

Chương 5

Ngôn ngữ mô hình hóa hợp nhất - UML

Giáo viên

: ThS. Trần Văn Thọ

Đơn vị

: Bộ môn KTHT & MMT

1. Giới thiệu

- Mô hình hóa: là mô tả bản chất một bài toán (một vấn đề, một hệ thống, cấu trúc...) một cách trừu tượng bằng cách sử dụng các mô hình.
- UML (Unified Modeling Language):
 - + Là một ngôn ngữ mô hình đa năng sử dụng để mô tả hệ thống phần mềm một cách trực quan .
 - + Sử dụng một tập hợp các ký hiệu cùng các quy tắc kết hợp để tạo nên các biểu đồ.
 - + Đã được thống nhất thành một chuẩn chung cho việc thiết kế hệ thống theo phương pháp hướng đối tượng

1. Giới thiệu

- Vai trò của UML:
 - Trực quan hóa hệ thống:
 - + UML là ngôn ngữ trực quan:
 - Sử dụng các hình khối và các đường kết nối đơn giản, quen thuộc để biểu diễn các biểu đồ
 - + Giúp công việc phát triển được xử lý nhất quán, giảm thiểu lỗi xảy ra
 - Dễ hình dung hơn cấu trúc của hệ thống
 - Hiệu quả hơn trong việc liên lạc, trao đổi
 - Trong tổ chức
 - Bên ngoài tổ chức

1. Giới thiệu

- **Vai trò của UML:**

- Chi tiết hóa
 - + UML tài liệu hóa mọi khía cạnh của sản phẩm phần mềm: kiến trúc, yêu cầu, kiểm thử, lập kế hoạch, quản lý bàn giao
- Chuẩn hóa tài liệu thiết kế
 - + Từ 2005, UML 2 được chuẩn hóa theo tiêu chuẩn ISO
- Sinh mã tự động
 - + Các mô hình UML phù hợp và có thể kết nối ánh xạ sang nhiều ngôn ngữ và nền tảng lập trình

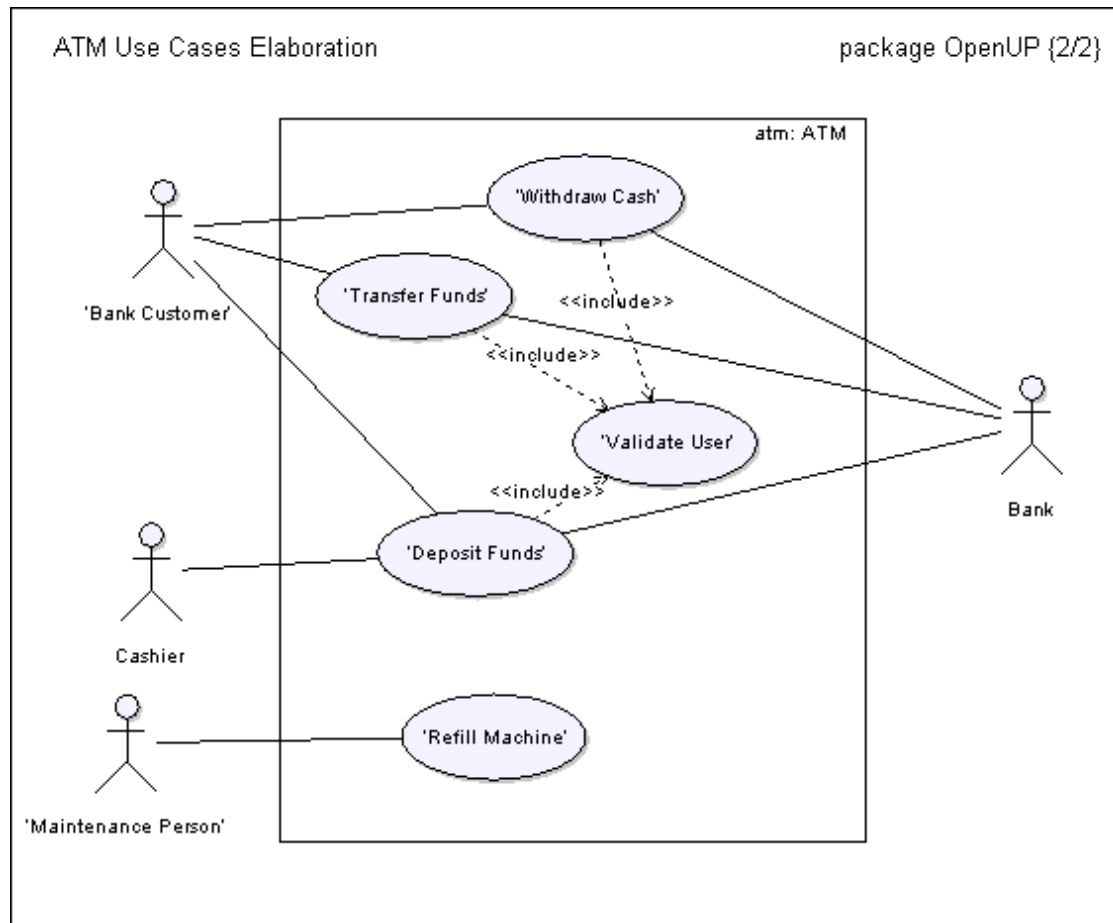
2. Các loại biểu đồ trong UML

- **Nhóm các biểu đồ ở góc nhìn cấu trúc:**
 - + Biểu đồ lớp (Class Diagram)
 - + Biểu đồ đối tượng
 - + Biểu đồ thành phần
 - + Biểu đồ triển khai
- **Nhóm các biểu đồ hành vi:**
 - + Biểu đồ Use case
 - + Biểu đồ hoạt động
 - + Biểu đồ trạng thái
 - + Biểu đồ tuần tự

3. Biểu đồ tình huống sử dụng (Use case)

- Thể hiện một tập hợp các tình huống và chủ thể cũng như mối quan hệ giữa chúng
- Minh họa góc nhìn tình huống tĩnh của hệ thống

3. Biểu đồ Use case

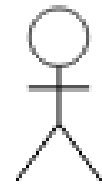
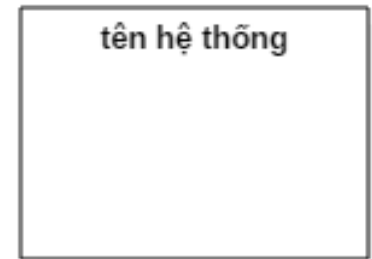


Biểu đồ Use case

3. Biểu đồ tình huống sử dụng (Use case)

3.1 Các biểu tượng dùng trong biểu đồ Tình huống sử dụng (use case)

- Phạm vi của hệ thống: được thể hiện bởi 1 hình chữ nhật, nó biểu diễn giới hạn của hệ thống.
- Tình huống sử dụng (use case): là một hình elíp, tên của tình huống được viết bên trong đường elíp này
- Actor : Chủ thể tương tác với hệ thống, có thể là người dùng hoặc 1 hệ thống bên ngoài nào đó

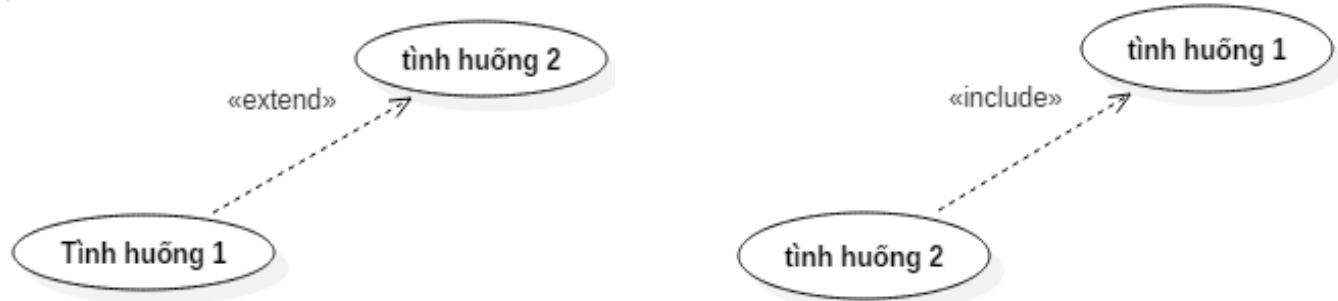
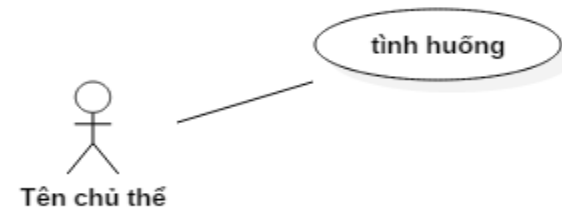


Tên chủ thể

3. Biểu đồ tình huống sử dụng (Use case)

- **Các đường nối:**

- Nói giữa Chủ thể với tình huống sử dụng
- Đường nối giữa chủ thể với tình huống không có nghĩa là chủ thể phải là người bắt đầu các tình huống này. Nó chỉ có nghĩa là giữa chủ thể và tình huống đó có liên quan với nhau.
- Nói giữa các tình huống sử dụng với nhau
- Đường nối thể hiện mối quan hệ <<include>> / <<extend>>



3. Biểu đồ tình huống sử dụng (Use case)

3.2. Mối quan hệ giữa các tình huống sử dụng

- Một tình huống sử dụng thể hiện một chức năng của hệ thống, do đó mối quan hệ giữa các tình huống cho biết mối quan hệ giữa các chức năng của hệ thống.
- Mối quan hệ này là tĩnh, nói cách khác việc diễn tả này không mang ý nghĩa về thứ tự thực thi (hay là chức năng nào có trước, chức năng nào có sau)
- Mối quan hệ tĩnh giữa 2 chức năng cho biết rằng một chức năng (hay tình huống) bao gồm hành vi của chức năng kia.
 - + Dạng thứ nhất: 1 tình huống luôn luôn bao gồm tình huống kia
 - + Dạng thứ hai: 1 tình huống đôi khi bao gồm tình huống kia.

3. Biểu đồ tình huống sử dụng (Use case)

a. Mối quan hệ sử dụng (Use Relationship)

Được dùng khi một tình huống (hay chức năng) nào đó của hệ thống luôn luôn bao gồm một tình huống khác.



3. Biểu đồ tình huống sử dụng (Use case)

b. Mối quan hệ mở rộng (Extend Relationship)

- Được dùng khi một tình huống (hay chức năng) nào đó của hệ thống đôi khi bao gồm một tình huống khác.
- Nói cách khác, tùy thuộc vào một số điều kiện nào đó mà một tình huống có thể kéo theo việc xảy ra một tình huống khác.
- Ví dụ: Xét một ứng dụng bán hàng trực tuyến. Tình huống Tạo một đơn hàng (Place Order): mô tả việc yêu cầu một đơn hàng mới. Tình huống Thêm khách hàng (Add Customer): mô tả việc thêm vào một khách hàng mới vào trong cơ sở dữ liệu về khách hàng. Tình huống này sẽ xảy ra nếu vào thời điểm khách hàng đặt một hóa đơn mua hàng, thông tin về khách hàng này vẫn chưa có trong cơ sở dữ liệu về khách hàng.



3. Biểu đồ tình huống sử dụng (Use case)

3.4 Các ưu điểm của biểu đồ tình huống

- Các tình huống thể hiện các yêu cầu cụ thể như là chuỗi các tương tác giữa hệ thống với một hay nhiều chủ thể.
- Mỗi liên hệ giúp mô tả các chức năng dự định của hệ thống đối với người sử dụng
- Trong hệ thống phức tạp, biểu đồ tình huống cung cấp một điểm xuất phát cho việc phân tích hệ thống theo chiều hướng là những chức năng nào cần được xây dựng
- Biểu đồ tình huống cung cấp nguồn gốc cho các dịch vụ cung cấp bởi hệ thống và giúp xác định các lớp mà hệ thống cần phải cài đặt.

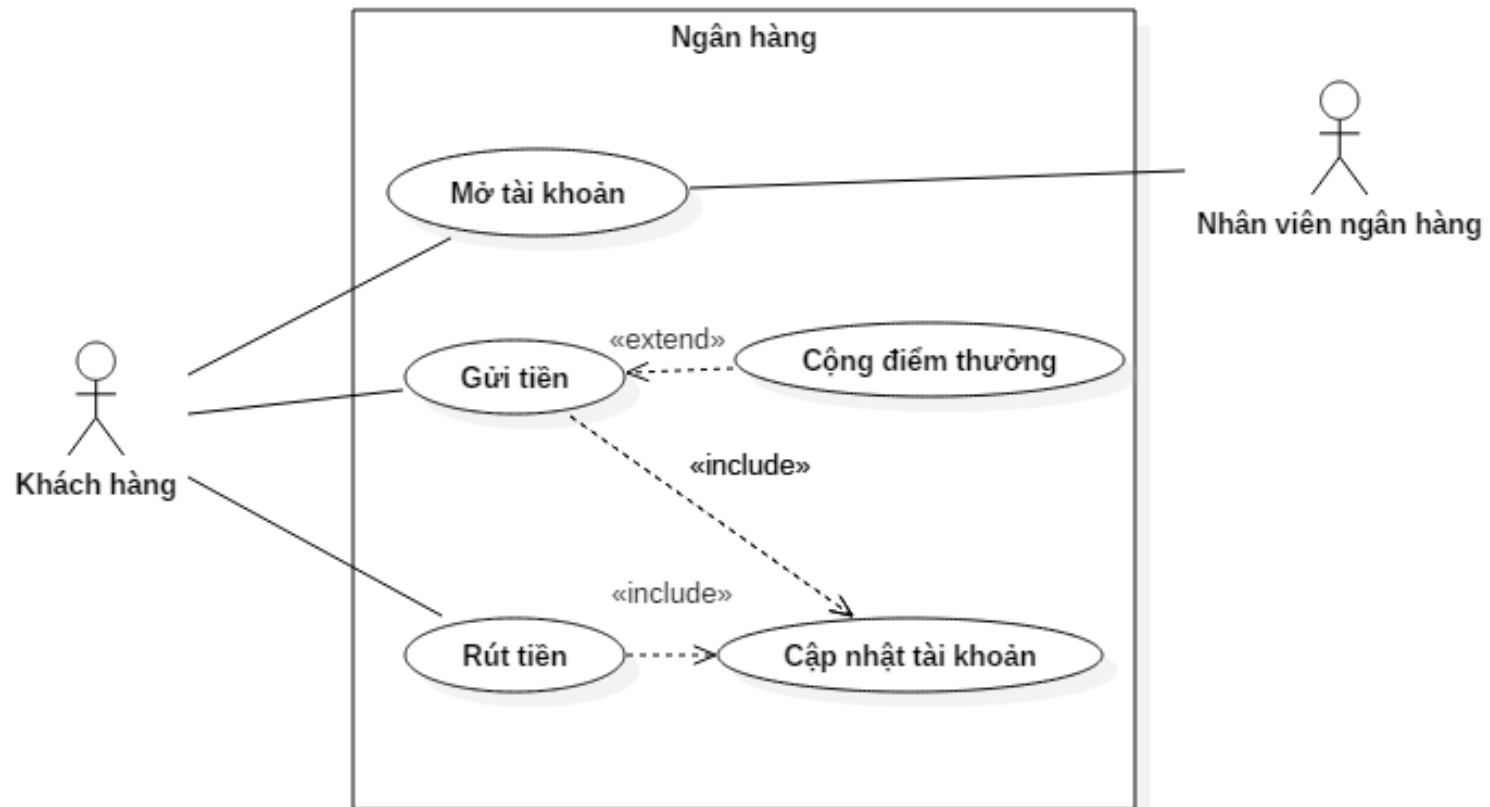
3. Biểu đồ tình huống sử dụng (Use case)

3.5 Các bước xây dựng biểu đồ tình huống

- Xác định Chủ thể (Actor) và Hệ thống
 - + Chủ thể là gì ?, có bao nhiêu chủ thể tương tác với hệ thống?
- Xác định các Tình huống (Use case):
 - + Xác định các tình huống ở mức cao nhất , chính là các chức năng tương tác trực tiếp với chủ thể
 - + Tiếp tục xác định tiếp các Tình huống ở mức chi tiết hơn, có thể mở rộng thêm Chủ thể mới nếu cần.
- Xác định mối liên hệ giữa các Tình huống:
 - + Xác định các liên hệ Sử dụng (thường là các Tình huống được sử dụng chung như Login, checkValid..)
 - + Xác định các liên hệ mở rộng

3. Biểu đồ tình huống sử dụng (Use case)

Ví dụ:



3. Biểu đồ tình huống sử dụng (Use case)

3.6 Đặc tả các Use case

Một use case thường phải đi kèm với phần đặc tả cụ thể nội dung của Use case đó. Nội dung phần đặc tả thường có các phần sau:

- + Tên Use case: gồm tên của Use case viết dưới dạng đầy đủ kèm theo mã của Use case.
- + Mô tả: sơ lược về nội dung của use case
- + Actor: các chủ thể tham gia vào use case
- + Điều kiện bắt đầu
- + Luồng chính: các hoạt động diễn ra tuần tự giữa chủ thể và hệ thống thực hiện luồng công việc chính, có đánh thứ tự cho mỗi hoạt động.
- + Luồng phụ: các hoạt động xảy ra không nằm trong luồng chính, thường là các phát sinh bất thường do lỗi. Được ký hiệu bắt đầu từ số thứ tự của hoạt động phát sinh luồng phụ.
- + Điều kiện sau: trạng thái của hệ thống ngay sau khi Use case hoàn thành

3. Biểu đồ tình huống sử dụng (Use case)

3.6 Đặc tả các Use case

Use Case 1.1 *Withdraw money from an ATM*

Withdraw money from an ATM

Primary actors: Customer
ATM Technician
Bank

Preconditions: Network connection is active
ATM has available cash

Basic flow of events:

1. Bank customer inserts debit card and enters PIN.
2. Customer is validated.
3. ATM displays actions available on ATM unit. Customer selects Withdraw Cash.
4. ATM prompts account.
5. Customer selects account.
6. ATM prompts amount.
7. Customer enters desired amount.
8. Information sent to Bank, inquiring if sufficient funds/allowable withdrawal limit.
9. Money is dispensed and receipt prints.

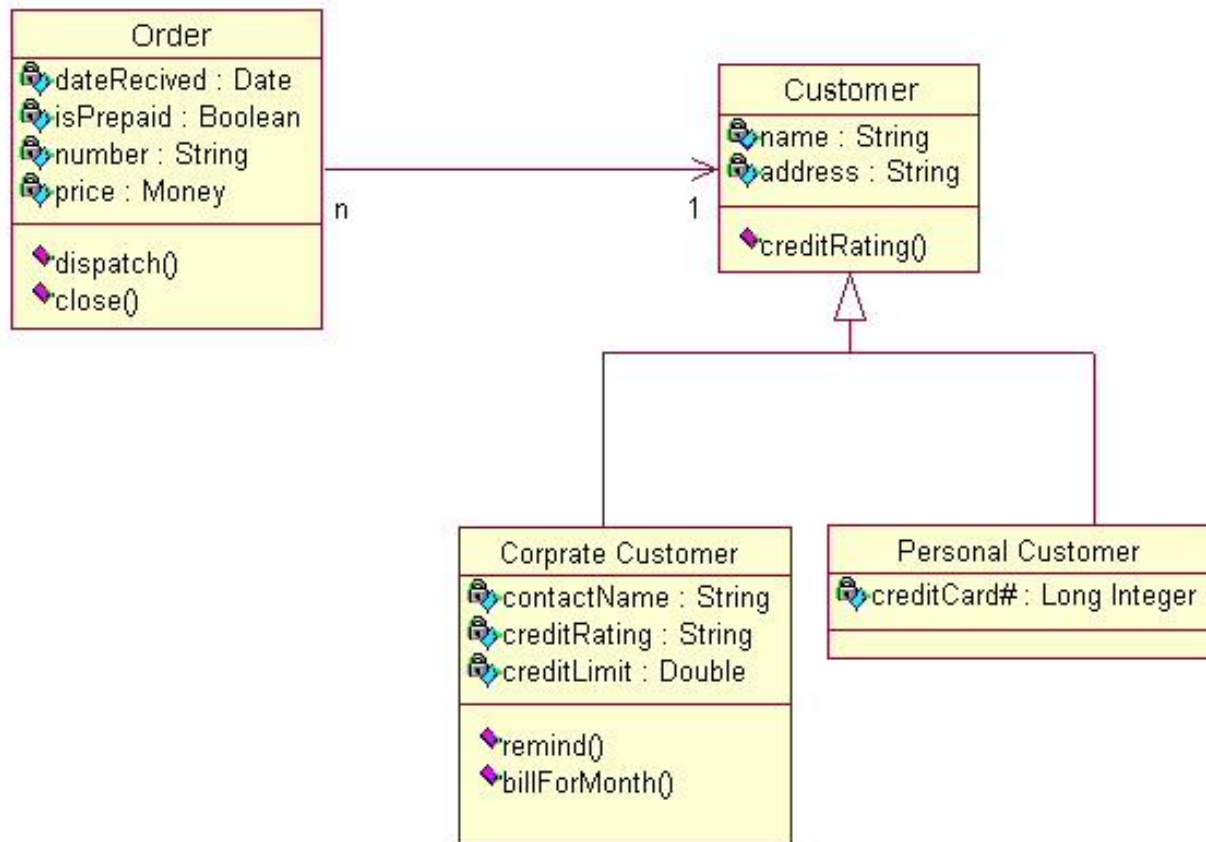
Alternative flows

- 2a. Customer is not validated.
 - 2a1. ATM displays error message.
- 7a. Customer selects invalid amount.
 - 7a1. ATM prompts user to re-enter valid amount.
- 8a. Customer has insufficient funds.
 - 8a1. ATM displays error message.
 - 8a2. ATM shows available withdrawal limit, redirects to step 6.
- 9a. ATM has insufficient cash.
 - 9a1. ATM Technician is alerted.
 - 9a2. ATM displays error message and phone number to call.
- 9b. Cash gets stuck in dispensing.
 - 9b1. ATM displays error message.

4. Biểu đồ Lớp (Class Diagram)

- Thể hiện một tập hợp các **lớp**, giao diện và quan hệ giữa chúng.
- Đây là biểu đồ phổ biến nhất và quan trọng nhất trong việc mô hình hóa các hệ thống hướng đối tượng.
- Chỉ ra các đối tượng trong hệ thống và mối quan hệ logic giữa chúng
- Sử dụng để minh họa góc nhìn thiết kế tĩnh của hệ thống, hỗ trợ thiết kế các chức năng của hệ thống

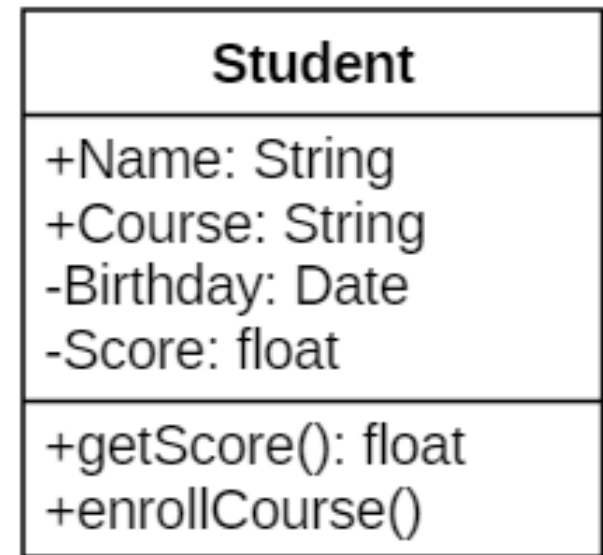
4. Biểu đồ lớp (Class Diagram)



4. Biểu đồ Lớp (Class Diagram)

4.1 Ký hiệu Lớp:

- Là một hình chữ nhật gồm 3 phần:
 - + Tên lớp
 - + Các thuộc tính của lớp: gồm tên thuộc tính và kiểu dữ liệu
 - + Các phương thức của lớp
- Các ký hiệu cho thuộc tính truy cập:
 - + Public: +
 - + Private: -
 - + Protected : #



4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp:

- Kết hợp (Association)
- Tích tụ (Aggregation)
- Kết cấu (Composition)
- Thừa kế (Generation)

4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp:

a. Kết hợp (Association relationship):

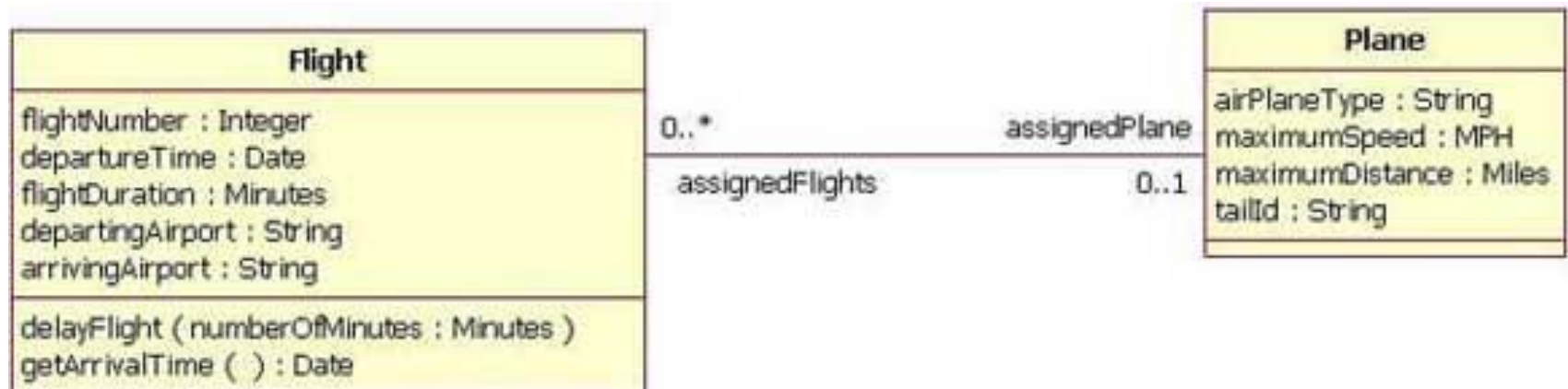
- Chỉ sự liên quan giữa các đối tượng với nhau
- Có 2 kiểu kết hợp:
 - + Kết hợp hai chiều: cả hai lớp đều biết về mối liên hệ giữa 2 lớp
 - + Kết hợp một chiều: chỉ một lớp biết về mối liên hệ giữa 2 lớp
 - + Thường có ghi chú về số lượng đối với liên kết lớp:
 - 1 :một và chỉ một
 - 0..1 :không hay một
 - m..n :từ m tới n (m và n là các số tự nhiên)
 - 0..* hay * :từ 0 tới nhiều
 - 1..* :một tới nhiều

4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp:

a. Kết hợp (Association relationship):

VD: kết hợp 2 chiều

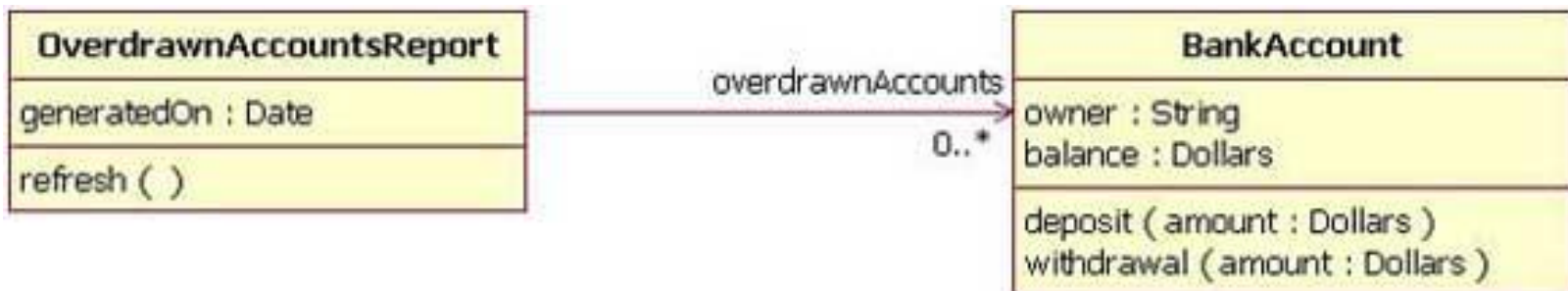


4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp.

a. Kết hợp (Association relationship):

- Ví dụ: kết hợp 1 chiều



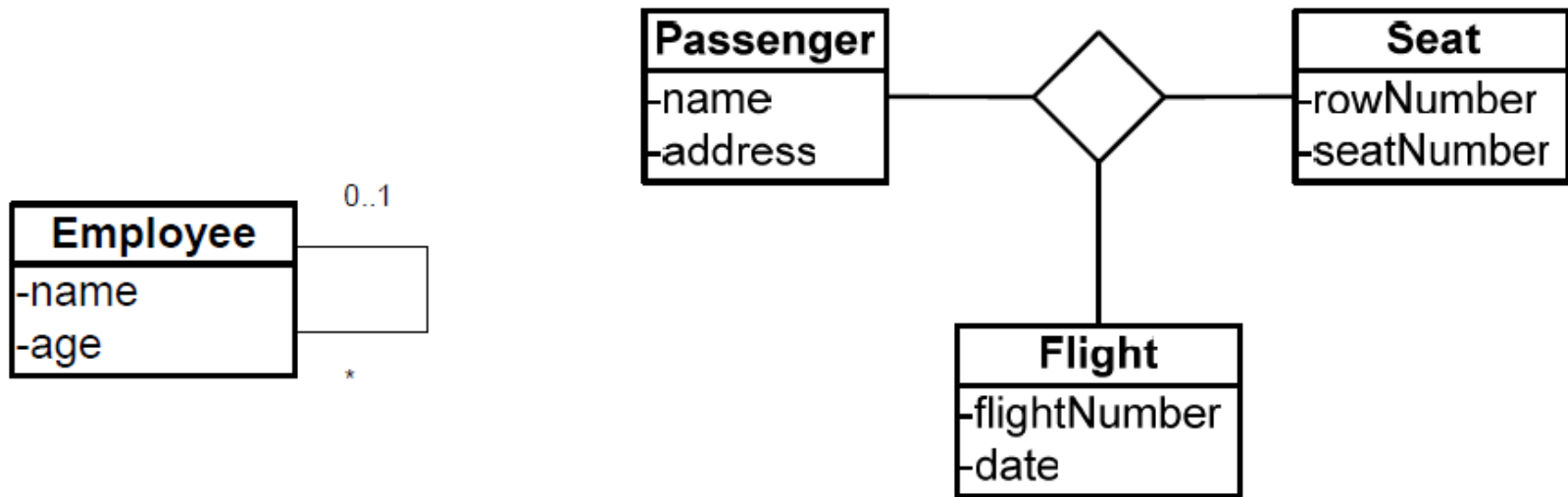
4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp

a. Kết hợp (Association relationship):

Liên kết và tự liên kết: UML cho phép tạo ra liên kết ở bất kỳ mức độ nào. Với trường hợp tự liên kết, nó cho biết hai đối tượng cùng lớp có thể có liên hệ với nhau.

Lớp kết hợp: lớp được thêm vào để chứa các thông tin về mối quan hệ



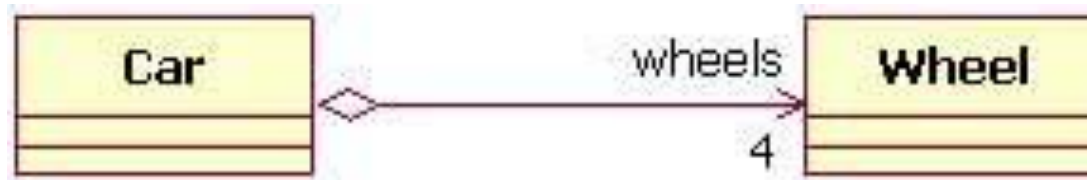
4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp

b. Tích tụ (Aggregation relationship):

- Là một dạng đặc biệt của kết hợp, biểu diễn quan hệ **một-phần-của**
- Dòng dữ liệu đi từ lớp tổng thể sang lớp thành phần. Một lớp thành phần có thể thuộc về một hoặc nhiều lớp tổng thể
- Vòng đời của đối tượng thành phần độc lập với vòng đời của đối tượng tổng thể
- Biểu diễn bằng mũi tên nét liền có đầu là hình thoi rỗng

Ví dụ: quan hệ giữa xe và bánh xe



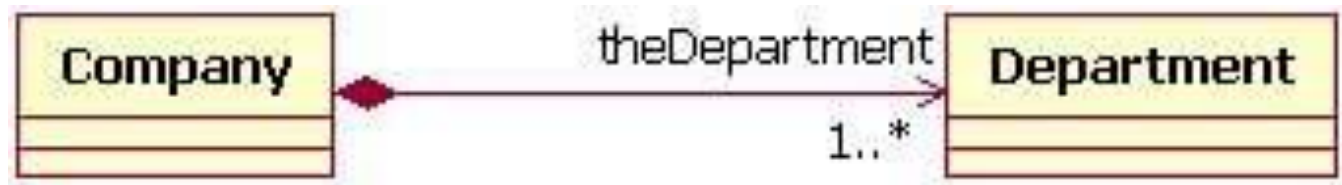
4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp

c. **Kết cấu**(Composition relationship):

- Là một dạng đặc biệt của tích tụ, biểu diễn quan hệ **thành phần** mạnh hơn so với tích tụ
- Vòng đời đối tượng thành phần phụ thuộc vào đối tượng tổng thể, nếu đối tượng tổng thể bị xóa thì đối tượng thành phần cũng phải bị xóa theo
- Biểu diễn bằng mũi tên nét liền có đầu là hình thoi đặc

VD: quan hệ giữa công ty và phòng ban

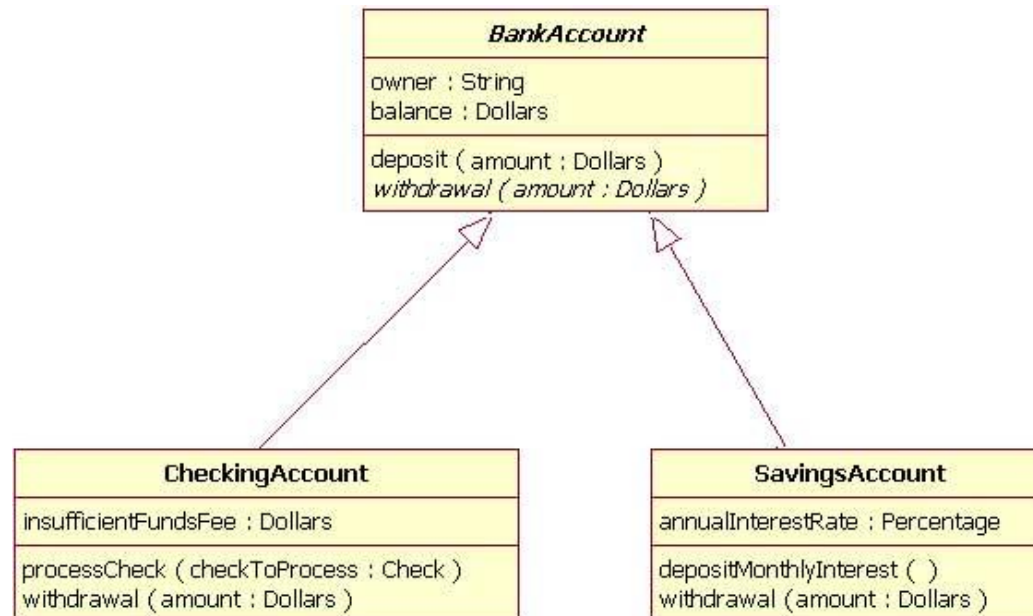


4. Biểu đồ Lớp (Class Diagram)

4.2 Các liên kết giữa các Lớp:

d. Thừa kế (Generalization relationship):

- Biểu diễn quan hệ thừa kế giữa các lớp
- Thể hiện bằng mũi tên nét liền có đầu là hình tam giác nổi từ lớp con lên lớp cha



4. Biểu đồ Lớp (Class Diagram)

4.3 Các bước triển khai biểu đồ Lớp

- Xuất phát từ các phát biểu bài toán và phân tích hoạt động của hệ thống, tìm ra các **đối tượng** dữ liệu trong hệ thống và các **mối quan hệ** giữa chúng.
 - + Các nguồn thông tin có thể phân tích để phát hiện các đối tượng: mô tả yêu cầu hệ thống, biểu đồ use case, kinh nghiệm từ các hệ thống tương tự, ý kiến chuyên gia,...
- Lập biểu đồ Domain Model (một dạng biểu đồ lớp sơ lược) để biểu diễn các đối tượng và các quan hệ
- Chi tiết hóa biểu đồ domain thành biểu đồ lớp nhờ bổ sung các thuộc tính và phương thức cho mỗi lớp

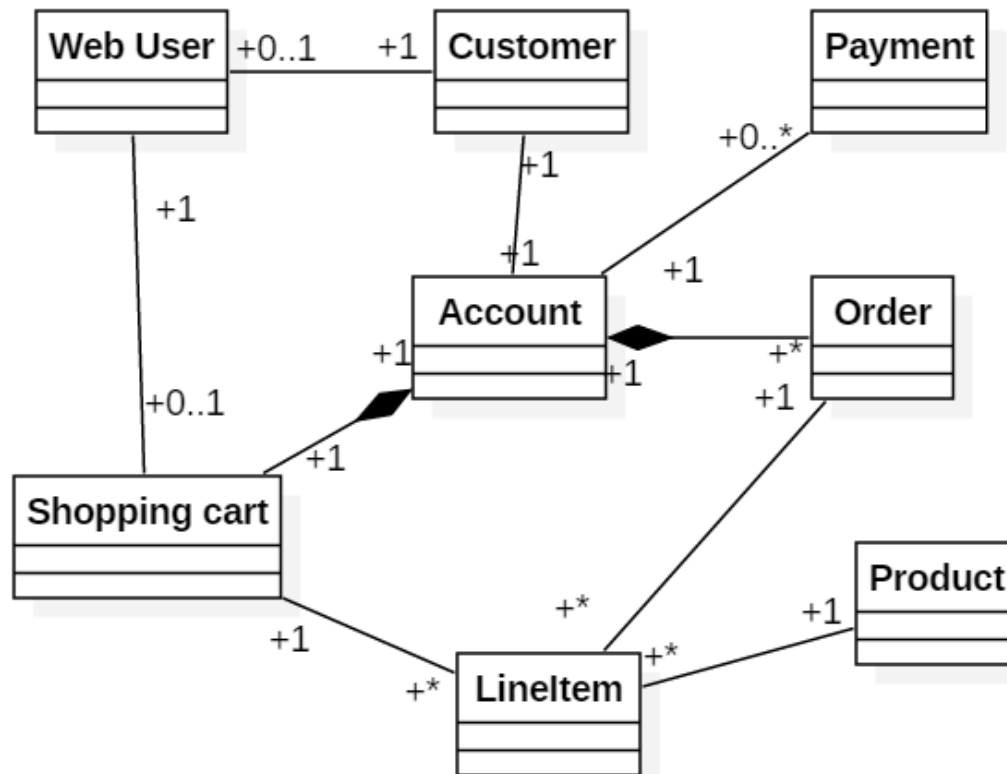
4. Biểu đồ Lớp (Class Diagram)

- Ví dụ:
- Bài toán:

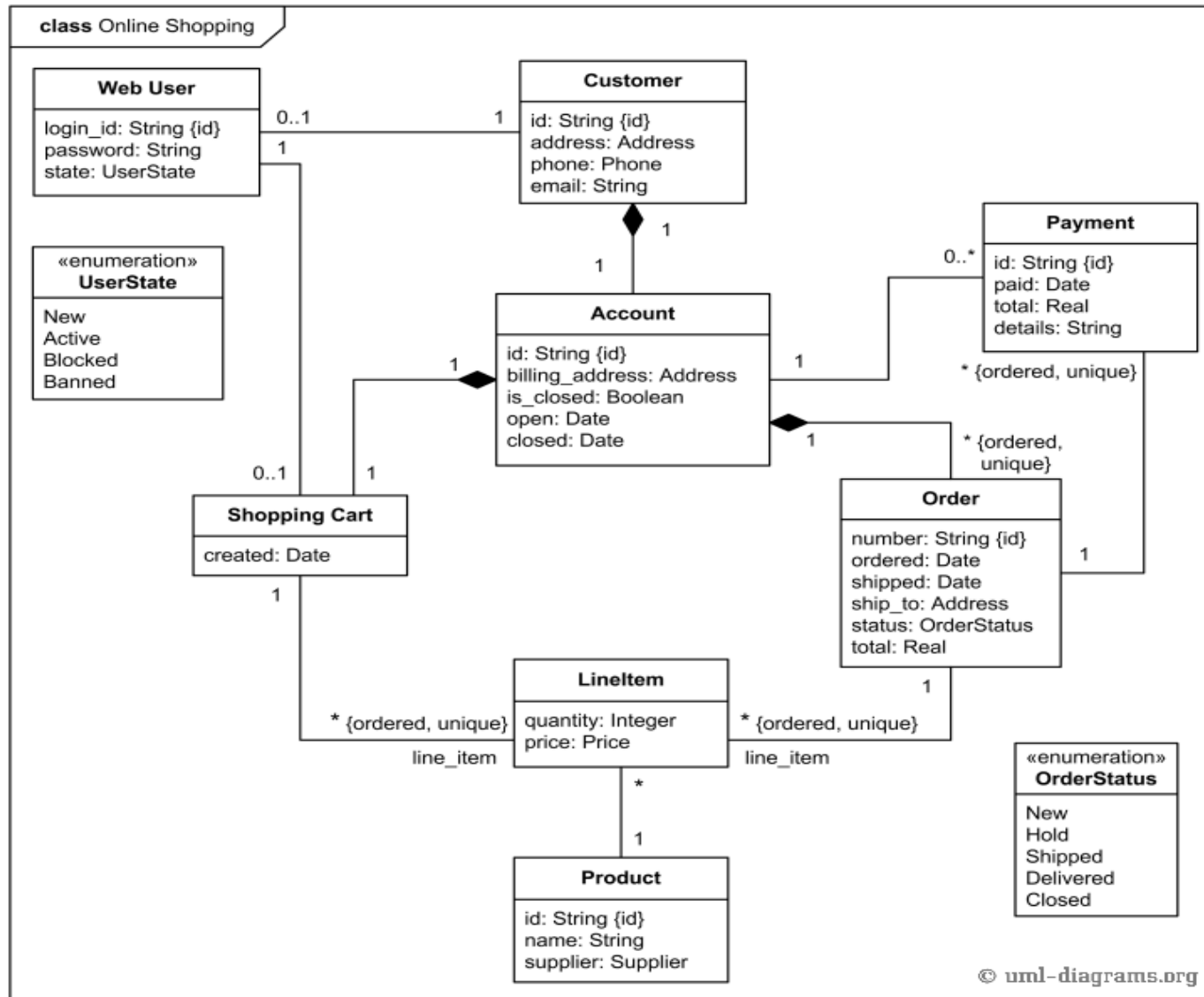
Một cửa hàng bán đồ lưu niệm muốn xây dựng một website bán hàng. Để mua hàng trên web, **khách hàng** phải đăng ký một **tài khoản** trên web. Sau đó, khách hàng có thể lựa chọn các **mặt hàng** để cho vào **giỏ hàng**. Khi thanh toán, khách hàng cần phải hoàn thiện **thông tin cá nhân chi tiết** và thông tin giao hàng. Sau đó một **hóa đơn** được sinh ra trong đó lưu chi tiết thông tin đơn hàng. Sau khi khách hàng **thanh toán**, chủ cửa hàng sẽ giao hàng cho nhân viên đi giao hàng. Khi giao hàng, cửa hàng phải phát ra một **phiếu giao hàng** để nhân viên vận chuyển sau khi giao hàng xong thì yêu cầu khách hàng xác nhận đã nhận hàng.

4. Biểu đồ Lớp (Class Diagram)

- Biểu đồ Domain Model:



4. Biểu đồ Lớp (Class Diagram)



4. Biểu đồ Lớp (Class Diagram)

4.4 Ứng dụng của biểu đồ lớp

- Thể hiện rõ các đối tượng của hệ thống và các mối quan hệ giữa chúng :
 - + Để hiểu rõ hơn các chức năng của hệ thống, cơ sở để xây dựng tiếp các biểu đồ tiếp theo
 - + Cho thấy cấu trúc của hệ thống
- Chỉ rõ các mô tả về thuộc tính và phương thức của các đối tượng:
 - + Coding
 - + Xây dựng cơ sở dữ liệu

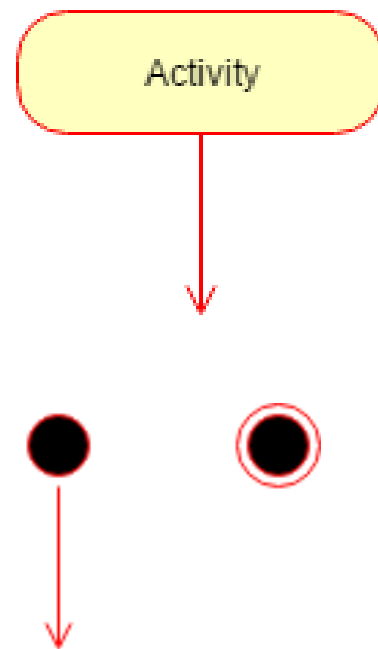
5. Biểu đồ hoạt động (Activity Diagram)

- Mô tả luồng hoạt động:
 - + Bao gồm các hành động nhỏ kết nối với nhau thể hiện thứ tự thực hiện của chúng
- Đối tượng mô tả:
 - + Các hoạt động nghiệp vụ của hệ thống
 - + Các use case

5. Biểu đồ hoạt động (Activity Diagram)

Hệ thống ký hiệu của biểu đồ:

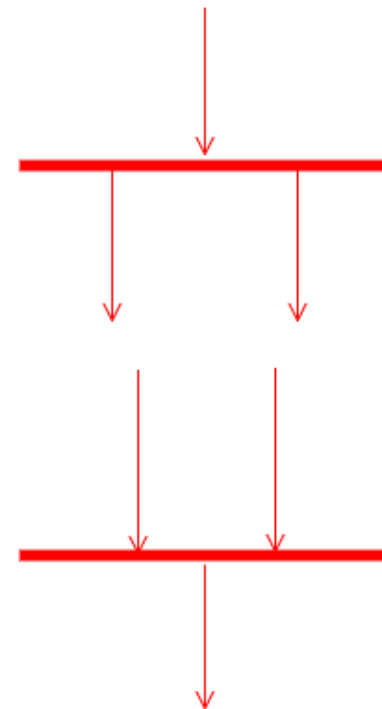
- Hoạt động (Activity): thể hiện một hành động của người hay hệ thống
 - + Tên hoạt động
 - + Mũi tên hướng đến hành động tiếp theo
- Điểm đầu và điểm cuối của luồng hoạt động



5. Biểu đồ hoạt động (Activity Diagram)

Hệ thống ký hiệu của biểu đồ:

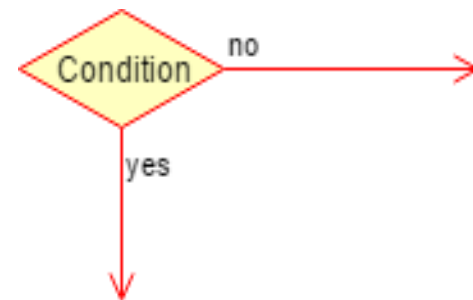
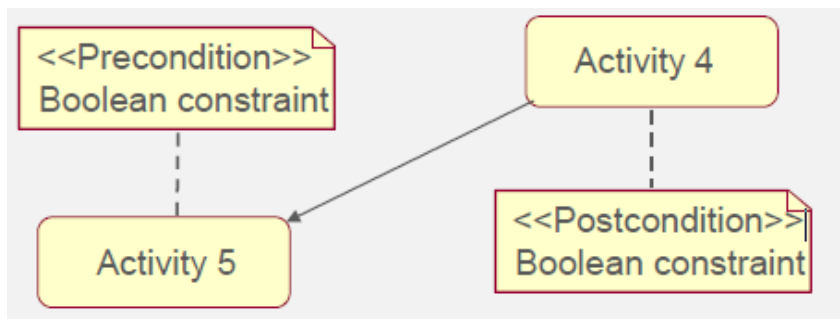
- Thanh đồng bộ :
 - + Phân nhánh: phân chia luồng hiện tại thành các luồng song song
 - + Kết hợp: kết hợp các luồng song song thành một luồng



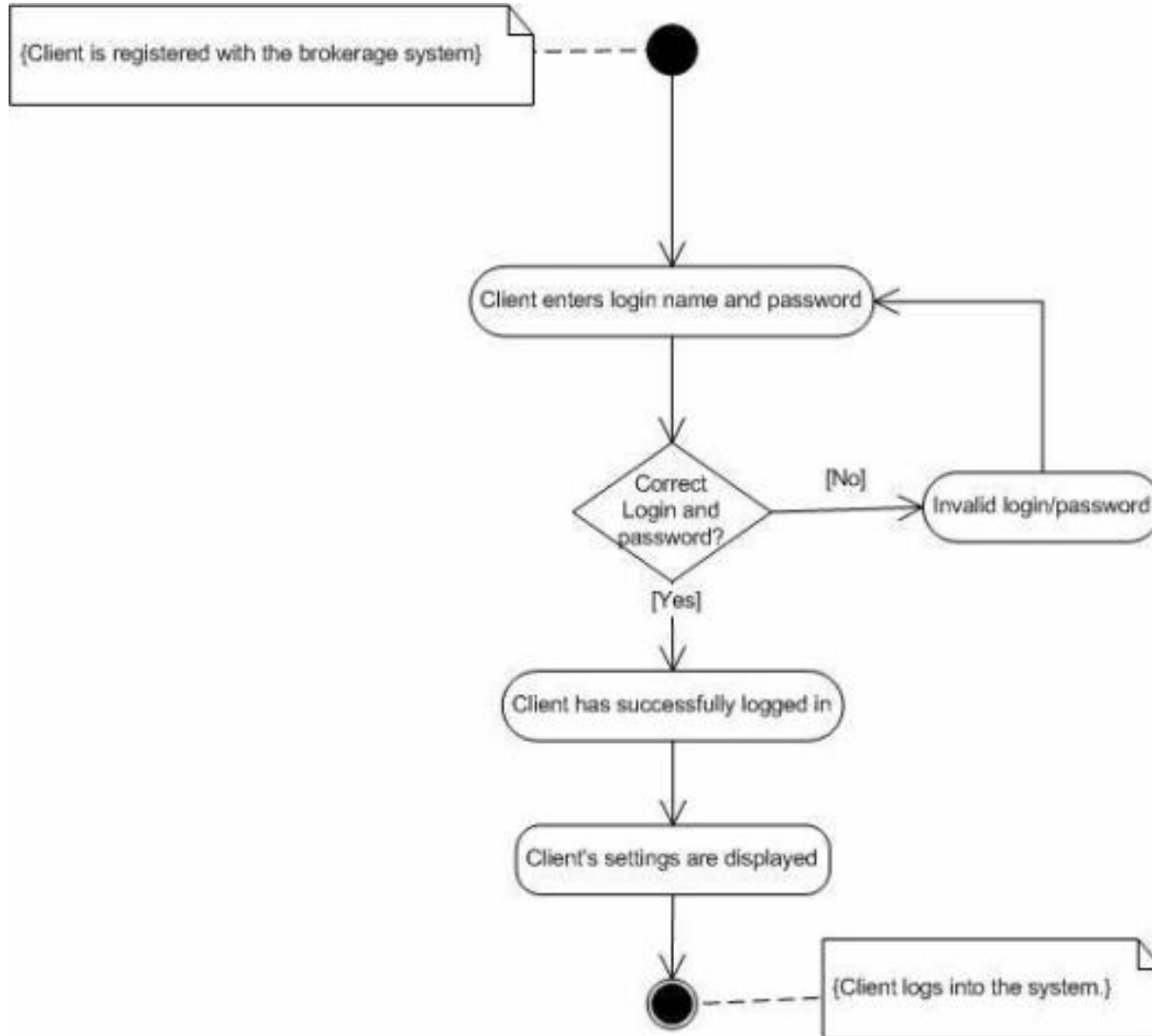
5. Biểu đồ hoạt động (Activity Diagram)

Hệ thống ký hiệu của biểu đồ:

- Điều kiện phân nhánh: Phân luồng thành 2 luồng dựa trên điều kiện hoặc quyết định
- Các ràng buộc khi bắt đầu luồng và kết thúc luồng (tương ứng với đặc tả use case)

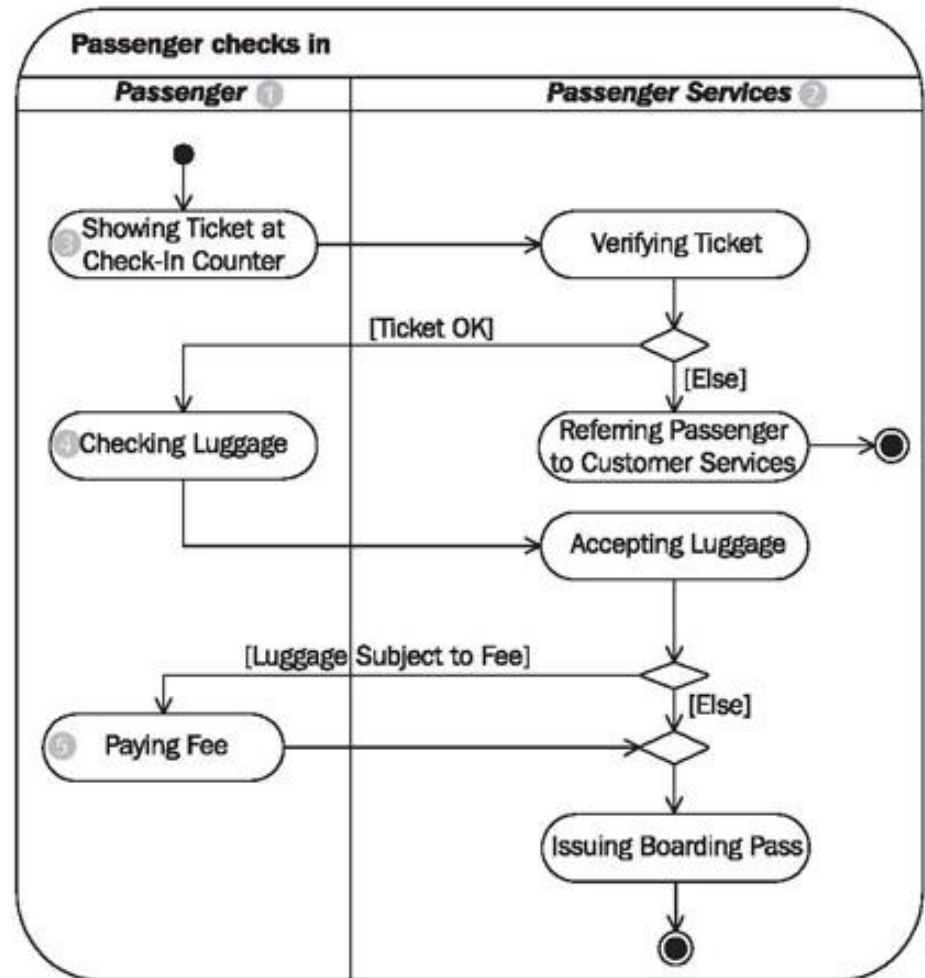


5. Biểu đồ hoạt động (Activity Diagram)



5. Biểu đồ hoạt động (Activity Diagram)

Để chỉ rõ các hành động do ai thực hiện, có thể phân chia biểu đồ thành các miền tương ứng với mỗi đối tượng thực hiện



5. Biểu đồ hoạt động (Activity Diagram)

Cách vẽ biểu đồ hoạt động:

- Xác định hành động của các tác nhân (chủ thể) trong hệ thống:
 - Từ mô tả hệ thống
 - Từ biểu đồ và đặc tả Use case
- Xác định phạm vi cho biểu đồ (giới hạn trong 1 use case , một vài use case hay cả hệ thống)
- Xác định trình tự thực hiện (luồng) của các hoạt động, đặc biệt chú ý đến các điều kiện phân nhánh và kết hợp

Nếu cần, có thể xác định phạm vi cho mỗi chủ thể

6. Biểu đồ tuần tự (Sequence Diagram)

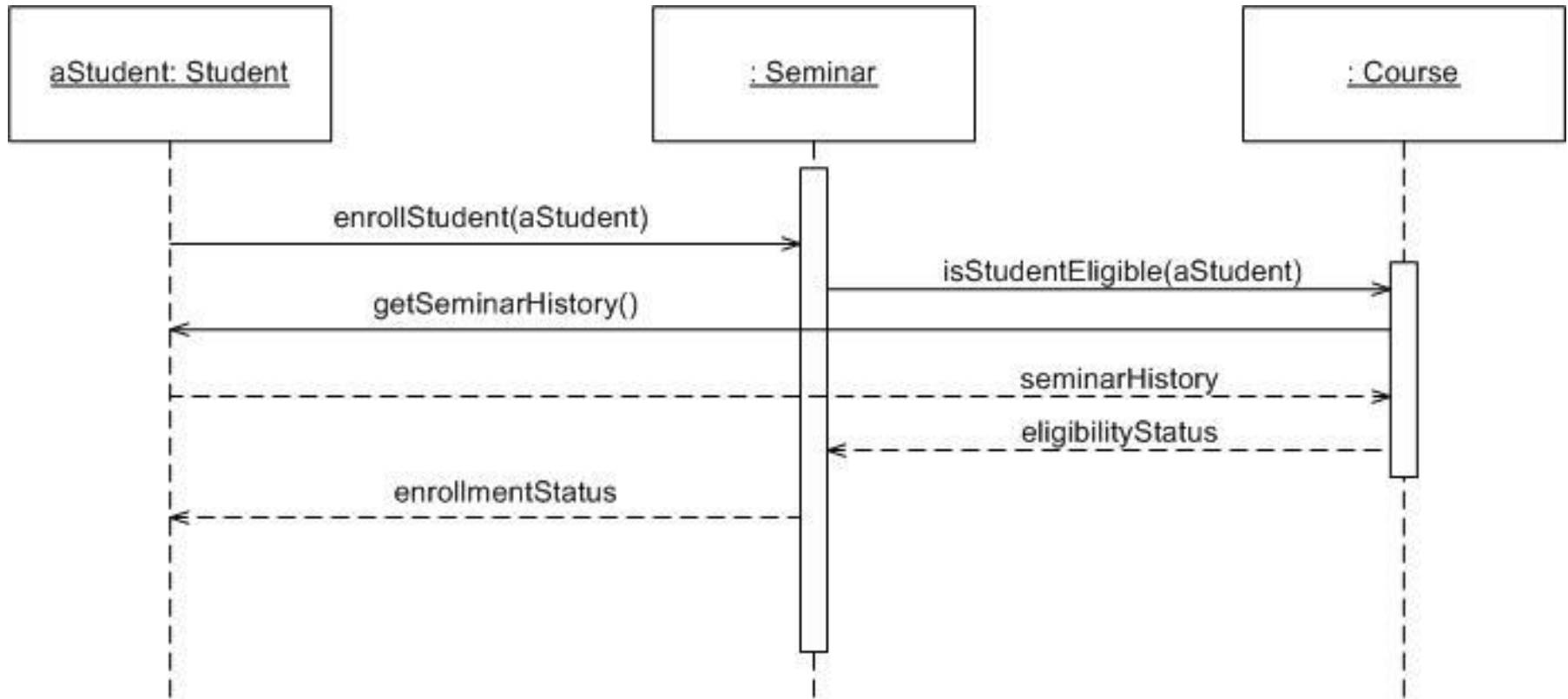
- Thể hiện một tập hợp các đối tượng và thông điệp truyền giữa chúng
- Chú trọng đến trình tự thời gian của thông điệp
- Thể hiện góc nhìn động của hệ thống.
- Là loại biểu đồ hành vi mô tả chuỗi các thông điệp gửi qua lại giữa các đối tượng theo trình tự thời gian

6. Biểu đồ tuần tự (Sequence Diagram)

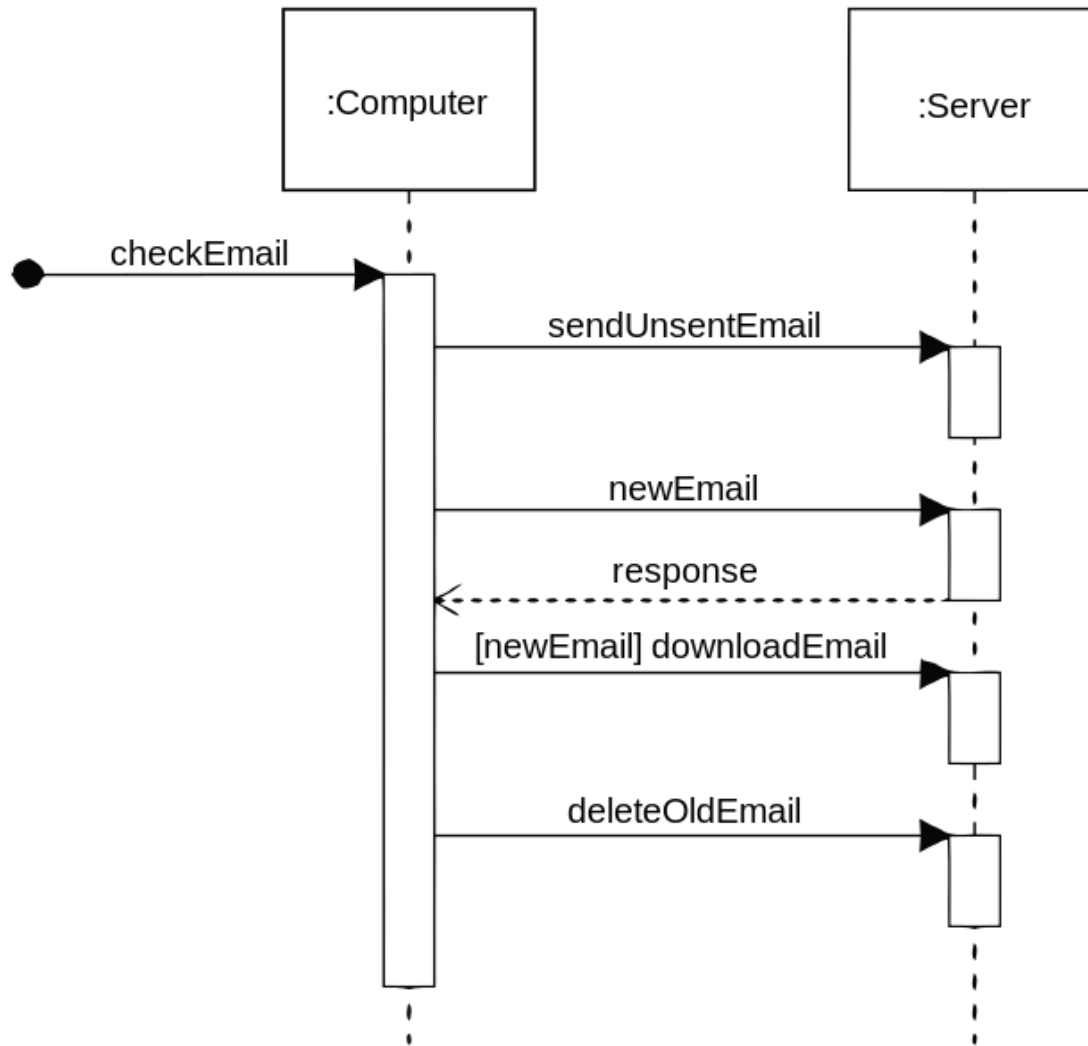
- **Các thành phần của biểu đồ tuần tự:**

- + **Đối tượng:** có thể là người, lớp hoặc một thành phần của hệ thống, được vẽ là một ô chữ nhật có trục thời gian là một đường nét đứt
- + **Tiến trình:** thời gian xử lý một yêu cầu mà đối tượng nhận được, biểu diễn là 1 hình chữ nhật hẹp nằm trên trục thời gian của đối tượng
- + **Frame:** khung bao quanh một mảng các tiến trình trên các đối tượng, nhằm phân biệt hoặc làm nổi bật một số tiến trình cụ thể.
- + **Thông điệp:** gồm 2 loại
 - **Lời gọi:** thường là một yêu cầu hoặc lời gọi hàm xuất phát từ 1 đối tượng kéo sang đối tượng kia, sinh ra một tiến trình ở đối tượng đích. Biểu diễn là một mũi tên nét liền, phía trên ghi tên lời gọi
 - **Phản hồi:** thường là dữ liệu trả về sau khi xử lý xong yêu cầu, được biểu diễn là một mũi tên nét đứt đi từ đối tượng xử lý đến đối tượng gọi. Phản hồi xảy ra khi tiến trình xử lý kết thúc.

6. Biểu đồ tuần tự (Sequence Diagram)



6. Biểu đồ tuần tự (Sequence Diagram)



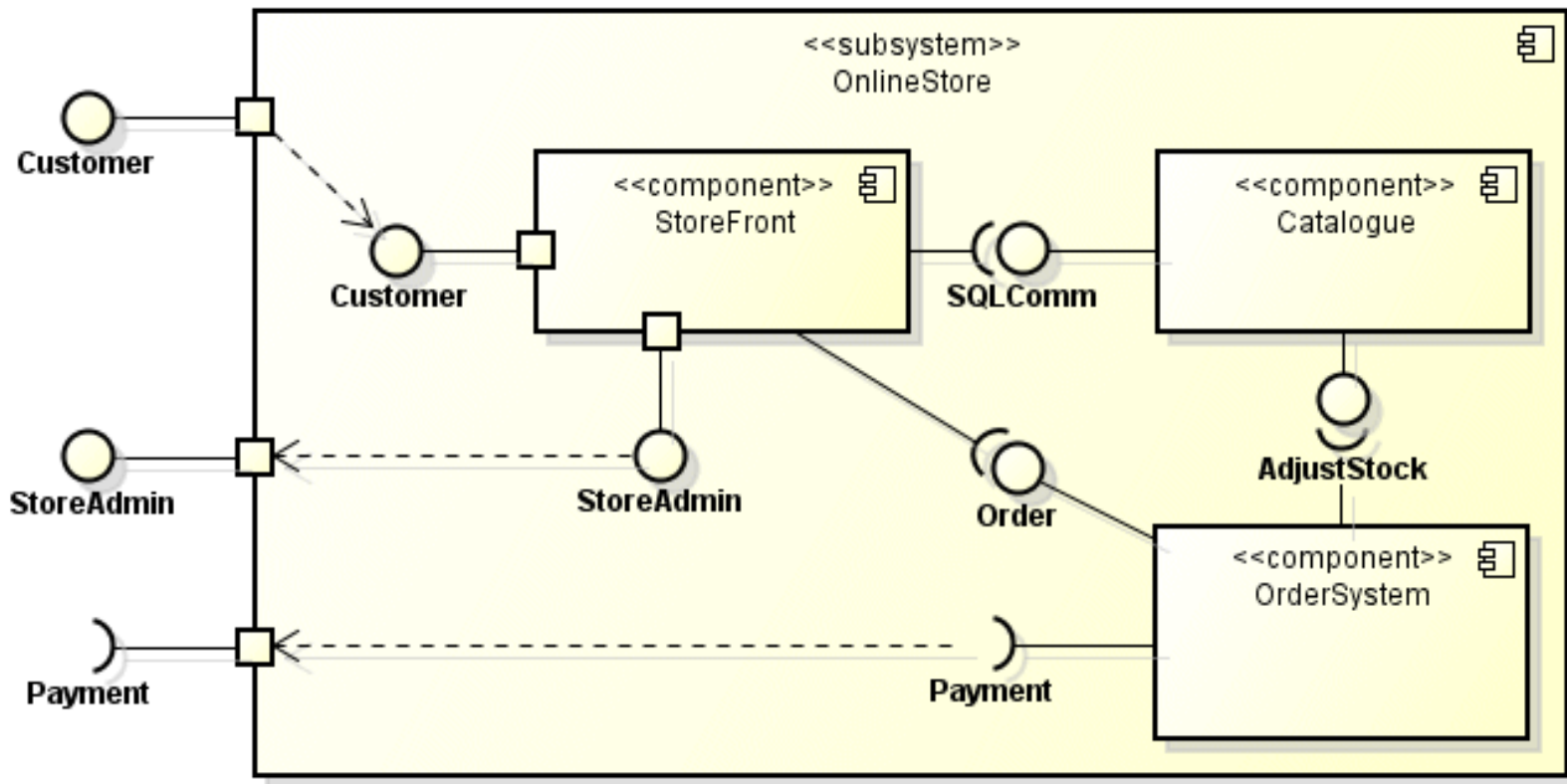
6. Biểu đồ tuần tự (Sequence Diagram)

- Biểu đồ tuần tự thường được vẽ dựa trên Use Case (các hành động, chức năng) và biểu đồ Lớp (các đối tượng)
- Cách vẽ:
 - Xác định chức năng được thể hiện bằng biểu đồ tuần tự
 - Xác định các đối tượng tham gia.
 - Dựa trên đặc tả Use case và vai trò của các lớp, xác định các tiến trình tuần tự trên các đối tượng, từ đó xác định các lời gọi và phản hồi.

7. Biểu đồ thành phần

- Là loại biểu đồ biểu diễn các thành phần của hệ thống và mối quan hệ giữa chúng
- Các thành phần của hệ thống:
 - + Các thành phần logic: các thành phần nghiệp vụ, các tiến trình, các lớp,...
 - + Các thành phần vật lý: file mã nguồn, thư viện, hệ thống con,...
- **Vai trò của biểu đồ thành phần:**
 - + Đóng vai trò quan trọng trong việc mô tả hệ thống dưới khía cạnh triển khai và cài đặt.
 - Cho phép người phát triển có cái nhìn tổng quan về hệ thống khi xây dựng.
 - Mô tả các quan hệ giữa các thành phần, từ đó cho phép xác định trình tự hoạt động và biên dịch của các thành phần.
 - + Được sử dụng không chỉ trong giai đoạn phát triển, mà còn trong giai đoạn bảo trì và mở rộng.

7. Biểu đồ thành phần

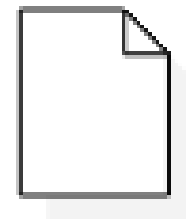
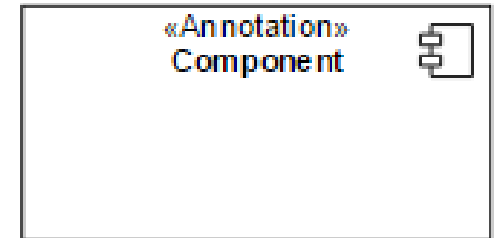


7. Biểu đồ thành phần

- **Ký hiệu:**

- + **Thành phần:**

- UML 1.x: biểu diễn bởi 1 hình chữ nhật, có hai hình chữ nhật nhỏ ở cạnh trái
 - UML 2.x: biểu diễn bởi 1 hình chữ nhật, có biểu tượng nhỏ ở góc phải
 - Ngoài ra, trong trường hợp biểu diễn các thành phần là các file mã nguồn, người ta còn dùng biểu tượng File



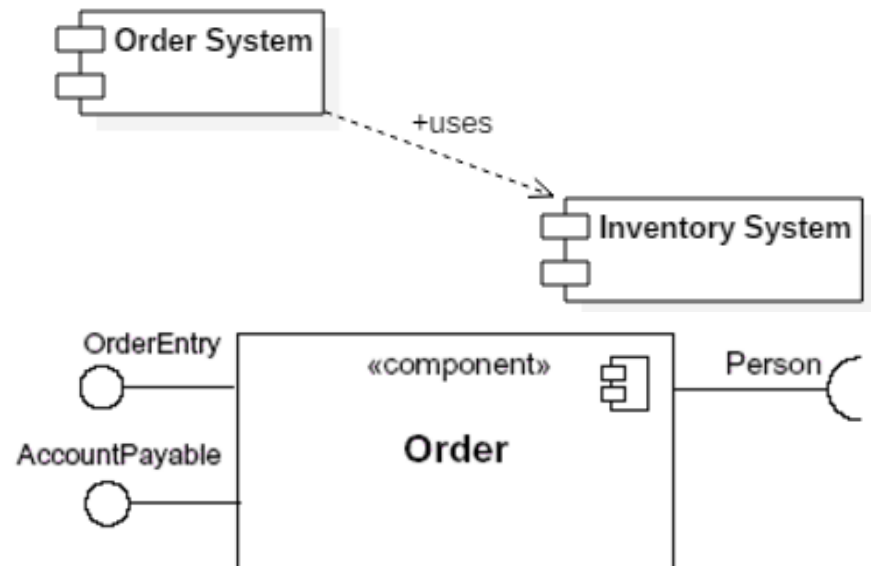
File 1

7. Biểu đồ thành phần

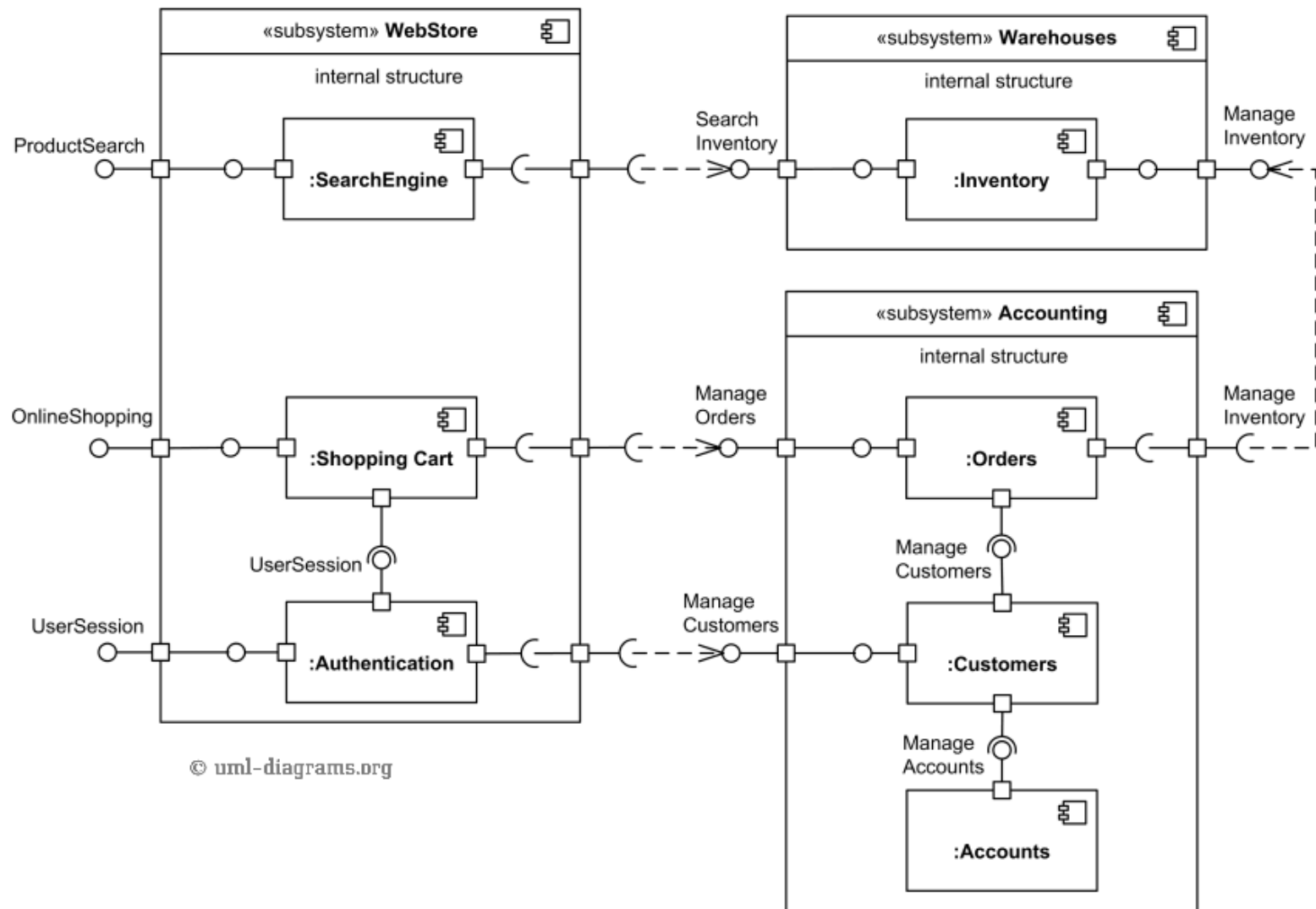
- **Ký hiệu:**

- + **Liên kết:**

- Dạng đơn giản: chỉ là một mũi tên nét đứt, nối từ thành phần sử dụng sang thành phần được sử dụng
 - Dạng giao diện: mỗi thành phần liên kết ra bên ngoài qua hai loại giao diện chính:
 - ✓ Giao diện cung cấp (Provided Interface)
 - ✓ Giao diện yêu cầu (Required Interface)
 - ✓ Một kết nối là một mũi tên nét đứt nối giao diện cung cấp của thành phần này với giao diện yêu cầu của thành phần kia



7. Biểu đồ thành phần



7. Biểu đồ thành phần

- **Các vẽ biểu đồ thành phần:**

- Xác định mục đích, đối tượng của biểu đồ thành phần cần vẽ:
 - Biểu diễn hệ thống dạng logic: quan hệ giữa các tiến trình, các module chứa các lớp, ...
 - Biểu diễn hệ thống dạng vật lý: các file mã nguồn, các gói cài đặt, các hệ thống con,....
- Xác định các thành phần cần vẽ:
 - Xác định phạm vi hệ thống cần vẽ, các thành phần liên quan trong phạm vi, các thành phần cha, thành phần con
 - Thể hiện các thành phần với tên gọi tương ứng
 - Xác định các giao diện cung cấp và yêu cầu của mỗi thành phần
 - Thành phần cần thông tin gì, dịch vụ gì, đối tượng dữ liệu gì -> giao diện yêu cầu
 - Thành phần có thể cung cấp dịch vụ gì, thông tin gì -> giao diện cung cấp
 - Nối các thành phần lại với nhau qua các giao diện tương ứng

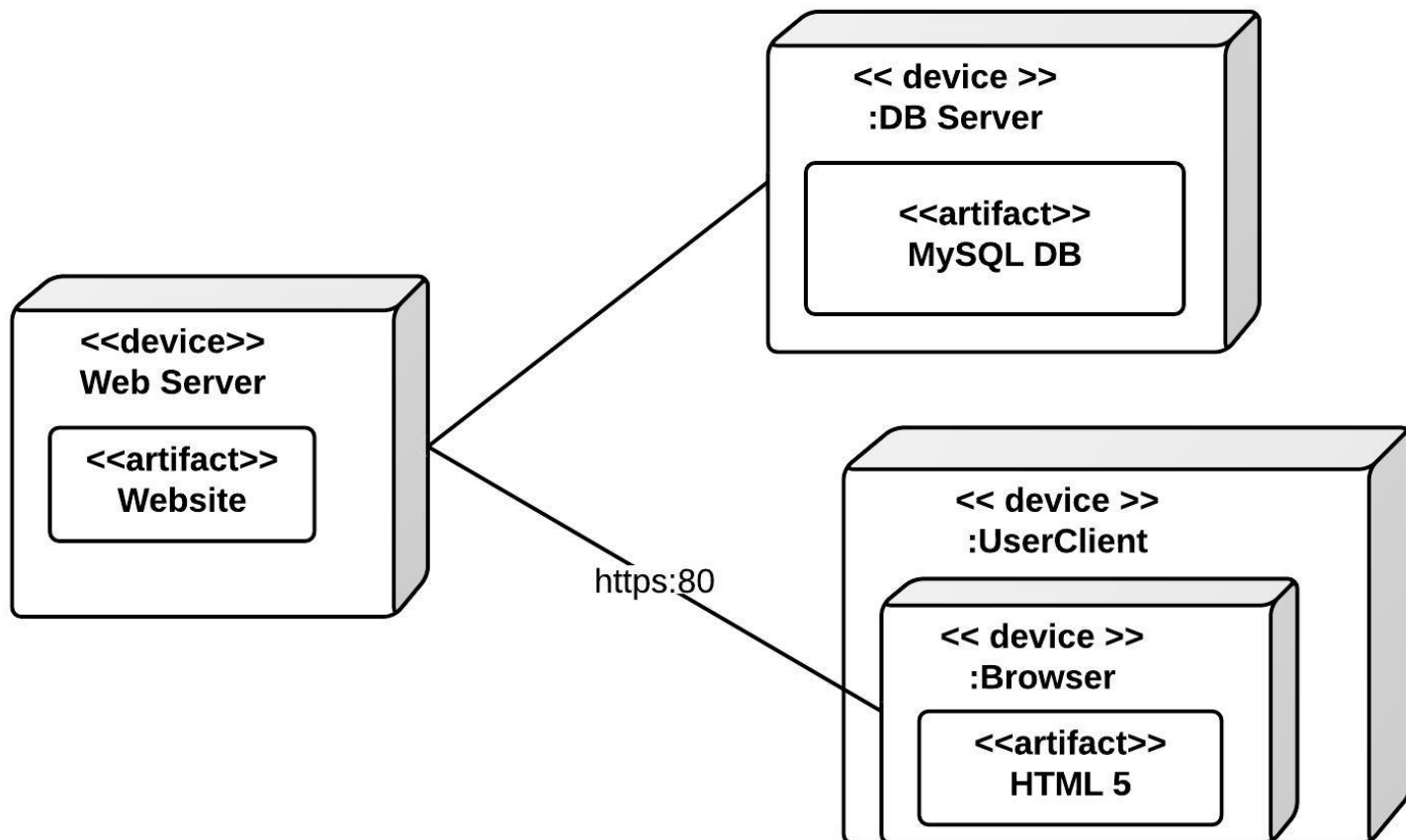
7. Biểu đồ thành phần

- Thể hiện tập hợp các thành phần và quan hệ giữa chúng
- Minh họa góc nhìn tĩnh của hệ thống
- Một thành phần được ánh xạ ra một hoặc nhiều lớp, giao diện

8. Biểu đồ triển khai (Deployment Diagram)

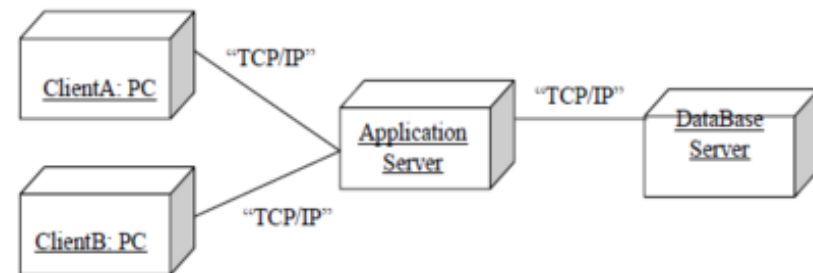
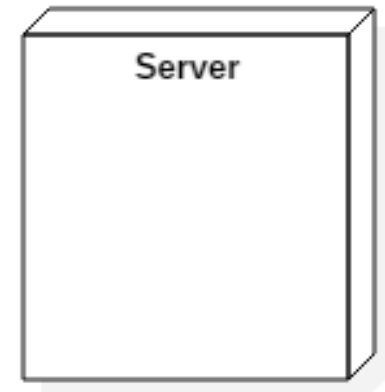
- Là loại biểu đồ biểu diễn sơ đồ của nút trong hệ thống khi thực thi. Nói cách khác, biểu đồ triển khai thể hiện cấu trúc phần cứng của hệ thống, các phần mềm cài đặt trên phần cứng.
- Được tạo ra khi cần biểu diễn cấu trúc của các hệ thống có nhiều thành phần phần cứng tham gia, ví dụ như các hệ thống dạng client-server
- Mỗi nút trong sơ đồ là một thiết bị phần cứng, trong mỗi nút có thể có các nút con hoặc các thành phần phần mềm

8. Biểu đồ triển khai (Deployment Diagram)



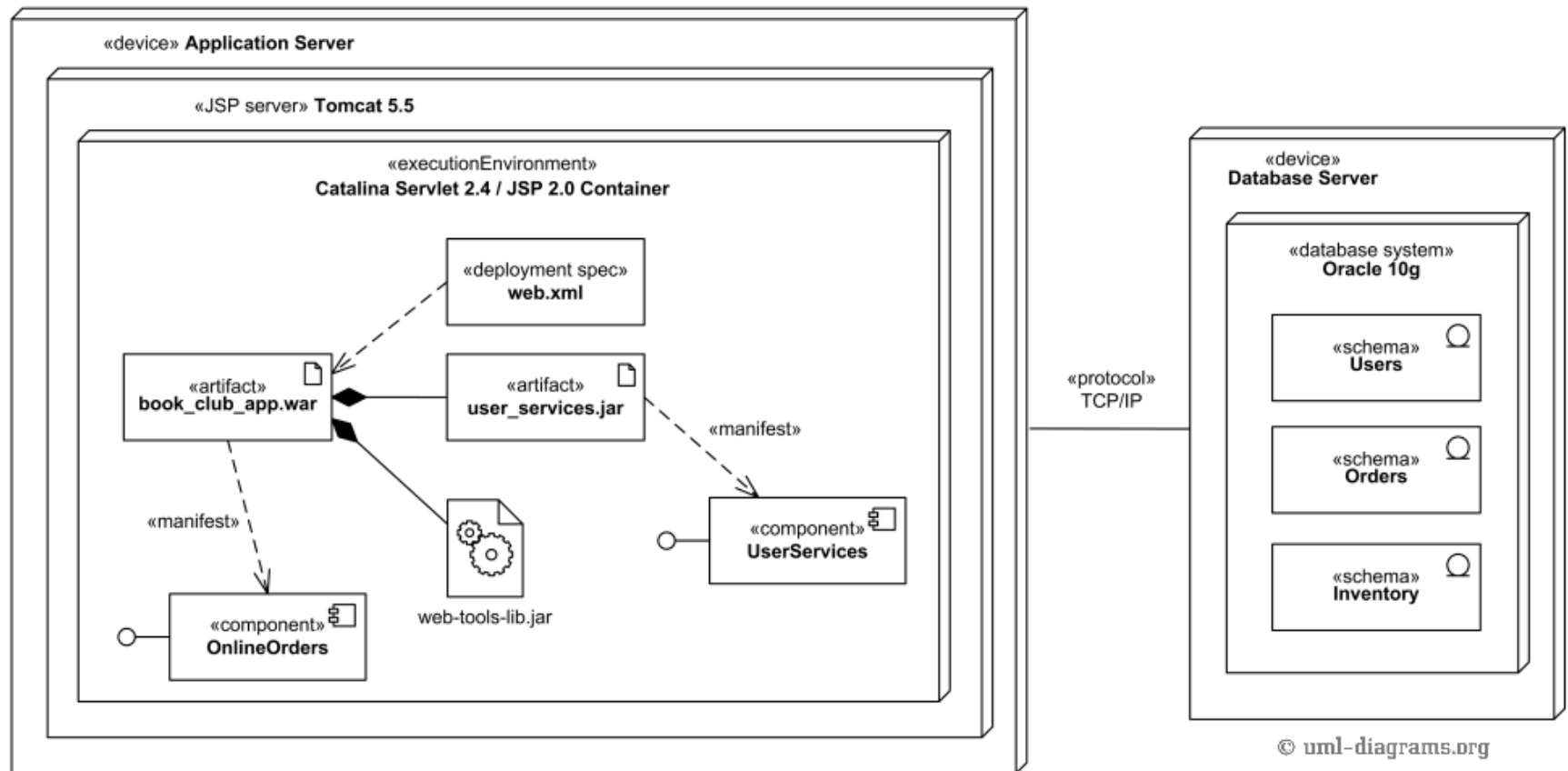
8. Biểu đồ triển khai (Deployment Diagram)

- Ký hiệu:
 - + Nút (node) : được ký hiệu là một khối hộp chữ nhật. Thực tế trong nhiều sơ đồ triển khai, người ta có thể dùng các loại ký hiệu khác nhau để biểu diễn các node như ký hiệu server, thiết bị mạng, thiết bị di động, máy tính cá nhân....
 - + Liên kết: liên kết giữa các nút thường thể hiện **đặc điểm** kết nối giữa các node, được biểu diễn bằng đường nét liền có ghi chú về đặc điểm của kết nối.



8. Biểu đồ triển khai (Deployment Diagram)

- Ví dụ:



8. Biểu đồ triển khai (Deployment Diagram)

- **Cách vẽ biểu đồ triển khai:**
 - + Xác định phạm vi hệ thống cần biểu diễn
 - Biểu đồ chỉ thể hiện sơ đồ của 1 hệ thống phần mềm hay toàn bộ hệ thống quản lý?
 - + Xác định kiến trúc của hệ thống:
 - Trọng tâm việc xử lý của hệ thống nằm ở client hay server?
 - Hệ thống bao gồm 2,3 hay nhiều tầng?
 - + Xác định các nút và kết nối giữa chúng:
 - Tên gọi, chủng loại, hệ điều hành trên các nút
 - Đặc điểm kết nối giữa các nút, vd : tcp/ip (kiểu kết nối) , RMI (chuẩn công nghệ kết nối),...
 - + Phân bố các thành phần phần mềm lên các nút

Bài tập 1

Cho:

- Các tác nhân: Người mua, hệ thống Email, Hệ thống cho vay và Hệ thống báo cáo tín dụng
- Các Use Case: Tìm người môi giới, Quản lý hồ sơ cá nhân, Tìm kiếm nhà và Yêu cầu vay
- Các mối liên kết:
 - Từ người mua tới Tìm người môi giới
 - Từ người mua tới Quản lý hồ sơ cá nhân
 - Từ người mua tới Tìm kiếm nhà
 - Từ người mua tới Yêu cầu vay
 - Quản lý hồ sơ cá nhân tới Hệ thống Email
 - Tìm kiếm nhà tới Hệ thống Email
 - Yêu cầu vay tới Hệ thống Email, Hệ thống cho vay
 - Yêu cầu vay tới Hệ thống báo cáo tín dụng

Yêu cầu: Hãy vẽ biểu đồ Use Case.

Bài tập 2

Cho:

- Các trạng thái hành động:
 - Chọn hồ sơ
 - Tìm hồ sơ người mua
 - Tạo hồ sơ mới
 - Đăng nhập
- Luồng hoạt động:
 - Bắt đầu từ Chọn hồ sơ tới Tìm hồ sơ người mua rồi đi từ Tìm hồ sơ người mua đến Tạo hồ sơ mới nếu hồ sơ không tồn tại. Nếu hồ sơ tồn tại thì có thể Đăng nhập.

Yêu cầu: Hãy vẽ biểu đồ Hoạt động.