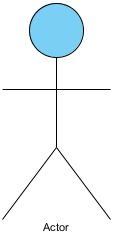
**Tìm hiểu về use case**

1. Khái niệm

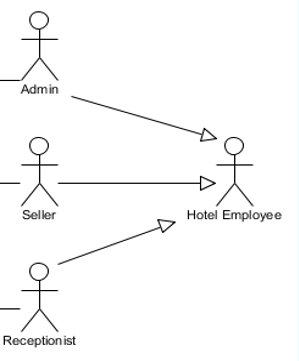
* Use case là một kỹ thuật được dùng trong [kỹ thuật phần mềm](https://vi.wikipedia.org/wiki/K%E1%BB%B9_thu%E1%BA%ADt_ph%E1%BA%A7n_m%E1%BB%81m) và [hệ thống](https://vi.wikipedia.org/w/index.php?title=K%E1%BB%B9_thu%E1%BA%ADt_h%E1%BB%87_th%E1%BB%91ng&action=edit&redlink=1) để nắm bắt [yêu cầu chức năng](https://vi.wikipedia.org/w/index.php?title=Y%C3%AAu_c%E1%BA%A7u_ch%E1%BB%A9c_n%C4%83ng&action=edit&redlink=1) của hệ thống
* Use case giúp mô tải tương tác đặc trưng giữa người dùng và hệ thống. Nó thể hiện ứng xử của hệ thống đối với bên ngoài trong 1 hoàn cảnh điều kiện nhất định. Nó mô tả các yêu cầu đối với hệ thống nghĩa là nó mô tả những j hệ thống phải làm chứ ko mô tả hệ thống làm như nào. Tập hợp các use case sẽ là tất cả các trường hợp mà hệ thống có thể được sử dụng .

1. Actor:

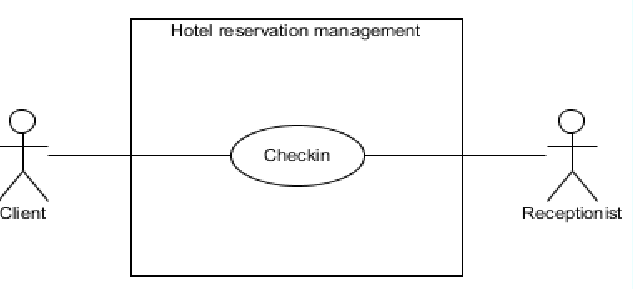
* Actor: Tác nhân, người dùng tương ứng với use đó, actor thường là người khởi tạo use case hoặc tác nhận chính để use case hoạt động.



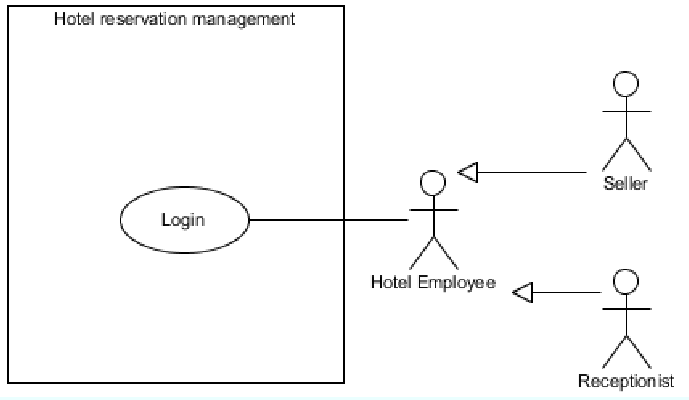
* 1 người có thể là nhiều actor khác nhau, 1 actor có thể tham gia vào nhiều use case khác nhau.
* Actor có thể là 1 người, 1 tổ chức, hay 1 thiết bị đầu cuối như máy in, điện thoại, …
* Các actor có thể có quan hệ kế thừa:



* Use case có 2 actor:

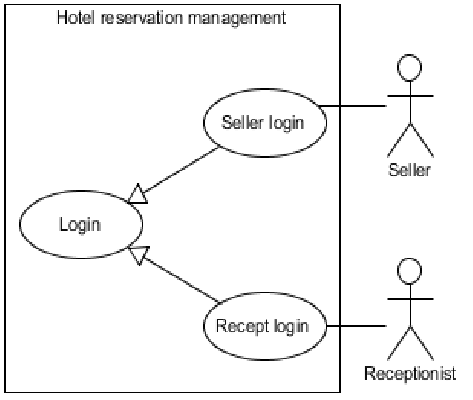


Nghĩa là phải có sự có mặt của 2 ng thì use case mới thực hiện được.

* Actor trừu tượng:
* 

được tạo ra để giảo quyết vấn đề khi nhiều actor có cùng hành động nhưng ko cần phải có mặt tất cả thì hành động mới xảy ra.

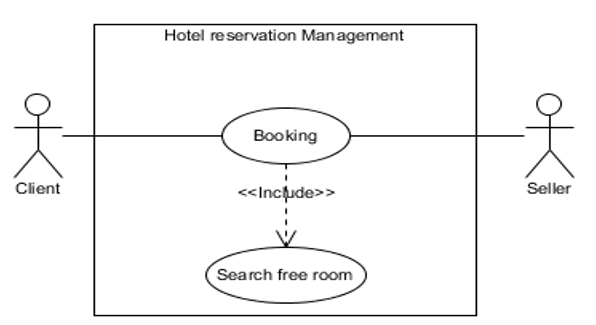
* Tương tự có use case trưù tượng.



Quan hệ giữa các use case : include, extend, generalize.()

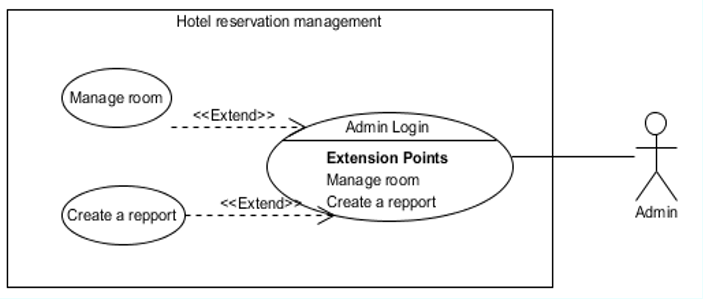
1. Include: Uc A có quan hệ include với uc B nếu việc hoàn thành B là một phần công việc để hoàn thành A

* Nếu không hoàn thành B thì A không thể hoàn thành
* Việc hoàn thành B có thể lặp lại nhiều lần, thì người ta tạo ra uc riêng để tránh trùng lặp
* Quan hệ này được biểu diễn bằng một mũi tên nét đứt đi từ A đến B. Mũi tên có nhãn « include »



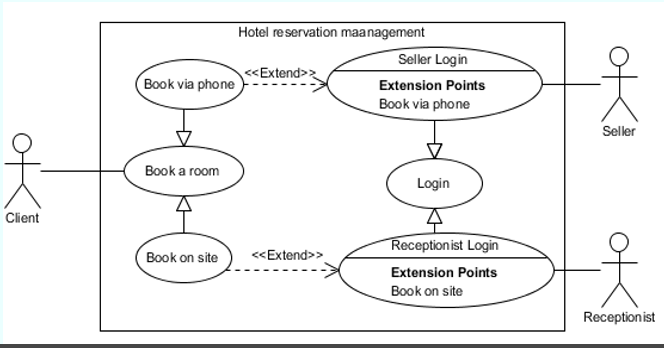
1. Extend:

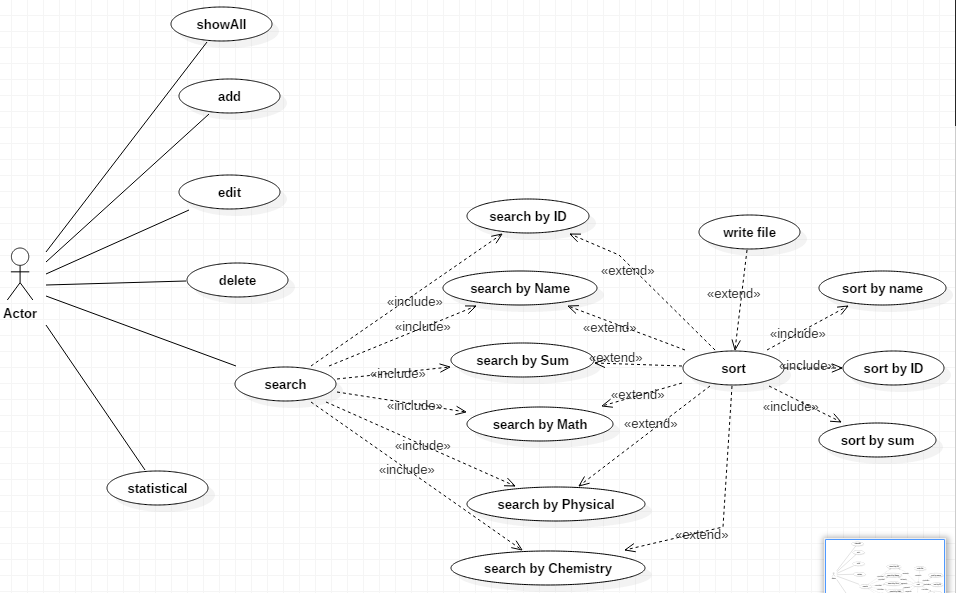
* Uc A có quan hệ extend với uc B nếu việc hoàn thành A là một tùy chọn công việc để hoàn thành B
* Trong một số trường hợp, làm B bao gồm làm A
* Nhưng trong một số trường hợp khác, làm B không cần làm A
* Quan hệ này được biểu diễn bằng một mũi tên nét đứt đi từ A đến B. Mũi tên có nhãn « extend »



1. Quan hệ kế thừa:

* Uc A có quan hệ kế thừa với uc B nếu B là một phần dạng tổng quát của A, hay A là một thể hiện chi tiết của B
* Quan hệ này được biểu diễn bằng một mũi tên nét liền (đầu hình tam giác rỗng) đi từ A đến B.





Mô tả use case:

* add: use case này cho phép người sử dụng thêm 1 sinh viên vào hệ thống.
* edit: use case này cho phép người sử dụng sửa 1 sinh viên trong hện thống.
* delete: use case này cho phép người sử dụng xoá 1 sinh viên trong hệ thống.
* search: use case này cho phép người sử dụng tìm kiếm sinh viên.
* search by ID: use case này cho phép người sử dụng tìm kiếm sinh viên theo ID.
* search by Name : use case này cho phép người sử dụng tìm kiếm sinh viên theo tên
* search by Sum: use case này cho phép người sử dụng tìm kiếm sinh viên theo tổng điểm.
* search by Math: use case này cho phép người sử dụng tìm kiếm sinh viên theo điểm toán.
* search by Physical: use case này cho phép người sử dụng tìm kiếm sinh viên theo điểm Lý.
* search by Chemistry: use case này cho phép người sử dụng tìm kiếm sinh viên theo điểm hoá.
* sort : use case này cho phép người sử dụng sắp xếp sinh viên sau khi đã tìm kiếm.
* sort by ID: use case này cho phép người sử dụng sắp xếp sinh viên sau khi đã tìm kiếm theo ID.
* sort by name: use case này cho phép người sử dụng sắp xếp sinh viên sau khi đã tìm kiếm theo tên.
* sort by sum: use case này cho phép người sử dụng sắp xếp sinh viên sau khi đã tìm kiếm theo tổng điểm.
* statistical : use case này cho phép người sử dụng thống kê sinh viên.
* write file : use case này cho phép người sử dụng ghi file sau khi đã tìm kiếm.
* showAll: use case này cho phép người sử dụng xem tất cả các sinh vên trong danh sách.