CÔNG NGHỆ PHẦN MỀM

CHƯƠNG 2

Xử lý yêu cầu

Nghệ An, 2021

📖 Nội dung giảng dạy

2.1. Khái niệm và tầm quan trọng

2.2. Phân loại các yêu cầu

2.3. Phân tích và xác định yêu cầu

2.4. Đặc tả yêu cầu

2.5. Định dạng tài liệu yêu cầu

2.6. Thẩm định yêu cầu

2.1. Khái niệm và tầm quan trọng

- Yêu cầu phần mềm:
  - Những gì muốn có trong phần mềm được phát triển.
- Tiêu chí gì quan trọng nhất đối với chất lượng phần mềm?
  - Phần mềm thỏa mãn được yêu cầu của người dùng
- Phân tích yêu cầu là sự phối hợp của nhà phát triển và khách hàng. Hiểu rõ, cụ thể hóa và biểu diễn yêu cầu.
- Quyết định chất lượng phần mềm đạt được với chi phí dự kiến và thời hạn cho trước.

📖 Yêu cầu và mục đích

- Yêu cầu
  - Mô tả từ trừu tượng đến chi tiết về các dịch vụ mà hệ thống cung cấp
  - Các ràng buộc lên sự phát triển và hoạt động của nó
- Mục đích
  - Làm cơ sở cho việc mời thầu
  - Làm cơ sở cho ký kết hợp đồng
  - Làm tài liệu đầu vào cho bước thiết kế và triển khai

Yêu cầu phần mềm

- Ví dụ: Các yêu cầu của Phần mềm Quản lý cửa hàng:
  - Nhập hàng mới
  - Lập hóa đơn bán hàng
  - Tra cứu hàng hóa
  - Lập báo cáo doanh thu trong tháng
  - Lập báo cáo tồn kho trong tháng
  - ...

🕮 Khó khăn của việc xác định yêu cầu

- Các yêu cầu mang tính đặc thù → khó hiểu, khó định nghĩa, không có chuẩn biểu diễn.
- Các hệ thống lớn, có nhiều người dùng → yêu cầu thường đa dạng, có các mức ưu tiên khác nhau, có khi mâu thuẫn nhau
- Khách hàng thường là nhà quản lý, ko trực tiếp sử dụng → diễn đạt yêu cầu ko rõ ràng

Ví dụ Travel Agency: Yêu cầu người dùng

- Hãng du lịch TravelGood đến gặp người làm phần mềm và đề nghị làm dự án phần mềm sau:
  - Mô tả bài toán / yêu cầu người dùng:
    - TravelGood muốn cung cấp cho khách hàng của họ một ứng dụng đặt vé và lập kế hoạch du lịch.
    - Ứng dụng này cần cho phép khách lập kế hoạch về các chuyến bay và khách sạn.
    - Đầu tiên, khách hàng có thể sắp xếp một chuyến đi, sau đó đặt vé và đặt phòng khách sạn cho chuyến đi đó. Người dùng có thể lập kế hoạch cho nhiều chuyến đi. Ngoài ra, phần mềm còn cho phép hủy các chuyến đã đặt.

Ví dụ Travel Agency: Yêu cầu người dùng

- Sau khi nhận làm phần mềm cho TravelGood, đội phát triển chi tiết hóa thành các yêu cầu hệ thống:
  - Người dùng có thể lập kế hoạch một chuyến đi bằng cách chọn một trình tự các điểm đến, rồi lưu lại. (kèm theo sơ đồ mô tả kịch bản ca sử dụng)
  - Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt
  - Ứng dụng Web phải triển khai được tại các Server tiêu chuẩn
  - Hệ thống phải dễ sử dụng: đạt một test usability (kèm chi tiết cụ thể)
  - …

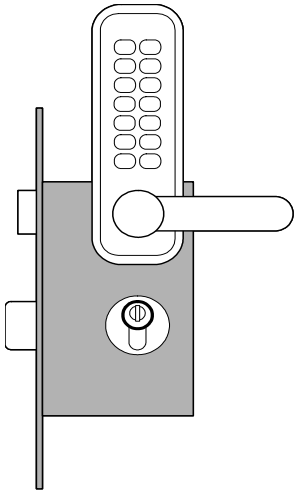
2.2. Phân loại các yêu cầu

- Yêu cầu người dùng (User requirements)
  - Các phát biểu bằng ngôn ngữ tự nhiên cộng với các sơ đồ về các dịch vụ mà hệ thống cung cấp và các ràng buộc về vận hành.
  - Được viết cho khách hàng.
- Yêu cầu hệ thống (System requirements)
  - Một tài liệu có cấu trúc bao gồm các mô tả chi tiết về các chức năng và dịch vụ của hệ thống cùng với các ràng buộc về vận hành.
  - Định nghĩa cái gì cần được cài đặt
    - Có thể là một phần của một hợp đồng giữa khách hàng và người nhận thầu.

Yêu cầu người dùng / Yêu cầu hệ thống

- Ví dụ yêu cầu hệ thống:

Identifier	Priority	Requirement
REQ5	2	The system shall maintain a history log of all attempted accesses for later review.
REQ6	2	The system should allow adding new authorized persons at runtime or removing existing ones.
REQ7	2	The system shall allow configuring the preferences for device activation when the user provides a valid key code, as well as when a burglary attempt is detected.
REQ8	1	The system should allow searching the history log by specifying one or more of these parameters: the time frame, the actor role, the door location, or the event type (unlock, lock, power failure, etc.). This function shall be available over the Web by pointing a browser to a specified URL.
REQ9	1	The system should allow filing inquiries about “suspicious” accesses. This function shall be available over the Web.



📖 Yêu cầu người dùng / Yêu cầu hệ thống

- Ví dụ yêu cầu hệ thống:

Identifier	Priority	Requirement
REQ1	5	The system shall keep the door locked at all times, unless commanded otherwise by authorized user. When the lock is disarmed, a countdown shall be initiated at the end of which the lock shall be automatically armed (if still disarmed).
REQ2	2	The system shall lock the door when commanded by pressing a dedicated button.
REQ3	5	The system shall, given a valid key code, unlock the door and activate other devices.
REQ4	4	The system should allow mistakes while entering the key code. However, to resist “dictionary attacks,” the number of allowed failed attempts shall be small, say three, after which the system will block and the alarm bell shall be sounded.

Yêu cầu người dùng / Yêu cầu hệ thống

- Ví dụ yêu cầu người dùng:

As a tenant, I can unlock the doors to enter my apartment.

user-role (benefactor) | capability | business-value

  - Tương tự với yêu cầu hệ thống, nhưng tập trung vào những gì người dùng nhận được từ hệ thống, thay vì các tính năng hệ thống.
  - Được sử dụng phổ biến trong các phương pháp Agile.

📖 Yêu cầu người dùng / Yêu cầu hệ thống

- Ví dụ yêu cầu người dùng:

Identifier	User Story	Size
ST-1	As an authorized person (tenant or landlord), I can keep the doors locked at all times.	**4 points**
ST-2	As an authorized person (tenant or landlord), I can lock the doors on demand.	**3 pts**
ST-3	The lock should be automatically locked after a defined period of time.	**6 pts**
ST-4	As an authorized person (tenant or landlord), I can unlock the doors. (Test: Allow a small number of mistakes, say three.)	**9 points**
ST-5	As a landlord, I can at runtime manage authorized persons.	**10 pts**
ST-6	As an authorized person (tenant or landlord), I can view past accesses.	**6 pts**
ST-7	As a tenant, I can configure the preferences for activation of various devices.	**6 pts**
ST-8	As a tenant, I can file complaint about “suspicious” accesses.	**6 pts**

Yêu cầu chức năng

- Yêu cầu chức năng: mô tả chức năng, dịch vụ mà hệ thống cần phải cung cấp
  - Phụ thuộc: loại phần mềm, yêu cầu của khách hàng, loại hệ thống mà phần mềm trợ giúp
  - Mức độ yêu cầu:
    - Trừu tượng (hệ thống làm gì)
    - Chi tiết (từng chức năng của hệ thống làm gì)
      - Hệ thống cần phản ứng như thế nào với các input cụ thể
      - Hệ thống cần ứng xử như thế nào trong các tình huống cụ thể

Yêu cầu chức năng

- Một số yêu cầu chức năng:
  - Chứcnăng tính toán
  - Chứcnăng lưu trữ
  - Chứcnăng tìm kiếm
  - Chứcnăng kết xuất
  - Chứcnăng backup, restore
  - Chứcnăng đa người dùng
  - Chứcnăng đa phương tiện
  - …
- Ví dụ:
  - Người dùng có thể tìm kiếm, download, in những bài báo
  - Người dùng được cấp một vùng lưu trữ riêng để có thể copy để lưu trữ tài liệu lâu dài

Yêu cầu phi chức năng

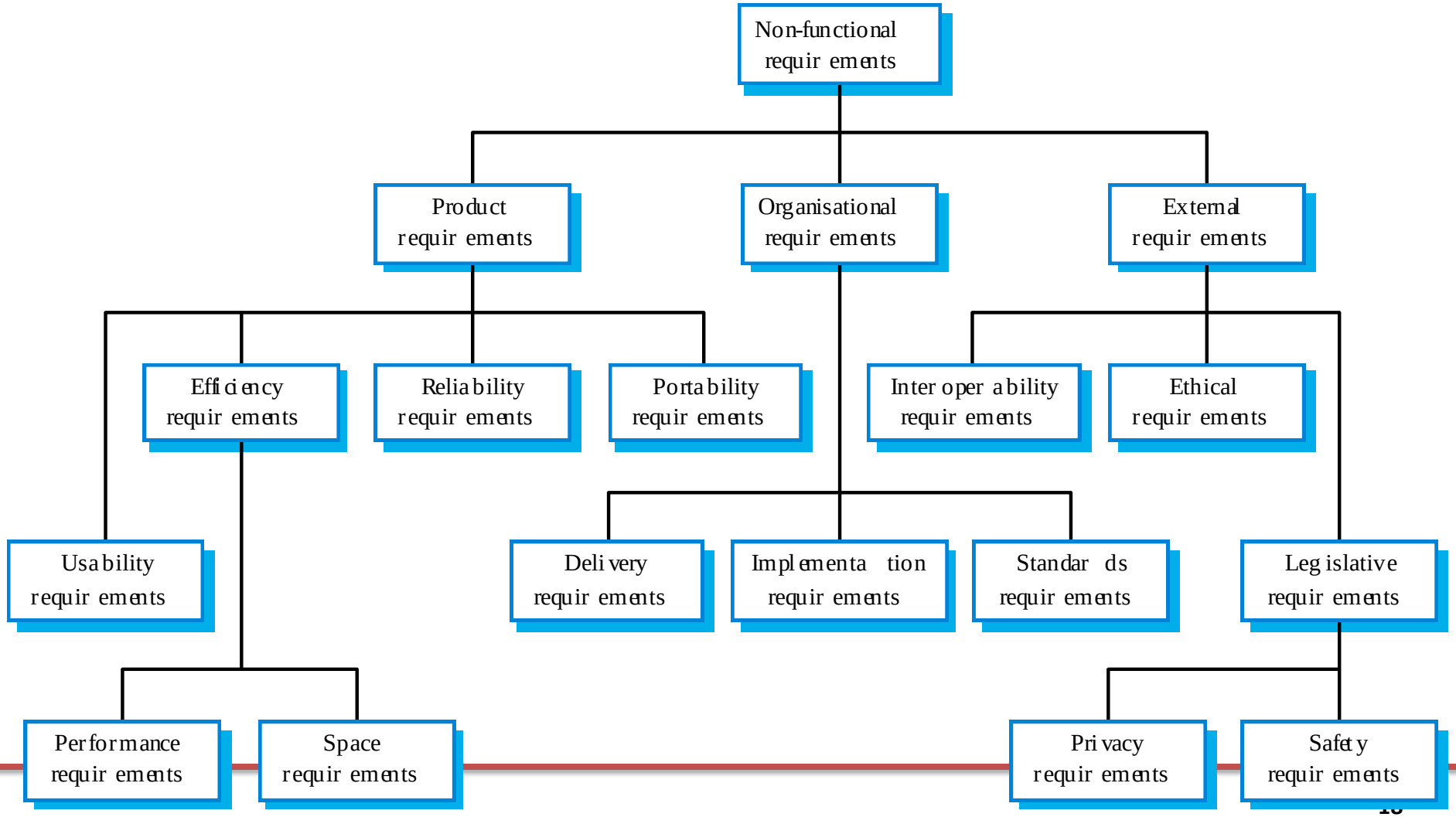
- Yêu cầu phi chức năng
  - Mô tả các ràng buộc lên dịch vụ và quá trình phát triển hệ thống (về chất lượng, môi trường, chuẩn sử dụng, quy trình phát triển...)
  - Yêu cầu về sản phẩm: tốc độ, độ tin cậy, bộ nhớ, giao diện.
  - Yêu cầu tiến trình phát triển: chuẩn áp dụng, phương pháp thiết kế, ngôn ngữ lập trình, mô hình tiến trình...
  - Yêu cầu từ bên ngoài: chi phí, thời gian, bản quyền, liên kết...

Yêu cầu chức năng / phi chức năng

- Yêu cầu chức năng (functional requirement):
  - Người dùng có thể lập kế hoạch một chuyến đi, đặt vé, đặt phòng, lưu một kế hoạch để sau này sẽ đặt vé đặt phòng…
- Yêu cầu phi chức năng (non-functional requirement):
  - Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt → không rõ ràng
  - Ứng dụng Web phải triển khai được tại các server tiêu chuẩn
  - Hệ thống phải dễ sử dụng – phải đạt một test usability

Yêu cầu chức năng / phi chức năng

- Các loại yêu cầu phi chức năng:



Yêu cầu được diễn đạt tốt

- Kiểm thử được (testability)
  - Test được (thủ công hoặc tự động)
- Đo được
  - Ví dụ về yêu cầu không đo được:
    - Hệ thống cần dễ sử dụng bởi các nhân viên và cần được tổ chức sao cho người dùng ít làm nhầm nhất
  - Đo được:
    - Nhân viên cần sử dụng được toàn bộ các chức năng của hệ thống sau 04 tiếng huấn luyện. Sau huấn luyện, số lỗi trung bình mà một người dùng có kinh nghiệm phạm phải trong mỗi giờ không vượt quá 02 lỗi

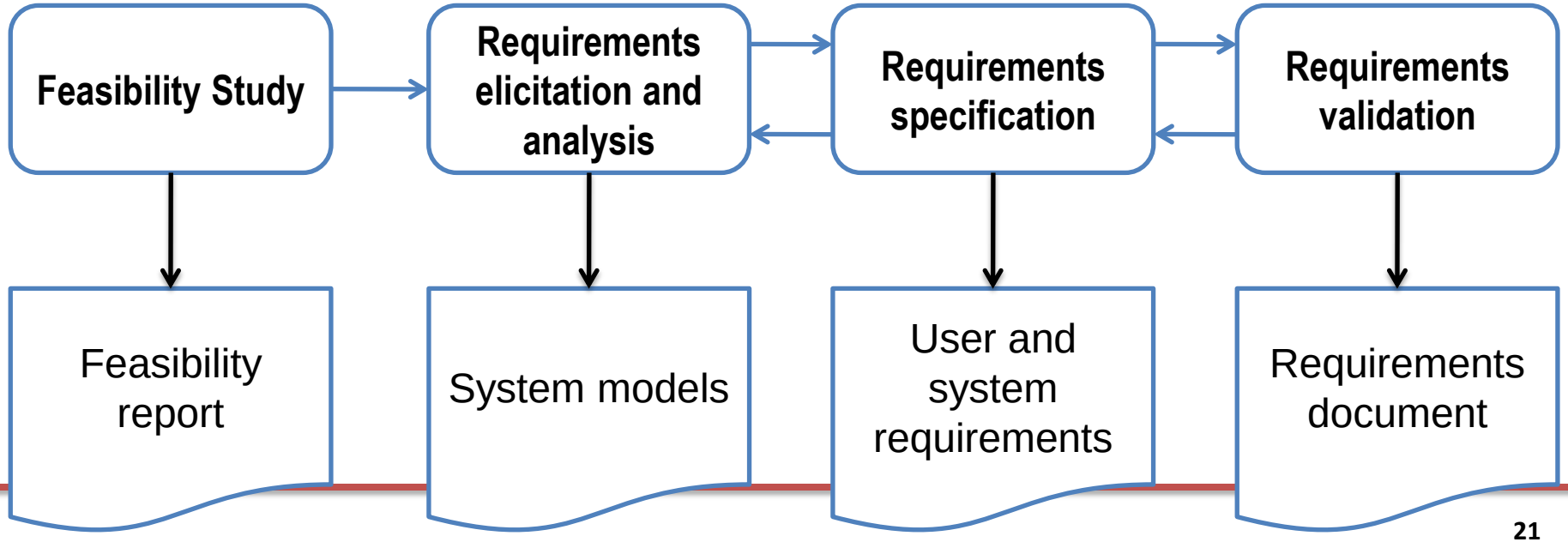
Các độ đo

- Các độ đo có thể sử dụng

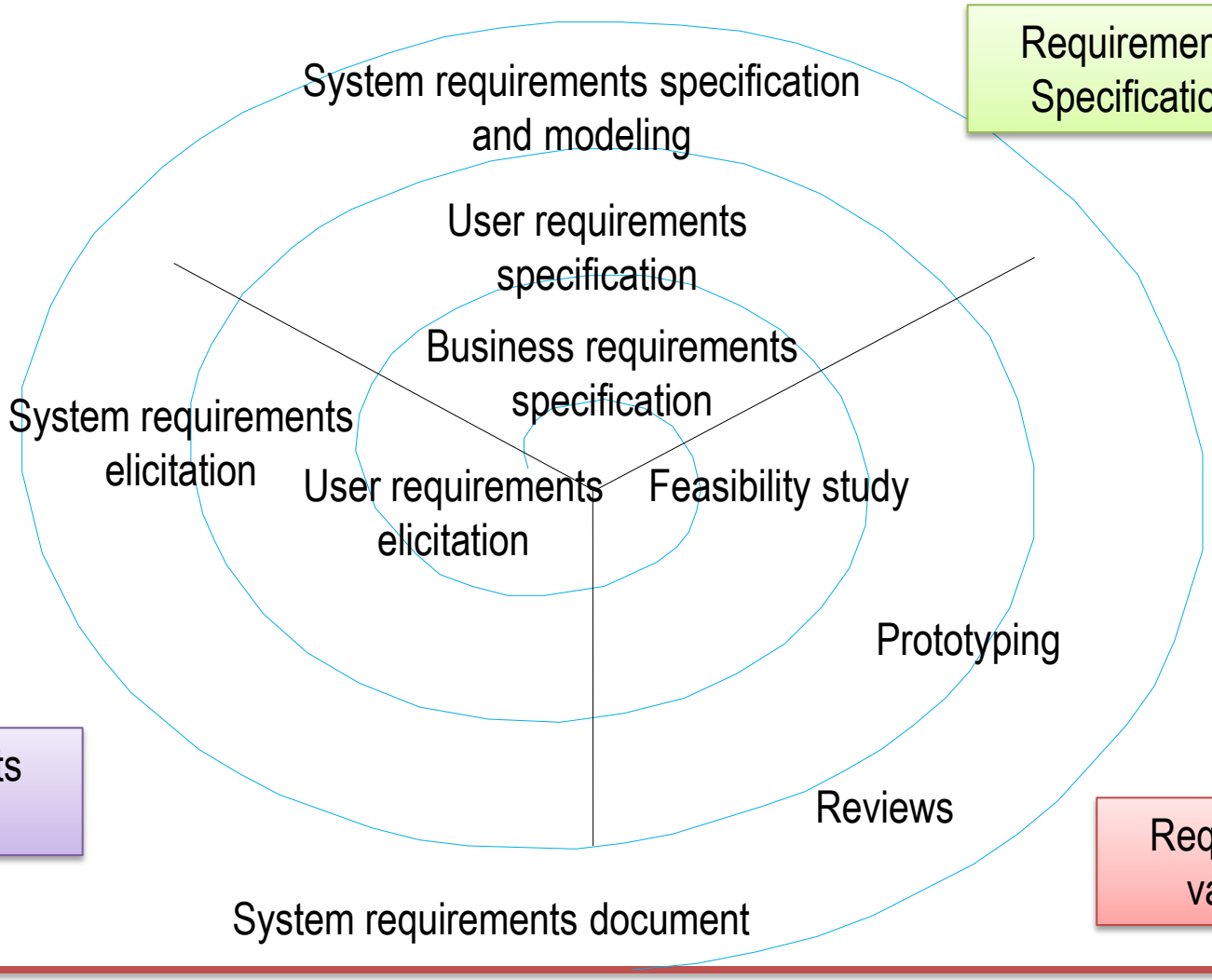
Đặc điểm	Độ đo
Tốc độ	Số giao dịch được xử lý mỗi giây Thời gian đáp ứng mỗi sự kiện Tần suất làm tươi màn hình
Kích thước	M Bytes
Dễ sử dụng	Thời gian huấn luyện Số trang tài liệu hướng dẫn sử dụng
Độ tin cậy (Reliability)	Khoảng thời gian trung bình giữa các sự cố Xác suất hệ thống không hoạt động tại một thời điểm Số lần xảy ra sự cố trong mỗi giờ
Vững mạnh (Robustness)	Thời gian cần để hoạt động lại sau sự cố Phần trăm sự kiện gây sự cố Xác xuất hỏng dữ liệu do sự cố

2.3. Phân tích và xác định yêu cầu

- Quy trình kỹ thuật yêu cầu:
  - Nghiên cứu khả thi
  - Thu thập và phân tích yêu cầu
  - Đặc tả yêu cầu
  - Thẩm định yêu cầu



Kĩ thuật yêu cầu



Nghiên cứu khả thi

- Mục tiêu: có nên phát triển hệ thống không
- Nội dung nghiên cứu khả thi:
  - Về kinh tế: tài chính, lợi ích mang lại của hệ thống, chi phí bảo trì.
  - Về kỹ thuật: xem xét khả năng kỹ thuật hiện tại đủ đáp ứng không? Những rủi ro sẽ gặp phải, tài nguyên phát triển (con người, phần cứng, phần mềm hỗ trợ), công nghệ hiện tại, khả năng tích hợp với hệ thống khác
  - Về pháp lý, tính khả thi về vận hành hệ thống

Nghiên cứu khả thi

- Thực hiện nghiên cứu khả thi:
  - Báo cáo khả thi được viết dựa trên thông tin thu thập được, những đánh giá về hệ thống hiện tại và phác họa phương án dự kiến
  - Các câu hỏi dành cho nhân viên của tổ chức:
    - Nếu hệ thống không được cài đặt thì sao?
    - Quy trình hiện hành có những vấn đề gì?
    - Hệ thống được đề xuất sẽ giúp được gì và như thế nào?
    - Khi tích hợp sẽ gặp những rắc rối nào?
    - Có cần công nghệ mới hay không? Cần kỹ năng gì?
    - Hệ thống mới cần hỗ trợ những tiện ích nào?

Thu thập và phân tích yêu cầu

- Các hoạt động quy trình
  - Phát hiện yêu cầu
    - Quy trình thu thập thông tin về hệ thống đề xuất và các hệ thống sẵn có, lọc ra các yêu cầu người dùng và yêu cầu hệ thống từ các thông tin này.
  - Phân loại và tổ chức
    - Phân nhóm các yêu cầu có liên quan đến nhau và tổ chức chúng thành các cụm có quan hệ gắn kết với nhau.
  - Đặt thứ tự ưu tiên và giải quyết mâu thuẫn giữa các yêu cầu
    - Xếp thứ tự ưu tiên cho các yêu cầu và giải quyết các xung đột/mâu thuẫn giữa các yêu cầu.
  - Documentation – Viết tài liệu
    - Ghi lại các yêu cầu làm tài liệu đầu vào cho vòng xoắn tiếp theo.

Thu thập và phân tích yêu cầu

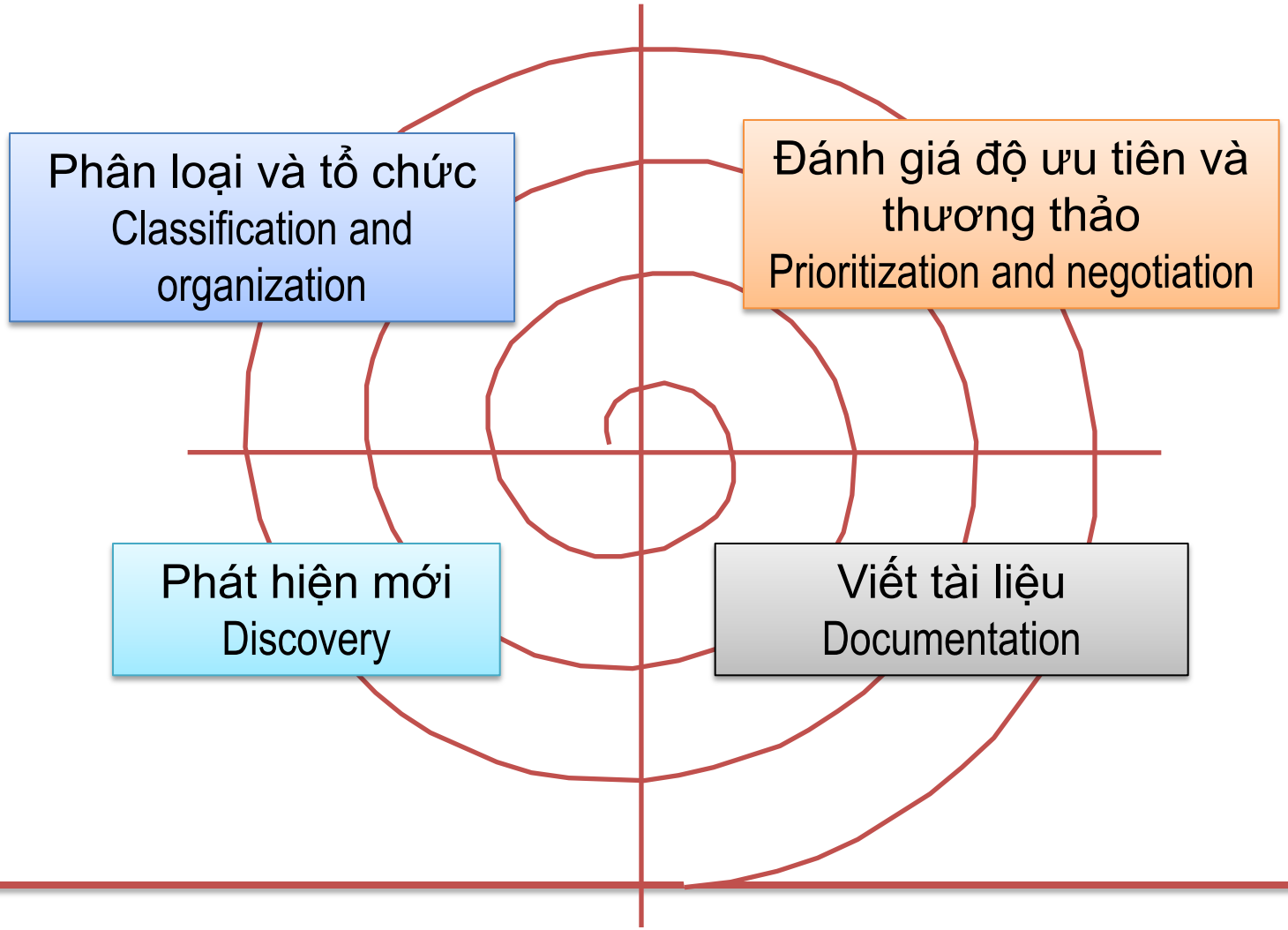
- Thu thập yêu cầu từ đâu?
  - Làm việc với khách hàng để tìm hiểu thông tin về:
    - Miền ứng dụng
    - Các dịch vụ mà hệ thống cần cung cấp
    - Các ràng buộc về vận hành hệ thống
  - Những người có thể cần tham gia (Stakeholders): khách hàng, người sử dụng, lập trình viên, chuyên gia kĩ thuật...
  - Tài liệu về hoạt động doanh nghiệp
  - Đặc tả của các hệ thống tương tự

Thu thập và phân tích yêu cầu

- Các hiện trạng cần khảo sát:
  - Hiện trạng nghiệp vụ: Lập danh sách các nghiệp vụ của từng bộ phận.
    - Mô tả nghiệp vụ: Tên công việc; Biểu mẫu liên quan; Qui định liên quan; Người thực hiện; Tần suất thực hiện; Thời điểm thực hiện; Cách thức thực hiện
  - Hiện trạng tổ chức
    - Sơ đồ tổ chức của các phòng ban, bộ phận (chú trọng các bộ phận có giao tiếp đối tác bên ngoài)
  - Hiện trạng tin học
    - Hệ thống phần cứng
    - Hệ thống phần mềm
    - Con người

Thu thập và phân tích yêu cầu

- Vòng xoắn ốc yêu cầu



Thu thập và phân tích yêu cầu

- Ví dụ: ATM stakeholder
  - Khách hàng của ngân hàng (người sử dụng dịch vụ)
  - Đại diện của các ngân hàng khác (ATM của ngân hàng này có thể dùng để giao dịch với ngân hàng khác)
  - Quản lý ngân hàng (dùng thông tin quản lý từ hệ thống ATM)
  - Nhân viên lại các chi nhánh ngân hàng (vận hành hệ thống)
  - Quản trị cơ sở dữ liệu (tích hợp hệ thống với CSDL của ngân hàng)
  - Quản lý an ninh
  - Phòng marketing (muốn dùng ATM để quảng cáo)
  - Kĩ sư bảo trì phần mềm và phần cứng
  - Những người điều phối hệ thống ngân hàng quốc gia (đảm bảo hệ thống tuân theo nguyên tắc chung)

Thu thập và phân tích yêu cầu

- Các kỹ thuật
  - Lấy yêu cầu
    - Phỏng vấn (Interviews), điều tra bằng bảng câu hỏi (Questionnaires)
    - Danh mục khái niệm (glossary) để hiểu miền ứng dụng
    - Ca sử dụng / user story
    - Quan sát
    - Nghiên cứu tài liệu
    - Phân tích thiết kế nhóm (Joint Application Design – JAD)
    - Làm bản mẫu
  - Đặc tả yêu cầu
    - Danh mục khái niệm
    - Use case / user story

Thu thập và phân tích yêu cầu

- Những khó khăn của phân tích:
  - Khách hàng thường mơ hồ về yêu cầu, không biết rõ mình muốn gì
  - Họ thể hiện yêu cầu theo thuật ngữ riêng
  - Khách hàng đa dạng → có thể có yêu cầu mâu thuẫn nhau
  - Những yếu tố tổ chức và chính sách có thể ảnh hưởng đến yêu cầu
  - Yêu cầu thường mang tính đặc thù, khó hiểu và không có chuẩn chung
  - Yêu cầu thường thay đổi trong quá trình phân tích: môi trường nghiệp vụ thay đổi, nhân sự mới đến...

2.4. Đặc tả yêu cầu

- Đặc tả phi hình thức
  - Bằng ngôn ngữ tự nhiên
  - Bằng bảng, sơ đồ, hình vẽ tùy chọn
- Đặc tả bán hình thức (đặc tả dựa trên mô hình)
  - Ký pháp đồ họa
  - Quy tắc biểu diễn
- Đặc tả hình thức
  - Biểu diễn bởi mô hình toán học
    - Ký pháp toán học
    - Quy tắc khai báo, biểu diễn biểu thức
    - Luật biến đổi biểu thức
  - Có thể tự động hóa mô hình
  - Sử dụng cho kiểm chứng, kiểm thử dựa trên mô hình

📖 Kịch bản

- Kịch bản (scenario) là các ví dụ đời thực về việc hệ thống có thể được sử dụng như thế nào.
- Các kịch bản nên bao gồm
  - Một miêu tả về tình huống ban đầu
  - Một miêu tả về luồng sự kiện thông thường
  - Một miêu tả về những trục trặc gì có thể xảy ra
  - Thông tin về các hoạt động xảy ra đồng thời
  - Một miêu tả về trạng thái khi kịch bản kết thúc

Kịch bản LIBSYS

- Initial Assumption: Người dùng đã đăng nhập hệ thống LIBSYS và đã tìm thấy tạp chí có đăng tài liệu cần tìm.
- Normal:
  - Người dùng chọn tài liệu cần copy. Hệ thống sẽ yêu cầu người dùng nhập thông tin thuê bao hoặc chọn cách trả phí dùng tài liệu. Có thể thanh toán bằng thẻ tín dụng hoặc dùng số tài khoản của một tổ chức.
  - Sau đó người dùng được yêu cầu điền một form bản quyền trong đó có chi tiết về giao dịch này, rồi submit form đó cho hệ thống LIBSYS.
  - Hệ thống kiểm tra form bản quyền, nếu OK, bản PDF của tài liệu sẽ được tải xuống máy tính của người dùng và người dùng được thông báo về việc này. Sau đó người dùng được chọn một máy in, và tài liệu sẽ được in tại đó.
  - Nếu tài liệu đã được gắn cờ ‘print-only’ thì nó sẽ được xóa khỏi máy của người dùng ngay sau khi người dùng khẳng định rằng đã in xong.

Kịch bản LIBSYS

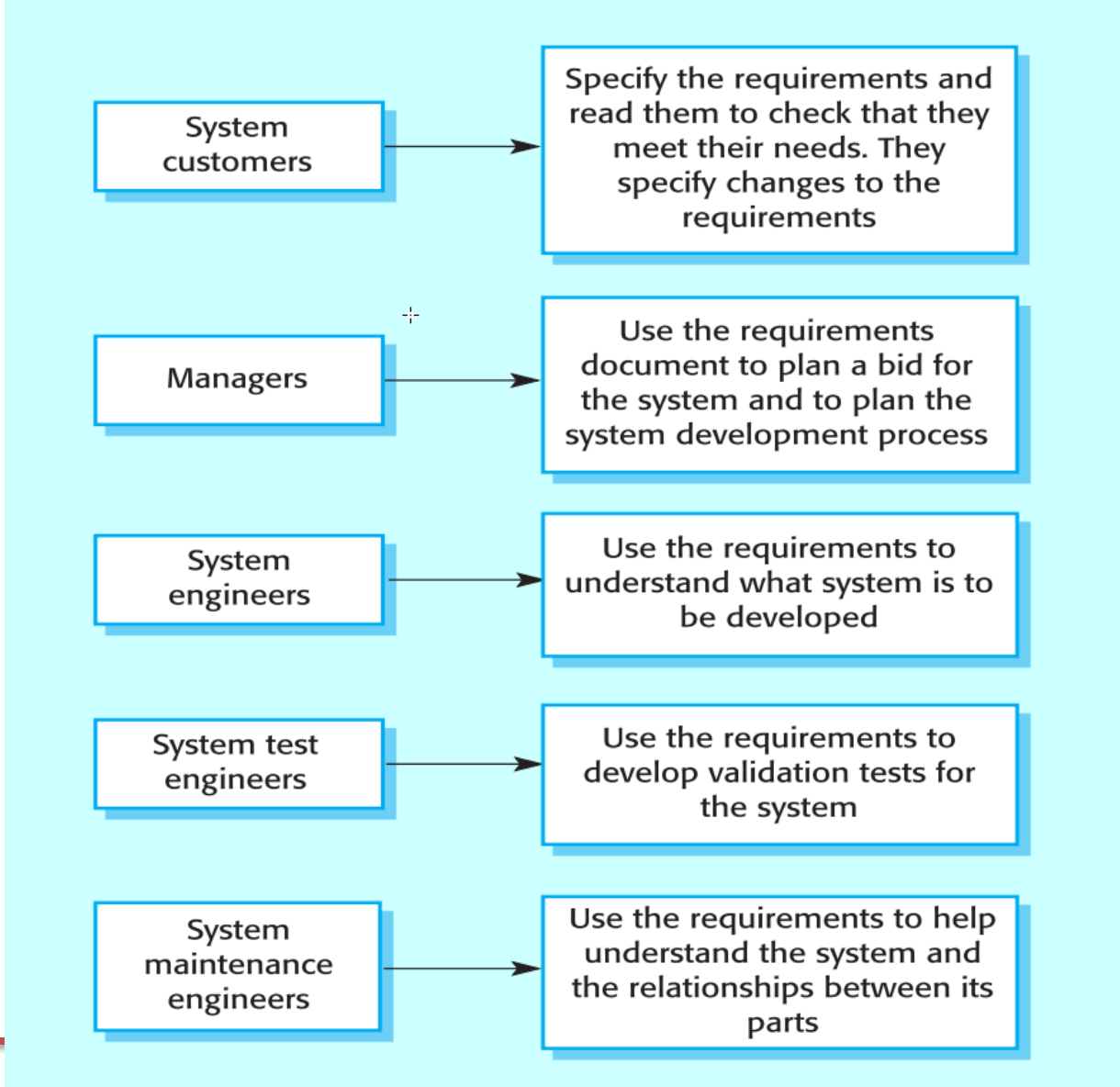
- What can go wrong:
  - Người dùng có thể điền form sai. Khi đó hệ thống cần hiện lại form để người dùng sửa lại. Nếu form được submit sau đó vẫn sai thì hủy yêu cầu đọc tài liệu của người dùng.
  - Hệ thống có thể không chấp nhận giao dịch thanh toán tiền. Hủy yêu cầu đọc tài liệu của người dùng.
  - Việc download tài liệu có thể thất bại. Làm lại cho đến khi thành công hoặc khi người dùng chấm dứt phiên làm việc.
  - Có thể không in được tài liệu. Nếu bài báo không có gắn cờ ‘print-only’ thì giữ nó trong workspace của LIBSYS. Nếu không, xóa tài liệu và hoàn lại chi phí cho người dùng.
- Other activities: Song song download các tài liệu khác nhau.
- System state on completion: Người dùng đang ở trạng thái đăng nhập. Nếu tài liệu có gắn cờ 'print-only' thì nó đã bị xóa khỏi LIBSYS workspace.

2.5. Định dạng tài liệu yêu cầu

- Tài liệu đặc tả theo IEEE:

1. Giới thiệu
    - 1.1. Mục đích của tài liệu yêu cầu
    - 1.2. Phạm vi của sản phẩm
    - 1.3. Các định nghĩa, từ viết tắt
    - 1.4. Các tham chiếu
    - 1.5. Tổng quan về tài liệu yêu cầu
2. Mô tả chung
    - 2.1. Giới thiệu chung về sản phẩm
    - 2.2. Các chức năng của sản phẩm
    - 2.3. Đặc điểm của người sử dụng
    - 2.4. Các ràng buộc
    - 2.5. Giả thiết và các phụ thuộc
3. Đặc tả yêu cầu: bao gồm các yêu cầu chức năng, phi chức năng, miền ứng dụng và giao diện.
4. Phụ lục
5. Chỉ mục

Người dùng với tài liệu yêu cầu

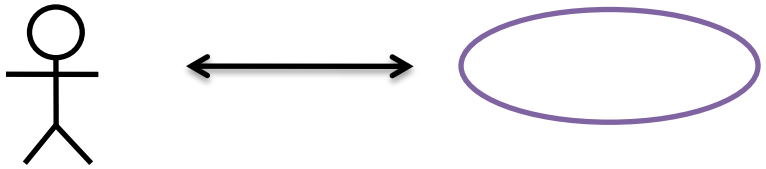


Use case

- Ca sử dụng (Use case) là một kĩ thuật kiểu kịch bản bằng ngôn ngữ UML
  - Mô tả các chức năng của hệ thống dựa trên quan điểm người sử dụng.
  - Mô tả sự tương tác giữa người dùng và hệ thống.
  - Cho biết hệ thống được sử dụng như thế nào?
- Là một tập các kịch bản tương tác giữa một hoặc vài actor với hệ thống nhằm thực hiện một mục tiêu chung
- Một bộ ca sử dụng có thể mô tả được tất cả các tương tác có thể đối với hệ thống.

Use case

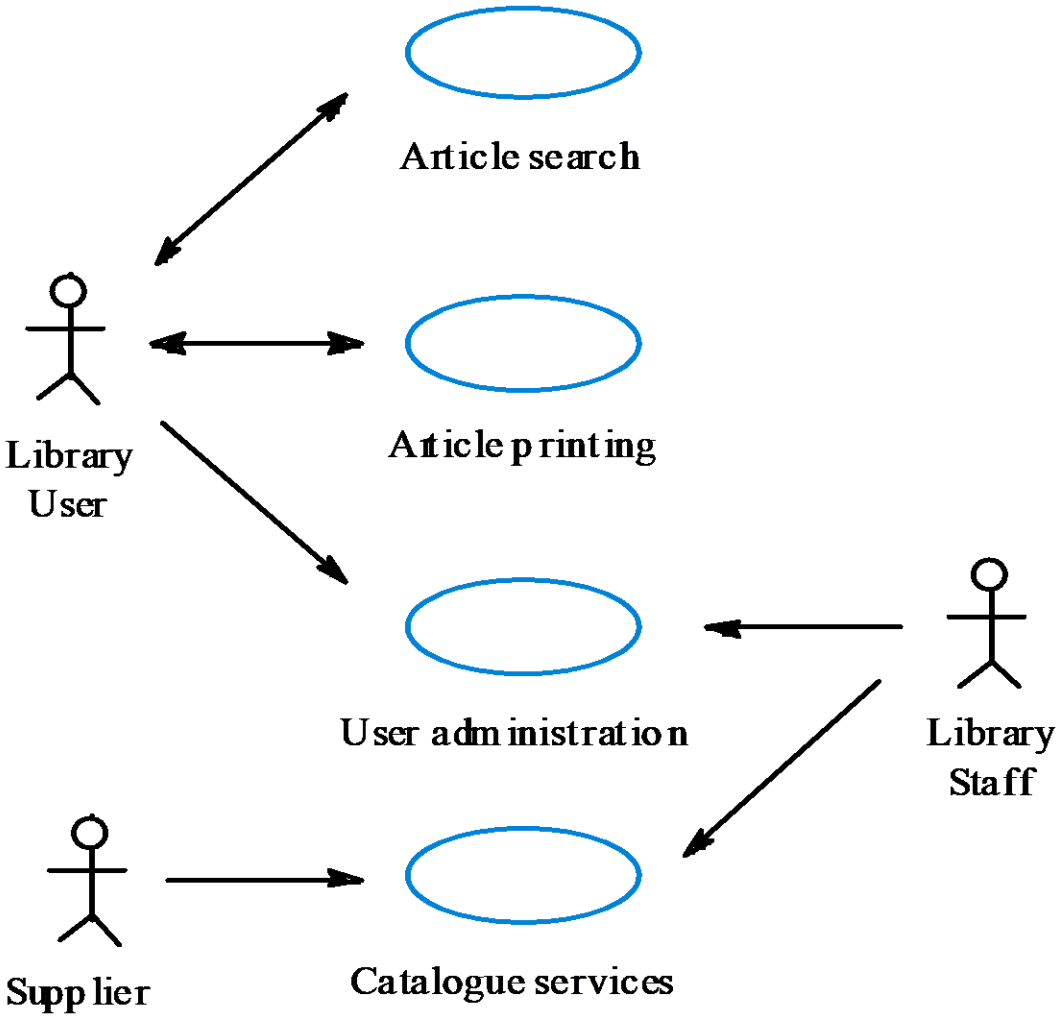
- Sơ đồ use case (đồ họa)
  - Sơ đồ mô tả tổng quan các ca sử dụng của một hệ thống và ai dùng chức năng nào
- Mô tả chi tiết use case (văn bản)
  - Mô tả chi tiết tương tác giữa người dùng và hệ thống trong một tập các kịch bản
- Có thể dùng các sơ đồ tuần tự (sequence diagram) để bổ sung chi tiết cho các ca sử dụng
  - Minh họa chuỗi xử lý sự kiện



In tài liệu

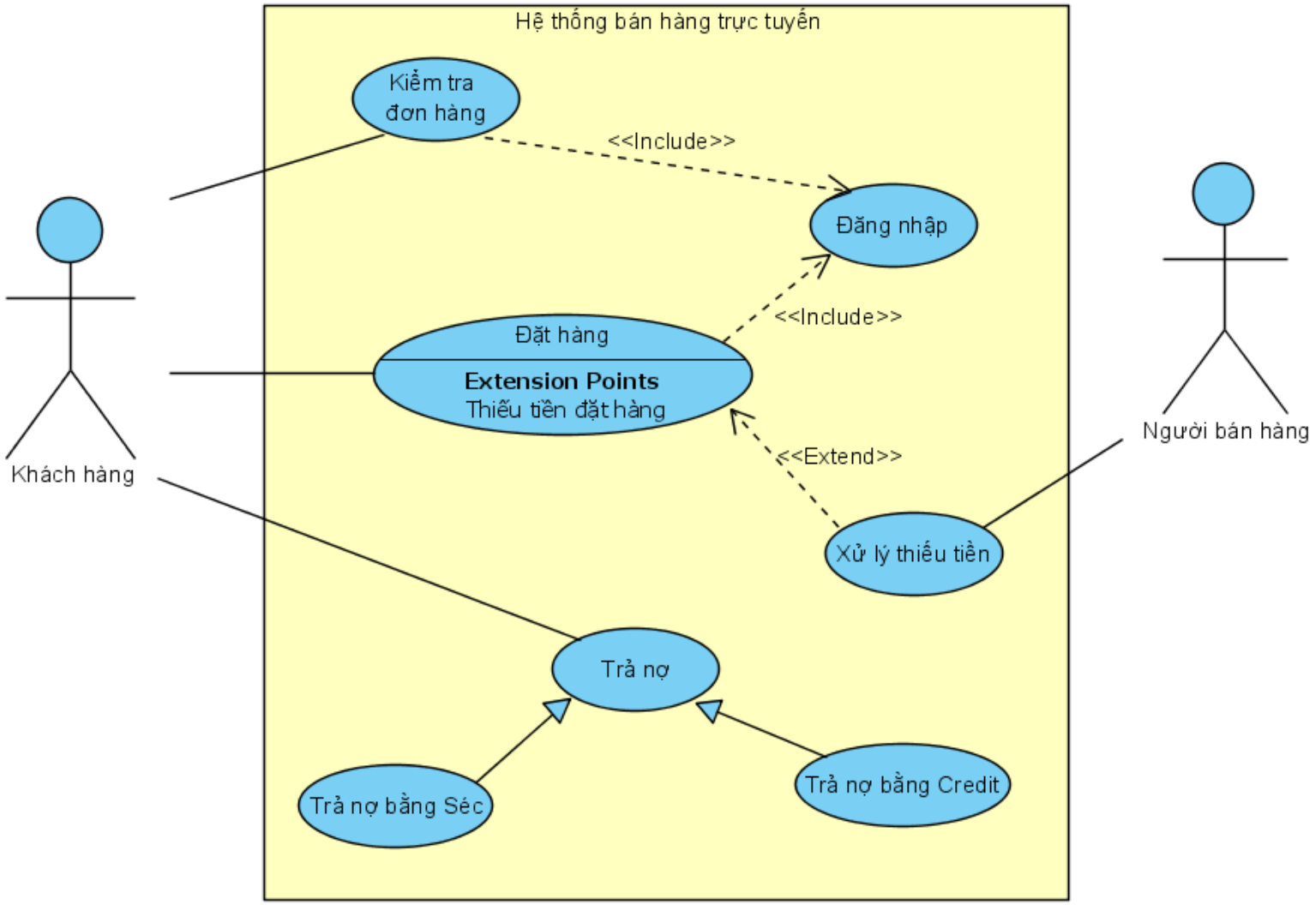
Use case

- LIBSYS use case:



Use case

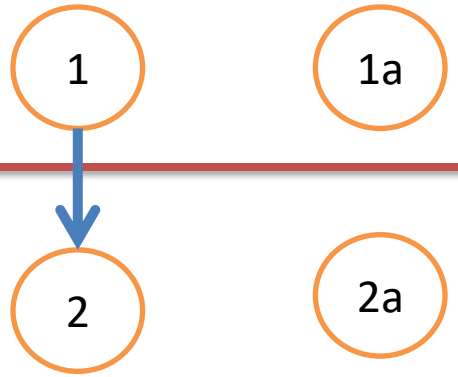
- Use case bán hàng trực tuyến:



Use case

- Một số quy tắc:
  - Use case A được gọi là Include B nếu trong xử lý của A có gọi đến B ít nhất 1 lần.
  - Use case B được gọi là Exntend A nếu use case B được gọi bởi A nếu thỏa mãn điều kiện nào đó.
  - Use case A được gọi là Generalization B nếu B là một trường hợp riêng của A.

Use case



- Travel Agency use case: List available flights

Tên: List available flights

Mô tả: Người dùng xem danh sách các chuyến bay có thể đặt

Actor: Người dùng

Kịch bản chính:

1. Người dùng nhập thông tin về thành phố cần đến, ngày đi và ngày đến
2. Hệ thống cung cấp một danh sách các chuyến bay phù hợp kèm theo giá vé và booking number

Kịch bản phụ:

1a. Dữ liệu vào không đúng

2. Hệ thống báo lỗi và kết thúc, use case quay lại từ đầu

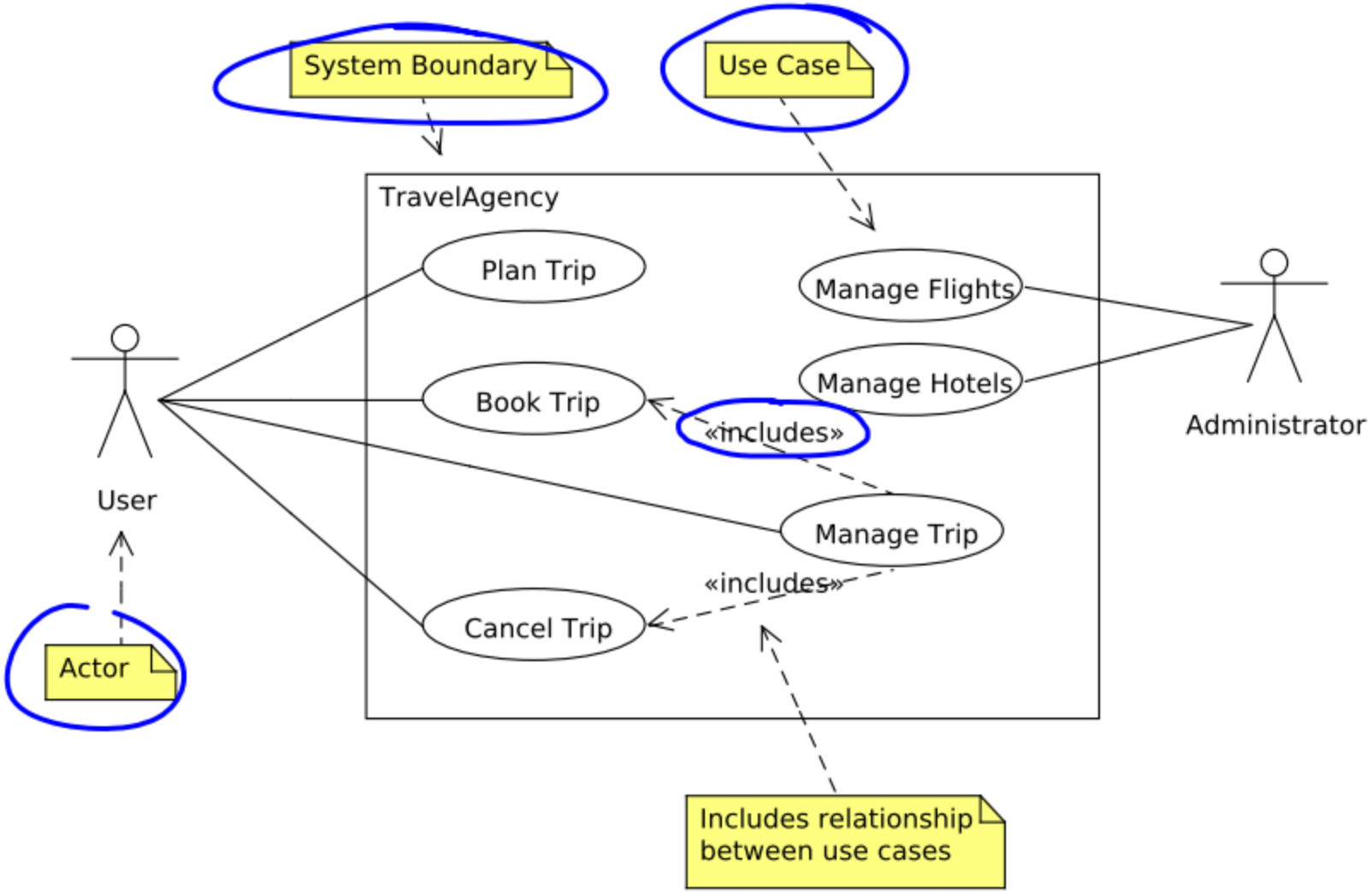
2.a. Không có chuyến bay nào phù hợp

3. Use case quay lại từ đầu

Ghi chú: Dữ liệu vào là đúng nếu tên thành phố đúng, ngày đi và ngày đến là các ngày hợp lệ, ngày đi sớm hơn ngày đến, ngày đến muộn hơn thời điểm hiện tại ít nhất 2 ngày, và ngày đi không muộn hơn một năm kể từ hiện tại

Use case

- Sơ đồ Use case

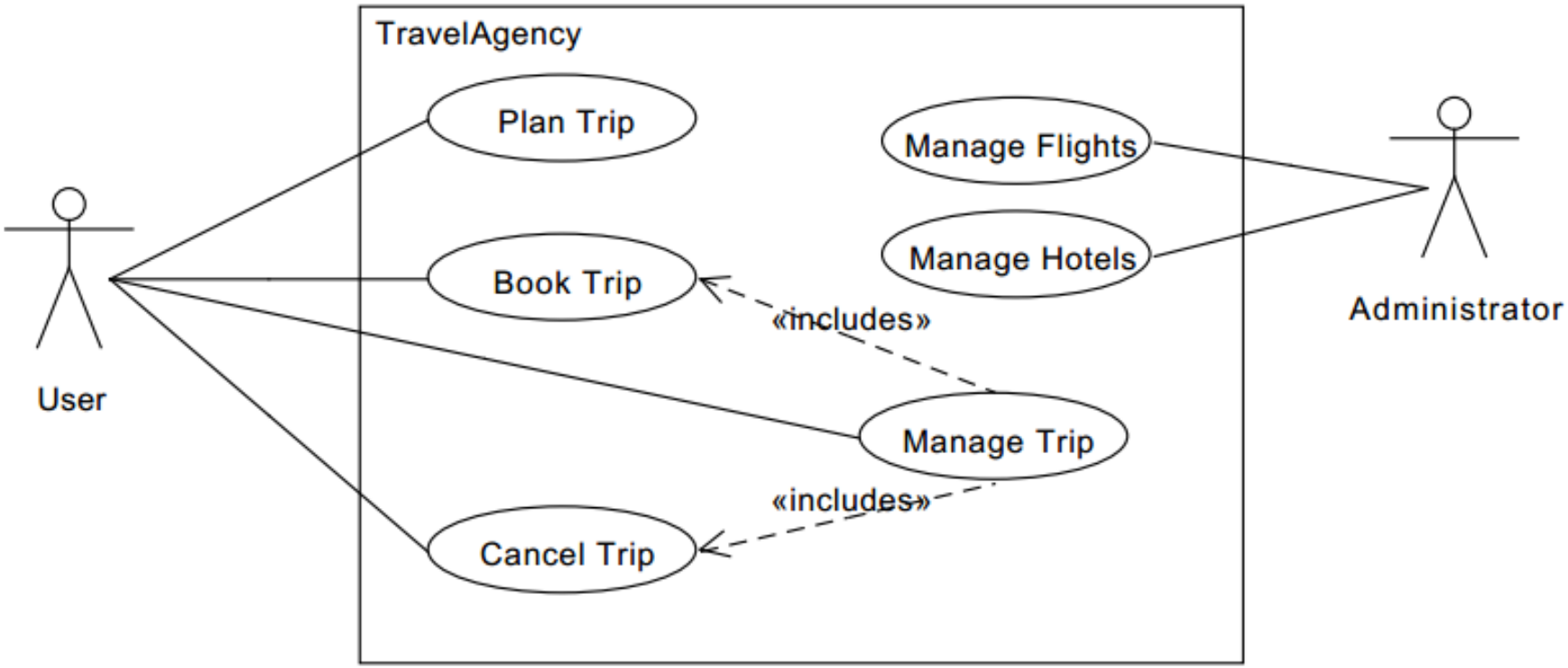


Use case

- Các loại sơ đồ use case
  - Business use case
    - Một phần của tài liệu yêu cầu người dùng
    - Mô tả chức năng từ góc nhìn của người dùng
  - System use case
    - Một phần của tài liệu kĩ thuật của đội phát triển
    - Mang tính kĩ thuật và chi tiết hơn
    - Tập trung vào mô tả những gì cần cài đặt

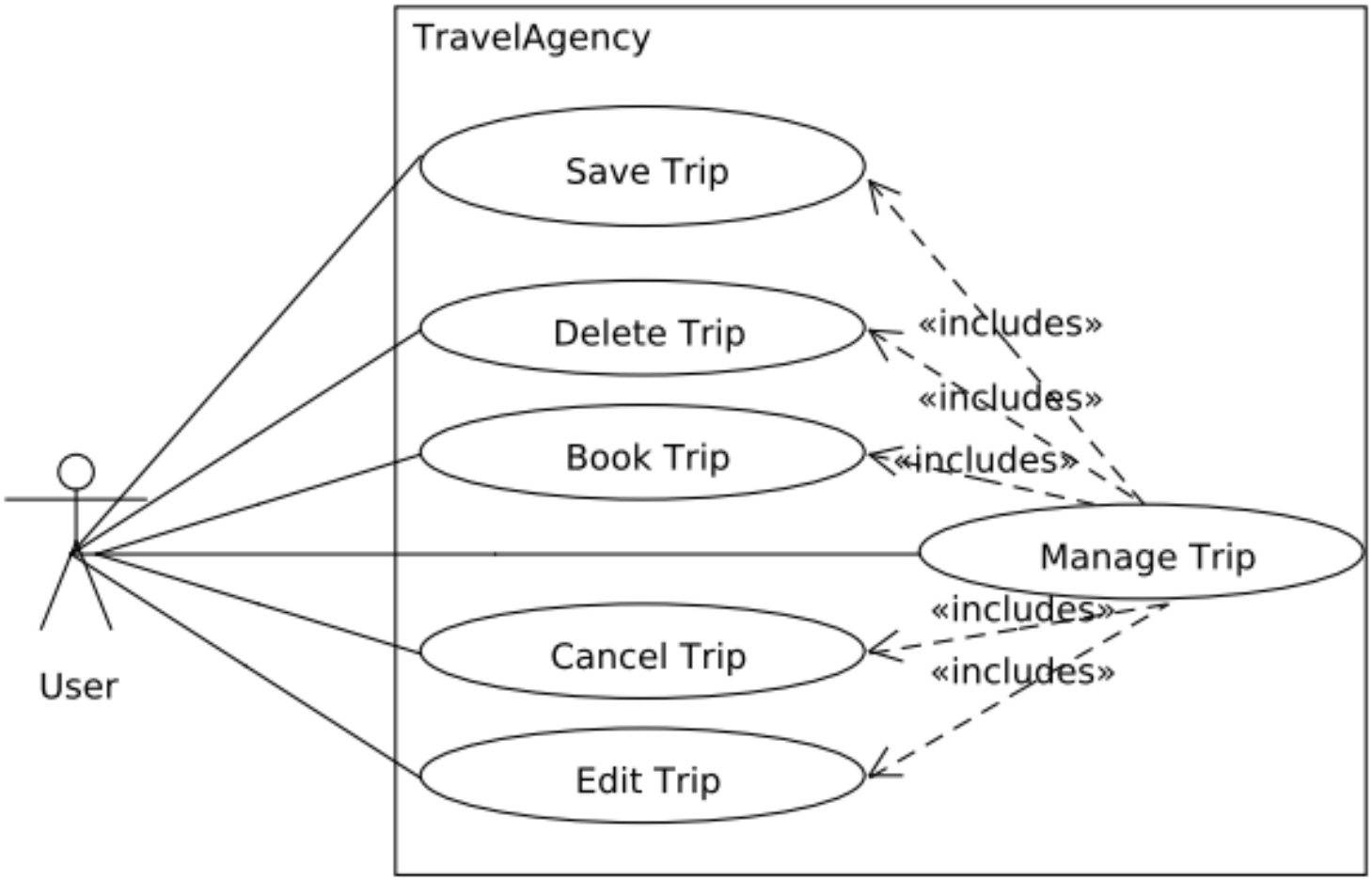
Use case

- Yêu cầu chức năng của TravelAgency:
  - Business use case



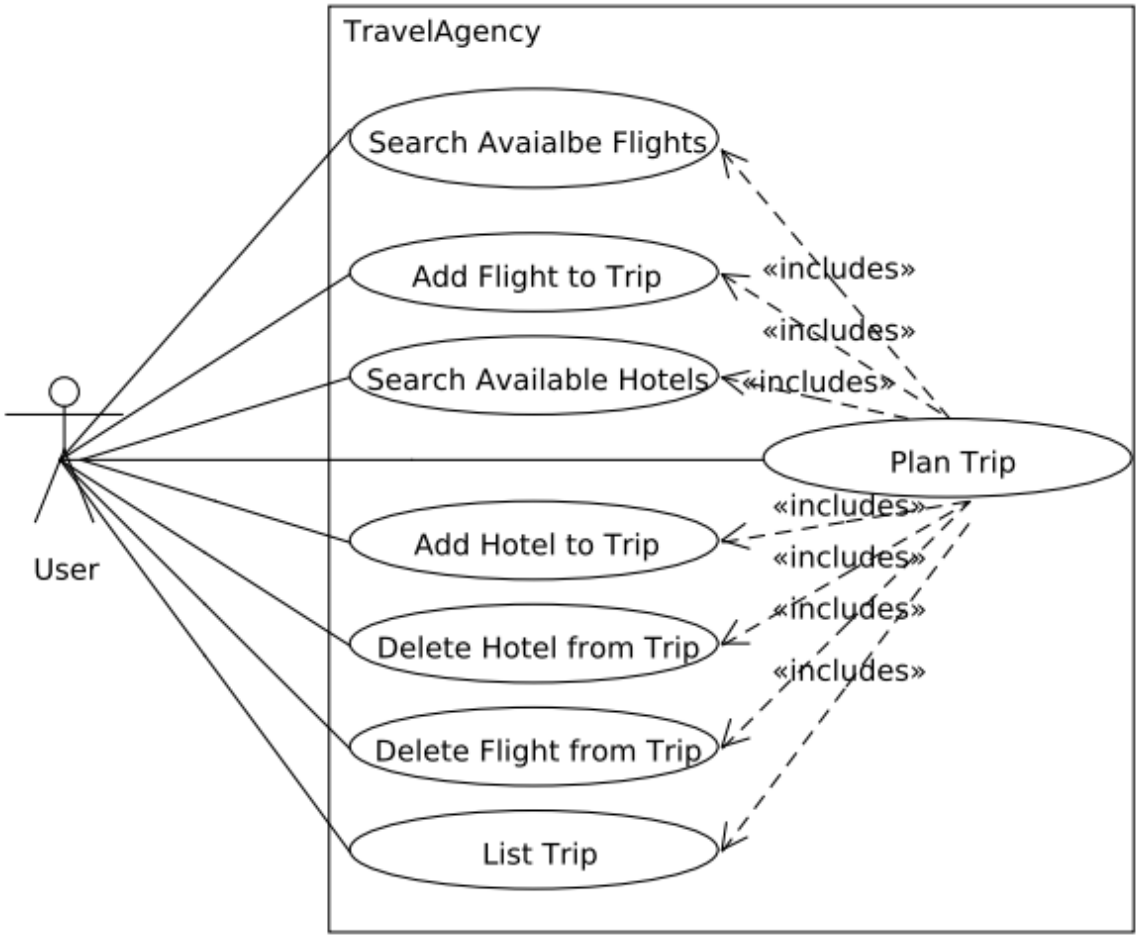
Use case

- Yêu cầu chức năng của TravelAgency:
  - System use case (phần 1: Manage Trip)



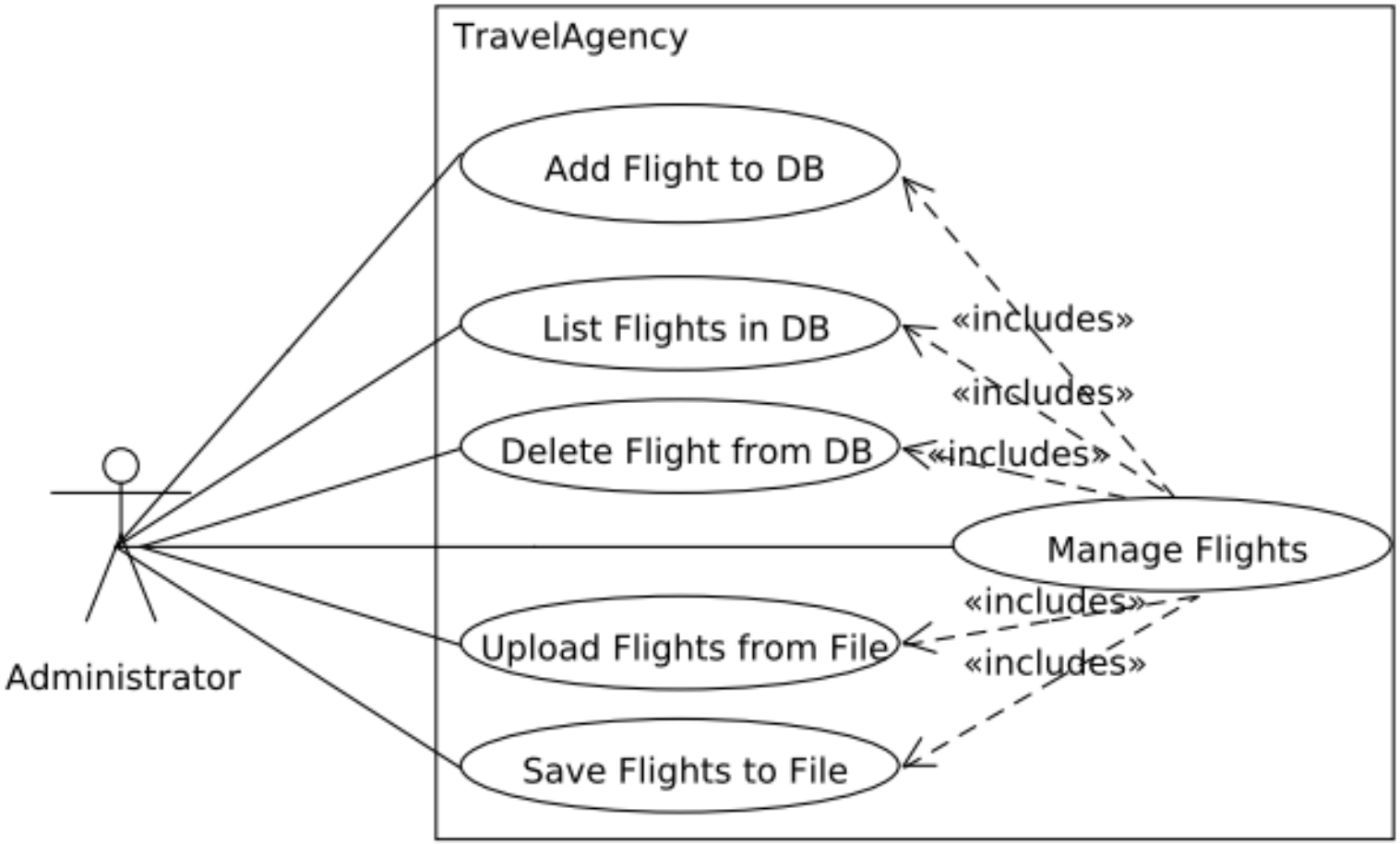
Use case

- Yêu cầu chức năng của TravelAgency:
  - System use case (phần 2: Plan Trip)



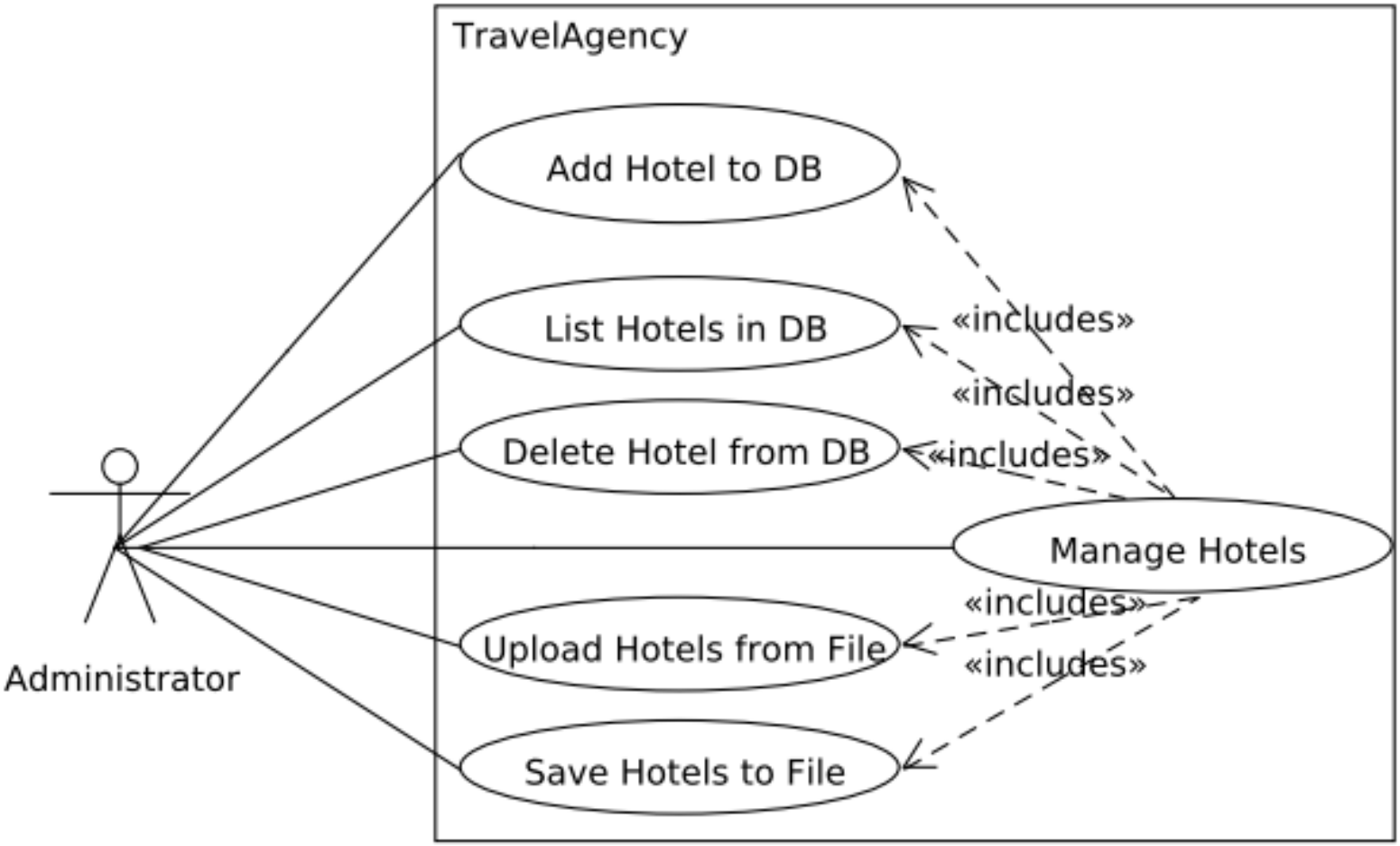
Use case

- Yêu cầu chức năng của TravelAgency:
  - System use case (phần 3: Manage Flights)



Use case

- Yêu cầu chức năng của TravelAgency:
  - System use case (phần 4: Manage Hotels)



Use case

- Kịch bản
  - Tương tác giữa một actor và hệ thống
    - Người dùng tác động vào hệ thống
    - Hệ thống phản ứng
    - Các hiệu ứng quan trọng đối với người dùng (người dùng thấy)
  - Hoạt động bên trong của hệ thống không phải là một phần của tương tác

Use case

- Mẫu tài liệu mô tả use case
  - **Tên**: Tên của use case
  - **Mô tả**: Mô tả ngắn gọn của use case (mục tiêu của ca sử dụng)
  - **Actor**: Một hoặc vài actor (nhân tố tương tác với hệ thống)
  - **Tiền điều kiện**: Các điều kiện hệ thống cần thỏa mãn để use case có thể hoạt động
  - **Kịch bản chính**: Mô tả chuỗi tương tác chính giữa actor và hệ thống
    - Chú ý: chỉ nên mô tả hệ thống từ góc nhìn của người sử dụng
  - **Các kịch bản phụ**: Có thể chứa các kịch bản thất bại
  - **Ghi chú**: Dùng cho tất cả những gì cần thiết nhưng lại không phù hợp với các thể loại trên

📖 Use case

- Travel Agency. Mô tả chi tiết use case: Cancel trip

Tên: Cancel trip

Mô tả: Người dùng hủy một chuyến đi đã đặt

Actor: Người dùng

Tiền điều kiện:

– Chuyến đi đã được đặt chỗ

– Ngày đầu tiên của thời gian đặt phòng khách sạn hoặc chuyến bay phải muộn hơn thời điểm hiện tại ít nhất 1 ngày.

Kịch bản chính:

1. Người dùng chọn chuyến đi để hủy
2. Hệ thống thông báo chi phí của việc hủy chuyến đi
3. Chuyến đi được chọn sẽ bị hủy sau khi người dùng khẳng định việc hủy

📖 Use case

- Travel Agency. Mô tả chi tiết use case: Plan trip (Use case này chứa/include các use case khác)

  Tên: Plan trip

  Mô tả: Người dùng lập kế hoạch một chuyến đi bao gồm khách sạn và các chuyến bay

  Actor: Người dùng

  Kịch bản chính: Lặp đi lặp lại hoạt động bất kì trong danh sách sau cho đến khi xong

  1. List available flights (use case)
  2. Add flight to trip (use case)
  3. List available hotels (use case)
  4. Add hotel to trip (use case)
  5. List trip (use case)
  6. Delete hotel from trip (use case)
  7. Delete flight from tip (use case)

Use case

- Travel Agency. Mô tả chi tiết use case: Save trip

Tên: Save trip

Mô tả: Người dùng đặt tên cho chuyến đi hiện hành và lưu lại cho sử dụng sau này

Actor: Người dùng

Tiền điều kiện: Chuyến đi hiện hành không rỗng

Kịch bản chính:

1. Người dùng nhập tên cho chuyến đi

Các kịch bản phụ:

1: Tên nhập không hợp lệ

2: Thông báo cho người dùng và kết thúc use case

1: Tên trùng với tên của một chuyến khác

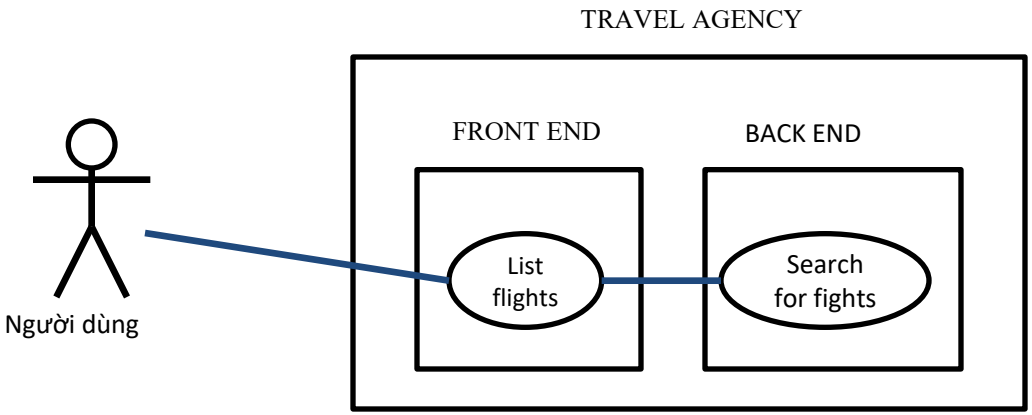
2: Hỏi người dùng xem có nên ghi đè lên chuyến đã ghi lần trước hay không

3a: Nếu người dùng đồng ý, ghi đè lên chuyến cũ

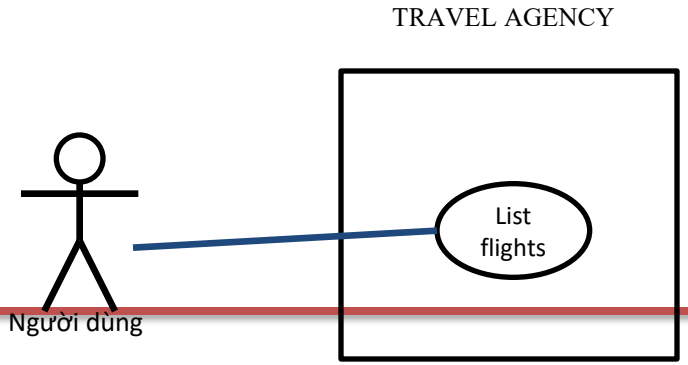
3b: Nếu người dùng không đồng ý, kết thúc use case

Use case

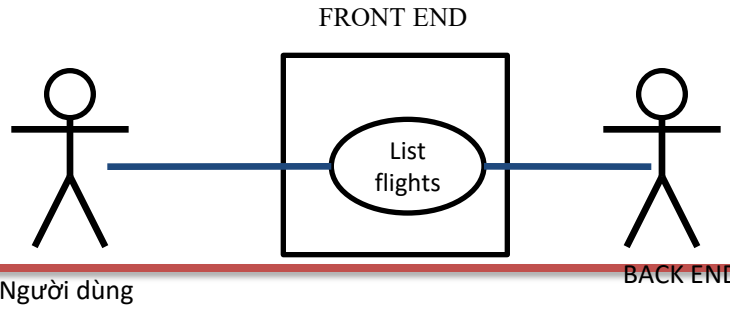
- Use case và ranh giới hệ thống
  - Use case và actor được xác định tùy theo ranh giới hệ thống



Biên hệ thống: TravelAgency



Biên hệ thống: Front End



Class diagram

- Class diagram (biểu đồ lớp)
  - Là biểu đồ quan trọng nhất
  - Mô tả các đối tượng và mối quan hệ của chúng trong hệ thống.
  - Mô tả các thuộc tính và các hành vi (behavior) của đối tượng.
  - Có biểu đồ lớp mức phân tích và mức cài đặt.

Class diagram

- Hai dạng lớp: phân tích và thiết kế

Analysis

Order
Placement Date Delivery Date Order Number
Calculate Total Calculate Taxes

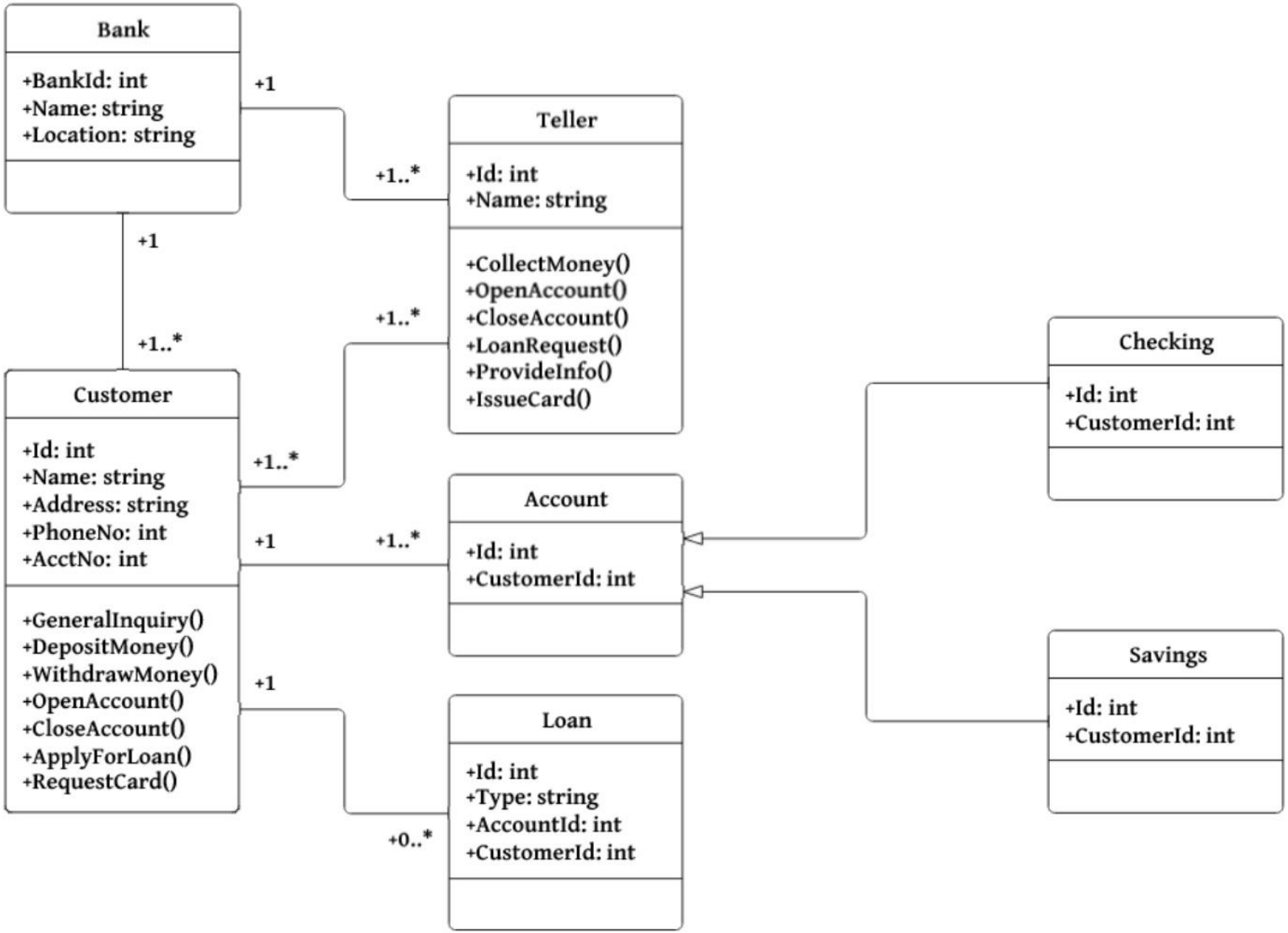
Bỏ qua các chi tiết không cần thiết

Design

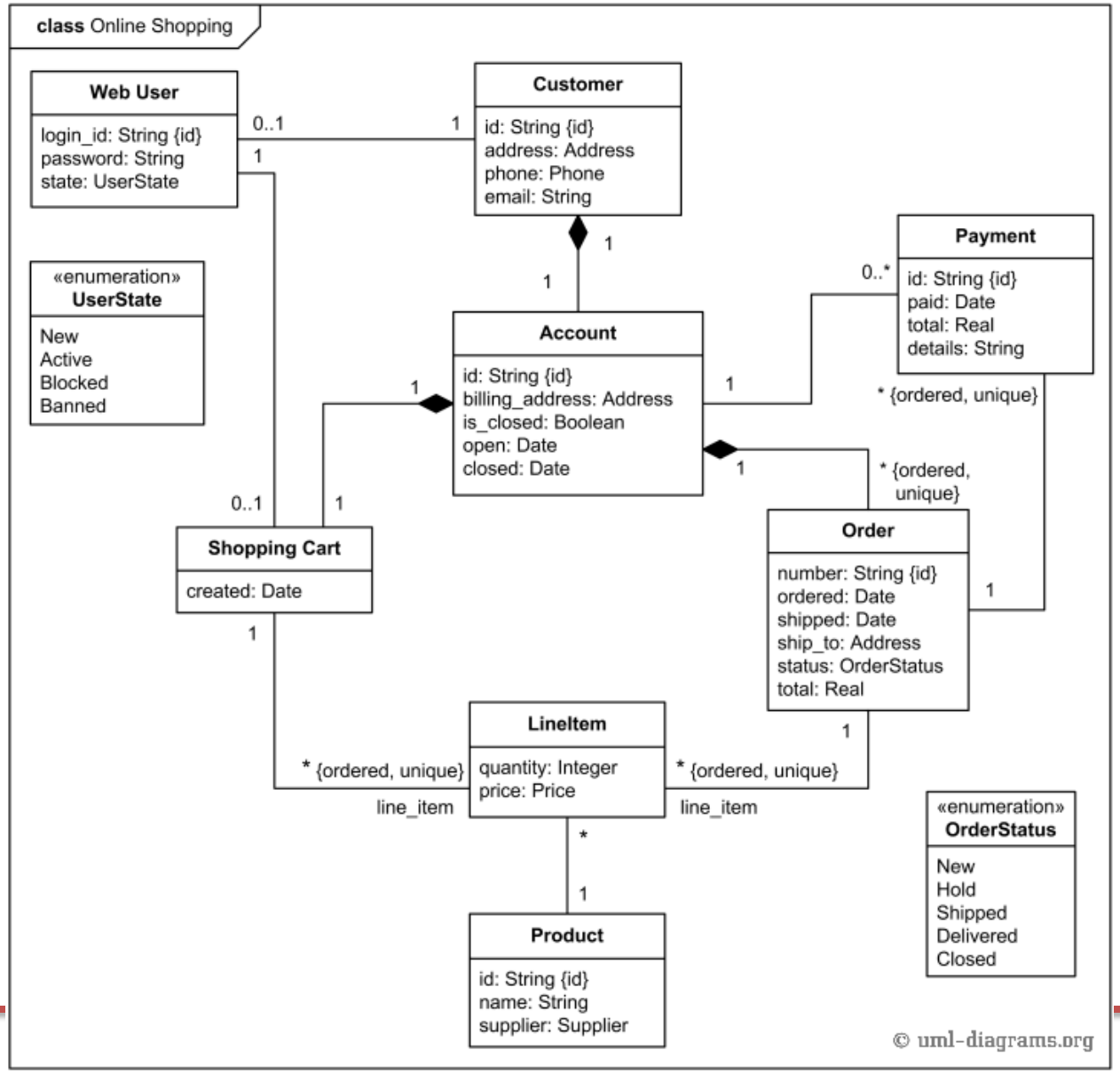
Order
- deliveryDate: Date - orderNumber: int - placementDate: Date - taxes: Currency - total: Currency
calculateTaxes(Country, State): Currency # calculateTotal(): Currency getTaxEngine() {visibility=implementation}

Phải đầy đủ & chi tiết các thành phần

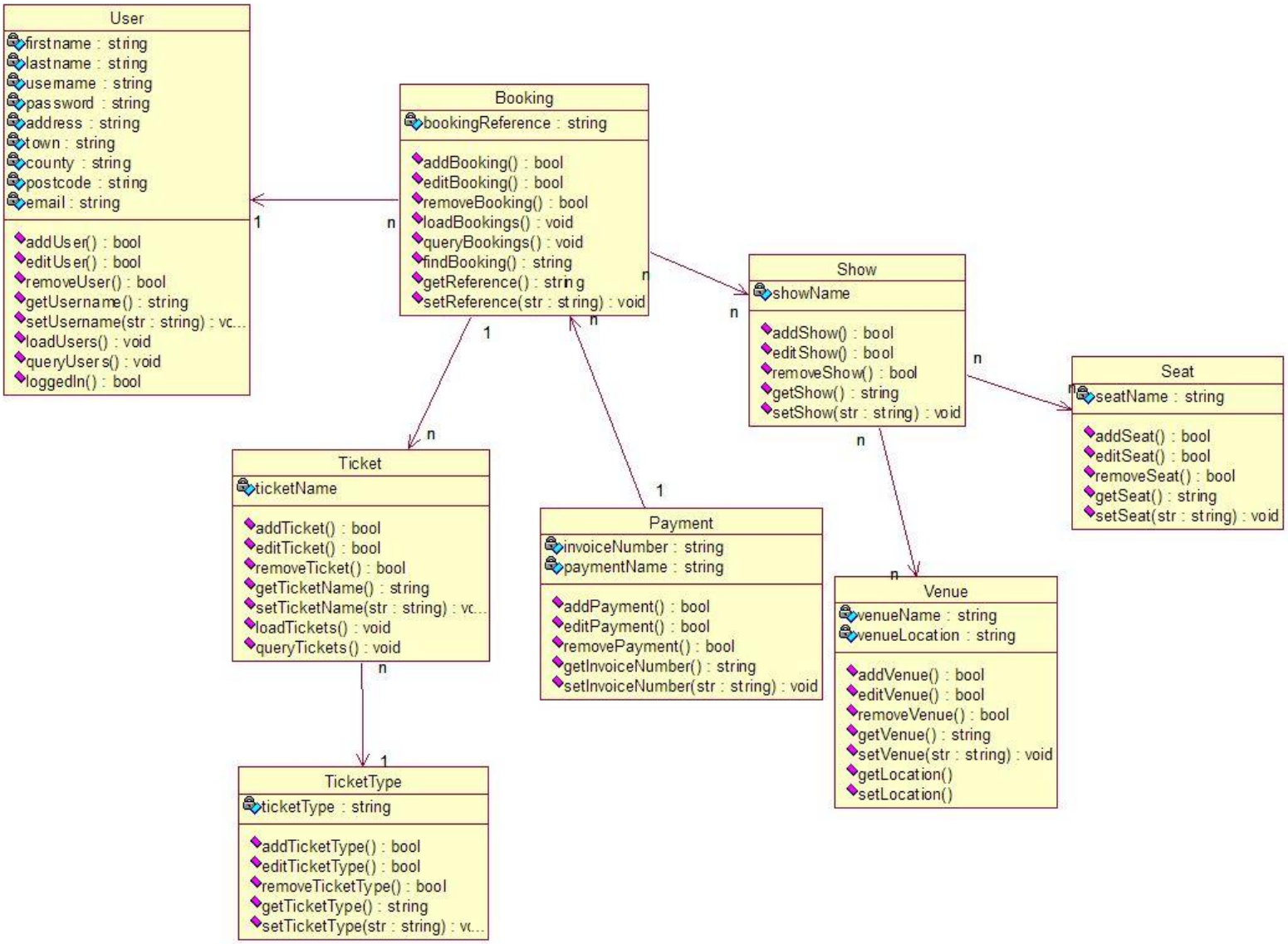
Class diagram



Class diagram

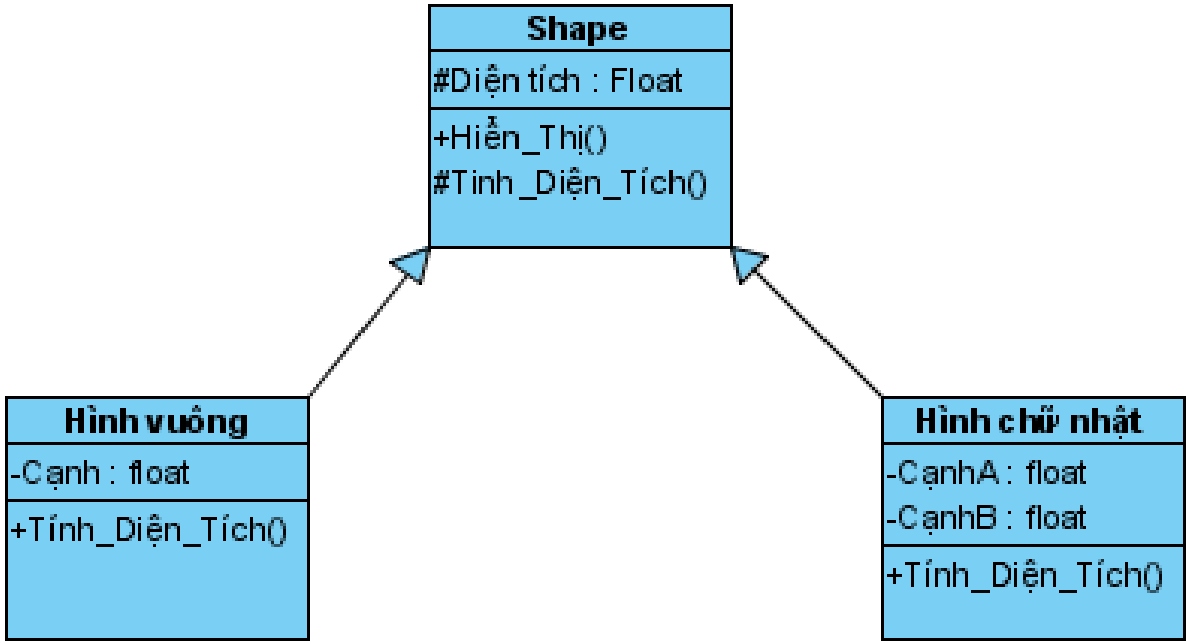


Class diagram



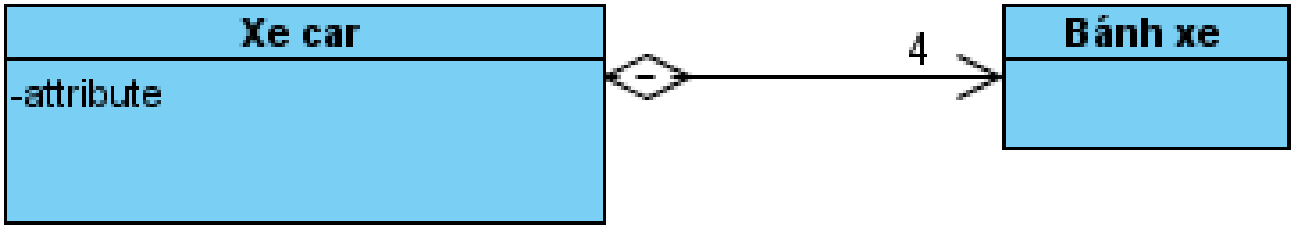
Class diagram

- Một số quy tắc:
  - Quan hệ Generalization: Thể hiện rằng một lớp A kế thừa từ một lớp B (Hay A là trường hợp riêng của B; B là tổng quát của A)



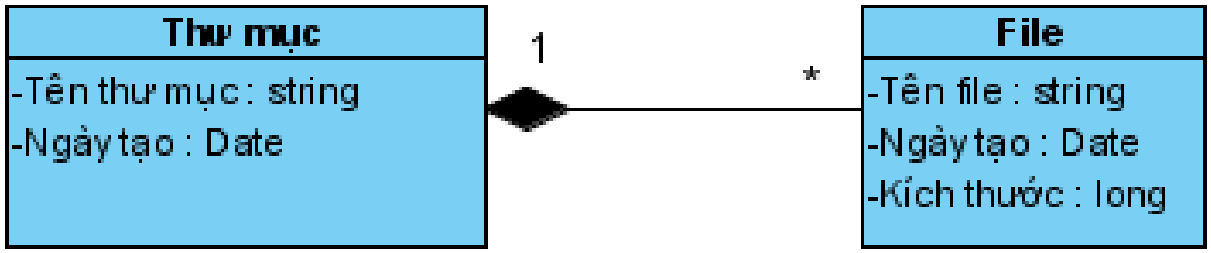
Class diagram

- Một số quy tắc:
  - Quan hệ Aggregation: Thể hiện rằng một lớp A nào đó bao gồm lớp B. Lớp B này có thể tồn tại độc lập mà không cần lớp A.



Class diagram

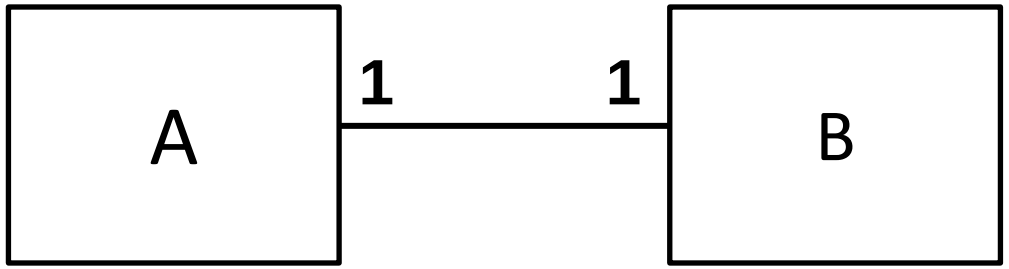
- Một số quy tắc:
  - Quan hệ Composition: thể hiện rằng một lớp A bao hàm lớp B. Nhưng lớp B không thể tồn tại độc lập (Tức không thuộc lớp nào). Tức là, nếu có B thì phải suy ra được A.



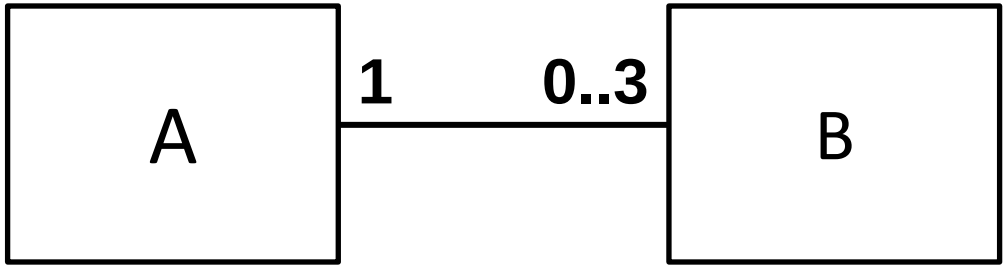
Class diagram

- Multiplicity:
  - Thể hiện rằng ứng với mỗi lớp A thì có (chứa, dạy, có, mua, đặt,...) bao nhiêu phần tử lớp B?

- Một phần tử lớp A có 1 phần tử lớp B



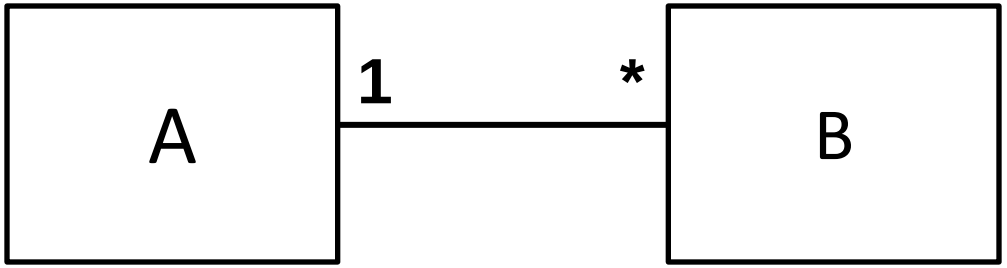
- Một phần tử lớp A có tối đa 3 phần tử lớp B
- Mỗi phần tử lớp B có đúng 1 phần tử lớp A



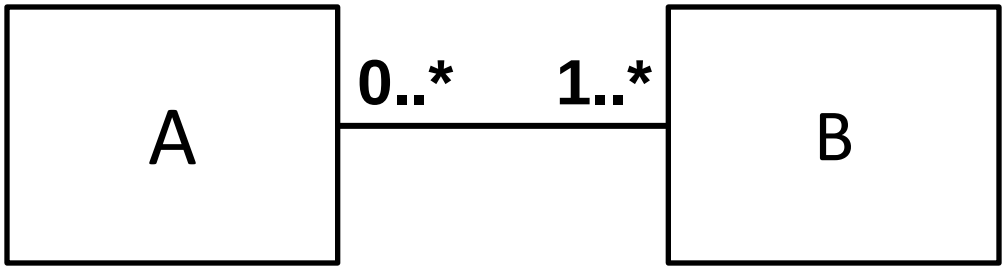
Class diagram

- Multiplicity:

- Một phần tử lớp A có nhiều phần tử lớp B
- Mỗi phần tử lớp B có đúng 1 phần tử lớp A



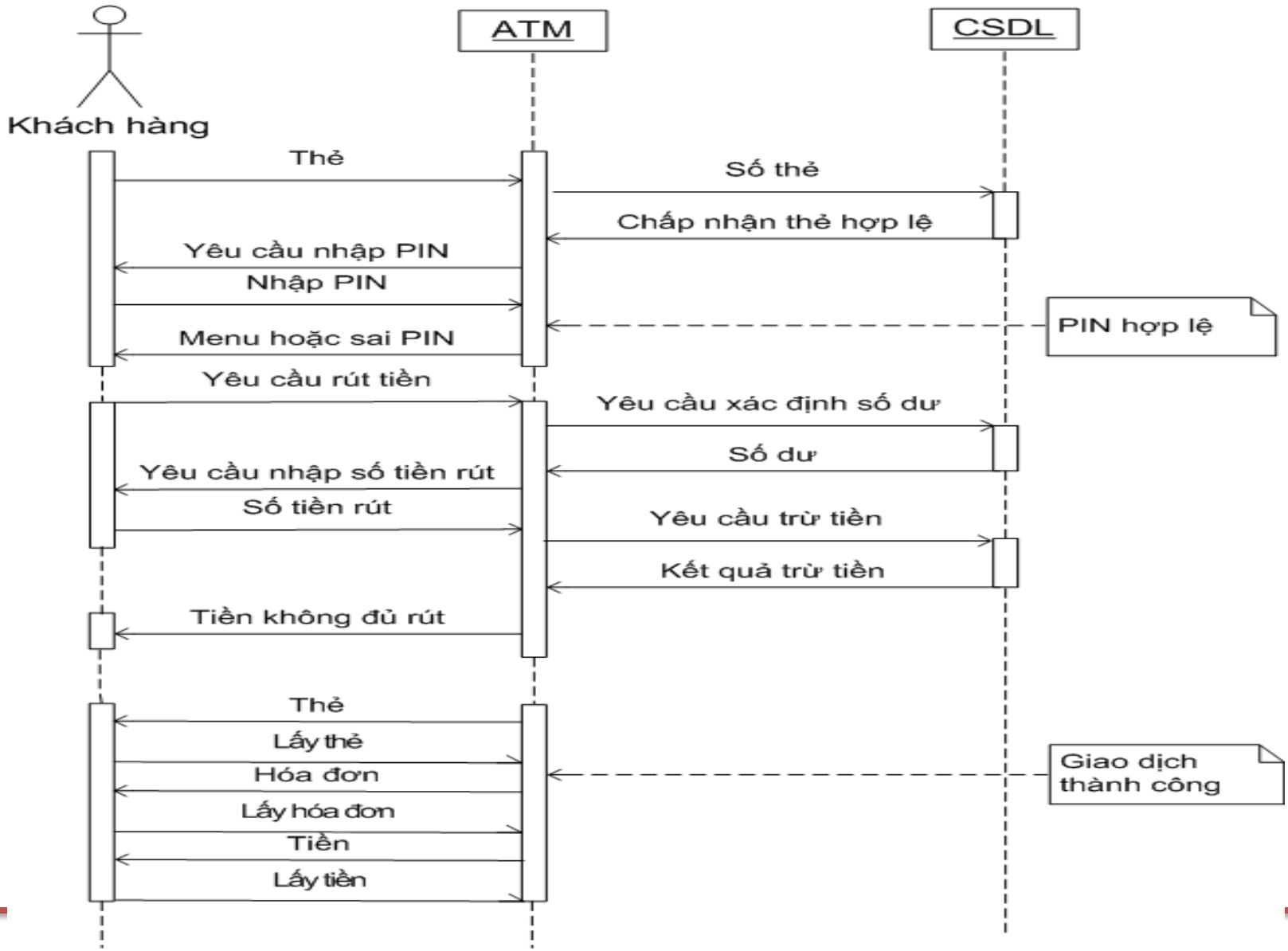
- Mỗi sinh viên tham gia ít nhất 1 khóa học
- Mỗi khóa học có thể có 0 hoặc nhiều sv tham gia



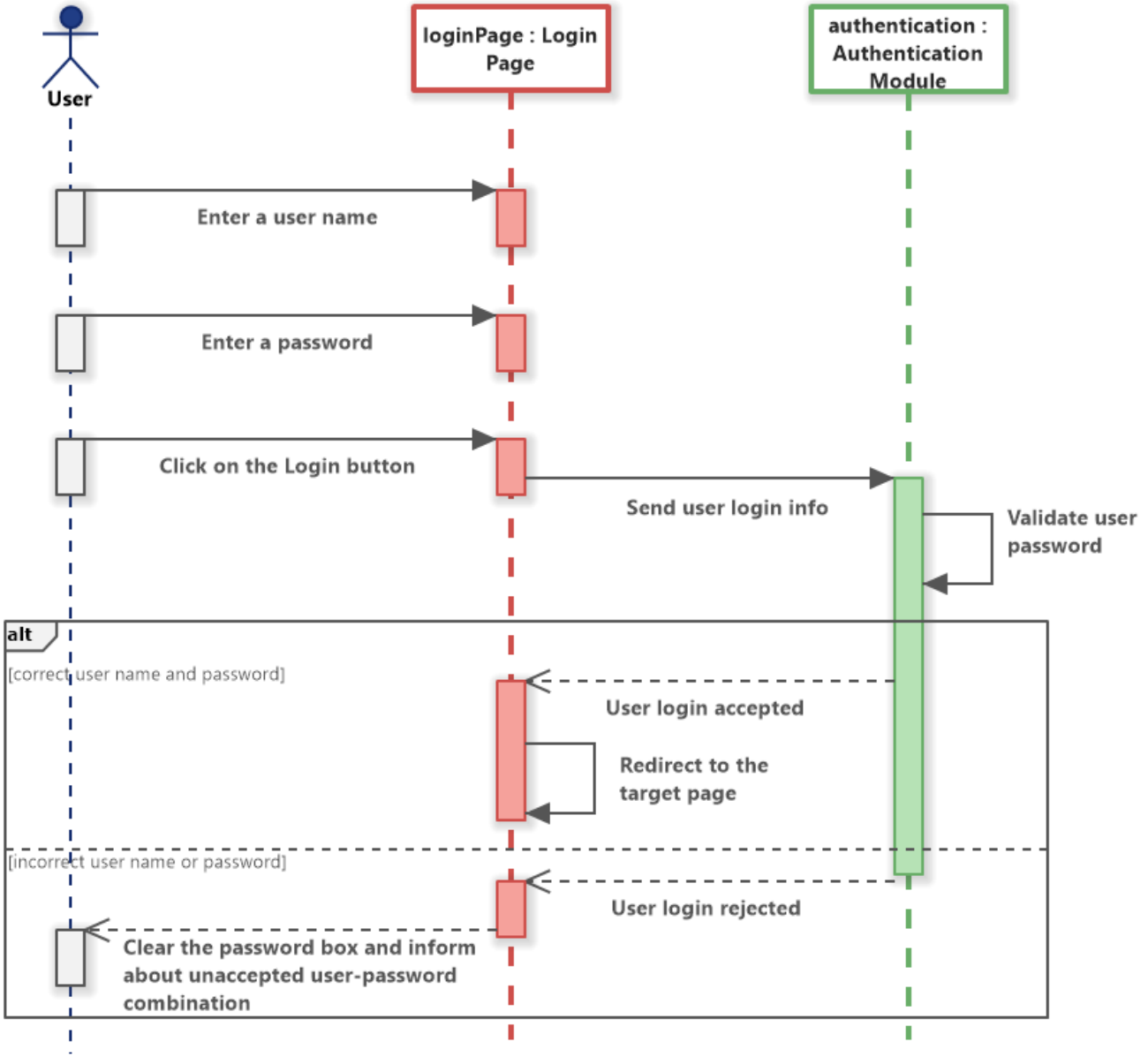
🕮 Sequence diagram

- Sequence diagram (biểu đồ tuần tự):
  - Mô tả sự tương tác của các đối tượng theo trình tự về thời gian.
  - Có sự liên kết chặt chẽ với biểu đồ lớp.
  - Mỗi biểu đồ tuần tự mô tả một tình huống xử lý.

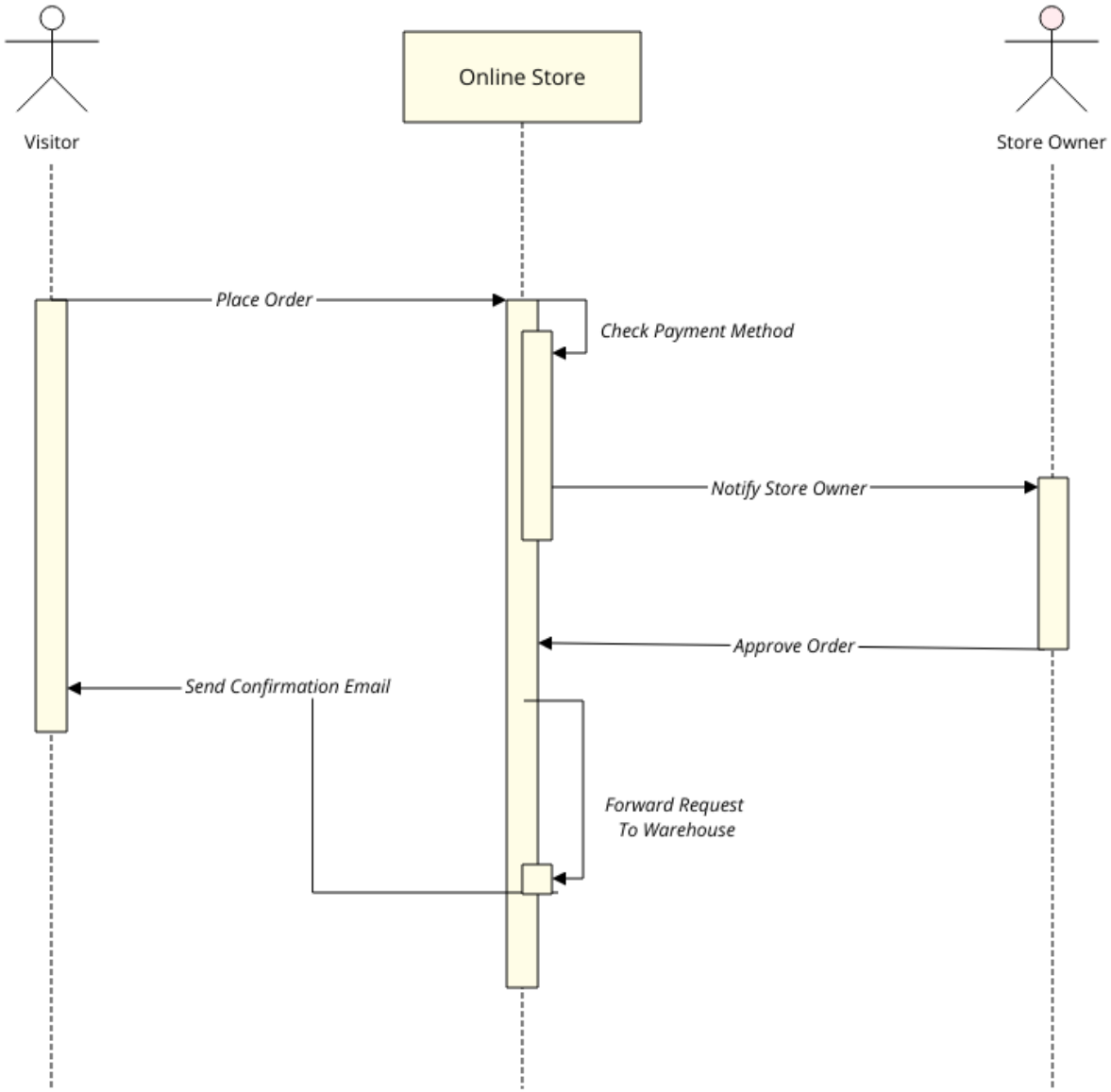
Sequence diagram



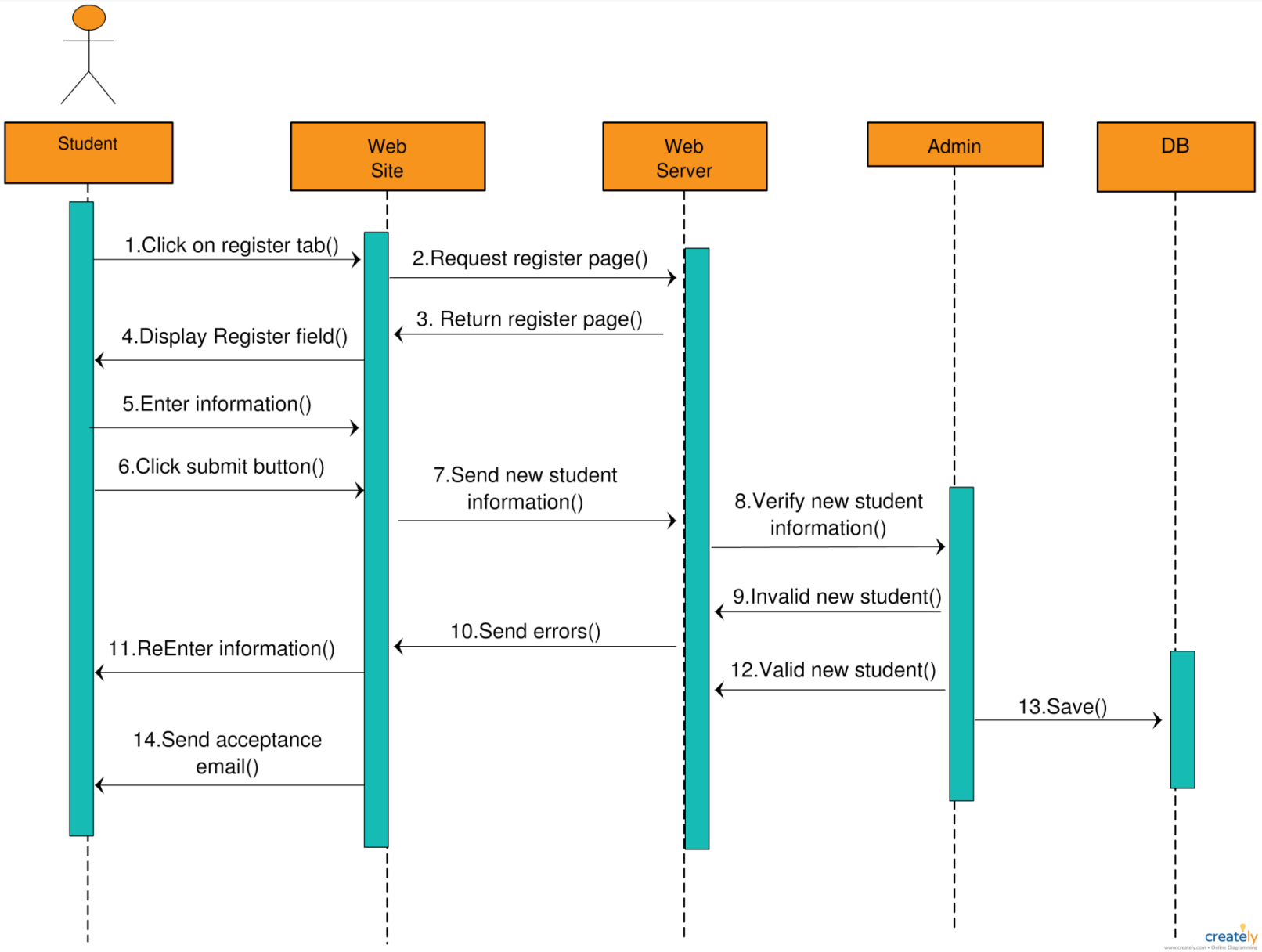
Sequence diagram



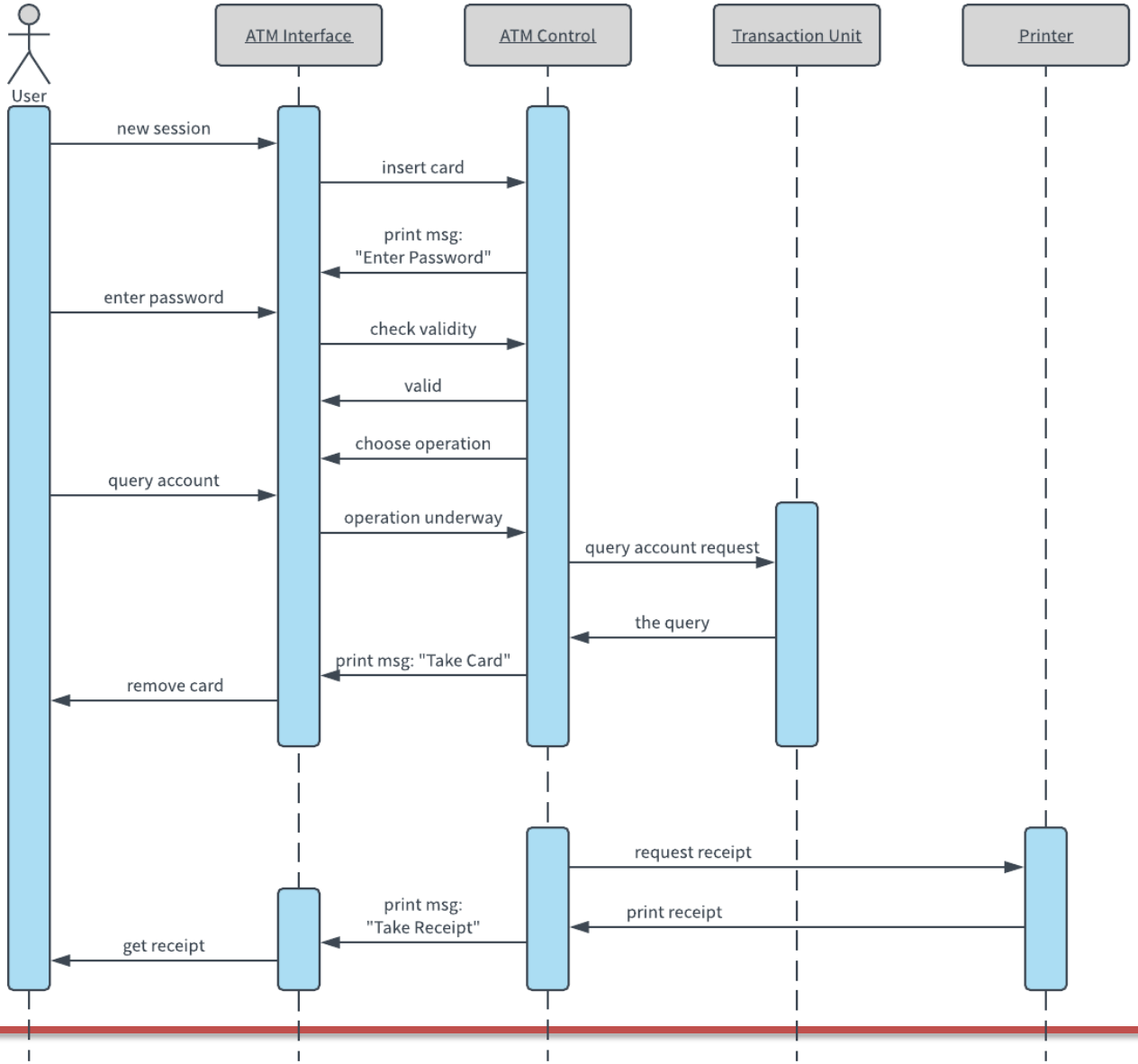
Sequence diagram



Sequence diagram



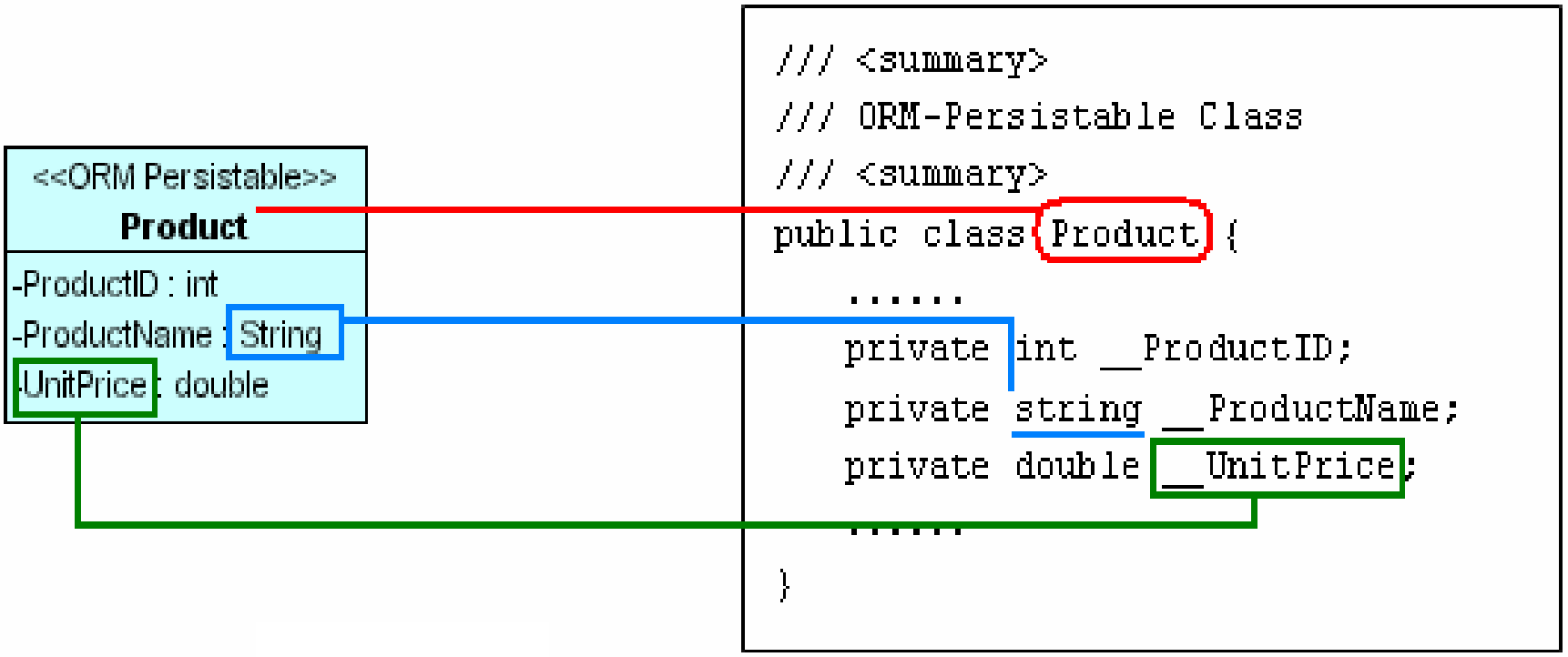
Sequence diagram



Một số biểu đồ khác

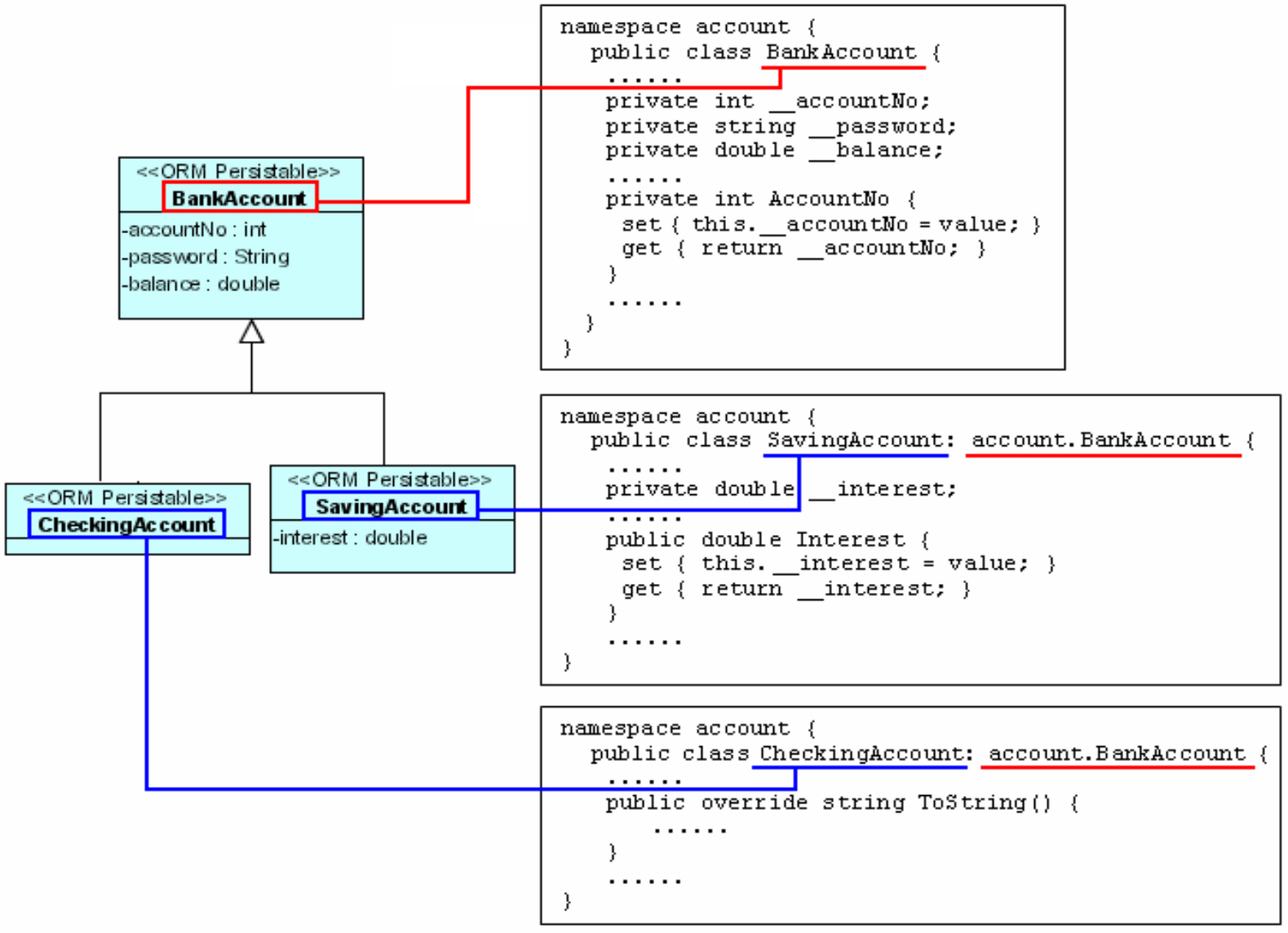
- Biểu đồ truyền thông: Communication diagram
- Biểu đồ tương tác: Interaction Diagram
- Biểu đồ thời gian: Timming diagram
- Biểu đồ trạng thái: State Diagram
- Biểu đồ đối tượng: Object Diagram
- Bểu đồ gói: Package Diagram
- Biểu đồ cấu trúc kết hợp: Composite Structured
- Biểu đồ thành phần: Component Diagram
- Biểu đồ triển khai: Deployment Diagram

Ánh xạ biểu đồ sang Code



- Mapping Classes, Attributes and Data Type

Ánh xạ biểu đồ sang Code



Mapping generalization

2.6. Thẩm định yêu cầu

- Thẩm định yêu cầu (Requirement validation): quan tâm đến việc chứng tỏ rằng các yêu cầu định nghĩa được hệ thống mà khách hàng thực sự muốn.
- Chi phí của lỗi yêu cầu cao đến mức công đoạn thẩm định rất quan trọng
  - Việc sửa một lỗi yêu cầu sau khi đã bàn giao phần mềm có thể tốn kém gấp nhiều lần chi phí cho việc sửa một lỗi cài đặt.

Các tiêu chí cho thẩm định

- Hiệu lực (Validity)
  - Hệ thống có cung cấp những chức năng phục vụ tốt nhất yêu cầu của khách hàng hay không?
- Nhất quán (Consistency)
  - Có những yêu cầu nào xung đột nhau không?
- Đầy đủ (Completeness)
  - Có đủ các chức năng mà khách hàng đòi hỏi hay không?
- Thực tế (Realism)
  - Có thể cài đặt các yêu cầu trong phạm vi công nghệ và ngân sách cho phép hay không?
- Kiểm định được (Verifiability)
  - Có cách kiểm tra các yêu cầu xem chúng đã được thỏa mãn chưa hay không?

Kĩ thuật thẩm định yêu cầu

- Duyệt yêu cầu (Requirements reviews)
  - Đọc và phân tích lại một cách có hệ thống (không dùng chương trình tự động).
- Phiên bản thử nghiệm (Prototyping)
  - Dùng một mô hình chạy được của hệ thống để kiểm tra các yêu cầu
- Tạo các ca kiểm thử (Test case)
  - Viết các test dành cho các yêu cầu để kiểm tra khả năng kiểm thử được.

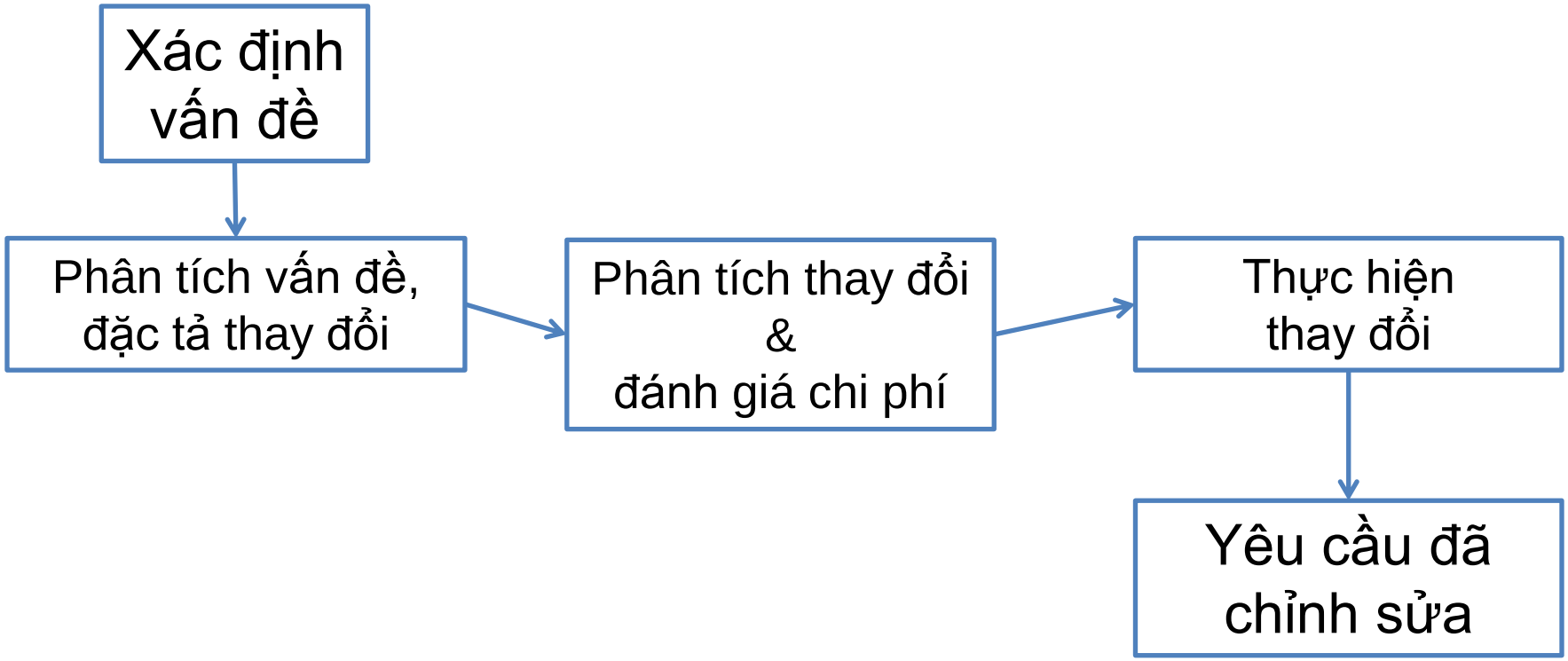
Quản lý yêu cầu

- Quản lý yêu cầu là quy trình quản lý sự thay đổi của các yêu cầu trong quá trình phát hiện yêu cầu và phát triển hệ thống.
- Các yêu cầu thường không đầy đủ và không đồng nhất là do:
  - Những yêu cầu mới xuất hiện khi các yêu cầu nghiệp vụ thay đổi và khi chúng ta có hiểu biết sâu hơn về hệ thống sẽ xây dựng.
  - Các yêu cầu thường xuất hiện các mâu thuẫn do khung nhìn khác nhau, khách hàng thường đặc tả sai về yêu cầu của stakeholder
  - Thứ tự ưu tiên của yêu cầu thay đổi trong quá trình phát triển hệ thống.
  - Môi trường nghiệp vụ và môi trường kỹ thuật của hệ thống thay đổi trong quá trình xây dựng.
  - Luật thay đổi, nhu cầu doanh nghiệp thay đổi → thay đổi chức năng
  - Xung đột giữa các yêu cầu mới nảy sinh, và giữa yêu cầu mới với yêu cầu cũ

Quản lý yêu cầu

- Để quản lý yêu cầu, cần xác định
  - Định danh yêu cầu: mỗi yêu cầu có định danh riêng để tiện cho việc tham chiếu giữa các yêu cầu và lần vết
  - Quy trình quản lý thay đổi: các hoạt động đánh giá ảnh hưởng và chi phí của thay đổi
  - Chính sách lần vết: cách ghi lại và lưu trữ quan hệ giữa các yêu cầu và giữa mỗi yêu cầu với thiết kế tương ứng với nó
  - Công cụ hỗ trợ: hỗ trợ thực hiện các công việc trên một cách có hiệu quả. VD: CASE tool, spreadsheet, cơ sở dữ liệu...

Quản lý thay đổi yêu cầu



Quản lý thay đổi yêu cầu

- Nên áp dụng cho tất cả các thay đổi được đề xuất đối với bộ yêu cầu.
- Các giai đoạn chính
  - Phân tích vấn đề: Thảo luận về vấn đề của các yêu cầu và đề xuất thay đổi; Bổ sung chi tiết; Chốt lại những điểm sẽ thay đổi.
  - Phân tích thay đổi và đánh giá chi phí; Đánh giá hiệu ứng của thay đổi đối với các yêu cầu khác; Ra quyết định có thực hiện thay đổi hay không.
  - Thực hiện thay đổi; Cập nhật tài liệu yêu cầu và các tài liệu khác để thực hiện thay đổi đã xét.

📖 Câu hỏi thảo luận