

Chương 6

MÔ HÌNH HOÁ USE CASE

Mục tiêu

Sau khi học xong sinh viên có thể:

- Hiểu ý nghĩa của việc sử dụng sơ đồ use case trong biểu diễn yêu cầu hệ thống
- Xác định được các tác nhân và mối quan hệ giữa các tác nhân của một hệ thống phần mềm
- Xác định được các use case biểu diễn chức năng phần mềm hệ thống và mối quan hệ giữa tác nhân và use case nhằm xây dựng sơ đồ use case mô tả yêu cầu phần mềm hệ thống
- Tinh chế sơ đồ use case nhằm làm gia tăng tính diễn đạt, tính tái sử dụng qua việc sử dụng các liên kết <<extend>>, <<include>>

Giới thiệu

Trong giai đoạn phân tích, kết quả của quá trình khảo sát yêu cầu phản ánh quá trình làm việc của người phát triển với người sử dụng. Các kết quả này phải nhắm đến yếu tố của người dùng. Có nghĩa là người phát triển trước tiên phải diễn đạt bức tranh của hệ thống tương lai theo cách nhìn của người sử dụng. Điều này sẽ giúp cho người dùng có thể thấy được hệ thống sẽ làm thỏa mãn các yêu cầu như thế nào và đó chính là chìa khoá đầu vào cho việc phát triển hệ thống trong các giai đoạn về sau. Một công cụ giúp diễn đạt điều này chính là mô hình use case.

Jacobson và cộng sự của ông (1992) là những người tiên phong trong việc sử dụng mô hình use case để phân tích yêu cầu hệ thống. Bởi vì mô hình use case đặt trọng tâm để biểu diễn hệ thống hiện tại làm gì, hệ thống mới sẽ làm gì và môi trường của nó. Nó giúp cho người phát triển có thể hiểu rõ về yêu cầu chức năng hệ thống mà không quan tâm đến chức năng này được cài đặt như thế nào.

Để hiểu yêu cầu của hệ thống, chúng ta phải tìm ra người dùng sẽ sử dụng hệ thống như thế nào. Do đó từ một quan điểm người dùng chúng ta phát hiện các tình huống sử dụng khác nhau của người dùng, các tình huống này được thiết lập bởi các use case, tổng hợp các use case và tác nhân cùng với quan hệ giữa chúng sẽ cho ta một mô hình use case mô tả yêu cầu của hệ thống.

Trong chương 5, quá trình mô hình hoá nghiệp vụ được áp dụng đối với các hệ thống nghiệp vụ và kết quả của nó sẽ cung cấp sơ đồ use case từ việc tổng nhất các yêu cầu hệ thống phần mềm để tự động hoá hoạt động của hệ thống nghiệp vụ đó. Tuy nhiên, trong những hệ thống mà không có hoạt động nghiệp vụ (ví dụ: hệ thống nhúng), hoặc các nghiệp vụ của hệ thống không quá phức tạp hoặc không quan tâm để mô hình hoá nghiệp vụ thì việc xây dựng mô hình use case phần mềm sẽ là bước tiếp cận mô hình hoá đầu tiên về hệ thống. Một tiến trình xây dựng sơ đồ use case bao gồm các bước sau:

- Xác định tác nhân hệ thống
 - o Ai đang sử dụng hệ thống?
 - o Hoặc trong trường hợp phát triển mới thì ai sẽ sử dụng hệ thống?
- Phát triển use case
 - o Người dùng (tác nhân) đang làm gì với hệ thống?

- Hoặc trong trường hợp hệ thống mới thì người dùng sẽ làm gì với hệ thống?
- Xây dựng sơ đồ use case
 - Xác định mối quan hệ giữa tác nhân – use case
 - Xác định mối quan hệ giữa các use case
- Phân chia sơ đồ use case thành các gói (package)

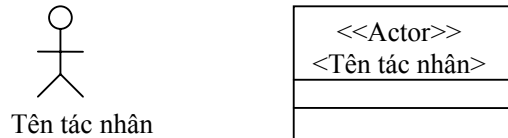
Xác định tác nhân

Tác nhân (actor)

Ý nghĩa: một tác nhân là một đối tượng bên ngoài hệ thống giao tiếp với hệ thống theo một trong những hình thức sau:

- Tương tác, trao đổi thông tin với hệ thống hoặc sử dụng chức năng hệ thống
- Cung cấp đầu vào hoặc nhận các đầu ra từ hệ thống
- Không điều khiển hoạt động của hệ thống

Ký hiệu

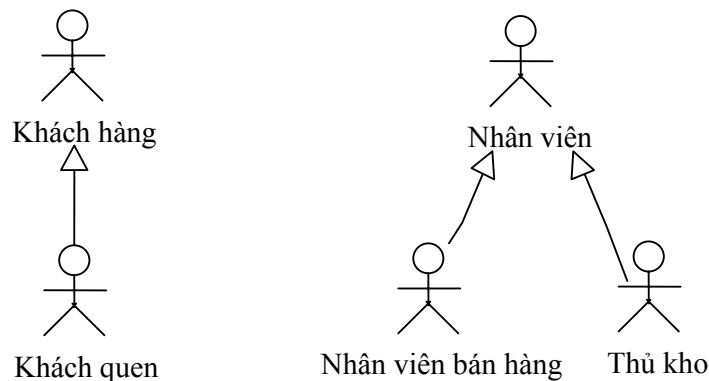


Tên tác nhân: tên tác nhân là một **danh từ**

Quan hệ giữa các tác nhân:

Là quan hệ tổng quát hóa và chuyên biệt hoá

Ví dụ:

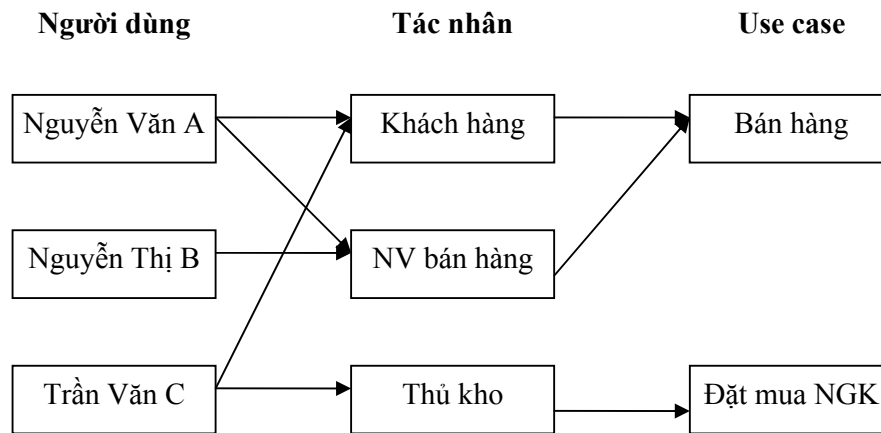


Xác định tác nhân

Xác định tác nhân cũng được xem có tầm quan trọng như xác định class, use case, liên kết,....

Khi xác định người sử dụng phần mềm hệ thống, chúng ta đừng quan trọng vấn đề quan sát người nào đang sử dụng hệ thống mà chúng ta nên xác định xem vai trò chịu trách nhiệm trong việc sử dụng hệ thống. Nghĩa là tác động lên hệ thống theo nghĩa cung cấp thông tin cho hệ thống hoặc nhận kết quả xử lý từ hệ thống.

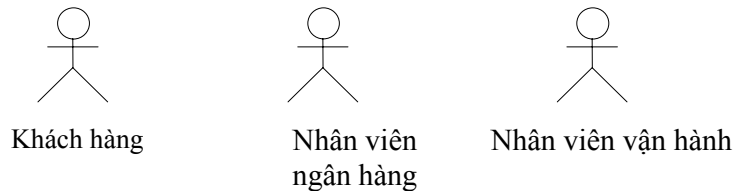
Tác nhân được hiểu là một vai trò tham gia vào hệ thống không giống như một con người cụ thể hoặc một công việc. Một đối tượng có thể tham gia vào một hoặc nhiều vai trò



Qua quá trình khảo sát và phân tích tài liệu hệ thống, chúng ta có thể nhận ra các tác nhân thông qua các câu hỏi sau:

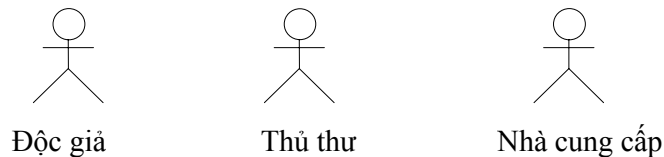
- Ai đang sử dụng hệ thống? Hoặc ai được tác động bởi hệ thống? Hoặc nhóm đối tượng nào cần hệ thống trợ giúp để làm công việc? (tác nhân chính)
- Ai tác động tới hệ thống? Những nhóm đối tượng nào hệ thống cần để thực hiện hoạt động của nó (hoạt động gồm chức năng chính và chức năng phụ, như là chức năng quản trị)?
- Những phần cứng hoặc hệ thống bên ngoài nào sử dụng hệ thống?

Ví dụ: trong hoạt động của máy ATM của một ngân hàng, các tác nhân được xác định là:



Trong đó, các tác nhân Khách hàng, Nhân viên ngân hàng là các tác nhân chính (primary actor) của hệ thống ATM. Bởi vì khách hàng là mục tiêu mà hệ thống tương tác; Nhân viên ngân hàng sử dụng hệ thống để trợ giúp công việc. Trong khi đó, Nhân viên vận hành là tác nhân phụ (secondary actor) bởi vì tác nhân này đảm nhận những chức năng phụ mà hệ thống cần có để thực hiện hoạt động của nó.

Hoặc trong một thư viện của trường đại học, các tác nhân của hệ thống phần mềm quản lý thư viện gồm:



Xác định use case

Use case

Một Use case được xem như một chức năng hệ thống từ quan điểm người dùng, như vậy tập hợp tất cả use case biểu diễn bộ mặt của hệ thống bao gồm các chức năng cần có để cung cấp cho các đối tượng tương tác làm việc với hệ thống. Như vậy, use case dùng để mô tả yêu cầu

của hệ thống mới về mặt chức năng, mỗi chức năng sẽ được biểu diễn như một hoặc nhiều use case.

Ví dụ: hệ thống cửa hàng NGK ta có một vài use case

Bán hàng, quản trị tồn kho,...

Ký hiệu



Xác định use case

Chúng ta bắt đầu từ tập các tác nhân đã xác định trong bước đầu tiên. Ứng với mỗi tác nhân:

- Tìm các nhiệm vụ và chức năng mà tác nhân sẽ thi hành hoặc hệ thống cần tác nhân để thi hành và mô hình hoá nó như là use case. Use case sẽ đại diện một dòng sự kiện dẫn tới một mục tiêu rõ ràng (hoặc trong một vài trường hợp, dẫn tới một vài mục tiêu riêng biệt có thể là các phương án thay thế cho tác nhân hoặc cho hệ thống so với dòng sự kiện chính)
- Đặt tên cho use case: tên use case nên đặt nhằm phản ánh một mô tả tổng quan về chức năng của use case. Tên nên đề cập những gì xảy ra khi một thể hiện của use case được thi hành. Một hình thức đặt tên use case phổ biến là : động từ (do) + danh từ (what).
- Mô tả use case một cách ngắn gọn bằng việc áp dụng các thuật ngữ gần gũi với người sử dụng. Điều này sẽ làm cho mô tả use case ít mơ hồ.

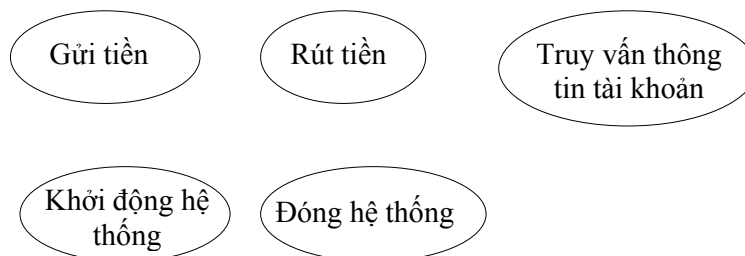
Ví dụ: trong hệ thống ATM

Tác nhân Khách hàng sẽ sử dụng hệ thống qua các chức năng:

- Gửi tiền
- Rút tiền
- Truy vấn thông tin về tài khoản

Tác nhân Nhân viên vận hành sẽ sử dụng các chức năng

- Khởi động hệ thống
- Đóng hệ thống



Gửi tiền: khách hàng đăng nhập vào hệ thống và yêu cầu gửi tiền vào tài khoản. Khách hàng sẽ xác định tài khoản và số tiền gửi, hệ thống sẽ tạo một giao tác gửi tiền và lưu vào hệ thống. Các bước như sau:

- Yêu cầu xác định tài khoản
- Hệ thống hỏi số tiền gửi

- Nhập vào số tiền gửi
- Khách hàng đưa tiền vào bao thư và chuyển vào máy ATM

Rút tiền: khách hàng đăng nhập hệ thống và yêu cầu rút tiền từ tài khoản. Khách hàng xác định tài khoản và lượng tiền rút. Sau khi kiểm tra số dư tài khoản còn đủ, hệ thống sẽ tạo một giao tác rút tiền và lưu vào hệ thống. Các bước như sau:

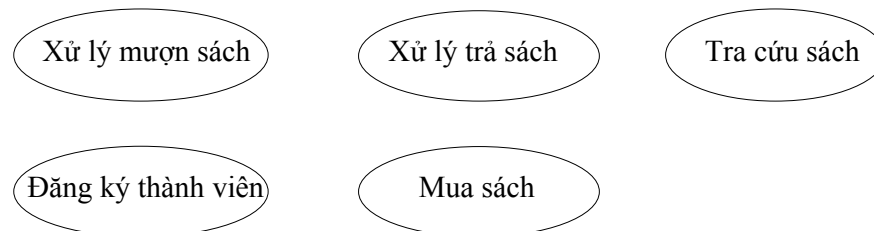
- Yêu cầu xác định tài khoản
- Yêu cầu xác định số tiền cần rút
- Nhập số tiền rút
- Kiểm tra số dư có đủ không
- Chuyển tiền ra ngoài

Truy vấn thông tin tài khoản: khách hàng đăng nhập vào hệ thống và yêu cầu xem thông tin về các giao dịch của tài khoản. Hệ thống hiển thị các thông tin về các giao tác đã tạo lên màn hình cho khách hàng.

Khởi động hệ thống: hệ thống được khởi động khi nhân viên vận hành bật công tắc của máy. Nhân viên vận hành sẽ được yêu cầu nhập vào số tiền hiện hành của máy nằm trong két đựng tiền. Sau đó, hệ thống sẽ thiết lập một kết nối tới ngân hàng và các dịch vụ của máy ATM bắt đầu vận hành.

Đóng hệ thống: hệ thống được đóng lại khi nhân viên vận hành đảm bảo rằng không có khách hàng nào đang sử dụng máy. Khi đó, nhân viên vận hành sẽ lấy các bao tiền gửi ra, bổ sung lượng tiền, giấy,...

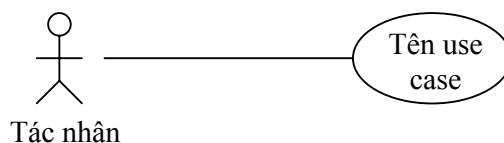
Trong hệ thống quản lý thư viện, các use case được xác định như sau:



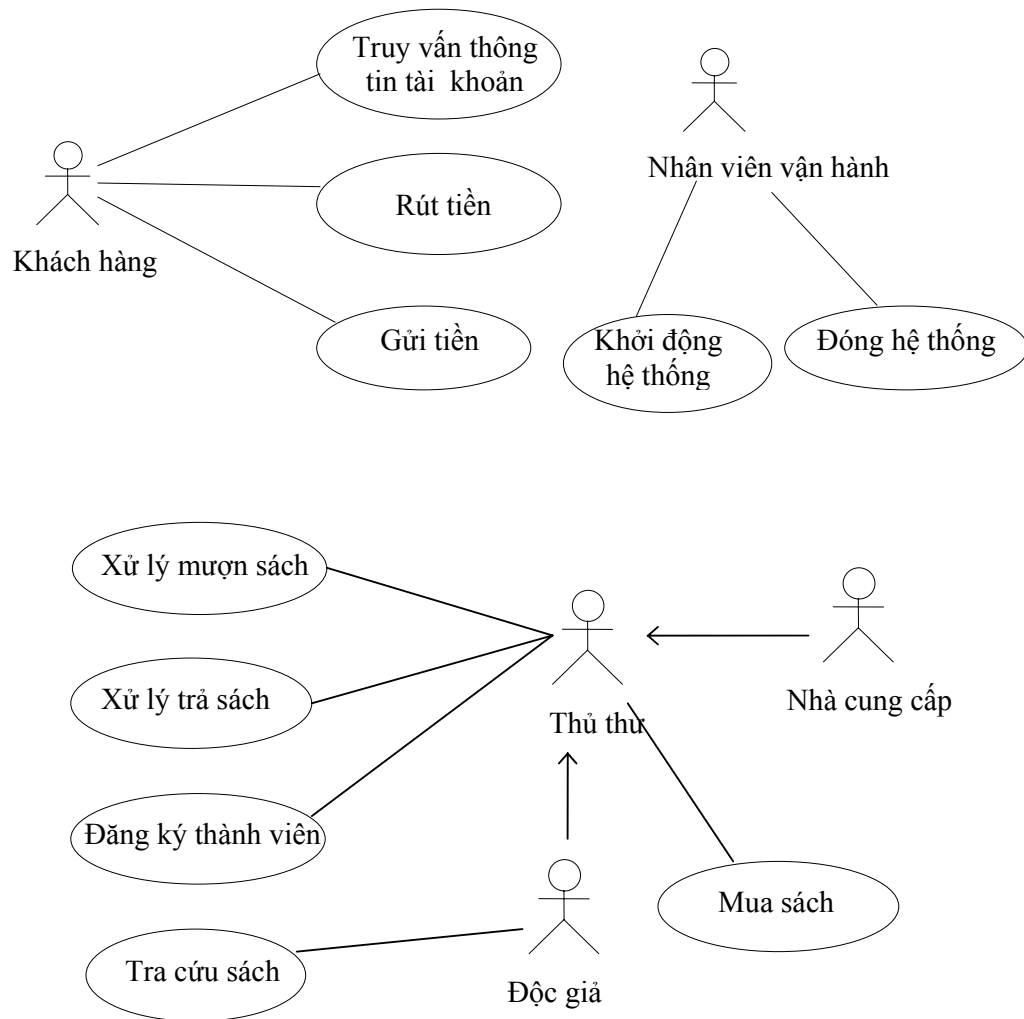
Xác định mối quan hệ

Quan hệ tác nhân – use case: quan hệ này cho biết tác nhân sẽ tương tác với use case. Một use case luôn luôn được khởi tạo bởi một tác nhân và có thể tương tác với nhiều tác nhân.

Ký hiệu



Ví dụ:



Mối quan hệ giữa các use case

Việc mô tả use case có thể sẽ khó hiểu nếu use case này chứa đựng nhiều dòng phụ hoặc dòng ngoại lệ chỉ xử lý cho những sự kiện trong những điều kiện đặc biệt. Để làm đơn giản mô tả này chúng ta sử dụng thêm các mối kết hợp <<extend>> và <<include>>.

Liên kết mở rộng (<<extend>>): được dùng khi chúng ta có một use case tương tự như use case khác nhưng có nhiều hơn một vài xử lý đặc biệt. Giống như liên kết tổng quát - chuyên biệt, trong đó, use case chuyên biệt là một mở rộng của use case tổng quát bằng việc đưa thêm vào các hoạt động hoặc ngữ nghĩa mới vào use case tổng quát, hoặc bỏ qua hoạt động của use case tổng quát.

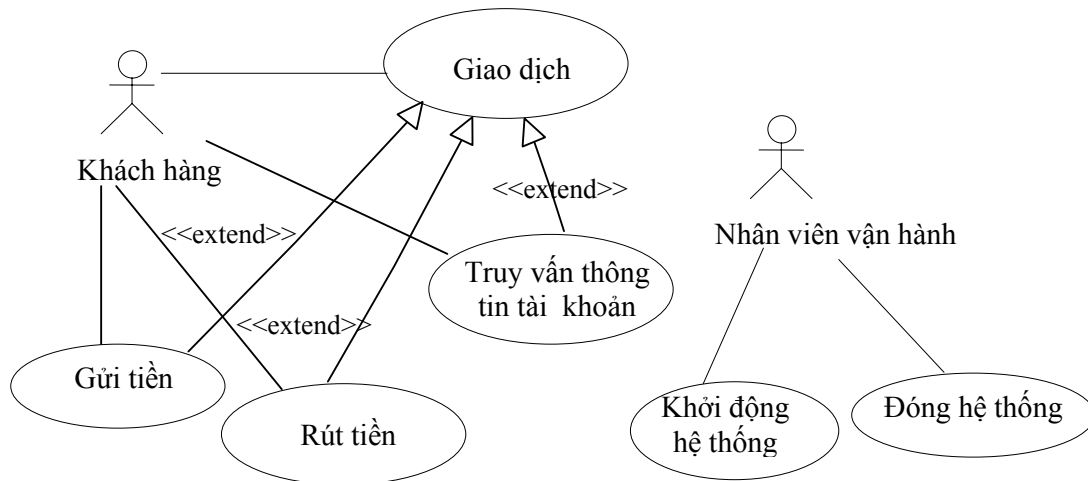
Ví dụ: giả sử Đăng nhập là một use case cơ bản. Use case này sẽ đại diện cho tất cả những gì được xem là thực hiện đăng nhập một cách xuyên suốt. Tuy nhiên, nhiều vấn đề có thể tác động đến dòng sự kiện chính. Ví dụ, mã số PIN không hợp lệ, hoặc thẻ không đọc được do bị hư,... Do đó, chúng ta không phải luôn luôn thi hành các hoạt động thường xuyên của một use case được cho và như vậy, cần thiết tạo ra các use case mới để giải quyết những tình huống mới. Tất nhiên, chúng ta có thể đưa vào use case cơ bản các nội dung xử lý đặc biệt đó. Tuy nhiên, điều này có thể dẫn đến sự phức tạp với nhiều luận lý riêng biệt và sẽ làm giảm vai trò của dòng chính.

Để giải quyết vấn đề này chúng ta có thể sử dụng quan hệ <<extend>>. Ở đây chúng ta gom các xử lý cơ bản hoặc bình thường vào trong một use case (cơ bản). Các xử lý đặc biệt vào

những use case (chuyên biệt) khác. Rồi tạo một liên kết <<extend>> giữa use case cơ bản tới các use case chuyên biệt để khai báo rằng: ngoài xử lý dòng chính (cơ bản), use case cơ bản có mở rộng đến các tình huống xử lý đặc biệt được giải quyết trong các use case chuyên biệt.



Tạo một use case tổng quát có tên là Giao dịch của các use case Rút tiền, Gửi tiền và Truy vấn thông tin tài khoản. Tạo các liên kết <<extend>> từ use case Giao dịch đến các use case này. Như vậy, một rút tiền, hoặc gửi tiền, hoặc truy vấn thông tin tài khoản là một loại giao dịch mà khách hàng có thể sử dụng trên máy ATM. Có nghĩa rằng, các xử lý trong use case Giao dịch sẽ cung cấp một dòng chung và khi khách hàng chọn một loại giao dịch đặc biệt nào đó thì use case này sẽ mở rộng việc giải quyết thông qua các use case chuyên biệt.

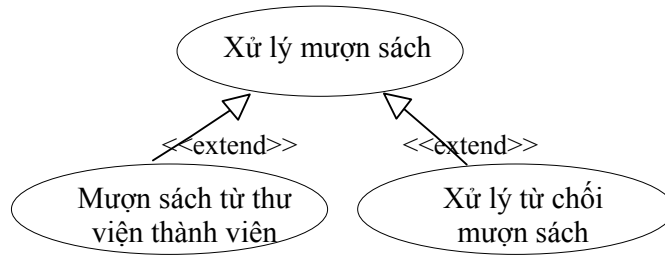


Giao dịch: khách hàng tương tác với hệ thống bắt đầu bằng việc đăng nhập hệ thống. Sau khi đăng nhập, khách hàng có thể thực hiện các giao dịch. Sau đây là các bước:

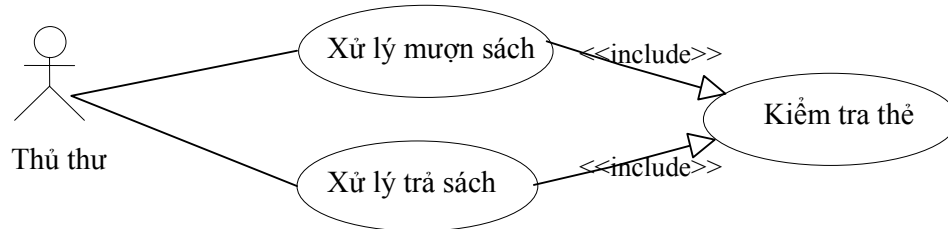
- Đưa thẻ vào máy
- Thực hiện đăng nhập
- Yêu cầu loại giao dịch
- Nhập loại giao dịch
- Thực hiện giao dịch
- Đẩy thẻ ra
- Yêu cầu lấy thẻ
- Lấy thẻ

Trong hệ thống quản lý thư viện, use case Mượn sách ngoài dòng hoạt động chính còn có các dòng phụ. Dòng phụ này sẽ được kích hoạt để giải quyết vấn đề khi một độc giả đến mượn tài liệu nhưng không có trong thư viện và thư viện sẽ mượn tài liệu đó từ những thư viện khác có liên kết. Hoặc do độc giả không thoả các điều kiện để được mượn (mượn sách quá hạn chưa

trả của lần mượn trước). Do đó, chúng ta tách dòng phụ này và use case “Mượn sách từ thư viện thành viên” và “” và tạo một liên kết <<extend>> từ use case này đến use case Xử lý mượn sách.

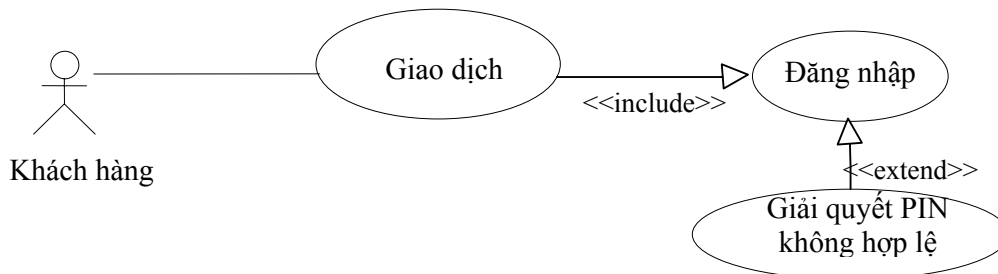


Liên kết sử dụng (<<include>>): được thành lập khi chúng ta có các use case mà tìm thấy một vài use case có những dòng hoạt động chung, và để tránh mô tả dòng hoạt động chung đó lặp lại trên những use case này, chúng ta có thể tách những dòng hoạt động chung đó ra thành một use case. Use case mới này có thể sử dụng bởi những use case khác. Quan hệ giữa những use case với use case được trích ra này gọi là quan hệ <<include>>. Quan hệ sử dụng giúp chúng ta tránh sự trùng lặp bằng cách cho phép một use case có thể được chia sẻ.



Trong ví dụ trên, use case *mượn sách* và *trả sách* đều phải thực hiện công việc kiểm tra thẻ thư viện của đọc giả, do đó chúng ta phát sinh một use case mới là *kiểm tra thẻ* bằng cách trích ra hoạt động kiểm tra thẻ thư viện từ hai use case trên và tạo một liên kết <<include>> tới use case từ hai use case đó tới use case mới. Các use case *Xử lý mượn sách* và *Xử lý trả sách* đều thừa hưởng tất cả hoạt động của use case của use case *kiểm tra thẻ*.

Trong hệ thống ATM, use case *Giao dịch* sẽ có mối liên kết <<include>> với use case *Đăng nhập*.



Đăng nhập: khách hàng nhập vào mã số PIN gồm bốn ký số. Nếu mã số PIN hợp lệ, tài khoản của khách hàng sẽ sẵn sàng cho các giao dịch. Các bước như sau:

- Yêu cầu password
- Nhập password
- Kiểm tra password

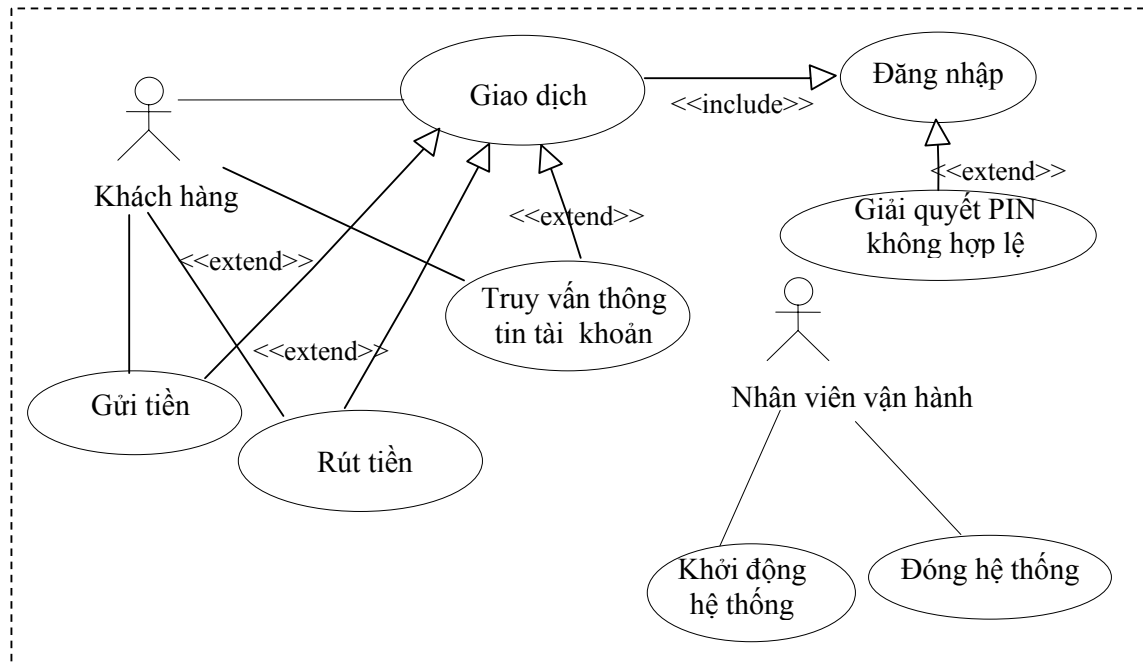
Giải quyết PIN không hợp lệ: nếu mã số PIN không hợp lệ, hệ thống sẽ hiển thị một thông báo tới khách hàng.

Sự giống nhau giữa liên kết <<extend>> và liên kết <<include>> là tất cả đều được xem như là một loại kế thừa. Khi chúng ta muốn chia sẻ một số hoạt động chung trong nhiều use case, dùng liên kết <<include>> bằng cách trích các hoạt động chia sẻ đó thành một use case mới. Khi chúng ta muốn thêm vào một ít khác biệt cho một use case để mô tả một tình huống đặc biệt trong một tình huống chung, chúng ta sẽ tạo một use case mới có liên kết <<extend>> với use case chung đó.

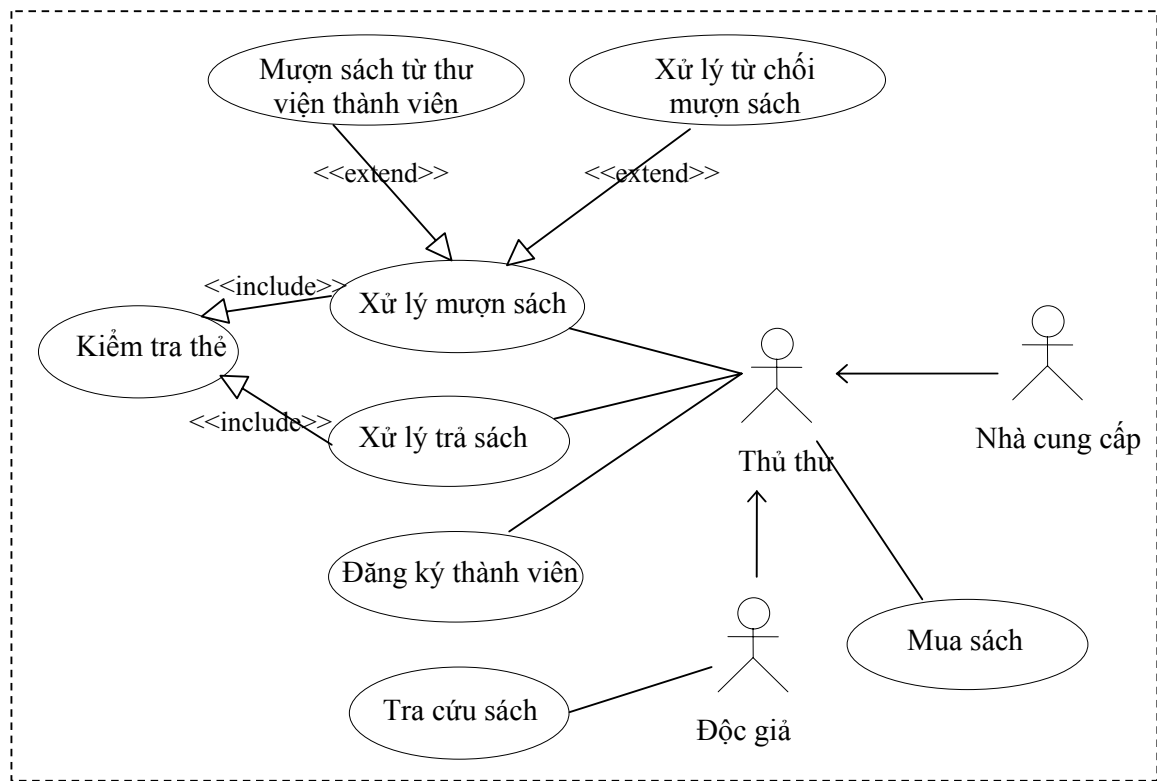
Dựa vào các liên kết được thiết lập cho các use case chúng ta phân use case thành hai loại:

Use case trừu tượng: là use case chưa hoàn hảo nghĩa là không tương tác với bất kỳ một tác nhân nào mà được sử dụng bởi một use case khác. Use case trừu tượng cũng có thể có liên kết <<extend>> hoặc liên kết <<include>> trong những mức độ khác. Ví dụ: các use case *Kiểm tra thẻ*, *Xử lý từ chối mượn sách*, ... là các use case trừu tượng.

Use case cụ thể: là use case có tương tác trực tiếp với một tác nhân. Ví dụ: các use case *Xử lý mượn sách*, *Xử lý trả sách*, hoặc *Khởi động máy*, *Đóng máy*,



Mô hình use case của hệ thống máy ATM

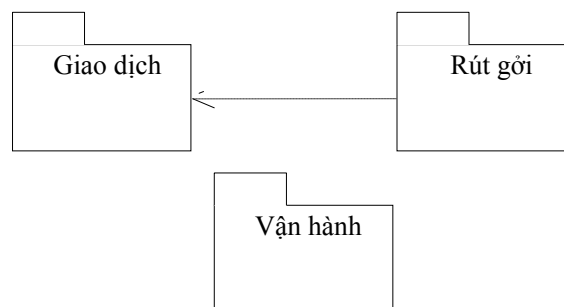


Mô hình use case hệ thống phần mềm quản lý thư viện

Phân chia các use case thành các gói (package)

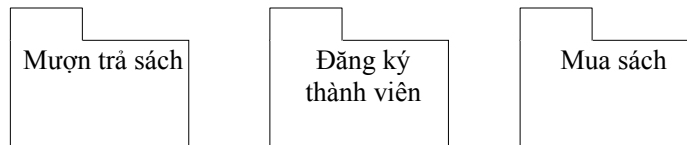
Mỗi use case minh họa một kịch bản trong hệ thống. Khi gặp những hệ thống tương đối phức tạp thì chúng ta nên thu hẹp tiêu điểm của các kịch bản trong hệ thống bằng cách phân chia thành các gói. Mỗi gói phản ánh một phạm vi của hệ thống mà chúng ta chỉ muốn quản lý nó khi chúng ta truy cập gói đó.

Ví dụ, có thể chia các se case của hệ thống máy ATM thành ba gói: Giao dịch, Rút gửi và Vận hành



Trong đó, gói Giao dịch gồm các use case: Giao dịch, Đăng nhập, Giải quyết PIN không hợp lệ; gói Rút gửi gồm các use case: Gởi tiền, Rút tiền, Truy vấn thông tin tài khoản; và gói Vận hành gồm các use case: Khởi động hệ thống, Đóng hệ thống

Hệ thống quản lý thư viện được chia thành ba gói như sau: Mượn trả sách, đăng ký thành viên, và Mua sách.



Trong đó, gói Mượn trả sách gồm các use case: Xử lý mượn sách, Xử lý trả sách, Kiểm tra thẻ, Mượn sách từ thư viện thành viên, Xử lý từ chối mượn sách; gói Đăng ký thành viên gồm use case: Đăng ký thành viên; gói mua sách gồm use case: Mua sách.

Câu hỏi và bài tập

Câu hỏi

1. Mô hình use case là gì?
2. Tại sao mô hình hoá use case là hữu dụng trong phân tích hệ thống?
3. Ai có thể là tác nhân?
4. Khi nào chúng ta sử dụng liên kết <<extend>> và <<include>> trong mô hình hoá use case?
5. Xác định tác nhân bằng cách nào?
6. Phân biệt sự khác nhau giữa người dùng và tác nhân?
7. Tại sao nên phân chia mô hình use case thành các gói?

Bài tập

1. Hãy xây dựng sơ đồ use case mô tả một hệ thống diễn đàn trao đổi học tập của khoa Công Nghệ Thông Tin. Hoạt động của diễn đàn có thể tóm mô tả như sau:
 - Tất cả mọi người khi truy cập vào địa chỉ của diễn đàn đều có thể xem được thông tin nội dung trao đổi của diễn đàn, tin tức trong diễn đàn.
 - Khi một thành viên muốn gửi thông tin lên diễn đàn. Thông tin đó có thể là một chủ đề cần trao đổi, hoặc một ý kiến liên quan đến một chủ đề đã được đưa ra, hoặc là một tin tức. Trước tiên, thành viên phải đăng nhập vào hệ thống với một tên và mã đăng nhập. Sau khi đăng nhập thành công, thành viên có thể sử dụng chức năng soạn thảo của hệ thống để soạn thảo và gửi thông tin lên diễn đàn. Thành viên có thể xem được các thông báo kết quả duyệt tin từ quản trị về những lần gửi trước.
 - Một người dùng khi truy cập thì có thể đăng ký thành viên của diễn đàn. Khi đăng ký, người dùng sẽ phải nhập các thông tin liên quan như: họ tên, tên đăng nhập, mã sinh viên, mã đăng nhập. Việc đăng ký có hiệu lực sau khi quản trị duyệt và chấp nhận.
 - Khi một thành viên đăng nhập vào quyền quản trị (quản trị hệ thống). Hệ thống sẽ hiển thị các thông tin được gửi bởi các thành viên. Ứng với mỗi thông tin, quản trị có thể chấp nhận hoặc từ chối đăng thông tin lên diễn đàn:
 - o Khi thông tin được đánh dấu là chấp nhận, hệ thống sẽ cập nhật lại trạng thái của thông tin và sẽ hiển thị thông tin này tới diễn đàn.
 - o Khi thông tin được đánh dấu là từ chối, hệ thống cũng sẽ cập nhật lại trạng thái thông tin và ghi chú về lý do từ chối thông tin.
 - Ngoài ra, người quản trị cũng có thể duyệt và chấp nhận hoặc từ chối một thành viên mới đăng ký. Huỷ một thành viên. Cập nhật thông tin của một thành viên.
2. Phân chia sơ đồ use case của hệ thống trên thành các gói. Mô tả sự liên kết giữa các gói (nếu có) và sơ đồ use case cho từng gói.