

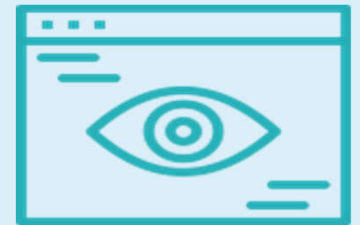
# CÔNG NGHỆ PHẦN MỀM

Khoa Công nghệ Thông tin – Đại học Đà Lạt



# CHƯƠNG 4.

## PHÂN TÍCH YÊU CẦU PHẦN MỀM



# Mục tiêu

---

- Sau khi hoàn thành bài học này, sinh viên có thể:
  - Hiểu được phân tích là gì, tầm quan trọng của việc phân tích
  - Nắm được các nguyên tắc và các bước trong giai đoạn phân tích yêu cầu
  - Xây dựng được mô hình phân tích khía cạnh tĩnh của hệ thống
  - Hiểu rõ vai trò của việc phân tích khía cạnh động
  - Xây dựng được sơ đồ tuần tự, cộng tác, hoạt động

# Nội dung

---

- Giới thiệu
- Một số nguyên tắc phân tích yêu cầu
- Các bước trong quá trình phân tích yêu cầu
- Xây dựng sơ đồ lớp
- Xây dựng sơ đồ cộng tác
- Xây dựng sơ đồ tuần tự
- Xây dựng sơ đồ hoạt động

# Giới thiệu



How the customer explained it



How the Project Leader understood it



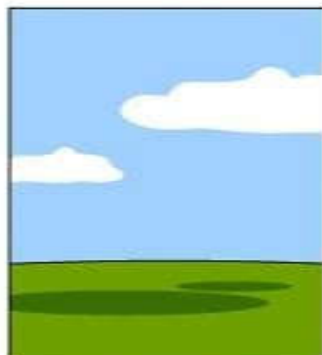
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



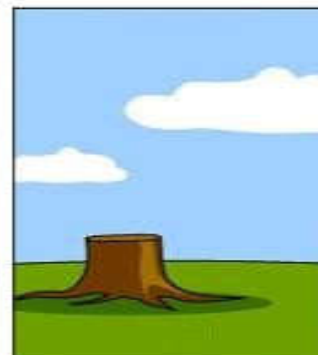
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

## Phân tích yêu cầu phần mềm là gì?

---

- Phân tích có nghĩa là mổ xẻ một vấn đề bằng nhiều khía cạnh mà mình nhận thấy và quan sát được, sau đó khái quát lên quan điểm của mình (có điểm nhấn).
- Phân tích yêu cầu phần mềm:
  - Nhằm chỉ ra những việc mà hệ thống sẽ xử lý, không phải mô tả cách thức hệ thống xử lý
  - Phân rã các yêu cầu phức tạp thành các thành phần cơ bản và mối quan hệ giữa chúng nhằm mục đích hiểu chức năng, tính chất và mối quan hệ giữa chúng
- Giúp nhóm phát triển hệ thống có thể hiểu rõ được hệ thống thông qua việc mô hình hóa thế giới thực thành các đối tượng.

# Mô hình hóa hệ thống là gì?

---

- Mô hình hóa hệ thống
  - Là quá trình phát triển các mô hình trừu tượng về hệ thống, mỗi mô hình biểu diễn một cách nhìn khác nhau hay một khía cạnh của hệ thống.
  - Là biểu diễn hệ thống dưới dạng các ký pháp đồ họa, thường là dựa trên UML.
- Tác dụng:
  - Giúp các nhà phân tích hiểu được chức năng của hệ thống
  - Các mô hình sẽ được dùng để trao đổi, giao tiếp với khách hàng

## Các khía cạnh của hệ thống

---

- Khía cạnh bên ngoài: ngữ cảnh hay môi trường của hệ thống.
- Khía cạnh tương tác: các tương tác giữa hệ thống với môi trường của nó hoặc giữa các thành phần của hệ thống.
- Khía cạnh cấu trúc (tĩnh): cách thức tổ chức của hệ thống hoặc cấu trúc của dữ liệu được xử lý bởi hệ thống.
- Khía cạnh hoạt động (động): các hoạt động bên trong hệ thống và cách nó trả lời/phản hồi các sự kiện.



# Tầm quan trọng của phân tích yêu cầu

---

- Trực quan hóa những hệ thống phức tạp
  - Dễ giao tiếp, dễ truyền đạt
  - Giúp thực hiện những giải pháp dễ dàng hơn. Dễ so sánh → tối ưu hóa
- Nắm bắt được các yêu cầu của tổ chức, xác định phạm vi hệ thống
- Biểu diễn sự thay đổi, cải tiến qui trình đã tồn tại, hoặc xây dựng qui trình mới, hoặc nâng cấp môi trường, ...
- Hiểu được chi tiết cấu trúc và các hoạt động của tổ chức được triển khai hệ thống.
- Hiểu rõ các vấn đề hiện tại trong tổ chức và xác định những hạn chế, các vấn đề bất cập cần cải tiến và hướng giải quyết.
- Bảo đảm rằng các khách hàng, người dùng cuối, và các nhà phát triển có sự hiểu biết chung về tổ chức.

# Input - Output

---

- Input:

- Mô hình các yêu cầu nghiệp vụ: mô tả các dòng công việc nghiệp vụ tự động hóa hoặc được làm bằng tay, thường được mô tả qua các đối tượng, thuật ngữ, sơ đồ use case, sơ đồ hoạt động nghiệp vụ.
- Mô hình các yêu cầu hệ thống: cách nhìn hệ thống từ bên ngoài, được mô tả thông qua các sơ đồ use case hệ thống, các bản phác thảo giao diện người dùng và các yêu cầu phi chức năng

- Output: Bản đặc tả yêu cầu (đặc tả đặc trưng hoạt động của phần mềm)

- Giải thích về hệ thống ở mức độ chi tiết vừa phải.
- Chỉ ra giao diện của các thành phần của hệ thống
- Xây dựng các yêu cầu bắt buộc hệ thống phải tuân thủ

## Các sơ đồ được sử dụng trong phân tích

---

- Sơ đồ hoạt động: chỉ ra các hoạt động liên quan tới một xử lý hoặc trong việc xử lý dữ liệu.
- Sơ đồ use case: chỉ ra các tương tác giữa hệ thống với môi trường bên ngoài.
- Sơ đồ tuần tự: cho thấy tương tác giữa các actor với hệ thống và giữa các thành phần của hệ thống.
- Sơ đồ cộng tác: thể hiện sự cộng tác giữa các đối tượng để hoàn thành một xử lý.
- Sơ đồ lớp: cho thấy các lớp đối tượng trong hệ thống và mối kết hợp giữa chúng.
- Sơ đồ trạng thái: chỉ ra cách mà hệ thống phản ứng lại với các sự kiện ở bên trong hoặc bên ngoài.

# Nguyên tắc khi phân tích

---



Mô hình cần phải tập trung vào các yêu cầu có thể nhìn thấy được trong vấn đề cần phân tích



Mỗi thành phần của mô hình yêu cầu phải đóng góp vào sự hiểu biết tổng thể về các yêu cầu phần mềm và cung cấp một cách nhìn sâu sắc về phạm vi của thông tin, chức năng và các hoạt động của hệ thống.



Không xem xét cơ sở hạ tầng và các mô hình phi chức năng khác cho đến khi đi vào giai đoạn thiết kế

## Nguyên tắc khi phân tích (tt.)

---



Tối thiểu hóa sự phụ thuộc bên trong hệ thống.



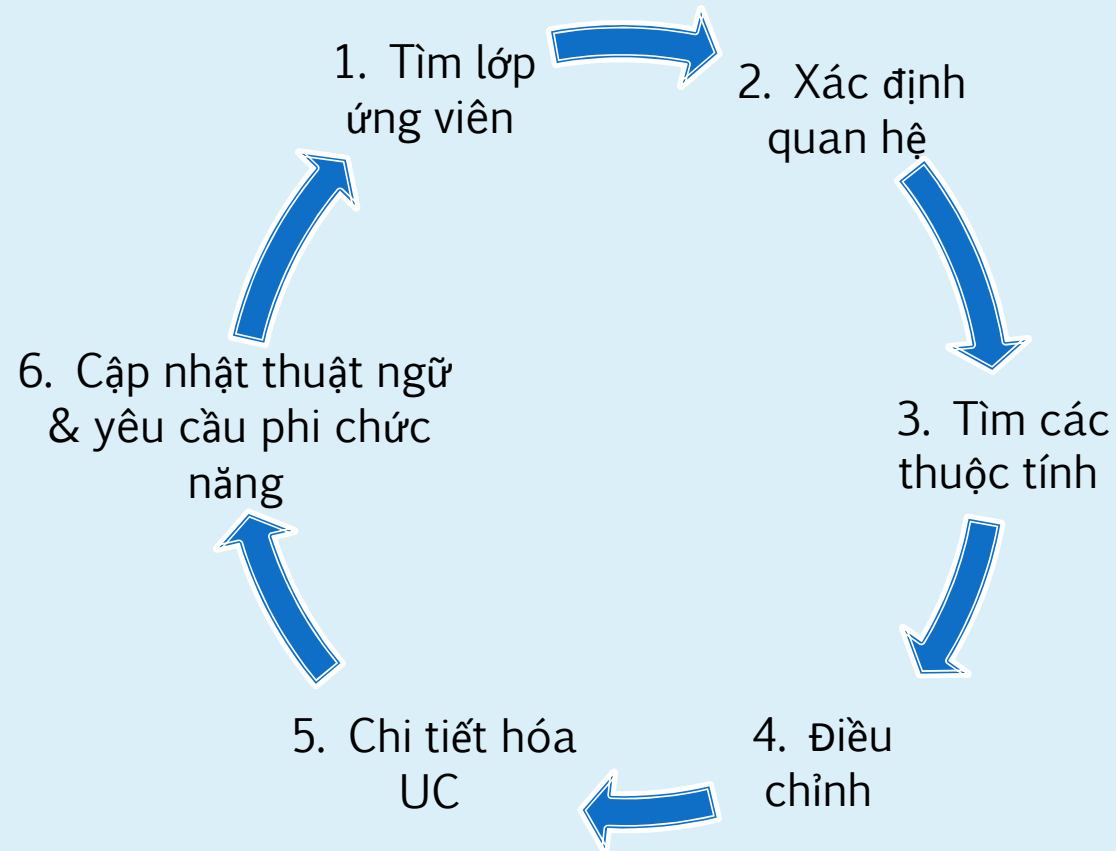
Cần chắc chắn rằng mô hình yêu cầu có giá trị cho tất cả các bên liên quan vì mỗi bên sử dụng mô hình theo một cách khác nhau.



Giữ cho mô hình ở mức đơn giản nhất có thể. Đừng tạo thêm sơ đồ khi chúng không bổ sung thông tin gì mới.

# Các bước phân tích

---



## Xây dựng sơ đồ lớp

---

- Là một quá trình lặp lại, kết quả bước sau có thể làm thay đổi kết quả đã có từ bước trước
- Kết thúc khi có sự thống nhất với chuyên gia trong lĩnh vực ứng dụng và nghiệp vụ.
- Không có phương thức chung để xác định các lớp
- Các bước thực hiện
  - Tìm các lớp ứng viên
  - Nhận biết mối quan hệ giữa các lớp
  - Vẽ sơ đồ lớp
  - Bổ sung thuộc tính
  - Bổ sung các lớp kết hợp

## Tìm các lớp ứng viên

---

- Vì kết quả bước sau có thể làm thay đổi kết quả đã có từ bước trước nên các lớp tìm thấy tại mỗi bước thường được gọi là các lớp ứng viên
- Lớp ứng viên thường được xác định bởi các danh từ trong văn bản mô tả các use case hoặc các mô tả yêu cầu.
- Những danh từ này thường biểu diễn
  - Bản thân hệ thống: e-Learning, iCoot, ...
  - Actor: Assistant, Head officer, ...
  - Phạm vi: Head office link, ...
  - Các kiểu dữ liệu: String, Number, ...
  - Giao diện người dùng: Customer applet, Catalog View, ...



## Tìm các lớp ứng viên (tt)

---

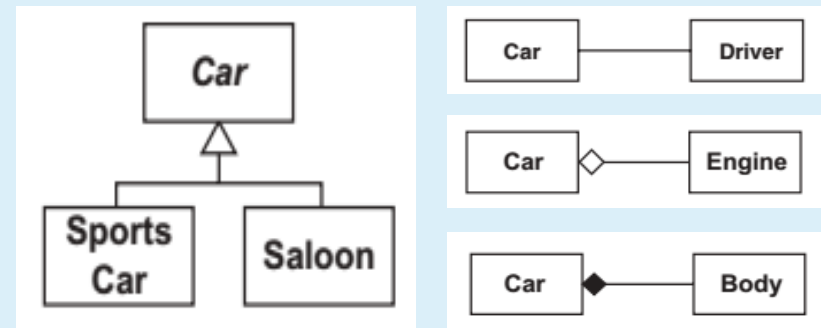
- Các bước thực hiện
  - Tìm và liệt kê các danh từ trong mô tả use case hệ thống.
  - Bổ sung thêm các danh từ (các lớp) phát hiện được nhờ kinh nghiệm
  - Sàng lọc danh sách lớp ứng viên
  - Viết ra các lớp ứng viên sau khi sàng lọc
- Ở bước sàng lọc, loại bỏ những danh từ
  - Nói về bản thân hệ thống: e-Learning, iCoot, ...
  - Các Actor không có thông tin gì đặc biệt
  - Nói về giao diện người dùng hoặc chi tiết xử lý
  - Tầm thường: danh từ chỉ kiểu dữ liệu hoặc có thể trở thành 1 thuộc tính của lớp khác

# Nhận biết mối quan hệ giữa các lớp

---

- Có 4 kiểu quan hệ được dùng

- Kế thừa (inheritance)
- Kết hợp (association)
- Tụ hợp (aggregation)
- Cấu thành (composition)



- Kế thừa mô tả một quan hệ giữa các lớp tại thời điểm biên dịch trong khi các dạng quan hệ còn lại mô tả mối quan hệ giữa các lớp tại thời điểm thực thi.
- Tần suất xuất hiện (trong giai đoạn phân tích):

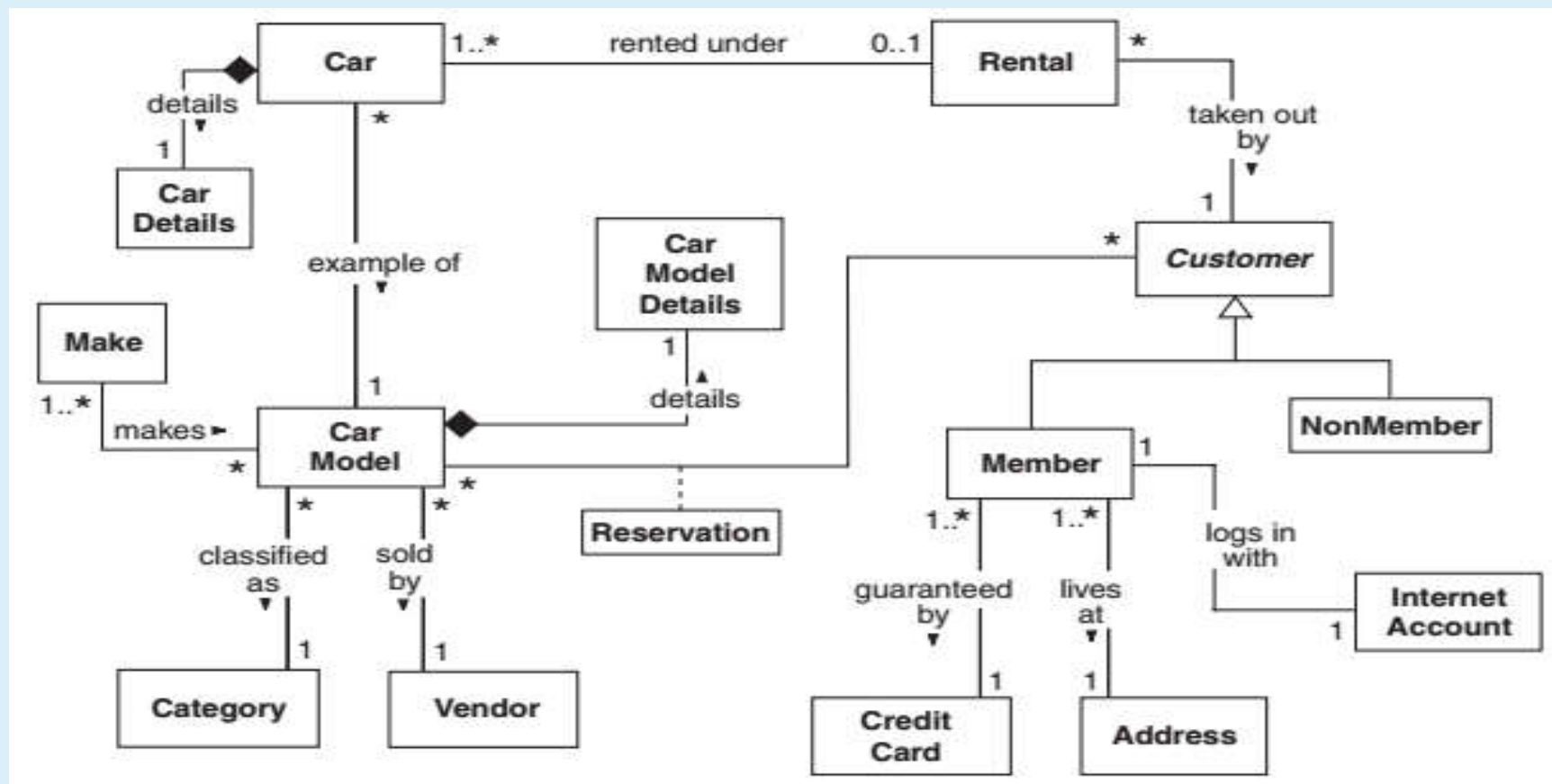
Association > Aggregation > Inheritance > Composition

## Vẽ sơ đồ lớp

---

- Mỗi lớp được biểu diễn bởi một hình chữ nhật, bên trong ghi tên lớp ở dạng in đậm
- Nếu là lớp trừu tượng, tên được viết in nghiêng hoặc được đánh dấu bằng từ khóa {abstract}
- Mỗi quan hệ giữa các lớp được thể hiện bằng các đoạn thẳng kèm theo chú thích.
- Trừ quan hệ kế thừa, các mối quan hệ còn lại thường đi kèm với bản số. Bản số là một hoặc một khoảng giá trị cho phép một đối tượng của một lớp có thể tham gia bao nhiêu lần vào mỗi kết hợp với các đối tượng của lớp khác.
- Bản số có thể ở các dạng sau:
  - n: chính xác n đối tượng tham gia vào mỗi quan hệ.
  - m..n: một số bất kỳ nằm trong đoạn [m..n].
  - p..\*: có ít nhất là p đối tượng tham gia vào mỗi quan hệ.
  - \*: dạng rút gọn của 0..\*
  - 0..1: tùy chọn, có thể có hoặc không có đối tượng tham gia.

## Vẽ sơ đồ lớp (tt)



## Bổ sung thuộc tính

---

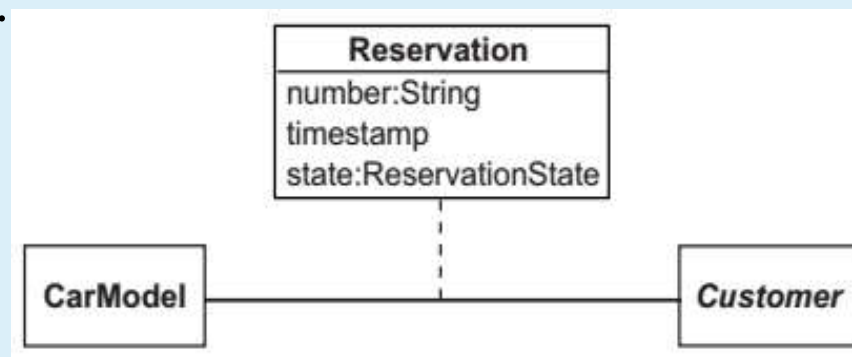
- Một thuộc tính là một đặc điểm của đối tượng
- Mỗi thuộc tính thường đi kèm với kiểu dữ liệu
- Kiểu dữ liệu được thêm vào sau tên thuộc tính và phân cách nhau bởi dấu hai chấm
- Không nên lạm dụng việc đưa kiểu dữ liệu của thuộc tính vào sơ đồ lớp ở giai đoạn phân tích.
- Các thuộc tính dẫn xuất cũng không nên đưa vào sơ đồ lớp.
- Vấn đề khá quan trọng khi bổ sung thuộc tính là lựa chọn giữa việc thêm một thuộc tính hay tạo ra một mối quan hệ.

Engine
capacity
horsePower
manufacturer:String
numberOfCylinders
fuelInjection:boolean

## Bổ sung các lớp kết hợp

---

- Lớp kết hợp được tạo ra từ các lớp có quan hệ nhiều – nhiều.
- Chỉ nên tạo khi một mối kết hợp có những thông tin hoặc thao tác riêng.
- Một lớp kết hợp biểu diễn các thuộc tính và phương thức chỉ tồn tại do có sự tồn tại của mối kết hợp.
- Các thuộc tính và phương thức đó không được đưa vào 2 lớp đầu cuối của mối kết hợp.



# Phân tích khía cạnh động

---

- Lý do
  - Để chắc chắn rằng sơ đồ lớp được thiết kế đầy đủ và chính xác.
  - Để có thể tin tưởng rằng, đến thời điểm này, mô hình có thể được cài đặt vào phần mềm.
  - Để xác thực rằng các chức năng của giao diện người dùng sẽ xuất hiện trong hệ thống cuối cùng.
- Phần quan trọng nhất trong việc phân tích khía cạnh động của hệ thống chính là việc hiện thực hóa (chi tiết hóa) các use case.
- Việc chi tiết hóa use case tuân thủ theo các bước sau:
  - Với mỗi use case hệ thống, mô phỏng các thông điệp được gửi giữa các đối tượng và ghi nhận kết quả bằng các sơ đồ giao tiếp (sơ đồ cộng tác).
  - Bổ sung các thao tác vào các đối tượng nhận thông điệp.
  - Thêm các lớp biểu diễn biên (giao diện) của hệ thống và các lớp điều khiển (nơi chứa các xử lý nghiệp vụ phức tạp hoặc để tạo, nhận đối tượng mới) nếu thấy cần thiết.

## Xây dựng sơ đồ cộng tác (giao tiếp)

---

- Hai loại sơ đồ cho phép mô phỏng các thông điệp được gửi giữa các đối tượng phân tích:
  - Sơ đồ giao tiếp (sơ đồ cộng tác)
  - Sơ đồ tuần tự
- Mặc dù cả hai loại sơ đồ này đều có thể ghi nhận cùng một thông tin nhưng sơ đồ giao tiếp phù hợp hơn cho việc chi tiết hóa use case
- Một sơ đồ giao tiếp có thể cho thấy
  - Các tác nhân tương tác với biên hệ thống (giao diện người dùng).
  - Biên của hệ thống tương tác với các đối tượng khác bên trong hệ thống.
  - Các đối tượng bên trong hệ thống tương tác với biên của các hệ thống bên ngoài.



# Các đối tượng trong sơ đồ giao tiếp

---



- Tác nhân (actor): một người (thông thường) hoặc hệ thống (hiếm khi) nằm bên ngoài hệ thống.



- Biên (boundary): một đối tượng tại cạnh (edge) của hệ thống, nằm giữa hệ thống và tác nhân.

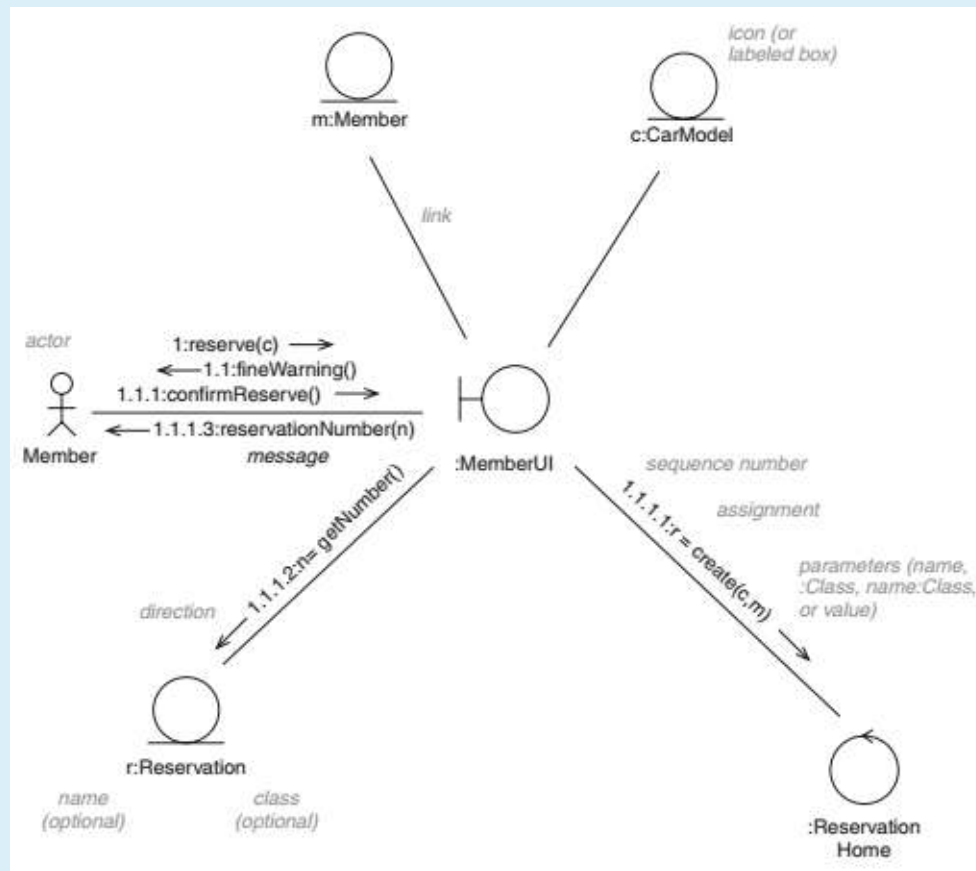


- Thực thể (entity): một đối tượng bên trong hệ thống, biểu diễn một khái niệm nghiệp vụ chứa những thông tin hữu ích.



- Lớp điều khiển (controller): một đối tượng bên trong hệ thống thực hiện việc đóng gói một quy trình phức tạp hoặc logic.

# Các thành phần của sơ đồ giao tiếp



## Các bước tạo sơ đồ giao tiếp

---

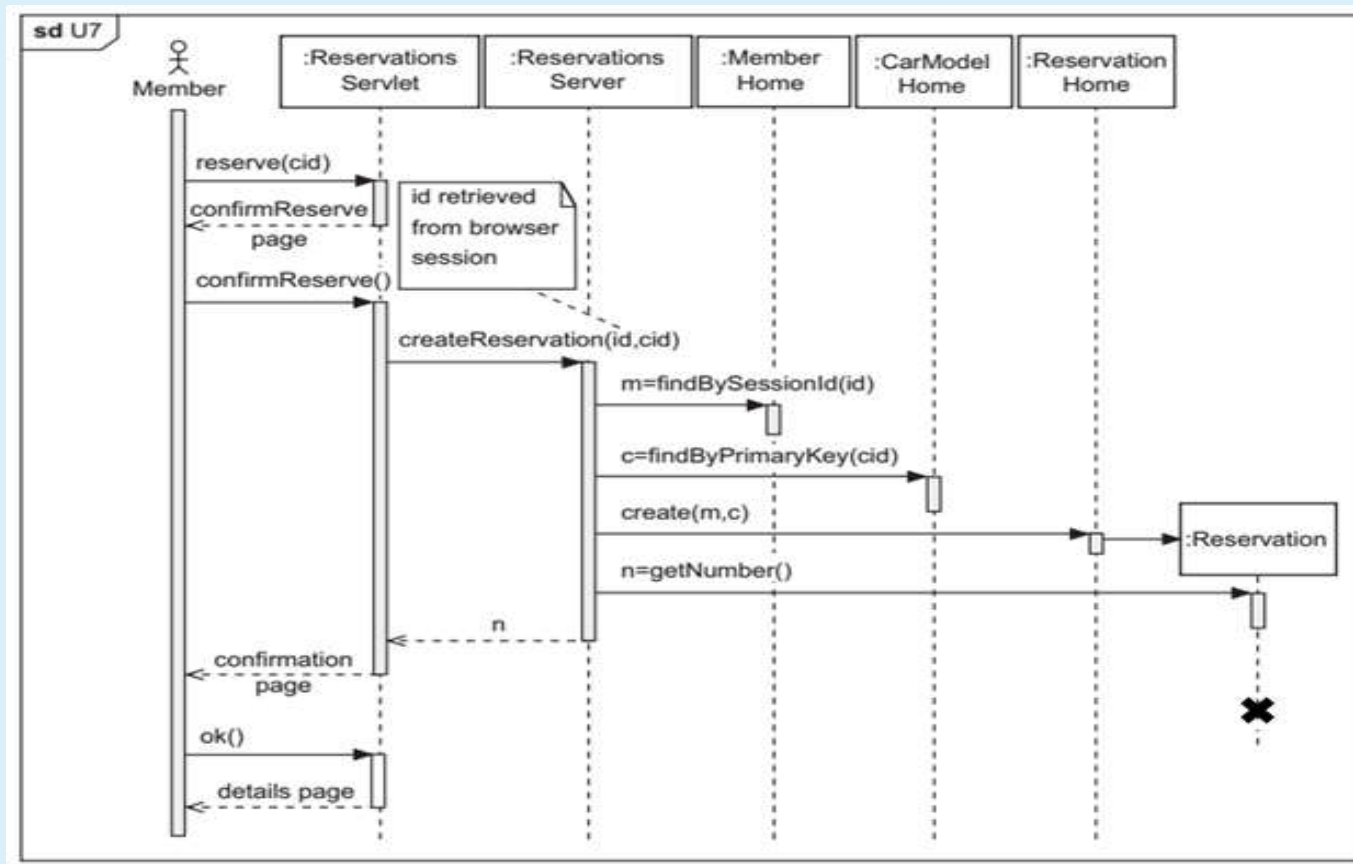
- Xác định ngữ cảnh của sơ đồ giao tiếp
- Xác định các đối tượng, tác nhân và mối kết hợp liên kết giữa các đối tượng (và tác nhân) tham gia vào sự cộng tác động.
- Bố trí các đối tượng (và tác nhân) cùng với mối kết hợp giữa chúng lên sơ đồ giao tiếp
- Bổ sung các thông điệp vào các mối kết hợp giữa các đối tượng
- Đánh giá lại sơ đồ giao tiếp để bảo đảm rằng sơ đồ giao tiếp mô tả đầy đủ các xử lý

# Xây dựng sơ đồ tuần tự

---

- Sơ đồ tuần tự:
  - Là một trong hai loại sơ đồ tương tác.
  - Sơ đồ tuần tự minh họa các đối tượng tham gia vào một use case và các thông điệp được gửi giữa các đối tượng theo thời gian.
  - Đặc biệt hữu ích trong việc hiểu rõ các đặc tả thời gian thực và các use case phức tạp.
- Mỗi sơ đồ tuần tự được dùng cho một use case.
- Được sử dụng xuyên suốt cả hai giai đoạn phân tích và thiết kế

# Các thành phần của sơ đồ tuần tự



## Các nguyên tắc khi vẽ sơ đồ tuần tự

---

- Cố gắng sắp xếp actor và các đối tượng từ trái sang phải và theo thứ tự mà chúng tham gia vào tình huống của use case. Đặt chúng ở trên cùng của sơ đồ.
- Nếu có một actor và một đối tượng giống nhau về mặt khái niệm nhưng một khái niệm ở trong phần phần, khái niệm kia ở thế giới thực thì cũng đặt chúng cùng một tên.
- Đối tượng khởi đầu hoạt động (tình huống) – có thể là actor hoặc object – phải được đặt nằm bên trái nhất của sơ đồ.
- Khi có nhiều đối tượng cùng kiểu tham gia vào cùng một sơ đồ tuần tự, phải thêm tên của đối tượng vào trước dấu hai chấm và tên lớp.
- Chỉ thể hiện các giá trị trả về trên sơ đồ nếu chúng thật sự cần thiết hoặc không phải giá trị hiển nhiên.
- Đặt tên của thông điệp và giá trị trả về nằm gần về phía bên trái của mũi tên.

## Các bước tạo sơ đồ tuần tự

---

- Xác định ngữ cảnh của sơ đồ tuần tự
- Nhận biết các tác nhân và đối tượng tham gia theo trình tự được mô hình hóa.
- Tạo ra đường sống cho mỗi đối tượng
- Thêm các thông điệp vào sơ đồ bằng cách vẽ các mũi tên nằm ngang hướng từ đối tượng này đến đối tượng khác
- Đặt các đoạn thực thi (execution occurrence) nằm chồng lên đường sống của các đối tượng khi chúng gửi và nhận thông điệp.
- Đánh giá sơ đồ tuần tự để đảm bảo rằng nó biểu diễn đầy đủ quy trình công việc hay nói khác hơn là sơ đồ thể hiện được tất cả các bước của quy trình

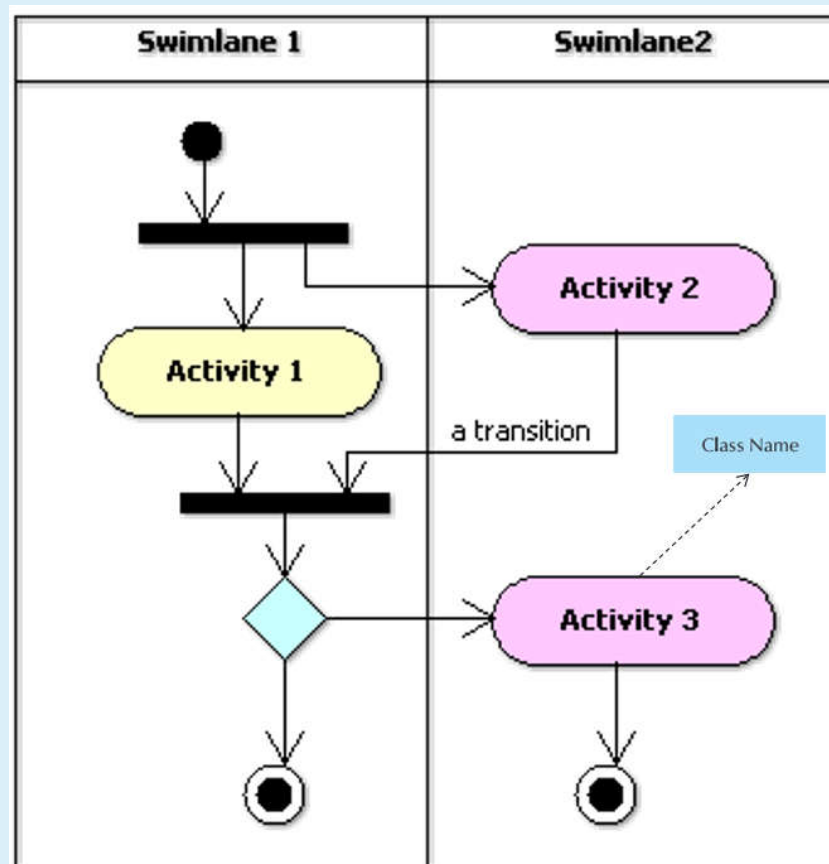
# Xây dựng sơ đồ hoạt động

---

- Sơ đồ hoạt động:
  - Có thể được xem như một sơ đồ dòng dữ liệu phức tạp được dùng cho việc phân tích cấu trúc.
  - Có các ký hiệu để giải quyết việc mô hình hóa các hoạt động xảy ra đồng thời, song song và các quy trình quyết định phức tạp.
  - Cung cấp tài liệu mô tả một thuật toán.
  - Có thể được dùng để mô hình hóa mọi quy trình.
- Các sơ đồ hoạt động mô tả các hoạt động chính và mối quan hệ giữa chúng trong một quy trình.



# Các thành phần của sơ đồ hoạt động



## Một số lưu ý khi vẽ sơ đồ hoạt động

---

- Chỉ vẽ một nút khởi đầu và đặt nó ở phía trên hoặc góc trên bên trái
- Với hầu hết các quy trình nghiệp vụ, chỉ có duy nhất một nút kết thúc hoạt động, được đặt ở phía dưới cùng hoặc góc dưới bên phải của sơ đồ.
- Vì các quy trình nghiệp vụ mức cao có tính tuần tự, không chạy song song nên cần hạn chế sử dụng nút kết thúc dòng.
- Khi mô hình hóa các quy trình nghiệp vụ mức cao hoặc dòng công việc, chỉ những quyết định quan trọng mới được thể hiện trong sơ đồ.
- Hạn chế tối thiểu các đường cắt nhau để đảm bảo sơ đồ rõ ràng, dễ đọc.
- Các hoạt động nên sắp xếp theo thứ tự từ trái qua phải, từ trên xuống dưới theo trình tự mà chúng được thực hiện.
- Các phân tuyến chỉ nên dùng khi muốn làm tăng tính rõ ràng, dễ đọc cho sơ đồ hoạt động.

## Các bước tạo sơ đồ hoạt động

---

- Chọn một quy trình xử lý đã xác định rõ để mô hình hóa
- Xác định tập các hoạt động cần thiết để hỗ trợ cho quy trình xử lý.
- Xác định các dòng điều khiển và các nút cần thiết để dẫn chứng sự logic của quy trình xử lý
- Xác định các dòng đối tượng và các nút cần thiết hỗ trợ cho sự logic của quy trình xử lý
- Bố trí các thành phần và vẽ sơ đồ hoạt động để cung cấp tài liệu về quy trình

# Kết chương

---

- Một số kiến thức cơ bản cần nắm:
  - Phân tích yêu cầu phần mềm là gì? Tại sao giai đoạn phân tích lại quan trọng?
  - Tổng quan về quá trình phân tích yêu cầu.
  - Cách thức xây dựng một mô hình phân tích khía cạnh tĩnh của hệ thống nhằm chỉ ra các đối tượng liên quan đến nghiệp vụ trong hệ thống đề xuất.
  - Cách xây dựng một sơ đồ lớp để thể hiện được các lớp đối tượng kém theo thuộc tính, mối quan hệ giữa chúng.
  - Phân tích khía cạnh động của hệ thống nhằm cải tiến và xác thực mô hình tĩnh bằng cách dùng các sơ đồ tương tác.
  - Xây dựng sơ đồ hoạt động để hiểu rõ hơn quy trình nghiệp vụ.

# THẢO LUẬN





Question?