# 2. Một số dạng biểu đồ UML phổ biến

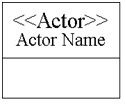
**2.1. Biểu đồ Use case (Use Case Diagram)**

Một biểu đồ Use case chỉ ra một số lượng các tác nhân ngoại cảnh và mối liên kết của chúng đối với Use case mà hệ thống cung cấp. Một Use case là một lời miêu tả của một chức năng mà hệ thống cung cấp. Lời miêu tả Use case thường là một văn bản tài liệu, nhưng kèm theo đó cũng có thể là một biểu đồ hoạt động. Các Use case được miêu tả duy nhất theo hướng nhìn từ ngoài vào của các tác nhân (hành vi của hệ thống theo như sự mong đợi của người sử dụng), không miêu tả chức năng được cung cấp sẽ hoạt động nội bộ bên trong hệ thống ra sao. Các Use case định nghĩa các yêu cầu về mặt chức năng đối với hệ thống.

* Hệ thống: Với vai trò là thành phần của biểu đồ use case, hệ thống biểu diễn ranh giới giữa bên trong và bên ngoài của một chủ thể trong phần mềm chúng ta xây dựng.Một hệ thống ở trong biểu đồ use case không nhất thiết là một hệ phần mềm; nó có thể là một chiếc máy,hoặc là một hệ thống thực như một doanh nghiệp, một trường đại học,…
* Tác nhân(actor):là người dùng của hệ thống, một tác nhân có thể là một người dùng thực hoặc các hệ thống máy tính khác có vai trò nào đó trong hoạt động của hệ thống. Như vậy, tác nhân thực hiện các use case. Một tác nhân có thể thực hiện nhiều use case và ngược lại một use case cũng có thể được thực hiện bởi nhiều tác nhân

Tác nhân được kí hiệu: 

hoặc



* Các use case: Đây là thành phần cơ bản của biểu đồ use case. Các use case được biểu diễn bởi các hình elip.Tên các use case thể hiện một chức năng xác định của hệ thống.

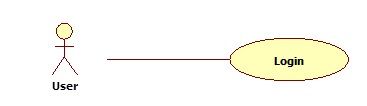
Các Use case được kí hiệu bằng hình elips.



* Mối quan hệ giữa các use case:
  + Association: thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau

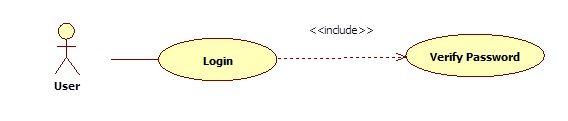
Association.jpg

Ví dụ quan hệ association:



* + 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.

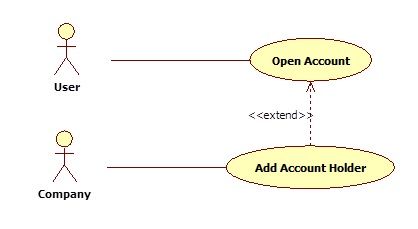


Ví dụ quan hệ include: 

* + Extent: Extend dùng để mô tả quan hệ giữa 2 Use Case. Quan hệ Extend 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.



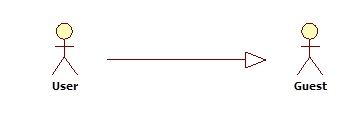
Ví dụ quan hệ extent:



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

Generalization.jpg

Ví dụ quan hệ Generalization:



**2.2.Biểu đồ lớp (Class Diagram)**

Một biểu đồ lớp chỉ ra cấu trúc tĩnh của các lớp trong hệ thống. Các lớp là đại diện cho các “đối tượng” được xử lý trong hệ thống. Các lớp có thể quan hệ với nhau trong nhiều dạng thức:

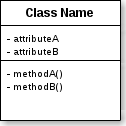
* liên kết (associated - được nối kết với nhau),
* phụ thuộc (dependent - một lớp này phụ thuộc vào lớp khác),
* chuyên biệt hóa (specialized - một lớp này là một kết quả chuyên biệt hóa của lớp khác),
* hay đóng gói ( packaged - hợp với nhau thành một đơn vị).

Tất cả các mối quan hệ đó đều được thể hiện trong biểu đồ lớp, đi kèm với cấu trúc bên trong của các lớp theo khái niệm thuộc tính (attribute) và thủ tục (operation). Biểu đồ được coi là biểu đồ tĩnh theo phương diện cấu trúc được miêu tả ở đây có hiệu lực tại bất kỳ thời điểm nào trong toàn bộ vòng đời hệ thống.

Một hệ thống thường sẽ có một loạt các biểu đồ lớp – không phải bao giờ tất cả các biểu đồ lớp này cũng được nhập vào một biểu đồ lớp tổng thể duy nhất – và một lớp có thể tham gia vào nhiều biểu đồ lớp.

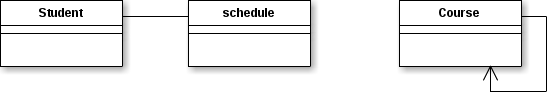
**2.2.1.Một lớp có các thành phần sau**

* Tên lớp
* Các thuộc tính
* Các phương thức

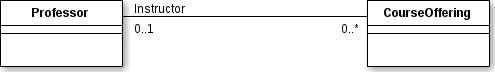


**2.2.2.Liên kết giữa các lớp**

* Liên kết (Association)
  + Mối liên hệ ngữ nghĩa giữa hai hay nhiều lớp chỉ ra sự liên kết giữa các thể hiện của chúng
  + Mối quan hệ về mặt cấu trúc chỉ ra các đối tượng của lớp này có kết nối với các đối tượng của lớp khác.



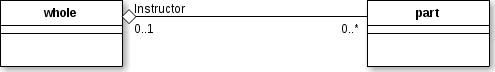
Bội số quan hệ: là số lượng thể hiện của một lớp liên quan tới một thể hiện của lớp khác. Với mỗi liên kết, có hai bội số quan hệ cho hai đầu của liên kết.

Ví dụ: 

Với mỗi đối tượng của Professor, có nhiều Course Offerings có thể được dạy. Với mỗi đối tượng của Course Offering, có thể có 1 hoặc 0 Professor giảng dạy.

| **Biểu diễn** | **Ý nghĩa** |
| --- | --- |
|  | Unspecified |
| 1 | chính xác 1 |
| 0..\* | 0 hoặc nhiều |
| \* | 0 hoặc nhiều |
| 1..\* | 1 hoặc nhiều |
| 0..1 | 0 hoặc 1 |
| 2..4 | Specified Range |
| 2, 4..6 | Multiple, Disjoint Ranges |

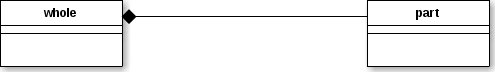
* Biểu diễn bội số quan hệ:
* Kết tập (Aggregation)
  + Là một dạng đặc biệt của liên kết mô hình hóa mối quan hệ toàn thể-bộ phận (whole-part) giữa đối tượng toàn thể và các bộ phận của nó.
  + Kết tập là mối quan hệ “là một phần” (“is a part-of”).
  + Bội số quan hệ được biểu diễn giống như các liên kết khác



* + Cấu thành (Composition) là :Một dạng của kết tập với quyền sở hữu mạnh và các vòng đời trùng khớp giữa hai lớp

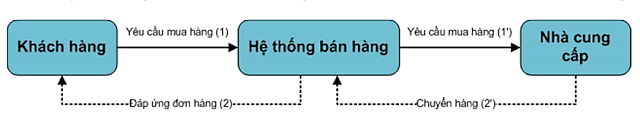
▫ Whole sở hữu Part, tạo và hủy Part.

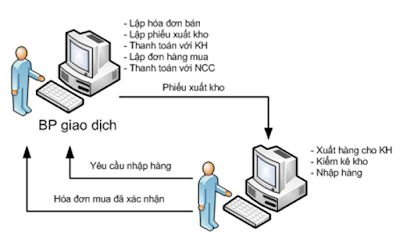
▫ Part bị bỏ đi khi Whole bị bỏ, Part không thể tồn tại nếu Whole không tồn tại.



**SAMPLE – Project Architecture & Detail Design**

### SYSTEM Architecture

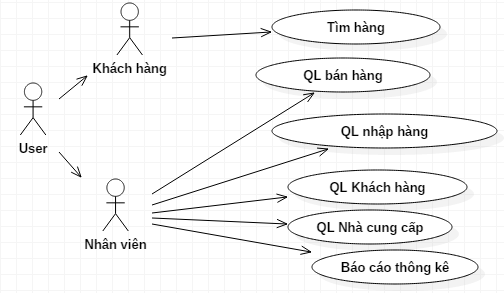




* 1. **USECASE Diagram**

**2.1 Biểu đồ use case tổng quát**

Khái quát chức năng chính của hệ thống. Các chức năng này có tính tổng quát dễ dàng nhìn thấy được trên quan điểm của các tác nhân. Dựa vào yêu cầu của bài toán ta có use case tổngquát như sau:



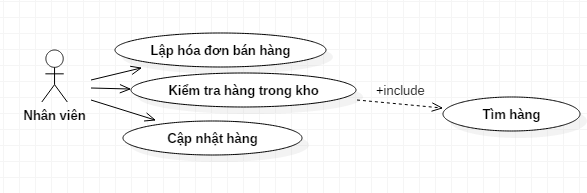
**2.2 Use case quản lý bán hàng**

Tác nhân: Nhân viên

Điểu kiện: Phải đăng nhập được vào hệ thống.

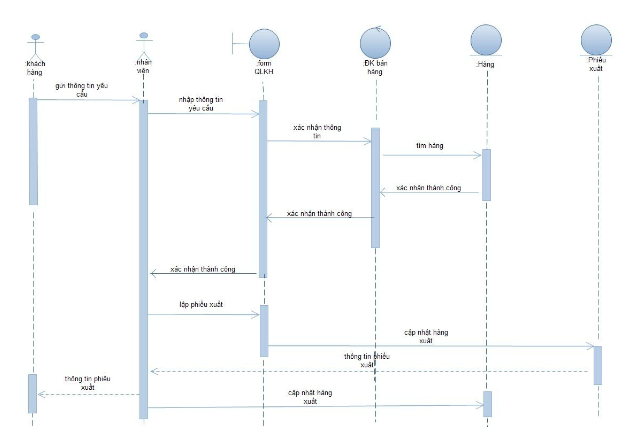
Mô tả: Dựa vào yêu cầu của khách hàng, nhân viên sẽ lập hóa đơn bán hàng, kiểm tra hàng trong kho xem có còn hayđáp ứng được không (tìm hàng), nếu đáp ứng được thì sẽ giao hàng cho khách hàng và cập nhật thông tin hàng bán vào hệ thống.

Nếu hàng trong kho không đáp ứng được thì thông báo cho khách hàng.

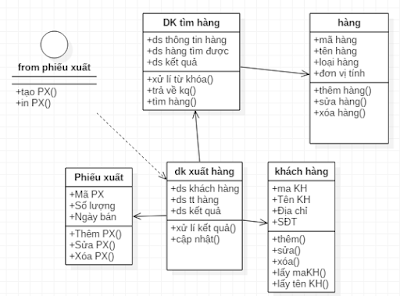
****

* 1. **SEQUENCE Diagram**

**Biểu đồ tuần tự cho chức năng cho chức năng quản lý bán hàng**



* 1. **CLASS Diagram**



* 1. **DB Diagram**
  2. **User Interface Design**