

# Nhập môn công nghệ phần mềm

## Phân tích hướng đối tượng

GV: ThS. Ngô Tiến Đức



- 2

# Luồng 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

# Luồng phân tích (2)

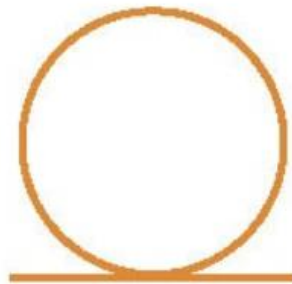
Phân tích **hướng đối tượng**:

- Mô hình hóa các chức năng: Viết các kịch bản
- Mô hình hóa các lớp – Phân tích tĩnh: Trích các lớp (lớp thực thể, lớp biên) xác định quan hệ ban đầu giữa các lớp
- Mô hình hóa các hoạt động - Phân tích động: Vẽ sơ đồ tuần tự

# Luồng phân tích (3)

Lớp thực thể (model):

- 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
- Ký hiệu trong UML:

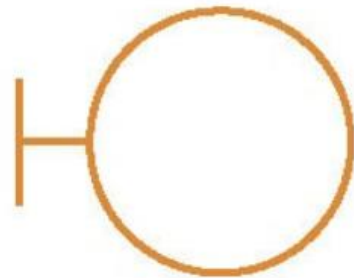


**Entity Class**

# Luồng phân tích (4)

Lớp biên (view):

- 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
- Ký hiệu trong UML:



**Boundary Class**

# Luồng phân tích (5)

Lớp điều khiển – Control class:

- Mô hình hóa 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
- Ký hiệu trong UML:



# Viết kịch bản (1)

- Scenario: Một kịch bản cụ thể khi người sử dụng tương tác với hệ thống
- Mục đích: Mô hình hóa các chức năng
- Với mỗi use case (kết quả lấy yêu cầu), viết ít nhất một scenario cho use case đó
- Scenario chuẩn: Thực hiện đầy đủ các bước từ đầu tới cuối
- Một scenario có thể có nhiều ngoại lệ



# Viết kịch bản (2)

Yêu cầu với các scenario:

- Đánh số thứ tự các bước bắt đầu từ 1
- Mỗi một bước chỉ nên 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 có cùng một chủ thể thì gộp lại thành một
- Thông tin trong mỗi bước phải cụ thể
- Scenario kết thúc khi chủ thể ban đầu đạt được mục đích ban đầu (được nhắc tới ở bước 1)

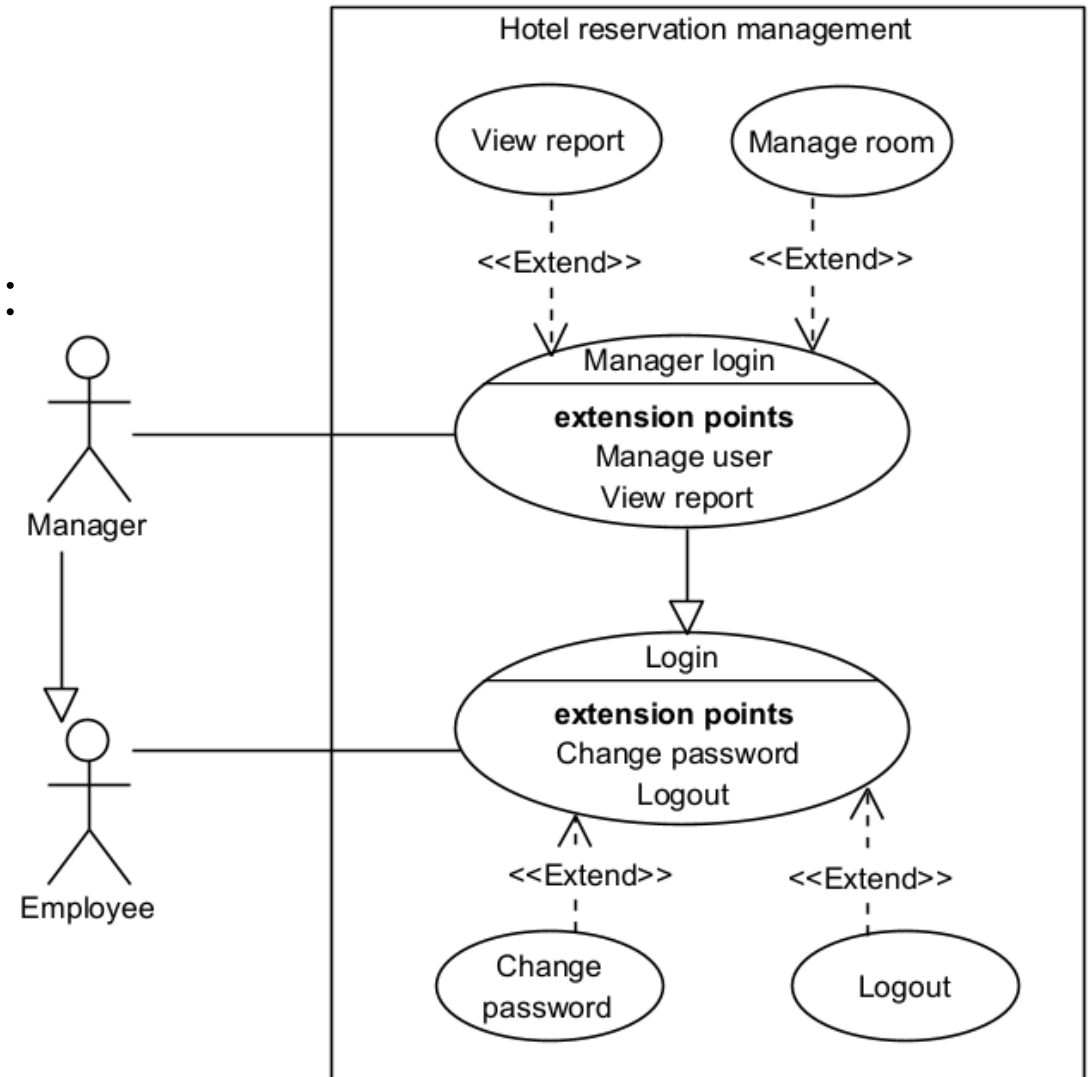
# Viết kịch bản (3)

Ví dụ về cách viết không đúng:

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

# Viết kịch bản (4)

Ví dụ các use case của người quản lý trong hệ thống đặt phòng khách sạn:





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

# Viết kịch bản (6)

## **Scenario chuẩn cho chức năng thêm phòng**

1. Nhân viên quản lý A chọn chức năng quản lý phòng từ trang chủ sau khi đăng nhập. A muốn thêm thông tin một phòng mới.
2. Giao diện quản lý phòng hiện ra với 3 nút: Thêm phòng, sửa phòng, xóa phòng.
3. A click vào nút thêm phòng.
4. Giao diện thêm phòng hiện ra với các ô nhập: ID phòng, tên phòng, kiểu phòng, giá phòng, mô tả, và 2 nút: nút lưu và nút hủy.

# Viết kịch bản (7)

5. A nhập các thông tin phòng mới vào các ô: ID = 555, name = Suite, type = single, price = 1000, description để trống, và click nút lưu.
6. Thông báo thêm phòng thành công hiện ra.
7. A click vào nút OK.
8. Hệ thống quay về trang chủ của người quản lý.



### 5.1. A không nhập thông tin và click nút thêm.

### 5.1.2. A click vào nút OK.

### 5.1.3. Hệ thống quay lại giao diện thêm phòng.

## 5.2. A click vào nút hủy.

### 5.2.1. Hệ thống quay về trang quản lý phòng.



### 6.1. Thông báo đã tồn tại phòng với ID vừa nhập hiện ra.

### 6.1.2. Hệ thống quay lại giao diện thêm phòng với các thông tin đã nhập.



# Viết kịch bản (10)

## **Scenario chuẩn cho chức năng sửa phòng**

1. Nhân viên quản lý A chọn chức năng quản lý phòng sau khi đăng nhập.  
A muốn sửa thông tin phòng có ID 404, tên Suite.
2. Giao diện quản lý phòng hiện ra với 3 nút: thêm phòng, sửa phòng, xóa phòng.
3. A click vào nút sửa phòng.
4. Giao diện tìm kiếm phòng hiện ra với một ô tìm kiếm (theo ID hoặc tên phòng) và một nút tìm.
5. A nhập "Suite" vào ô tìm kiếm phòng và click vào nút tìm.

# Viết kịch bản (11)

6. Giao diện kết quả tìm kiếm hiện ra gồm một bảng, mỗi dòng có đầy đủ thông tin một phòng với các cột: ID phòng, tên phòng, kiểu phòng, giá phòng, mô tả, và 1 nút chọn sửa ở cuối dòng.

ID	Name	Type	Price	Description	
404	Suite	Single	1000		<input type="button" value="Select"/>
505	Suite VIP	Triple	2000	New	<input type="button" value="Select"/>

7. A chọn click vào nút chọn sửa của dòng thứ nhất tương ứng với ID là 404.

# Viết kịch bản (12)

8. Giao diện sửa phòng hiện ra với các ô chứa sẵn thông tin phòng đã chọn gồm có: ID = 404 (không sửa được), name = Suite, type = Single, price = 1000, mô tả trống, nút lưu và nút hủy.
9. A sửa thông tin mô tả phòng thành "city view", và click vào nút cập nhật.
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 lý.

**Ngoại lệ:**

**6.1. Hệ thống thông báo không có phòng nào trong kết quả tìm kiếm...**

# Viết kịch bản(13)

## **Scenario chuẩn cho chức năng xóa phòng**

1. Nhân viên quản lý A chọn chức năng quản lý phòng sau khi đăng nhập. A muốn xóa phòng có ID là 404.
2. Giao diện quản lý phòng hiện ra với 3 nút: thêm phòng, sửa phòng, xóa phòng.
3. A click vào nút xóa phòng.
4. Giao diện tìm kiếm phòng hiện ra với một ô tìm kiếm và nút tìm.
5. A nhập 404 vào ô tìm kiếm và click vào nút tìm.

# Viết kịch bản (14)

6. Giao diện kết quả tìm kiếm hiện ra gồm một bảng, mỗi dòng có đầy đủ thông tin một phòng với các cột: ID phòng, tên phòng, kiểu phòng, giá phòng, mô tả, và 1 nút chọn xóa ở cuối dòng.

ID	Name	Type	Price	Description	
404	Suite	Single	1000	City view	<input type="button" value="Select"/>

7. A chọn click vào nút chọn xóa của dòng thứ nhất tương ứng với ID là 404.

8. Một hộp thoại hiện ra yêu cầu nhân viên quản lý xác nhận có muốn xóa phòng 404 hay không.



- Ngoại lệ:

## 6.1. Hệ thống thông báo không có phòng nào trong kết quả tìm kiếm...

## 9.1. A click vào nút hủy...

# Trích các lớp thực thể (1)

## Biểu đồ lớp – Class diagram:

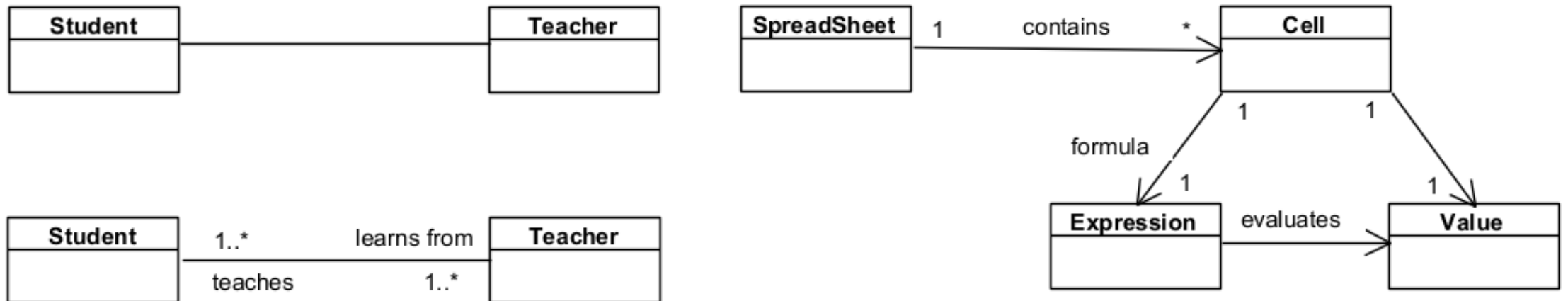
- Biểu diễn các lớp: tên, thuộc tính, phương thức
- Biểu diễn quan hệ giữa các lớp
- Đặt tên theo code convention
- Trong VP, chọn *Diagram > New > Class Diagram*

ClassName
-attribute : type
+operation() : return type

# Trích các lớp thực thể (2)

## Quan hệ giữa các lớp:

- **Association:** Các class có mối quan hệ với nhau
  - Có nhiều cách biểu diễn sử dụng chiều mũi tên, số lượng, tên



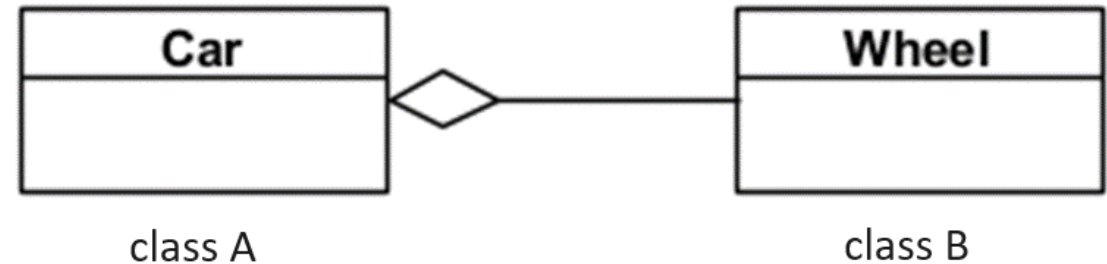


# Trích các lớp thực thể (3)

## Quan hệ giữa các lớp:

- **Aggregation:** Biểu diễn quan hệ "part of" (là một phần)

- A chứa (thông tin của) B
- B có thể tồn tại độc lập



- **Composition:** Dạng đặc biệt của Aggregation

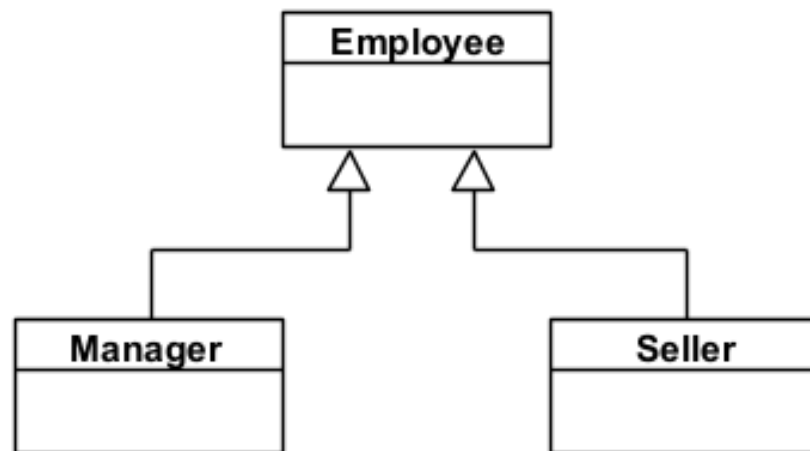
- B không thể tồn tại độc lập với A



# Trích các lớp thực thể (4)

**Quan hệ giữa các lớp:**

- **Inheritance/Generalization:** Biểu diễn quan hệ "is-a"
  - Class con là thể hiện chi tiết hơn của class cha



# Trích các lớp thực thể (5)

Thực hiện việc trích các lớp thực thể:

- Sử dụng kỹ thuật trích danh từ để trích lớp thực thể
- Xác định các thuộc tính cho từng lớp
- Xác định quan hệ giữa các lớp:
  - Nếu 2 lớp có quan hệ n-n thì tạo thêm lớp thực thể trung gian
- Biểu diễn trên sơ đồ lớp thực thể

# Trích các lớp thực thể (6)

Kĩ thuật trích danh từ để trích các lớp:

- Mô tả hoạt động của ứng dụng trong một đoạn văn
  - Có thể mô tả theo từng actor
- Trích các danh từ xuất hiện trong đoạn văn
- Xét duyệt từng danh từ và đề xuất nó là lớp thực thể, thuộc tính của lớp thực thể hoặc bị loại bỏ

# Trích các lớp thực thể (7)

Ví dụ: **Hệ thống** quản lý **thông tin** về **phòng** của **khách sạn**, **thông tin** về **khách hàng** đặt **phòng**. Trong đó **người quản trị** có thể quản lý **tài khoản người dùng**, **người quản lý** có thể quản lý **thông tin** về **phòng**, xem các loại **báo cáo thống kê** về **phòng**, **thống kê** về **khách hàng**, **thống kê** về **dịch vụ**, **thống kê doanh thu** theo **thời gian**. **Khách hàng** có thể đặt phòng qua **điện thoại** thông qua **nhân viên bán hàng**, hoặc đặt phòng trực tiếp tại **quầy** thông qua **nhân viên lễ tân**. **Nhân viên lễ tân** có thể thực hiện nhận phòng, trả phòng, thanh toán khi có **yêu cầu** từ **khách hàng**. Mỗi khi thanh toán, **hóa đơn** sẽ được in ra, bao gồm **tiền phòng** và **chi phí** cho các **dịch vụ** mà **khách hàng** đã sử dụng khi nghỉ tại **khách sạn**.

# Trích các lớp thực thể (8)

Đánh giá các danh từ:

- Hệ thống, thông tin: Các danh từ chung chung → loại
- Điện thoại, quầy: Nằm ngoài phạm vi của phần mềm → loại
- Tiền phòng, chi phí: Các danh từ chung chung nhưng nằm trong hóa đơn → đề xuất làm thuộc tính của hóa đơn
- Người quản lý, người quản trị, nhân viên bán hàng, nhân viên lễ tân: Các đối tượng được quản lý theo kiểu người dùng trực tiếp của hệ thống → đề xuất 1 lớp thực thể chung với tài khoản người dùng

# Trích các lớp thực thể (9)

Đánh giá các danh từ (tiếp): Các danh từ còn lại là đối tượng xử lý của hệ thống được đề xuất thành các lớp thực thể:

Phòng → lớp **Room**, khách sạn → lớp **Hotel**, tài khoản người dùng → lớp **User**, khách hàng → lớp **Client**, hóa đơn → lớp **Bill**, dịch vụ → lớp **Service**, thống kê khách sạn → lớp **HotelStat**, thống kê phòng → lớp **RoomStat**, thống kê khách hàng → lớp **ClientStat**, thống kê dịch vụ → lớp **ServiceStat**, thống kê doanh thu → lớp **IncomeStat**



- Lớp Hotel: tên khách sạn, hạng sao, địa chỉ, mô tả
- Lớp Room: tên phòng, kiểu phòng, giá hiển thị, mô tả
- Lớp User: tên đăng nhập, mật khẩu, họ tên, email, chức vụ
- Lớp Client: số CCCD, họ tên, địa chỉ, SĐT, email, ghi chú
- Lớp Bill: ngày thanh toán, số tiền thanh toán, hình thức thanh toán, ghi chú
- Lớp Service: tên, đơn vị, đơn giá
- Các lớp Stat: ...

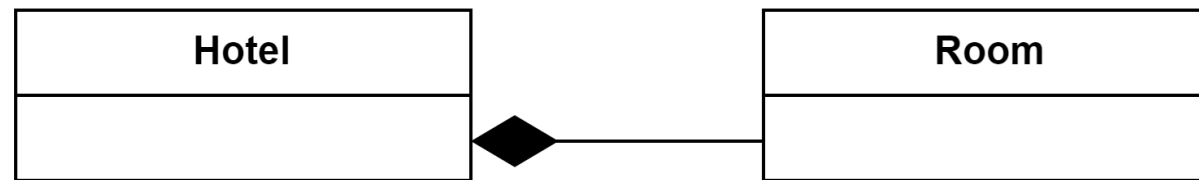


# Trích các lớp thực thể (11)

Quan hệ giữa các lớp thực thể:

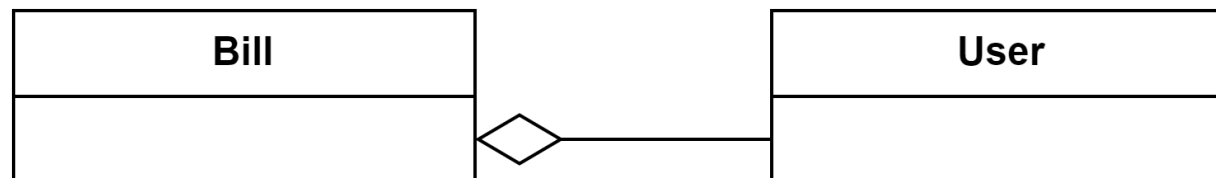
- Một Hotel có nhiều Room, một Room phải thuộc vào một Hotel

→ Quan hệ giữa Hotel và Room là composition 1-n



- Một User có thể lập nhiều Bill, một Bill chỉ có thể do một User lập

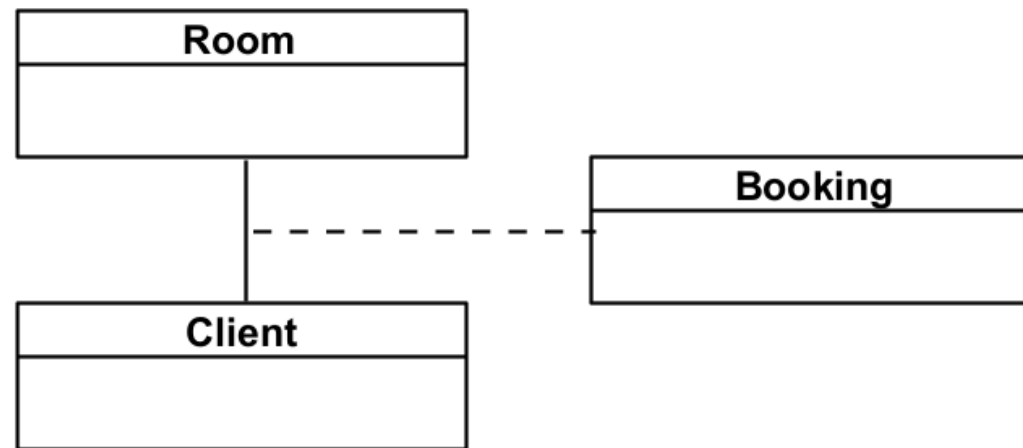
→ Quan hệ giữa User và Bill là aggregation 1-n



# Trích các lớp thực thể (12)

Quan hệ giữa các lớp thực thể:

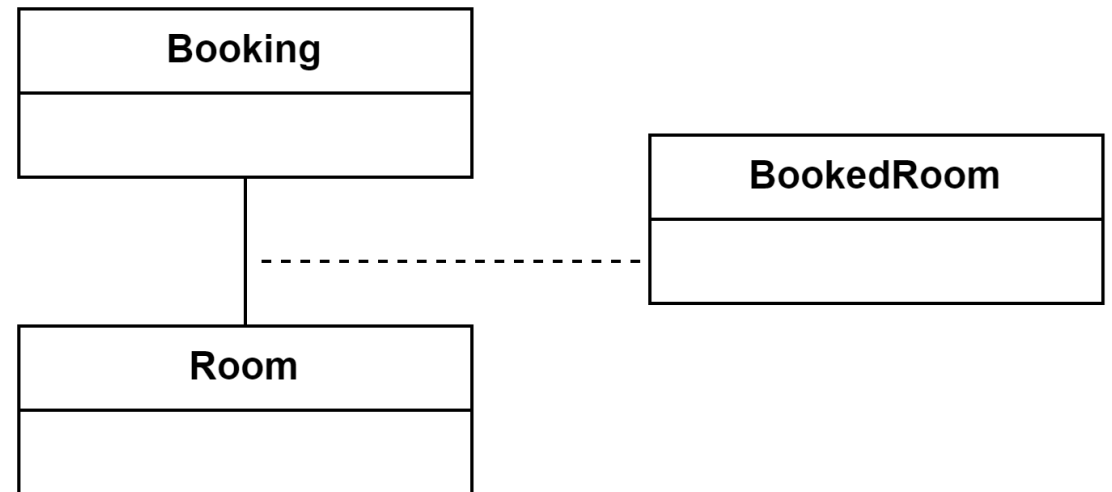
- Một Room có thể đặt bởi nhiều Client, một Client lại có thể đặt nhiều Room tại nhiều thời điểm khác nhau → Quan hệ giữa Client và Room là n-n  
→ Đề xuất thêm lớp thực thể **Booking (thông tin đặt phòng)** với các thuộc tính ngày đặt, tổng số tiền phải trả, ghi chú



# Trích các lớp thực thể (13)

Quan hệ giữa các lớp thực thể:

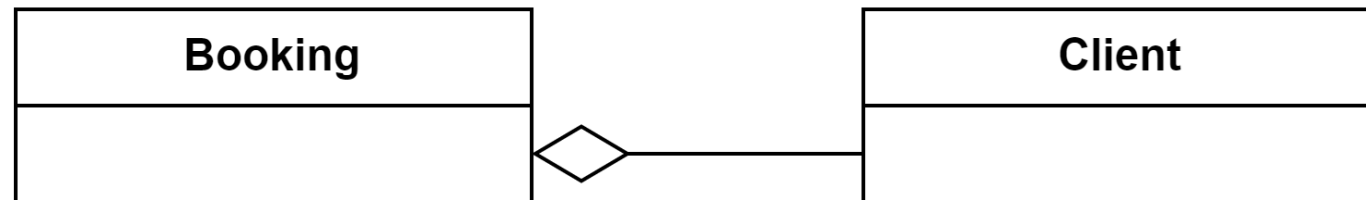
- Một phòng có thể được đặt nhiều lần, một lần đặt có thể đặt nhiều phòng  
→ Quan hệ giữa Room và Booking là n-n → Đề xuất thêm lớp thực thể **BookedRoom (thông tin phòng được đặt)** gồm các thuộc tính ngày nhận phòng, ngày trả phòng, giá thuê phòng (có thể khác với giá niêm yết), đã checkin hay chưa, ghi chú



# Trích các lớp thực thể (12)

Quan hệ giữa các lớp thực thể:

- Một khách hàng có thể có nhiều lần đặt phòng, một lần đặt phòng chỉ có thể của 1 khách hàng → Quan hệ giữa Client và Booking là aggregation 1-n



# Trích các lớp thực thể (14)

Quan hệ giữa các lớp thực thể:

- Khách hàng là người sử dụng dịch vụ nhưng dịch vụ được tính theo phòng được đặt nên chúng ta xét quan hệ giữa BookedRoom và Service (thay vì Client và Service): Một phòng có thể sử dụng nhiều dịch vụ, một dịch vụ có thể được sử dụng bởi nhiều phòng → Quan hệ giữa BookedRoom và Service là n-n → Đề xuất thêm lớp thực thể **UsedService (dịch vụ được sử dụng)** gồm các thuộc tính số lượng (theo đơn vị của dịch vụ), tổng tiền

# Trích các lớp thực thể (15)

Quan hệ giữa các lớp thực thể:

- Đối với các lớp thống kê, do chúng phải dùng lại một số thuộc tính của lớp thực thể tương ứng nên có quan hệ kế thừa từ các lớp thực thể tương ứng ấy: HotelStat kế thừa từ Hotel, RoomStat kế thừa từ Room, ServiceStat kế thừa từ Service, ClientStat kế thừa từ Client.
- Riêng lớp IncomeStat không dùng lại thuộc tính của lớp nào nên không có quan hệ kế thừa với lớp nào. Nó chỉ có quan hệ với lớp Bill vì có thể liên quan đến chi phí trong các hóa đơn.

```

classDiagram
    class RoomStat {
        -fillDay
        -fillRate
        -revenue
    }
    class Room {
        -name
        -type
        -price
        -description
    }
    class Booking {
        -bookingDate
        -totalPrice
        -note
    }
    class Client {
        -idCardNumber
        -fullName
        -address
        -phone
        -email
        -note
    }
    class HotelStat {
        -fillRate
        -revenue
    }
    class Hotel {
        -name
        -starLevel
        -address
        -description
    }
    class BookedRoom {
        -checkin
        -checkout
        -price
        -isCheckedIn
        -note
    }
    class User {
        -username
        -password
        -fullName
        -mail
        -position
        -description
    }
    class ClientStat {
        -day
        -payment
    }
    class Bill {
        -paymentDate
        -paymentAmount
        -paymentType
        -note
    }
    class ServiceStat {
        -amount
        -revenue
    }
    class Service {
        -name
        -unit
        -price
    }
    class UsedService {
        -quantity
        -totalPrice
    }
    class IncomeStat {
        -client
        -revenue
    }

    RoomStat --|> Room
    HotelStat --|> Hotel
    ServiceStat --|> Service
    ClientStat --|> Client
    Booking o-- Room
    Booking o-- Client
    Booking o-- User
    Booking o-- Bill
    Booking o-- User
    Booking .. BookedRoom
    BookedRoom .. UsedService
    User o-- Bill
    User o-- Bill
    Bill o-- IncomeStat

```

# Phân tích tĩnh (1)

Thực hiện việc trích các lớp biên (giao diện):

- Mỗi giao diện (trang web, form) nên để là một lớp biên. Ví dụ: Trang chủ của người quản lý -> lớp **ManagerHomeView**, trang giao diện thêm phòng -> lớp **AddRoomView**
- Các thông báo, các hộp thoại (dialog) xác nhận có thể xem xét tạo thành một lớp biên hay là thành phần của một lớp biên. Ví dụ: Thông báo thành công không phải một lớp riêng mà là thành phần nằm trong lớp **AddRoomView**



# Phân tích tĩnh (2)

Ví dụ modul thêm phòng của người quản lý (bỏ qua bước đăng nhập):

- Sau khi đăng nhập thành công, hệ thống hiện giao diện trang chủ của nhân viên quản lý -> đề xuất lớp **ManagerHomeView**, có ít nhất một nút chọn vào quản lý thông tin phòng.
- Click nút quản lý thông tin phòng -> giao diện quản lý phòng hiện lên -> đề xuất lớp **ManageRoomView**, có ít nhất một nút thêm phòng.
- Click nút thêm phòng -> hiển thị giao diện thêm phòng -> đề xuất lớp **AddRoomView**, có các ô nhập: tên, kiểu, giá, mô tả, nút lưu và nút hủy.

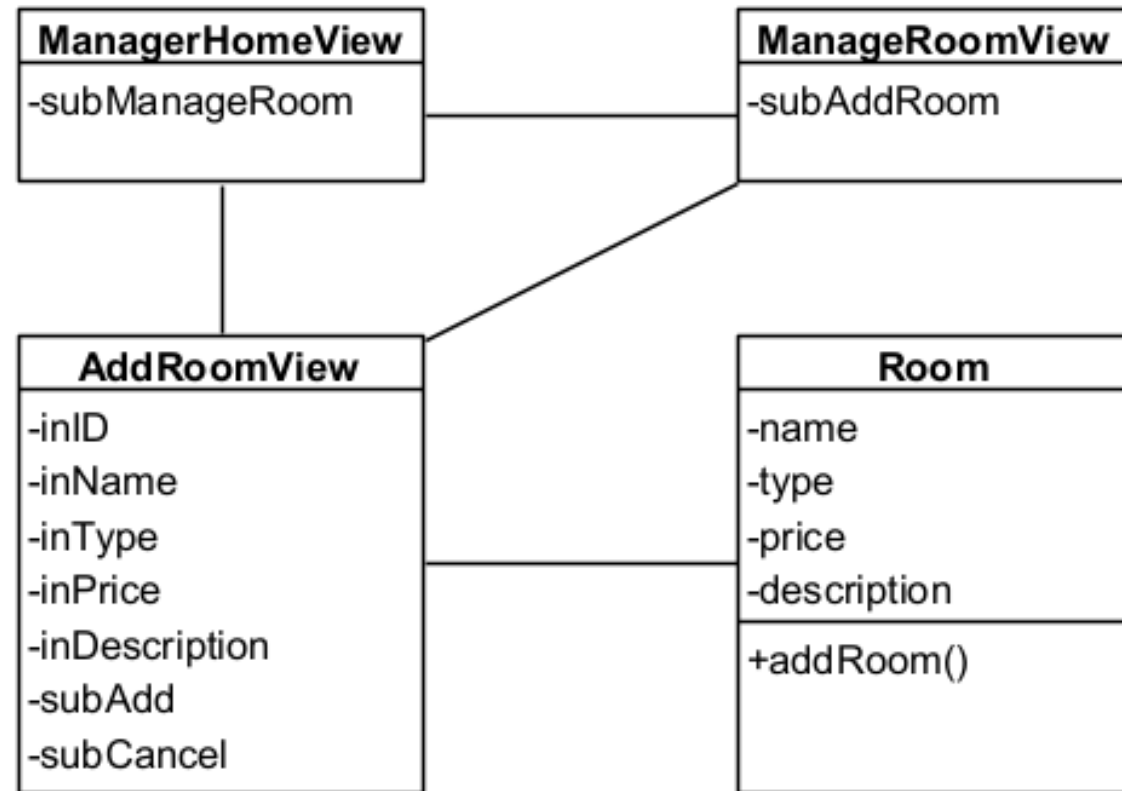
# Phân tích tĩnh (3)

Ví dụ chức năng thêm phòng của người quản lý (tiếp):

- Nhập thông tin và click nút lưu -> hệ thống lưu vào CSDL -> cần chức năng **addRoom()**, đây là hành động của đối tượng **Room**
- Sau khi thêm phòng mới, hệ thống quay về trang chủ của nhân viên quản lý **ManagerHomeView**

# Phân tích tĩnh (4)

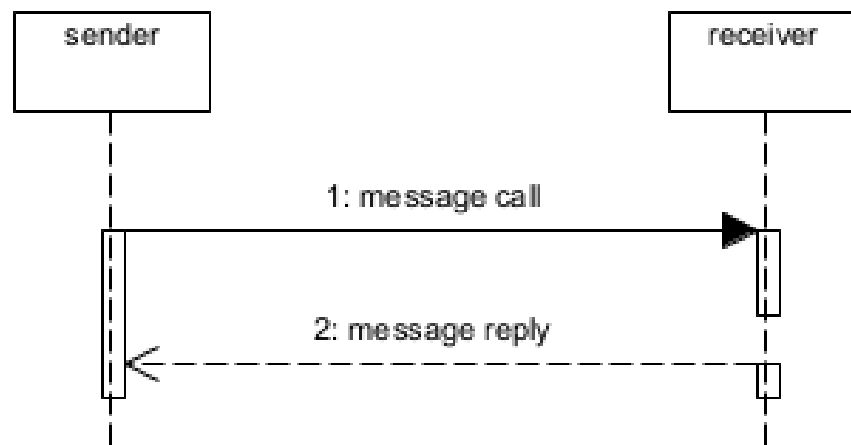
Sơ đồ lớp phân tích cho modul thêm phòng:



# Phân tích động (1)

## Biểu đồ tuần tự – Sequence Diagram:

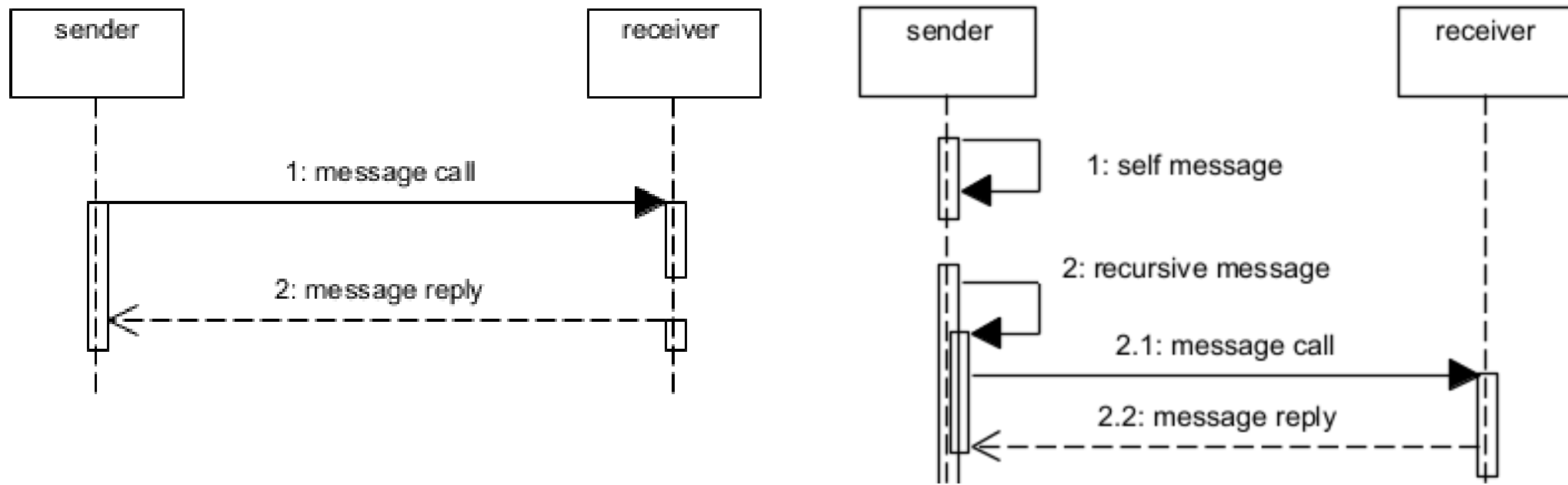
- Biểu diễn thứ tự của các hành động → Thể hiện luồng chương trình
- Bao gồm các đối tượng và các lớp liên quan đến scenario
- Trong VP: *Diagram > New > Sequence Diagram*



# Phân tích động (2)

## Biểu đồ tuần tự – Sequence Diagram:

- Các đối tượng giao tiếp với nhau qua các message



# Phân tích động (3)

Từ scenario chuẩn đã viết:

- Các chủ thể hành động từ kịch bản trước 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
- > Vẽ sơ đồ tuần tự phân tích dựa trên kịch bản viết lại

# Phân tích động (4)

Modul thêm phòng:

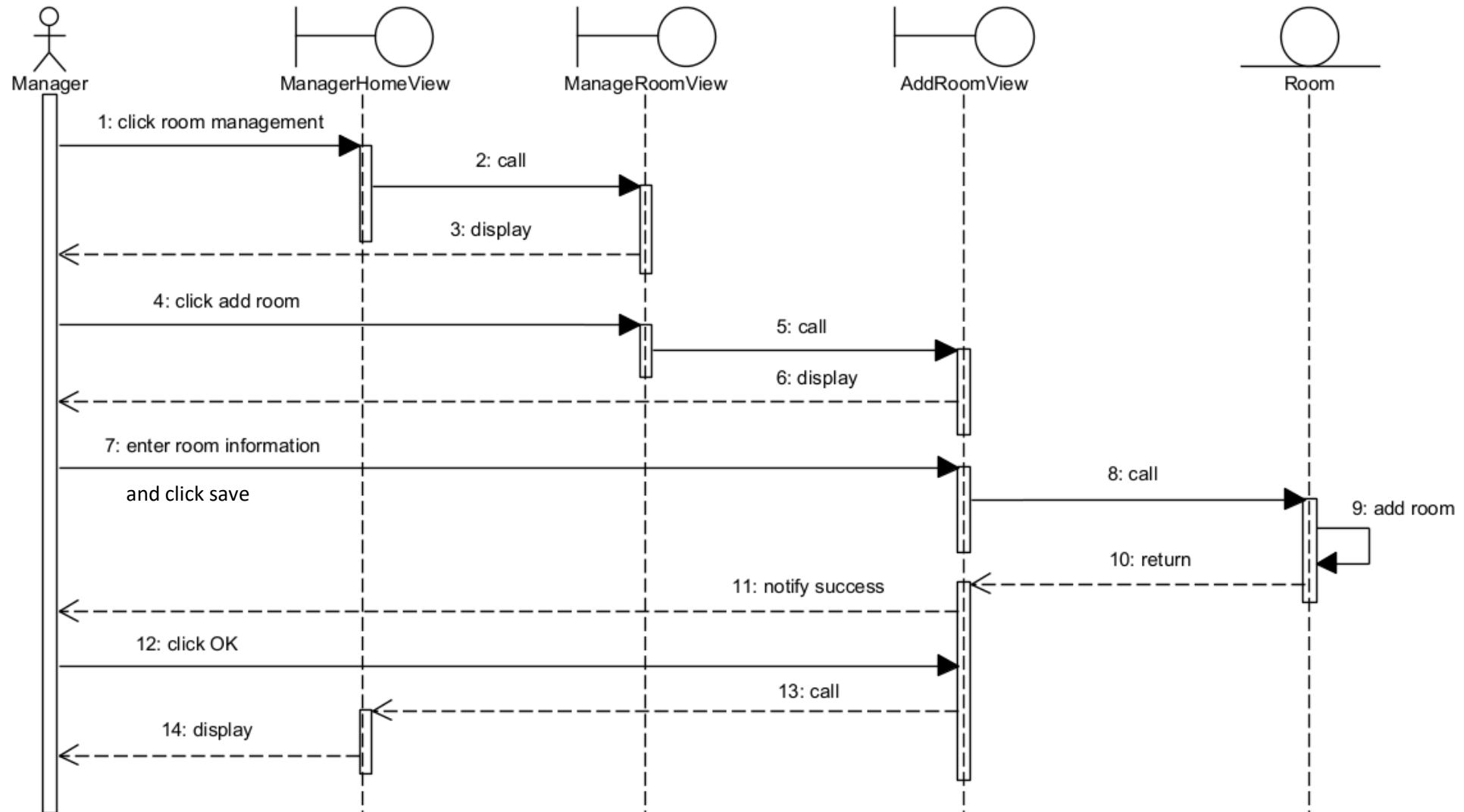
1. User/Manager chọn chức năng quản lý phòng từ ManagerHomeView (do lớp User sử dụng chung cho các nhân viên, User ở đây hiểu là Manager)
2. Lớp ManagerHomeView gọi lớp ManageRoomView yêu cầu hiển thị
3. Lớp ManageRoomView hiển thị cho Manager
4. Manager click vào nút thêm phòng
5. Lớp ManageRoomView gọi lớp AddRoomView yêu cầu hiển thị
6. Lớp AddRoomView hiển thị cho Manager với các ô nhập thông tin phòng mới

# Phân tích động (5)

7. Manager nhập các thông tin phòng mới vào các ô và click nút lưu
8. Lớp AddRoomView gọi lớp Room yêu cầu xử lý
9. Lớp Room gọi phương thức thêm phòng
10. Lớp Room trả lại kết quả cho lớp AddRoomView
11. Lớp AddRoomView hiện thông báo thành công
12. Manager click vào nút OK của thông báo
13. Lớp AddRoomView gọi lại lớp ManagerHomeView
14. Lớp ManagerHomeView hiển thị cho Manager



# Phân tích động (6)



# Bài tập tại lớp

- Bài tập nhóm: Từ đề tài đã đăng ký:
  - Thực hiện trích các lớp thực thể toàn hệ thống (theo các bước trong slide) và vẽ sơ đồ lớp thực thể phân tích
  - Nộp file pdf qua email. Title: NMCNPM – nhóm lớp – nhóm BTL – BT nhóm số 3 (*VD: NMCNPM - 1 - 2 - BT nhóm số 3*)
  - Tên file là số thứ tự nhóm BTL (*VD: 1.pdf*), trong file ghi tên các thành viên nhóm và tên đề tài



- 51



- 52