

# Bài 2: TẠO LƯỢC ĐỒ USE - CASE TRONG PTTKPM

Xem bài học trên website để ủng hộ Kteam: [Tạo lược đồ Use - Case trong PTTKPM](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Phân tích thiết kế hệ thống, việc mấu chốt là phải hiểu rõ yêu cầu. Và để hiểu rõ yêu cầu thì chúng ta sẽ phải đọc hiểu yêu cầu. Để thể hiện rằng mình đã hiểu yêu cầu đó, đồng thời có tài liệu phân tích về sau. Thì chúng ta tạo ra sơ đồ **Use – Case**.

Cùng nhau tìm hiểu về sơ đồ **Use – Case** với Kteam nhé!

---

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- Đã từng sử dụng qua vài phần mềm
- Đã từng suy nghĩ đến việc cấu thành của một phần mềm ra sao
- Biết sơ bộ về tư duy logic của lập trình
- Biết sử dụng máy tính cũng như các công cụ thành thạo.
- Đã đọc hiểu rõ bài [GIỚI THIỆU VỀ PHÂN TÍCH THIẾT KẾ PHẦN MỀM](#).

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

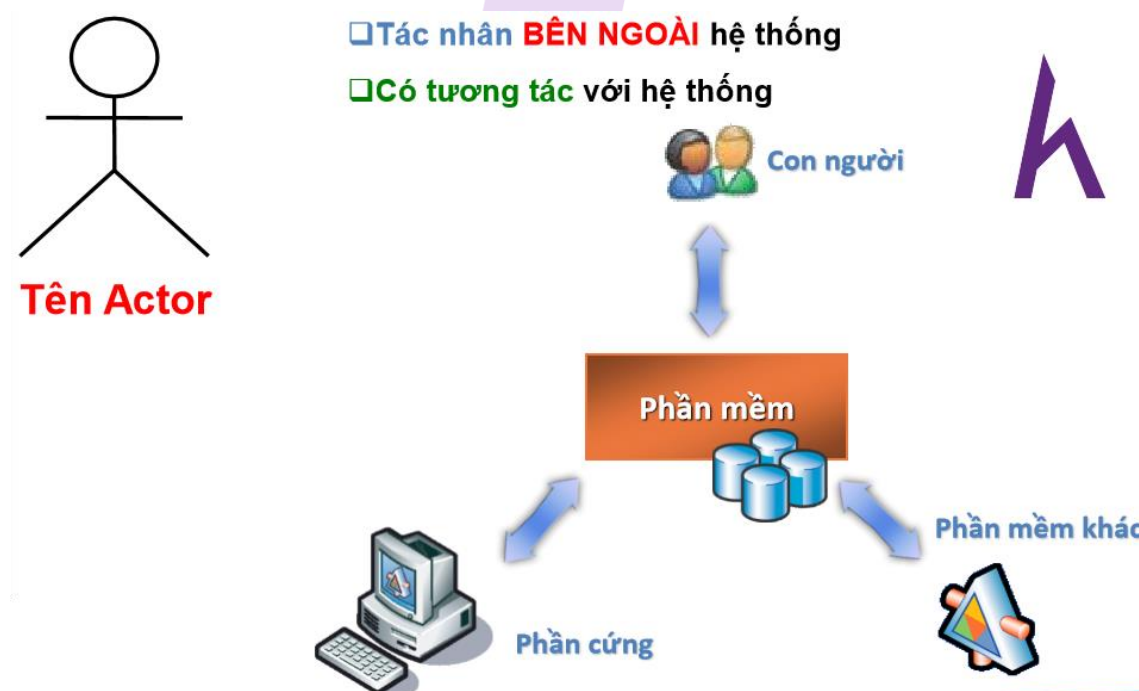
- Khái niệm về Actor và Use - Case
- Các quan hệ trong lược đồ Use – Case: Quan hệ thống quát hóa giữa Actor, quan hệ <<include>> và <<extend>>.
- Sơ đồ Use – Case hoàn chỉnh.
- Đặc tả Use – Case.

# Mô hình hóa yêu cầu

Chỉ mô tả chủ yếu các thông tin liên quan đến việc thực hiện các nghiệp vụ trong thế giới thực, chưa thể hiện rõ nét việc thực hiện các nghiệp vụ trên máy tính.

Mô tả thông quá các văn bản dễ gây ra nhầm lẫn và không trực quan.

## Actor



**Actor** là các tác nhân bên ngoài tác động vào phần mềm:

- Người dùng
- Thiết bị ngoại vi
- Phần mềm khác.

Một nhóm người dùng tương ứng với một Actor.

Mỗi Actor được phép sử dụng một hay nhiều chức năng trong hệ thống như:

- Nhân viên được phép thêm, sửa hóa đơn.
- Admin được phép toàn quyền với hệ thống.

Một chức năng có thể cho phép nhiều Actor sử dụng như:

- Nhân viên và Quản lý đều được phép thông báo giờ làm của mình với hệ thống.

Nhiều Actor có thể có các quyền hạn giống nhau như:

- Admin và Quản lý đều có thể xem thống kê báo cáo của hệ thống.

**Một hay nhiều Actor tùy vào ngữ cảnh:** Với trường hợp nhân viên dùng phần mềm thì có thể có nhiều nhân viên. Nhưng vẫn gọi chung Actor đó là Nhân Viên. Máy quét mã vạch thì chỉ có một tương tác với phần mềm.

## Nhóm người sử dụng

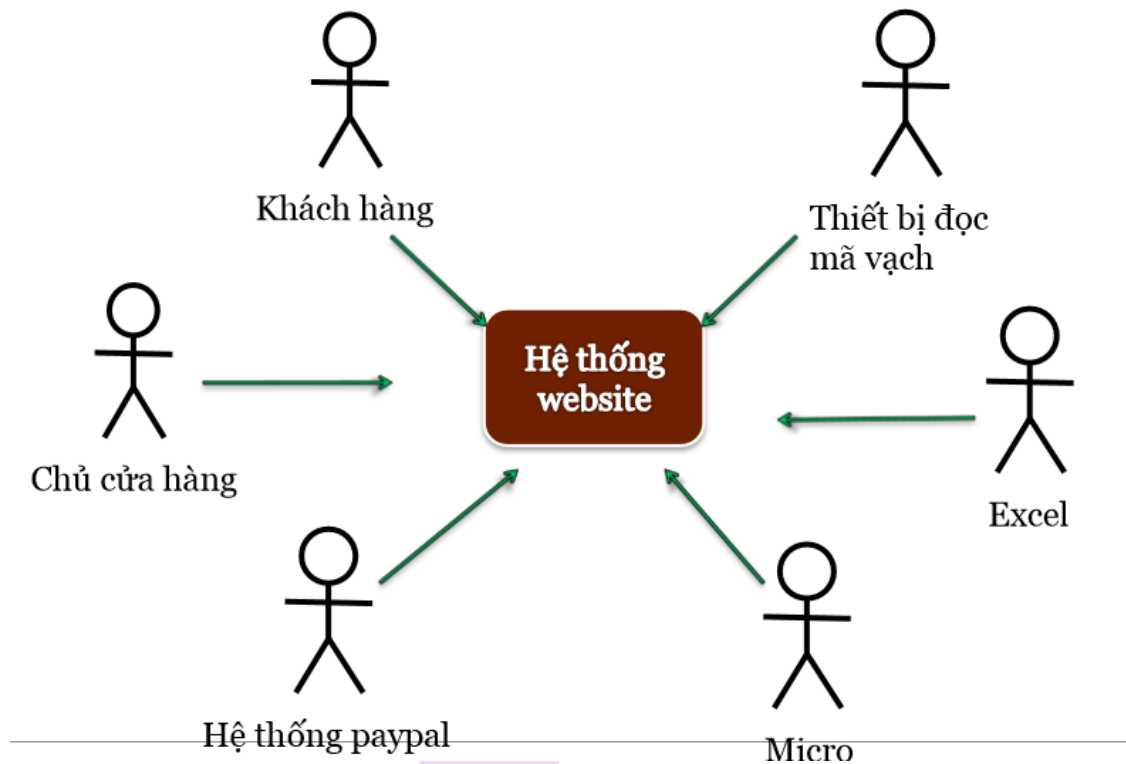
### Xét phần mềm Quản lý học sinh cấp III

STT	Yêu cầu	Nhóm người dùng
1	Tiếp nhận học sinh	Giáo vụ?
2	Lập danh sách lớp	Giáo vụ?
3	Tra cứu học sinh	Mọi người? Phụ huynh? Học sinh?
4	Nhập bảng điểm môn	Giáo viên? Giáo vụ?
5	Xem báo cáo tổng kết	Ban giám hiệu?
6	Thay đổi quy định	Ban giám hiệu? Quản trị hệ thống?


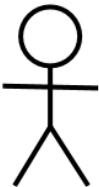
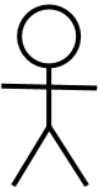
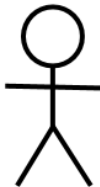

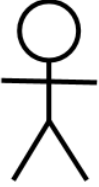
### Xét phần mềm Quản lý thư viện

STT	Yêu cầu	Nhóm người dùng
1	Lập thẻ độc giả	Thủ thư
2	Nhận sách mới	Thủ thư
3	Tra cứu sách	Thủ thư? Độc giả? Khách bất kỳ?
4	Lập phiếu mượn	Thủ thư
5	Nhận trả sách	Thủ thư
6	Thay đổi quy định	Thủ thư? Quản trị hệ thống?

Chúng ta có thể xác định các Actor qua hình sau.




Phân loại các Actor ra theo tác nhân.

Người dùng	Hệ thống khác	Phần cứng	Phần mềm
 Khách hàng	 Hệ thống paypal	 Thiết bị đọc mã vạch	 Excel
 Chủ cửa hàng		 Micro	

## Nhóm phần cứng ngoại vi

- ❑ Phần mềm quản lý Siêu thị:
  - Đọc thông tin từ thiết bị đọc mã vạch
- ❑ Phần mềm quản lý cửa tự động:
  - Đọc thông tin từ camera
  - Phát lệnh điều khiển mở cửa
- ❑ Phần mềm quản lý ra vào các phòng trong công sở
  - Đọc tín hiệu từ đầu đọc thẻ từ
  - Phát lệnh điều khiển mở cửa
- ❑ Phần mềm chống trộm
  - Đọc tín hiệu từ camera, sensor
  - Phát lệnh điều khiển ra loa, đèn, điện thoại...



Các thiết bị ngoại vi  
mà phần mềm  
cần tương tác

Có cần liệt kê  
tất cả thiết bị ngoại vi?

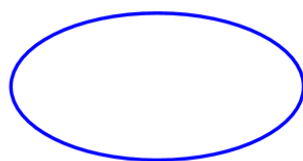
---

## Nhóm phần mềm khác

### Ví dụ:

- Nhập xuất file Excel.
  - Gửi mail thông qua Microsoft Outlook.
  - Phần mềm phân tích dữ liệu...
-

## Use – Case



Use-Case

- Một **Use-Case** là một chuỗi các hành động mà hệ thống thực hiện mang lại một kết quả quan sát được đối với actor.
- Có thể hiểu một Use-Case là một chức năng của hệ thống, mang một ý nghĩa nhất định đối với người dùng

### Ví dụ

#### Xét phần mềm Quản lý học sinh cấp III

STT	Yêu cầu
1	Tiếp nhận học sinh
2	Lập danh sách lớp
3	Tra cứu học sinh
4	Nhập bảng điểm môn
5	Xem báo cáo tổng kết
6	Thay đổi quy định

Trong ví dụ này, có tổng cộng 6 **Use – Case**. Tính năng Thêm, xóa, sửa là tính năng `<<include>>` và `<<extent>>` của **Use – Case** Tiếp nhận học sinh. `<<include>>` và `<<extent>>` sẽ được giới thiệu trong bài.

## Xét phần mềm Quản lý thư viện

STT	Yêu cầu
1	Lập thẻ đọc giả
2	Nhận sách mới
3	Tra cứu sách
4	Lập phiếu mượn
5	Nhận trả sách
6	Thay đổi quy định

❖ Có bao nhiêu Use-case trong Ví dụ này?

## Phần mềm thí nghiệm mạch điện

STT	Yêu cầu
1	Sắp đặt mạch điện
2	Cung cấp nguồn điện
3	Thay đổi thông số
4	Lưu bài thí nghiệm
5	Lấy lại thí nghiệm
6	Thay đổi quy định

❖ Có bao nhiêu Use-case trong Ví dụ này?

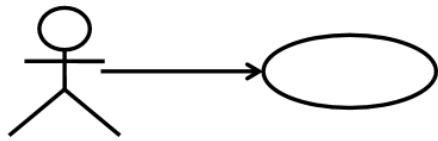
Một ví dụ về Use – Case:



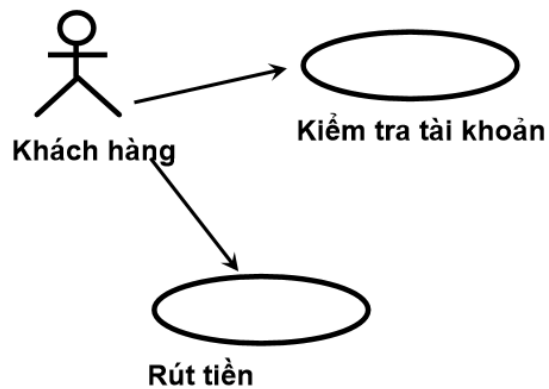
- **Xét trang web ở trên có các chức năng:**
  - **Khách hàng** *đăng ký tài khoản & mua sản phẩm*
  - **Chủ cửa hàng** *chấp nhận các đơn đặt hàng, đăng sản phẩm*
  - Trang web trên có thêm chức năng *xuất báo cáo* ra tập tin **excel**
  - Trang web có chức năng *hỗ trợ khách hàng trực tuyến* bằng cách trao đổi trực tiếp bằng âm thanh của **micro**
  - Trang web có thêm chức năng có khả năng giao tiếp với **thiết bị đọc mã vạch** để *thêm thông tin sản phẩm* vào dữ liệu
  - Trang web hỗ trợ *thanh toán* qua mạng bằng cách kết nối với **"Hệ thống thanh toán Paypal"**



## Sơ đồ Use – Case

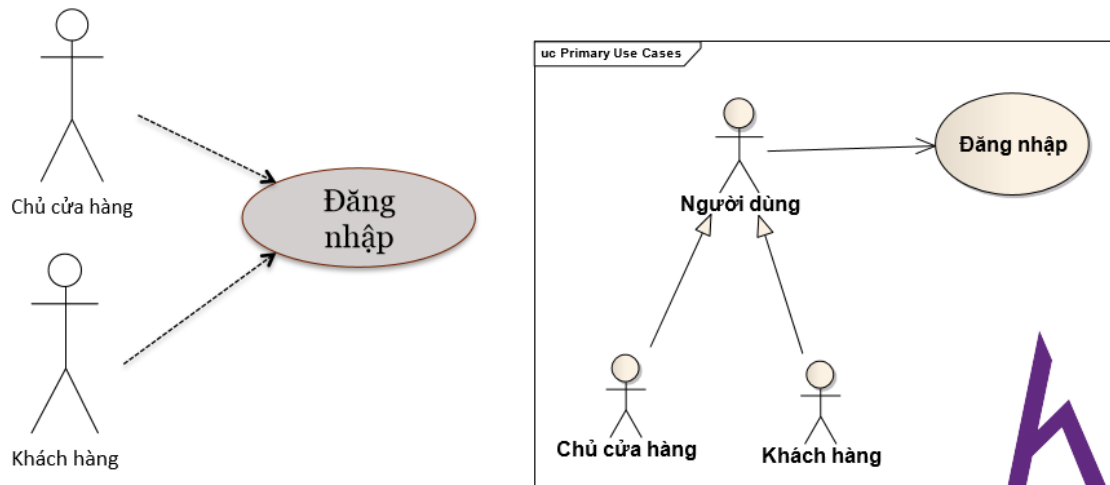


→ Sự tương tác giữa Actor và Use-case  
Chiều của mũi tên thể hiện vai trò chủ động trong sự tương tác



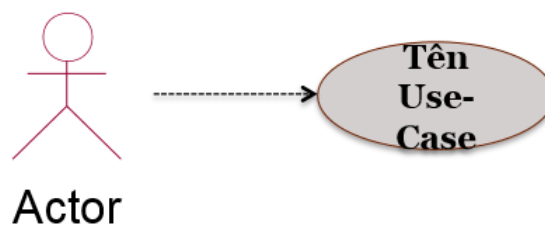
## Tổng quát hóa giữa các Actor

- Khách hàng và chủ cửa hàng đều có thể thực hiện chức năng đăng nhập

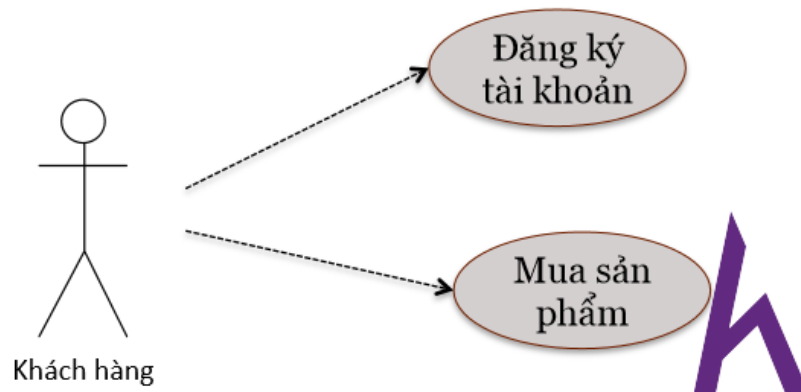


Ta có thể gom những **Use-Case** của nhiều Actor dùng chung thành một. Và đặt cho Actor đó một cái tên riêng. Các Actor có Actor chung có **Use-Case** riêng thì sẽ vẽ tách biệt ra.

Chiều của mũi tên thể hiện vai trò chủ động của tác nhân khi thực hiện Use-Case



## Khách hàng *đăng ký tài khoản & mua sản phẩm*

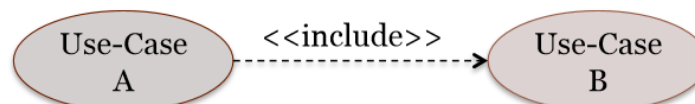


## <<include>> và <<extend>>

Có 2 loại quan hệ:

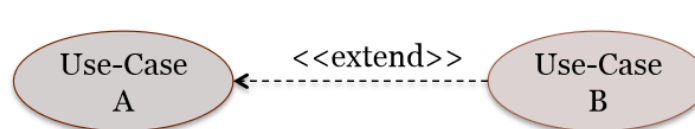
○ **Include:** Use-Case A <<include>> Use-Case B:

- ✦ trong quá trình thực thi Use-Case A sẽ luôn cần thực hiện Use-Case B



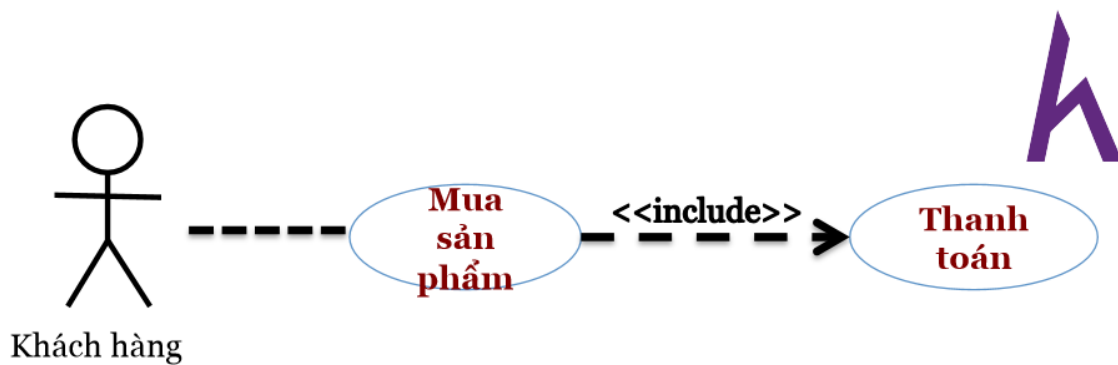
○ **Extend:** Use-Case A <<extend>> Use-Case B:

- ✦ trong quá trình thực thi Use-Case A, trong 1 số trường hợp sẽ thực hiện Use-Case B

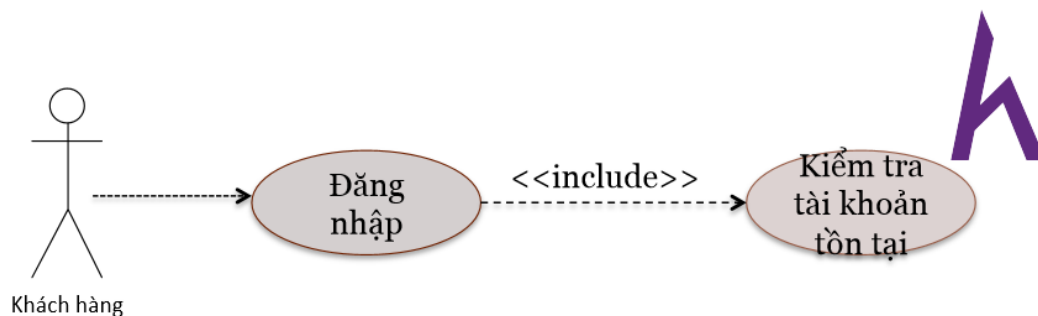


## Quan hệ <<include>>

- Quá trình **khách hàng** *mua hàng luôn luôn* phải thực hiện bước *thanh toán* tiền cho **chủ cửa hàng**

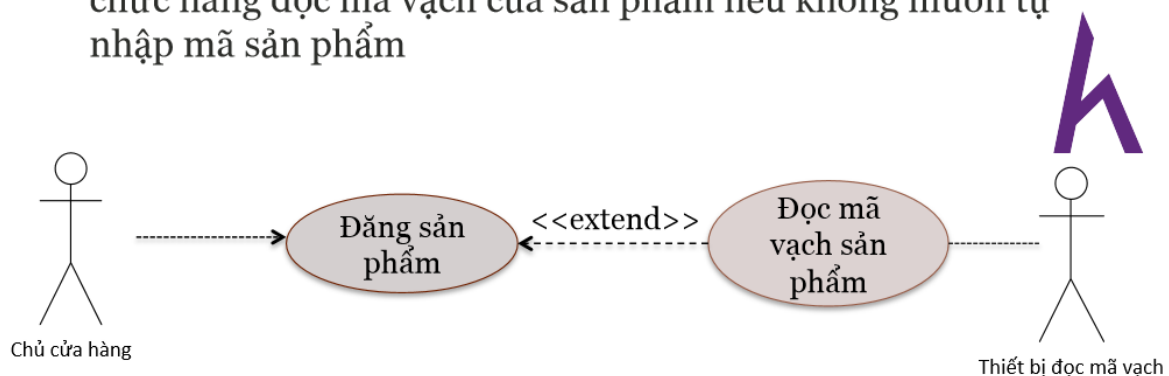


- Khi **khách hàng** thực hiện *đăng nhập luôn luôn* phải thực hiện bước *kiểm tra tài khoản tồn tại*

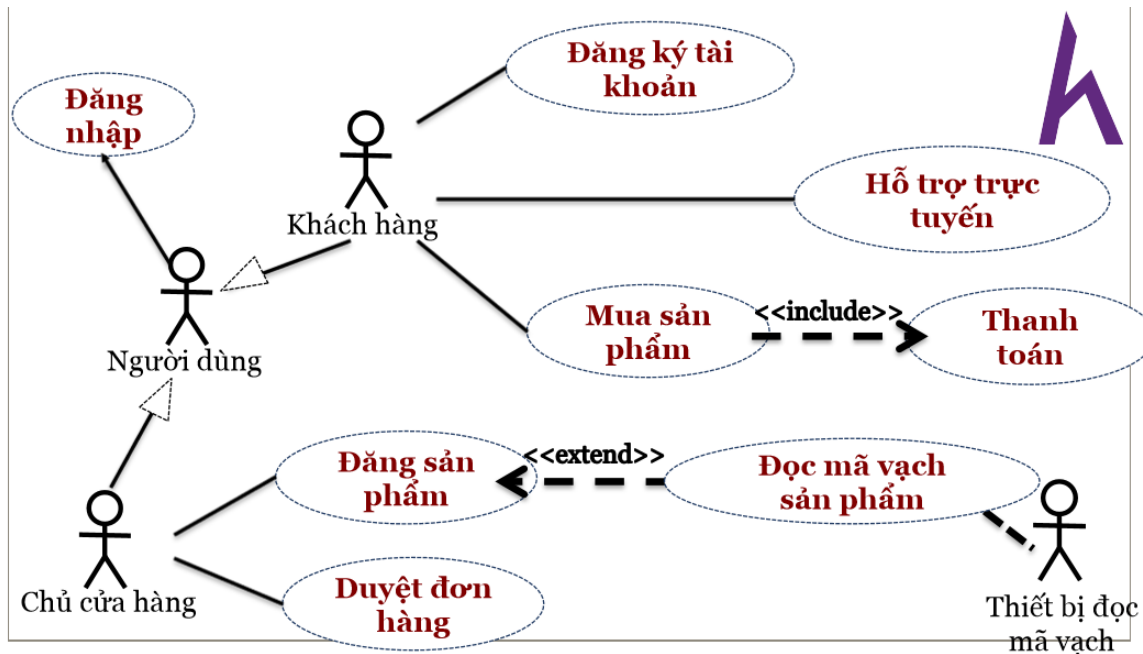


## Quan hệ <<extend>>

- Khi chủ cửa hàng thêm sản phẩm vào dữ liệu có thể sử dụng chức năng đọc mã vạch của sản phẩm nếu không muốn tự nhập mã sản phẩm



## Ví dụ về Sơ đồ Use – Case



Chúng ta có thể dùng các công cụ để vẽ sơ đồ Use – Case như:

- IBM Rational Rose
- Enterprise Architecture
- Visio


## Đặc tả Use – Case

Sau khi mô hình hóa yêu cầu hệ thống. Chúng ta cần Đặc Tả lại hệ thống theo phong cách chuyên ngành. Với mỗi **Use – Case** sẽ có một đặc tả riêng.

Tiêu đề	Nội dung
Tên Use-Case	Tên ngắn gọn của Use-Case
Mô tả	Mô tả ngắn gọn nội dung của Use-Case
Sự kiện kích hoạt	Sự kiện kích hoạt Use-Case hoạt động
Actors	Các tác nhân liên quan đến Use-Case
Use-Case liên quan	Các Use-Case được gọi khi thực hiện Use-Case này. Gồm 2 loại include và extend.
Tiền điều kiện	Các điều kiện cần thiết phải có trước khi Use-Case thực hiện
Hậu điều kiện	Trạng thái của hệ thống sau khi usecase được thực hiện
Biến thể	Điều kiện dẫn đến phân nhánh. Sẽ quay lại luồng sự kiện chính
Ngoại lệ	Những điều kiện không mong đợi dẫn đến việc phân nhánh. Không quay lại luồng sự kiện chính.

Chúng ta sẽ cùng tham khảo về Ví dụ **Use – Case** đăng nhập.

Use-Case	Nội dung
Tên Use-Case	Đăng nhập
Mô tả	Use-case cho phép người dùng đăng nhập vào hệ thống để thực hiện những chức năng của mình
Actor	Người dùng
Điều kiện kích hoạt	Khi người dùng chọn chức năng đăng nhập từ trang chủ của hệ thống
Tiền điều kiện	Người dùng phải có tài khoản trên hệ thống
Hậu điều kiện	Người dùng đăng nhập thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>Hệ thống hiển thị màn hình đăng nhập</li> <li>Người dùng nhập tên đăng nhập và mật khẩu.</li> <li>Hệ thống hiển thị kiểm tra thông tin đăng nhập.</li> <li>Nếu thành công hệ thống hiển thị màn hình đăng nhập thành công.</li> <li>Kết thúc Use-Case.</li> </ol>

<p>Luồng sự kiện phụ</p> 	<p><u><b>A1 - Mật khẩu không hợp lệ:</b></u> Khi người dùng nhập sai tên đăng nhập và mật khẩu</p> <ol style="list-style-type: none"><li>1. Hệ thống hiển thị lại màn hình đăng nhập để người dùng nhập lại thông tin kèm theo thông báo tên đăng nhập và mật khẩu bị sai.</li><li>2. Quay lại bước 2 trong luồng sự kiện chính.</li></ol> <p><u><b>A2 – Quên mật khẩu:</b></u> Khi người dùng chọn chức năng quên mật khẩu trên màn hình đăng nhập.</p> <ol style="list-style-type: none"><li>1. Hệ thống hiển thị màn hình để người dùng nhập email.</li><li>2. Người dùng nhập email và chọn nút chức năng Lấy lại mật khẩu</li><li>3. Hệ thống kiểm tra email hợp lệ và gửi liên kết để reset mật khẩu cho người dùng qua email.</li><li>4. Hệ thống hiển thị màn hình thông báo thành công.</li><li>5. Use case kết thúc.</li></ol>
--	--

## Kết luận

Bài viết có sử dụng nhiều hình ảnh của tài liệu PTTKHT của trường ĐH KHTN

Qua bài này các bạn đã nắm được sơ đồ Use – Case là gì. Các luồng sự kiện.

Bài sau chúng ta sẽ cùng tìm hiểu cách để tạo ra [TAO ACTIVITY – DIAGRAM](#) trong phân tích thiết kế phần mềm.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **“Luyện tập – Thử thách – Không ngại khó”**.