

Bài tập tuần 05

Phân tích usecase : Trích xuất đối tượng và thiết kế chuỗi thông điệp giữa các đối tượng

Mục tiêu

- Mục tiêu của bước này là tìm các đối tượng / lớp phân tích cho từng ca sử dụng sau đó thiết kế chuỗi thông điệp giữa chúng để xây dựng các sơ đồ tương tác. Chúng ta xây dựng biểu đồ tuần tự để mô hình hóa các hành vi, truyền thông điệp giữa các đối tượng.
- 1. Trích xuất các lớp phân tích
- 2. Xây dựng sơ đồ trình tự

Đánh giá

- Hoàn thành xác định các lớp phân tích và xây dựng sơ đồ trình tự cho các usecase

Phần I: Trích xuất các lớp phân tích (Analysis classes)

Các lớp phân tích đại diện cho một mô hình khái niệm ban đầu cho “những thứ trong hệ thống có trách nhiệm và hành vi”. Các lớp phân tích xử lý các yêu cầu chức năng chủ yếu. Chúng mô hình hóa các đối tượng từ miền vấn đề. Các lớp phân tích có thể được sử dụng để đại diện cho "các đối tượng mà chúng ta muốn hệ thống hỗ trợ" mà không cần đưa ra quyết định về mức độ hỗ trợ của chúng bao nhiêu với phần cứng và bao nhiêu với phần mềm.

Ba khía cạnh của hệ thống có thể sẽ thay đổi:

1. Ranh giới giữa hệ thống và các tác nhân của nó

2. Thông tin hệ thống sử dụng

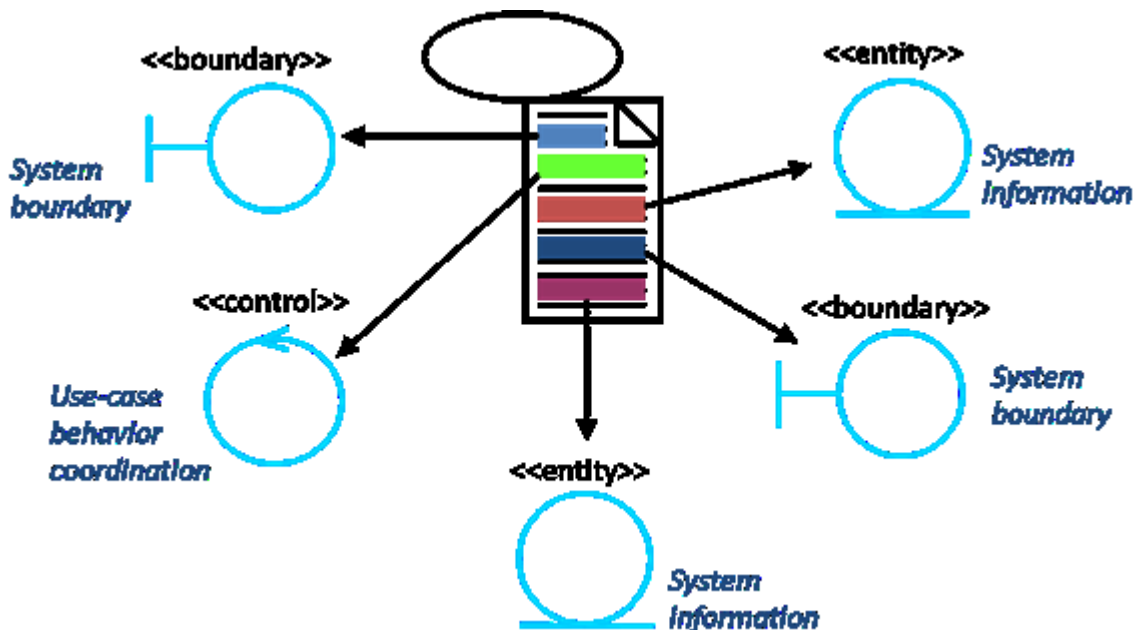
3. Logic điều khiển của hệ thống

Trong nỗ lực cô lập các bộ phận của hệ thống sẽ thay đổi, các loại lớp phân tích sau được xác định với một nhóm trách nhiệm “đóng hộp”:

- Ranh giới
- Thực thể
- Điều khiển

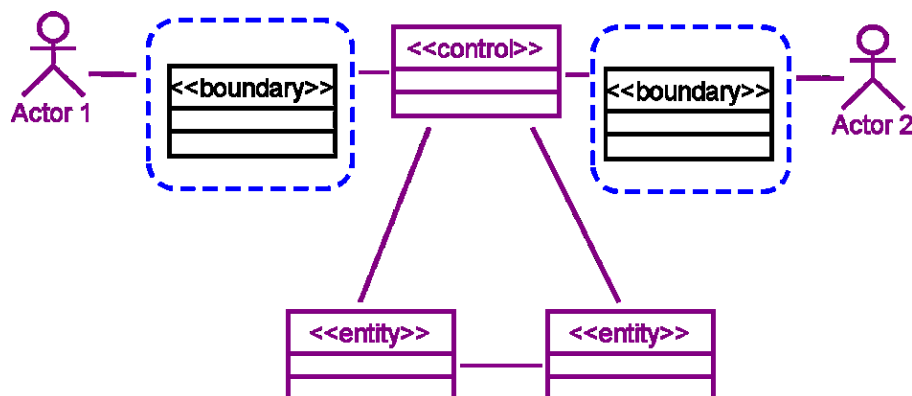
Khuôn mẫu có thể được xác định cho từng loại. Những điểm khác biệt này được sử dụng trong giai đoạn Phân tích, và có thể biến mất trong giai đoạn Thiết kế. Các loại lớp

phân tích khác nhau có thể được biểu diễn bằng các biểu tượng khác nhau hoặc với tên của khuôn mẫu trong cặp dấu (<< >>): <<boundary>>, <<control>>, <<entity>>.



Boundary classes

- Một lớp biên là **trung gian giữa giao diện và một cái gì đó bên ngoài hệ thống**. Các lớp ranh giới cách ly hệ thống khỏi những thay đổi của môi trường xung quanh (ví dụ, những thay đổi về giao diện đối với các hệ thống khác và những thay đổi về yêu cầu của người dùng), giữ cho những thay đổi này không ảnh hưởng đến phần còn lại của hệ thống.

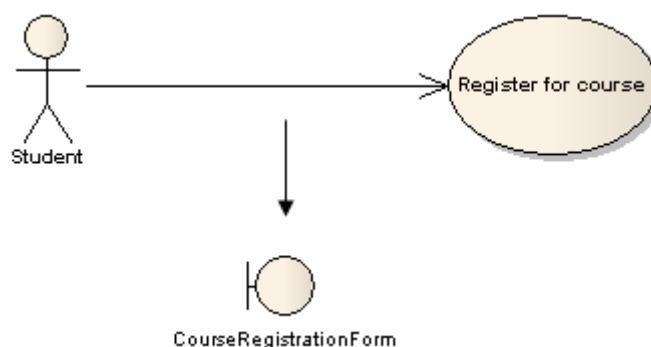


Một số hướng dẫn để tìm các lớp biên:

- Một khuyến nghị cho việc xác định ban đầu các lớp biên là một lớp biên cho mỗi cặp tác nhân / ca sử dụng.
- Một hệ thống có thể có một số loại lớp biên:
 - Các lớp giao diện người dùng: Các lớp giao tiếp trung gian với con người của hệ thống.

- Các lớp giao diện hệ thống: Các lớp giao tiếp trung gian với các hệ thống khác. Một lớp ranh giới giao tiếp với hệ thống bên ngoài chịu trách nhiệm quản lý đối thoại với hệ thống bên ngoài; ví dụ: các giao diện gọi các API của một hệ thống khác.
- Các lớp giao diện thiết bị: Các lớp cung cấp giao diện cho các thiết bị phát hiện các sự kiện bên ngoài. Các lớp ranh giới này nắm bắt các trách nhiệm của thiết bị hoặc cảm biến.

Ví dụ cho usecase Register for course:

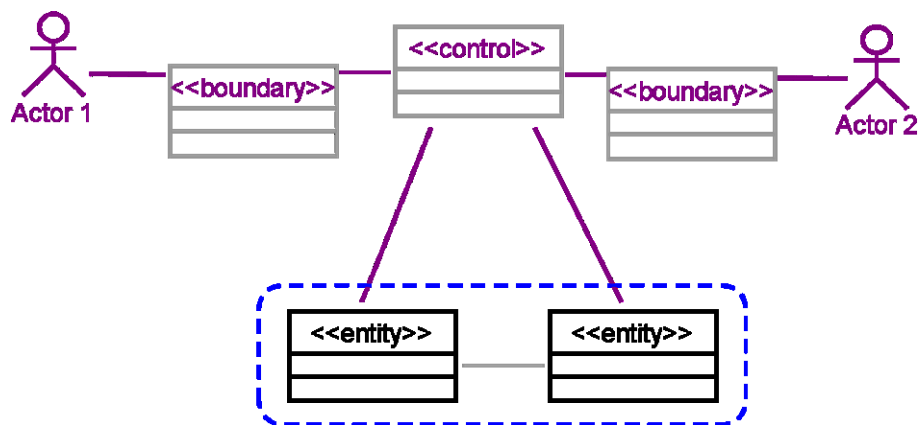


Entity classes

Các đối tượng thực thể đại diện cho các **khái niệm chính của hệ thống đang được phát triển**. Các lớp thực thể cung cấp một quan điểm khác để từ đó hiểu hệ thống, bởi vì chúng thể hiện cấu trúc dữ liệu logic. Biết cấu trúc dữ liệu có thể giúp hiểu hệ thống phải cung cấp cho người dùng những gì.

Các lớp thực thể đại diện cho các kho thông tin trong hệ thống. Chúng thường được sử dụng để đại diện cho các khái niệm chính mà hệ thống quản lý. Đối tượng thực thể (các thể hiện của lớp thực thể) được sử dụng để lưu giữ và cập nhật thông tin về một số hiện tượng, chẳng hạn như một sự kiện, một người hoặc một đối tượng ngoài đời thực.

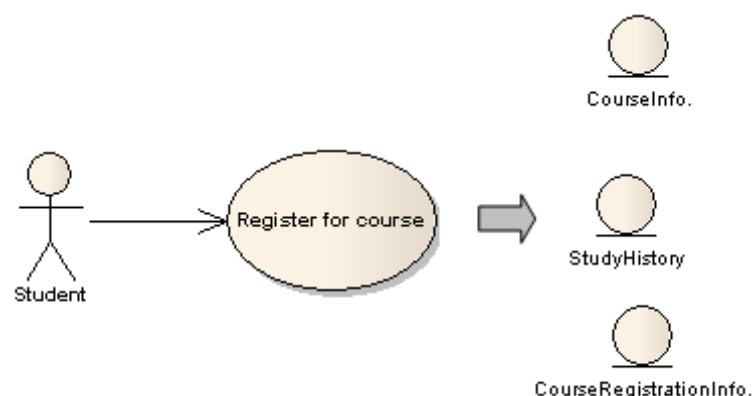
Trách nhiệm chính của các lớp thực thể là lưu trữ và quản lý thông tin trong hệ thống.



Một số hướng dẫn để tìm các lớp thực thể:

- Lấy luồng sự kiện theo trường hợp sử dụng làm đầu vào, gạch dưới các cụm danh từ trong luồng sự kiện. Chúng tạo thành danh sách ứng viên ban đầu của các lớp phân tích.
- Tiếp theo, thực hiện một loạt các bước lọc trong đó một số lớp ứng viên bị loại bỏ. Phương pháp lọc danh từ truyền thống
 - Gạch chân mệnh đề danh từ trong dòng sự kiện tình huống sử dụng
 - Loại bỏ các ứng viên thừa
 - Loại bỏ các ứng viên mơ hồ
 - Loại bỏ các tác nhân (ngoài phạm vi)
 - Loại bỏ các cấu trúc triển khai
 - Loại bỏ các thuộc tính (lưu để sử dụng sau)
 - Loại bỏ các hoạt động

Ví dụ qua đặc tả usecase “Register for course” chúng ta có thể tìm kiếm các ứng viên qua các danh từ, từ đó tinh chỉnh để xác định các lớp thực thể:



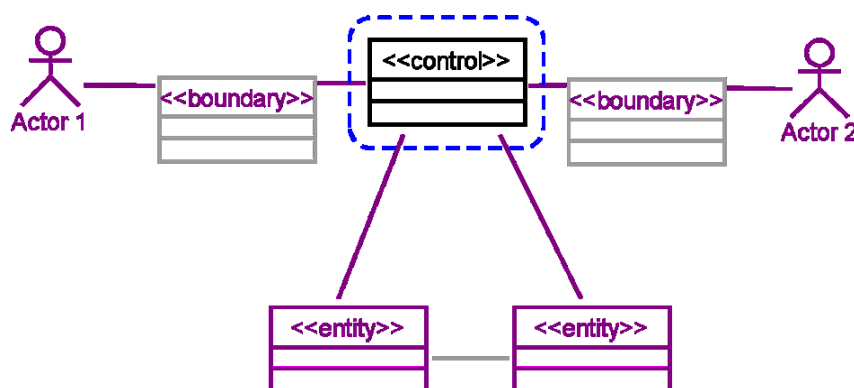
Control classes

Các lớp điều khiển cung cấp hành vi phối hợp trong hệ thống.

Các lớp điều khiển tách các đối tượng ranh giới và thực thể khỏi nhau một cách hiệu quả, làm cho hệ thống dễ chịu hơn với những thay đổi trong ranh giới hệ thống. Chúng cũng tách hành vi cụ thể của trường hợp sử dụng khỏi các đối tượng thực thể, làm cho chúng có thể tái sử dụng nhiều hơn trên các trường hợp sử dụng và hệ thống.

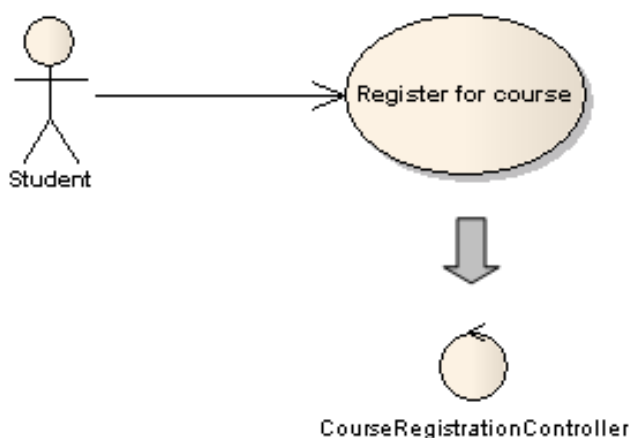
Các lớp điều khiển cung cấp hành vi:

- Không phụ thuộc vào môi trường xung quanh (không thay đổi khi môi trường xung quanh thay đổi).
- **Xác định logic điều khiển (thứ tự giữa các sự kiện) và các giao dịch trong một ca sử dụng.**
- Thay đổi ít nếu cấu trúc bên trong hoặc hành vi của các lớp thực thể thay đổi.
- Sử dụng hoặc thiết lập nội dung của một số lớp thực thể, và do đó cần phối hợp hành vi của các lớp thực thể này.
- Không được thực hiện theo cùng một cách mỗi khi nó được kích hoạt (luồng sự kiện có một số trạng thái).



Một số hướng dẫn để tìm các lớp điều khiển:

- Một khuyến nghị là xác định một lớp điều khiển cho mỗi trường hợp sử dụng.
- Hệ thống có thể thực hiện một số trường hợp sử dụng mà không có lớp điều khiển bằng cách chỉ sử dụng các lớp thực thể và lớp biên. Điều này đặc biệt đúng đối với các trường hợp sử dụng chỉ liên quan đến thao tác đơn giản đối với thông tin được lưu trữ.
- Các trường hợp sử dụng phức tạp hơn thường yêu cầu một hoặc nhiều lớp điều khiển để điều phối hành vi của các đối tượng khác trong hệ thống. Mỗi lớp điều khiển chịu trách nhiệm điều phối / kiểm soát quá trình thực hiện chức năng được mô tả trong trường hợp sử dụng liên quan. Ví dụ về các lớp kiểm soát bao gồm quản lý giao dịch, điều phối tài nguyên và xử lý lỗi.



Phần II: Xây dựng biểu đồ trình tự

Sau khi trích xuất các đối tượng phân tích, đối với mỗi trường hợp sử dụng, chúng ta cần:

- Phân bổ trách nhiệm ca sử dụng cho các lớp phân tích
- Các tương tác của lớp phân tích mô hình trong Sơ đồ tương tác, chẳng hạn như sơ đồ tuần tự.

Việc phân bổ trách nhiệm trong phân tích là một hoạt động quan trọng và đôi khi khó khăn. Ba khuôn mẫu này làm cho quá trình dễ dàng hơn bằng cách cung cấp một tập hợp các trách nhiệm đóng hộp có thể được sử dụng để xây dựng một hệ thống mạnh mẽ. Các trách nhiệm được xác định trước này tách biệt các phần của hệ thống có nhiều khả năng thay đổi: giao diện (các lớp ranh giới), luồng sự kiện ca sử dụng (lớp điều khiển) và dữ liệu liên tục (lớp thực thể).

Một số hướng dẫn để phân bổ trách nhiệm ca sử dụng cho các lớp phân tích:

- Các lớp ranh giới
 - o Hành vi liên quan đến giao tiếp với một tác nhân
- Lớp thực thể
 - o Hành vi liên quan đến dữ liệu được đóng gói trong phần trừu tượng
- Lớp kiểm soát
 - o Hành vi cụ thể cho một ca sử dụng hoặc một phần của luồng sự kiện rất quan trọng

Bài mẫu:

1. Với các usecase “Add course” & “Cancel registration” có đặc tả kịch bản như sau, thực hiện xác định các lớp phân tích và xây dựng biểu đồ trình tự.

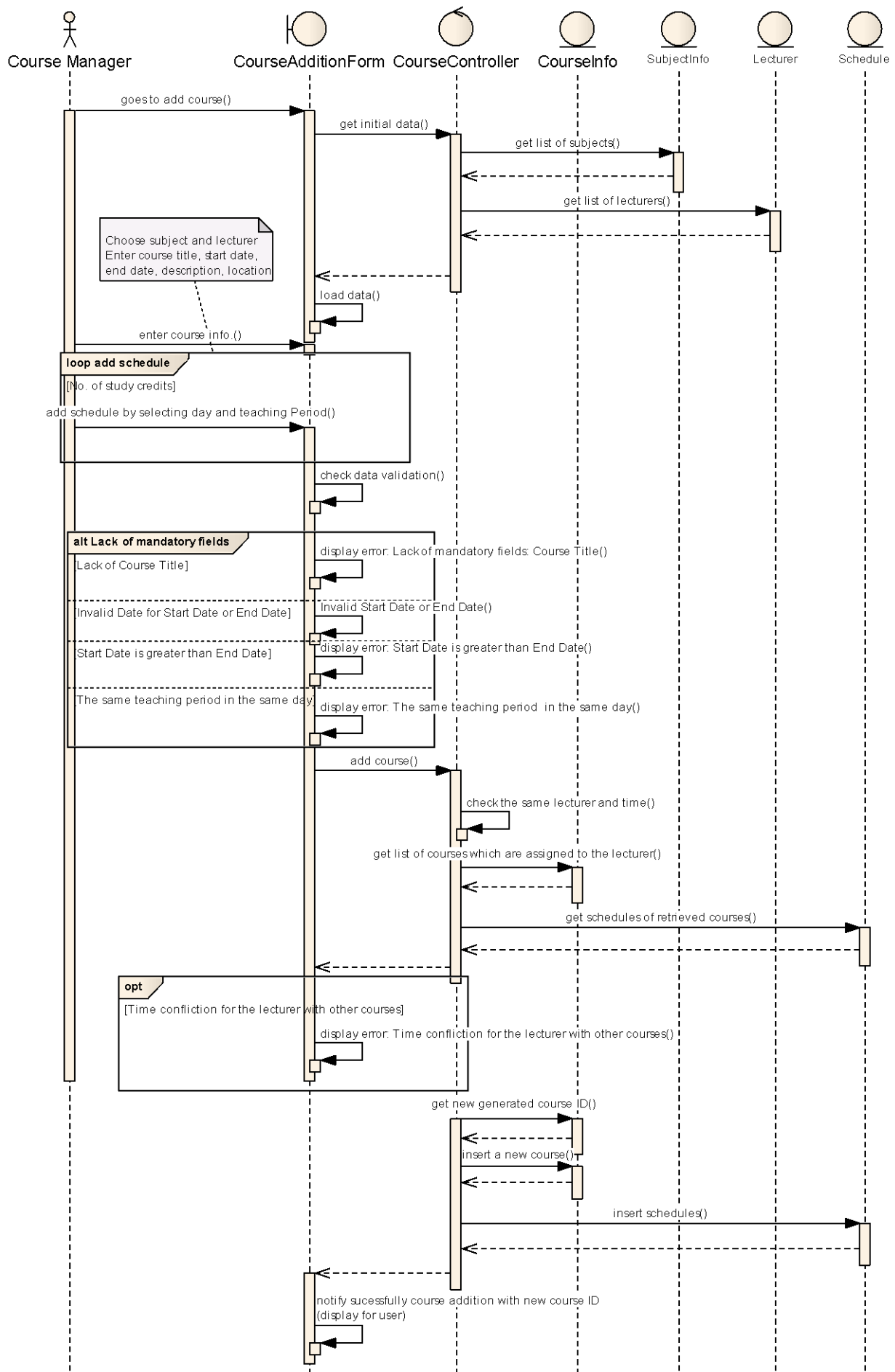
1.1. “Add course”

UC Code (UC #)	UC001	UC Name	Add course
Actor	Course manager		
Pre-condition	Course registration mustnot be opened		
Main Success Scenario (Flow of event)	No.	Doer	Action
	1.	Course manager	goes to Add course
	2.	System	loads the initial data
	3.	Course manager	enters course information (described below)
	4.	Course manager	submits to add course

	5.	System	checks if there is any blank mandatory field.
	6.	System	checks if Start Date and End Date is valid and if End Date is greater than Start Date
	7.	System	checks if time confliction for the lecturer
	8.	System	generates new course ID and add course
	9.	System	notifies successful course addition
Extension Scenarios (Alternative flows)	No.	Doer	Action
	6a.	System	notifies error: Lack of mandatory fields
	7a.	System	notifies error: Invalid Start Date and/or End Date
	8a.	System	notifies error: Time confliction for the lecturer

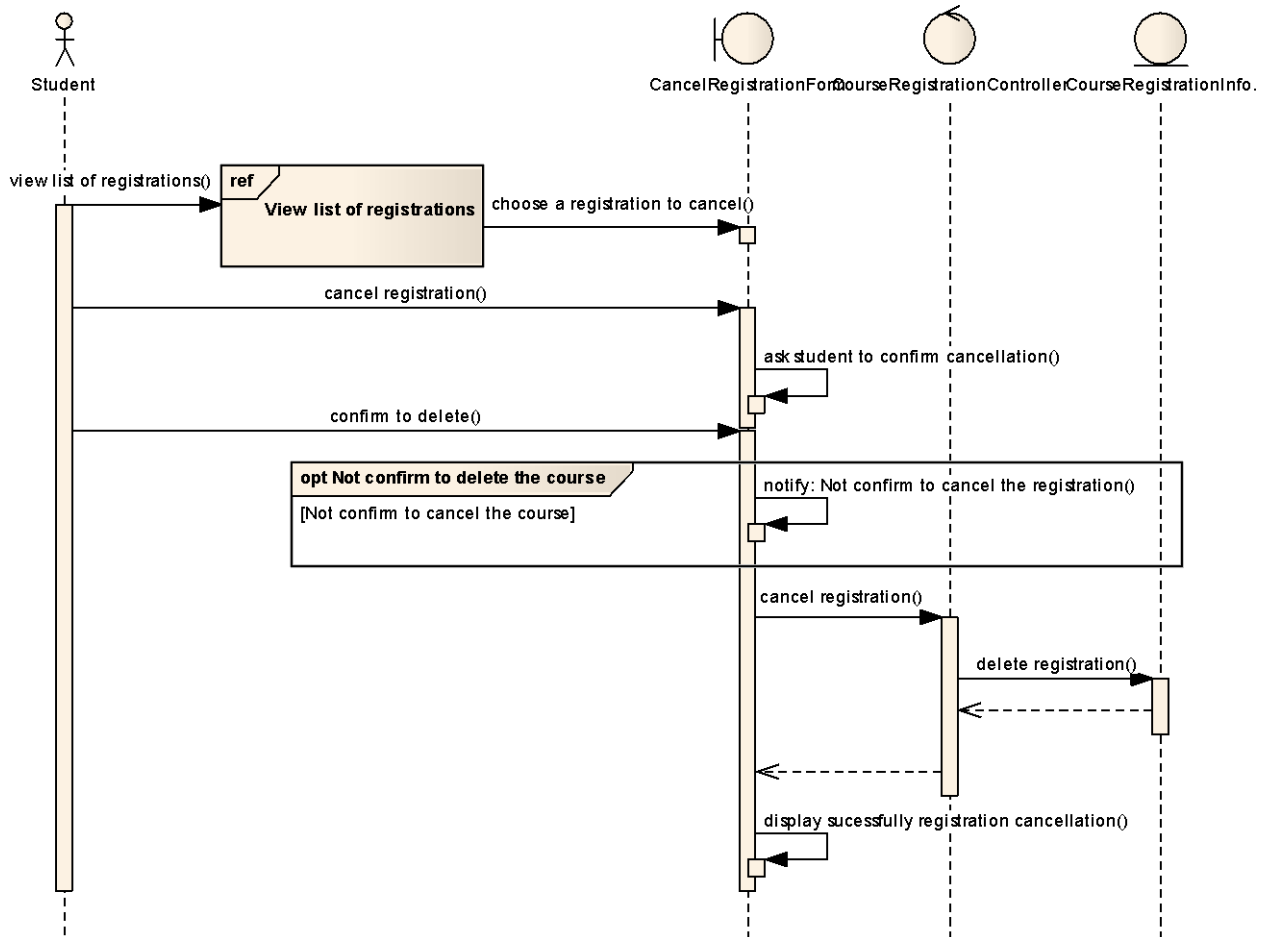
Input Data for course information:

N	Data item	Description	Mandatory	Condition
1.	Course Title	Title of the course	Yes	
2.	Subject	Choose from a list of subjects	No	
3.	Lecturer	Choose from a list of lecturers	No	
4.	Start Date		No	Valid date, Must be less than End Date
5.	End Date		No	Valid date, Must be greater than Start Date
6.	Description		No	
7.	Goal		No	
8.	Schedule	Choose schedule for the course with days and teaching period	No	



1.2. "Cancel registration"

UC Code (UC #)	UC005	UC Name	Cancel registration
Actor	Student		
Pre-condition	Logged in as a student, course registration must be opened, not closed		
Guarantee			
Includes	View list of course		
Extends			
Main Success Scenario (Flow of event)	No.	Doer	Action
	1.	Student	views list of course registrations and selects the course to be canceled
	2.	Student	goes to cancel registration
	3.	System	asks to confirm to cancel the registration.
	4.	Student	confirms to cancel the registration
	5.	System	deletes the registration
	6.	System	notifies successful registration.
Extensions (Alternative flows)	No.	Doer	Action
	5a.	System	notifies: Not confirm to cancel registration.



HẾT