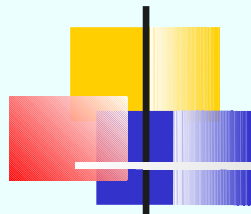




Công nghệ phần mềm

Pha phân tích

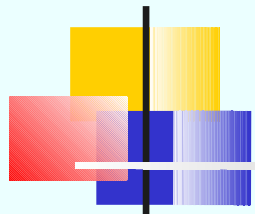
Giảng viên: TS. Nguyễn Mạnh Hùng
Học viện Công nghệ Bưu chính Viễn thông (PTIT)



Pha phân tích (1)

Mục đích:

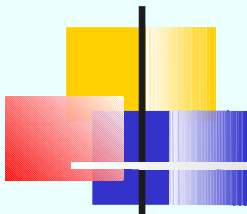
- Giúp đội phát triển hiểu sâu hơn yêu cầu của khách hàng
- Đặc tả yêu cầu của khách hàng dưới dạng có thể làm đầu vào cho thiết kế và cài đặt được



Pha phân tích (2)

Thực hiện:

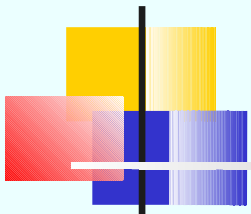
- Trích các lớp: lớp thực thể, lớp biên, lớp điều khiển
- Xác định quan hệ (ban đầu giữa các lớp)



Pha phân tích (3)

Lớp thực thể (còn gọi là lớp model):

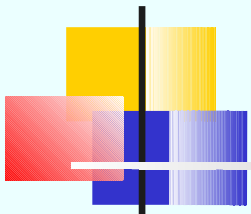
- Dùng để biểu diễn dữ liệu để xử lý, trao đổi giữa các đối tượng trong hệ thống
- Thường chỉ có các thuộc tính và các phương thức truy nhập *get/set*



Pha phân tích (4)

Lớp biên (còn gọi là lớp view):

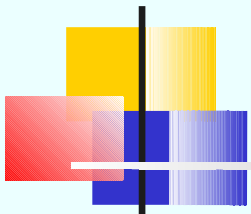
- Dùng để biểu diễn các dạng giao diện, giao tiếp giữa người dùng và hệ thống
- Mỗi lớp biên thường liên quan đến một thiết bị đầu vào, hoặc đầu ra của hệ thống



Pha phân tích (5)

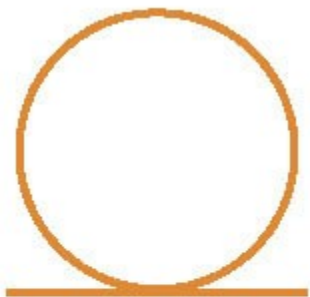
Lớp điều khiển (còn gọi là lớp control):

- Dùng để mô hình các tính toán và thuật toán phức tạp trong hệ thống
- Có thể chỉ cần dùng một lớp điều khiển cho các hệ thống đơn giản, mỗi phương thức là một hàm xử lý, tính toán độc lập

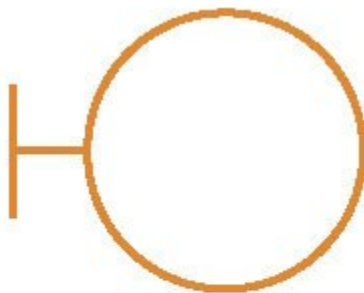


Pha phân tích (6)

Biểu diễn các dạng lớp trong UML:



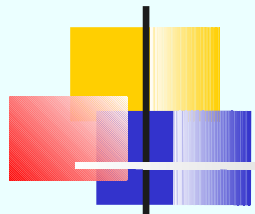
Entity Class



Boundary Class



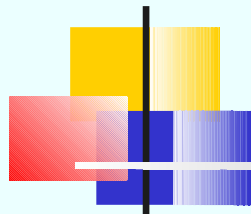
Control Class



Trích các lớp

Lặp lại 3 bước chính sau:

- B1: Mô hình hóa chức năng
- B2: Mô hình hóa các lớp
- B3: Mô hình hóa hoạt động



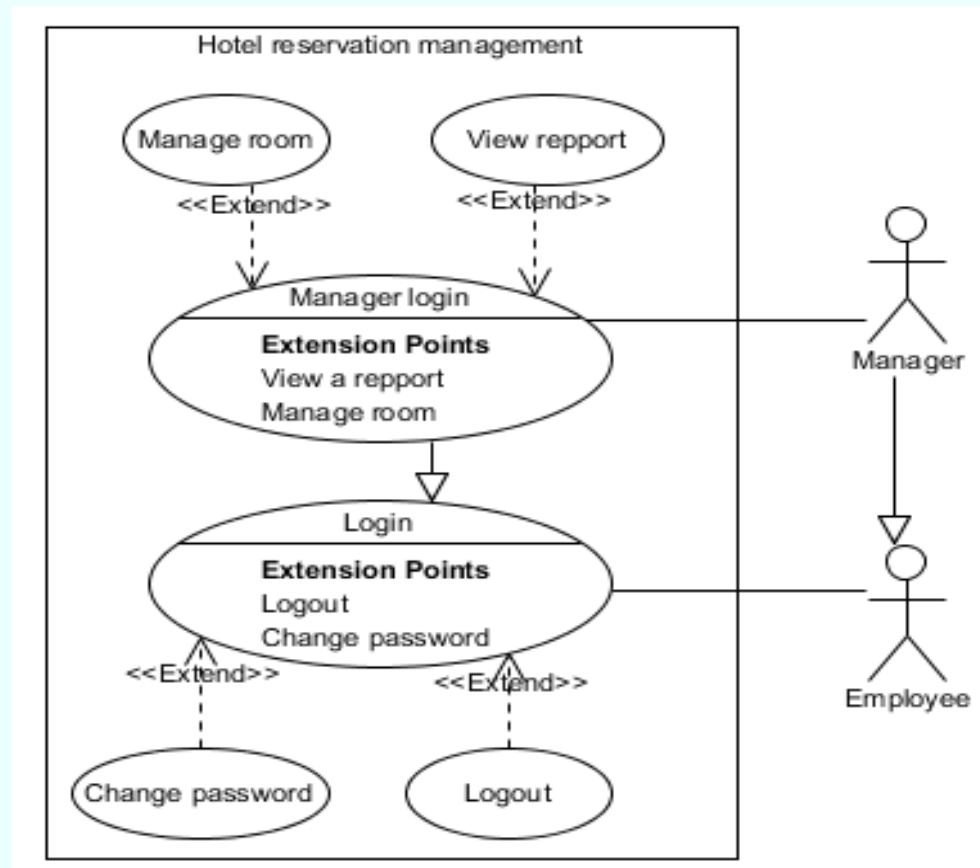
Mô hình hóa chức năng

Thực hiện:

- Với mỗi use case (kết quả từ pha yêu cầu), viết ít nhất một scenario cho use case đấy
- Một scenario là một kịch bản cụ thể khi người sử dụng tương tác với hệ thống

Viết scenario (1)

Ví dụ với bài toán đặt phòng khách sạn, các use case của người quản lí:





Viết scenario (2)

Mô tả các use case:

- Manager login: Use case này cho phép người quản lí đăng nhập theo tài khoản của mình
- View a repport: use case này cho phép người quản lí xem một báo cáo về doanh thu hoặc tỉ lệ phòng trống
- Manage room: use case này cho phép người quản lí thêm, hoặc sửa, hoặc xóa thông tin về phòng của khách sạn



Viết scenario (3)

Manage room: scenario chuẩn cho thêm phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn thêm thông tin một phòng mới.
2. Giao diện quản lí phòng hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút thêm phòng.
4. Giao diện thêm phòng hiện ra với các ô nhập: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 2 nút: nút thêm phòng, và nút hủy bỏ.
5. Nhân viên A nhập các thông tin phòng mới vào các ô và click nút thêm phòng
6. Thông báo thêm phòng thành công hiện ra.
7. A click vào nút ok
8. Hệ thống quay về trang chủ người quản lí



Viết scenario (4)

Manage room: scenario ngoại lệ cho thêm phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn thêm thông tin một phòng mới.
2. Giao diện quản lí phòng hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút thêm phòng.
4. Giao diện thêm phòng hiện ra với các ô nhập: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 2 nút: nút thêm phòng, và nút hủy bỏ.
5. Nhân viên A nhập các thông tin phòng mới vào các ô và click nút thêm phòng
6. Thông báo phòng với id vừa nhập đã tồn tại hiện ra.
7. A click vào nút ok



Viết scenario (5)

Manage room: scenario ngoại lệ cho thêm phòng (tt)

8. Hệ thống quay lại giao diện nhập phòng với các thông tin đã nhập lần trước

9. Nhân viên A nhập lại id mới và click nút thêm phòng

10. Thông báo thêm phòng thành công hiện ra.

11. A click vào nút ok

12. Hệ thống quay về trang chủ người quản lí



Viết scenario (6)

Manage room: scenario chuẩn cho sửa phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn sửa thông tin phòng 305.
2. Giao diện quản lí phòng hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút sửa phòng.
4. Giao diện tìm kiếm phòng hiện ra với một ô nhập tên phòng và một nút tìm kiếm
5. A nhập 305 vào ô tên phòng và click vào nút tìm kiếm
6. Giao diện kết quả tìm kiếm hiện ra gồm một bảng các phòng có tên 305, mỗi dòng có đầy đủ thông tin một phòng với các cột: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 1 nút chọn sửa.



Viết scenario (7)

Manage room: scenario chuẩn cho sửa phòng (tt)

7. A chọn click vào nút chọn sửa của dòng thứ nhất.

8. Giao diện sửa phòng hiện ra với các ô chứa sẵn thông tin phòng đã chọn gồm có: id phòng(không sửa được), tên phòng, kiểu phòng, giá hiển thị, mô tả, 1 nút hủy bỏ và 1 nút sửa.

9. A sửa thông tin loại phòng và mô tả phòng, và click vào nút sửa.

10. Thông báo sửa phòng thành công hiện ra.

11. A click vào nút OK

12. Hệ thống quay lại trang chủ của người quản lí.



Viết scenario (8)

Manage room: scenario chuẩn cho xóa phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn xóa thông tin phòng 503.
2. Giao diện quản lí phòng hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút sửa phòng.
4. Giao diện tìm kiếm phòng hiện ra với một ô nhập tên phòng và một nút tìm kiếm
5. A nhập 503 vào ô tên phòng và click vào nút tìm kiếm
6. Giao diện kết quả tìm kiếm hiện ra gồm một bảng các phòng có tên 503, mỗi dòng có đầy đủ thông tin một phòng với các cột: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 1 nút chọn xóa.



Viết scenario (9)

Manage room: scenario chuẩn cho xóa phòng (tt)

7. A chọn click vào nút chọn xóa của dòng thứ nhất.
8. Một dialog hiện ra yêu cầu nhân viên quản lí xác nhận có muốn xóa thông tin phòng 503 hay không
9. A click vào nút xác nhận có muốn xóa.
10. Thông báo xóa phòng thành công hiện ra.
11. A click vào nút OK
12. Hệ thống quay lại trang chủ của người quản lí.



Viết scenario (10)

Yêu cầu của một scenario:

- Luôn phải nêu rõ ý định, mục đích của actor trước khi bắt đầu scenario hoặc ngay trong bước 1
- Các bước luôn đánh số thứ tự từ 1
- Mỗi một bước chỉ có một hành động đơn. Nếu có hai hành động trở lên thì các hành động đó phải cùng một chủ thể
- Hai bước liên tiếp mà có cùng một chủ thể thì nên gộp lại thành một bước
- Scenario kết thúc khi chủ thể ban đầu đạt được mục đích ban đầu
- Một scenario có thể có nhiều ngoại lệ, các ngoại lệ cần được mô tả hết



Viết scenario (11)

Yêu cầu của một scenario:

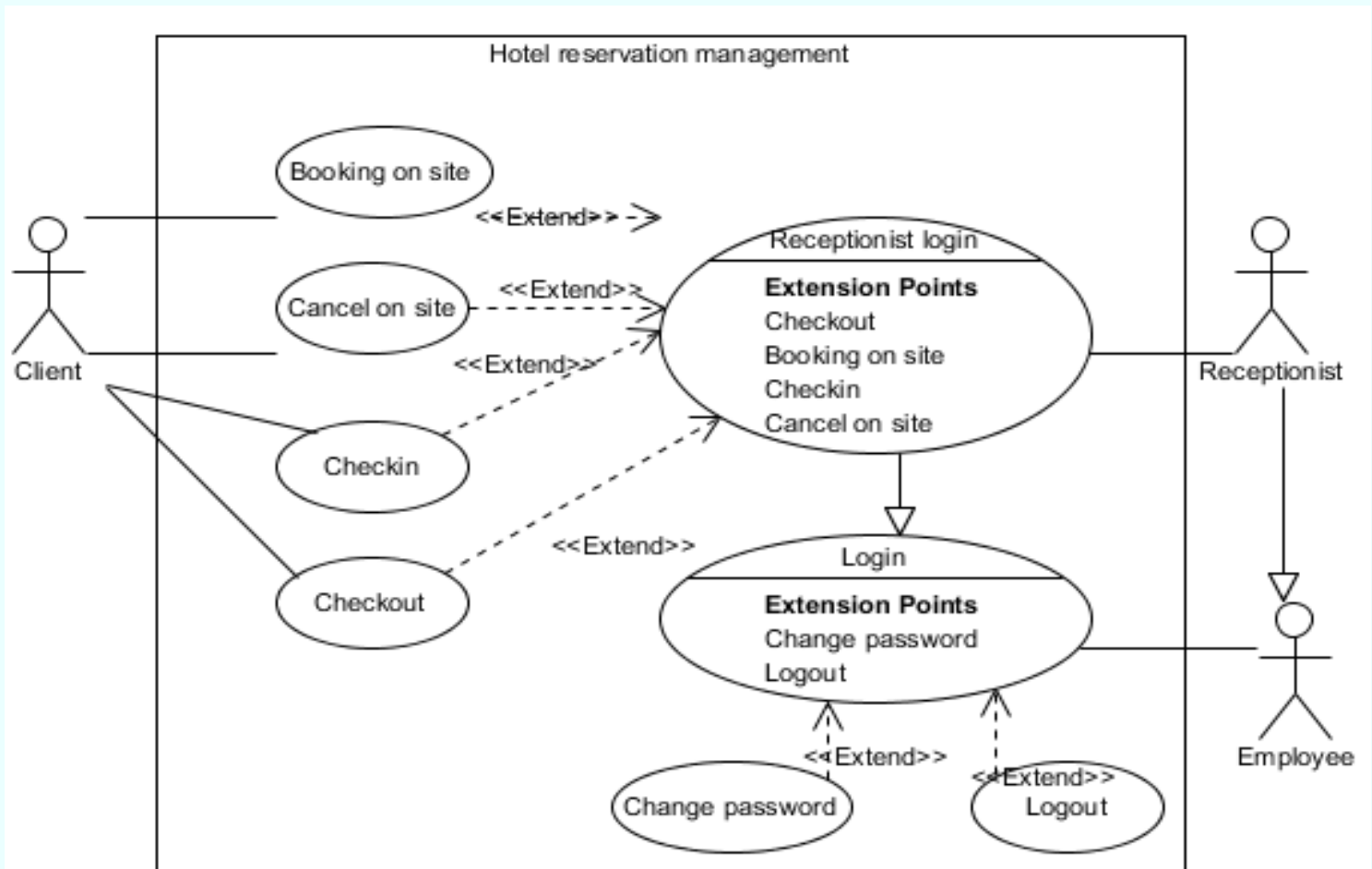
- Thông tin trong mỗi bước là phải cụ thể, không được viết chung chung

Ví dụ viết thể này là **không được**:

1. Nhân viên click vào nút xóa phòng
2. Giao diện tìm kiếm phòng hiện ra
3. Nhân viên nhập tên phòng vào
4. Giao diện kết quả hiện ra
5. Nhân viên chọn xóa 1 phòng
6. Hệ thống thông báo xóa thành công

Viết scenario (12)

Ví dụ với bài toán đặt phòng khách sạn, các use case của nhân viên lễ tân:





Viết scenario (13)

Mô tả các use case:

- Receptionist login: Use case này cho phép nhân viên lễ tân đăng nhập theo tài khoản của mình
- Booking on site: use case này cho phép nhân viên lễ tân đặt phòng khi có yêu cầu từ khách hàng tại quầy.
- Cancel on site: use case này cho phép nhân viên lễ tân hủy đặt phòng khi có yêu cầu từ khách hàng tại quầy.
- Checkin: use case này cho phép nhân viên lễ tân cập nhật thông tin khách đã nhận phòng
- Checkout: use case này cho phép nhân viên lễ tân cập nhật thông tin khách trả phòng và thanh toán cho khách hàng



Viết scenario (14)

Scenario chuẩn cho đặt phòng tại chỗ

1. Nhân viên tiếp tân A chọn chức năng quản lí đặt phòng sau khi login. A muốn thêm thông tin đặt phòng do khách hàng B yêu cầu.
2. Giao diện quản lí đặt phòng hiện ra với 3 nút: thêm, sửa, hủy đặt phòng
3. Nhân viên A click vào nút thêm đặt phòng.
4. Giao diện tìm phòng trống hiện ra với các ô nhập: ngày bắt đầu, ngày kết thúc và nút tìm kiếm.
5. Nhân viên A hỏi khách hàng B ngày bắt đầu, kết thúc mong muốn.
6. Khách hàng B nói với nhân viên A ngày bắt đầu, kết thúc.
7. Nhân viên A nhập ngày bắt đầu, kết thúc và click nút tìm kiếm.



Viết scenario (15)

Scenario chuẩn cho đặt phòng tại chỗ (tt)

8. Giao diện kết quả tìm kiếm phòng trống hiện ra với bảng danh sách các phòng trong trong khoảng ngày đã chọn. Mỗi phòng tương ứng với một dòng với các thông tin: id phòng, tên phòng, kiểu phòng, giá, mô tả, và nút chọn đặt.
9. Nhân viên A thông báo danh sách các phòng trống cho khách hàng B chọn.
10. Khách hàng B chọn phòng thứ 2 trong danh sách.
11. A click vào nút chọn đặt của dòng thứ hai.
12. Giao diện nhập thông tin khách hàng hiện ra, bao gồm các ô nhập thông tin khách hàng: họ tên, số CMND/passport, kiểu giấy id, địa chỉ, mô tả, ghi chú. Một nút thêm, một nút tìm kiếm.



Viết scenario (16)

Scenario chuẩn cho đặt phòng tại chỗ (tt)

13. Nhân viên A hỏi khách hàng B tên

14. Khách hàng B nói tên mình là B cho nhân viên A.

15. Nhân viên A nhập tên B vào ô họ tên và click nút tìm kiếm.

16. Giao diện kết quả tìm kiếm thông tin khách hàng hiện ra gồm một bảng danh sách các khách hàng có tên B, mỗi khách hàng chứa các thuộc tính tương ứng với các cột: họ tên, số CMND/passport, kiểu giấy id, địa chỉ, mô tả, ghi chú, một nút chọn. Dưới cùng là nút thêm khách hàng mới.

17. Nhân viên A xác nhận các thông tin với khách hàng B

18. Khách hàng B xác nhận thông tin của mình chưa có trong danh sách đã tìm thấy.



Viết scenario (17)

Scenario chuẩn cho đặt phòng tại chỗ (tt)

19. Nhân viên A click vào nút thêm mới khách hàng ở phía dưới.
20. Giao diện nhập thông tin khách hàng mới hiện ra với các ô nhập: họ tên, số CMND/passport, kiểu giấy id, địa chỉ, mô tả, ghi chú, một nút chọn. Dưới cùng là nút thêm khách hàng mới.
21. Nhân viên A nhập các thông tin khách hàng B và click vào nút thêm mới
22. Giao diện xác nhận đặt phòng hiện lên với đầy đủ các thông tin: phòng đặt, ngày checkin, ngày checkout, giá đặt, khách hàng đặt. Dưới cùng là nút xác nhận và nút hủy bỏ.
23. Nhân viên A click vào nút xác nhận
24. Hệ thống báo đặt chỗ thành công và quay về trang chủ của nhân viên lễ tân



Bài tập trên lớp

Viết tất cả các scenario ngoại lệ cho:

- Sửa thông tin phòng
- Xóa thông tin phòng
- Đặt chỗ

Viết scenario chuẩn và ngoại lệ cho chức năng:

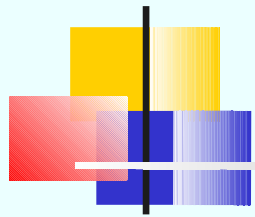
- Checkin
- Sửa đặt phòng
- Trả phòng + thanh toán tại chỗ



Bài tập về nhà

Với chủ đề bài tập lớn của nhóm:

- Chọn một use case trong modul của mình (mà bản thân cho là khó nhất và hay nhất)
- Viết lại phần mô tả modul đó trong pha yêu cầu
- Vẽ lại sơ đồ use case
- Viết 1 scenario chuẩn và tất cả các scenario ngoại lệ cho use case đã chọn



Mô hình hóa các lớp (1)

Thực hiện:

- Trích các lớp thực thể và các thuộc tính của chúng
- Xác định quan hệ và tương tác giữa các lớp này
- Biểu diễn các thông tin này trên sơ đồ lớp (khởi tạo)



Trích lớp thực thể (1)

Kĩ thuật trích danh từ để trích các lớp:

- Mô tả hoạt động của ứng dụng trong một đoạn văn
- Trích các danh từ xuất hiện trong đoạn văn đó, coi như là các ứng cử viên của lớp thực thể
- Xét duyệt từng danh từ và đề xuất nó là **lớp thực thể** hay là **thuộc tính của lớp thực thể**

Lưu ý:

- Có thể thay đoạn văn trong bước 1 bằng cách tập hợp các scenario đã viết trong bước trước



Trích lớp thực thể (2)

Ví dụ mô tả bài toán đặt phòng khách sạn:

- Hệ thống phục vụ hoạt động quản lí đặt phòng của một khách sạn. Trong đó, nhân viên quản lí có thể quản lí thông tin phòng và xem các báo cáo. Nhân viên quản trị có thể quản lí các tài khoản người dùng hệ thống. Nhân viên bán hàng có thể đặt phòng, thay đổi và hủy đặt phòng cho khách hàng thông qua điện thoại. Nhân viên tiếp tân có thể đặt phòng, thay đổi đặt phòng, hủy đặt phòng, làm thủ tục checkin, checkout và thanh toán trực tiếp tại chỗ cho khách hàng. Khi thanh toán có thể xuất hóa đơn theo yêu cầu của khách hàng, bao gồm tiền phòng và chi phí các dịch vụ gia tăng của khách sạn mà khách hàng đã dùng.



Trích lớp thực thể (3)

Ví dụ mô tả bài toán đặt phòng khách sạn:

- **Hệ thống** phục vụ hoạt động quản lí đặt **phòng** của một **khách sạn**. Trong đó, **nhân viên quản lí** có thể quản lí thông tin **phòng** và xem các **báo cáo**. **Nhân viên quản trị** có thể quản lí các **tài khoản người dùng hệ thống**. **Nhân viên bán hàng** có thể đặt **phòng**, thay đổi và hủy đặt **phòng** cho **khách hàng** thông qua **điện thoại**. **Nhân viên tiếp tân** có thể đặt **phòng**, thay đổi đặt **phòng**, hủy đặt **phòng**, làm thủ tục checkin, checkout và thanh toán trực tiếp tại chỗ cho **khách hàng**. Khi thanh toán có thể xuất **hóa đơn** theo **yêu cầu** của **khách hàng**, bao gồm **tiền phòng** và **chi phí các dịch vụ gia tăng** của **khách sạn** mà **khách hàng** đã dùng.



Trích lớp thực thể (4)

Các danh từ:

- Hệ thống, phòng, khách sạn, nhân viên quản lí, báo cáo, nhân viên quản trị, tài khoản người dùng, nhân viên bán hàng, khách hàng, điện thoại, nhân viên tiếp tân, hóa đơn, yêu cầu, tiền phòng, chi phí, dịch vụ gia tăng.

Đánh giá:

- **Điện thoại** nằm ngoài phạm vi của phần mềm → loại
- **Hệ thống, yêu cầu, tiền phòng, chi phí** là các danh từ trừu tượng → loại
- **Báo cáo** nên là một lớp biên hơn là lớp thực thể
- **Nhân viên quản lí, nhân viên quản trị, nhân viên bán hàng, nhân viên tiếp tân** đều có thể là các danh từ cụ thể của **tài khoản người dùng**



Trích lớp thực thể (5)

Như vậy chỉ còn các lớp thực thể:

- Phòng: Room
- Khách sạn: Hotel
- Tài khoản người dùng: User
- Hóa đơn: Bill
- Khách hàng: Client
- Dịch vụ gia tăng: Service

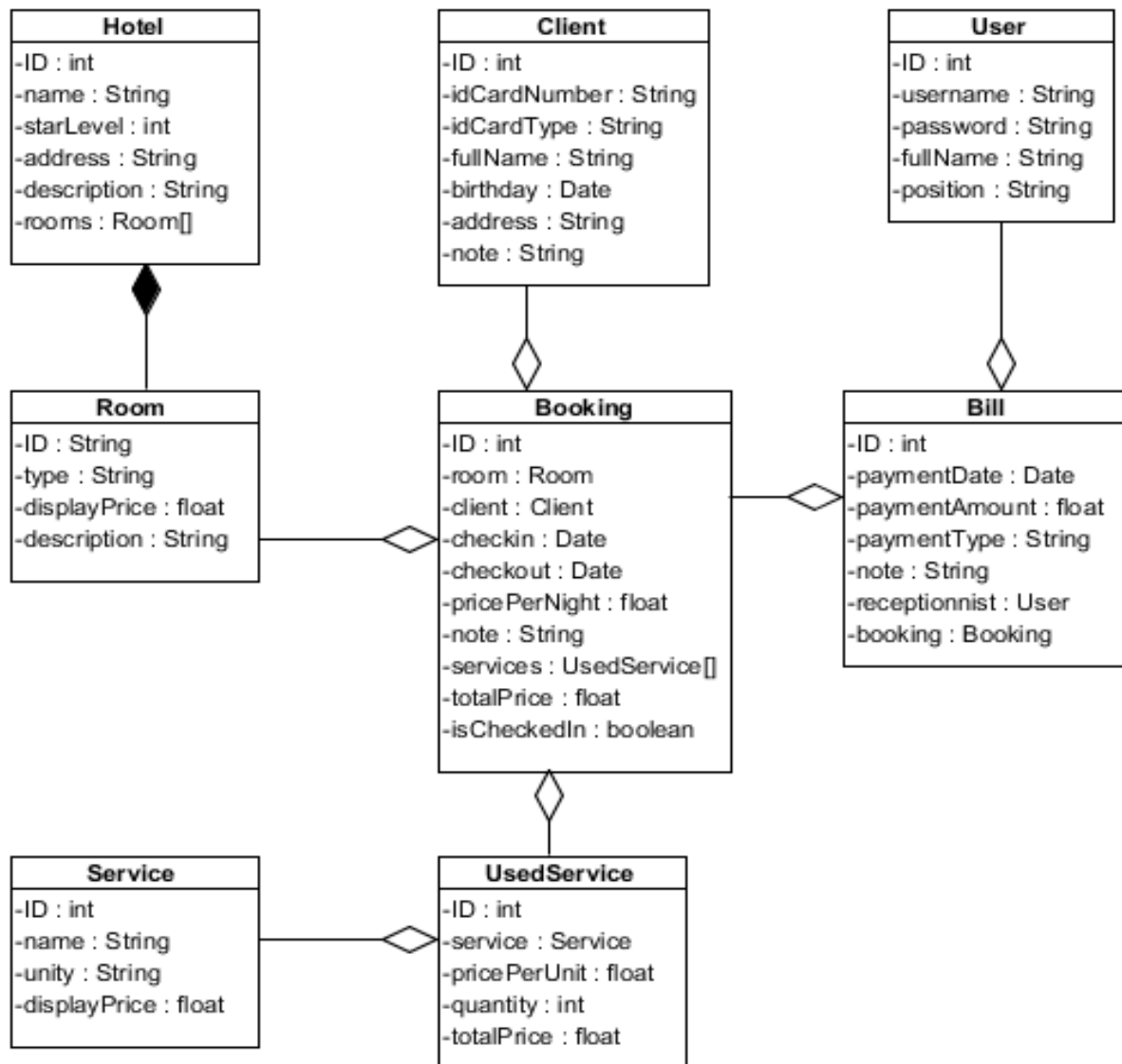


Trích lớp thực thể (6)

Quan hệ giữa các lớp thực thể:

- Một Hotel có nhiều Room, một Room phải thuộc vào một Hotel nhất định
- Một Room có thể đặt bởi nhiều Client, một Client lại có thể đặt nhiều Room tại nhiều thời điểm khác nhau → Đề xuất thêm một lớp **Booking**
- Một Booking có thể dùng nhiều Service khác nhau, một Service lại có thể được sử dụng bởi nhiều Booking khác nhau → Đề xuất thêm lớp **UsedService**
- Một Booking có thể được thanh toán nhiều lần khác nhau nên có thể có nhiều Bill
- Mỗi Bill có tối đa một User lập và nhận thanh toán.

Trích lớp thực thể (7)





Trích lớp điều khiển (1)

Đề xuất các lớp điều khiển:

- Toàn bộ hệ thống dùng chung một lớp điều khiển
- Mỗi modul dùng riêng một lớp điều khiển



Trích lớp điều khiển (2)

Đề xuất mỗi modul dùng riêng lớp điều khiển:

- Lớp điều khiển cho modul Manager: ManagerCtr
- Lớp điều khiển cho modul Admin: AdminCtr
- Lớp điều khiển cho modul Seller: SellerCtr
- Lớp điều khiển cho modul Receptionist: ReceptCtr



Trích lớp biên (1)

Đề xuất các lớp biên:

- Mỗi giao diện (trang web, form) nên để là một lớp biên
- Mỗi báo cáo, biểu mẫu nên để là một lớp biên
- Các thông báo, các thông điệp xác nhận có thể xem xét tạo thành một lớp biên hoặc là thành phần của một lớp biên



Trích lớp biên (2)

Đề xuất các lớp biên cho modul quản lí phòng của Manager:

- Giao diện chính: RoomManagerFrm
- Chức năng thêm: form thêm (AddRoomFrm)
- Chức năng sửa: form tìm kiếm (SearchEditRoomFrm), form kết quả (chung với SearchEditRoomFrm), form sửa (EditRoomFrm)
- Chức năng xóa: form tìm kiếm (SearchDeleteRoomFrm), form kết quả dùng chung với SearchDeleteRoomFrm.
- Các dialog và cửa sổ con đều là thành phần của các form chính

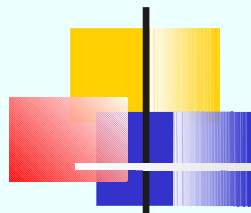


Mô hình hóa các lớp (2)

Dùng thẻ CRC để mô hình hóa quan hệ giữa các lớp:

- C: class. Biểu diễn tên lớp
- R: responsibility. Trách nhiệm của lớp
- C: collaboration. Quan hệ của lớp

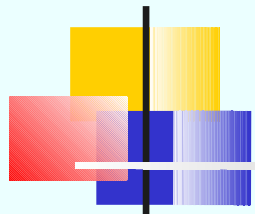
Trong VP: chọn new CRC Card diagram



Mô hình hóa các lớp (3)

Thẻ CRC cho lớp điều khiển modul Manager:

ManagerCtr	
Responsibilities:	
Name	Collaborator
Hiện giao diện quản lí phòng	
Hiện giao diện thêm phòng	
Hiện giao diện tìm kiếm phòng để sửa	
Hiện giao diện kết quả tìm kiếm phòng để sửa	
Hiện giao diện sửa phòng	
Hiện giao diện tìm kiếm phòng để xóa	
Hiện giao diện kết quả tìm kiếm phòng để xóa	
Lưu thông tin phòng mới vào CSDL	
Tìm kiếm phòng trong CSDL theo tên	
Cập nhật thông tin một phòng vừa sửa vào CSDL	
Xóa thông tin một phòng khỏi CSDL	
Đóng gói dữ liệu vào đối tượng Room	



Mô hình hóa các lớp (4)

Thẻ CRC cho lớp điều khiển (tt):

- Vì hướng đối tượng che giấu dữ liệu của các lớp nên không thể viết:

- Lớp điều khiển hiện giao diện quản lí phòng

Mà phải viết theo dạng tương tác giữa các lớp:

- Lớp **ManagerCtr** gửi thông điệp yêu cầu lớp **RoomManagerFrm** hiển thị giao diện chính

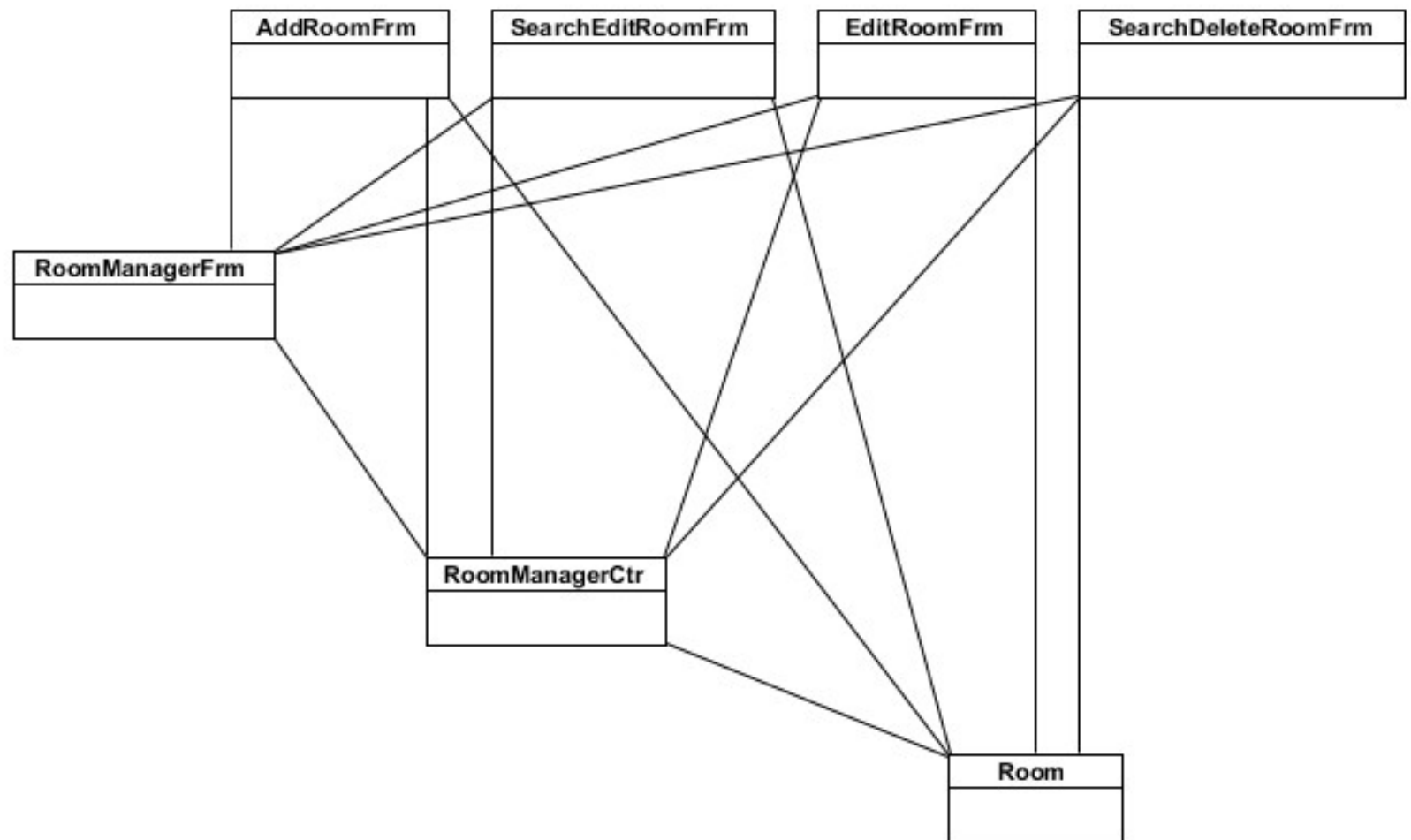
Mô hình hóa các lớp (5)

Thẻ CRC cho lớp điều khiển (tt):

ManagerCtr	
Responsibilities:	
Name	Collaborator
Yêu cầu lớp RoomManagerFrm hiện giao diện quản lí phòng	RoomManagerFrm
Yêu cầu lớp AddRoomFrm hiện giao diện thêm phòng	AddRoomFrm
Yêu cầu lớp SearchEditRoomFrm hiện giao diện tìm kiếm phòng để sửa	SearchEditRoomFrm
Yêu cầu lớp SearchEditRoomFrm hiện giao diện kết quả tìm kiếm phòng để sửa	SearchEditRoomFrm
Yêu cầu lớp EditRoomFrm hiện giao diện sửa phòng	EditRoomFrm
Yêu cầu lớp SearchDeleteRoomFrm hiện giao diện tìm kiếm phòng để xóa	SearchDeleteRoomFrm
Yêu cầu lớp SearchDeleteRoomFrm hiện giao diện kết quả tìm kiếm phòng để xóa	SearchDeleteRoomFrm
Lưu thông tin phòng mới vào CSDL	
Tìm kiếm phòng trong CSDL theo tên	
Cập nhật thông tin một phòng vừa sửa vào CSDL	
Xóa thông tin một phòng khỏi CSDL	
yêu cầu lớp Room đóng gói dữ liệu vào đối tượng Room	Room

Mô hình hóa các lớp (6)

Sơ đồ lớp cho modul quản lí phòng:





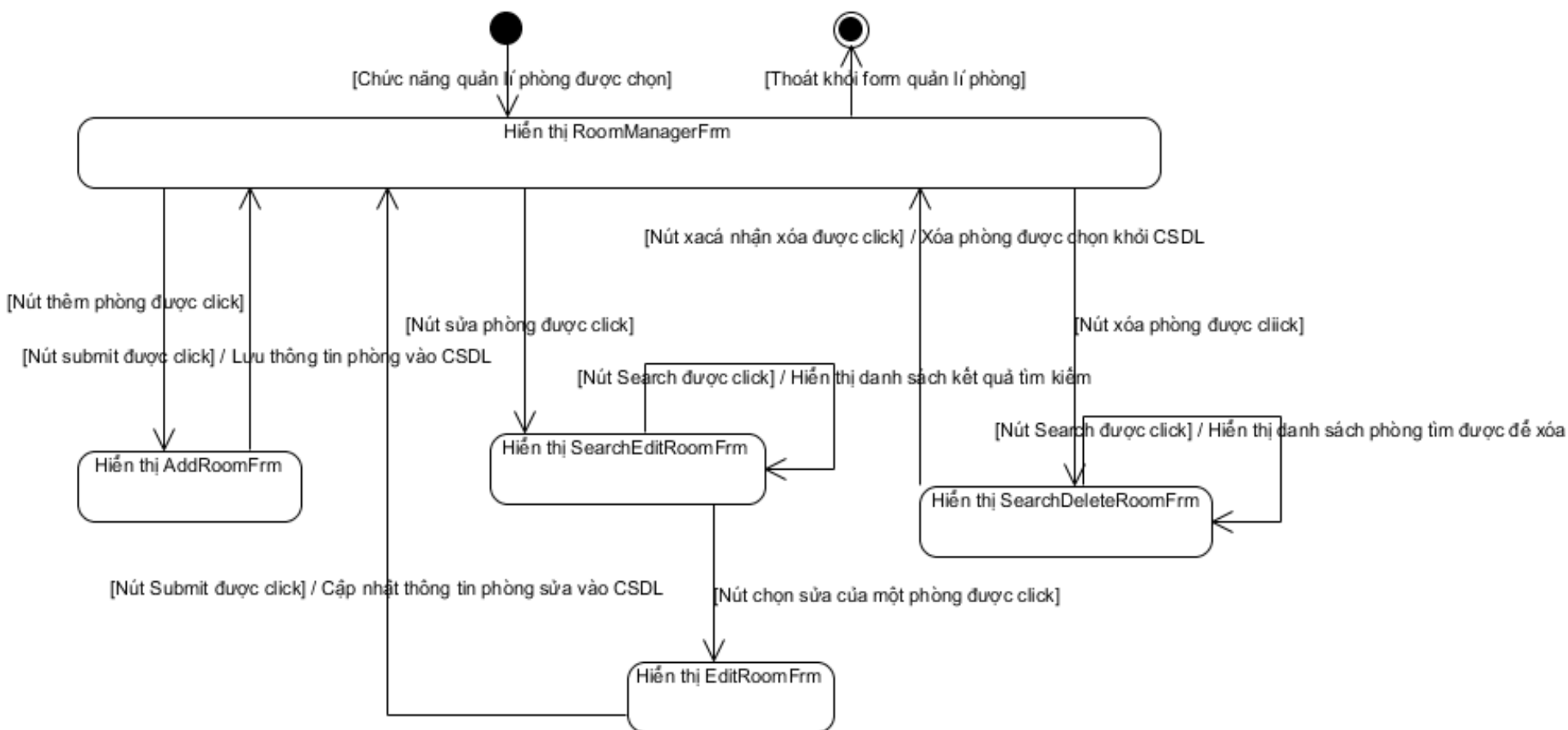
Mô hình hoạt động (1)

Sử dụng sơ đồ statechart:

- Mỗi trạng thái hệ thống được mô tả bằng một hình chữ nhật
- Khi có một sự kiện (event) xảy ra, thì trạng thái này sẽ chuyển sang trạng thái kia. Chuyển trạng thái biểu diễn bằng một mũi tên, nhãn là tên của sự kiện
- Các sự kiện và hoạt động tương tác được trích ra từ các scenario
- Trong VP: chọn new state machine diagram

Mô hình hoạt động (2)

Modul quản lí phòng:





Scenario cuối pha phân tích (1)

Trong scenario này:

- Các chủ thể hành động nằm trong phần mềm được thay thế bằng tên các lớp đã trích được
- Tương tác giữa các chủ thể chuyển thành hành động gửi thông điệp yêu cầu thực hiện hành động nào đó



Scenario cuối pha phân tích (2)

Manage room: scenario chuẩn cho thêm phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn thêm thông tin một phòng mới.
2. Lớp RoomManagerFrm hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút thêm phòng.
4. Lớp RoomManagerFrm gọi lớp AddRoomFrm yêu cầu hiển thị
5. Lớp AddRoomFrm hiện ra với các ô nhập: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 2 nút: nút thêm phòng, và nút hủy bỏ.
6. Nhân viên A nhập các thông tin phòng mới vào các ô và click nút thêm phòng
7. Lớp AddRoomFrm gọi lớp Room để đóng gói thông tin trên form thành một đối tượng kiểu Room



Scenario cuối pha phân tích (3)

Manage room: scenario chuẩn cho thêm phòng (tt)

8. Lớp Room đóng gói thông tin và trả lại cho lớp AddRoomFrm một đối tượng kiểu Room

9. Lớp AddRoomFrm chuyển đối tượng Room cho lớp RoomManagerCtr

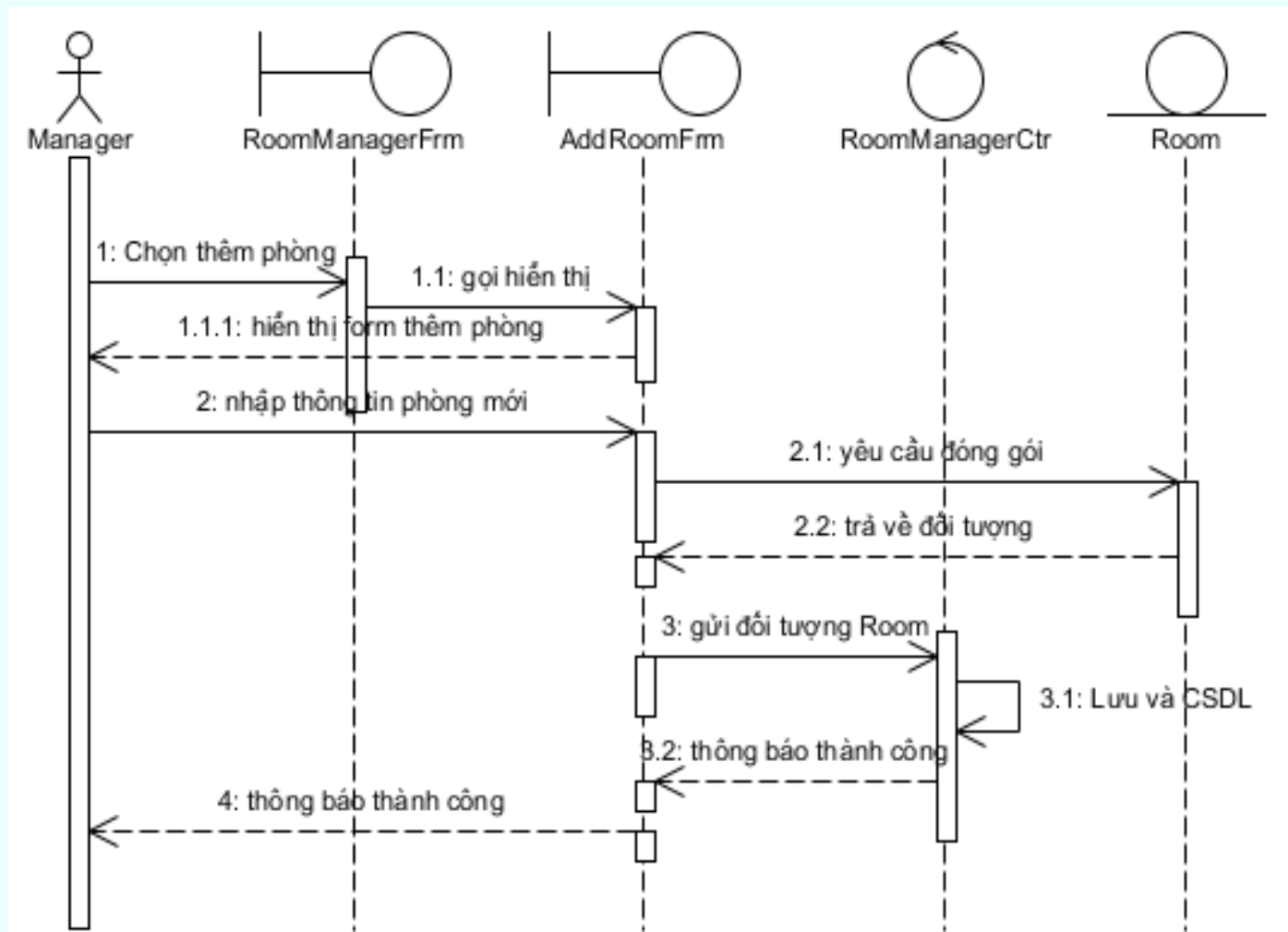
10. Lớp RoomManagerCtr lưu thông tin phòng vào CSDL

11. Lớp RoomManagerCtr thông báo cho lớp AddRoomFrm đã thêm thành công

12. Lớp AddRoomFrm thông báo thêm phòng thành công.

Scenario cuối pha phân tích (4)

Sơ đồ tuần tự cho scenario chuẩn cho thêm phòng





Scenario cuối pha phân tích (5)

Manage room: scenario chuẩn cho sửa phòng

1. Nhân viên quản lí A chọn chức năng quản lí phòng sau khi login. A muốn sửa thông tin phòng 305.
2. Lớp RoomManagerFrm hiện ra với 3 nút: thêm, sửa, xóa phòng
3. Nhân viên A click vào nút sửa phòng.
4. Lớp RoomManagerFrm gọi lớp SearchEditRoomFrm hiển thị
5. Lớp SearchEditRoomFrm hiện ra với một ô nhập tên phòng và một nút tìm kiếm
6. A nhập 305 vào ô tên phòng và click vào nút tìm kiếm
7. Lớp SearchEditRoomFrm gửi thông tin tên phòng 305 cho lớp RoomMaanagerCtr



Scenario cuối pha phân tích (6)

Manage room: scenario chuẩn cho sửa phòng (tt)

8. Lớp RoomManagerCtr tìm kiếm các phòng có tên 305 trong CSDL

9. Lớp RoomManagerCtr gửi kết quả đến lớp Room để đóng gói thành danh sách các đối tượng Room

10. Lớp Room gửi trả cho lớp RoomManagerCtr danh sách các đối tượng Room

11. Lớp RoomManagerCtr gửi danh sách các đối tượng Room cho lớp SearchEditRoomFrm để hiển thị

12. Lớp SearchEditRoomFrm hiện kết quả tìm kiếm gồm một bảng các phòng có tên 305, mỗi dòng có đầy đủ thông tin một phòng với các cột: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, và 1 nút chọn sửa.



Scenario cuối pha phân tích (7)

Manage room: scenario chuẩn cho sửa phòng (tt)

13. Nhân viên A click chọn sửa vào phòng thứ 2 trong danh sách

14. Lớp SearchEditRoomFrm gọi và truyền đối tượng vừa được chọn cho lớp EditRoomFrm

15. Lớp EditRoomFrm hiện ra với đầy đủ thông tin của đối tượng được chọn sửa: id phòng, tên phòng, kiểu phòng, giá hiển thị, mô tả, 1 nút hủy bỏ và một nút sửa.

16. Nhân viên A sửa một số thông tin về loại phòng, mô tả và giá của phòng và click vào nút sửa.

17. Lớp EditRoomFrm gửi thông tin trên form đến lớp Room để đóng gói đối tượng Room

18. Lớp Room đóng gói thông tin thành một đối tượng Room



Scenario cuối pha phân tích (8)

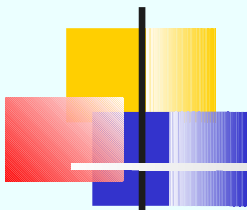
Manage room: scenario chuẩn cho sửa phòng (tt)

19. Lớp EditRoomFrm truyền đối tượng Room cho lớp RoomManagerCtr

20. Lớp RoomManagerCtr cập nhật thông tin phòng vào CSDL.

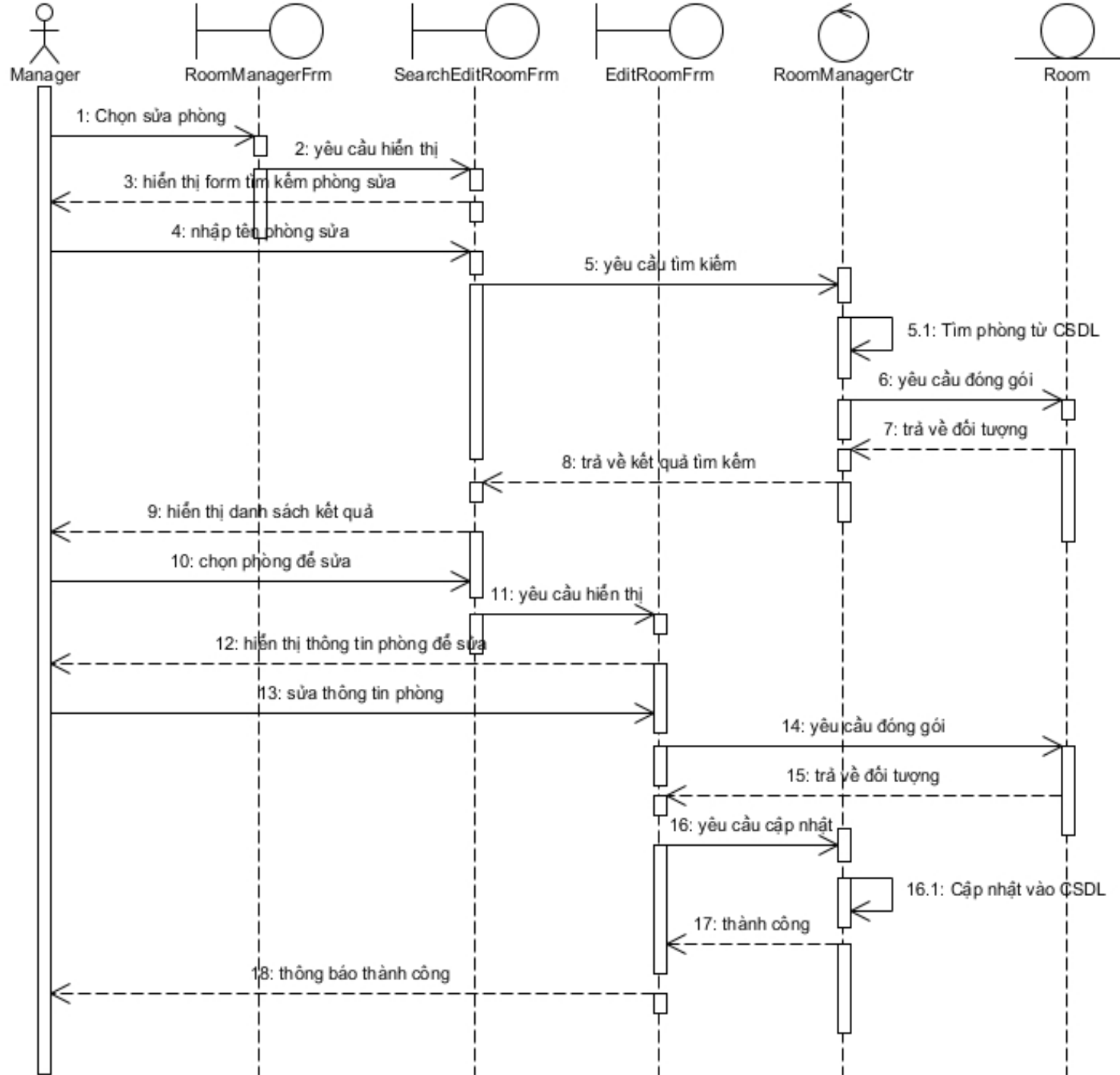
21. Lớp RoomManagerCtr thông báo cho lớp EditRoomFrm đã cập nhật thành công

22. Lớp EditRoomFrm thông báo cập nhật thành công cho nhân viên A



S

Sơ đồ tuần tự cho scenario chuẩn cho sửa phòng





Bài tập trên lớp

Thực hiện các bước sau lần lượt với các modul của Admin, của Seller, và của Receptionist:

- Trích các lớp biên, các lớp điều khiển
- Viết thẻ CRC cho lớp điều khiển
- Vẽ sơ đồ trạng thái cho modul
- Vẽ lại sơ đồ lớp cho modul
- Viết mỗi UC với một scenario phiên bản 2
- Thực tế hóa mỗi scenario này của mỗi UC thành sơ đồ tuần tự



Bài tập về nhà

Với mỗi modul của cá nhân:

- **Vẽ lại sơ đồ chi tiết các UC của modul cá nhân**
- Với mỗi UC, trích các scenario chuẩn và các ngoại lệ tương ứng (không cần xử lý các ngoại lệ sai kiểu dữ liệu đầu vào)
- Trích các lớp thực thể, trích các lớp biên, các lớp điều khiển. Vẽ sơ đồ lớp từ các lớp đã trích được.
- Xây dựng thẻ CRC cho các lớp điều khiển
- Xây dựng sơ đồ hoạt động (statechart) cho modul
- Viết lại các scenario với các lớp đã trích được
- Thực tế hóa mỗi scenario của mỗi UC thành sơ đồ tuần tự (hoặc cộng tác)



Questions?
