

# Nhập môn công nghệ phần mềm

## Thiết kế hướng đối tượng

GV: ThS. Ngô Tiến Đức

# Nội dung chính

- Luồng thiết kế
- Thiết kế lớp thực thể
- Thiết kế CSDL
- Thiết kế giao diện
- Mô hình MVC
- Thiết kế tĩnh
- Thiết kế động

# Luồng thiết kế (1)

Mục đích:

- Chuyển tài liệu phân tích từ dạng đặc tả nghiệp vụ hệ thống sang dạng có thể cài đặt và kiểm thử được
- Xác định các yêu cầu phi chức năng:
  - Chọn ngôn ngữ lập trình
  - Tính tái sử dụng - reusability
  - Tính khả chuyển - portability

# Luồng thiết kế (2)

## Tính tái sử dụng:

- Sử dụng các thành phần đã có để phát triển các thành phần khác
  - Đoạn code, modul, thiết kế, dữ liệu kiểm thử
- Hai loại tái sử dụng:
  - Opportunistic/Accidental reuse
  - Deliberate reuse

# Luồng thiết kế (3)

## **Tính khả chuyển:**

- Việc sửa đổi sản phẩm để chạy trên một môi trường khác dễ hơn so với việc xây dựng lại từ đầu
  - Hệ điều hành
  - Phần cứng
  - Trình biên dịch

# Luồng thiết kế (4)

## **Các bước thực hiện thiết kế:**

- Vẽ biểu đồ lớp thực thể thiết kế
- Thiết kế CSDL
- Thiết kế giao diện
- Thiết kế tĩnh: Vẽ sơ đồ lớp chi tiết
- Thiết kế động: Vẽ sơ đồ tuần tự

# Thiết kế lớp thực thể (1)

Từ biểu đồ lớp thực thể phân tích:

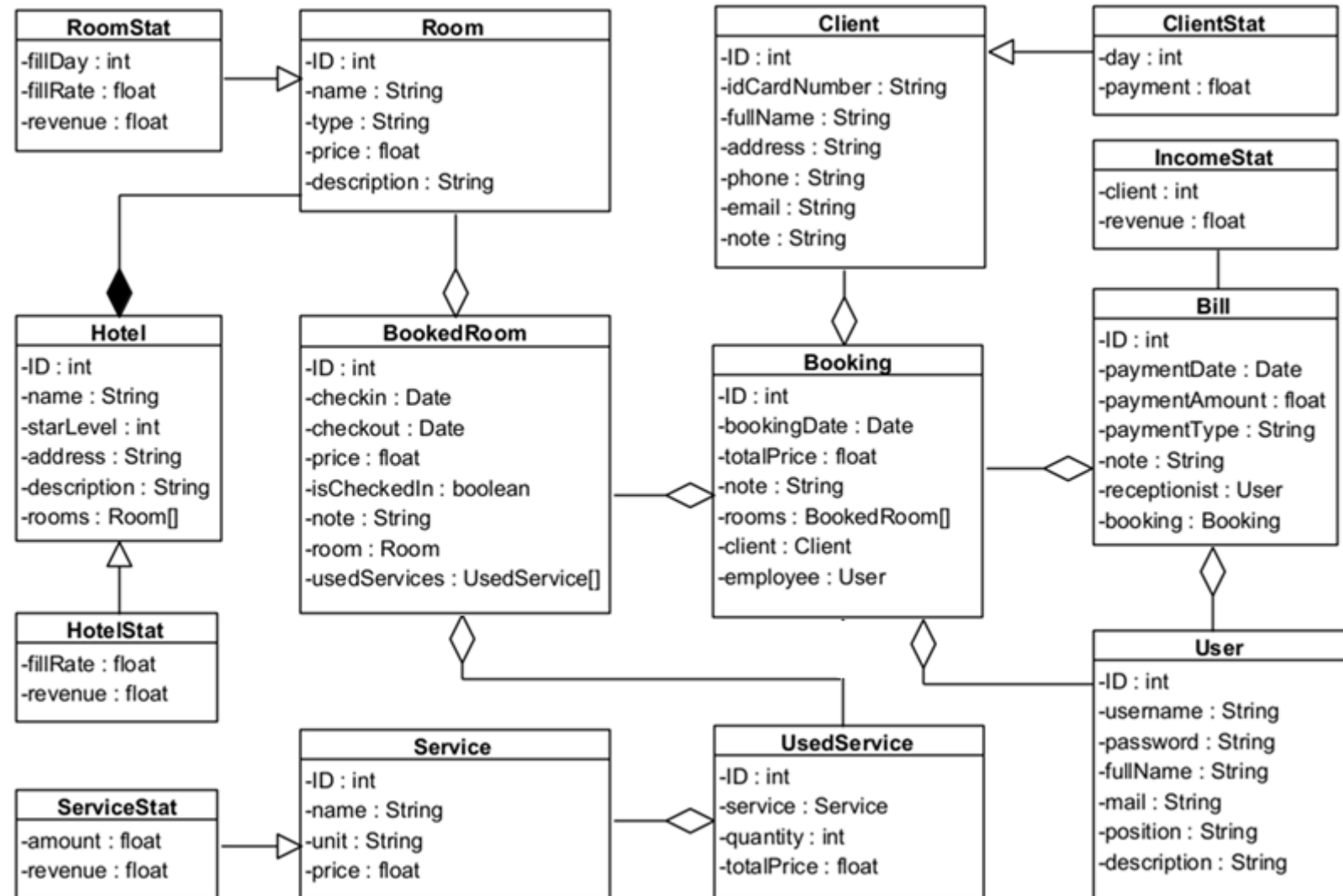
- Thêm thuộc tính id cho các lớp không kế thừa từ lớp khác
- Bổ sung kiểu dữ liệu cho các thuộc tính
- Chuyển các quan hệ association thành aggregation
- Bổ sung các thuộc tính tương ứng với quan hệ thành phần





# Thiết kế lớp thực thể (3)

Biểu đồ lớp thực thể:



# Thiết kế CSDL (1)

Xây dựng CSDL từ sơ đồ lớp thực thể của hệ thống:

- Bước 1: Đề xuất các bảng tương ứng với các lớp thực thể
- Bước 2: Xác định các thuộc tính
- Bước 3: Biểu diễn quan hệ giữa các bảng
- Bước 4: Định nghĩa khóa chính và khóa ngoại
- Bước 5: Mìn hóa

Trong VP: *Diagram > New > Entity Relationship Diagram*

# Thiết kế CSDL (2)

## Bước 1:

- Lớp Hotel → bảng tblHotel: Lưu các thông tin về khách sạn bao gồm ID, tên, địa chỉ, hạng sao, mô tả
- Lớp Room → bảng tblRoom: Lưu các thông tin cố định về phòng bao gồm ID, tên, loại phòng, giá hiển thị, mô tả
- Lớp User → bảng tblUser: Lưu các thông tin về người dùng hệ thống bao gồm ID, tên đăng nhập, mật khẩu, họ tên, SĐT, email, chức vụ, mô tả

# Thiết kế CSDL (3)

## Bước 1 (Tiếp):

- Bảng tblClient: Lưu các thông tin về khách đặt phòng...
- Bảng tblService: Lưu các thông tin về dịch vụ...
- Bảng tblBookedRoom: Lưu các thông tin về phòng được đặt...
- Lớp tblBooking: Lưu các thông tin đặt phòng...
- Lớp tblUsedService: Lưu các thông tin dịch vụ khách hàng sử dụng...
- Lớp tblBill: Lưu các thông tin về hóa đơn...

# Thiết kế CSDL (4)

## Bước 2:

- Bảng tblHotel có các thuộc tính tương ứng từ biểu đồ lớp: ID kiểu *int*, name kiểu *varchar*, starLevel kiểu *int*, address kiểu *varchar*, description kiểu *varchar*
- Bảng tblRoom có các thuộc tính: ID kiểu *varchar*, type kiểu *varchar*, price kiểu *float*, description kiểu *varchar*

Áp dụng tương tự với các bảng còn lại

# Thiết kế CSDL (5)

## Bước 3:

- Lớp Hotel và lớp Room có quan hệ thành phần, một Hotel có nhiều Room → Bảng tblHotel và tblRoom có quan hệ 1-n
- Thuộc tính rooms kiểu *Room[]* của Hotel sẽ không nằm trong bảng tblHotel mà được thể hiện qua hotelID trong bảng tblRoom

Áp dụng tương tự với các bảng còn lại

# Thiết kế CSDL (6)

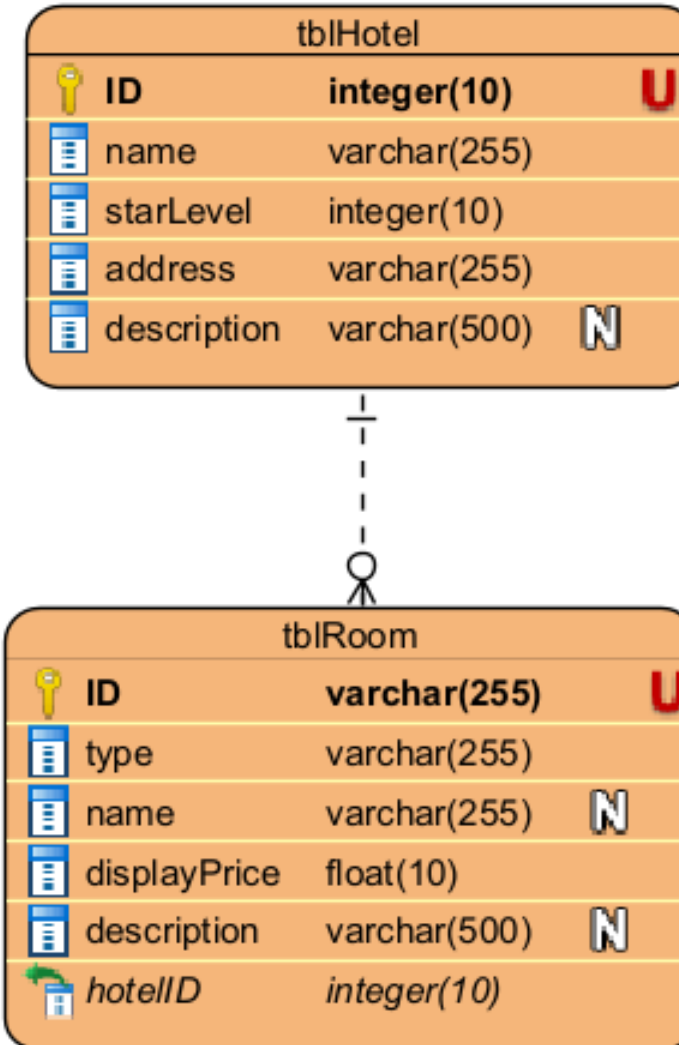
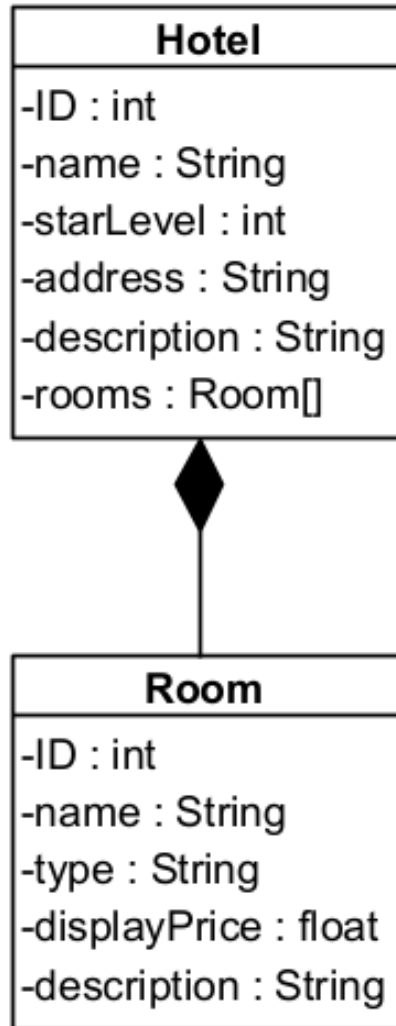
## **Bước 4:**

- Bảng tblHotel có ID là khóa chính
- Bảng tblRoom có ID là khóa chính và hotelID là khóa ngoại

Áp dụng tương tự với các bảng còn lại

# Thiết kế CSDL (7)

Ví dụ:

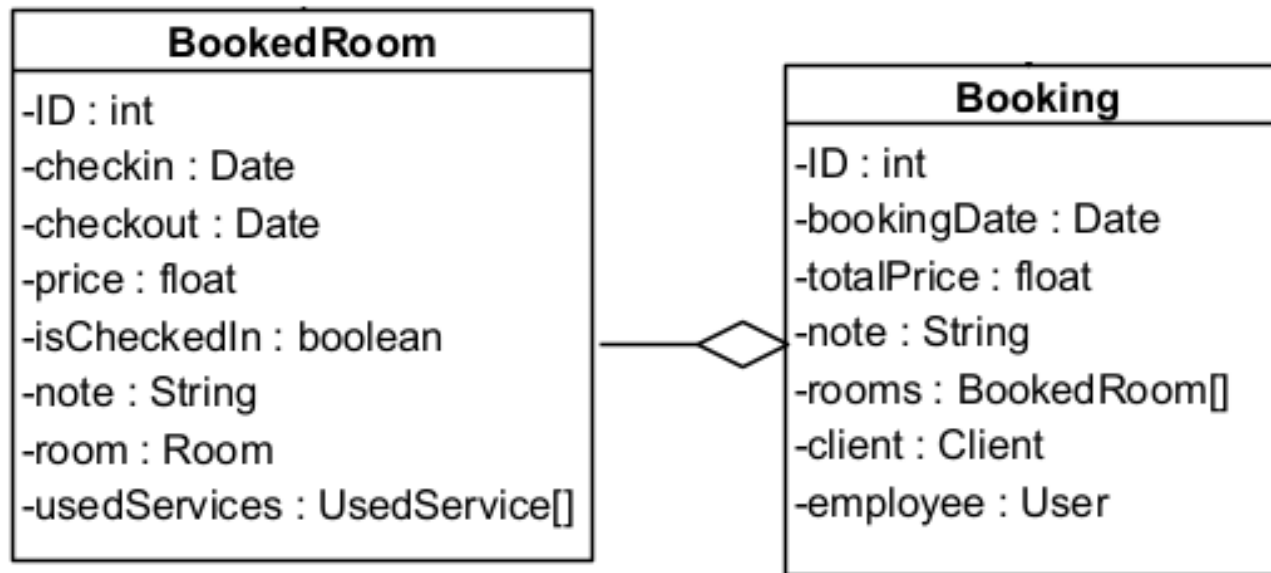




# Thiết kế CSDL (8)

## Bước 5:

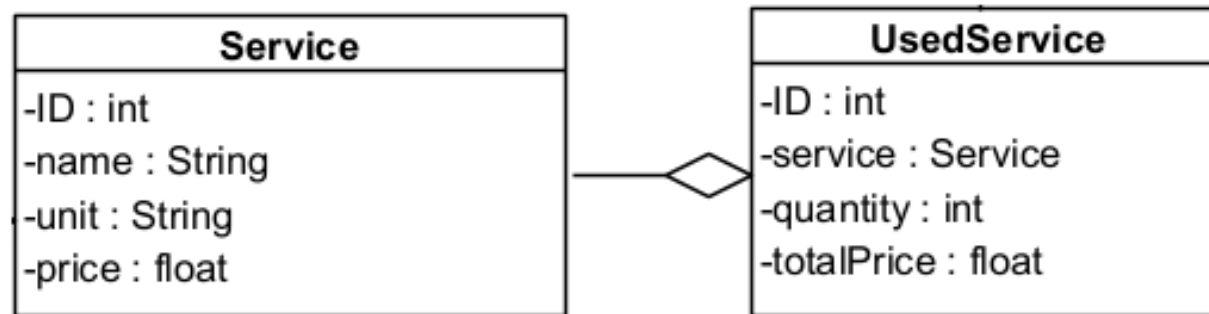
- Trong Booking thì tổng tiền có thể tính qua ngày nhận phòng, ngày trả phòng và giá của phòng được đặt -> bỏ totalPrice (có thể thay bằng discount)



# Thiết kế CSDL (9)

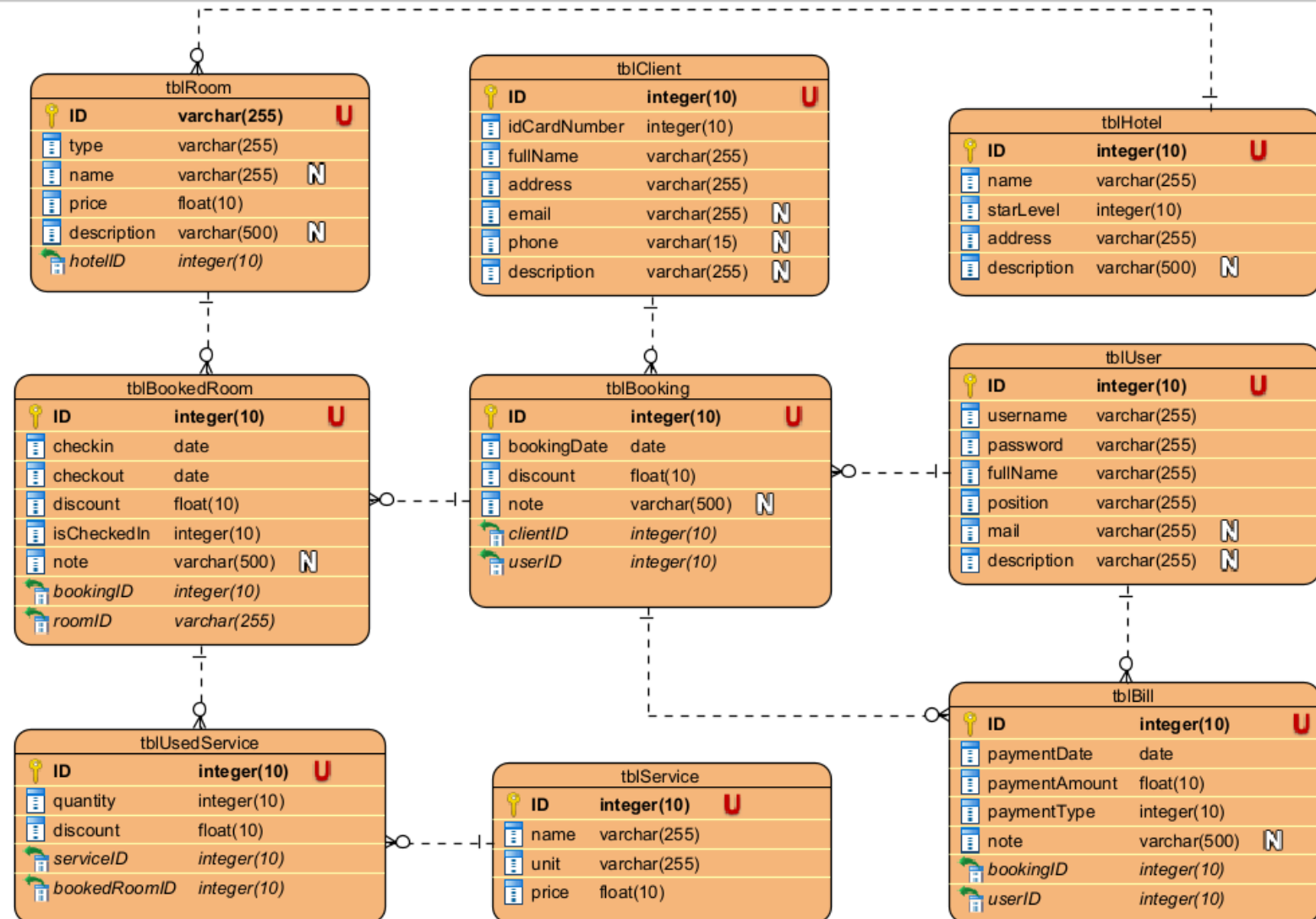
## Bước 5:

- Trong UsedService thì tổng tiền có thể tính qua đơn giá và đơn vị từ Service và kết hợp với số lượng sử dụng -> bỏ totalPrice (có thể thay bằng discount)



# Thiết kế CSDL (10)

Kết quả CSDL:



# Thiết kế giao diện

Ví dụ modul thêm phòng của người quản lý:

Hotel Reservation - Manager

Manage room

View report

Manage room

Add room

Update room

Delete room

Add room

ID

Name

Type

Display price

Description

Save Cancel

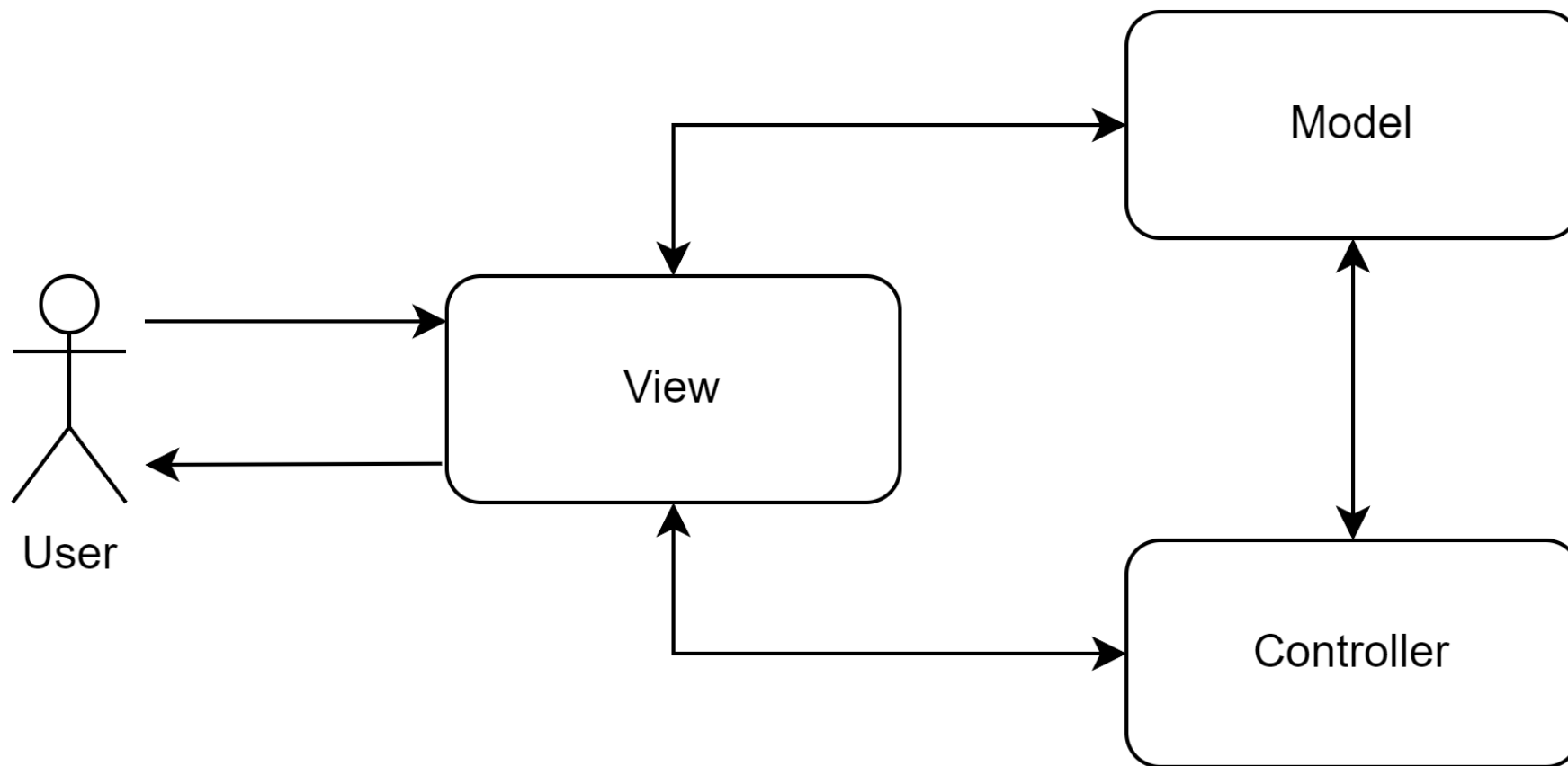


- 21

# Mô hình MVC (2)

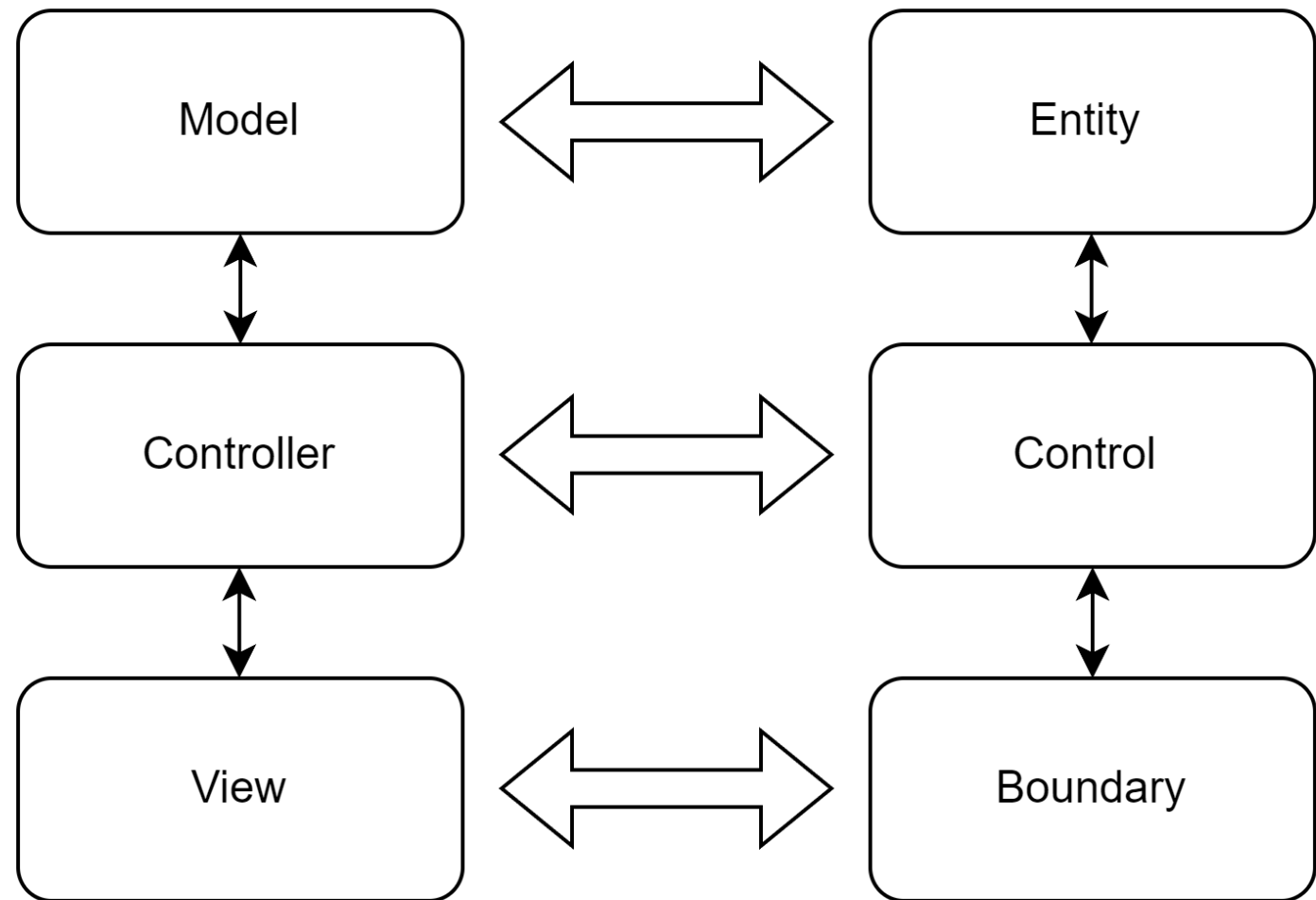
- C - Controller:
  - Định nghĩa các hành vi, hoạt động, xử lý của hệ thống
  - Đối chiếu hành động của người dùng (nhận từ view) vào tập chức năng để xử lý
  - Chọn view để hiển thị cho người dùng

# Mô hình MVC (3)



# Mô hình MVC (4)

- Mô hình MVC cải tiến:



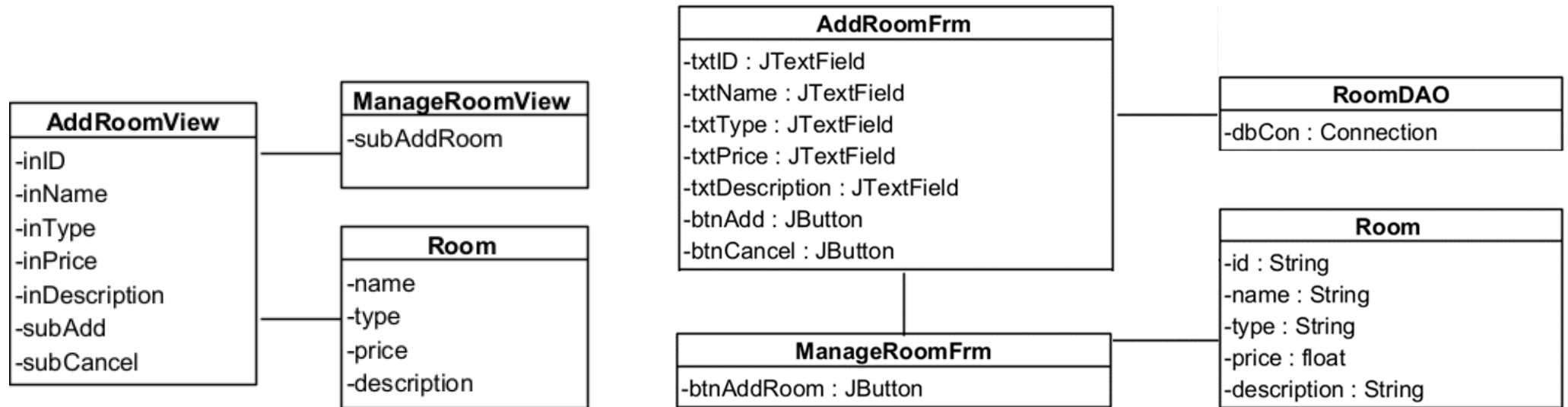




- Các lớp thực thể chỉ chứa các thuộc tính và các phương thức get/set cho mỗi thuộc tính
- Các thao tác liên quan đến CSDL đặt trong lớp điều khiển dạng DAO

# Thiết kế tĩnh (2)

Thêm thuộc tính cho các lớp giao diện và lớp điều khiển dạng DAO:



Sơ đồ lớp phân tích

Sơ đồ lớp thiết kế (Java concept)

# Thiết kế tĩnh (3)

Định nghĩa các phương thức cho lớp:

- Nguyên lý 1: Các thuộc tính của lớp phải để dạng private → cần các phương thức get/set
- Nguyên lý 2: Nếu có nhiều đối tượng **X** gọi đến một hành động **k** của đối tượng **Y**, thì phương thức để thực hiện hành động **k** nên gán cho lớp của đối tượng **Y**
- Nguyên lý 3: Nếu một hành động mà không thể gán thành phương thức cho lớp khác, thì lớp của đối tượng cần thực hiện hành động phải chứa phương thức tương ứng với hành động đó

# Thiết kế tĩnh (4)

Ví dụ với thực thể Room:

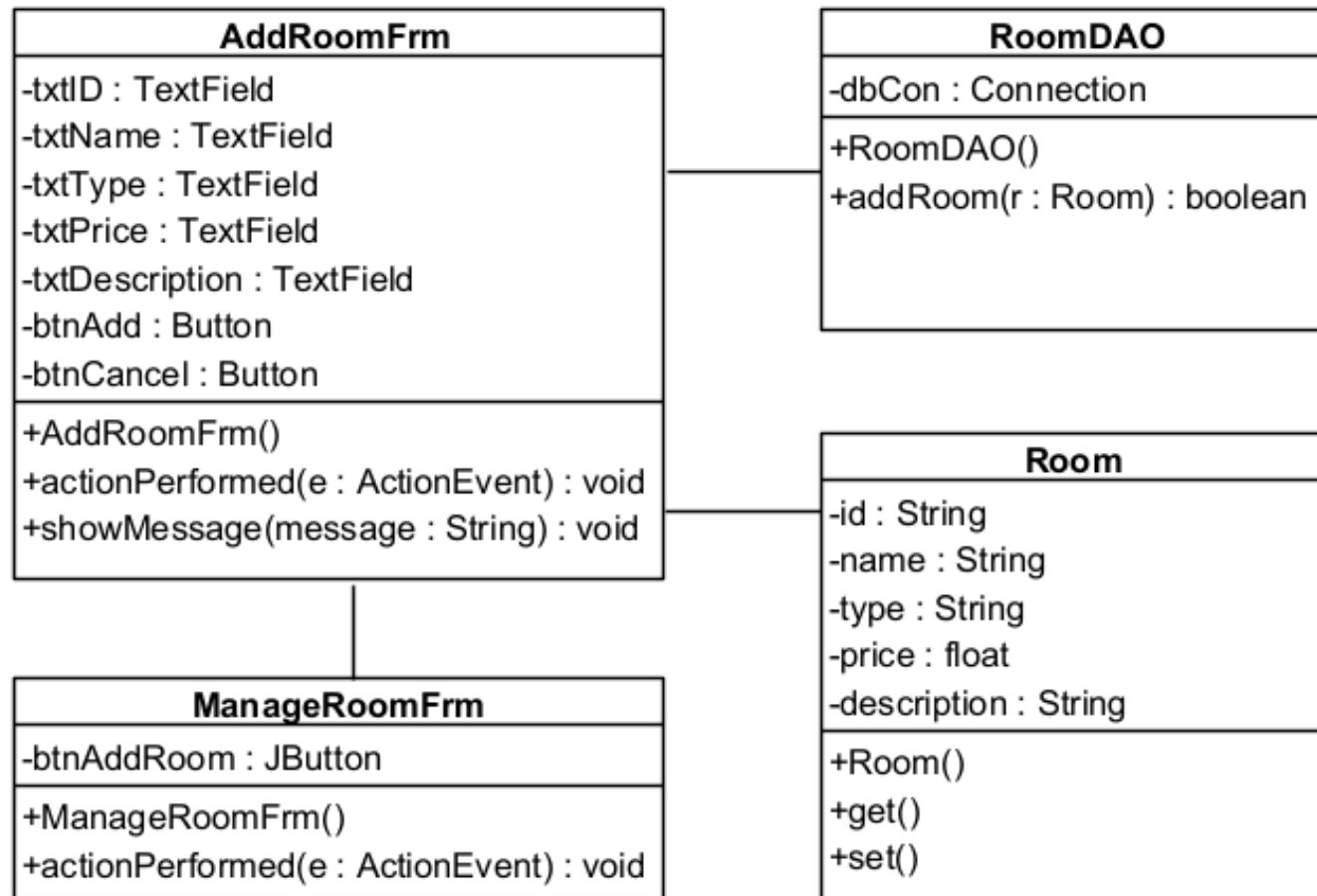
```
private String name;  
  
...  
  
public Room() {...} // constructor  
  
public void setName(String name) {...} // setter  
  
public String getName() {...} // getter
```



- Lớp ManageRoomFrm phải có các phương thức: hiển thị form (hàm khởi tạo), xử lý sự kiện gọi lớp AddRoomFrm hiển thị
- Lớp AddRoomFrm phải có các phương thức: hiển thị form, xử lý sự kiện nút Save và nút Cancel bị click, thông báo thành công
- Lớp RoomDAO phải có phương thức lưu thông tin phòng vào CSDL
- Lớp Room phải có các phương thức đóng gói thông tin đối tượng (hàm khởi tạo) và các phương thức get/set

# Thiết kế tĩnh (6)

Kết quả sơ đồ lớp MVC chi tiết:



# Thiết kế động (1)

Vẽ sơ đồ tuần tự thiết kế:

- Viết lại scenario tương tự như pha phân tích, thay thế các lớp và hành động cũ bằng các lớp và phương thức vừa trích được
- Ví dụ scenario cho modul thêm phòng của người quản lý:
  1. Manager (người quản lý) click nút thêm phòng từ ManageRoomFrm
  2. Lớp ManageRoomFrm kích hoạt phương thức actionPerformed() để yêu cầu lớp AddRoomFrm hiển thị
  3. Phương thức actionPerformed() gọi lớp AddRoomFrm

# Thiết kế động (2)

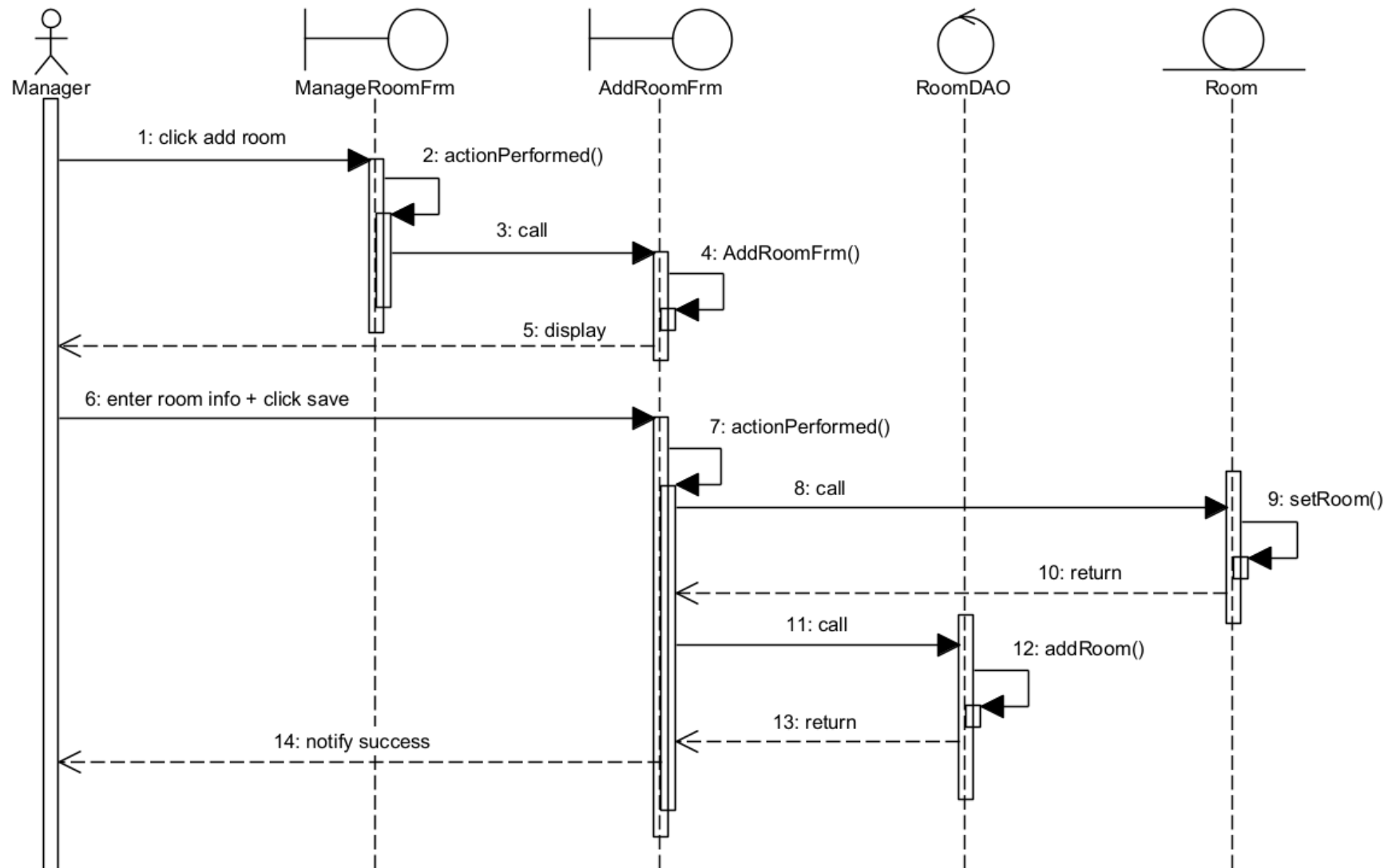
4. Hàm khởi tạo AddRoomFrm() được kích hoạt
5. Giao diện AddRoomFrm hiển thị cho Manager
6. Manager nhập thông tin phòng và click nút lưu
7. Lớp AddRoomFrm kích hoạt phương thức actionPerformed() để gọi lớp Room yêu cầu khởi tạo đối tượng Room
8. Phương thức actionPerformed() gọi lớp Room
9. Lớp Room gọi hàm set để đóng gói đối tượng từ AddRoomFrm thành một thực thể có kiểu dữ liệu Room



# Thiết kế động (3)

10. Thực thể được trả về cho phương thức actionPerformed()
  11. Phương thức actionPerformed() gọi phương thức addRoom() của lớp RoomDAO
  12. Phương thức addRoom() cập nhật thông tin vào mới vào CSDL
  13. Phương thức addRoom() trả quyền điều khiển về lại cho phương thức actionPerformed()
  14. Phương thức actionPerformed() hiện thông báo thành công cho Manager
- > Vẽ sơ đồ tuần tự thiết kế đủ 14 bước

# Thiết kế động (4)



# Bài tập tại lớp

- Bài tập nhóm: Từ đề tài đã đăng ký:
  - Vẽ sơ đồ thực thể (thiết kế) toàn hệ thống
  - Thiết kế CSDL cho toàn hệ thống
  - Nộp file pdf qua email. Title: NMCNPM – nhóm lớp – nhóm BTL – BT nhóm số 4 (*VD: NMCNPM - 1 - 2 - BT nhóm số 4*)
  - Tên file là số thứ tự nhóm BTL (*VD: 1.pdf*), trong file ghi tên các thành viên nhóm và tên đề tài



- 36

# Tài liệu tham khảo

- Stephen R. Schach. *Object-Oriented and Classical Software Engineering*. 8th Edition, WCB/McGraw-Hill, 2010
- T. Đ. Quế, N. M. Hùng. *Bài giảng Nhập môn công nghệ phần mềm*. HVCNBCVT, 2020