



# Phân Tích và Thiết Kế Hướng đối tượng Sử dụng UML

cuu duong than cong . com

## Tổng Quan Về Xác định Yêu Cầu Người Dùng

cuu duong than cong . com

# Mục tiêu : Tổng quan về xác định yêu cầu

- ✍️ Tìm hiểu các khái niệm cơ bản về xác định yêu cầu người dùng và tác dụng của chúng lên Phân tích và Thiết kế
- ✍️ Tìm hiểu cách ghi nhận và diễn dịch các yêu cầu của người dùng, là những thông tin được dùng để bắt đầu việc phân tích và thiết kế

cuu duong than cong . com

# Các chủ đề

---

## ★ ✍ Giới thiệu

✍ Các khái niệm chính

✍ Phát biểu bài toán

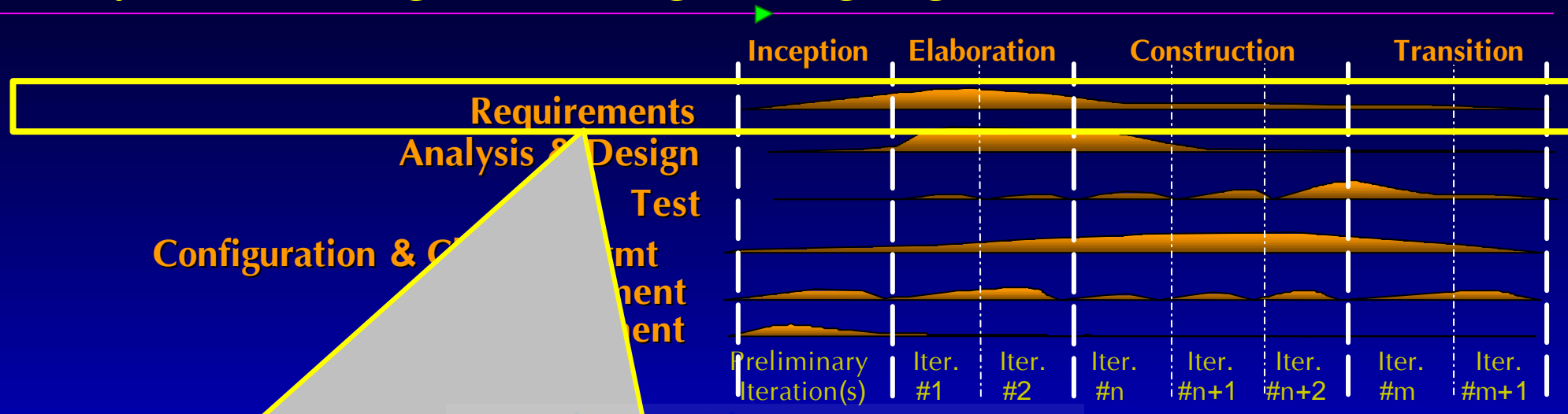
✍ Bảng chú giải

✍ Use-Case Model

✍ Các đặc tả bổ sung

✍ Checkpoints

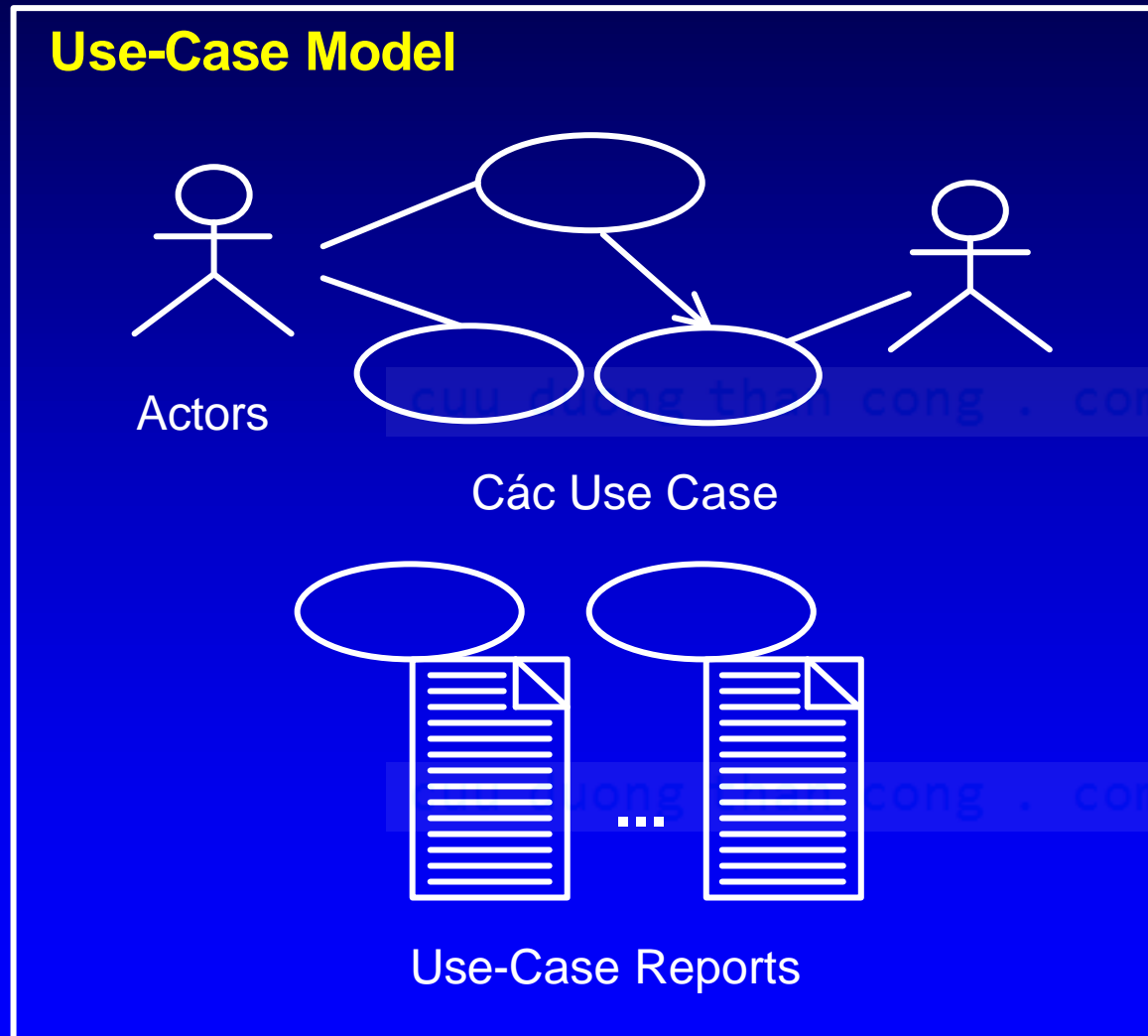
# Các yêu cầu người dùng trong ngữ cảnh



Mục đích của bước xác định y/c người dùng là:

- ? Đi đến thỏa thuận với khách hàng và người dùng về các chức năng của hệ thống (những gì hệ thống phải thực hiện).
- ? Cho phép các system developer hiểu rõ hơn các yêu cầu đối với hệ thống.
- ? Phân định ranh giới của hệ thống.
- ? Cung cấp cơ sở để hoạch định nội dung kỹ thuật của các vòng lặp.
- ? Xác định giao diện người dùng cho hệ thống.

# Các dạng thông tin về yêu cầu người dùng




B?ng chú gì?



Các d?c t? b? sung

# Các chủ đề

---

 Giới thiệu

★  Các khái niệm chính

 Phát biểu bài toán

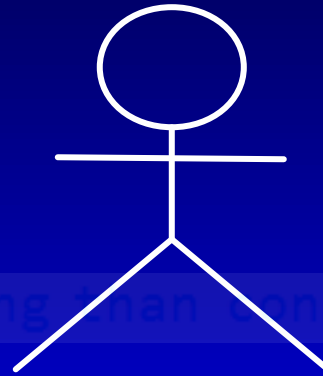
 Bảng chú giải

 Use-Case Model

 Các đặc tả bổ sung

 Checkpoints

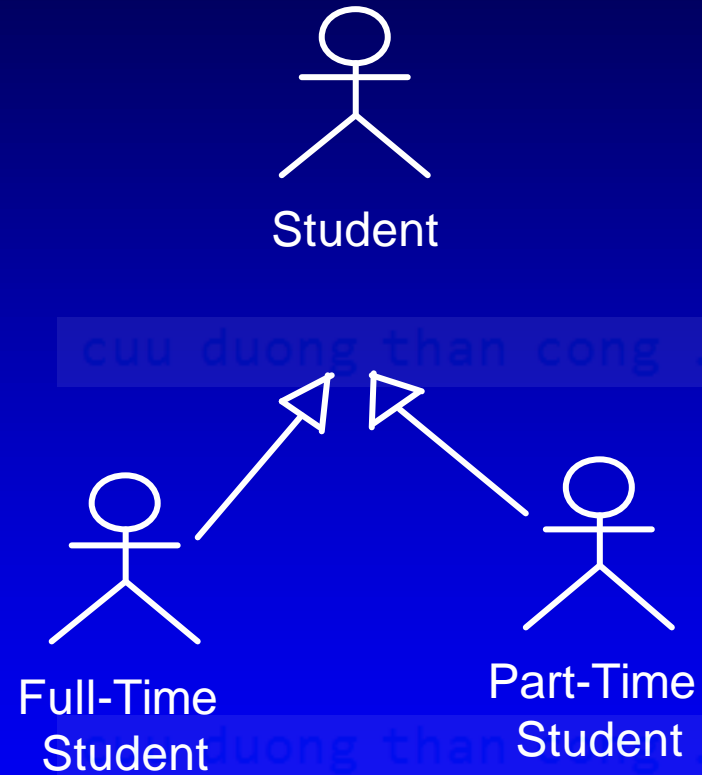
# Khái niệm trong Use-Case Modeling: Actor



**Actor (Tác nhân)**

***Các Actor nằm BÊN NGOÀI hệ thống***

# Actor Generalization



# Một User có thể có nhiều Vai trò (Role)



Charlie

Charlie như  
một sinh viên



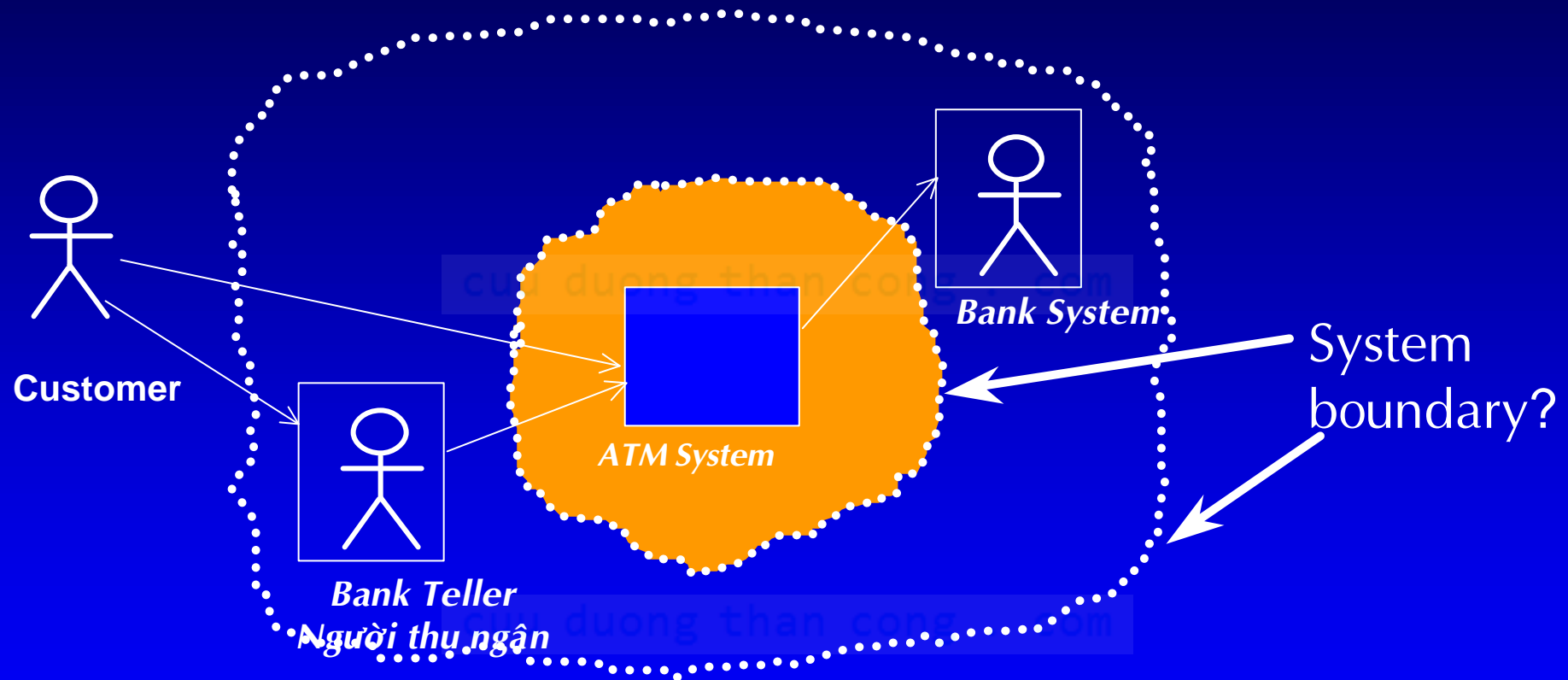
Student

Charlie như  
một giáo sư



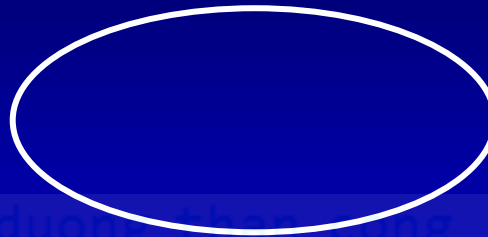
Professor

# Actors và giới hạn hệ thống(System Boundary)



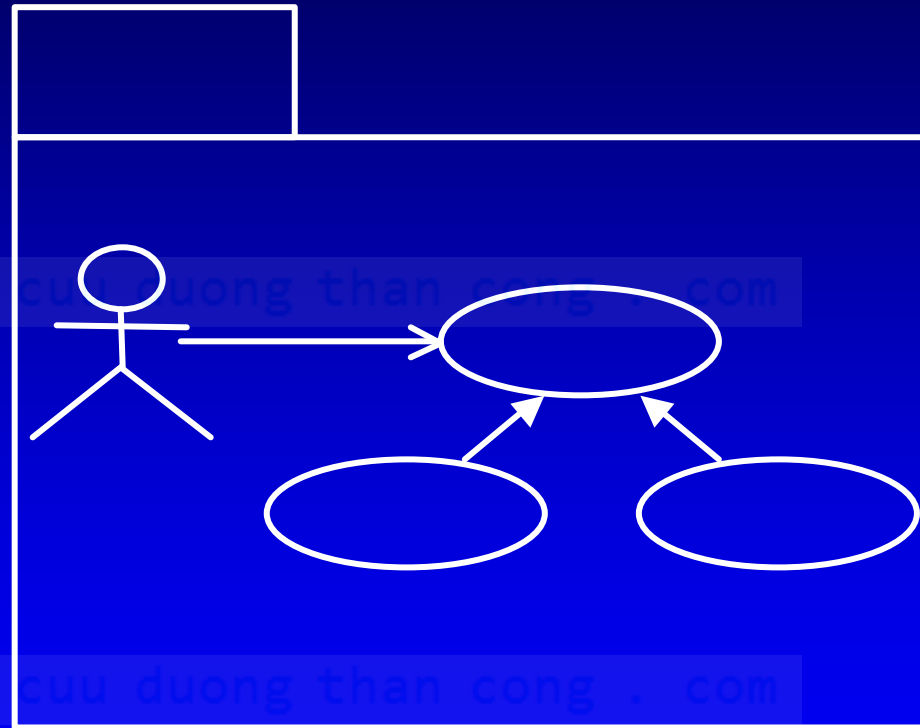
# Khái niệm trong Use-Case Modeling : Use-Case

---




**Use-Case**

# Các Package trong Use-Case Model



# Các chủ đề

---

 Giới thiệu

 Các khái niệm chính

★  Phát biểu bài toán (Problem Statement)

 Bảng chú giải

 Use-Case Model

 Các đặc tả bổ sung

 Checkpoints

# Ví dụ: Course Registration Problem Statement

- ✍ Là trưởng ban IT của trường Đại học KHTN, bạn được yêu cầu phát triển một hệ thống đăng ký học phần mới. Hệ thống mới cho phép sinh viên đăng ký học phần và xem phiếu điểm từ một máy tính cá nhân được kết nối vào mạng nội bộ của trường. Các giáo sư cũng có thể truy cập hệ thống này để đăng ký lớp dạy và nhập điểm cho các môn học.
- ✍ Do kinh phí bị giảm nên trường không đủ khả năng thay đổi toàn bộ hệ thống trong cùng một lúc. Trường sẽ giữ lại cơ sở dữ liệu (CSDL) sẵn có về danh mục học phần mà trong đó lưu trữ toàn bộ thông tin về học phần. Đây là một CSDL quan hệ và có thể truy cập bằng các câu lệnh SQL thông qua các server của trường. Hiệu suất của hệ thống cũ này rất kém nên hệ thống mới phải bảo đảm truy cập dữ liệu trên hệ thống cũ một cách hợp lý hơn. Hệ thống mới sẽ đọc các thông tin học phần trên CSDL cũ nhưng sẽ không cập nhật chúng. Phòng Đào tạo sẽ tiếp tục duy trì các thông tin học phần thông qua một hệ thống khác.
- ✍ Ở đầu mỗi học kỳ, sinh viên có thể yêu cầu danh sách các học phần được mở trong học kỳ đó. Thông tin về mỗi học phần, ví dụ như là tên giáo sư, khoa, và các môn học phần tiên quyết sẽ được cung cấp để giúp sinh viên chọn lựa.
- ✍ Hệ thống mới cho phép sinh viên chọn bốn học phần được mở trong học kỳ tới. Thêm vào đó mỗi sinh viên có thể đưa ra hai môn học thay thế trong trường hợp không thể đăng ký theo nguyện vọng chính. Các học phần được mở có tối đa là 100 và tối thiểu là 30 sinh viên. Các học phần có ít hơn 30 sinh viên sẽ bị hủy. Đầu mỗi học kỳ, sinh viên có một khoảng thời gian để thay đổi các học phần đã đăng ký. Sinh viên chỉ có thể thêm hoặc hủy học phần đã đăng ký trong khoảng thời gian này. Khi quá trình đăng ký hoàn tất cho một sinh viên, hệ thống đăng ký sẽ gửi thông tin tới hệ thống thanh toán (billing system) để sinh viên có thể đóng học phí. Nếu một lớp bị hết chỗ trong quá trình đăng ký, sinh viên sẽ được thông báo về sự thay đổi trước khi xác nhận việc đăng ký học phần.
- ✍ Ở cuối học kỳ, sinh viên có thể truy cập vào hệ thống để xem phiếu điểm. Bởi vì thông tin về điểm của mỗi sinh viên cần được giữ kín, nên hệ thống cần có cơ chế bảo mật để ngăn chặn những truy cập không hợp lệ.
- ✍ Các giáo sư có thể truy cập vào hệ thống để đăng ký những học phần mà họ sẽ dạy. Họ có thể xem danh sách các sinh viên đã đăng ký vào lớp của họ, cũng như nhập điểm sau mỗi khóa học.

# Các chủ đề

---

 Giới thiệu

 Các khái niệm chính

 Phát biểu bài toán

★  Bảng chú giải

 Use-Case Model

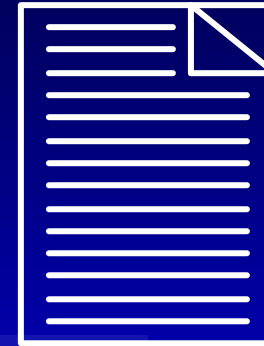
 Các đặc tả bổ sung

 Checkpoints

# Từ điển thuật ngữ (Glossary)

 Giới thiệu













 Bảng chú giải



Glossary

# Ví dụ: Glossary

## Bảng chú giải của ứng dụng Course Registration:

-  **Course (Học phần):** Một môn học được dạy trong trường.
-  **Course Offering (Lớp):** Một lớp học cụ thể được mở trong một học kỳ cụ thể cùng một học phần có thể được mở song song nhiều lớp trong một học kỳ. Thông tin gồm cả ngày học trong tuần và giờ học.
-  **Course Catalog (Danh mục học phần):** Danh mục đầy đủ của tất cả các học phần được dạy trong trường.
-  **Faculty:** Toàn bộ cán bộ giảng dạy của trường..
-  **Finance System (Hệ thống thanh toán):** Hệ thống dùng để xử lý các thông tin thanh toán học phí.
-  **Grade (Điểm số):** Sự đánh giá cho một sinh viên cụ thể trong một lớp cụ thể.
-  **Professor (Giáo sư):** Người giảng dạy trong trường.
-  **Report Card (Phiếu điểm):** Toàn bộ điểm số cho tất cả học phần một sinh viên đã học trong một học kỳ xác định.
-  **Roster (Danh sách sinh viên đăng ký):** Tất cả sinh viên đăng ký vào một lớp học cụ thể.
-  **Student (Sinh viên):** Người đăng ký vào học các lớp của trường.
-  **Schedule (Lịch học):** Các học phần mà một sinh viên đã chọn học trong học kỳ hiện tại.
-  **Transcript (Bản sao học bạ):** Bản sao tất cả điểm số cho tất cả các học phần của một sinh viên cụ thể được chuyển cho hệ thống thanh toán để hệ thống này lập hóa đơn cho sinh viên.

# Các chủ đề

---

✍ Giới thiệu

✍ Các khái niệm chính

✍ Phát biểu bài toán

✍ Bảng chú giải

★ ✍ Use-Case Model

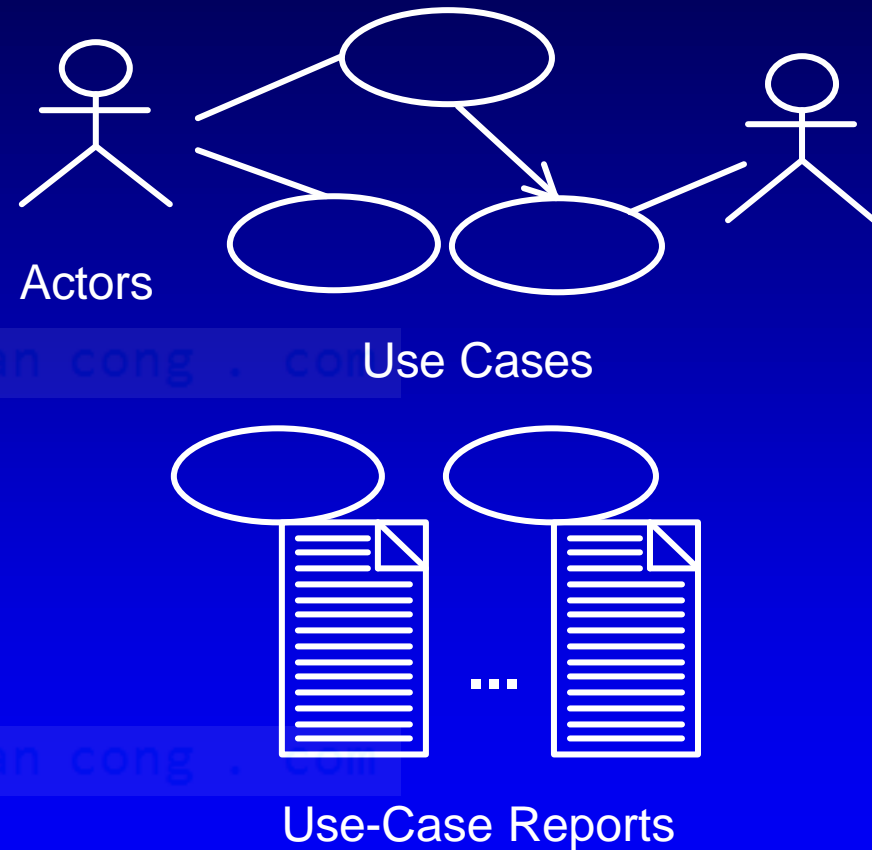
✍ Các đặc tả bổ sung

✍ Checkpoints

# Use-Case Model

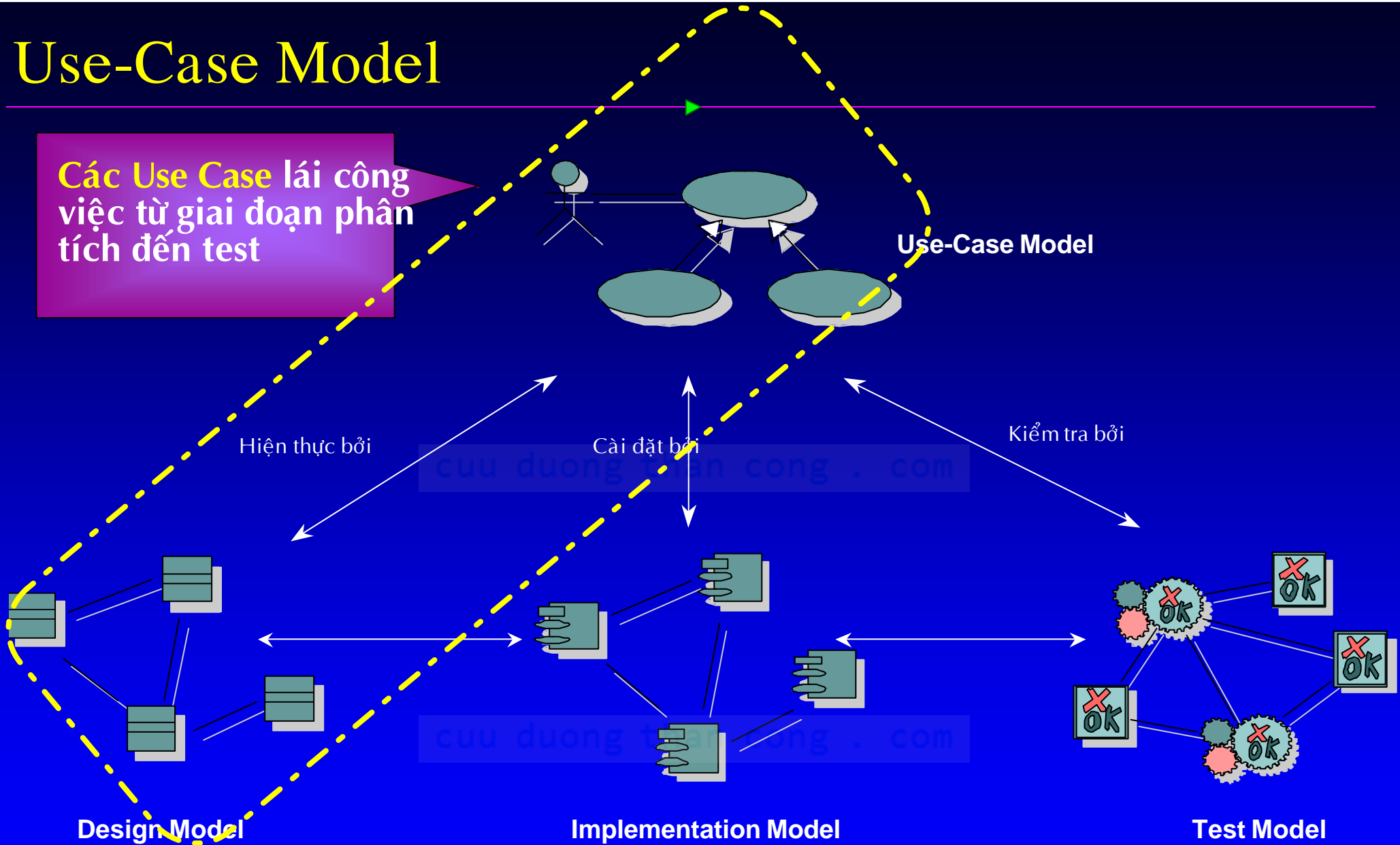
- ✍ Giới thiệu
- ✍ Survey Description
- ✍ Use-Case Packages
- ✍ Use Cases
- ✍ Actors
- ✍ Relationships
- ✍ Diagrams
- ✍ Use-Case View

## Use-Case Model

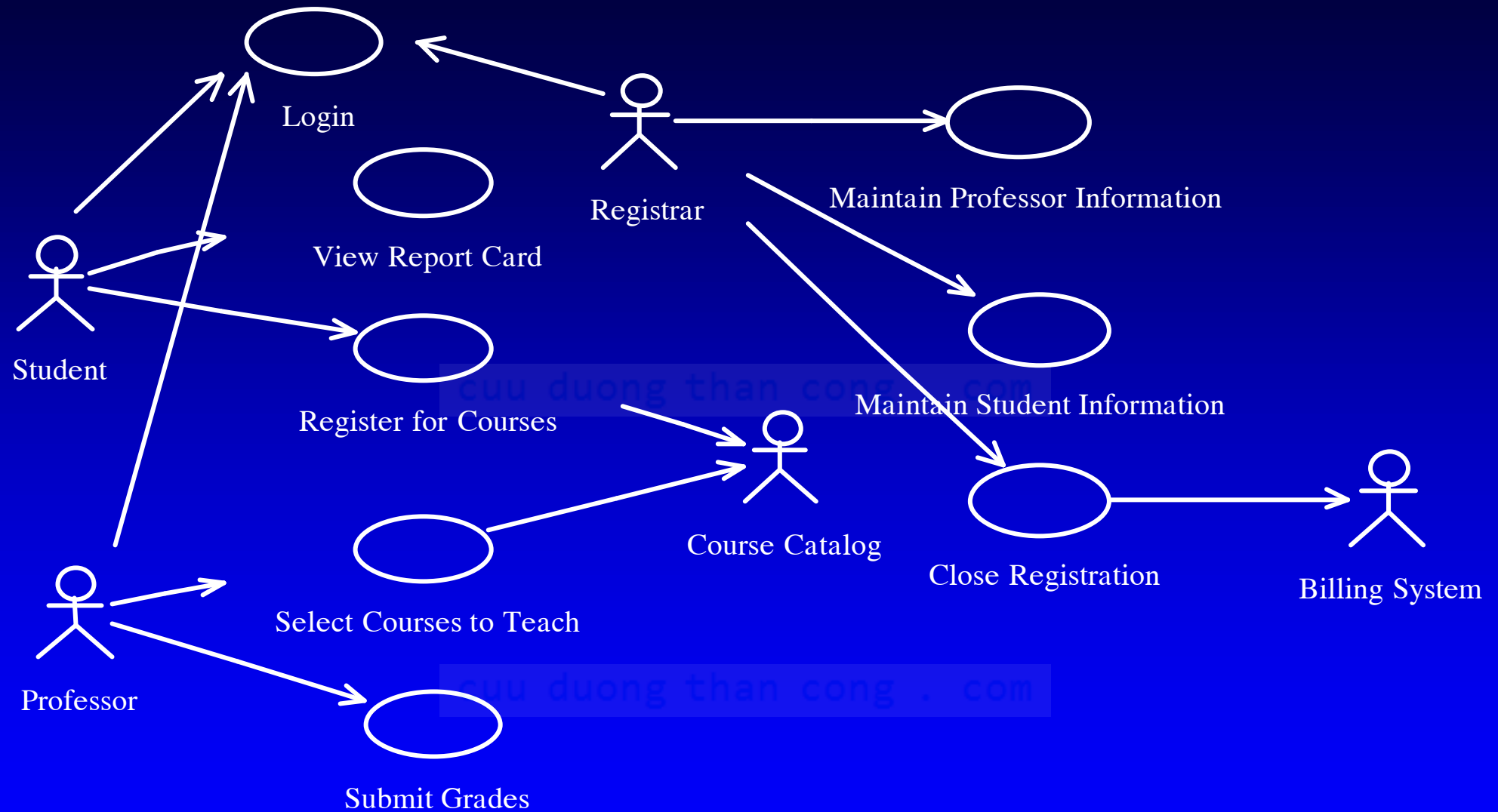


# Use-Case Model

Các Use Case lái công việc từ giai đoạn phân tích đến test

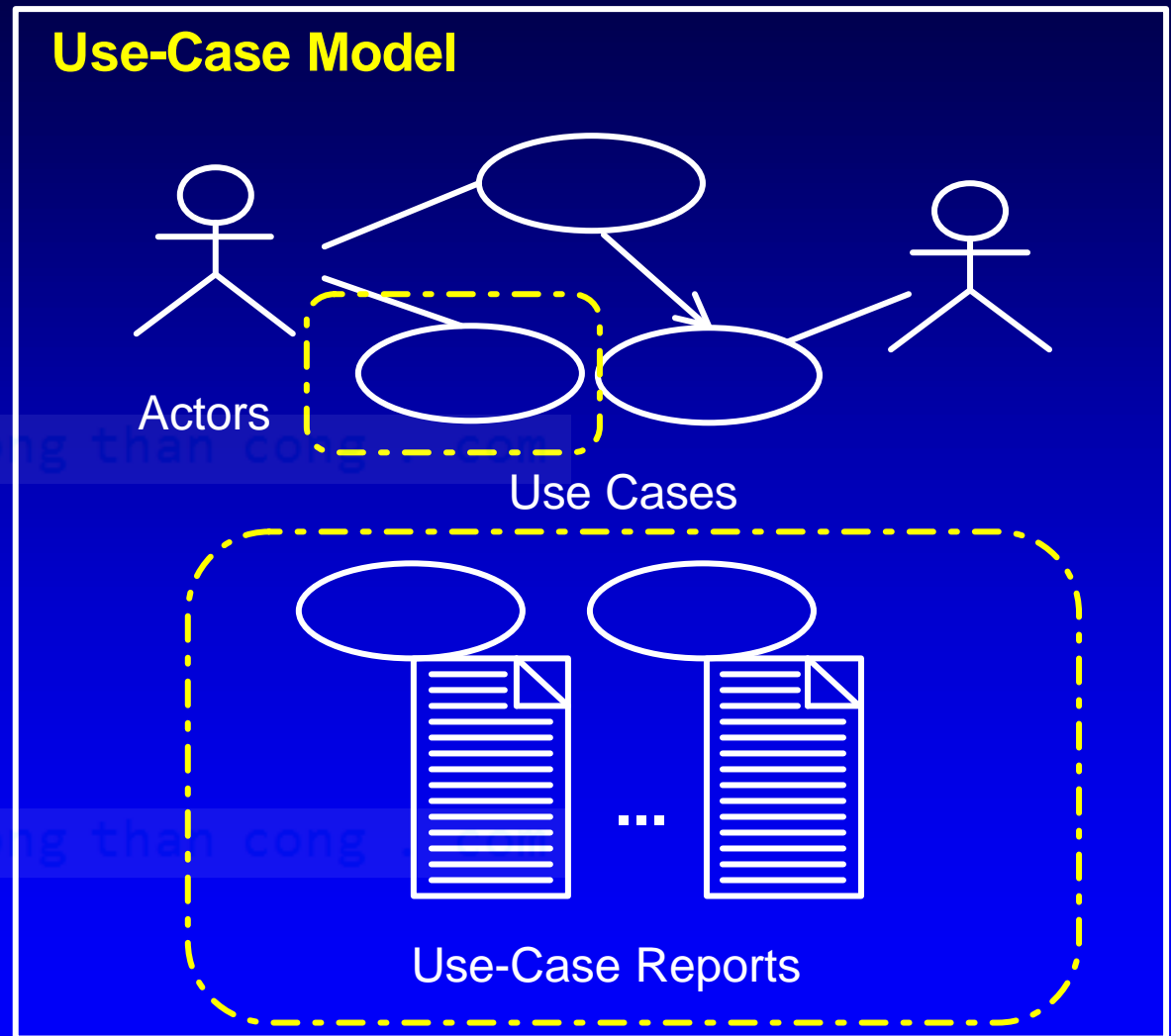


# Ví dụ: Use-Case Model: Use-Case Diagram



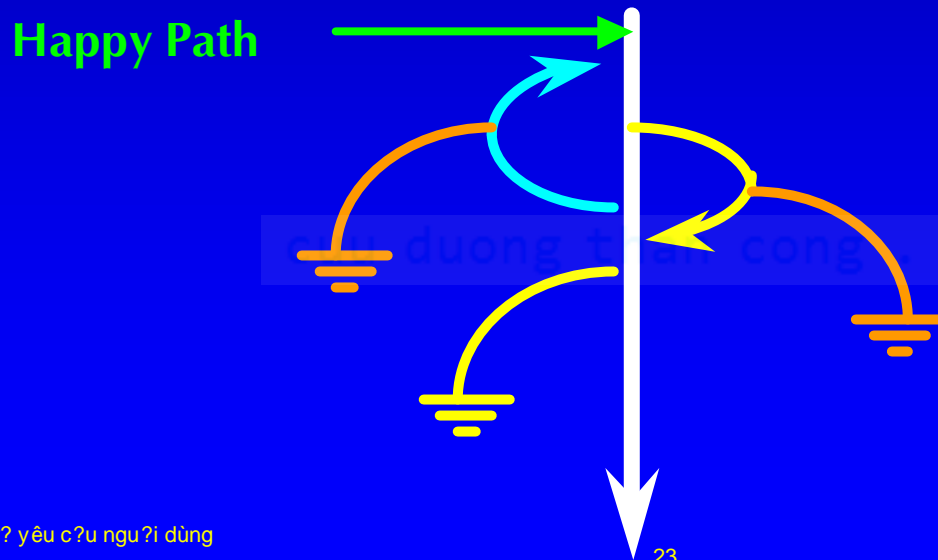
# Use Case

- ✍ Tên
- ✍ Brief description
- ✍ Luồng các sự kiện
- ✍ Relationships
- ✍ Activity và State diagrams
- ✍ Use-Case diagrams
- ✍ Special requirements
- ✍ Preconditions
- ✍ Postconditions
- ✍ Các diagram khác



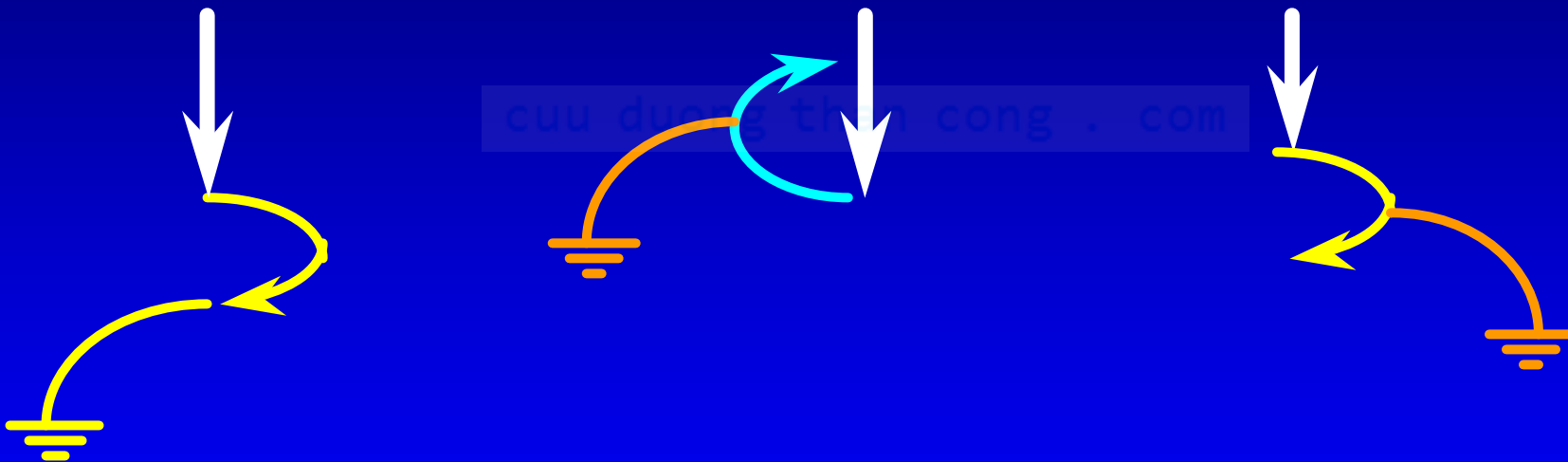
## Luồng các sự kiện (Use-Case Flows of Events)

- ✍ Có một *basic flow* (Happy Path)
- ✍ Một số *alternative flows*
  - ✍ Các biến thể thường gặp (Regular variants)
  - ✍ Các trường hợp bất thường (Odd cases)
  - ✍ **Exceptional flows** xử lý các tình huống lỗi

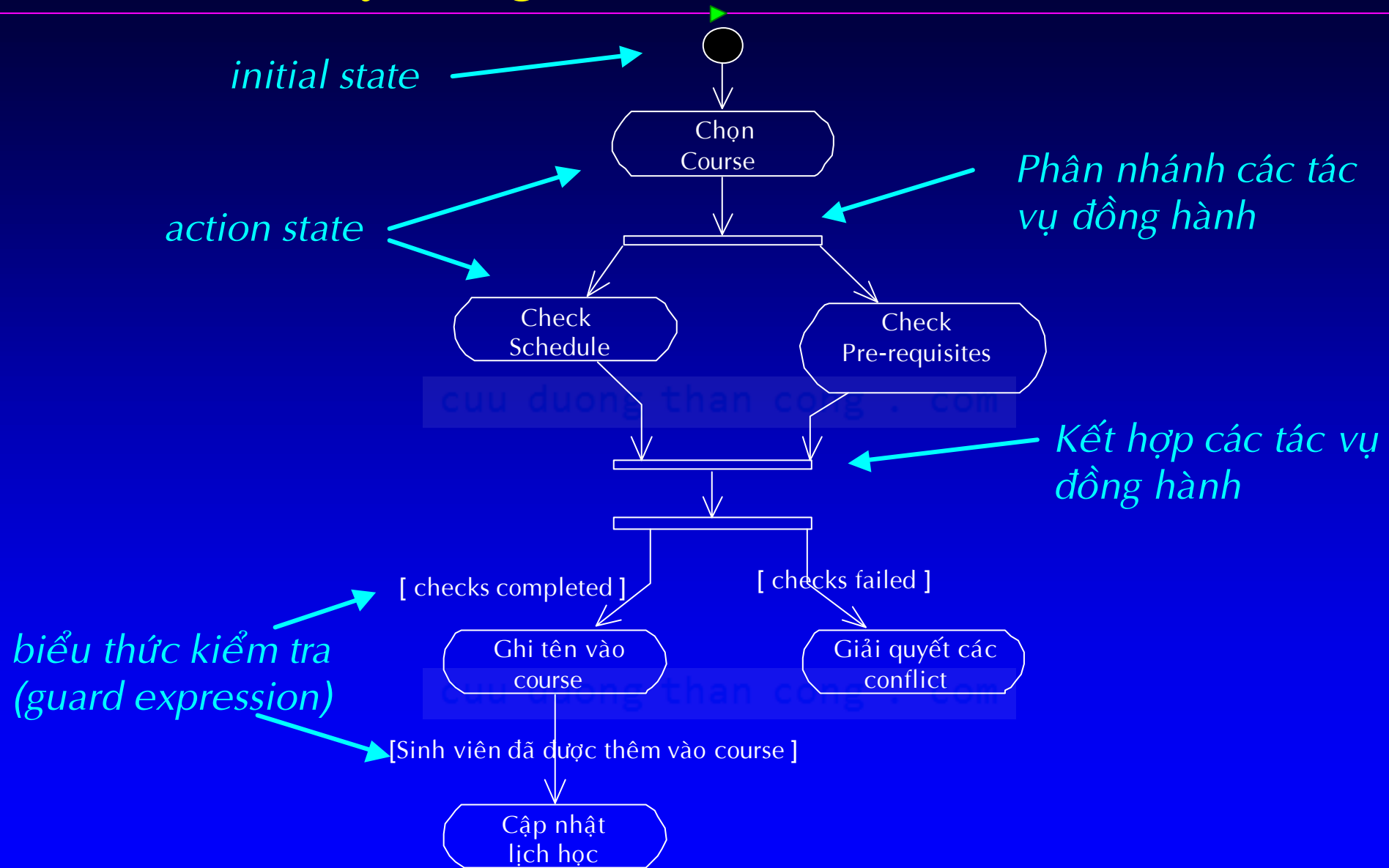


# Scenarios là gì ?

✍ Scenario là một thể hiện của use case



# Ví dụ: Activity Diagram



# Ví dụ: Đặc tả Use-Case

✍️ Điểm lại đặc tả của một use-case hoàn chỉnh được cung cấp trong tài liệu mô tả các yêu cầu của ứng dụng Course Registration

cuu duong than cong . com

cuu duong than cong . com

# Các chủ đề

---

 Giới thiệu

 Các khái niệm chính

 Phát biểu bài toán

 Bảng chú giải

 Use-Case Model

★  Các đặc tả bổ sung

 Checkpoints

# Các đặc tả bổ sung

- ✍ Functionality
- ✍ Tính khả dụng (Usability)
- ✍ Tính tin cậy (Reliability)
- ✍ Tính hiệu năng (Performance)
- ✍ Tính hỗ trợ (Supportability)
- ✍ Các ràng buộc thiết kế



Supplementary  
Specification

# Ví dụ: Các đặc tả bổ sung

## Tài liệu tham khảo

-  Không có.

## Chức năng

-  Hỗ trợ nhiều người dùng làm việc đồng thời.

-  Nếu một lớp bị hết chỗ khi một sinh viên đang đăng ký học có lớp đó thì sinh viên này phải được thông báo.

## Tính khả dụng

-  Giao diện người dùng tương thích Windows 95/98.

## Tính ổn định

-  Hệ thống phải hoạt động liên tục 24 giờ/ngày, 7 ngày/tuần, với thời gian ngưng hoạt động không quá 10%.

## Hiệu suất

-  Hệ thống phải hỗ trợ đến 2000 người dùng truy xuất CSDL trung tâm đồng thời bất kỳ lúc nào, và đến 500 người dùng truy xuất các server cục bộ.

-  Hệ thống phải cho phép truy xuất đến CSDL danh mục học phần cũ với độ trễ không quá 10 giây.

-  Hệ thống phải có khả năng hoàn tất 80% giao dịch trong vòng 2 phút.

## Sự hỗ trợ

-  Không có.

## Tính bảo mật

-  Hệ thống phải ngăn chặn sinh viên thay đổi lịch học của người khác, và ngăn các giáo sư thay đổi lớp dạy của giáo sư khác.

-  Chỉ có giáo sư mới có thể nhập điểm cho sinh viên.

-  Chỉ có cán bộ đào tạo mới được phép thay đổi thông tin của sinh viên.

## Các ràng buộc thiết kế

-  Hệ thống phải tích hợp với hệ thống có sẵn, Hệ thống danh mục học phần, một CSDL RDBMS.

-  Hệ thống phải cung cấp giao diện dựa trên Windows.

# Các chủ đề

---

✍ Giới thiệu

✍ Các khái niệm chính

✍ Phát biểu bài toán

✍ Bảng chú giải

✍ Use-Case Model

✍ Các đặc tả bổ sung

★ ✍ Checkpoints

# Checkpoints: Requirements: Use-Case Model

- ✍ Use-case model có dễ hiểu không?
- ✍ Sau khi nghiên cứu use-case model, bạn có hình thành được một ý tưởng rõ ràng về các chức năng của hệ thống và cách thức mà chúng liên hệ với nhau?
- ✍ Đã xác định hết tất cả các actor? Tất cả các yêu cầu chức năng được thỏa?
- ✍ Use-case model có chứa các hành vi vô dụng nào không?
- ✍ Việc chia model thành các use-case package có xác đáng?

# Checkpoints: Requirements: Actors

- ✍ Đã xác định hết tất cả các actor?
- ✍ Mỗi actor có tham gia vào ít nhất một use case?
- ✍ Mỗi actor thật sự có một vai trò (role)? Có cần ghép hoặc tách các actor không?
- ✍ Có tồn tại 2 actor đóng cùng một vai trò đối với 1 use case không?
- ✍ Tên của các actor có gợi nhớ không? Users và customers có hiểu tên của chúng ?

cuu duong than cong . com

# Checkpoints: Requirements: Use-Cases

- ✍ Mỗi use case có ít nhất một actor tương tác?
- ✍ Các use case có độc lập với nhau?
- ✍ Tồn tại các use case có các luồng sự kiện và các hành vi tương tự nhau không?
- ✍ Liệu các use case có tên duy nhất, gợi nhớ, và dễ hiểu để chúng không bị nhầm lẫn trong các giai đoạn sau?
- ✍ Các customer và user có hiểu tên và mô tả của các use case không?

# Checkpoints: Requirements: Các đặc tả Use-Case

- ✍ Is it clear who wishes to perform a use-case?
- ✍ Mục đích của use-case có rõ ràng?
- ✍ Brief description có cho ta hình ảnh trung thực của use-case?
- ✍ Có xác định rõ luồng sự kiện của use-case như thế nào và khi nào bắt đầu và kết thúc?
- ✍ Chuỗi các giao tiếp giữa actor và use-case có tiện nghi không (từ góc nhìn của user)?
- ✍ Các tương tác và các thông tin trao đổi của actor có rõ ràng?
- ✍ Có use-case nào quá phức tạp không?
- ✍ Các luồng sự kiện (basic và alternative) được mô hình đúng đắn?

# Checkpoints: Requirements: Glossary

---

- ✍ Các thuật ngữ có định nghĩa rõ ràng và súc tích?
- ✍ Mỗi thuật ngữ có dùng đâu đó trong các mô tả use-case?
- ✍ Các thuật ngữ có được sử dụng hợp lý trong các mô tả ngắn về các actor và use case?

cuu duong than cong . com

# Review:

---

- ✍ Các thành phần chính của Requirements?
- ✍ Các thành phần của Requirements được dùng để làm gì?
- ✍ Use-case model là gì?
- ✍ Actor là gì?
- ✍ Use case là gì? Liệt kê một số thuộc tính của use case.
- ✍ Sự khác nhau giữa use-case và scenario?
- ✍ Supplementary specification là gì và nó chứa những gì?
- ✍ Glossary là gì và nó chứa những gì?

# Bài tập:

---

✍ Mô tả các thành phần chính của Requirements:

✍ Problem statement

✍ Use-case model main diagram

✍ Supplementary specification

✍ Glossary

✍ Điềm lại các thông tin về y/c người dùng được sử dụng trong mô hình, ghi chú tất cả các câu hỏi, các vấn đề còn tranh cãi, mâu thuẫn