta nhận thấy:

- Chức năng “**Thu tiền**” thực tế là thanh toán trực tiếp tại quầy cho từng đơn hàng và chức năng “**Theo dõi chuyển hàng**” được thực hiện trên từng đơn hàng nên nó có thể là chức năng con của “**Quản lý đơn hàng**”.
- Chức năng “**Quản lý đơn hàng**” ở đây quản lý cho nhiều khách hàng nên sẽ khác với chức năng “**Quản lý đơn hàng**” của Actor “Khách hàng” nên để phân biệt chúng ta sửa chức năng “**Quản lý đơn hàng**” của Actor “Khách hàng” thành “**Quản lý đơn hàng cá nhân**”
- Chức năng “**Đăng nhập**” có thể dùng chung với Actor “Khách hàng”, chức năng Chat dùng chung với Actor “Khách hàng tiềm năng”

Về chúng chung với nhau chúng ta được bản vẽ như sau:



(Bản vẽ Use Case khi bổ sung các chức năng cho “Khách hàng tiềm năng”, “Khách hàng” và “Người bán hàng”)

Yêu cầu: hãy tiếp tục hoàn tất các chức năng cho các Actor còn lại để có một bản vẽ hoàn chỉnh về Use Case cho hệ thống.

Bài tập 2:

HỆ THỐNG QUẢN LÝ THƯ VIỆN cho trường Đại học gồm các hoạt động **Quản lý thông tin sách, Quản lý thông tin độc giả, Quản lý hoạt động mượn trả sách** được mô tả như sau:

- Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.

- Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngày nhập). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.

- Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).

- Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).

- Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.

- Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.

- Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.

- Người dùng hệ thống phải đăng nhập trước khi thực hiện.

Dựa vào bản mô tả hệ thống quản lý thư viện, hãy thực hiện các yêu cầu sau:

1. Xác định các Actor và Use case của hệ thống quản lý thư viện.
2. Xây dựng biểu đồ Use case mức tổng quát cho hệ thống quản lý thư viện.
3. Xây dựng các biểu đồ Use case phân rã cho các Use case tổng quát (nếu có).
4. Xây dựng kịch bản cho Use case.

Hướng dẫn thực hiện:

Bước 1. Xác định các Actor.

Để xác định được các Actor của hệ thống cần trả lời câu hỏi ai, hệ thống nào...tác động trực tiếp, gián tiếp vào hệ thống quản lý thư viện. Từ mô tả trên ta tìm thấy 2 Actor chính trong hệ thống là: **Actor thủ thư** và **Actor người quản trị**. Và 2 Actor nhận kết quả từ hệ thống là **Actor Độc giả** và **Actor Quản lý thư viện**.

Bước 2. Xác định các Use case.

Từ các Actor chính xác định các Use Case tổng quát Hệ thống có 2 Actor chính là thủ thư và người quản trị, trong đó người quản trị có quyền thực hiện các chức năng trong hệ thống như là một thủ thư, ngoài ra người quản trị có quyền quản lý người dùng.

- **Các Use case mà Thủ thư sử dụng gồm:** Quản lý độc giả, quản lý sách, quản lý mượn trả sách, thống kê báo cáo, Đăng Nhập.
- **Các Use case mà người quản trị sử dụng gồm:** Các Use case như của thủ thư và Use case Quản lý người dùng.

Bước 3. Xác định các mối quan hệ

- Hai Actor chính của hệ thống là Actor **Thủ thư** và Actor **QuanTri** là các Actor sử dụng trực tiếp các Use case của hệ thống, nên các Actor có mối quan hệ Association với các Actor mà Use case sử dụng.

- Actor **Quản trị** có vai trò như một Thủ thư, ngoài ra có quyền thực hiện Use case **QuanLyNguoiDung**. Nên Actor QuanTri có thể thực hiện tất cả các Use case mà Thủ thư thực hiện, nên Actor QuanTri và Actor ThủThư có mối quan hệ kế thừa.

- Khi thực hiện tất cả các Use case người dùng đều phải Đăng nhập, vì vậy Use case đăng nhập là phần chung được tách ra từ các Use case khác. Vì vậy các Use case này đều có mối quan hệ include với Use case đăng nhập.

- Hai Actor **Bạn đọc** và **Quản lý thư viện** là Actor không trực tiếp mà nhận kết quả từ hệ thống, nên có mối quan hệ phụ thuộc vào các Use case mà tác nhân nhận kết quả

Thực hiện Vẽ biểu đồ Use case tổng quát.

***. Xây dựng kịch bản cho Use case**

Kịch bản Use case là một chuỗi tương tác giữa Actor và phản ứng hệ thống để thực hiện Use case, khi đó tùy thuộc vào hành động tác nhân tác động lên hệ thống mà có thể dẫn đến. Các phản ứng khác nhau với cùng một hành động tùy thuộc vào dữ liệu đầu vào và các yếu tố khác. Nên một Use case có thể có luồng sự kiện chính và các luồng sự kiện phụ.

Khi xây dựng kịch bản chúng ta xây dựng kịch bản Use case tổng quát trước và kịch bản của Use case phân rẽ sau:

Kịch bản Use case Quản lý Sách

Tên use case	QuanLySach
Tên Actor	Thủ thư
Mức	1
Tiền điều kiện	Thủ thư phải đăng nhập hệ thống
Đảm bảo tối thiểu	
Đảm bảo thành công	
Kích hoạt	Thủ thư yêu cầu chức năng QL sách
Hành động tác nhân	Phản ứng hệ thống
1.Thủ thư yêu cầu chức năng	1.1. Hệ thống lấy về thông tin các chuyên ngành 1.2. Lấy về thông tin ngày giờ hệ thống. 1.3. Hệ thống chọn chuyên ngành mặc định. 1.4. Hệ thống lấy về các đầu sách thuộc chuyên ngành, hiển thị lên giao diện
2. Thực hiện các chức năng theo yêu cầu	

Kịch bản Use case Thêm đầu sách mới:

Tên use case	QuanLySach
Tên Actor	Thủ thư
Mức	1
Tiền điều kiện	Thủ thư phải đăng nhập hệ thống
Đảm bảo tối thiểu	
Đảm bảo thành công	
Kích hoạt	Thủ thư yêu cầu chức năng QL sách
Hành động tác nhân	Phản ứng hệ thống
1. Thủ thư yêu cầu chức năng nhập đầu sách	
2. Thủ thư chọn chuyên ngành	2.1. Hệ thống lấy về các đầu sách thuộc chuyên ngành
3. Thủ thư nhập thông tin về đầu sách	2.1. Hệ thống hợp lệ hóa thông tin
4. Thủ thư yêu cầu chức năng nhập đầu sách	2.2. Hệ thống sinh mã tự động cho đầu sách (Theo nguyên tắc 1), sinh mã tự động cho các bản sao theo nguyên tắc 2. 2.3. Hệ thống cập nhật đầu sách, cập nhật các bản sao của đầu sách.

- Nguyên tắc 1: Mã đầu sách = Mã chuyên ngành.Số thứ tự. Ví dụ đầu sách thứ 5 của chuyên ngành ck sẽ là ck.5

- Nguyên tắc 2: Mã sách = mã đầu sách.số thứ tự. Ví dụ đầu sách ck.5, có 2 cuốn thì mã sách lần lượt của 2 cuốn là ck.5.1, ck.5.2

Kịch bản Use case sửa sách

Tên use case	Sửa sách
Tên Actor	Thủ thư
Mức	2
Tiền điều kiện	Thủ thư phải đăng nhập hệ thống
Đảm bảo tối thiểu	
Đảm bảo thành công	
Kích hoạt	Thủ thư yêu cầu chức năng QL sách
Hành động tác nhân	Phản ứng hệ thống
1. Chọn, nhập đầu sách cần sửa	1.1. Hiện thị các thuộc tính của đầu sách lên các điều khiển 1.2. Vô hiệu hóa chức năng nhập đầu sách
2. Sửa thông tin đầu sách	
3. Yêu cầu sửa thông tin đầu sách	3.1. Sửa thông tin đầu sách 3.2. Nếu số lượng mới > số lượng cũ chuyển sang luồng phụ 1. Nếu số lượng mới < số lượng cũ chuyển sang luồng phụ 2
Ngoại lệ	
Luồng phụ 1	
	3.2.1. Bổ sung thêm các bản sao chênh lệch
Luồng phụ 2	
	3.2.1. Xóa đi các bản sao chênh lệch

Vẽ biểu đồ phân rã cho các Use case tổng quát còn lại.

Xây dựng kịch bản cho các Use case đó

BÀI SỐ 4

XÂY DỰNG BIỂU ĐỒ LỚP (Class Diagram)

MỤC TIÊU:

- Trình bày được các thành phần của biểu đồ lớp và các bước xây dựng biểu đồ lớp (Class Diagram);
- Phân biệt và xác định được các lớp cơ bản, các phương thức và thuộc tính của các lớp cơ bản đó;
- Sử dụng thành thạo phần mềm để biểu diễn biểu đồ lớp của hệ thống;
- Xây dựng được biểu đồ lớp thực thể; Xây dựng được biểu đồ lớp cho từng kịch bản Use case;
- Rèn luyện tính cẩn thận, chính xác, tư duy logic, sáng tạo của người học.

YÊU CẦU

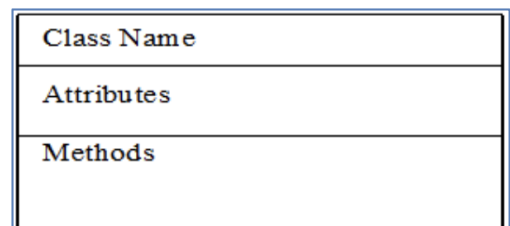
Phần 1. Lý thuyết

1. Các thành phần trong biểu đồ lớp

- *Classes* (Các lớp): Class là thành phần chính của bản vẽ Class Diagram. Class mô tả về một nhóm đối tượng có cùng tính chất, hành động trong hệ thống. Ví dụ mô tả về khách hàng “Customer”. Class được mô tả gồm tên Class, thuộc tính và phương thức

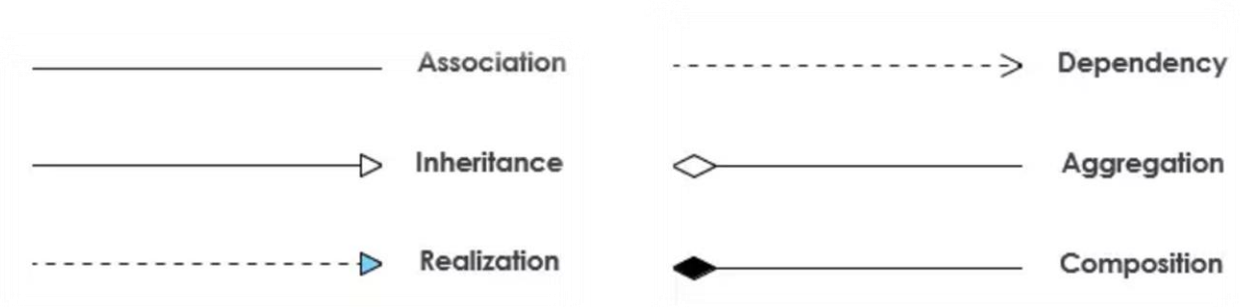
Trong đó:

- + *Class Name*: là tên của lớp.
- + *Attributes* (thuộc tính): mô tả tính chất của các đối tượng. Ví dụ như khách hàng có Mã khách hàng, Tên khách hàng, Địa chỉ, Ngày sinh v.v...



- + *Method* (Phương thức): chỉ các hành động mà đối tượng này có thể thực hiện trong hệ thống. Nó thể hiện hành vi của các đối tượng do lớp này tạo ra

- *Relationship* (Quan hệ): Relationship thể hiện mối quan hệ giữa các Class với nhau. các quan hệ thường sử dụng như sau:



- + **Association**: Association là quan hệ giữa hai lớp với nhau, thể hiện chúng có liên quan với nhau. Association thể hiện qua các quan hệ như “has: có”, “Own: sở hữu” v.v...

- + **Aggregation**: Aggregation là một loại của quan hệ Association nhưng mạnh hơn. Nó có thể cùng thời gian sống (cùng sinh ra hoặc cùng chết đi)
- + **Composition** **Composition** là một loại mạnh hơn của Aggregation thể hiện quan hệ class này là một phần của class kia nên dẫn đến cùng tạo ra hoặc cùng chết đi.
- + **Generalization** **Generalization** là quan hệ thừa kế được sử dụng rộng rãi trong lập trình hướng đối tượng:

2. Các bước xây dựng biểu đồ lớp (Class Diagram)

Bước 1: Tìm các Classes dự kiến Entity Classes(các lớp thực thể) là các thực thể có thật và hoạt động trong hệ thống, thường dựa vào các nguồn sau để xác định chúng.

- Phân tích các danh từ trong các yêu cầu để tìm ra các thực thể.
- Phân tích các Use Case sẽ cung cấp thêm các Classes dự kiến.

Lưu ý: Chỉ các thực thể bên trong hệ thống được xem xét, các thực thể bên ngoài hệ thống không được xem xét.

Bước 2: Tìm các thuộc tính và phương thức cho lớp

- *Tìm thuộc tính:* phân tích thông tin từ các form mẫu có sẵn, bạn sẽ tìm ra thuộc tính cho các đối tượng của lớp. Ví dụ các thuộc tính của lớp Customer sẽ thể hiện trên Form đăng ký thông tin khách hàng.
- *Tìm phương thức:* phương thức là các hoạt động mà các đối tượng của lớp này có thể thực hiện. Chúng ta sẽ bổ sung phương thức đầy đủ cho các lớp khi phân tích Sequence Diagram sau này.

Bước 3: Xây dựng các quan hệ giữa các lớp và phát hiện các lớp phát sinh Phân tích các quan hệ giữa các lớp và định nghĩa các lớp phát sinh do các quan hệ sinh ra. Chúng ta phân tích các thực thể ở trên để xác định các kiểu quan hệ.

Phần 2. Thực hành

Bài tập 1.

“Một công ty chuyên kinh doanh về các thiết bị điện tử và công nghệ thông tin trong nhiều năm nay và đã có một lượng khách hàng nhất định. Để mở rộng hoạt động kinh doanh của mình, công ty mong muốn xây dựng một hệ thống thương mại điện tử nhằm mở rộng phạm vi kinh doanh trên mạng Internet. Hệ thống mới phải đảm bảo cho khách hàng viếng thăm Website dễ dàng lựa chọn các sản phẩm, xem các khuyến mãi cũng như mua hàng. Việc thanh toán có thể được thực hiện qua mạng hoặc thanh toán trực tiếp tại cửa hàng. Khách hàng có thể nhận hàng tại cửa hàng hoặc sử dụng dịch vụ chuyển hàng có phí của công ty. Ngoài ra, hệ thống cũng cần có phân hệ để đảm bảo cho công ty quản lý các hoạt động kinh doanh như số lượng hàng có trong kho, quản lý đơn đặt hàng, tình trạng giao hàng, thanh toán v.v...”. Thông tin chi tiết các chức năng các bạn có thể tham khảo thêm tại các Website bán hàng.

Dựa vào bản mô tả hệ thống trên, hãy thực hiện các yêu cầu sau:

1. Phân tích các yêu cầu để xác định các lớp thực thể (gồm: tên lớp, các thuộc tính cơ bản, các phương thức cơ bản).
2. Xác định mối quan hệ giữa các lớp.
3. Sử dụng Rational Rose để biểu diễn các lớp đó.
4. Xây dựng biểu đồ lớp tổng quát cho mô hình trên

Hướng dẫn thực hiện

Bước 1: Tìm các Classes dự kiến

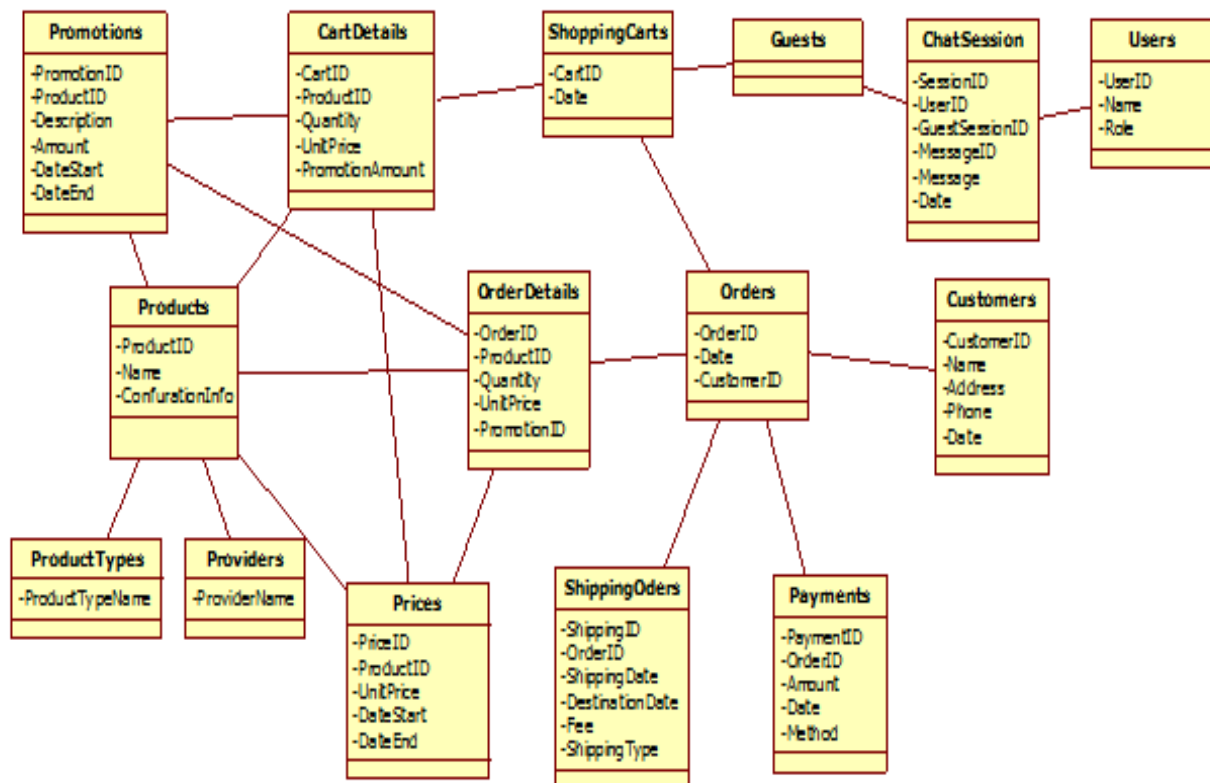
- Phân tích Use Case “*Xem sản phẩm*” chúng ta xác định thực thể sản phẩm (**Products**). Sản phẩm được phân loại theo chủng loại (**Product Types**) và Nhà sản xuất (**Providers**) nên đây có thể là 2 lớp có quan hệ với class **Products**.
- Xem xét Use Case “*Xem khuyến mãi*” xác định Class Chương trình khuyến mãi (**Promotions**).
- Use Case “*Quản lý giỏ hàng*” → Class giỏ hàng (**Shopping Carts**)
- Use Case *Chat* → Class **Chat session**. Những người dùng tham gia Chat là **Sales** và **Guest** có thể là hai class dự kiến.
- Use Case “*Đăng ký thành viên*” → Khách hàng (**Customers**)
- Use Case “*Quản lý đơn hàng*” → Class đơn hàng (**Orders**), class thu tiền (**Payments**) và Quản lý chuyển hàng (**Shipping Orders**) có thể là 2 lớp có liên quan với Class **Orders**.

Bước 2: Xác định thuộc tính và quan hệ cho các lớp

- **Products:** xem xét tài liệu mô tả sản phẩm của hệ thống chúng ta có thể thấy Class Products cần những thuộc tính sau: *Tên sản phẩm, mô tả, cấu hình, Giá bán, khuyến mãi, bảo hành, ...* Trong đó, thuộc tính giá thay đổi theo thời gian nên chúng ta nên tách ra thành lớp riêng là Giá (**Prices**). Tương tự thuộc tính khuyến mãi cũng được tách ra thành lớp **Promotions**.
- **Prices:** có các thuộc tính là *Mã sản phẩm, Giá, ngày bắt đầu, ngày hết hạn*.
- **Promotions:** tương tự như giá nó cần có lớp riêng với các thuộc tính là *Mã sản phẩm, Mô tả khuyến mãi, Giá trị khuyến mãi, Ngày bắt đầu, Ngày hết hạn*.
- **ProductTypes:** chứa loại sản phẩm
- **Providers:** chứa tên nhà sản xuất
- **ShoppingCarts:** chứa các thông tin như: *cartID, ngày, mã sản phẩm, số lượng, đơn giá*. Chúng ta nhận thấy nếu để nguyên lớp này khi tạo đối tượng chúng sẽ lặp thông tin *cartID* và *ngày mua* nên tách chúng ra thành **ShoppingCarts** với các thuộc tính *CartID, ngày* và **CartDetails** với các thuộc tính *ProductID, số lượng, đơn giá*.
- Tương tự chúng ta có class **Orders** với *OrderID, ngày, customerID* và class **Orderdetails** với *ProductID, số lượng, đơn giá*.
- **Payments:** chứa các thông tin như *PaymentID, OrderID, ngày trả, số tiền, hình thức thanh toán*.

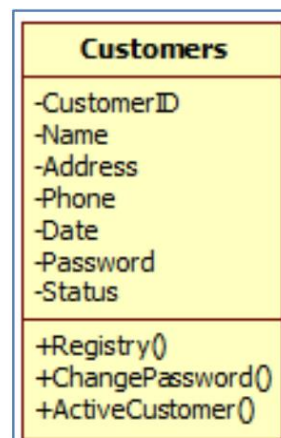
- **Shippings:** có thể chứa *ShippingID*, *OrderID*, Ngày chuyển, ngày đến, số tiền, phương thức vận chuyển.
- **Customers:** *CustomerID*, Họ và tên, địa chỉ, điện thoại, ngày đăng ký v.v...
- **Guests:** có thể chứa *sessionID* để xác định thông tin khi chat
- **Sales:** có thể gộp với lớp người dùng (**Users**) chứa *UserID*, *Name*
- **ChatSessions:** *ChatsessionID*, tên người bán hàng, mã khách, mã tin nhắn, nội dung tin nhắn, ngày

Nhập đầy đủ thuộc tính và vẽ, chúng ta có bản vẽ như sau:



Bước 3: Bổ sung phương thức cho các lớp

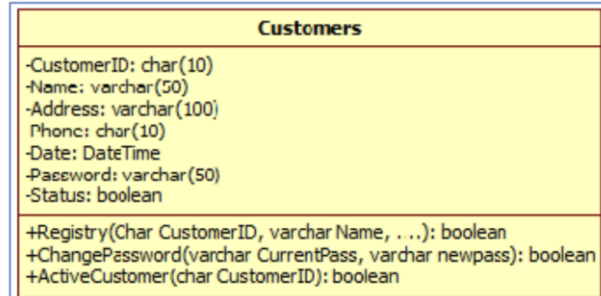
Phương thức là các hành động mà đối tượng sinh ra từ lớp đó có thể thực hiện trong hệ thống. Ví dụ các đối tượng của lớp **Customers** có thể đăng ký mới, có thể thay đổi mật khẩu (*password*), kích hoạt người dùng (*Active*) v.v..



Bước 4: Thiết kế chi tiết các thuộc tính và phương thức cho lớp

Khi đã có được Class Diagram, chúng ta cần thiết kế chi tiết các lớp bằng cách đặc tả các thuộc tính và phương thức của nó.

- **Đặc tả thuộc tính:** chúng ta xác định kiểu dữ liệu và kích thước.
- **Đặc tả phương thức:** chúng ta xác định dữ liệu đầu vào, dữ liệu đầu ra.



Việc sử dụng các kiểu dữ liệu và mô tả các phương thức của một lớp chúng ta đã học trong lập trình hướng đối tượng.

Bài tập 2:

HỆ THỐNG QUẢN LÝ THƯ VIỆN cho trường Đại học gồm các hoạt động **Quản lý thông tin sách, Quản lý thông tin độc giả, Quản lý hoạt động mượn trả sách** được mô tả như sau:

- *Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.*

- *Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngày nhập). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.*

- *Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).*

- *Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).*

- Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.

- Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.

- Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.

- Người dùng hệ thống phải đăng nhập trước khi thực hiện.

Dựa vào bản mô tả hệ thống quản lý thư viện, hãy thực hiện các yêu cầu sau:

1. Xác định các lớp thực thể (gồm: tên lớp, các thuộc tính cơ bản, các phương thức cơ bản).
2. Xác định mối quan hệ giữa các lớp.
3. Sử dụng Rational Rose để biểu diễn các lớp đó.
4. Phân tích kịch bản cho từng Use case để xây dựng biểu đồ lớp cho từng Use case đó.

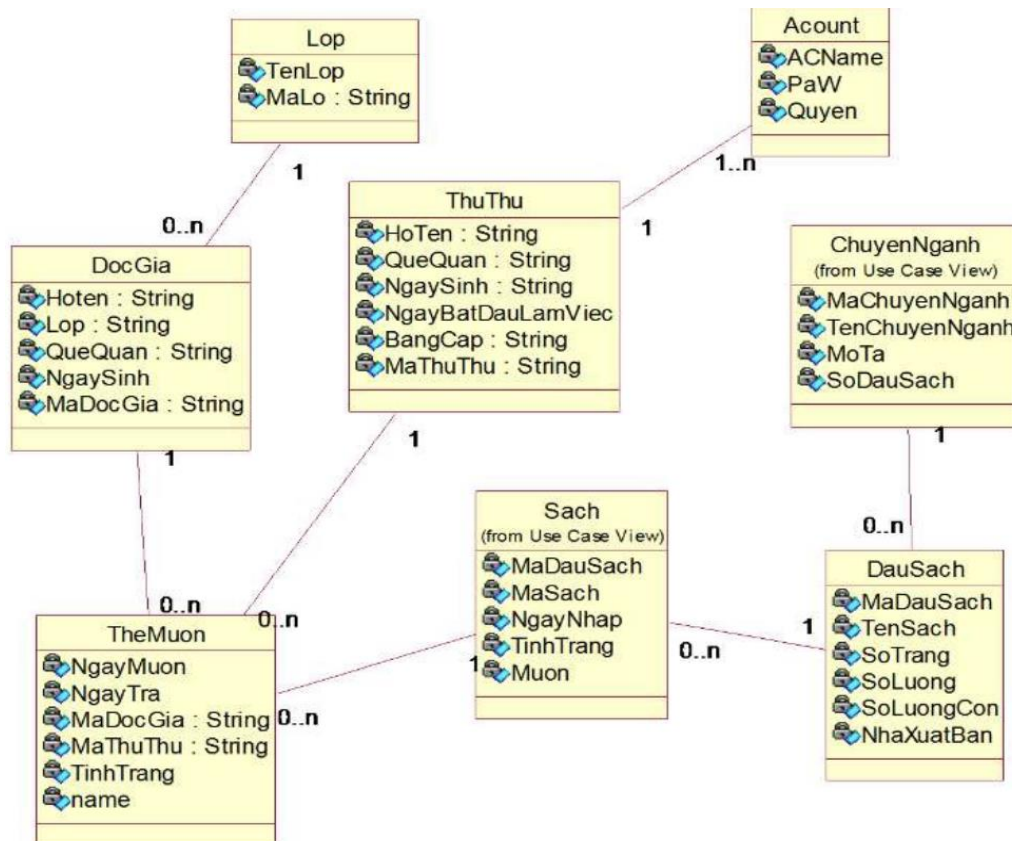
Hướng dẫn thực hiện:

Bước 1. Xác định các lớp cơ bản

- Xem xét kỹ lại đặc tả hệ thống, kịch bản Use case, sử dụng phương pháp trích chọn danh từ: *Sinh viên, lớp, thẻ thư viện, Mã độc giả, họ tên, lớp, ngày sinh, giới tính, thủ thư, sách, đầu sách, bản sao, mã đầu, mã sách, tình trạng, chuyên ngành, Mã chuyên ngành, tên chuyên ngành, mô tả, độc giả, phiếu mượn, Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng, tài khoản người dùng.*
- Loại bỏ các danh từ trùng nhau hoặc không có giá trị. Ví dụ danh từ *Sinh viên, thẻ thư viện, bạn đọc, độc giả* đều chỉ một đối tượng người tham gia mượn sách thư viện là độc giả.
- Xác định các thuộc tính cho từng lớp. Các danh từ có thể trở thành lớp, và các danh từ khác có thể trở thành thuộc tính của lớp. Ví dụ danh từ *phiếu mượn* thành lớp **PhieuMuon** các danh từ *Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng* thành thuộc tính của lớp **PhieuMuon**.

Bước 2. Xác định quan hệ của các lớp

Minh họa

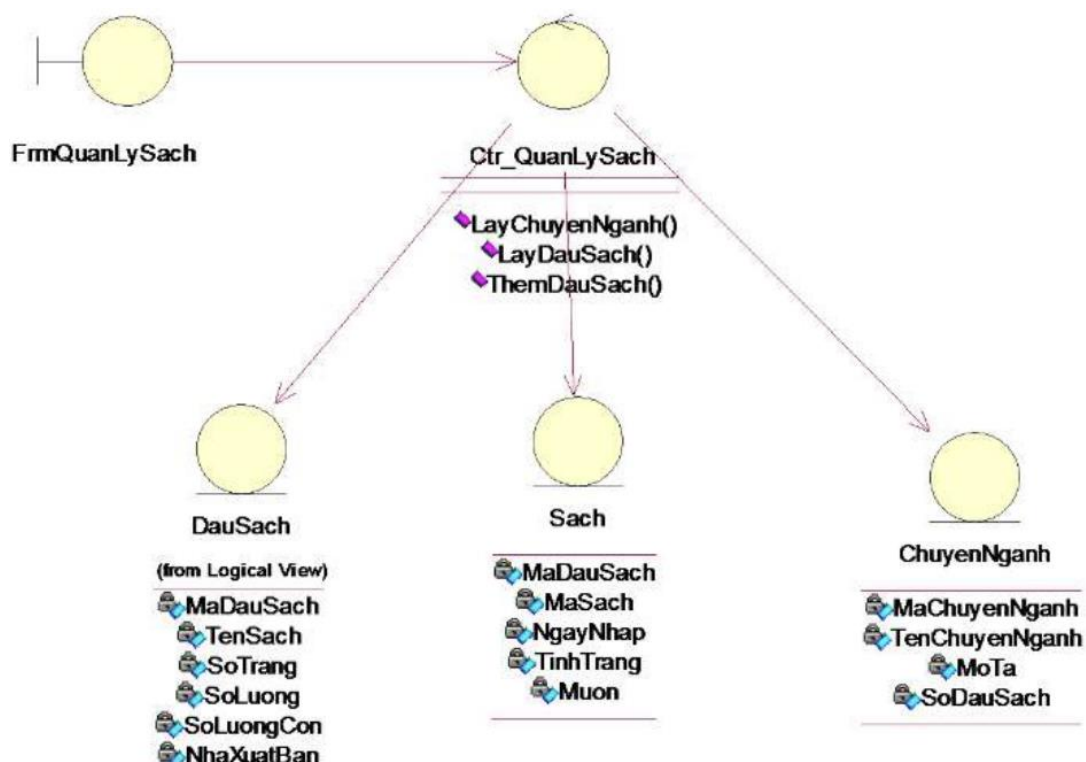


Bước 3. Phân tích kịch bản và xây dựng biểu đồ lớp cho từng Use case.

Ví dụ: Use case Thêm đầu sách mới

- Tìm lớp Bound: theo nguyên tắc giữa Actor và Use case có một lớp Bound.
- Xác định lớp Control: theo nguyên tắc mỗi Use case có ít nhất một lớp Control.
- Xác định các lớp thực thể: Đọc kịch bản xác định các danh từ tham gia Use case để tìm ra lớp thực thể.

Minh họa



BÀI SỐ 5

XÂY DỰNG BIỂU ĐỒ TRẠNG THÁI (State Diagram)

MỤC TIÊU:

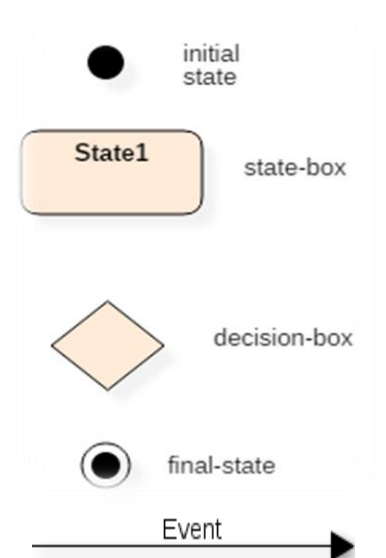
- Trình bày được các thành phần của biểu đồ trạng thái và các bước xây dựng biểu đồ trạng thái (State Diagram);
- Xác định được các trạng thái khác nhau theo vòng đời của đối tượng và các sự kiện chịu trách nhiệm cho các thay đổi trạng thái đó;
- Sử dụng thành thạo phần mềm để mô hình hóa các trạng thái của đối tượng;
- Xây dựng được biểu đồ trạng thái của các lớp trong ứng dụng, trong các Use case;
- Rèn luyện tính cẩn thận, chính xác, tư duy logic, sáng tạo của người học.

YÊU CẦU

Phần 1. Lý thuyết

1. Các thành phần trong biểu đồ trạng thái

- **Initial state:** Trạng thái ban đầu, được sử dụng để chỉ ra sự bắt đầu của biểu đồ trạng thái.
- **State-box:** Hộp trạng thái, là trạng thái tại một thời điểm cụ thể trong vòng đời của đối tượng.
- **Decision-box:** Hộp quyết định chứa điều kiện chuyển đổi trạng thái.
- **Final-state:** Trạng thái kết thúc, được sử dụng để chỉ ra sự kết thúc của biểu đồ trạng thái.
- **Event:** Sự kiện, là quá trình chuyển đổi gây ra sự thay đổi trạng thái của đối tượng.



2. Các bước xây dựng biểu đồ trạng thái

Bước 1: Xác định trạng thái ban đầu và trạng thái kết thúc cuối cùng.

Bước 2: Xác định các trạng thái khả dĩ mà đối tượng có thể tồn tại (các giá trị biên tương ứng với các thuộc tính khác nhau hướng dẫn chúng ta xác định các trạng thái khác nhau).

Bước 3: Gắn nhãn các sự kiện kích hoạt các chuyển đổi này.

Phần 2. Thực hành

Bài tập 1

HỆ THỐNG QUẢN LÝ THƯ VIỆN cho trường Đại học gồm các hoạt động **Quản lý thông tin sách, Quản lý thông tin độc giả, Quản lý hoạt động mượn trả sách** được mô tả như sau:

- Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao

cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.

- Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngày nhập). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.

- Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).

- Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).

- Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.

- Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.

- Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.

- Người dùng hệ thống phải đăng nhập trước khi thực hiện.

Dựa vào bản mô tả hệ thống quản lý thư viện, hãy thực hiện các yêu cầu sau:

1. Phân tích kịch bản cho từng Use case và xác định các lớp trong các Use case đó.
2. Xây dựng biểu đồ trạng thái cho các lớp trong các Use case phân tích được.

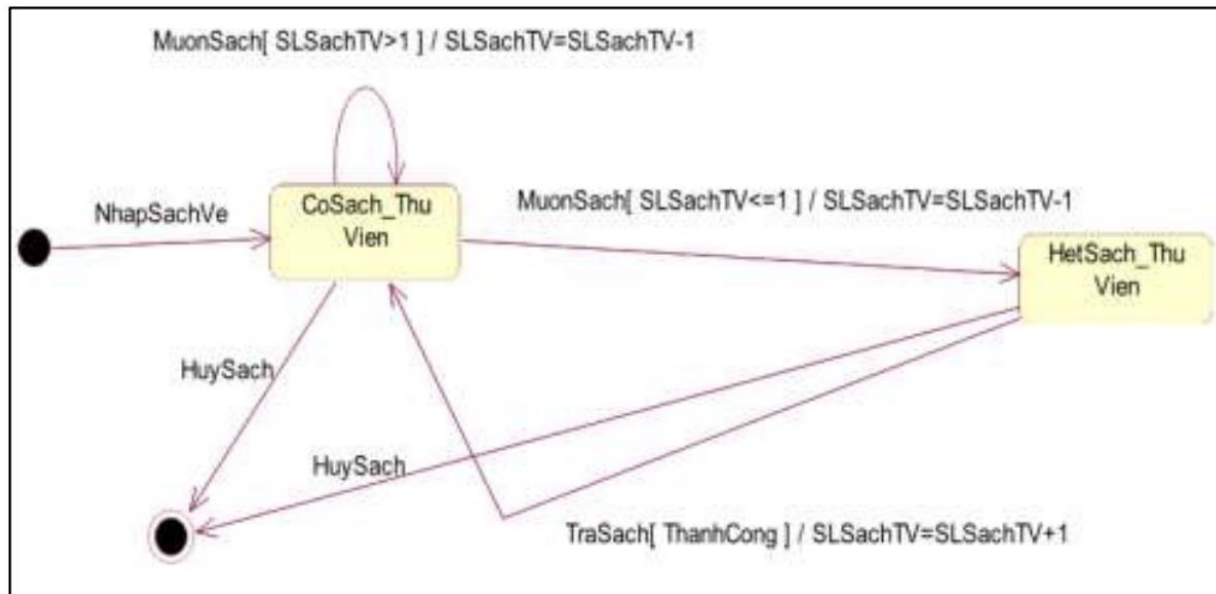
Hướng dẫn thực hiện

Bước 1: Đọc đặc tả phần mềm, hoặc đặc tả chi tiết dựa vào nghiệp vụ hệ thống.

Bước 2: Xác định các trạng thái có thể của lớp.

Bước 3: Xác định các chuyển trạng thái, và sự kiện, điều kiện chuyển, hành động để chuyển giữa các trạng thái.

Minh họa: Biểu đồ trạng thái của lớp DOCGIA trong Use case Trả sách



Bài tập tự thực hiện:

3. Phân tích kịch bản cho từng Use case và xác định các lớp khác (Sách, Phiếu mượn, Phiếu trả, Thủ thư, ...) trong các Use case đó.
4. Xây dựng biểu đồ trạng thái cho các lớp trong các Use case phân tích được ở trên.

BÀI SỐ 6

XÂY DỰNG BIỂU ĐỒ TUẦN TỰ (Sequence Diagram)

MỤC TIÊU:

- Trình bày được các thành phần của biểu đồ tương tác theo thời gian Sequence Diagram (biểu đồ tuần tự).
- Xây dựng được các biểu đồ tuần tự
- Sử dụng thành thạo phần mềm để biểu diễn các biểu đồ
- Xây dựng được biểu đồ lớp chi tiết dựa vào biểu đồ tuần tự và biểu đồ lớp ở mức phân tích

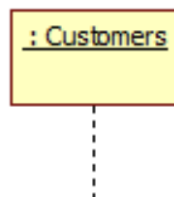
YÊU CẦU

Phần 1. Lý thuyết

1. Các thành phần của Sequence Diagram

- ✓ Objects: mô tả một đối tượng trong hệ thống. (*Object có dấu ":" phía trước tên của nó, phân biệt với Class*). Đường gạch chấm bên dưới đối tượng thể hiện thời gian sống của đối tượng.

Ký hiệu



- ✓ Stimulus (message): thể hiện thông điệp từ một đối tượng này tương tác với một đối tượng khác. Ký hiệu:



- ✓ Axes (Trục tọa độ): trục ngang thể hiện các đối tượng, trục đứng thể hiện thời gian.

Chúng ta, dễ dàng nhận thấy các đối tượng tương tác với nhau theo tuần tự các bước để hình thành nên chức năng của hệ thống

2. Ứng dụng Sequence Diagram

- ✓ Thiết kế các chức năng
- ✓ Kiểm chứng và bổ sung method cho các Class
- ✓ Sử dụng trong việc coding các chức năng

3. Các bước xây dựng biểu đồ tuần tự

Bước 1: Xác định chức năng cần thiết kế. Dựa vào biểu đồ ca sử dụng (Use Case Diagram) để xác định xem chức năng nào cần thiết kế.

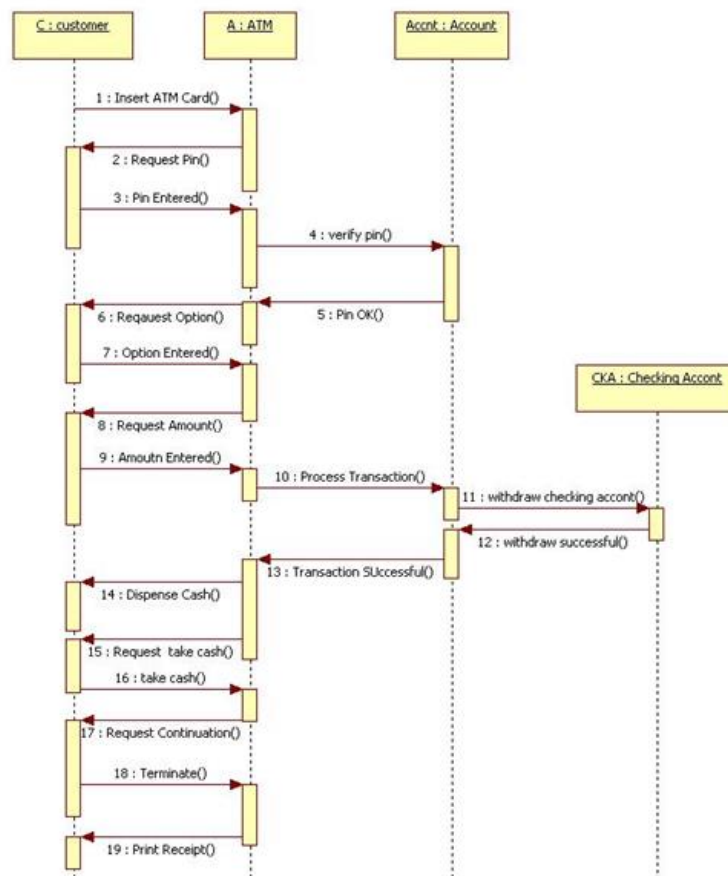
Bước 2: Dựa vào biểu đồ hành động (Activity Diagram) để xác định các bước thực hiện theo nghiệp vụ.

Bước 3: Đối chiếu với biểu đồ lớp (Class Diagram) để xác định lớp trong hệ thống tham gia vào nghiệp vụ.

Bước 4: Vẽ biểu đồ tuần tự (Sequence Diagram)

Bước 5: Cập nhật lại bản vẽ Class Diagram.

Ví dụ: Sequence Diagram cho hoạt động rút tiền ở ATM



Phần 2. Thực hành

Bài tập 1.

Thực hành xây dựng Sequence Diagram cho hệ thống eCommerce

Bước 1: Xác định các Use Case cần thiết kế. Xem xét bản vẽ Use Case Diagram đã vẽ ở bài 3, có thể thấy các Use Case sau cần thiết kế:

- Xem sản phẩm theo chủng loại
- Thêm sản phẩm theo nhà cung cấp
- Thêm giỏ hàng
- Chat
- Quản lý đơn hàng
- Thanh toán

- Theo dõi chuyên hàng
- Đăng nhập

Ví dụ: chúng ta sẽ thiết kế cho chức năng “*Xem sản phẩm theo chủng loại*”

Bước 2: Xác định các hoạt động cho Use Case này:

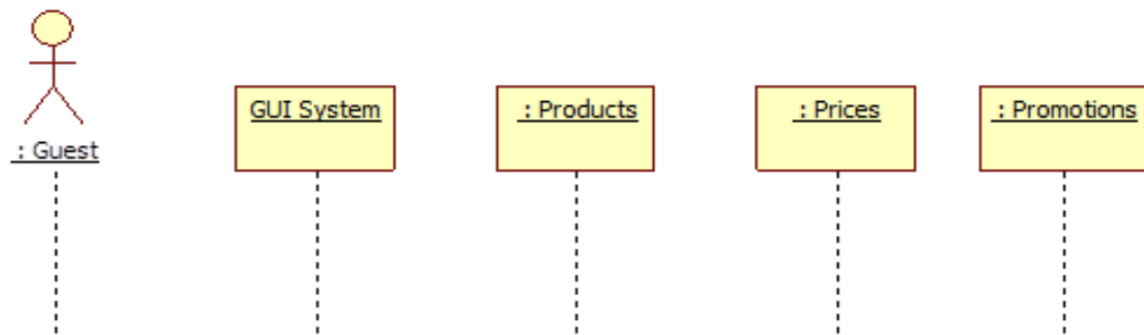
- Người dùng chọn loại sản phẩm
- Hệ thống sẽ lọc lấy loại sản phẩm tương ứng, sau đó lấy giá, lấy khuyến mãi và hiển thị lên màn hình.
- Người dùng xem sản phẩm

Bước 3: Đối chiếu với Class Diagram để xác định các đối tượng thực hiện như sau:

- ✓ **Người dùng:** chọn loại sản phẩm qua giao diện
- ✓ **Giao diện:** sẽ lấy danh sách sản phẩm tương ứng từ Products
- ✓ **Giao diện:** lấy giá của từng sản phẩm từ Class Prices và Promotion Amount từ lớp Promotions
- ✓ **Giao diện:** tổng hợp danh sách và hiển thị
- ✓ **Người dùng:** Xem sản phẩm

Bước 4: Vẽ Sequence Diagram

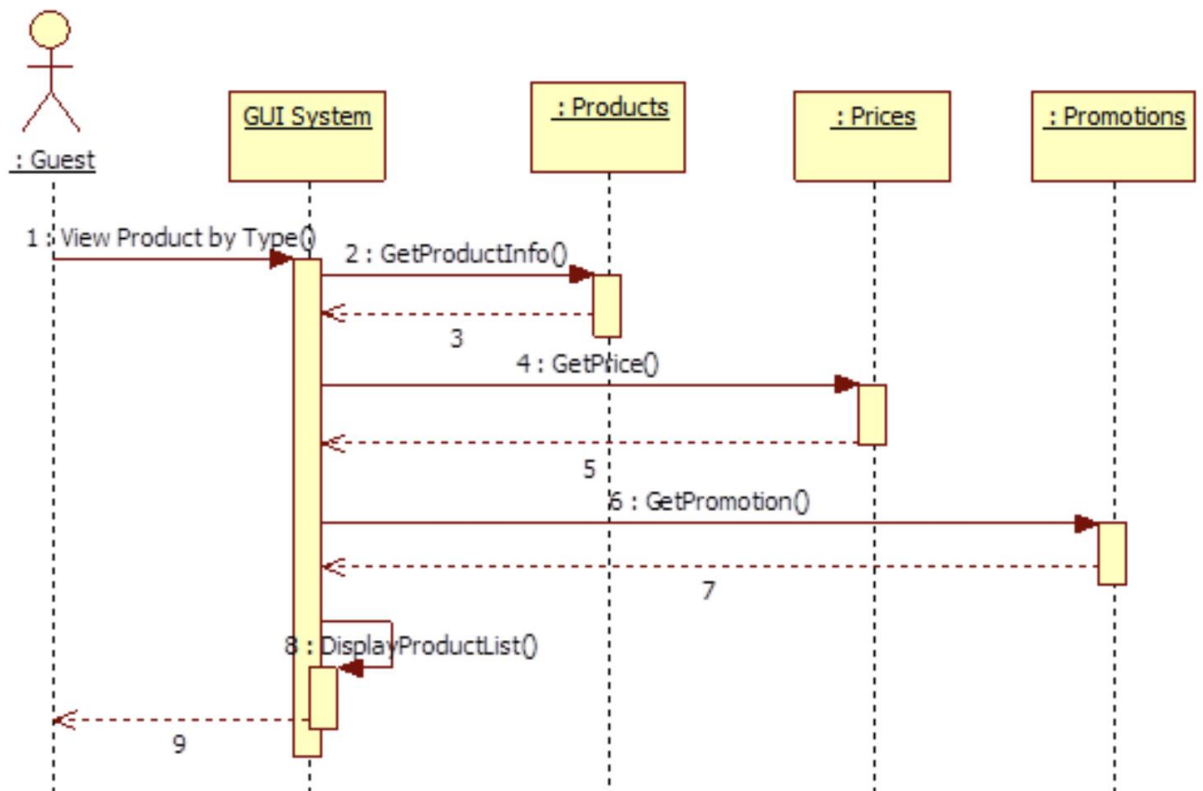
- Xác định các lớp tham gia vào hệ thống gồm: người dùng (Guest), Giao diện (GUI System), Sản phẩm (Products), Giá (Prices), Khuyến mãi (Promotions).
- Trong đó: GUI System để sử dụng chung cho giao diện, có thể sử dụng cụ thể trang Web nào nếu đã có Mockup (*thiết kế chi tiết của giao diện*).



Các đối tượng tham gia vào bản vẽ

Các bước thực hiện của Use Case này như sau:

- ✓ Guest gửi yêu cầu xem sản phẩm lên giao diện kèm theo chủng loại
- ✓ GUI system: gửi yêu cầu lấy danh sách các sản phẩm tương ứng với chủng loại cho lớp sản phẩm và nhận lại danh sách.
- ✓ GUI system: gửi yêu cầu lấy Giá cho từng sản phẩm từ Prices
- ✓ GUI system: gửi yêu cầu lấy khuyến mãi cho từng sản phẩm từ Promotions và nhận lại kết quả
- ✓ GUI system: ghép lại danh sách và hiển thị lên browser và trả về cho Guest Thể hiện lên bản vẽ như sau:

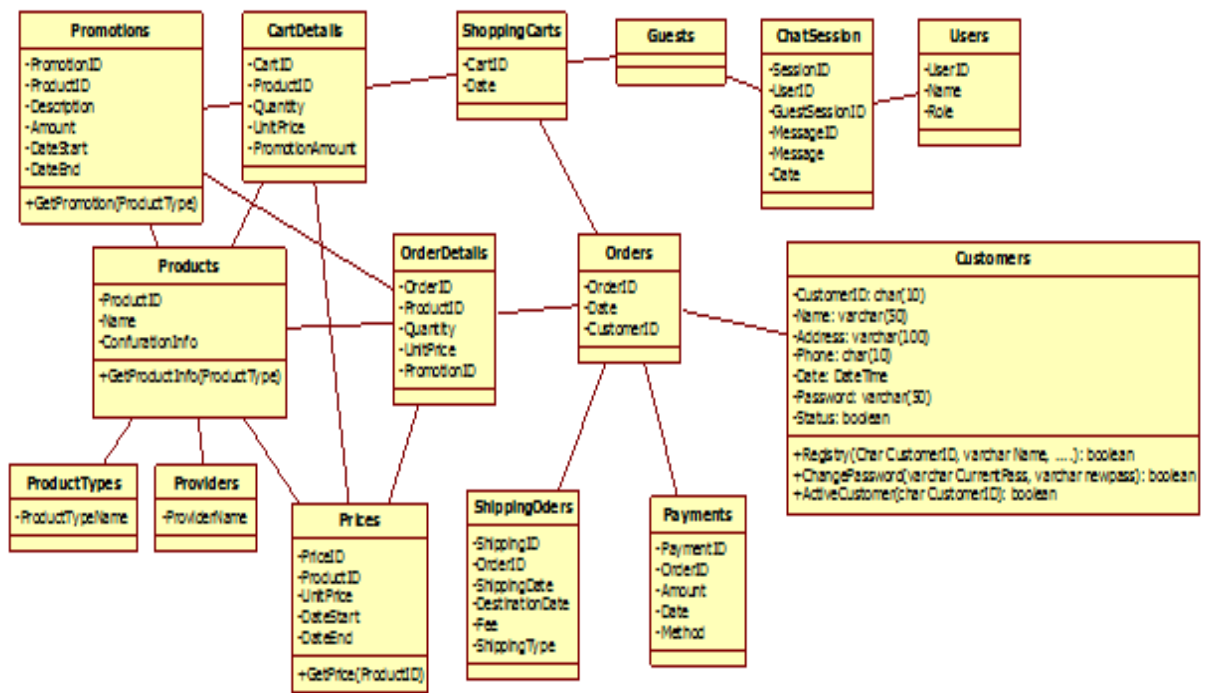


Bản vẽ Sequence Diagram cho chức năng Xem sản phẩm theo chủng loại.

Bước 5: Kiểm tra và cập nhật bản vẽ Class Diagram. Cần bổ sung các phương thức cho các lớp như sau:

- ✓ **Products class:** bổ sung phương thức `GetProductInfo(Product Type)`: trả về thông tin sản phẩm có loại được truyền vào. Việc này các đối tượng của lớp Products hoàn toàn làm được vì họ đã có thuộc tính `ProductType` nên họ có thể trả về được thông tin này.
- ✓ **Prices:** bổ sung phương thức `GetPrice(ProductID)`: `UnitPrice`. Sau khi lấy được `ProductID` từ Products, GUI gọi phương thức này để lấy giá của sản phẩm từ lớp giá. Các đối tượng từ lớp Prices hoàn toàn đáp ứng điều này.
- ✓ **Promotions:** tương tự bổ sung phương thức `GetPromotion(ProductID)`.
- ✓ **GUI System (View Product Page):** bổ sung phương thức `DisplayProductList (List of product)` để hiển thị danh sách lên sản phẩm. Ngoài ra, cần có thêm một phương thức `ViewProductbyType(ProductType)` để mô tả chính hoạt động này khi người dùng kích chọn

Như vậy, chúng ta thấy các phương thức trên đều thực hiện được trên các đối tượng của các lớp nên thiết kế của trên là khả thi. Bổ sung các phương thức trên vào các Class tương ứng chúng ta có bản vẽ Class Diagram như sau:



Class Diagram sau khi đã bổ sung các phương thức mới

Ngoài ra, có thể bổ sung các lớp giao diện vào Class Diagram để hoàn chỉnh thiết kế cho hệ thống. Hoàn tất sequence diagram cho tất cả các Use Case chúng ta sẽ hoàn thành việc thiết kế, đồng thời cũng hoàn tất bản vẽ Class Diagram.

Bài tập 2.

Thực hiện vẽ các biểu đồ sau cho hệ thống Quản lý thư viện

Yêu cầu:

1. Xây dựng biểu đồ tuần tự cho Use case quản lý sách.
2. Xây dựng biểu đồ tuần tự cho Use case thêm đầu sách mới.
3. Xây dựng biểu đồ lớp chi tiết của Use case quản lý sách.

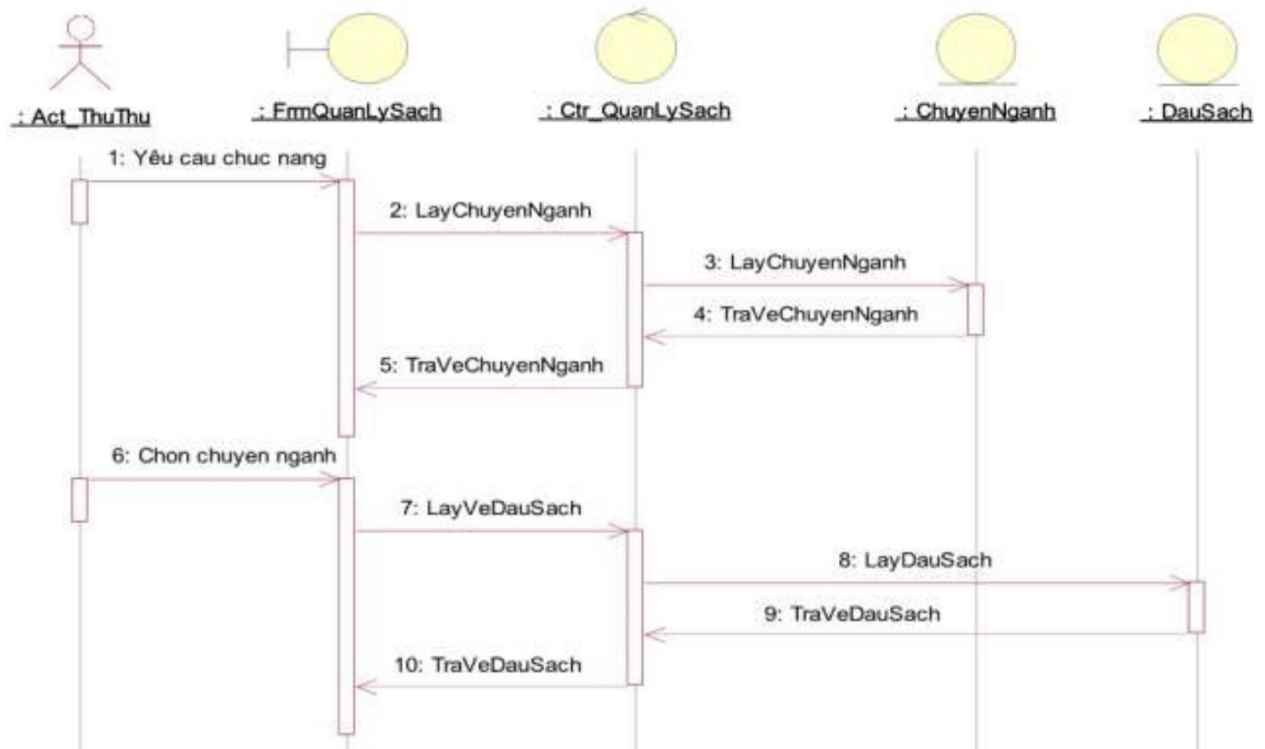
Hướng dẫn thực hiện:

1. Xây dựng biểu đồ tuần tự cho *Use case quản lý sách*.

**** Phân tích bài toán:***

- ✓ *Bước 1:* Kiểm tra lại Use case quản lý sách, biểu đồ phân rã Use case quản lý sách, kịch bản Use case
- ✓ *Bước 2:* Xác định rõ các kiểu đối tượng tham gia (giao diện, điều kiện hay thực thể) dựa vào biểu đồ lớp phân tích Use case và đối tượng Actor thực hiện.
- ✓ *Bước 3:* Căn cứ vào kịch bản Use case, xây dựng biểu đồ tuần tự cho mỗi một luồng chính và các luồng phụ, xác các thông điệp được gửi đi giữa các đối tượng, thời gian xử lý thông điệp.
- ✓ *Bước 4:* Xây dựng biểu đồ tuần tự

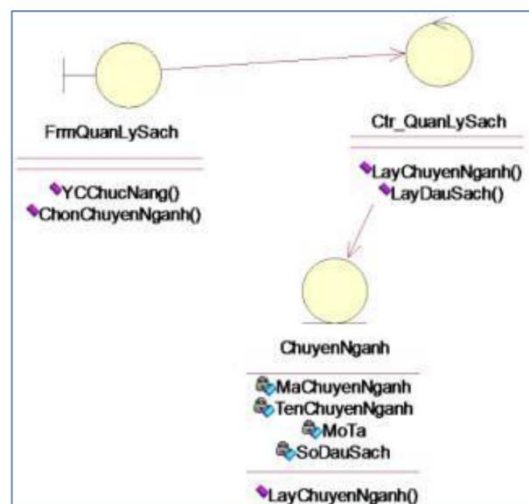
Minh họa:



* Hoàn thiện biểu đồ lớp: Xác định thêm các phương thức dựa vào biểu đồ tuần tự:

Ví dụ: dựa vào biểu đồ tuần tự Use case Quản lý sách trên, Lớp **Ctrl_QuanLySach** sẽ có thêm 2 phương thức *LayChuyenNganh*, *LayVeDauSach*, vì thông điệp 2 và 7 được gửi đến lớp **Ctrl_QuanLySach**.

Minh họa:



Tương tự như vậy sau khi hoàn thành các biểu đồ tuần tự cho các Use case phân rõ khác, xác định thêm các phương thức cho biểu đồ chi tiết.

BÀI SỐ 7

XÂY DỰNG BIỂU ĐỒ HOẠT ĐỘNG (Activity Diagram)

MỤC TIÊU:

- Trình bày được các thành phần trong biểu đồ hoạt động - Activity Diagram.
- Xây dựng được các biểu đồ hoạt động của hệ thống quản lý.
- Sử dụng thành thạo phần mềm để biểu diễn các biểu đồ
- Xây dựng được biểu đồ hoạt động theo đúng yêu cầu.

YÊU CẦU

Phần 1. Lý thuyết

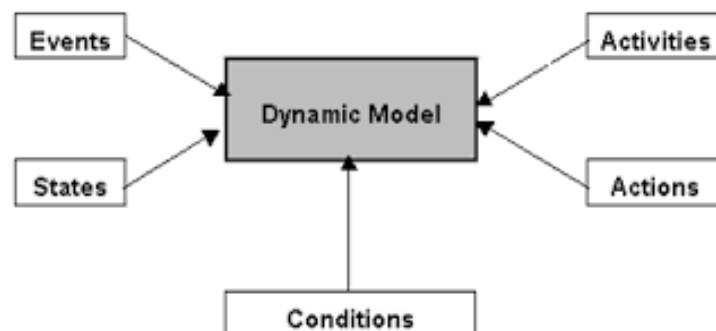
1. Biểu đồ hoạt động - Activity Diagram:

Theo UML thì khía cạnh hoạt động của hệ thống có thể được mô tả theo 2 mô hình tĩnh (Static Model) và mô hình động (Dynamic Model).

Static Model: mô tả cấu trúc của hệ thống bao gồm các bản vẽ Class Diagram, Object Diagram, Component Diagram và Deployment Diagram.

Dynamic Model: mô tả các hoạt động bên trong hệ thống bao gồm các bản vẽ Activity Diagram, State Diagram, Sequence Diagram, Collaboration Diagram

2. Các thành phần cơ bản của Dynamic Model



- ✓ *Event*: là sự kiện, mô tả một hoạt động bên ngoài tác động vào đối tượng và được đối tượng nhận biết và có phản ứng lại.
- ✓ *Activity*: mô tả một hoạt động trong hệ thống. Hoạt động có thể do một hoặc nhiều đối tượng thực hiện.
- ✓ *State*: là trạng thái của một đối tượng trong hệ thống, được mô tả bằng giá trị của một hoặc nhiều thuộc tính.
- ✓ *Action*: chỉ hành động của đối tượng.
- ✓ *Condition*: mô tả một điều kiện.

3. Activity Diagram

Activity Diagram là bản vẽ tập trung vào mô tả các hoạt động, luồng xử lý bên trong hệ thống. Nó có thể được sử dụng để mô tả các quy trình nghiệp vụ trong hệ thống, các luồng của một chức năng hoặc các hoạt động của một đối tượng.

Các ký hiệu Activity Diagram:

- ✓ *Swimlane*: Swimlane được dùng để xác định đối tượng nào tham gia hoạt động nào trong một quy trình.



- ✓ Nút *Start, End*:

Start thể hiện điểm bắt đầu quy trình

End thể hiện điểm kết thúc quy trình.



Ký hiệu về nút Start



Ký hiệu về nút kết thúc

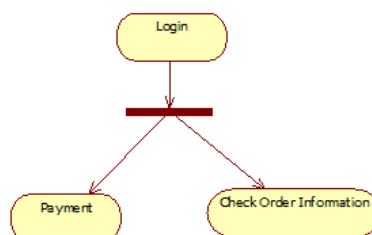
- ✓ *Activity*: mô tả một hoạt động trong hệ thống. Các hoạt động này do các đối tượng thực hiện.



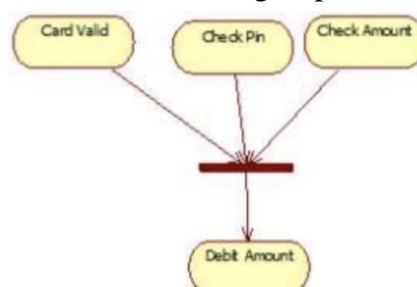
- ✓ *Branch*: thể hiện rẽ nhánh trong mệnh đề điều kiện



- ✓ *Fork*: thể hiện cho trường hợp thực hiện xong một hoạt động rồi sẽ rẽ nhánh thực hiện nhiều hoạt động tiếp theo.



- ✓ *Join*: Cùng ký hiệu với Fork nhưng thện trường hợp phải thực hiện hai hay nhiều hành động trước rồi mới thực hiện hành động tiếp theo.



4. Ứng dụng của Activity Diagram

- ✓ Phân tích nghiệp vụ để hiểu rõ hệ thống
- ✓ Phân tích Use Case
- ✓ Cung cấp thông tin để thiết kế bản vẽ Sequence Diagram

5. Cách xây dựng Activity Diagram

Bước 1: Xác định các nghiệp vụ cần mô tả Xem xét bản vẽ Use Case để xác định nghiệp vụ nào bạn cần mô tả.

Bước 2: Xác định trạng thái đầu tiên và trạng thái kết thúc.

Bước 3: Xác định các hoạt động tiếp theo Xuất phát từ điểm bắt đầu, phân tích để xác định các hoạt động tiếp theo cho đến khi gặp điểm kết thúc để hoàn tất bản vẽ này.

Phần 2. Thực hành

Bài tập 1. Thực hành xây dựng Activity Diagram cho hệ thống eCommerce.

Hướng dẫn thực hiện

Bước 1: Xác định các nghiệp vụ cần phân tích.

Trước tiên, chúng ta xem xét các Use Case. Về nguyên tắc phải phân tích và mô tả tất cả các nghiệp vụ của hệ thống để làm rõ hệ thống. Xem xét bản vẽ Use Case Diagram, có thể thấy các Use Case sau cần làm rõ:

- ✓ Xem sản phẩm theo chủng loại
- ✓ Thêm sản phẩm theo nhà cung cấp
- ✓ Thêm giỏ hàng
- ✓ Chat
- ✓ Quản lý đơn hàng
- ✓ Thanh toán
- ✓ Theo dõi chuyển hàng
- ✓ Đăng nhập

Tiếp theo, chúng ta bắt đầu phân tích và vẽ cho chức năng xem sản phẩm theo chủng loại.

Bước 2: Xác định các bước thực hiện và đối tượng liên quan

Để thực hiện chức năng xem sản phẩm theo chủng loại hệ thống sẽ thực hiện như sau:

- + Điều kiện ban đầu: ở trang chủ
- + Điều kiện kết thúc: Hiển thị xong trang sản phẩm

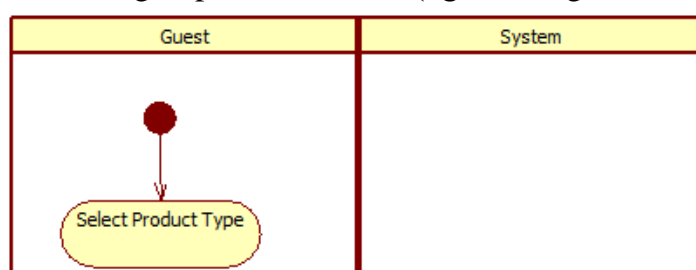
Các bước như sau:

- + Người dùng chọn loại sản phẩm
- + Hệ thống sẽ lọc lấy loại sản phẩm tương ứng, sau đó lấy giá, lấy khuyến mãi cho tất cả các sản phẩm đã được chọn và hiển thị lên màn hình
- + Người dùng xem sản phẩm

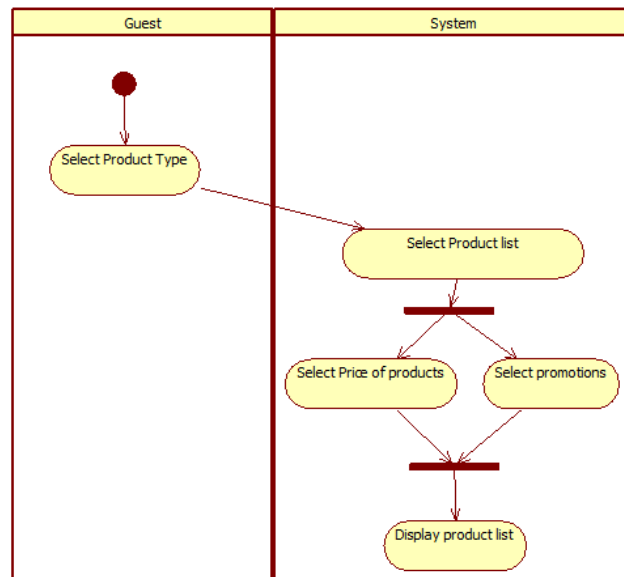
Bước 3: Thực hiện bản vẽ

Chúng ta thấy có 2 đối tượng tham gia vào giao dịch này là Người dùng và Hệ thống. Chúng ta sử dụng Swimlane để thể hiện người dùng tham gia vào nghiệp vụ hệ thống.

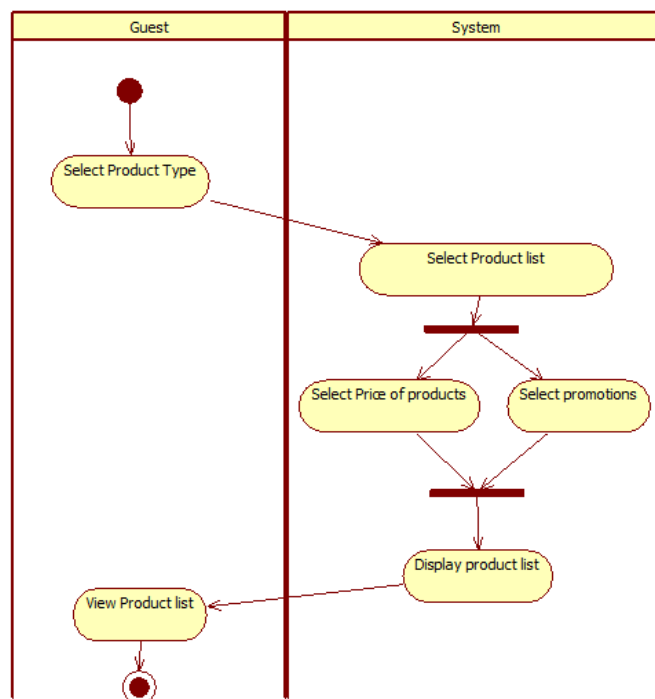
- ✓ Xác định trạng thái đầu tiên.
- ✓ Hành động tiếp theo là Guest (người dùng) chọn loại sản phẩm



Hệ thống sẽ lấy danh sách sản phẩm tương ứng với loại đó, sau đó lấy giá, lấy khuyến mãi của chúng và hiển thị ra màn hình. Hành động lấy giá và khuyến mãi của sản phẩm có thể làm song song nên chúng ta dùng *Fork* và *Join* để thể hiện.



Hệ thống tập hợp danh sách sản phẩm và thông tin liên quan để hiển thị lên Browser. Người dùng xem danh sách sản phẩm và kết thúc nghiệp vụ của Use Case này.



Tương tự, hãy hoàn tất Activity Diagram cho các nghiệp vụ còn lại cho hệ thống.

Bài tập 2: Mô tả và Xây dựng biểu đồ hoạt động cho các chức năng và yêu cầu của HỆ THỐNG QUẢN LÝ THƯ VIỆN

Yêu cầu:

1. Xây dựng biểu đồ hoạt động của Use case ThemDauSach.
2. Xây dựng biểu đồ hoạt động cho phương thức LayChuyenNganh của lớp Ctr_QuanLySach.

3. Xây dựng các bảng cơ sở dữ liệu cho ứng dụng quản lý thư viện

Hướng dẫn thực hiện:

1. Xây dựng biểu đồ hoạt động cho Use case ThemDauSach:

Đọc lại kịch bản của Use case, và đưa ra giải pháp thực hiện Use case sau đó xây dựng biểu đồ hoạt động cho Use case.

2. Xây dựng biểu đồ hoạt động cho phương thức LayChuyenNganh.

3. Xây dựng cơ sở dữ liệu.

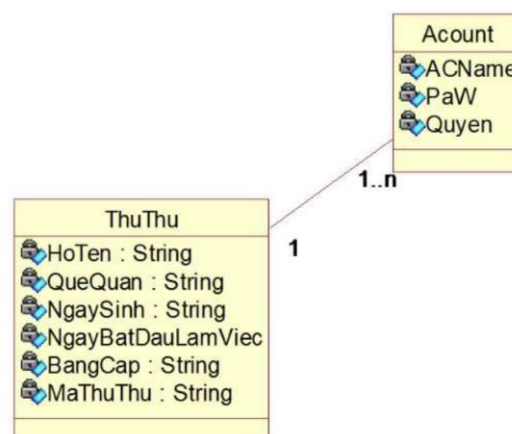
Ánh xạ các lớp trong biểu đồ thực thể theo nguyên tắc:

- ✓ Mỗi lớp thực thể sẽ được ánh xạ thành một bảng cơ sở dữ liệu
- ✓ Quan hệ giữa Association 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 hai cách chuyển

+Cách 1: Nếu thuộc của các lớp con, khác nhau nhiều thì chuyển các lớp con thành bảng, các thuộc tính lớp con thành trường dữ liệu + Các thuộc tính của lớp cha chuyển thành trường dữ liệu các lớp con. Bỏ lớp cha.

+Cách 2: Nếu thuộc tính của các lớp con, giống nhau: Bỏ các lớp con, chuyển lớp cha thành bảng: Gộp các thuộc tính của các lớp con và lớp cha lại chuyển thành trường dữ liệu, và thêm vào bảng một thuộc tính loại để phân biệt đối tượng.



Ví dụ hai lớp thực thể trên chuyển thành hai bảng như sau:

Bảng tblThuThu

STT	Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaThuThu	Nvarchar(10)	Not null	Khóa chính
2	HoTen	Nvarchar(25)	Not null	Họ tên của thủ thư
3	QueQuan	Nvarchar(25)	Allow null	
4	NgaySinh	DateTime	Tuổi >=18	Ngày tháng năm sinh
5	BangCap	nVarchar		Bảng của thủ thư
6	NgayBatDauLamViec	DateTime		Ngày vào làm ở thư viện

Bảng tblAccount

STT	Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaThuThu	Nvarchar(10)	Not null	Khóa ngoại
2	AcountName	Nvarchar(15)	Not null	Tên tài khoản người dùng
3	Pass	Nvarchar(15)	NotNull	Mật khẩu tài khoản người dùng
4	Quyen	Nvarchar(15)		Vai trò người dùng trong hệ thống

Bài tập tự thực hiện

2. Xây dựng biểu đồ hoạt động cho phương thức LayChuyenNganh của lớp Ctr_QuanLySach
3. Xây dựng hoàn thiện cơ sở dữ liệu.

BÀI SỐ 8

XÂY DỰNG BIỂU ĐỒ THÀNH PHẦN VÀ TRIỂN KHAI (Component Diagram & Deployment Diagram)

MỤC TIÊU:

- Trình bày được các thành phần của biểu đồ thành phần và triển khai (Component Diagram & Deployment Diagram).
- Xây dựng được các biểu đồ thành phần và biểu đồ triển khai cho hệ thống.
- Sử dụng thành thạo phần mềm để biểu diễn các biểu đồ trên.

YÊU CẦU

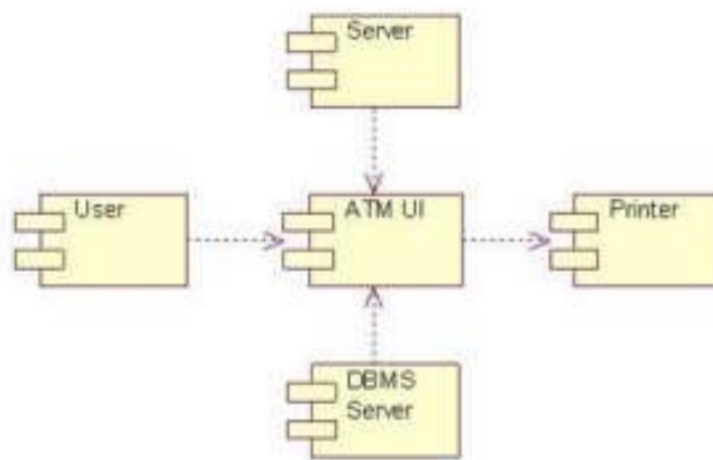
Phần 1. Lý thuyết

1. Biểu đồ thành phần - Component Diagram

Khi thiết kế các hệ thống phức tạp ta nên chia chúng ra thành nhiều hệ thống con (subsystem) để dễ thiết kế. Mỗi hệ thống con sau khi xây dựng có thể đóng gói thành một thành phần phần mềm được triển khai độc lập.

Component Diagram sẽ giúp chúng ta thể hiện cách chia hệ thống ra nhiều thành phần và quan hệ của chúng. Component Diagram là bản vẽ cho biết cấu trúc của hệ thống theo thành phần phần mềm.

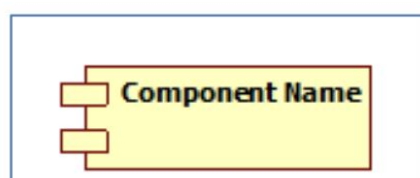
Ví dụ về Component Diagram:



Ví dụ trên cho thấy hệ thống phần mềm ATM chia ra thành 5 thành phần là ATM UI, Server, User, DBMS Server và Printer. Trong đó ATM UI sử dụng chức năng của các thành phần còn lại để vận hành hệ thống.

*** Các thành phần của Component Diagram**

- ✓ **Component:** là một thành phần phần mềm được đóng gói độc lập, nó có thể được triển khai độc lập trên hệ thống và có khả năng tương tác với các thành phần khác khi thực hiện các chức năng của hệ thống.



- ✓ *Component Dependence*: thể hiện quan hệ giữa các thành phần với nhau. Các thành phần phần mềm luôn cần sử dụng một số chức năng ở các thành phần khác trong hệ thống nên quan hệ Dependence được sử dụng thường xuyên.



* *Ứng dụng của Component Diagram Component*

- ✓ Thể hiện cấu trúc của hệ thống
- ✓ Cung cấp đầu vào cho bản vẽ Deployment
- ✓ Hỗ trợ cho việc thiết kế kiến trúc phần mềm

* *Xây dựng Component Diagram*

- ✓ **Bước 1**: Chia hệ thống thành những SubSystem
- ✓ **Bước 2**: Xác định các thành phần và vẽ

* *Hướng dẫn thực hành*

“Một công ty chuyên kinh doanh về các thiết bị điện tử và công nghệ thông tin trong nhiều năm nay và đã có một lượng khách hàng nhất định. Để mở rộng hoạt động kinh doanh của mình, công ty mong muốn xây dựng một hệ thống thương mại điện tử nhằm mở rộng phạm vi kinh doanh trên mạng Internet. Hệ thống mới phải đảm bảo cho khách hàng viếng thăm Website dễ dàng lựa chọn các sản phẩm, xem các khuyến mãi cũng như mua hàng. Việc thanh toán có thể được thực hiện qua mạng hoặc thanh toán trực tiếp tại cửa hàng. Khách hàng có thể nhận hàng tại cửa hàng hoặc sử dụng dịch vụ chuyển hàng có phí của công ty. Ngoài ra, hệ thống cũng cần có phân hệ để đảm bảo cho công ty quản lý các hoạt động kinh doanh như số lượng hàng có trong kho, quản lý đơn đặt hàng, tình trạng giao hàng, thanh toán v.v...”. Thông tin chi tiết các chức năng các bạn có thể tham khảo thêm tại các Website bán hàng

Bước 1: Chia hệ thống thành các SubSystem:

Chia phần Website phục vụ cho đối tượng bên ngoài công ty là Guest và Customer ra một gói riêng để dễ triển khai và bảo mật. Thành phần này gọi là EcommerceWeb.

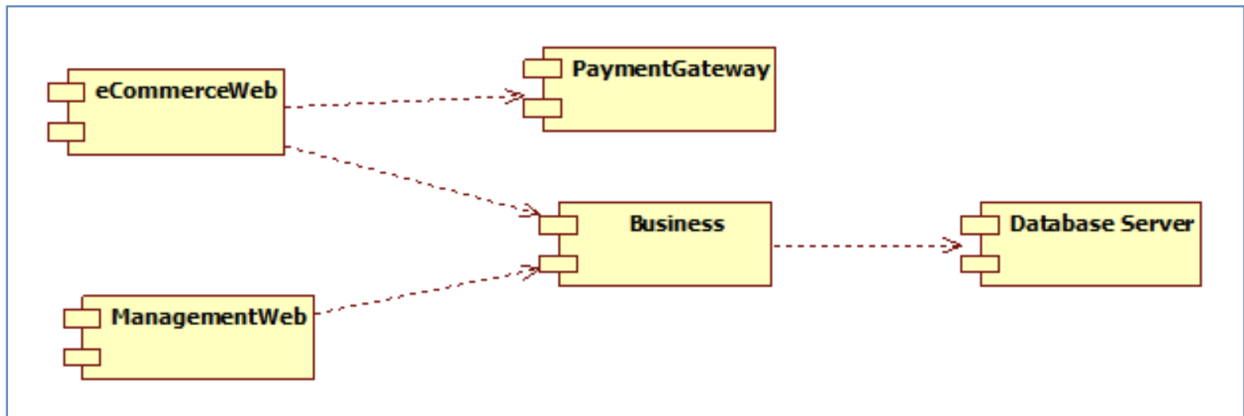
- ✓ Phần Website phục vụ cho đối tượng bên trong công ty được chia thành một gói gọi là ManagementWeb.
- ✓ Phần Bussiness được sử dụng để tương tác CSDL và xử lý các nghiệp vụ.
- ✓ Phần PaymentGateway để xử lý thanh toán trực tuyến.
- ✓ Phần Database Server cũng được tách ra một gói riêng.

Lưu ý: việc phân chia này tùy thuộc vào nhu cầu tổ chức phát triển và triển khai của hệ thống, vì vậy cần có kinh nghiệm về kiến trúc hệ thống khi tham gia thiết kế bản vẽ này.

Bước 2: Xác định quan hệ và vẽ ta được bản vẽ Component Diagram

Việc chia ra các gói sẽ giúp thuận tiện trong quá trình thiết kế, phát triển và triển khai. Chúng ta có thể triển khai mỗi thành phần trên một Server riêng để tăng năng lực cho hệ thống.

Ví dụ:

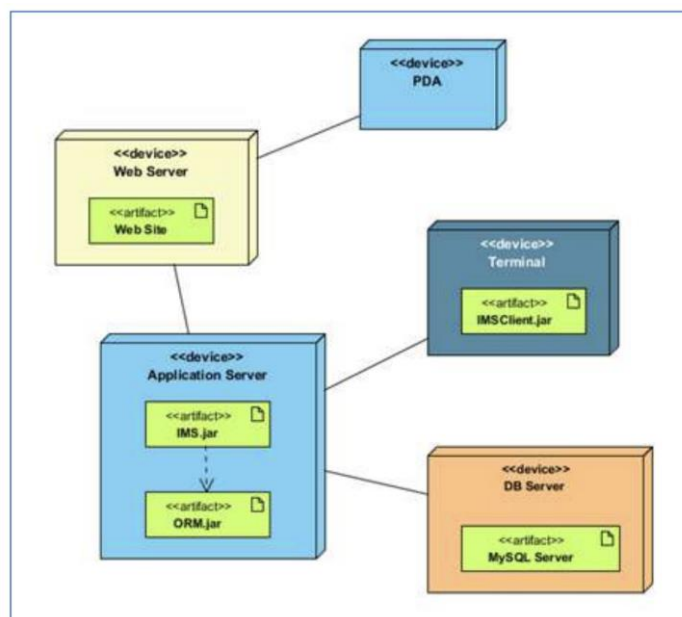


2. Biểu đồ triển khai - Deployment Diagram

Deployment Diagram là bản vẽ giúp chúng ta xác định sẽ triển khai hệ thống phần mềm như thế nào. Đồng thời, xác định việc sẽ đặt các thành phần phần mềm (component) lên hệ thống ra sao.

Deployment Diagram thể hiện rõ kiến trúc triển khai nên nó sẽ ảnh hưởng đến sự thiết kế, phát triển, hiệu năng, khả năng mở rộng của hệ thống v.v...

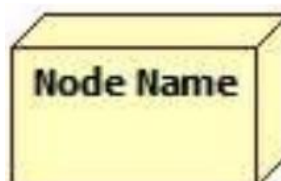
Ví dụ về deployment diagram:



Bản vẽ trên mô tả hệ thống được triển khai trên 03 Server khác nhau gồm Webserver, Application Server, DB server và 02 thiết bị truy cập đầu cuối

* Các thành phần của Deployment Diagram Node

- ✓ *Node*: là một thành phần vật lý, nó có thể là thiết bị phần cứng hoặc một môi trường nào đó mà các thành phần phần mềm được thực hiện.



- ✓ *Relationship*: thể hiện mối quan hệ giữa các node với nhau, Deployment Diagram sử dụng quan hệ Association và Dependence.

Ký hiệu về Association

Ký hiệu về Dependence



Việc xác định quan hệ giữa các thành phần và kết nối chúng lại với nhau chúng ta sẽ có bản vẽ *Deployment Diagram*

* Ứng dụng của *Deployment Diagram*

- ✓ Làm tài liệu để triển khai hệ thống.
- ✓ Sử dụng trong thiết kế kiến trúc cho hệ thống.
- ✓ Dùng trong giao tiếp với khách hàng, các nhà đầu tư.

Cũng như *Component Diagram*, *Deployment diagram* là một bản vẽ khá đơn giản và dễ xây dựng nhưng có ảnh hưởng rất lớn đến quá trình phát triển, triển khai và kinh phí xây dựng dự án.

* Xây dựng *Deployment Diagram*

Bước 1: Xác định các thành phần phần cứng sẽ tham gia vào việc triển khai hệ thống Việc này liên quan đến kiến trúc hệ thống, hiệu năng, khả năng mở rộng và cả vấn đề tài chính và hạ tầng của hệ thống nên bạn cần có kinh nghiệm về kiến trúc hệ thống để làm được việc này.

Bước 2: Xác định các thành phần để triển khai lên các Node Khi đã có phần cứng, bước tiếp theo chúng ta xác định những component liên quan để triển khai trên mỗi node.

Bước 3: Xác định các quan hệ và hoàn tất bản vẽ Xác định các mối quan hệ giữa các thành phần với nhau và nối chúng lại để hoàn tất bản vẽ

* Hướng dẫn thực hành

Xây dựng *Deployment Diagram* cho hệ thống eCommerce

Xem xét hệ thống eCommerce và tiến hành xây dựng bản vẽ *Deployment Diagram* cho hệ thống này. Thực hiện các bước sau đây:

Bước 1: Xác định các Node và bố trí các thành phần lên node

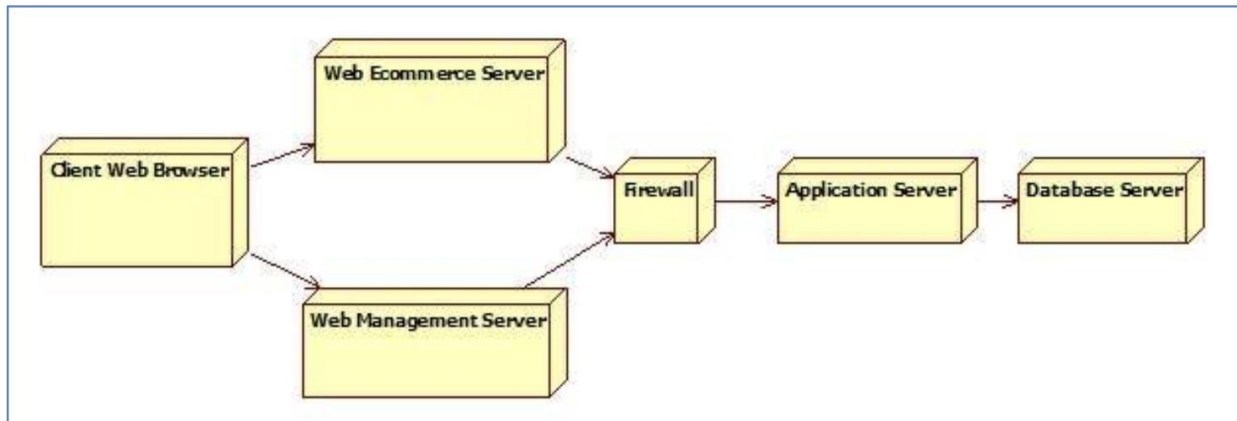
- ✓ Để tăng cường an ninh và sức chịu đựng cho hệ thống chúng ta bố trí phần cho người dùng bên ngoài công ty (Guest và Customer) ra một Server riêng gọi là Web eCommerce Server.
- ✓ Website chứa phần tương tác với nhân viên công ty đặt lên một Node riêng gọi là Web Management Server.
- ✓ Phần Bussiness được đưa ra một Server ứng dụng gọi là Application Server.
- ✓ Database được đặt lên một Server gọi là Database Server.
- ✓ Phần PaymentGateWay có thể được đặt trên Web eCommerce Server

- ✓ Bổ sung thêm các thiết bị bảo mật và hạ tầng

Bước 2: Xác định quan hệ giữa các thành phần và hoàn tất bản vẽ.

Xem xét các thành phần gọi với nhau để hoàn tất chức năng, chúng ta sẽ xác định các quan hệ của chúng.

Biểu diễn lên bản vẽ chúng ta sẽ có Deployment Diagram cho hệ thống eCommerce như sau:



Nếu theo mô hình này, hệ thống cần 4 Server và 01 thiết bị mạng là Firewall để triển khai nhằm đảm bảo an ninh, sức chịu đựng khi có đông người sử dụng và khả năng mở rộng hệ thống.

Tuy nhiên, nếu hệ thống có ít người sử dụng và cần tiết kiệm chi phí, có thể triển khai trên một hoặc hai Server chứ không cần thiết phải triển khai trên nhiều Server như mô hình trên.

Phần 2. Thực hành

Cho HỆ THỐNG QUẢN LÝ THƯ VIỆN với mô tả chức năng và yêu cầu như sau:

Xây dựng hệ thống quản lý thư viện cho trường Đại học gồm các hoạt động quản lý thông tin sách, quản lý thông tin độc giả, quản lý hoạt động mượn trả sách:

Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.

Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngaynhap). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.

Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).

Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).

Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.

Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.

Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.

Người dùng hệ thống phải đăng nhập trước khi thực hiện.

Yêu cầu:

1. Xây dựng biểu đồ thành phần của hệ thống quản lý thư viện
2. Xây dựng biểu đồ triển khai của hệ thống quản lý thư viện

Hướng dẫn thực hiện

1. Xây dựng biểu đồ thành phần của hệ thống quản lý thư viện

*** Phân tích bài toán**

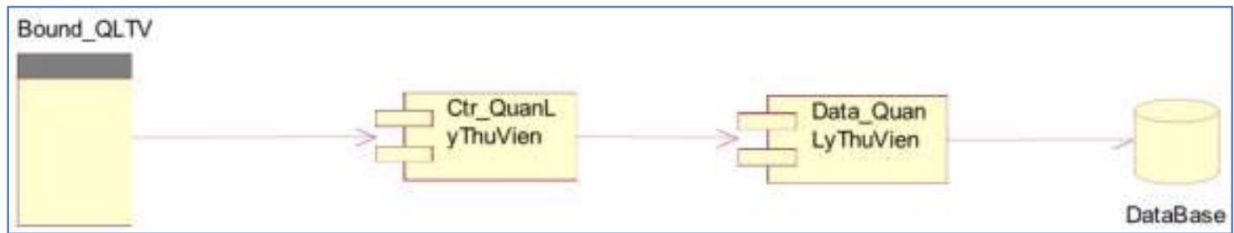
Bước 1: Dựa vào kiến trúc hệ thống phần mềm sẽ triển khai, xác định các thành phần của hệ thống. Xác định các lớp, các gói thuộc mỗi một thành phần (*Một thành phần có thể gồm nhiều thành phần nhỏ bên trong, tùy thuộc vào quy mô của dự án*).

Ví dụ trong hệ thống quản lý thư viện chọn kiến trúc tập trung dữ liệu, Multi layer có thể gồm các gói sau:

- ✓ Bound_QuanLyThuVien: gồm các lớp bound thuộc tầng trình diễn.
- ✓ Ctr_QuanLyThuVien: gồm các lớp control đảm nhận nghiệp vụ các UC
- ✓ DaTa_QuanLyThuVien: gồm các lớp đảm nhận nhiệm vụ tương tác với cơ sở dữ liệu.
- ✓ Entity_QuanLyThuVien: gồm các lớp mô tả cấu trúc dữ liệu

Bước 2:

Xác định mối quan hệ phụ thuộc giữa các thành phần (*thành phần A phụ thuộc vào thành phần B nếu trong A có ít nhất một lớp sử dụng dịch vụ của một lớp khác trong thành phần B*).

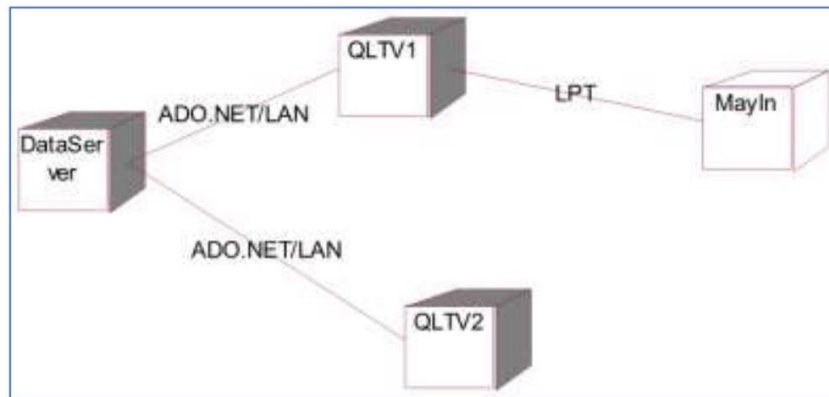


2. Xây dựng biểu đồ triển khai hệ thống.

* Phân tích bài toán:

- ✓ Dựa trên kiến trúc hệ thống, xác định các Node triển khai.
- ✓ 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.

Ví dụ: các dạng truyền thông giữa các hệ thống như: TCP/IP, hồng ngoại, Bluetooth... với hệ thống quản lý thư viện trên, lựa chọn kiến trúc tập trung dữ liệu → có biểu đồ triển khai như sau:



Bài tập tự thực hiện

Lựa chọn kiến trúc triển khai khác, xây dựng lại biểu đồ thành phần và biểu đồ triển khai cho hệ thống quản lý thư viện.