



FPT POLYTECHNIC

NHẬP MÔN KỸ THUẬT PHẦN MỀM

Bài 5: Unified Modeling Language (UML)

www.poly.edu.vn



Điểm danh

Nội dung bài học

1. Khái niệm về UML
2. Phân tích về thiết kế UML



1. Khái niệm về UML

Unified Modeling Language (UML)

- UML là ngôn ngữ đồ họa tiêu chuẩn
- Sử dụng cho rất nhiều lĩnh vực kỹ thuật trong đó có công nghệ phần mềm.
- UML là ngôn ngữ ký hiệu sử dụng trong lập trình hướng đối tượng.
- UML biểu diễn, quy định cụ thể, xây dựng và tài liệu về hệ thống phần mềm

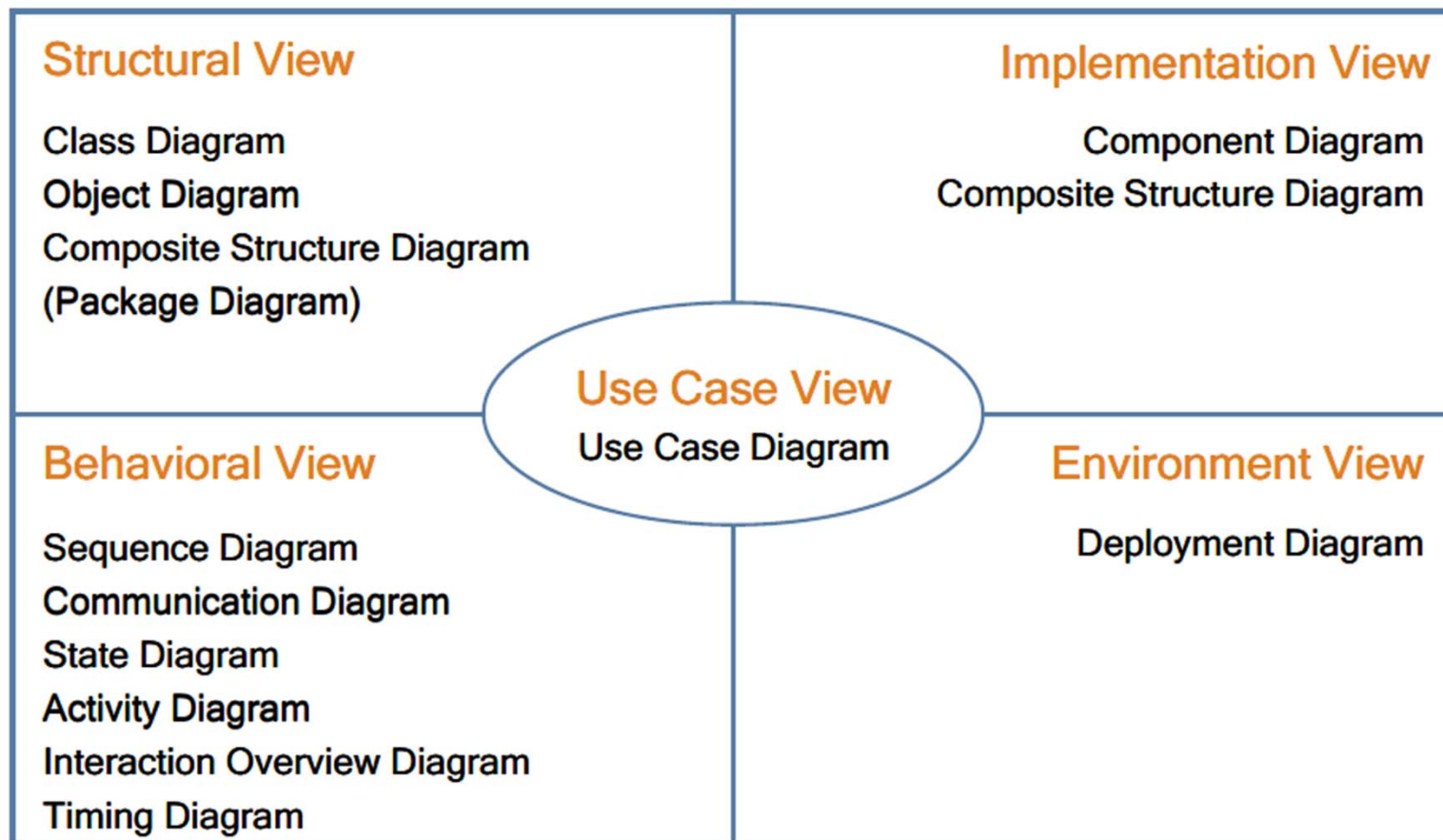
Unified Modeling Language (UML)

- Vì sao nên sử dụng UML? => Vì UML là:
 - ✓ Ký hiệu đồ họa
 - ✓ Một bức tranh đáng giá ngàn lời nói
 - ✓ Ngôn ngữ giao tiếp chuẩn
 - ✓ Cung cấp nhiều sơ đồ để chụp kiến trúc khác nhau
 - ✓ Thúc đẩy việc tái sử dụng các thành phần
 - ✓ UML là một ngôn ngữ tiêu chuẩn cho phép biểu diễn, quy định cụ thể, xây dựng và tài liệu về hệ thống phần mềm

Unified Modeling Language (UML)

UML định nghĩa 13 sơ đồ mô tả 4+1 quan điểm kiến trúc

4+1 architectural views model was proposed by Philippe Kruchten, IBM



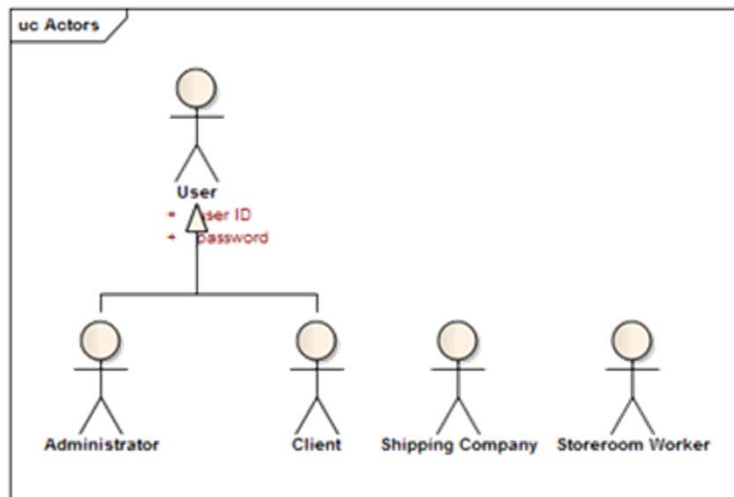
2. Phân tích về thiết kế UML

UML – Use case diagram

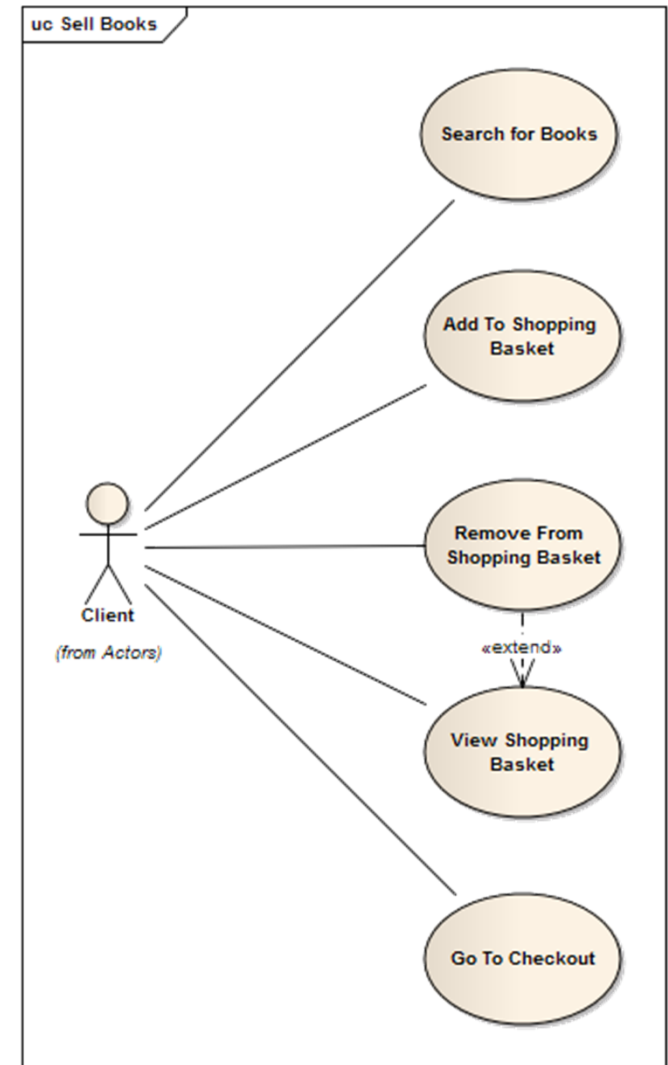
Mô tả các chức năng được cung cấp bởi hệ thống

Mỗi trường hợp sử dụng cho một chức năng riêng biệt của sự tương tác giữa người sử dụng (người hoặc máy) và hệ thống.

Bao gồm: **actors**, **use cases**, và **relationships**



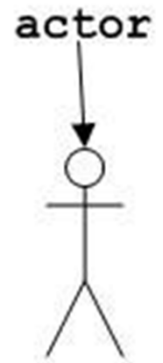
BookShop Sample: Actors + Sell Book Use Cases



UML – Use case diagram

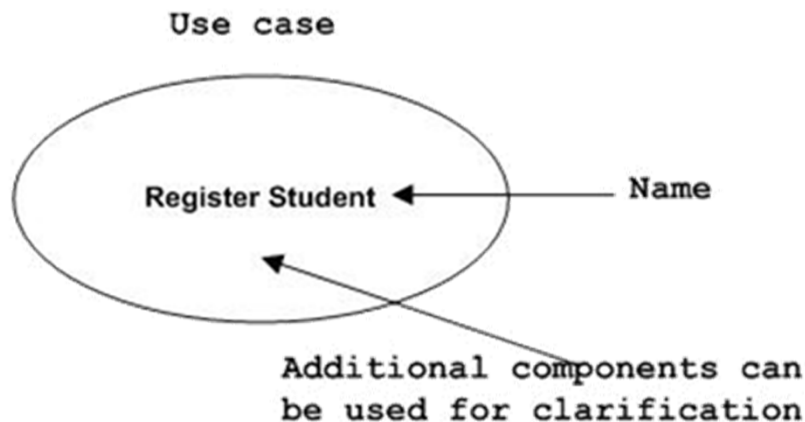
Actor:

Có thể là người dùng, ứng dụng nội bộ hoặc bên ngoài



Use case

Chức năng chính của hệ thống



UML – Use case diagram

Actor Relationship: Generalization

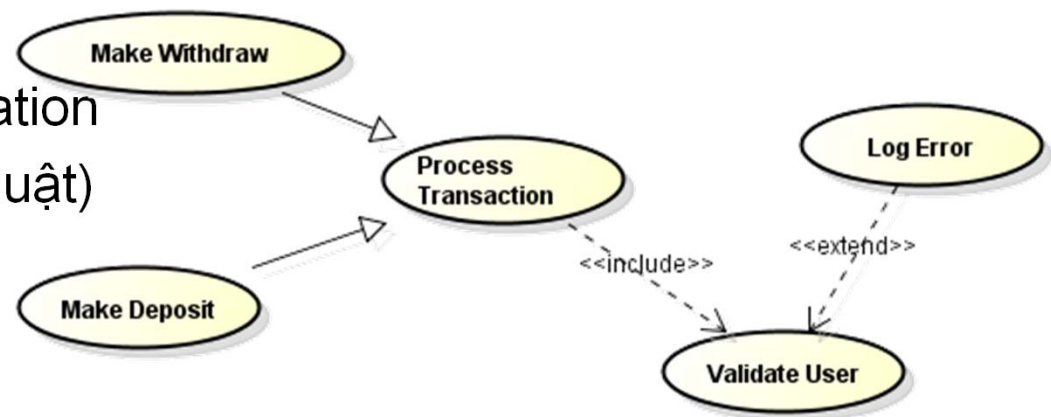
- Định nghĩa vai trò người dùng

Actor – Use case relationship

- Association (Quan hệ người dùng với chức năng)

Use case Relationship

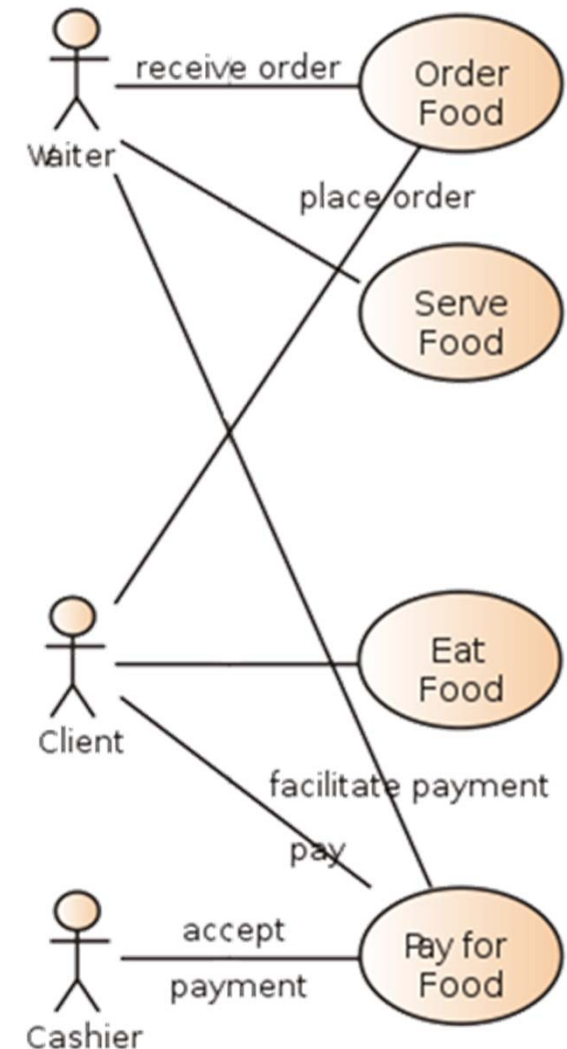
- Include (Bao gồm)
- Extend (Mở rộng)
- Generalization/Specification
(Khái quát / Thông số kỹ thuật)



UML – Use case diagram

Association

- Quan hệ người dùng với chức năng
- Actor tham gia vào tương tác mô tả bởi use case
- Sử dụng đường thẳng nhằm:
 - ✓Chỉ ra tương tác của actor tới chức năng
 - ✓Chỉ luồng điều khiển (có data hoặc không)

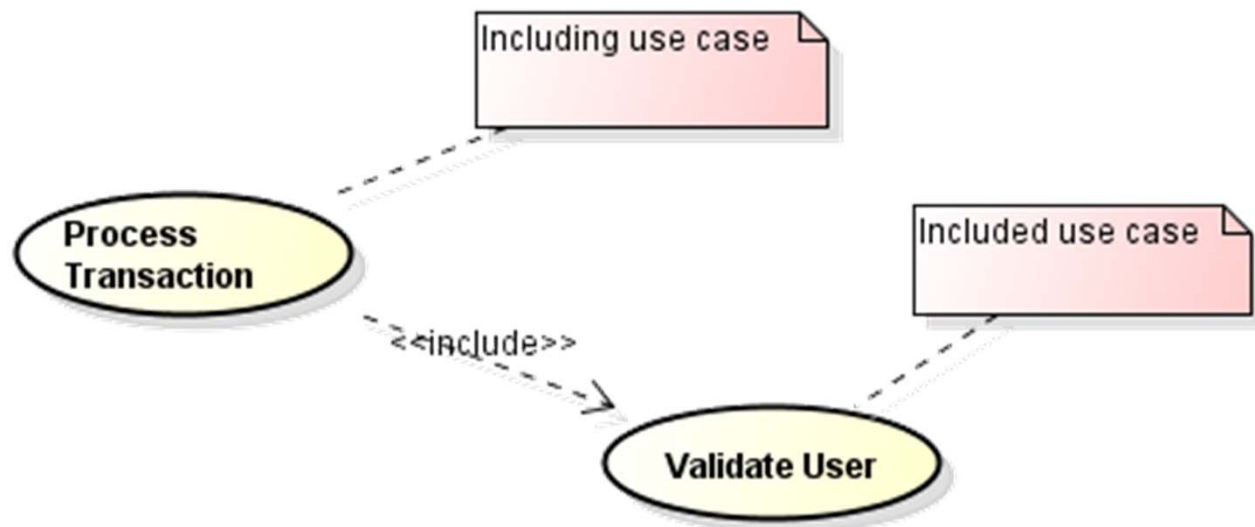


UML – Use case diagram

Include:

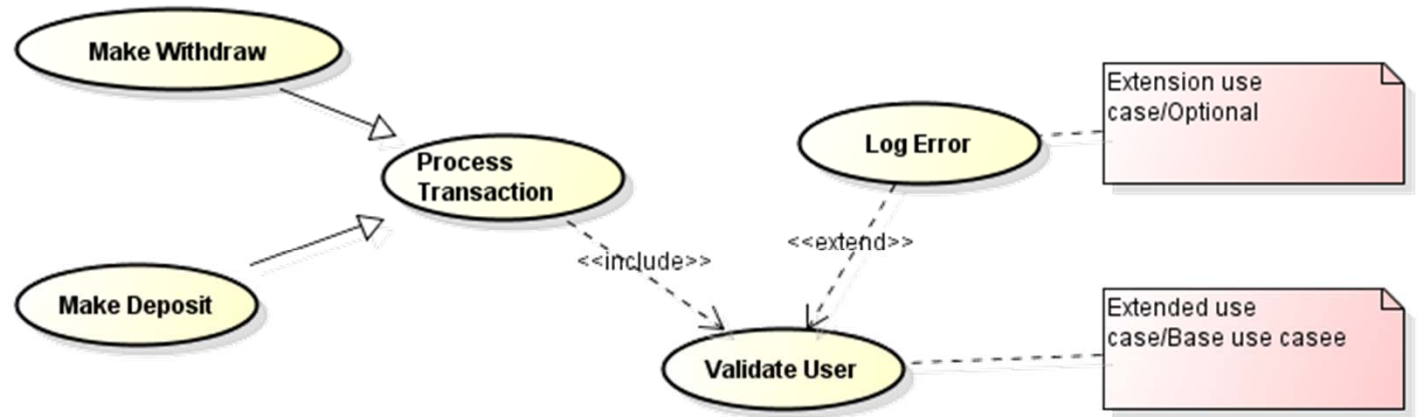
Mối quan hệ trực tiếp giữa 2 chức năng.

Hành vi của chức năng được chèn vào hành vi của chức năng bao gồm nó.



UML – Use case diagram

Extends



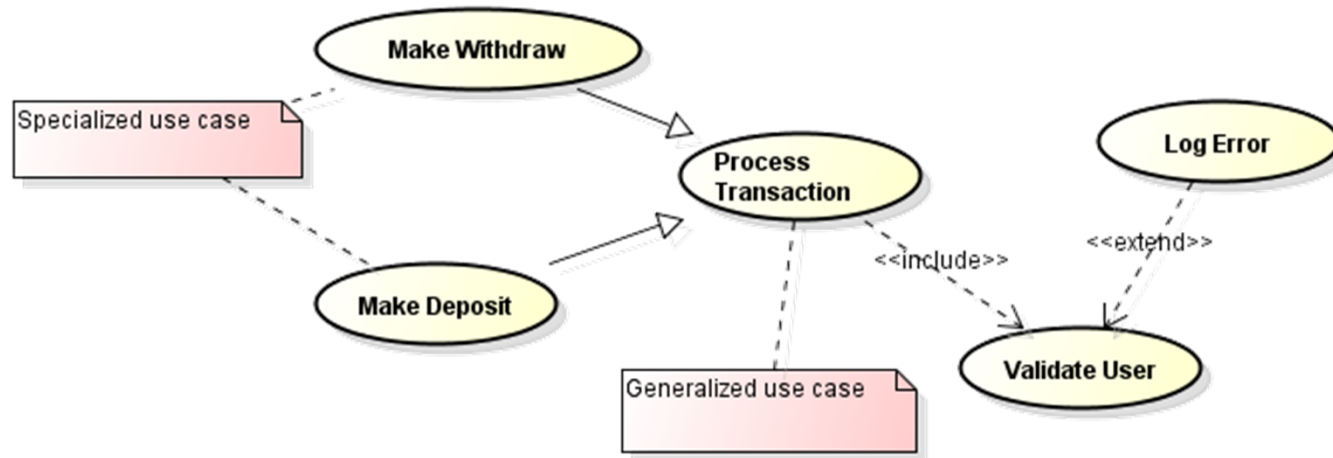
Hành vi của chức năng có thể được chèn vào chức năng mở rộng với một điều kiện nào đó

Extension use case :

- Không bắt buộc ,
- Có khả năng không được thực hiện với các chức năng cơ bản,
- Không cần thiết để đạt được mục tiêu.

UML – Use case diagram

Generalization/Specialization



- General use case: (Chức năng khái quát) có chung hành vi, yêu cầu, ràng buộc, giả định.
- General use case được mô tả chung 1 lần
- Sự khác nhau thì được mô tả trong Specialized use case.

UML – Use case diagram

Vẽ use case diagram, phải xác định:

- Các chức năng của hệ thống chính là use case
- Actor (người dùng)
- Quan hệ giữa use cases và actors

Use case diagram tốt phải:

- Tên của use case rất quan trọng.
 - ✓ Đặt tên có thể xác định các chức năng được thực hiện
 - ✓ Đặt tên phù hợp cho Actor.
- Thể hiện mối quan hệ và sự phụ thuộc rõ ràng.
- Không bao gồm tất cả các loại quan hệ
- Mục đích chính của Use case diagram để nắm bắt yêu cầu.
- Khi cần thiết làm rõ một số điểm quan trọng cần có ghi chú

UML – Use case diagram

THẢO LUẬN MỞ VỀ USE CASE



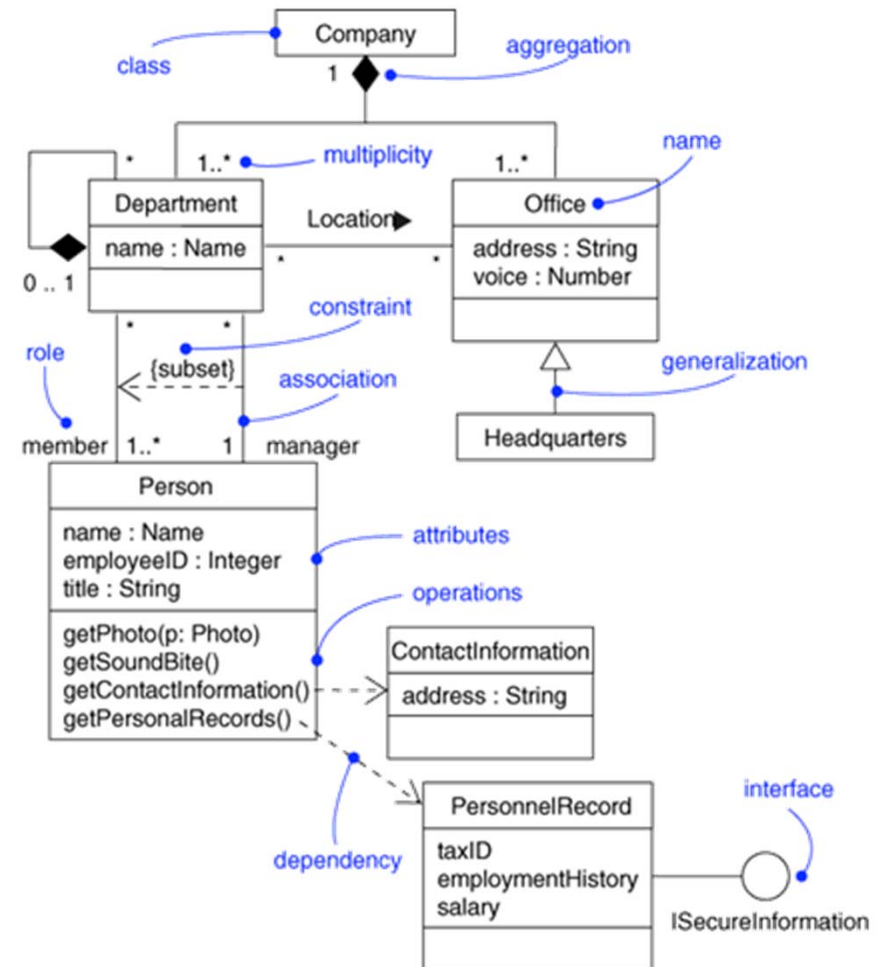
UML – Class diagram

Mô tả cấu trúc hệ thống

- Mô tả các Class của hệ thống, thuộc tính của chúng và mối quan hệ giữa các Class.

Mục đích:

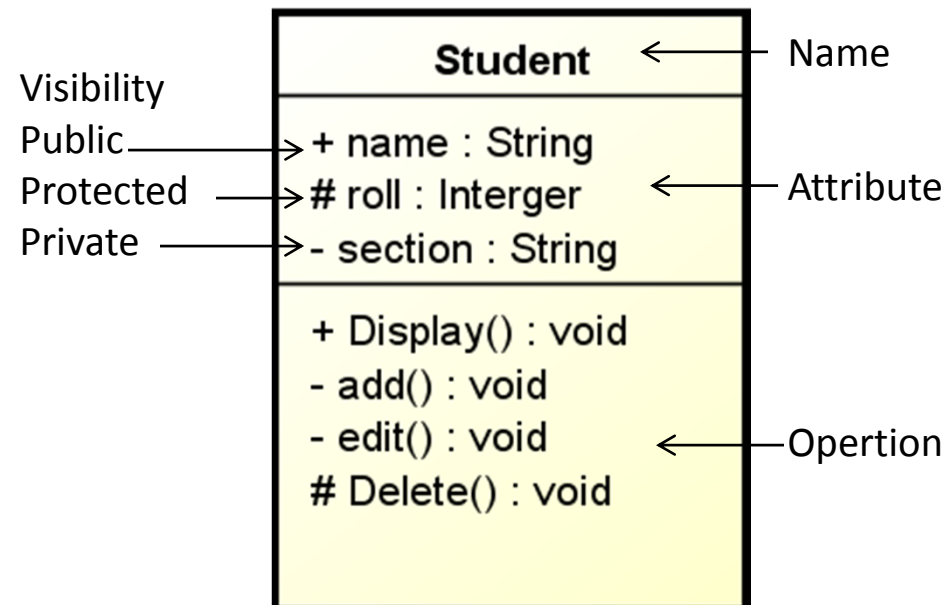
- Phân tích và thiết kế các View tĩnh của ứng dụng.
- Thể hiện sự thống nhất giữa các thành phần trong View tĩnh.
- Mô tả sự phản hồi của hệ thống.
- Cơ sở cho các thành phần và triển khai sơ đồ.



UML – Class diagram

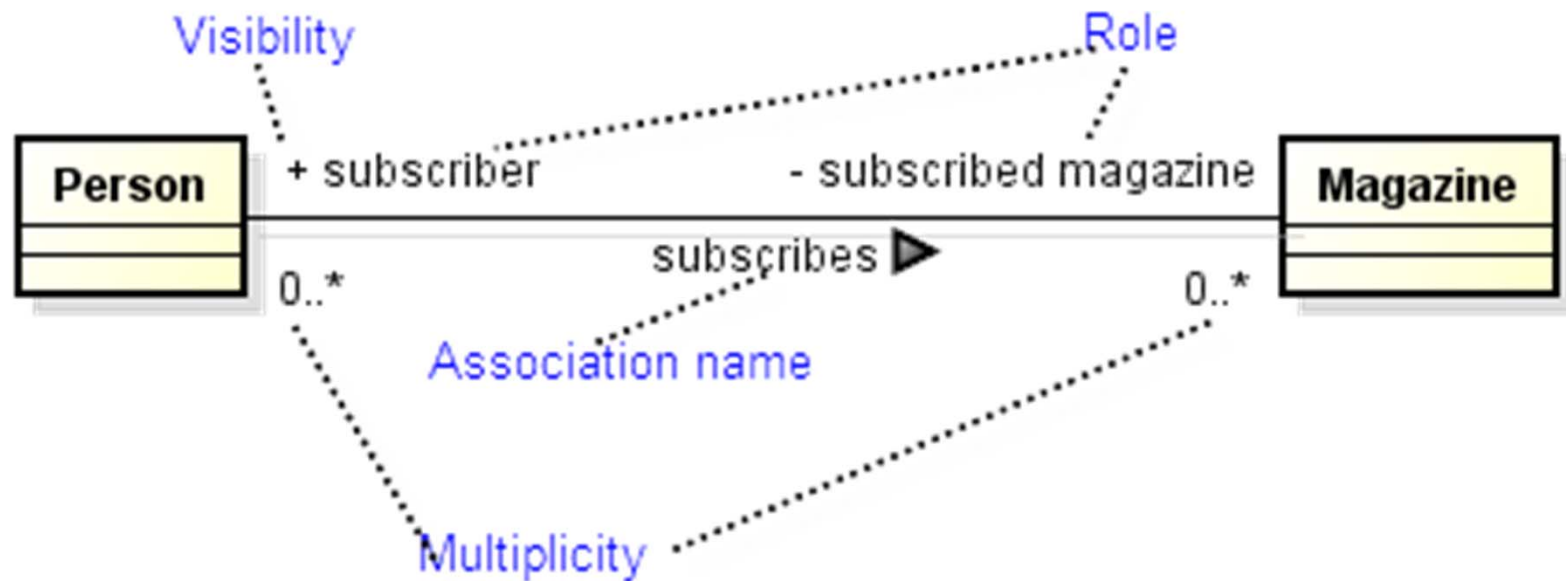
Class

- Đại diện cho đối tượng (Object)
- Có các thuộc tính
- Và các function.



UML – Class diagram

Thể hiện mối quan hệ giữa các Class



UML – Class diagram

Class diagram tốt phải:

- Tên class, tên diagram nên mô tả được các đối tượng tồn tại trong hệ thống.
- Thành phần, mối quan hệ nên được định nghĩa rõ ràng.
- Attributes và methods nên được định nghĩa rõ ràng.
- Trong mỗi Class: Tối thiểu các thuộc tính.
 - ✓ Những thuộc tính không cần thiết có thể làm sơ đồ phức tạp hơn.
- Dùng các ghi chú khi được yêu cầu.
- Diagram có thể được vẽ trên giấy thường và làm lại nhiều lần để làm cho nó chính xác hơn.

UML – Activity diagram

Activity diagrams là sơ đồ quy trình công việc từng bước của các hành động với sự hỗ trợ cho sự lựa chọn, lặp đi lặp lại và đồng thời.

Activity diagrams thể hiện dòng chảy hành động được kiểm soát như thế nào.

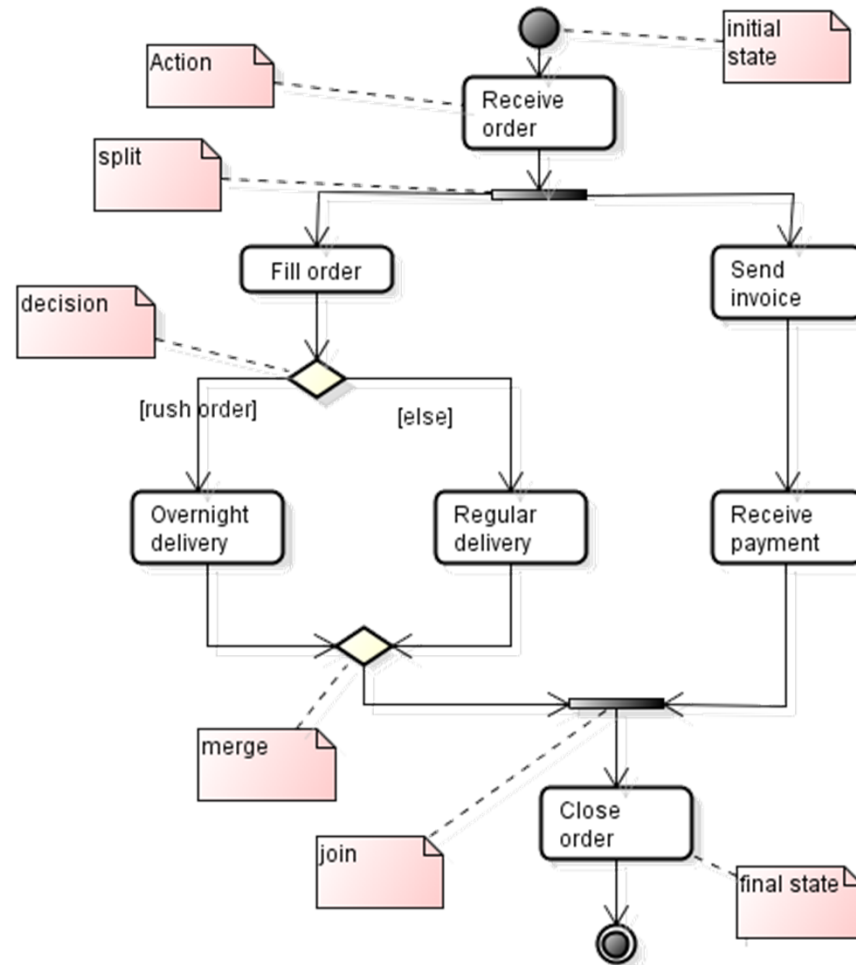
UML – Activity diagram

Activity diagrams được tạo thành từ các shape, kết nối bằng arrow (mũi tên):

- Rounded rectangles: actions (hành động)
- Diamonds: lựa chọn
- Bars: bắt đầu (split) hoặc kết thúc (join) của hành động.
- A black circle: Bắt đầu (initial state) của workflow
- An encircled black circle: kết thúc (final state) của workflow.

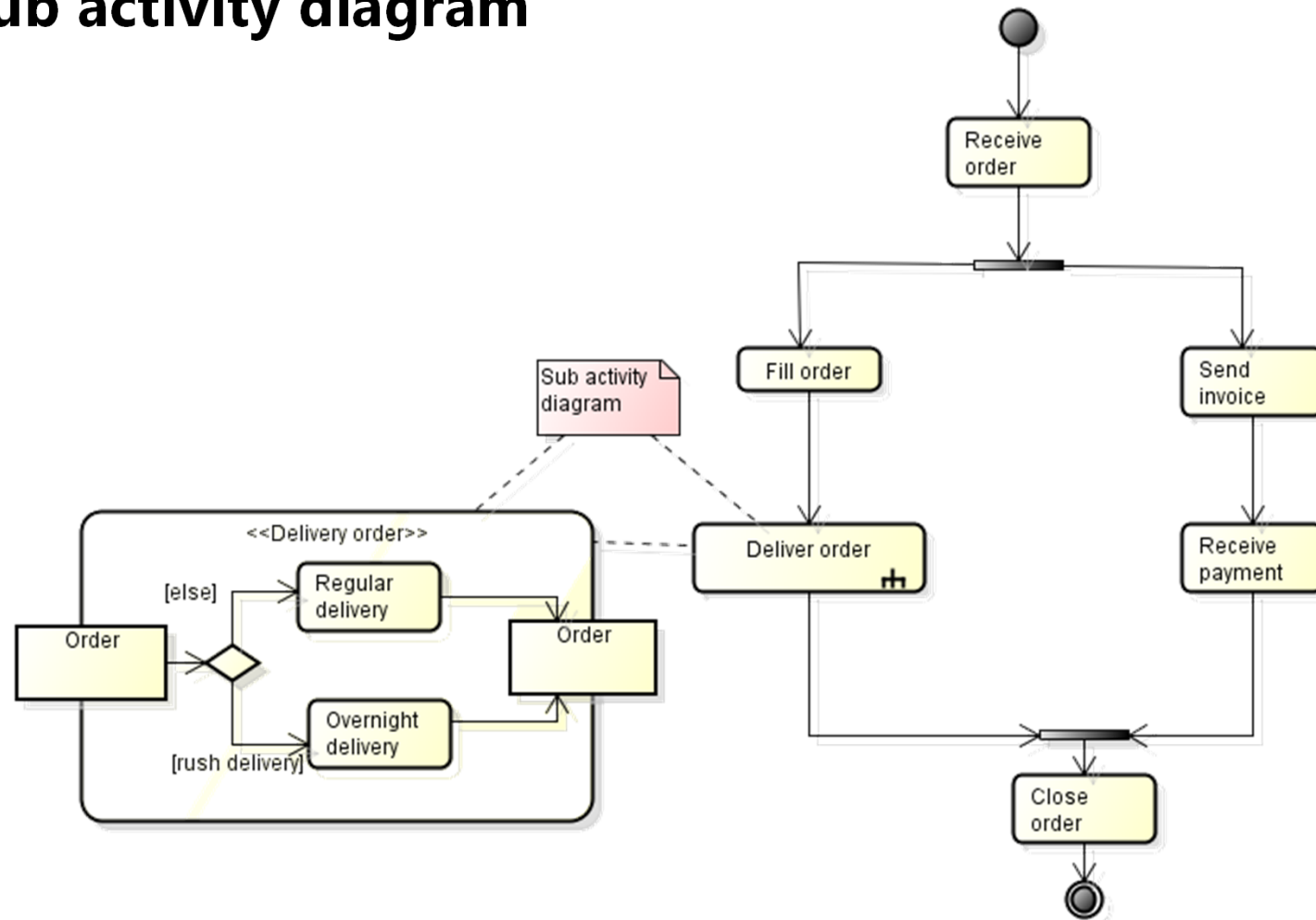
UML – Activity diagram

Example of an activity diagram



UML – Activity diagram

Sub activity diagram



DEMO

Microsoft Visio

File Home Insert Design Data Process Review View

Save
Save As
Open
Close
Info
Recent
New
Print
Save & Send
Help
Options
Exit

Choose a Template

Home

Recently Used Templates

Basic Flowchart (US units)
Cross-Functional Flowchart (US units)
Organization Chart (US units)
Detailed Network Diagram (US units)

Template Categories

Business Engineering Flowchart General Maps and Floor Plans Network Schedule Software and Database

Other Ways to Get Started

Blank drawing Office.com templates Sample diagrams New from existing

Basic Flowchart (US units)

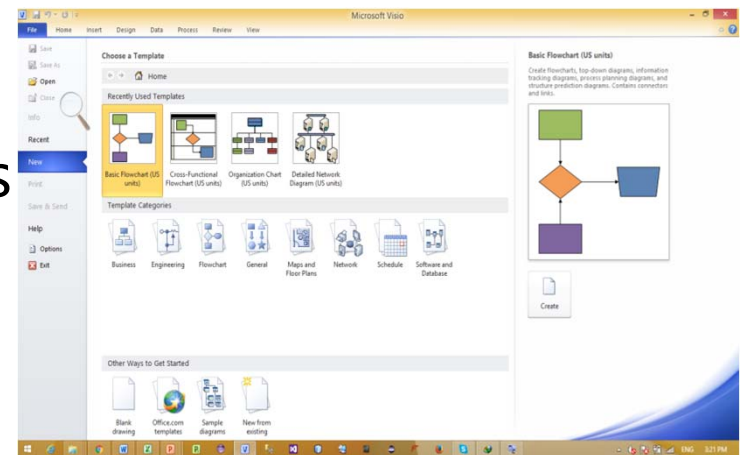
Create flowcharts, top-down diagrams, information tracking diagrams, process planning diagrams, and structure prediction diagrams. Contains connectors and links.

Create

Sử dụng Visio vẽ sơ đồ

Chuẩn bị Workshop 4

- Sử dụng công cụ Visio thiết kế Class diagram và Active diagram theo Requirement của đề tài.



Tổng kết nội dung bài học

1. Khái niệm về UML
2. Phân tích về thiết kế UML





KẾT THÚC