

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THÔNG THÔNG TIN



ĐỒ ÁN MÔN HỌC  
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU  
**QUẢN LÝ KHÁCH SẠN**

**Sinh viên:** 18520396 \_ Nguyễn Lê Mỹ Uyên  
18520403 \_ Lương Nguyễn Tường Vy  
18520466 \_ Nguyễn Vũ Tú Anh  
18520476 \_ Võ Thị Vân Anh

*Thành phố Hồ Chí Minh, Tháng 3/2020*

## MỤC LỤC

1	Chương 1: TỔNG QUAN ĐỀ TÀI .....	5
1.1	Đặt vấn đề.....	5
1.2	Mục tiêu.....	5
1.3	Công cụ sử dụng.....	6
2	Chương 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU .....	6
2.1.1	Mô tả bài toán .....	6
2.1.2	Phân tích hiện trạng và yêu cầu cho tương lai .....	7
2.1.2.1	Tìm hiểu sơ bộ về hệ thống cũ.....	7
2.1.2.2	Đánh giá hiện trạng.....	20
2.1.2.3	Yêu cầu mới.....	21
2.2	Mô hình dữ liệu quan hệ .....	22
2.2.1	Thiết kế mô hình dữ liệu quan hệ mức vật lý .....	22
2.2.1.1	Table LOAITAIKHOAN .....	22
2.2.1.2	Table NHANVIEN .....	22
2.2.1.3	Table LOAIPHONG .....	23
2.2.1.4	Table PHONG .....	23
2.2.1.5	Table KHACHHANG .....	24
2.2.1.6	Table DICHVU.....	24
2.2.1.7	Table MONAN .....	25
2.2.1.8	Table CSVC.....	25
2.2.1.9	Table PHIEUNHANPHONG .....	25
2.2.1.10	Table PHIEUDATTRUOC.....	26
2.2.1.11	Table NGAYLE.....	26
2.2.1.12	Table HOADONTANTHOAN.....	26
2.2.1.13	Table BANGKE_DV.....	27
2.2.1.14	Table BANGKE_MONAN .....	27
2.2.1.15	Table HOADON_CSVC .....	27
2.2.2	Ràng buộc toàn vẹn.....	27
2.2.2.1	Bảng NHANVIEN.....	27
2.2.2.2	Bảng LOAIPHONG .....	31
2.2.2.3	Bảng PHONG .....	32
2.2.2.4	Bảng PHIEUDATTRUOC .....	33
2.2.2.5	Bảng PHIEUNHANPHONG.....	34

2.2.2.6	Bảng KHACHHANG .....	36
2.2.2.7	Bảng HOADONTANTHOAN .....	38
2.2.2.8	Bảng BANGKE_DICHVU.....	41
2.2.2.9	Bảng DICHVU .....	42
2.2.2.10	Bảng BANGKE_MONAN .....	43
2.2.2.11	Bảng MONAN.....	44
2.2.2.12	Bảng HOADON_CSVC .....	45
2.2.2.13	Bảng CSVC .....	46
2.2.2.14	Bảng LOAITAIKHOAN .....	46
3	Chương 3: CÁC CÔNG NGHỆ SỬ DỤNG.....	47
3.1	Oracle .....	47
3.2	JDBC (Java Database Connectivity) .....	47
3.3	Java Swing.....	48
3.4	Netbeans .....	48
4	CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC.....	49
4.1	Trigger trong Oracle.....	49
4.1.1	Danh sách các trigger.....	49
4.1.2	Mô tả một số Trigger .....	52
4.2	Stored Procedure: .....	54
4.2.1	Stored Procedure trong Oracle:.....	54
4.2.2	Transaction trong Oracle.....	54
4.2.3	Danh sách Stored Procedure .....	55
4.3	Mô tả một số Stored Procedure .....	56
4.3.1	Cập nhật thông tin Phiếu đặt trước .....	56
4.3.2	Thanh toán hóa đơn.....	61
4.3.3	Tính tiền bảng kê món ăn.....	62
4.3.4	Tính tiền phòng .....	63
5	CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI .....	64
5.1	Các mức độ lập trong Oracle.....	64
5.1.1	Read Committed .....	64
5.1.2	Serializable.....	64
5.1.3	Read-Only .....	65
5.2	Cơ chế khóa.....	65
5.3	Một số vấn đề xảy ra khi nhiều giao tác truy xuất đồng thời .....	66
5.3.1	Dirty reads.....	66

5.3.2	Non – repeatable reads .....	66
5.3.3	Phantom reads .....	66
5.3.4	Lost update .....	66
5.3.5	Deadlock .....	66
5.4	Mô tả đồ án môn học .....	66
5.4.1	Lost update .....	66
5.4.2	Non - repeatable read: .....	71
5.4.3	Phantom read .....	73
5.4.4	Deadlock .....	76
6	Chương 6: THIẾT KẾ GIAO DIỆN .....	77
6.1	Danh sách các màn hình .....	77
6.1.1	Đăng nhập .....	77
6.1.2	Manager .....	77
6.1.3	Receptionists .....	79
6.2	Mô hình các màn hình chính .....	80
6.2.1	Đăng nhập .....	80
6.2.2	Manager .....	81
6.2.3	Quản lý khách hàng .....	82
6.2.4	Quản lý phiếu đặt trước .....	83
6.2.5	Thêm phiếu đặt trước .....	84
6.2.6	Cập nhật phiếu đặt trước .....	85
6.2.7	Quản lý dịch vụ .....	87
6.2.8	Thêm dịch vụ .....	88
6.2.9	Cập nhật dịch vụ .....	89
6.2.10	Quản lý món ăn .....	90
6.2.11	Thêm món ăn .....	91
6.2.12	Cập nhật món ăn .....	92
6.2.13	Quản lý Cơ sở vật chất .....	93
6.2.14	Thêm Cơ sở vật chất .....	94
6.2.15	Cập nhật Cơ sở vật chất .....	95
6.2.16	Quản lý hóa đơn thanh toán .....	96
6.2.17	Hóa đơn thanh toán .....	97
6.2.18	Dặt dịch vụ .....	98
6.2.19	Xóa bảng kê dịch vụ .....	99
6.2.20	Dặt món ăn .....	100

## Đồ án HQT \_ Quản Lý Khách sạn

6.2.21	Xóa bảng kê món ăn .....	101
6.2.22	Thêm hóa đơn CSVC .....	102
7	Chương 7: KẾT LUẬN .....	102
7.1	Kết quả đạt được .....	102
7.2	Hạn chế.....	103
7.3	Hướng phát triển .....	103
8	Phụ lục 1: BẢNG PHÂN CHIA CÔNG VIỆC .....	103
9	Phụ lục 2: TÀI LIỆU THAM KHẢO .....	104

## Chương 1: TỔNG QUAN ĐỀ TÀI

### 1.1 Đặt vấn đề

Hiện nay, các quốc gia trên thế giới đều chú trọng vào lĩnh vực công nghệ thông tin, đặc biệt là các cường quốc hiện tại và các quốc gia đang phát triển như Việt Nam. Kéo theo sự phát triển không ngừng đó là sự áp dụng công nghệ vào trong nhiều lĩnh vực khác như khách sạn. Các phần mềm thường được thiết kế với giao diện thân thiện với người dùng cùng cách tính năng giúp người dùng quản lý hệ thống của mình tự động, dễ dàng và logic hơn.

Do vậy mà trong việc phát triển phần mềm, đòi hỏi không chỉ là sự chính xác, xử lý được nhiều nghiệp vụ thực tế mà còn phải đáp ứng các yêu cầu khác như về tốc độ, giao diện thân thiện, mô hình hoá được thực tế vào máy tính để người sử dụng tiện lợi, quen thuộc, tính tương thích cao, bảo mật cao (đối với các dữ liệu nhạy cảm), ... Các phần mềm giúp tiết kiệm một lượng lớn thời gian, công sức của con người, và tăng độ chính xác và hiệu quả trong công việc (nhất là việc sửa lỗi và tự động đồng bộ hoá). Đối với công việc quản lý khách sạn, nếu không có sự hỗ trợ của các chức năng trong hệ thống, việc quản lý cần mất rất nhiều thời gian, nhân lực, bị chia nhỏ thành nhiều bộ phận (quản lý đặt phòng, nhận phòng, các dịch vụ, món ăn ...). Các công việc như đặt phòng thì khách hàng cần phải có trong dnah sách khách hàng của khách sạn nếu như không có sự hỗ trợ từ hệ thống để phải tra cứu trên giấy tờ rất mất thời gian và dễ xảy ra sai sót. Nhờ có một hệ thống quản lý, người dùng sẽ dễ dàng hơn trong công tác quản lý cũng như một số tính năng tự động hóa như tính tiền, phát hiện đặt trùng phòng...

Ứng dụng phần mềm quản lý khách sạn thì khá phổ biến tại nhiều khách sạn hiện tại. Tuy nhiên đa phần hệ thống chỉ được dùng để quản lý, lưu trữ và tra cứu mà không có những tác vụ tự động hóa hay ngăn chặn các tính huống cập nhật đồng thời.

Do đó hệ thống cần thiết phải giải quyết được các vấn đề đáy đồng thời phải được xây dựng giao diện dễ tương tác với người dùng.

### 1.2 Mục tiêu

Xây dựng một hệ thống ứng dụng hoàn thiện để lê tân và ban giám đốc có thể trao đổi với hệ thống một cách dễ dàng và thuận tiện. Cung cấp chức năng riêng cho một bộ phận nào đó như: người quản lý thì có thể quản lý thông tin dịch vụ, món ăn, CSVC nhưng lê tân thì chỉ có thể tra cứu các thông tin ấy.

Trong hệ thống sẽ phải đáp ứng các yêu cầu sau:

- Quản lý thông tin nhân viên, khách hàng, loại phòng, phòng, dịch vụ, món ăn, CSVC.
- Quản lý các hoạt động của lê tân: đặt phòng, nhận phòng, yêu cầu dịch vụ, đặt món ăn hay thêm hóa đơn CSVC, quản lý khách hàng, xem thông tin dịch vụ, món ăn, CSVC.
- Đảm bảo việc tìm kiếm và xem thông tin của khách hàng, thông tin của các bộ phận khác: các thông tin về phòng, hóa đơn, bảng kê, phiếu đặt trước, phiếu nhận phòng.
- Quản lý việc thanh toán hóa đơn
- Quản lý các hoạt động kê doanh thu, số lượng đặt phòng của từng loại phòng, những món ăn được yêu cầu nhiều trong tháng.
- Thông tin lưu trữ sao cho việc xuất dữ liệu được nhanh chóng, hiệu quả.
- Hệ thống hỗ trợ phân quyền người dùng, đảm bảo tính bảo mật an toàn của hệ thống.

### 1.3 Công cụ sử dụng

Trong quá trình thực hiện, nhóm đã sử dụng một số phần mềm phục vụ cho việc tìm và xây dựng đề tài:

- Oracle developer
- Netbeans
- Ireport
- Power designer

## Chương 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU

### 2.1.1 Mô tả bài toán

“Paradis” Thiên đường là một khách sạn cỡ lớn nhiều phòng.

Vì đang sử dụng hệ quản trị cũng như phần mềm không thuận tiện, gặp nhiều khó khăn trong việc quản lý thông tin đặt phòng trước, theo dõi sự lưu trú, yêu cầu dịch vụ cũng như việc thanh toán khi rời đi. Để rút ngắn thời gian, tăng hiệu suất làm việc vì vậy ban giám đốc muốn tin học hóa công việc của phòng tiếp tân.

Theo yêu cầu, hệ thống phải đảm bảo xử lý việc đặt chỗ trước của khách một cách nhanh chóng. Đảm bảo rằng sau khi phân tích yêu cầu của khách (loại phòng, vị trí, số lượng phòng), nhân viên có thể truy cập hệ thống để tham khảo danh sách phòng trống phù hợp các yêu cầu của khách để tiến hành đặt phòng cũng như sẽ lưu trữ các thông tin đặt chỗ của khách hàng trên hệ thống: họ tên, giấy tờ tùy thân, ngày đến và rời khách sạn...

Khi khách hàng đến, nhân viên sẽ kiểm tra hồ sơ đặt chỗ trước của khách hàng trên hệ thống, nếu có thông tin đặt chỗ trước, nhân viên sẽ lưu giữ giấy tờ tùy thân để tiện cho việc khai báo tạm trú tạm vắng, tiến hành bố trí phòng ở cho khách cũng như hoàn thành các thông tin cần lưu trữ trên hệ thống: họ tên, cmnd, sdt, email, giới tính, ngày sinh của người đại diện thanh toán và số người đi cùng. Ngoài ra, cập nhật tình trạng phòng đã bố trí cho khách hàng.

Không chỉ cung cấp chỗ ở, khách sạn còn cung cấp một số dịch vụ cho khách hàng như: giặt ủi, gọi điện thoại, karaoke, spa, phòng xông hơi, thuê xe máy, xe 4 chỗ, phòng gym. Nếu có yêu cầu dịch vụ, khách hàng liên hệ với quầy tiếp tân để tiếp tân sẽ lập một bảng kê các dịch vụ trên hệ thống. Bảng kê bao gồm các dữ liệu: số thứ tự, tên dịch vụ, vào thời điểm nào, chi phí của dịch vụ. Khách hàng có thể thanh toán ngay khi sử dụng dịch vụ hoặc thanh toán khi trả phòng. Việc gọi món cũng là một dịch vụ đặc biệt ở khách sạn. Khách hàng yêu cầu món ăn. Từ đó nhân viên tiếp tân sẽ liên hệ với quầy dịch vụ hay quầy bếp để thông báo yêu cầu của khách hàng và sẽ xác nhận thời gian sử dụng dịch vụ tới khách hàng.

Cũng như nhận phòng, trả phòng được thực hiện ở quầy tiếp tân. Nhân viên sẽ tổng hợp các hóa đơn dịch vụ, hóa đơn ăn uống, hóa đơn cơ sở vật chất (nếu có), tiền phòng và đưa ra hóa đơn tổng số tiền khách hàng cần trả. Sau khi thanh toán (tiền mặt, quẹt thẻ) thì nhân viên sẽ trả lại giấy tờ tùy thân mà khách sạn đã giữ lại khi khách hàng nhận phòng. In hóa đơn, giao cho khách hàng, ký xác nhận vào phiếu đã thu.

Theo nhu khách sạn, khách sạn không phân khách hàng thành khách hàng thành viên hay khách vãng lai. Cũng như các ưu đãi cho khách hàng thành viên. Ý kiến của ban giám đốc, khách sạn sẽ

## Đồ án HQT \_ Quản Lý Khách sạn

khuyến mãi vào những ngày lễ, những ngày được quy định sẵn. Việc khuyến mãi đó sẽ áp dụng với tất cả các khách đặt phòng cũng như lưu trú ở khách sạn tại thời điểm đó.

### 2.1.2 Phân tích hiện trạng và yêu cầu cho tương lai

#### 2.1.2.1 Tìm hiểu sơ bộ về hệ thống cũ

##### 2.1.2.1.1 Tổng hợp các xử lý

Dự án: Quản lý nhà hàng khách sạn 'Paradis' Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Tra cứu đặt trước</b>	Số thứ tự: 1 Ngày lập: 17/3/2020
<p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p> <p>4. Kết quả <b>đầu ra</b>:</p> <p>5. <b>Nơi sử dụng</b>:</p> <p>6. <b>Những quy tắc điều kiện</b> đi kèm:</p>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp tân muốn tra cứu thông tin khách hàng đặt chỗ trước</li> <li>- Khi khách hàng nhận phòng và có liên hệ đặt trước</li> <li>- Ban lãnh đạo có nhu cầu tra cứu thông tin khách hàng đặt chỗ trước</li> <li>- Ban lãnh đạo và nhân viên tiếp tân chọn chức năng tra cứu thông tin cơ sở vật chất</li> <li>- Thông tin khách hàng đặt chỗ trước hiển thị trên màn hình hệ thống</li> <li>- Giao diện chức năng tra cứu đặt trước của hệ thống</li> <li>- Chỉ có ban lãnh đạo và nhân viên tiếp tân nhân viên trong khách sạn đều sử dụng được chức năng này</li> </ul>	

Dự án: Quản lý nhà hàng khách sạn 'Paradis' Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Quản lí đặt trước</b>	Số thứ tự: 2 Ngày lập: 17/3/2020
<p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp tân có thể thêm, xóa hoặc cập nhật thông tin khách hàng đặt chỗ trước</li> <li>- Ban lãnh đạo và nhân viên tiếp tân truy cập vào hệ thống và có nhu cầu quản lý thông tin khách hàng đặt chỗ trước</li> <li>- Nhân viên tiếp tân chọn chức năng quản lí thông tin khách hàng đặt chỗ trước</li> </ul>	

Đồ án HQT \_ Quản Lý Khách sạn

<p><b>4. Kết quả đầu ra:</b></p> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p>	<ul style="list-style-type: none"> <li>- Giao diện quản lý thông tin khách hàng đặt chỗ trước thể hiện trên hệ thống</li> <li>- Giao diện chức năng quản lý đặt trước của hệ thống</li> <li>- Ban lãnh đạo khách sạn và nhân viên tiếp tân mới có quyền truy cập chức năng này</li> </ul>
---	---

<p>Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p> <p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p> <p><b>3. Thông tin đầu vào:</b></p> <p><b>4. Kết quả đầu ra:</b></p> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p>	<p>Công việc/ Xử lý: <b>Tra cứu lưu trú</b></p> <p>Số thứ tự: 3 Ngày lập: 17/3/2020</p> <ul style="list-style-type: none"> <li>- Nhân viên tiếp tân và ban lãnh đạo muốn tra cứu thông tin khách hàng lưu trú</li> <li>- Khách hàng muốn lưu trú ở khách sạn</li> <li>- Ban lãnh đạo và nhân viên tiếp tân muốn truy cập hệ thống và nhu cầu quản lý thông tin cơ sở vật chất của khách sạn</li> <li>- Ban lãnh đạo và nhân viên tiếp tân chọn chức năng tra cứu thông tin khách hàng lưu trú</li> <li>- Thông tin khách hàng lưu trú hiển thị trên màn hình hệ thống</li> <li>- Giao diện chức năng tra cứu lưu trú của hệ thống</li> <li>- Chỉ có ban lãnh đạo và nhân viên tiếp tân nhân viên trong khách sạn đều sử dụng được chức năng này</li> </ul>
---	--

<p>Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p> <p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p> <p><b>3. Thông tin đầu vào:</b></p>	<p>Công việc/ Xử lý: <b>Quản lý lưu trú</b></p> <p>Số thứ tự: 4 Ngày lập: 17/3/2020</p> <ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp tân có thêm, xóa hoặc cập nhật thông tin khách hàng lưu trú</li> <li>- Ban lãnh đạo và nhân viên tiếp tân truy cập vào hệ thống và có nhu cầu quản lý thông tin khách hàng lưu trú</li> <li>- Nhân viên tiếp tân chọn chức năng quản lý thông tin khách hàng lưu trú</li> </ul>
---	---

<b>4. Kết quả đầu ra:</b>  <b>5. Nơi sử dụng:</b>  <b>6. Những quy tắc điều kiện đi kèm:</b>	<ul style="list-style-type: none"> <li>- Giao diện quản lý thông tin khách hàng lưu trú thể hiện trên hệ thống</li> <li>- Giao diện chức năng quản lý lưu trú của hệ thống</li> <li>- Ban lãnh đạo khách sạn và nhân viên tiếp tân đều có quyền truy cập chức năng này</li> </ul>
--	---

Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’ Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Tra cứu phòng</b>	Số thứ tự: 5 Ngày lập: 17/3/2020
<b>1. Mô tả công việc:</b>  <b>2. Điều kiện bắt đầu (kích hoạt):</b>  <b>3. Thông tin đầu vào:</b>  <b>4. Kết quả đầu ra:</b>  <b>5. Nơi sử dụng:</b>  <b>6. Những quy tắc điều kiện đi kèm:</b>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp kiểm tra tình trạng các phòng của khách sạn</li> <li>- Khi có khách hàng liên hệ đặt phòng</li> <li>- Ban lãnh đạo có nhu cầu tra cứu thông tin phòng</li> <li>- Nhân viên tiếp tân chọn chức năng tra cứu thông tin phòng</li> <li>- Thông tin trạng thái của các phòng thể hiện trên hệ thống</li> <li>- Giao diện chức năng tra cứu phòng của hệ thống</li> <li>- Ban lãnh đạo khách sạn và nhân viên tiếp tân đều có quyền truy cập chức năng này</li> </ul>	

Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’ Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Quản lý phòng</b>	Số thứ tự: 6 Ngày lập: 17/3/2020
<b>1. Mô tả công việc:</b>  <b>2. Điều kiện bắt đầu (kích hoạt):</b>  <b>3. Thông tin đầu vào:</b>  <b>4. Kết quả đầu ra:</b>  <b>5. Nơi sử dụng:</b>  <b>6. Những quy tắc điều kiện đi kèm:</b>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo khách sạn có thể thêm, xóa hoặc cập nhật thông tin phòng</li> <li>- Ban lãnh đạo có nhu cầu quản lý thông tin phòng</li> <li>- Ban lãnh đạo chọn chức năng quản lý thông tin phòng</li> <li>- Giao diện quản lý thông tin phòng thể hiện trên hệ thống</li> <li>- Giao diện chức năng quản lý phòng của hệ thống</li> <li>- Chỉ có ban lãnh đạo khách sạn mới có quyền truy cập chức</li> </ul>	

Đồ án HQT \_ Quản Lý Khách sạn

	năng này
--	----------

Dự án: Quản lý <b>nhà hàng</b> <b>khách sạn</b> ‘Paradis’ Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Tra cứu dịch vụ</b>	Số thứ tự: 7 Ngày lập: 17/3/2020
1. Mô tả công việc:  2. Điều kiện bắt đầu ( <b>kích hoạt</b> ):  3. Thông tin <b>đầu vào</b> :  4. Kết quả <b>đầu ra</b> :  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm:	<ul style="list-style-type: none"> <li>– Ban lãnh đạo và nhân viên tiếp xem thông tin dịch vụ</li> <li>– Khi khách hàng muốn sử dụng dịch vụ của khách sạn</li> <li>– Ban lãnh đạo có nhu cầu tra cứu thông tin dịch vụ</li> <li>– Nhân viên tiếp tân chọn chức năng tra cứu thông tin dịch vụ</li> <li>– Thông tin dịch vụ hiển thị trên màn hình hệ thống</li> <li>– Giao diện chức năng tra cứu dịch vụ của hệ thống</li> <li>– Ban lãnh đạo và nhân viên tiếp tân khách sạn đều có quyền truy cập chức năng này</li> </ul>	

Dự án: Quản lý <b>nhà hàng</b> <b>khách sạn</b> ‘Paradis’ Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Quản lí dịch vụ</b>	Số thứ tự: 8 Ngày lập: 17/3/2020
1. Mô tả công việc:  2. Điều kiện bắt đầu ( <b>kích hoạt</b> ):  3. Thông tin <b>đầu vào</b> :  4. Kết quả <b>đầu ra</b> :  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm:	<ul style="list-style-type: none"> <li>– Ban lãnh đạo có thể thêm, xóa hoặc cập nhật thông tin dịch vụ của khách sạn</li> <li>– Ban lãnh đạo có nhu cầu quản lý thông tin dịch vụ</li> <li>– Ban lãnh đạo chọn chức năng quản lý thông tin dịch vụ</li> <li>– Giao diện quản lý thông tin dịch vụ hiển thị trên hệ thống</li> <li>– Giao diện chức năng quản lý dịch vụ của hệ thống</li> <li>– Chỉ có ban lãnh đạo khách sạn mới có quyền truy cập chức năng này</li> </ul>	

Dự án: Quản lý <b>nhà hàng</b> <b>khách sạn</b> ‘Paradis’ Người lập: Nguyễn Lê Mỹ	Công việc/ Xử lý: <b>Tra cứu cơ sở vật chất</b>	Số thứ tự: 9 Ngày lập: 17/3/2020
--	---	-------------------------------------

Đồ án HQT \_ Quản Lý Khách sạn

<p><b>Uyên</b></p> <p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p> <p>4. Kết quả <b>đầu ra</b>:</p> <p>5. <b>Nơi sử dụng</b>:</p> <p>6. <b>Những quy tắc điều kiện</b> đi kèm:</p>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp tân tra cứu cơ sở vật chất để biết thông tin của cơ sở vật chất</li> <li>- Khi khách hàng hàng làm hư hỏng cơ sở vật chất của khách sạn</li> <li>- Ban lãnh đạo có nhu cầu tra cứu cơ sở vật chất</li> <li>- Ban lãnh đạo và nhân viên tiếp tân chọn chức năng tra cứu thông tin cơ sở vật chất</li> <li>- Thông tin cơ sở vật chất hiển thị trên màn hình hệ thống</li> <li>- Giao diện chức năng tra cứu cơ sở vật chất của hệ thống</li> <li>- Ban lãnh đạo và nhân viên tiếp tân đều được sử dụng chức năng này</li> </ul>
---	--

<p>Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p> <p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p> <p>4. Kết quả <b>đầu ra</b>:</p> <p>5. <b>Nơi sử dụng</b>:</p> <p>6. <b>Những quy tắc điều kiện</b> đi kèm:</p>	<p>Công việc/ Xử lý: <b>Quản lí cơ sở vật chất</b></p> <ul style="list-style-type: none"> <li>- Ban lãnh đạo có thể thêm, xóa hoặc cập nhật thông tin cơ sở vật chất của khách sạn</li> <li>- Ban lãnh đạo có nhu cầu quản lý thông tin cơ sở vật chất của khách sạn</li> <li>- Ban lãnh đạo chọn chức năng quản lý cơ sở vật chất</li> <li>- Giao diện quản lý thông tin cơ sở vật chất hiển thị trên hệ thống</li> <li>- Giao diện chức năng quản lý cơ sở vật chất của hệ thống</li> <li>- Chỉ có ban lãnh đạo khách sạn mới có quyền truy cập chức năng này</li> </ul>	<p>Số thứ tự: 10</p> <p>Ngày lập: 17/3/2020</p>
--	--	---

<p>Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p>	<p>Công việc/ Xử lý: <b>Tra cứu món ăn</b></p>	<p>Số thứ tự: 11</p> <p>Ngày lập: 17/3/2020</p>
---	--	---

Đồ án HQT \_ Quản Lý Khách sạn

<p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p> <p><b>3. Thông tin đầu vào:</b></p> <p><b>4. Kết quả đầu ra:</b></p> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo và nhân viên tiếp kiểm tra tình trạng các món ăn của khách sạn</li> <li>- Khi có khách hàng liên hệ đặt món ăn</li> <li>- Ban lãnh đạo có nhu cầu tra cứu thông tin món ăn</li> <li>- Nhân viên tiếp tân chọn chức năng tra cứu thông tin món ăn</li> <li>- Thông tin trạng thái của các món ăn thể hiện trên hệ thống</li> <li>- Giao diện chức năng tra cứu món ăn của hệ thống</li> <li>- Ban lãnh đạo khách sạn và nhân viên tiếp tân đều có quyền truy cập chức năng này</li> </ul>
---	--

<p>Dự án: Quản lý nhà hàng khách sạn 'Paradis'</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p>	<p>Công việc/ Xử lý: <b>Quản lí món ăn</b></p>	<p>Số thứ tự: 12</p> <p>Ngày lập: 17/3/2020</p>
<p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p> <p><b>3. Thông tin đầu vào:</b></p> <p><b>4. Kết quả đầu ra:</b></p> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p>	<ul style="list-style-type: none"> <li>- Ban lãnh đạo có thể thêm, xóa hoặc cập nhật thông tin món ăn của khách sạn</li> <li>- Ban lãnh đạo truy cập hệ thống và nhu cầu quản lý thông tin món ăn của khách sạn</li> <li>- Ban lãnh đạo chọn chức năng quản lý món ăn</li> <li>- Giao diện quản lý thông tin món ăn hiển thị trên hệ thống</li> <li>- Giao diện chức năng quản lý món ăn của hệ thống</li> <li>- Chỉ có ban lãnh đạo khách sạn mới có quyền truy cập chức năng này</li> </ul>	

<p>Dự án: Quản lý nhà hàng khách sạn 'Paradis'</p> <p>Người lập: Nguyễn Lê Mỹ Uyên</p>	<p>Công việc/ Xử lý: <b>Lập hóa đơn dịch vụ</b></p>	<p>Số thứ tự: 13</p> <p>Ngày lập: 17/3/2020</p>
<p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p>	<ul style="list-style-type: none"> <li>- Khi khách hàng yêu cầu dịch vụ, nhân viên tiếp tân phải lập một hóa đơn, ghi nhận khách yêu dịch vụ gì? Vào thời điểm nào? Chi phí tương ứng là bao nhiêu?</li> <li>- Khi khách hàng yêu cầu dịch vụ</li> </ul>	

	<p><b>3. Thông tin đầu vào:</b></p> <ul style="list-style-type: none"> <li>– Mã bản, số thứ tự, mã khách hàng (CMND), những dịch vụ mà khách hàng yêu cầu đặt cũng như là thông tin về giá dịch vụ, số lần yêu cầu và số tiền khách hàng trả</li> </ul> <p><b>4. Kết quả đầu ra:</b></p> <ul style="list-style-type: none"> <li>– Thông tin dịch vụ khác hàng yêu cầu</li> <li>– Tổng tiền dịch vụ</li> </ul> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p> <ul style="list-style-type: none"> <li>– Giao diện chức năng lập hóa đơn dịch vụ của hệ thống</li> <li>– Khi nhập từ 2 dịch vụ lên thì hệ thống sẽ tự cộng dồn tổng tiền trước đó với giá tiền của dịch vụ vừa nhập</li> </ul>
--	---

Dự án: Quản lý nhà hàng khách sạn 'Paradis' Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Lập hóa đơn món ăn</b>	Số thứ tự: 14 Ngày lập: 17/3/2020
<p><b>1. Mô tả công việc:</b></p> <p><b>2. Điều kiện bắt đầu (kích hoạt):</b></p> <p><b>3. Thông tin đầu vào:</b></p> <p><b>4. Kết quả đầu ra:</b></p> <p><b>5. Nơi sử dụng:</b></p> <p><b>6. Những quy tắc điều kiện đi kèm:</b></p>	<ul style="list-style-type: none"> <li>– Khi khách hàng yêu cầu món ăn, nhân viên tiếp tân phải lập một hóa đơn, ghi nhận khách yêu món ăn gì? Vào thời điểm nào? Giá bao nhiêu?</li> <li>– Khi khách hàng yêu cầu món ăn</li> </ul> <ul style="list-style-type: none"> <li>– Mã hóa đơn, số thứ tự, mã khách hàng (CMND), những món ăn mà khách hàng yêu cầu cũng như là thông tin về giá món ăn, số lần yêu cầu và tổng số tiền khách hàng trả</li> </ul> <ul style="list-style-type: none"> <li>– Thông tin dịch vụ khách hàng yêu cầu</li> <li>– Tổng tiền dịch vụ</li> </ul> <ul style="list-style-type: none"> <li>– Giao diện chức năng lập hóa đơn món ăn của hệ thống</li> <li>– Khi nhập từ 2 món trở lên thì hệ thống sẽ tự cộng dồn tổng tiền trước đó với giá tiền của món ăn vừa nhập</li> </ul>	

Dự án: Quản lý nhà hàng khách sạn 'Paradis' Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Lập hóa đơn cơ sở vật chất</b>	Số thứ tự: 15 Ngày lập: 17/3/2020
<p><b>1. Mô tả công việc:</b></p>	<ul style="list-style-type: none"> <li>– Liệt kê các cơ sở vật chất của khách sạn mà khách hàng đã làm hư hỏng</li> </ul>	

	<ul style="list-style-type: none"> <li>- Khi khách hàng làm hư hỏng cơ sở vật chất</li> </ul>
2. Điều kiện bắt đầu ( <b>kích hoạt</b> ):	
3. Thông tin <b>đầu vào</b> :	<ul style="list-style-type: none"> <li>- Mã hóa đơn, số thứ tự, mã khách hàng (CMND), số phòng, những cơ sở vật chất khách hàng làm hư hỏng và giá của chúng</li> </ul>
4. Kết quả <b>đầu ra</b> :	<ul style="list-style-type: none"> <li>- Thông tin cơ sở vật chất khách hàng làm hư hỏng</li> <li>- Tổng tiền cơ sở vật chất</li> </ul>
5. <b>Nơi sử dụng</b> :	<ul style="list-style-type: none"> <li>- Giao diện chức năng lập hóa đơn cơ sở vật chất của hệ thống</li> </ul>
6. <b>Những quy tắc điều kiện</b> đi kèm:	<ul style="list-style-type: none"> <li>- Khi nhập từ 2 cơ sở vật chất trở lên thì hệ thống sẽ tự cộng dồn tổng tiền trước đó với giá tiền của cơ sở vật chất vừa nhập</li> </ul>

Dự án: Quản lý <b>nhà hàng</b> khách sạn ‘Paradis’ Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Lập hóa đơn thanh toán tổng</b>	Số thứ tự: 16 Ngày lập: 17/3/2020
<ul style="list-style-type: none"> <li>1. Mô tả công việc:</li> <li>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</li> <li>3. Thông tin <b>đầu vào</b>:</li> <li>4. Kết quả <b>đầu ra</b>:</li> <li>5. <b>Nơi sử dụng</b>:</li> <li>6. <b>Những quy tắc điều kiện</b> đi kèm:</li> </ul>	<ul style="list-style-type: none"> <li>- Khi khách hàng trả phòng, nhân viên tiếp tân sẽ lập hóa đơn thanh toán tổng, tính tổng tiền số tiền khách hàng phải trả</li> <li>- Khi khách hàng trả phòng</li> <li>- Mã hóa đơn</li> <li>- Hóa đơn dịch vụ (nếu có)</li> <li>- Hóa đơn món ăn (nếu có)</li> <li>- Hóa đơn cơ sở vật chất (nếu có)</li> <li>- Hóa đơn tiền phòng</li> <li>- Tổng tiền khách hàng phải trả</li> <li>- Giao diện chức năng lập hóa đơn thanh toán tổng của hệ thống</li> <li>- Chỉ lập một lần duy nhất</li> </ul>	

Đồ án HQT \_ Quản Lý Khách sạn

Dự án: Quản lý <b>nhà hàng khách sạn 'Paradis'</b> Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Thống kê doanh thu</b>	Số thứ tự: 17 Ngày lập: 17/3/2020
<p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p> <p>4. Kết quả <b>đầu ra</b>:</p> <p>5. Nơi sử dụng:</p> <p>6. <b>Những quy tắc điều kiện</b> đi kèm:</p>	<ul style="list-style-type: none"> <li>- Báo cáo doanh thu cho khách sạn theo ngày, theo tháng, theo năm</li> <li>- Ban lãnh đạo có nhu cầu xem doanh thu của khách sạn</li> <li>- Tổng số tiền khách sạn thu được theo ngày, theo tháng, theo năm</li> <li>- Doanh thu theo ngày, theo tháng, theo năm</li> <li>- Giao diện chức năng báo cáo doanh thu của hệ thống</li> <li>- Chỉ có ban lãnh đạo mới được sử dụng chức năng này</li> </ul>	

Dự án: Quản lý <b>nhà hàng khách sạn 'Paradis'</b> Người lập: Nguyễn Lê Mỹ Uyên	Công việc/ Xử lý: <b>Đăng nhập</b>	Số thứ tự: 18 Ngày lập: 17/3/2020
<p>1. Mô tả công việc:</p> <p>2. Điều kiện bắt đầu (<b>kích hoạt</b>):</p> <p>3. Thông tin <b>đầu vào</b>:</p> <p>4. Kết quả <b>đầu ra</b>:</p> <p>5. Nơi sử dụng:</p> <p>6. <b>Những quy tắc điều kiện</b> đi kèm:</p>	<ul style="list-style-type: none"> <li>- Khi nhân viên tiếp tân và ban lãnh đạo muốn sử dụng hệ thống</li> <li>- Nhân viên tiếp tân và ban lãnh đạo phải có tài khoản trên hệ thống</li> <li>- Tên đăng nhập và mật khẩu</li> <li>- Đăng nhập thành công vào hệ thống</li> <li>- Giao diện đăng nhập của hệ thống</li> <li>- Nhân viên nhà hàng được cấp tài khoản để đăng nhập vào hệ thống</li> <li>- Đăng nhập thành công nếu tên đăng nhập và mật khẩu đúng</li> </ul>	

### 2.1.2.1.2 Tổng hợp các dữ liệu

Dự án: Quản lý nhà hàng khách sạn ‘Paradis’	Công việc/Xử lý: Tra cứu thông tin phòng (1), Quản lý thông tin phòng (2)...	Số thứ tự: 1
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: Phòng	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để xác định phòng. Mỗi giá trị dữ liệu loại phòng xác định duy nhất một phòng.	
2. Định dạng:	- Kiểu kí tự, gồm 20 kí tự.	
3. Số lượng:	- Có tối đa $2.10^{18}$ .	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: phòng Thương gia, số phòng XXXX</li> <li>- Ngoài loại phòng, phòng còn kèm theo thông tin như số phòng, giá phòng, mô tả phòng, ...</li> </ul>	

Dự án: Quản lý nhà hàng khách sạn ‘Paradis’	Công việc/Xử lý: Tra cứu thông tin khuyến mãi (1), Quản lý thông tin khuyến mãi (2)...	Số thứ tự: 2
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: Khuyến mãi	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để xác định ngày áp dụng khuyến mãi. Mỗi giá trị dữ liệu xác định duy nhất một khuyến mãi.	
2. Định dạng:	- Kiểu kí tự, gồm 20 kí tự.	
3. Số lượng:	- Có tối đa $2.10^{18}$ .	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: khuyến mãi Tết, giảm giá XX%</li> <li>- Tên khuyến mãi ghi đầy đủ, không viết tắt. Ngoài tên khuyến mãi còn kèm theo thông tin như ngày áp dụng, giá giảm, ...</li> </ul>	

Dự án: Quản lý nhà hàng khách sạn ‘Paradis’	Công việc/Xử lý: Tra cứu thông tin dịch vụ (1), Quản lý thông tin dịch vụ (2)...	Số thứ tự: 3
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: Dịch vụ	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để chỉ một loại dịch vụ. Mỗi giá trị dữ liệu dịch vụ xác định duy nhất một dịch vụ.	
2. Định dạng:	- Kiểu kí tự, gồm 1 đến 50 kí tự.	
3. Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Dịch vụ Phòng Spa, mã XXXX</li> <li>- Tên dịch vụ ghi đầy đủ, không viết tắt. Ngoài tên dịch vụ còn</li> </ul>	

Đồ án HQT \_ Quản Lý Khách Sạn

	kèm theo thông tin khác như mã dịch vụ, giá, ...
--	--

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Tra cứu thông tin món ăn (1), Quản lý thông tin món ăn (2)...</b>	Số thứ tự: 4
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Món ăn</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để chỉ một loại món ăn. Mỗi giá trị dữ liệu món ăn xác định duy nhất một món ăn.	
2. Định dạng:	- Kiểu kí tự, gồm 1 đến 50 kí tự.	
3.Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Món ăn Súp có mã XXXX</li> <li>- Tên món ăn được ghi đầy đủ, không viết tắt. Ngoài tên món ăn còn kèm theo thông tin khác như mã món ăn, giá...</li> </ul>	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Tra cứu thông tin CSVC (1), Quản lý thông tin CSVC (2)...</b>	Số thứ tự: 5
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: Cơ sở vật chất	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để chỉ một vật dụng trong phòng. Mỗi giá trị dữ liệu xác định duy nhất một vật dụng.	
2. Định dạng:	- Kiểu kí tự, gồm 1 đến 50 kí tự.	
3.Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Vật dụng Drap Giường, mã XXXX</li> <li>- Tên vật dụng được ghi đầy đủ, không viết tắt. Ngoài tên vật dụng còn kèm theo thông tin khác như mã, giá, số lượng...</li> </ul>	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Tra cứu thông tin khách hàng đặt phòng...</b>	Số thứ tự: 6
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Khách hàng</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để chỉ một khách hàng. Mỗi giá trị dữ liệu khách hàng xác định duy nhất một khách hàng.	
2. Định dạng:	- Kiểu kí tự, gồm 1 đến 50 kí tự.	
3.Số lượng:	- Không giới hạn.	

Đồ án HQT \_ Quản Lý Khách Sạn

4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Khách hàng Trần Trúc Quân, số CMND XXXX</li> <li>- Họ tên được ghi đầy đủ, không viết tắt. Ngoài họ tên khách hàng còn kèm theo thông tin khác như số CMND, số điện thoại, năm sinh, ngày đặt phòng, ngày trả phòng, người đi kèm...</li> </ul>
-----------	---

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Lập hóa đơn tiền phòng</b>	Số thứ tự: 7
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Hóa đơn tiền phòng</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Xác định số tiền phòng khách hàng cần trả. Mỗi giá trị dữ liệu xác định duy nhất một hóa đơn dịch vụ.	
2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.	
3. Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Hóa đơn tiền phòng của khách hàng Quân, số thứ tự XXXX.</li> <li>- Ngoài số thứ tự hóa đơn tiền phòng còn kèm theo thông tin khác như số phòng, tên khách hàng, thời gian ở khách sạn, số lần, thành tiền...</li> </ul>	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Lập hóa đơn dịch vụ</b>	Số thứ tự: 8
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Hóa đơn dịch vụ</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Xác định tổng số tiền dịch vụ khách hàng đã sử dụng. Mỗi giá trị dữ liệu xác định duy nhất một hóa đơn dịch vụ.	
2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.	
3. Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Hóa đơn dịch vụ của khách hàng Quân, số thứ tự XXXX</li> <li>- Ngoài số thứ tự hóa đơn dịch vụ còn kèm theo thông tin khác như mã dịch vụ, tên khách hàng, giá, số lần, thành tiền...</li> </ul>	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Lập hóa đơn món ăn</b>	Số thứ tự: 9
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Hóa đơn món ăn</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Xác định tổng số tiền các món ăn khách hàng đã yêu cầu. Mỗi giá	

	trị dữ liệu xác định duy nhất một hóa đơn món ăn.
2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.
3. Số lượng:	- Không giới hạn.
4. Mô tả:	- Ví dụ: Hóa đơn món ăn của khách hàng Quân, số thứ tự XXXX - Ngoài số thứ tự hóa đơn còn kèm theo thông tin khác như mã món ăn, giá, số lượng, tên khách hàng...

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Lập hóa đơn cơ sở vật chất</b>	Số thứ tự: 10
Người lập: XXXX	Dữ liệu: <b>Hóa đơn CSVC</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Xác định số tiền cơ sở vật chất khách hàng đã làm hư hại. Mỗi giá trị dữ liệu xác định duy nhất một hóa đơn CSVC.	
2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.	
3. Số lượng:	- Không giới hạn.	
4. Mô tả:	- Ví dụ: Hóa đơn CSVC của khách hàng Quân, số thứ tự XXXX - Ngoài số thứ tự hóa đơn còn kèm theo thông tin khác như mã vật dụng, giá, số lượng, tên khách hàng...	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Lập hóa đơn thanh toán tổng</b>	Số thứ tự: 11
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Hóa đơn thanh toán tổng</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Xác định tổng số tiền khách hàng cần thanh toán trong thời gian ở khách sạn. Mỗi giá trị dữ liệu xác định duy nhất một hóa đơn thanh toán.	
2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.	
3. Số lượng:	- Không giới hạn.	
4. Mô tả:	- Ví dụ: Hóa đơn thanh toán của khách hàng Quân, số thứ tự XXXX - Ngoài số thứ tự hóa đơn còn kèm theo thông tin khác như tiền phòng, thời gian, hóa đơn dịch vụ, hóa đơn món ăn...	

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Thông kê doanh thu</b>	Số thứ tự: 12
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Báo cáo doanh thu</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Báo cáo tổng doanh thu của khách sạn theo mỗi ngày, mỗi tháng, mỗi năm. Mỗi giá trị dữ liệu xác định duy nhất một báo cáo.	

## Đồ án HQT \_ Quản Lý Khách Sạn

2. Định dạng:	- Kiểu tiền tệ, định dạng VND, không có giá trị âm.
3.Số lượng:	- Không giới hạn.
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Báo cáo doanh thu, ngày XXXX</li> <li>- Ngoài ngày báo cáo còn kèm theo thông tin khác như thời gian thống kê...</li> </ul>

Dự án: <b>Quản lý nhà hàng khách sạn 'Paradis'</b>	Công việc/Xử lý: <b>Đăng nhập tài khoản</b>	Số thứ tự: 13
Người lập: Nguyễn Vũ Tú Anh	Dữ liệu: <b>Nhân viên</b>	Ngày lập: 19/3/2020
1. Định nghĩa:	- Dùng để chỉ một nhân viên trong khách sạn. Mỗi giá trị dữ liệu dùng để xác định một nhân viên.	
2. Định dạng:	- Kiểu kí tự, gồm 1 đến 50 kí tự.	
3.Số lượng:	- Không giới hạn.	
4. Mô tả:	<ul style="list-style-type: none"> <li>- Ví dụ: Nhân viên Cao Thanh Tuấn, chức vụ lễ tân.</li> <li>- Họ tên được ghi đầy đủ, không viết tắt. Ngoài họ tên nhân viên còn kèm theo thông tin khác như mã nhân viên, số CMND, chức vụ, ...</li> </ul>	

### 2.1.2.2 Đánh giá hiện trạng

Thiếu	<ul style="list-style-type: none"> <li>- Thiếu sự tiện lợi: Tất cả các phiếu đơn, bảng kê và các hóa đơn đều lưu trên giấy, bất tiện cho việc tìm kiếm và thanh toán.</li> <li>- Thiếu sự nhất quán: Đối với quản lý đặt tiệc cần phải có 1 menu món ăn với giá cả rõ ràng. Như vậy sẽ dễ quản lý và chất lượng cũng như hương vị món ăn sẽ được đảm bảo hơn.</li> </ul>
-------	--

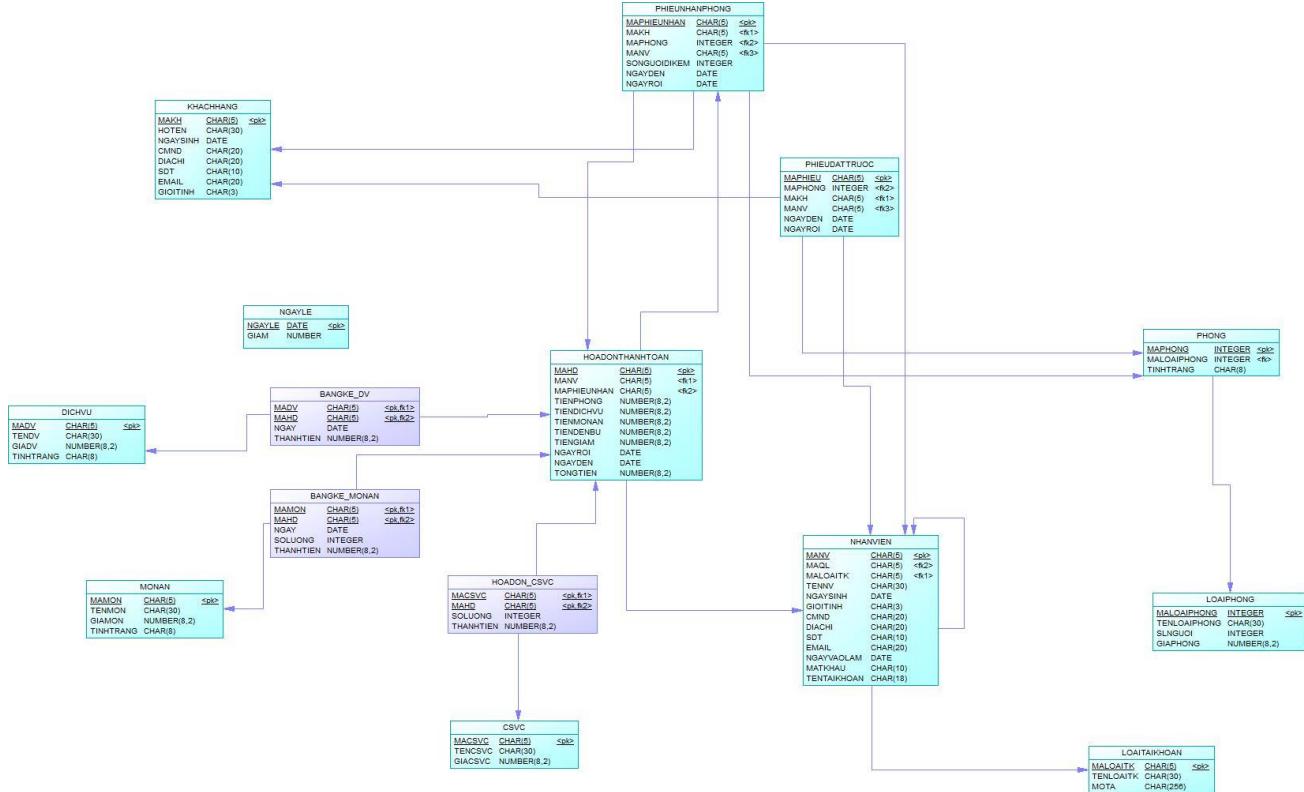
<b>Hiệu quả kém</b>	<ul style="list-style-type: none"> <li>- Lưu trữ các thông tin của khách sạn và khách hàng trên giấy rất tốn thời gian và dễ gây sai sót, rắc rối.</li> <li>- Hiệu suất làm việc sẽ không cao do nhân viên lễ tân sẽ mất khá nhiều thời gian trong lưu trữ, tra cứu và truy xuất thông tin.</li> <li>- Mất thời gian khi mỗi lần giao ca nhân viên phải bàn giao phần công việc đã làm cho nhân viên ca sau (dễ xảy ra nhầm lẫn, thiếu sót).</li> </ul>
<b>Tốn kém, dù thừa</b>	<ul style="list-style-type: none"> <li>- Lãng phí thời gian, tiền bạc.</li> </ul>

#### 2.1.2.3 Yêu cầu mới

- Chuyển đổi tất cả các dữ liệu của khách sạn cũng như dữ liệu của khách hàng thành dữ liệu số mà máy tính có thể quản lý được.
- Dễ hiểu, dễ sử dụng, dễ lưu trữ và tiện lợi hơn cho nhân viên lễ tân.
- Đảm bảo thông tin luôn chính xác không bị sai sót hay thất lạc.
- Khách hàng cũng sẽ yên tâm hơn về việc sử dụng dịch vụ của khách sạn.

## 2.2 Mô hình dữ liệu quan hệ

### 2.2.1 Thiết kế mô hình dữ liệu quan hệ mức vật lý



#### 2.2.1.1 Table

#### LOAITAIKHOAN

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MALOAITK	Mã loại tài khoản	Chuỗi	5		
2	TENLOAITK	Tên loại tài khoản	Chuỗi	30	Tài khoản nhân viên, tài khoản quản lý	
3	MOTA	Mô tả	Chuỗi	256		Gồm các chức năng mà loại tài khoản được phép sử dụng

#### 2.2.1.2 Table NHANVIEN

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MANV	Mã nhân	Chuỗi	5		

Đồ án HQT \_ Quản Lý Khách Sạn

		viên				
2	MALOAITK	Mã loại tài khoản	Chuỗi	5		Khóa ngoại tham chiếu tới bảng loại tài khoản
3	MAQL	Mã người quản lý	Chuỗi	5		Người quản lý phải là một nhân viên của khách sạn
4	TENNV	Tên nhân viên	Chuỗi	30		
5	NGAYSINH	Năm sinh	Ngày			
6	GIOITINH	Giới tính	Chuỗi	3	Nam, Nữ	
7	CMND	CMND	Chuỗi	20		CMND là duy nhất
8	DIACHI	Địa chỉ	Chuỗi	20		
9	SDT	Số điện thoại	Chuỗi	10		SDT là duy nhất
10	Email	Email	Chuỗi	20		Email là duy nhất
11	NGAYVAOLAM	Ngày vào làm	Ngày			Ngày vào làm phải lớn hơn ngày sinh nhân viên
12	TENTAIKHOAN	Tên tài khoản	Chuỗi	18		Tên tài khoản là duy nhất
13	MATKHAU	Mật khẩu	Chuỗi	10		

2.2.1.3 Table LOAIPHONG

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MALOAIPHONG	Mã loại phòng	Số		1,2,3,4	
2	TENLOAIPHONG	Tên loại phòng	Chuỗi	30		
3	GIAPHONG	Giá phòng	Tiền			
4	SLNGUOI	Số người	Số			

2.2.1.4 Table PHONG

STT	Thuộc tính	Diễn giải	Kiểu dữ	Chiều	Miền	Ghi chú
-----	------------	-----------	---------	-------	------	---------

			<b>liệu</b>	<b>dài</b>	<b>giá trị</b>	
1	MAPHONG	Mã phòng	Chuỗi	5		
2	MALOAIPHONG	Mã loại phòng	Số		1,2,3,4	Là ngoại chiêu đến bảng loại phòng
3	TINHTRANG	Tình trạng	Chuỗi	30	Bận, trống, đang sửa chữa	

#### 2.2.1.5 Table KHACHHANG

<b>STT</b>	<b>Thuộc tính</b>	<b>Diễn giải</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Miền giá trị</b>	<b>Ghi chú</b>
1	MAKH	Mã khách hàng	Chuỗi	5		
2	HOTEN	Họ tên khách hàng	Chuỗi	30		
3	NGAYSINH	Ngày sinh	Ngày			
4	CMND	CMND	Chuỗi	20		Mỗi CMND là duy nhất
5	DIACHI	Địa chỉ	Chuỗi	20		
6	SDT	Số điện thoại	Chuỗi	10		Mỗi sdt là duy nhất
7	EMAIL	EMAIL	Chuỗi	20		Mỗi email là duy nhất
8	GIOITINH	Giới tính	Chuỗi	3	Nam, nữ	

#### 2.2.1.6 Table DICHVU

<b>STT</b>	<b>Thuộc tính</b>	<b>Diễn giải</b>	<b>Kiểu dữ liệu</b>	<b>Chiều dài</b>	<b>Miền giá trị</b>	<b>Ghi chú</b>
1	MADV	Mã dịch vụ	Chuỗi	5		
2	TENDV	Tên dịch vụ	Chuỗi	30		
3	GIADV	Giá dịch vụ	Tiền			
4	TINHTRANG	Tình trạng	Chuỗi		True, false	True: phục vụ False: tạm ngưng

#### 2.2.1.7 Table MONAN

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MAMON	Mã món ăn	Chuỗi	5		
2	TENMON	Tên món	Chuỗi	30		
3	GIAMON	Giá món	Tiền			
4	TINHTRANG	Tình trạng	Chuỗi		True, false	True: phục vụ False: tạm ngưng

#### 2.2.1.8 Table CSVC

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MACSVC	Mã cơ sở vật chất	Chuỗi	5		
2	TENCSVC	Tên cơ sở vật chất	Chuỗi	30		
3	GIACSVC	Giá cơ sở vật chất	Tiền			

#### 2.2.1.9 Table PHIEUNHANPHONG

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MAPHIEUNHAN	Mã phiếu nhận	Chuỗi	5		
2	MAKH	Mã khách hàng	Chuỗi	5		
3	MAPHONG	Mã phòng	Chuỗi	5		
4	MANV	Mã nhân viên	Chuỗi	5		
5	SONGUOIDIKEM	Số người đi kèm	Số		Phải nhỏ hơn hoặc bằng SLNGUOI trong loại phòng mà khách hàng đặt	
6	NGAYDEN	Ngày đến	Ngày			
7	NGAYROI	Ngày rời	Ngày			Ngày rời > Ngày đến

#### 2.2.1.10 Table PHIEUDATTRUOC

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MAPHIEU	Mã phiếu	Chuỗi	5		
2	MAKH	Mã khách hàng	Chuỗi	5		
3	MAPHONG	Mã phòng	Chuỗi	5		
4	MANV	Mã nhân viên	Chuỗi	5		
5	NGAYDEN	Ngày đến	Ngày			
6	NGAYROI	Ngày rời	Ngày			

#### 2.2.1.11 Table NGAYLE

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	NGAYLE	Ngày lễ	Ngày			NULL
2	GIAM	Số % được giảm	Số		0 < GIAM < 100%	

#### 2.2.1.12 Table HOADONTANHTOAN

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MAHD	Mã hóa đơn	Chuỗi	5		
2	MANV	Mã nhân viên	Chuỗi	5		
3	MAPHIEUNHAN	Mã phiếu nhận	Chuỗi	5		
4	TIENPHONG	Tiền phòng	Tiền			Tiền phòng = (Ngày rời – Ngày đến) * Giá phòng
5	TIENDICHVU	Tiền dịch vụ	Tiền			
6	TIENMONAN	Tiền món ăn	Tiền			
7	TIENDENBU	Tiền đèn bù	Tiền			
8	TIENGIAM	Tiền giám	Tiền			
9	NGAYDEN	Ngày đến	Ngày			
10	NGAYROI	Ngày rời	Ngày			
11	TONGTIEN	Tổng tiền	Tiền			Tổng tiền = Tiền dịch vụ + tiền phòng + tiền món

						ăn + tiền đền bù – tiền giảm
--	--	--	--	--	--	---------------------------------

#### 2.2.1.13 Table BANGKE\_DV

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MADV	Mã dịch vụ	Chuỗi	5		
2	MAHD	Mã hóa đơn	Chuỗi	5		
3	NGAY	Ngày sử dụng	Ngày giờ		Phải trong khoảng ngày đến và ngày rời	
4	THANHTIEN	Thành tiền	Tiền			

#### 2.2.1.14 Table BANGKE\_MONAN

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MAMON	Mã món ăn	Chuỗi	5		
2	MAHD	Mã hóa đơn	Chuỗi	5		
3	NGAY	Ngày sử dụng	Ngày, giờ		Phải trong khoảng ngày đến và ngày rời	
4	SOLUONG	Số lượng	Số			
5	THANHTIEN	Thành tiền	Tiền			

#### 2.2.1.15 Table HOADON\_CSVC

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Chiều dài	Miền giá trị	Ghi chú
1	MACSVC	Mã cơ sở vật chất	Chuỗi	5		
2	MAHD	Mã hóa đơn	Chuỗi	5		
3	SOLUONG	Số lượng	Số			
4	THANHTIEN	Thành tiền	Tiền			

#### 2.2.2 Ràng buộc toàn vẹn

##### 2.2.2.1 Bảng NHANVIEN

- ❖ **R<sub>1</sub>** : "Tất cả các nhân viên phải có mã số phân biệt nhau"

RBTV liên bộ

Bối cảnh: NHANVIEN

Nội dung:

$\forall nv1, nv2 \in NHANVIEN: nv1 \neq nv2 \Rightarrow nv1.MANV \neq nv2.MANV$

R <sub>1</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+ (MANV)

❖ **R<sub>2</sub>:** "Người quản lý trực tiếp của một nhân viên phải là nhân viên của khách sạn"

RBTV tham chiếu

Bối cảnh: NHANVIEN

Nội dung:

$\forall nv \in NHANVIEN: (nv.MAQL = NULL) \vee (\exists nv' \in NHANVIEN: nv.MAQL = nv'.MANV)$

R <sub>2</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+ (MAQL)

❖ **R<sub>3</sub>:** “Tập các giá trị MALOAITK được tìm thấy trong quan hệ NHANVIEN phải được tìm thấy trong tập các giá trị MALOAITK trong quan hệ LOAITAIKHOAN”

RBTV tham chiếu

Bối cảnh: NHANVIEN, LOAITAIKHOAN

Nội dung:

$\forall nv \in NHANVIEN, \exists ltk \in LOAITAIKHOAN: nv.MALOAITK = ltk.MALOAITK$

R <sub>3</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+ (LOAITAIKHOAN)
LOAITAIKHOAN	-	+	-(*)

❖ **R<sub>4</sub>:** “Giới tính của nhân viên là ‘Nam’ hoặc ‘Nu’”

RBTV miền giá trị

Bối cảnh: NHANVIEN

Nội dung:

$$\forall \text{nv} \in \text{NHANVIEN}: \text{nv.GIOITINH} \in \{\text{'Nam'}, \text{'Nu'}\}$$

$R_4$	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(GIOITINH)

❖ **R<sub>5</sub>**: "Số chứng minh nhân dân của nhân viên là duy nhất"

RBTV liên bộ

Bối cảnh: NHANVIEN

Nội dung:

$$\forall \text{nv1}, \text{nv2} \in \text{NHANVIEN}: \text{nv1} \neq \text{nv2} \rightarrow \text{nv1.CMND} \neq \text{nv2.CMND}$$

$R_5$	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(CMND)

❖ **R<sub>6</sub>** : "Số điện thoại của nhân viên là duy nhất"

RBTV liên bộ

Bối cảnh: NHANVIEN

Nội dung:

$$\forall s1, s2 \in \text{NHANVIEN}: s1 \neq s2 \rightarrow s1.SDT \neq s2.SDT$$

$R_6$	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(SDT)

❖ **R<sub>7</sub>** : "Email của nhân viên là duy nhất"

RBTV liên bộ

Đồ án HQT \_ Quản Lý Khách Sạn

Bối cảnh: NHANVIEN

Nội dung :

$$\forall e1, e2 \in NHANVIEN: e1 \neq e2 \rightarrow e1.EMAIL \neq e2.EMAIL$$

R <sub>7</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(EMAIL)

❖ **R<sub>8</sub>**: "Ngày vào làm của nhân viên phải lớn hơn ngày sinh".

RBTV liên thuộc tính

Bối cảnh: NHANVIEN

Nội dung:

$$\forall nv \in NHANVIEN: nv.NGAYVAOLAM > nv.NGAYSINH$$

R <sub>8</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(NGAYVAOLAM, NGAYSINH)

❖ **R<sub>9</sub>** : "Tên tài khoản của nhân viên là duy nhất"

RBTV liên bộ

Bối cảnh: NHANVIEN

Nội dung :

$$\forall tk1, tk2 \in NHANVIEN: tk1 \neq tk2 \rightarrow tk1.TENTAIKHOAN \neq tk2.TENTAIKHOAN$$

R <sub>9</sub>	Thêm	Xóa	Sửa
NHANVIEN	+	-	+(TENTAIKHOAN)

### 2.2.2.2 Bảng LOAIPHONG

- ❖ **R<sub>10</sub>** : "Tất cả các loại phòng phải có mã phân biệt nhau"

RBTV liên bộ

Bối cảnh: LOAIPHONG

Nội dung:

$$\forall p_1, p_2 \in LOAIPHONG: p_1 \neq p_2 \rightarrow p_1.MALOAIPHONG \neq p_2.MALOAIPHONG$$

R <sub>10</sub>	Thêm	Xóa	Sửa
LOAIPHONG	+	-	+ (MALOAIPHONG)

- ❖ **R<sub>11</sub>**: “Mã loại phòng cho từng loại phòng phải là:

1. Standard
2. Superior
3. Deluxe
4. Suite”

RBTV miền giá trị

Bối cảnh: LOAIPHONG

$$\forall p \in LOAIPHONG: p.MALOAIPHONG = \{1,2,3,4\}$$

R <sub>11</sub>	Thêm	Xóa	Sửa
LOAIPHONG	+	-	+ (MALOAIPHONG)

- ❖ **R<sub>12</sub>**: “Tên loại phòng không trùng nhau”

RBTV liên bộ

Bối cảnh: LOAIPHONG

Nội dung:

$$\forall lp_1, lp_2 \in MONAN: lp_1 \neq lp_2 \rightarrow lp_1.TENLOAIPHONG \neq lp_2.TENLOAIPHONG$$

R <sub>38</sub>	Thêm	Xóa	Sửa
LOAIPHONG	+	-	+

| | | (TENLOAIPHONG)

- ❖ **R<sub>13</sub>:** “Giá tiền của loại phòng phải thấp hơn giá tiền của loại phòng cao cấp hơn”

RBTV liên bộ - liên thuộc tính

Bối cảnh: LOAIPHONG

$\forall p1, p2 \in LOAIPHONG: ((p1.MALOAPHONG < p2.MALOAPHONG) \rightarrow (p1.GIAPHONG < p2.GIAPHONG)) \vee ((p1.MALOAPHONG > p2.MALOAPHONG) \rightarrow (p1.GIAPHONG > p2.GIAPHONG))$

R <sub>13</sub>	Thêm	Xóa	Sửa
LOAIPHONG	+	-	+ (MALOAPHONG, GIAPHONG)

### 2.2.2.3 Bảng PHONG

- ❖ **R<sub>14</sub>:** "Tất cả các phòng phải có mã phân biệt nhau"

RBTV liên bộ

Bối cảnh: PHONG

Nội dung:

$\forall p1, p2 \in PHONG: p1 \neq p2 \rightarrow p1.MAPHONG \neq p2.MAPHONG$

R <sub>14</sub>	Thêm	Xóa	Sửa
PHONG	+	-	+ (MAPHONG)

- ❖ **R<sub>15</sub>:** “Tập các giá trị MALOAPHONG được tìm thấy trong quan hệ PHONG phải được tìm thấy trong tập các giá trị MALOAPHONG trong quan hệ LOAIPHONG.”

RBTV tham chiếu

Bối cảnh: PHONG, LOAIPHONG

Nội dung:

$\forall p \in PHONG, \exists p \in LOAIPHONG: p.MALOAPHONG = p.MALOAPHONG$

R <sub>15</sub>	Thêm	Xóa	Sửa
PHONG	+	-	+ (MALOAPHONG)
LOAIPHONG	-	+	-*

- ❖ **R<sub>16</sub>** : “Tình trạng phòng là: “”Trong”, “ Ban”, “Sua chua”

RBTV miền giá trị

Bối cảnh: PHONG

Nội dung:

$$\forall ph \in PHONG: ph.TINHTRANG \in \{‘Trong’, ‘Ban’, ‘Sua chua’\}$$

<b>R<sub>16</sub></b>	Thêm	Xóa	Sửa
PHONG	+	-	+ (TINHTRANG)

#### 2.2.2.4 Bảng PHIEUDATTRUOC

- ❖ **R<sub>17</sub>**: "Tất cả các phiếu đặt trước phải có mã phân biệt nhau"

RBTV liên bộ

Bối cảnh: PHIEUDATTRUOC

Nội dung:

$$\forall p1, p2 \in PHIEUDATTRUOC: p1 \neq p2 \rightarrow p1.MAPHIEU \neq p2.MAPHIEU$$

<b>R<sub>17</sub></b>	Thêm	Xóa	Sửa
PHIEUDATTRUOC	+	-	+ (MAPHIEU)

- ❖ **R<sub>18</sub>**: “Tập các giá trị MAPHONG được tìm thấy trong quan hệ PHIEUDATTRUOC phải được tìm thấy trong tập các giá trị MAPHONG trong quan hệ PHONG.”

RBTV tham chiếu

Bối cảnh: PHIEUDATTRUOC, PHONG

Nội dung:

$$\forall pdt \in PHIEUDATTRUOC, \exists p \in PHONG: pdt.MAPHONG = p.MAPHONG$$

<b>R<sub>18</sub></b>	Thêm	Xóa	Sửa
PHIEUDATTRUOC	+	-	+ (MAPHONG)
PHONG	-	+	-*

- ❖ **R<sub>19</sub>**: ““Tập các giá trị MAKH được tìm thấy trong quan hệ PHIEUDATTRUOC phải được tìm thấy trong tập các giá trị MAKH trong quan hệ KHACHHANG.”

RBTV tham chiếu

Bối cảnh: PHIEUDATTRUOC, KHACHHANG

Nội dung:

$\forall \text{pdt} \in \text{PHIEUDATTRUOC}, \exists \text{kh} \in \text{KHACHHANG}: \text{pdt.MAKH} = \text{kh.MAKH}$

$R_{19}$	Thêm	Xóa	Sửa
PHIEUDATTRUOC	+	-	+ (MAKH)
KHACHHANG	-	+	-*

- ❖ **R<sub>20</sub>:** ““Tập các giá trị MANV được tìm thấy trong quan hệ PHIEUDATTRUOC phải được tìm thấy trong tập các giá trị MANV trong quan hệ NHANVIEN.””

RBTV tham chiếu

Bối cảnh: PHIEUDATTRUOC, NHANVIEN

Nội dung:

$\forall \text{pdt} \in \text{PHIEUDATTRUOC}, \exists \text{nv} \in \text{NHANVIEN}: \text{pdt.MANV} = \text{nv.MAKH}$

$R_{20}$	Thêm	Xóa	Sửa
PHIEUDATTRUOC	+	-	+ (MANV)
NHANVIEN	-	+	-*

- ❖ **R<sub>21</sub>:** “Ngày đến nhận phòng phải nhỏ hơn hoặc bằng ngày rời”

RBTV liên thuộc tính

Bối cảnh: PHIEUDATTRUOC

Nội dung:

$\forall \text{pdt} \in \text{PHIEUDATTRUOC}: \text{pdt.NGAYDEN} \leq \text{pdt.NGAYROI}$

$R_{21}$	Thêm	Xóa	Sửa
PHIEUDATTRUOC	+	-	+ (NGAYDEN,NGAYROI)

### 2.2.2.5 Bảng PHIEUNHANPHONG

- ❖ **R<sub>22</sub>:** “Tất cả các phiếu nhận phòng phải có mã phân biệt nhau”

RBTV liên bộ

Bối cảnh: PHIEUNHANPHONG

Nội dung:

$\forall p1, p2 \in \text{PHIEUNHANPHONG}: p1 \neq p2 \rightarrow p1.MAPHIEUNHAN \neq p2.MAPHIEUNHAN$

R <sub>22</sub>	Thêm	Xóa	Sửa
PHIEUNHANPHONG	+	-	+ (MAPHIEUNHAN)

- ❖ R<sub>23</sub>: ““Tập các giá trị MAKH được tìm thấy trong quan hệ PHIEUNHANPHONG phải được tìm thấy trong tập các giá trị MAKH trong quan hệ KHACHHANG.””

RBTV tham chiếu

Bối cảnh: PHIEUNHANPHONG, KHACHHANG

Nội dung:

$$\forall pnp \in PHIEUNHANPHONG, \exists kh \in KHACHHANG: pnp.MAKH = kh.MAKH$$

R <sub>23</sub>	Thêm	Xóa	Sửa
PHIEUNHANPHONG	+	-	+ (MAKH)
KHACHHANG	-	+	-*

- ❖ R<sub>24</sub>: ““Tập các giá trị MAPHONG được tìm thấy trong quan hệ PHIEUNHANPHONG phải được tìm thấy trong tập các giá trị MAPHONG trong quan hệ PHONG.””

RBTV tham chiếu

Bối cảnh: PHIEUNHANPHONG, PHONG

Nội dung:

$$\forall pnp \in PHIEUNHANPHONG, \exists p \in PHONG: pnp.MAPHONG = p.MAPHONG$$

R <sub>24</sub>	Thêm	Xóa	Sửa
PHIEUNHANPHONG	+	-	+ (MAPHONG)
PHONG	-	+	-*

- ❖ R<sub>25</sub>: ““Tập các giá trị MANV được tìm thấy trong quan hệ PHIEUNHANPHONG phải được tìm thấy trong tập các giá trị MANV trong quan hệ NHANVIEN.””

RBTV tham chiếu

Bối cảnh: PHIEUNHANPHONG, NHANVIEN

Nội dung:

$$\forall pnp \in PHIEUNHANPHONG, \exists nv \in NHANVIEN: pnp.MANV = nv.MANV$$

R <sub>25</sub>	Thêm	Xóa	Sửa

PHIEUNHANPHONG	+	-	+ (MANV)
NHANVIEN	-	+	-*

- ❖ **R<sub>26</sub>:** “Số người đi kèm ở phiếu nhận phòng phải nhỏ hơn hoặc bằng số lượng người ở mỗi loại phòng”

RBTV về chu trình

Bối cảnh: PHIEUNHANPHONG, LOAIPHONG, PHONG

Nội dung:

$\forall \text{ppn} \in \text{PHIEUNHANPHONG}, \exists \text{plp} \in (\text{PHONG} \bowtie_{\text{MALOAIPHONG}=\text{MALOAIPHONG}} \text{MAPHONG})$ :

(ppn.MAPHONG = plp.MAPHONG)  $\wedge$  (ppn.SONGUOIDIKEM  $\leq$  plp.SLNGUOI)

R <sub>26</sub>	Thêm	Xóa	Sửa
PHIEUNHANPHONG	+	-	+ (SONGUOIDIKEM, MAPHONG)
PHONG	-	+	-*
LOAIPHONG	-	+	+ (SLNGUOI)

- ❖ **R<sub>27</sub>:** “Ngày đến nhận phòng phải nhỏ hơn hoặc bằng ngày rời”

RBTV liên thuộc tính

Bối cảnh: PHIEUNHANPHONG

Nội dung:

$\forall \text{ppn} \in \text{PHIEUNHANPHONG}: \text{ppn.NGAYDEN} \leq \text{ppn.NGAYROI}$

R <sub>27</sub>	Thêm	Xóa	Sửa
PHIEUDATPHONG	+	-	+ (NGAYDEN, NGAYROI)

### 2.2.2.6 Bảng KHACHHANG

- ❖ **R<sub>28</sub>:** “Tất cả các khách hàng phải có mã số phân biệt nhau”

RBTV liên bộ

Bối cảnh: KHACHHANG

Nội dung:

$\forall k1, k2 \in \text{KHACHHANG}: k1 \neq k2 \rightarrow k1.MAKH \neq k2.MAKH$

R <sub>28</sub>	Thêm	Xóa	Sửa
-----------------	------	-----	-----

KHACHHANG	+	-	+ (MAKH)

- ❖ **R<sub>29</sub>:** “Tập các giá trị CMND trong quan hệ KHACHHANG phải là duy nhất.”

RBTV liên bộ

Bối cảnh: KHACHHANG

Nội dung:

$$\forall kh_1, kh_2 \in KHACHHANG: kh_1 \neq kh_2 \rightarrow kh_1 . CMND \neq kh_2 . CMND$$

R <sub>29</sub> KHACHHANG	Thêm +   Xóa -   Sửa + (CMND)

- ❖ **R<sub>30</sub>:** “Tập các giá trị SDT trong quan hệ KHACHHANG phải là duy nhất.”

RBTV liên bộ

Bối cảnh: KHACHHANG

$$\forall kh_1, kh_2 \in KHACHHANG: kh_1 \neq kh_2 \rightarrow kh_1 . SDT \neq kh_2 . SDT$$

R <sub>30</sub> KHACHHANG	Thêm +   Xóa -   Sửa + (SDT)

- ❖ **R<sub>31</sub>:** “Tập các giá trị EMAIL trong quan hệ KHACHHANG phải là duy nhất.”

RBTV liên bộ

Bối cảnh: KHACHHANG

Nội dung:

$$\forall kh_1, kh_2 \in KHACHHANG: kh_1 \neq kh_2 \rightarrow kh_1 . EMAIL \neq kh_2 . EMAIL$$

R <sub>31</sub> KHACHHANG	Thêm +   Xóa -   Sửa + (EMAIL)

- ❖ **R<sub>32</sub> :** “Giới tính của nhân viên là ‘Nam’ hoặc ‘Nu’”

RBTV miền giá trị

Bối cảnh: KHACHHANG

Nội dung:

$\forall kh \in KHACHHANG: kh.GIOITINH \in \{\text{'Nam'}, \text{'Nu'}\}$

R <sub>32</sub>	Thêm	Xóa	Sửa
KHACHHANG	+	-	+ (GIOITINH)

#### 2.2.2.7 Bảng HOADONTANHTOAN

- ❖ R<sub>33</sub>: “Tất cả hóa đơn có mã hóa đơn khác nhau”

RBTV liên bộ

Bối cảnh: HOADONTANHTOAN

Nội dung:

$\forall hd1, hd2 \in HOADONTANHTOAN: hd1 \neq hd2 \rightarrow hd1.MAHD \neq hd2.MAHD$

R <sub>33</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	- (*)

- ❖ R<sub>34</sub>: “Tập các giá trị MANV được tìm thấy trong quan hệ HOADONTANHTOA N phải được tìm thấy trong tập các giá trị MANV trong quan hệ NHANVIEN.”

RBTV tham chiếu

Bối cảnh: HOADONTANHTOAN, NHANVIEN

Nội dung:

$\forall hd \in HOADONTANHTOAN, \exists nv \in NHANVIEN: hd.MANV = nv.MANV$

R <sub>34</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (MANV)
NHANVIEN	-	+	-*

- ❖ R<sub>35</sub>: “Tập các giá trị MAPHIEUNHAN được tìm thấy trong quan hệ HOADONTANHTOAN phải được tìm thấy trong tập các giá trị MAPHIEUNHAN trong quan hệ PHIEUNHANPHONG.”

RBTV tham chiếu

Bối cảnh: HOADONTANHTOAN, PHIEUNHANPHONG

Nội dung:

$\forall hd \in HOADONTANHTOAN, \exists pnp \in PHIEUNHANPHONG: hd.MAPHIEUNHAN = php.PHIEUNHANPHONG$

R <sub>35</sub>	Thêm	Xóa	Sửa
-----------------	------	-----	-----

HOADONTANHTOAN	+	-	+ (MAPHIEUNHAN)
PHIEUNHANPHONG	-	+	-*

- ❖ **R<sub>36</sub>:** “Tiền phòng bằng giá phòng nhân với thời gian ở khách sạn”

RBTV liên thuộc tính - liên quan hệ

Bối cảnh: HOADONTANHTOAN, PHIEUNHANPHONG, LOAIPHONG

Nội dung:

$\forall hd \in HOADONTANHTOAN, \exists ppm \in (PHIEUNHANPHONG \bowtie_{MAPHONG=MAPHONG} (PHONG \bowtie_{MALOAIPHONG=MALOAIPHONG} MALOAIPHONG))$

$hd.MAPHIEUNHAN = ppm.MAPHIEUNHAN \rightarrow hd.TIENPHONG = ppm.GIAPHONG * (hd.NGAYROI - ppm.NGAYDEN)$

R <sub>36</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENPHONG, NGAYROI, NGAYDEN)
PHIEUNHANPHONG	-	+	- (*)
LOAIPHONG	-	+	+(GIAPHONG)
PHONG	-	+	-(*)

- ❖ **R<sub>37</sub>:** “Tiền dịch vụ trong hóa đơn bằng tổng thành tiền của các bảng kê dịch vụ”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, BANGKE\_DICHVU

Nội dung:

$\forall hd \in HOADONTANHTOAN, hd.TIENDICHVU = \sum_{(bk \in BANGKE_DICHVU : hd.MAHD = bk.MAHD)} (bk.THANHTIEN)$

R <sub>37</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENDICHVU)
BANGKE_DICHVU	-	+	+ (THANHTIEN)

- ❖ **R<sub>38</sub>:** “Tiền món ăn trong hóa đơn bằng tổng thành tiền của các bảng kê món ăn”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, BANGKE\_MONAN

Nội dung:

$\forall hd \in HOADONTANHTOAN, hd.TIENMONAN = \sum_{(bk \in BANGKE_MONAN : hd.MAHD = bk.MAHD)} (bk.THANHTIEN)$

R <sub>38</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENMONAN)
BANGKE_MONAN	-	+	+ (THANHTIEN)

- ❖ **R<sub>39</sub>:** “Tiền đền bù trong hóa đơn bằng tổng thành tiền của các hóa đơn CSVC”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, HOADON\_CSVC

Nội dung:

$$\forall \text{hd} \in \text{HOADONTANHTOAN}, \text{hd.TIENDICHVU} = \sum_{(\text{csvc} \in \text{HOADON_CSVC} : \text{hd.MAHD} = \text{csvc.MAHD})} (\text{csvc.THANHTIEN})$$

R <sub>39</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENDENBU)
HOADON_CSVC	-	+	+ (THANHTIEN)

- ❖ **R<sub>40</sub>:** “Nếu giá trị NGAYDEN trùng với giá trị NGAYLE thì giá trị TIENPHONG sẽ bằng tích giá trị TIENPHONG trong quan hệ HOADONTANHTOAN và GIAM trong quan hệ NGAYLE.”

RBTV liên thuộc tính – liên quan hệ

Bối cảnh: HOADONTANHTOAN, NGAYLE

Nội dung:

$$\forall \text{hd} \in \text{HOADONTANHTOAN}, \exists \text{nl} \in \text{NGAYLE}: \text{hd.NGAYDEN} = \text{nl.NGAYLE} \rightarrow \text{hd.TIENPHONG} = \text{hd.TIENPHONG} * \text{nl.GIAM}$$

R <sub>40</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (NGAYLE,NGAYDEN, TIENG IAM, TIENPHONG, GIAM)
NGAYLE	-	+	+ (GIAM)

- ❖ **R<sub>41</sub>:** “Nếu NGAYDEN trùng với NGAYLE thì TIENPHONG sẽ được giảm”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, NGAYLE

Nội dung:

$$\forall \text{hd} \in \text{HOADONTANHTOAN}, \exists \text{nl} \in \text{NGAYLE}:$$

$$\text{hd.NGAYDEN} = \text{nl.NGAYLE} \rightarrow \text{hd.TIENPHONG} = \text{hd.TIENPHONG} * \text{nl.GIAM}$$

R <sub>41</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENPHONG, TIENGAM)
NGAY_LE	-	+	+ (GIAM)

- ❖ **R<sub>42</sub>:** “Ngày đến khách sạn phải nhỏ hơn hoặc bằng ngày rời khách sạn.”

RBTV liên thuộc tính

Bối cảnh: HOADONTANHTOAN

Nội dung:  $\forall hd \in HOADONTANHTOAN: hd.NGAYDEN \leq hd.NGAYROI$

R <sub>42</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (NGAYDEN, NGAYROI)

- ❖ **R<sub>43</sub>:** “Tổng tiền của hóa đơn bằng tổng các dịch vụ, món ăn, csvc”

RBTV liên thuộc tính

Bối cảnh: HOADONTANHTOAN

Nội dung:

$\forall hd \in HOADONTANHTOAN:$

$$hd.TONGTIEN = SUM(hd.TIENPHONG, hd.TIENDICHVU, hd.TIENMONAN, hd.TIENGIAM)$$

R <sub>43</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	+	-	+ (TIENPHONG, TIENDICHVU, TIENMONAN, TIENDENU, TIENGIAM)

### 2.2.2.8 Bảng BANGKE\_DICHVU

- ❖ **R<sub>44</sub>:** “Tập các giá trị MADV được tìm thấy trong quan hệ BANGKE\_DICHVU phải được tìm thấy trong tập các giá trị MADV trong quan hệ DICHVU.”

RBTV tham chiếu

Bối cảnh: BANGKE\_DICHVU, DICHVU

Nội dung:

$$\forall bk \in BANGKE_DICHVU, \exists dv \in DICHVU: bk.MADV = dv.MADV$$

R <sub>44</sub>	Thêm	Xóa	Sửa
BANGKE_DICHVU	+	-	+ (MADV)
DICHVU	-	+	-*

- ❖ **R45:** “Tập các giá trị MAHD được tìm thấy trong quan hệ BANGKE\_DICHVU phải được tìm thấy trong tập các giá trị MAHD trong quan hệ HOADONTANHTOAN.”

RBTV tham chiếu

Bối cảnh: BANGKE\_DICHVU, HOADONTANHTOAN

Nội dung:

$$\forall \text{bk} \in \text{BANGKE\_DICHVU}, \exists \text{hd} \in \text{HOADONTANHTOAN}: \text{bk.MAHD} = \text{hd.MAHD}$$

<b>R<sub>45</sub></b>	Thêm	Xóa	Sửa
<b>BANGKE_DICHVU</b>	+	-	+ (MAHD)
<b>HOADONTANHTOAN</b>	-	+	-*

- ❖ **R46:** “Thời gian sử dụng dịch vụ phải trong khoảng ngày đến, ngày rời”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, BANGKE\_DICHVU

Nội dung:

$$\forall \text{hd} \in \text{HOADONTANHTOAN}, \exists \text{dv} \in \text{BANGKE\_DICHVU}: \text{hd.MAHD} = \text{dv.MAHD} \wedge \text{dv.NGAY} = < \text{hd.NGAYROI} \wedge \text{hd.NGAYDEN} = < \text{dv.NGAY}$$

<b>R<sub>46</sub></b>	Thêm	Xóa	Sửa
<b>HOADONTANHTOAN</b>	-	+	+ (NGAYDEN, NGAYROI)
<b>BANGKE_DICHVU</b>	+	-	+ (NGAY)

### 2.2.2.9 Bảng DICHVU

- ❖ **R47:** “Tất cả dịch vụ có mã dịch vụ khác nhau”

RBTV liên bộ

Bối cảnh: DICHVU

Nội dung:

$$\forall \text{dv1}, \text{dv2} \in \text{DICHVU}: \text{dv1} \neq \text{dv2} \rightarrow \text{dv1.MADV} \neq \text{dv2.MADV}$$

<b>R<sub>47</sub></b>	Thêm	Xóa	Sửa
<b>DICHVU</b>	+	-	- (*)

- ❖ **R48:** “Tên dịch vụ không trùng nhau”

RBTV liên bộ

Bối cảnh: DICHVU

Nội dung:

$\forall dv1, dv2 \in MONAN: dv1 \neq dv2 \rightarrow dv1.TENDV \neq dv2.TENDV$

R <sub>48</sub>	Thêm	Xóa	Sửa
DICHVU	+	-	+ (TENDV)

- ❖ **R<sub>49</sub>** : “Tình trạng dịch vụ là: “CO SAN”, “ TAM NGUNG”

RBTV miền giá trị

Bối cảnh: DICHVU

Nội dung:

$\forall dv \in DICHVU: dv.TINHTRANG \in \{‘Co san’, ‘Tam ngung’\}$

R <sub>49</sub>	Thêm	Xóa	Sửa
DICHVU	+	-	+ (TINHTRANG)

### 2.2.2.10 Bảng BANGKE\_MONAN

- ❖ **R<sub>50</sub>**: “Tập các giá trị MAMON được tìm thấy trong quan hệ BANGKE\_MONAN phải được tìm thấy trong tập các giá trị MAMON trong quan hệ MONAN.”

RBTV tham chiếu

Bối cảnh: BANGKE\_MONAN, MONAN

Nội dung:

$\forall bk \in BANGKE_MONAN, \exists ma \in MONAN: bk.MAMON = ma.MAMON$

R <sub>50</sub>	Thêm	Xóa	Sửa
BANGKE_MONAN	+	-	+ (MAMON)
MONAN	-	+	-*

- ❖ **R<sub>51</sub>**: “Tập các giá trị MAHD được tìm thấy trong quan hệ BANGKE\_MONAN phải được tìm thấy trong tập các giá trị MAHD trong quan hệ HOADONTANHTOAN.”

RBTV tham chiếu

Bối cảnh: BANGKE\_MONAN, HOADONTANHTOAN

Nội dung:

$\forall bk \in BANGKE_MONAN, \exists hd \in HOADONTANHTOAN: bk.MAHD = hd.MAHD$

R <sub>51</sub>	Thêm	Xóa	Sửa
BANGKE_MONAN	+	-	+ (MAHD)

HOADONTANHTOAN	-	+	-*
----------------	---	---	----

- ❖ **R<sub>52</sub>:** “Thời gian sử dụng dịch vụ phải trong khoảng ngày đến, ngày rời”

RBTV liên thuộc tính liên quan hệ

Bối cảnh: HOADONTANHTOAN, BANGKE\_MONAN

Nội dung:

$\forall hd \in HOADONTANHTOAN, \exists bk \in BANGKE\_MONAN :$

$hd.MAHD = bk.MAHD \wedge bk.NGAY = < hd.NGAYROI \wedge hd.NGAYDEN = < bk.NGAY$

R <sub>52</sub>	Thêm	Xóa	Sửa
HOADONTANHTOAN	-	+	+ (NGAYDEN, NGAYROI)
BANGKE_MONAN	+	-	+ (NGAY)

- ❖ **R<sub>53</sub>:** “Thành tiền của món ăn bằng số lượng nhân với giá mỗi món”

RBTV liên quan hệ

Bối cảnh: BANGKE\_MONAN, MONAN

Nội dung:

$\forall bk_ma \in BANGKE\_MONAN, \exists ma \in MONAN :$

$bk_ma.MAMON = ma.MAMON \wedge bk_ma.THANHTIEN = bk_ma.SOLUONG * ma.GIA$

R <sub>53</sub>	Thêm	Xóa	Sửa
BANGKE_MONAN	+	-	+ (THANHTIEN, SOLUONG)
MONAN	-	+	+ (GIA)

### 2.2.2.11 Bảng MONAN

- ❖ **R<sub>54</sub>:** “Tất cả món ăn có mã món ăn khác nhau”

RBTV liên bộ

Bối cảnh: MONAN

Nội dung:

$\forall ma1, ma2 \in MONAN : ma1 \neq ma2 \rightarrow ma1.MAMON \neq ma2.MAMON$

R <sub>54</sub>	Thêm	Xóa	Sửa
MONAN	+	-	- (*)

- ❖ **R<sub>55</sub>:** “Tên món ăn không trùng nhau”

RBTV liên bộ

Bối cảnh: MONAN

Nội dung:

$\forall ma1, ma2 \in MONAN: ma1 \neq ma2 \rightarrow ma1.TENDV \neq ma2.TENDV$

$R_{55}$	Thêm	Xóa	Sửa
MONAN	+	-	+ (MAMON)

- ❖ **R<sub>56</sub>** : “Tình trạng món ăn là: “”CO SAN”, “ TAM NGUNG”

RBTV miền giá trị

Bối cảnh: MONAN

Nội dung:

$\forall ma \in MONAN: ma.TINHTRANG \in \{‘Co san’, ‘Tam ngung’\}$

$R_{56}$	Thêm	Xóa	Sửa
MONAN	+	-	+ (TINHTRANG)

### 2.2.2.12 Bảng HOADON\_CSVC

- ❖ **R<sub>57</sub>**: “Tập các giá trị MACSVC được tìm thấy trong quan hệ HOADON\_CSVC phải được tìm thấy trong tập các giá trị MACSVC trong quan hệ CSVC.”

RBTV tham chiếu

Bối cảnh: HOADON\_CSVC, CSVC

Nội dung:

$\forall hd \in HOADON_CSVC, \exists cs \in CSVC: hd.MACSVC = cs.MACSVC$

$R_{57}$	Thêm	Xóa	Sửa
HOADON_CSVC	+	-	+ (MACSVC)
CSVC	-	+	-*

- ❖ **R<sub>58</sub>**: “Tập các giá trị MAHD được tìm thấy trong quan hệ HOADON\_CSVC phải được tìm thấy trong tập các giá trị MAHD trong quan hệ HOADONTHANHTOAN.”

RBTV tham chiếu

Bối cảnh: HOADON\_CSVC, HOADONTHANHTOAN

Nội dung:

$\forall cs \in HOADON_CSVC, \exists hd \in HOADONTHANHTOAN: cs.MAHD = hd.MAHD$

R <sub>58</sub>	Thêm	Xóa	Sửa
HOADON_CSVC	+	-	+ (MAHD)
HOADONTHANHTOAN	-	+	-*

- ❖ **R<sub>59</sub>:** “Thành tiền của hóa đơn CSVC bằng số lượng nhân với giá mỗi món”

RBTW liên thuộc tính liên quan hệ

Bối cảnh: HOADON\_CSVC, CSVC

Nội dung:

$\forall \text{hd\_cs} \in \text{HOADON_CSVC}, \exists \text{cs} \in \text{CSVC}:$

$\text{hd\_cs.MACSV} = \text{cs.MACSV} \wedge \text{hd\_cs.THANHTIEN} = \text{hd\_cs.SOLUONG} * \text{cs.GIA}$

R <sub>59</sub>	Thêm	Xóa	Sửa
HOADON_CSVC	+	-	+ (THANHTIEN, SOLUONG)
CSVC	-	+	+ (GIA)

### 2.2.2.13 Bảng CSVC

- ❖ **R<sub>60</sub>:** “Tất cả CSVC có mã khác nhau”

RBTW liên bộ

Bối cảnh: CSVC

Nội dung:

$\forall \text{cs1}, \text{cs2} \in \text{MONAN}: \text{cs1} \neq \text{cs2} \rightarrow \text{cs1.MACSV} \neq \text{cs2.MACSV}$

R <sub>60</sub>	Thêm	Xóa	Sửa
CSVC	+	-	- (*)

- ❖ **R<sub>61</sub>:** “Tên cơ sở vật chất không trùng nhau”

RBTW liên bộ

Bối cảnh: CSVC

Nội dung:

$\forall \text{cs1}, \text{cs2} \in \text{CSVC}: \text{cs1} \neq \text{cs2} \rightarrow \text{cs1.TENCSVC} \neq \text{cs2.TENCSVC}$

R <sub>61</sub>	Thêm	Xóa	Sửa
CSVC	+	-	+ (TENCSVC)

### 2.2.2.14 Bảng LOAITAIKHOAN

- ❖ **R<sub>62</sub>:** “Tất cả LOAITAIKHOAN phải có mã khác nhau”

RBTW liên bộ

Bối cảnh: LOAITAIKHOAN

Nội dung:

$\forall ltk1, ltk2 \in LOAITAIKHOAN: ltk1 \neq ltk2 \rightarrow ltk1.MALOAITK \neq ltk2.MALOAITK$

R <sub>60</sub> LOAITAIKHOAN	Thêm +	Xóa -	Sửa - (*)
---------------------------------	-----------	----------	--------------

- ❖ **R<sub>63</sub>:** “Tên LOAITAIKHOAN không trùng nhau”

RBTV liên bộ

Bối cảnh: LOAITAIKHOAN

Nội dung:

$\forall ltk1, ltk2 \in LOAITAIKHOAN: ltk1 \neq ltk2 \rightarrow ltk1.TENLOAITK \neq ltk2.TENLOAITK$

R <sub>63</sub> LOAITAIKHOAN	Thêm +	Xóa -	Sửa + (TENLOAITK)
---------------------------------	-----------	----------	----------------------

## Chương 3: CÁC CÔNG NGHỆ SỬ DỤNG

### 3.1 Oracle

Năm 1977, Lawrence Ellison đã thành lập công ty Oracle. Sau đó 2 năm công ty đã cho ra sản phẩm đầu tiên là CSDL Oracle phiên bản 2.0 tích hợp ngôn ngữ truy vấn câu trúc SQL. Oracle sau nhiều năm phát triển đã có thể cung cấp đa dạng các giải pháp lưu trữ và xử lý dữ liệu chuyên nghiệp cho các khách hàng theo nhiều hướng ứng dụng khác nhau và trở nên rộng rãi với người dùng.

Cơ sở dữ liệu Oracle (hay Oracle Database) là một tập các đơn vị dữ liệu. Mục đích của cơ sở dữ liệu là lưu trữ và tìm lại được các thông tin được lưu trữ. Máy chủ cơ sở dữ liệu là chìa khóa của việc giải quyết vấn đề quản lý thông tin. Thông thường một máy chủ máy chủ có thể quản lý một lớn dữ liệu trong môi trường có nhiều người dùng có thể đồng thời truy cập cùng dữ liệu. Tất cả được hoàn thành với hiệu năng cao. Máy chủ cơ sở dữ liệu có thể ngăn cản truy cập trái phép và cung cấp giải pháp hiệu quả trong việc lấy lại dữ liệu khi có bất kỳ sự cố nào.

Oracle Database là cơ sở dữ liệu đầu tiên được thiết kế chuyên dụng cho tính toán mạng lưới (grid computing), tính mềm dẻo nhất và chi phí hiệu quả nhất cho việc quản lý thông tin và ứng dụng. Kiến trúc mạng lưới tính toán của tổ chức tạo ra các vũng (pools) rộng lớn của ngành công nghiệp(industry-standard), modun hóa lưu trữ và máy chủ. Với kiến trúc này, mỗi hệ thống mới có thể nhanh chóng chuẩn bị từ vũng của các thành phần. Không cần áp lực cao, bởi vì sức chứa có thể dễ dàng thêm hoặc đặt lại từ nguồn tài nguyên chung nếu cần.

### 3.2 JDBC (Java Database Connectivity)

JDBC, là viết tắt của Java Database Connectivity, là một Java API chuẩn để kết nối giữa ngôn ngữ lập trình Java và các cơ sở dữ liệu đa dạng. Sử dụng JDBC, bạn có thể thực hiện nhiều tác vụ đa dạng khi

làm việc với cơ sở dữ liệu như tạo, xóa cơ sở dữ liệu; tạo và thực thi các lệnh SQL hoặc MySQL; tạo, xóa các bản ghi; ...

JDBC API cung cấp các Class và Interface sau:

DriverManager: Lớp này quản lý các Database Driver. Ánh xạ các yêu cầu kết nối từ ứng dụng Java với Data driver thích hợp bởi sử dụng giao thức kết nối phụ.

Driver: Interface này xử lý các kết nối với Database Server. Hiếm khi, bạn tương tác trực tiếp với các đối tượng Driver này. Thay vào đó, bạn sử dụng các đối tượng DriverManager để quản lý các đối tượng kiểu này.

Connection: Đối tượng Connection biểu diễn ngữ cảnh giao tiếp. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database.

Statement: Bạn sử dụng các đối tượng được tạo từ Interface này để đệ trình các lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung cấp nhận thêm các tham số để thực thi các thủ tục đã được lưu trữ.

ResultSet: Các đối tượng này giữ dữ liệu được thu nhận từ một Database sau khi bạn thực thi một truy vấn SQL. Nó đóng vai trò như một Iterator để cho phép bạn đọc qua dữ liệu của nó.

SQLException: Lớp này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database.

### 3.3 Java Swing

Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng Window-Based. Nó được xây dựng ở trên cùng của AWT (Abstract Windowing Toolkit) API và được viết hoàn toàn bằng Java.

Không giống AWT, Java Swing cung cấp các thành phần (Component) gọn nhẹ và độc lập nền tảng. Javax.swing. Package cung cấp các lớp cho Java Swing chẳng hạn như JButton, JTextField,

JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, ...

### 3.4 Netbeans

NetBeans là một môi trường phát triển tích hợp (IDE) cho Java. NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là modules. NetBeans chạy trên Windows, macOS, Linux và Solaris.

Có một số mô-đun trong NetBeans IDE. Trình lược tả Netbeans để giám sát các ứng dụng Java chủ yếu được sử dụng để tìm rò rỉ bộ nhớ, tối ưu hóa tốc độ và làm được nhiều hơn thế. Công cụ GUI hoặc Graphical User Interface là một trong những công cụ thiết kế và phát triển tốt nhất để phát triển các prototype và thiết kế Swing GUI bằng cách kéo, thả các GUI Components. NetBean cũng đã tích hợp NetBeans JavaScript Editor, mở rộng hỗ trợ cho Ajax, CSS và JavaScript cùng với những thứ khác

## CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC

### 4.1 Trigger trong Oracle

Trigger có thể hiểu đơn giản là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như Insert, Update and Delete. Trigger được lưu trữ và quản lý trong server database, được dùng trong trường hợp ta muốn kiểm tra các ràng buộc toàn vẹn trong Database.

Cú pháp:

```

CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER } triggering_event ON table_name
[FOR EACH ROW]
[FOLLOWS | PRECEDES another_trigger ]
[ENABLE | DISABLE]
[WHEN condition]
DECLARE
    Declaration statements
BEGIN
    Executable statements
EXCEPTION
    Exception_handling statements
END;

```

#### 4.1.1 Danh sách các trigger

STT	Tên	Thao tác	Bảng	Nội dung	Ghi chú
1	TRIG_INSERT_PHIEU DATTRUOC	INSERT	PHIEUDAT TRUOC	Kiểm tra đã tồn tại phiếu đặt trước trong hệ thống hay chưa	
2	TRIG_BK_DV_NGAYSD	Insert	BANGKE_DV	Kiểm tra ngày sử dụng dịch vụ có trong	

Đồ án HQT \_ Quản Lý Khách Sạn

				khoảng thời gian lưu trú ở khách sạn hay không	
3	TRIG_BK_MONAN_NGAYSD	Insert	BANGKE_MONAN	Kiểm tra ngày sử dụng món ăn có trong khoảng thời gian lưu trú ở khách sạn hay không	
4	TRIG_DATTRUOC_NGAY	Insert	PHIEUDAT TRUOC	Kiểm tra ngày đến và ngày rời trong phiếu có hợp lệ hay không	
5	TRIG_DELETE_PHIEU_NHANPHONG	delete	PHIEUNHAN PHONG	Thực hiện xóa các hóa đơn trước khi xóa phiếu nhận	
6	TRIG_HOADON_NGAY	INSERT UPDATE	HOADON THANHTOAN	Kiểm tra ngày đến và ngày rời trong phiếu có hợp lệ hay không	
7	TRIG_INSERT_PHIEU_NHANPHONG	Insert	PHIEUNHAN PHONG	Kiểm tra đã tồn tại phiếu nhận phòng hoặc phiếu đặt	

Đồ án HQT \_ Quản Lý Khách Sạn

				trước trùng với dữ liệu nhập vào	
8	TRIG_NV_NGAYVAOLAM	INSERT UPDATE	NHANVIEN	Kiểm tra ngày vào làm của nhân viên có hợp lệ hay không	
9	TRIG_PHIEUNHAN_NGAY	INSERT UPDATE	PHIEUNHAN PHONG	Kiểm tra ngày đến và ngày rời của phiếu nhận phòng có hợp lệ	
10	TRIG_PNP_TAOHD	INSERT	PHIEUNHAN PHONG	Tự động thực hiện sinh hóa đơn thanh toán với mã phiếu nhận vừa được tạo	
11	TRIG_PNP_UPDATE_HD	UPDATE	PHIEUNHAN PHONG	Tự động cập nhật lại thông tin hóa đơn khi phiếu nhận thay đổi thông tin	
12	TRIG_TUOI	INSERT UPDATE	NHANVIEN	Kiểm tra nhân viên có đủ tuổi để được nhận vào làm hay không	
13	TRIG_UPDATE_TINHTRANG PHONG	UPDATE	PHONG	Thực hiện tự động cập nhật	

				mã phòng cũng loại phòng cho phiếu đặt trước có mã phòng bị cập nhật thành “SỬA CHỮA”	
14	TRIG_PHIEUNHAN_ NGDIKEM	INSERT UPDATE	PHIEUNHAN PHONG	Kiểm tra số người đi kèm có đúng với quy định số lượng người tối đa cho loại phòng hay không.	
15	TRIG_LOAIPHONG_GIA	INSERT UPDATE	LOAIPHONG	Kiểm tra giá tiền của loại phòng phải thấp hơn giá tiền của loại phòng cao cấp hơn	

#### 4.1.2 Mô tả một số Trigger

##### a. Trigger kiểm tra giá phòng cho loại phòng

Tên trigger: **TRIG\_LOAIPHONG\_GIA**

Thao tác: Inser/ update

Trên bảng: **LOAIPHONG**

Các bước thực hiện:

[1]: Khai báo một con trỏ chứa mã loại phòng, giá phòng của các loại phòng khác với mã loại phòng mới được thêm vào.

[2]: Thực hiện vòng lặp cho tới khi không còn phần tử nào còn trong con trỏ:

[3]: Thực hiện so sánh giữa mã loại phòng, giá phòng được lấy ra từ con trỏ và mã loại phòng với giá phòng mới được thêm hoặc cập nhật.

- Nếu vi phạm điều kiện thì thông báo lỗi “**-20002,'GIA TIEN CUA LOAI PHONG PHAI THAP HON GIA TIEN CUA LOAI PHONG CAO CAP HON'**”. Quay lại bước [2].

Mã PL/SQL:

```
CREATE OR REPLACE TRIGGER TRG_LOAIPHONG_GIA
BEFORE INSERT OR UPDATE ON LOAIPHONG
FOR EACH ROW
DECLARE
    CURSOR C_GIAPHONG IS SELECT MALOAIPHONG, GIAPHONG FROM LOAIPHONG WHERE
    MALOAIPHONG <>:NEW.MALOAIPHONG;
    GIA_PHONG LOAIPHONG.GIAPHONG%TYPE;
    MALP LOAIPHONG.MALOAIPHONG%TYPE;
BEGIN
    OPEN C_GIAPHONG;
    LOOP
        FETCH C_GIAPHONG INTO MALP,GIA_PHONG ;
        EXIT WHEN C_GIAPHONG%NOTFOUND;
        IF (:NEW.MALOAIPHONG < MALP AND :NEW.GIAPHONG > GIA_PHONG) THEN
            RAISE_APPLICATION_ERROR(-20002, 'GIA TIEN CUA LOAI PHONG PHAI THAP HON
GIA TIEN CUA LOAI PHONG CAO CAP HON');
        END IF;
    END LOOP;
    CLOSE C_GIAPHONG;
END;
```

### b. Trigger kiểm tra số lượng người đi kèm có hợp quy định.

Tên trigger: TRIG\_PHIEUNHAN\_NGDIKEM

Thao tác: INSERT/UPDATE

Trên bảng: PHIEUNHANPHONG

Các bước thực hiện:

[1]: Thực hiện truy vấn mã loại phòng, số lượng người quy định của phòng thích hợp với điều kiện mã phòng = mã phòng mới.

[2]: Tiến hành so sánh số lượng người với số người đi kèm mới thêm vào

- Nếu số người đi kèm >= số lượng người quy định thì thông báo lỗi: **-20000, 'SO LUONG NGUOI DI KEM TOI DA LA ' || SL\_NGUOI**.

Mã PL/SQL:

```
CREATE OR REPLACE TRIGGER TRIG_PHIEUNHAN_NGDIKEM
BEFORE INSERT OR UPDATE ON PHIEUNHANPHONG
FOR EACH ROW
DECLARE
    SL_NGUOI LOAIPHONG.SLNGUOI%TYPE;
    MALP LOAIPHONG.MALOAIPHONG%TYPE;
BEGIN
    SELECT LOAIPHONG.MALOAIPHONG, SLNGUOI INTO MALP, SL_NGUOI
    FROM PHONG INNER JOIN LOAIPHONG ON PHONG.MAPHONG = LOAIPHONG.MAPHONG
    WHERE MAPHONG = :NEW.MAPHONG;

    IF (:NEW.SONGUOIDIKEM > SL_NGUOI) THEN
        RAISE_APPLICATION_ERROR(-20000, 'SO LUONG NGUOI DI KEM TOI DA LA ' ||
        SL_NGUOI);
    END IF;
END;
```

```
END IF;  
END;
```

## 4.2 Stored Procedure:

### 4.2.1 Stored Procedure trong Oracle:

Procedure (Thủ tục) dùng để gom một nhóm lệnh SQL cùng xử lý một mục đích cụ thể nào đó. Về bản chất giống với hàm, thủ tục trong các ngôn ngữ lập trình khác.

Stored procedure giúp giảm thời gian giao tiếp giữa các ứng dụng với hệ quản trị. Thay vì gửi một/mehrere câu lệnh dài thì chỉ cần gọi tới thủ tục và trong thủ tục này sẽ thực hiện nhiều câu lệnh SQL.

Cú pháp:

```
CREATE [OR REPLACE] PROCEDURE procedure_name (parameter_list)  
IS  
[declaration statements]  
BEGIN  
[execution statements]  
EXCEPTION  
[exception handler]  
END [procedure_name];
```

### 4.2.2 Transaction trong Oracle

Transaction là một tập các công việc có tính logic bao gồm một hay nhiều câu lệnh SQL. Một transaction là một đơn vị nguyên tố. Những ảnh hưởng của tất cả các lệnh trong transaction có thể được committed tất cả hoặc rolled back tất cả. Tất cả transaction tuân theo tất cả các tính chất ACID của một database transaction. Tính chất ACID được mô tả như sau:

- Atomicity (tính nguyên tố): Hoặc toàn bộ hoạt động của transaction được phản ánh đúng đắn trong CSDL hoặc không có hoạt động nào cả.
- Consistency (tính nhất quán): Một transaction được thực hiện độc lập với các giao tác khác xử lý đồng thời với nó để đảm bảo tính nhất quán cho CSDL.
- Isolation (tính cô lập): Một giao tác không quan tâm đến các giao tác xử lý đồng thời với nó.
- Durability (tính bền vững): Mọi thay đổi mà giao tác thực hiện trên CSDL phải được ghi nhận bền vững.

Rollback một transaction chưa được commit sẽ hủy bất cứ một thay đổi nào đối với data được thực thi bởi các câu lệnh SQL trong transaction. Sau khi một transaction bị roll back, những tác động bởi các công việc trong transaction lên database sẽ không còn tồn tại. Nếu không đặt bất kì savepoints nào trong transaction, Oracle sẽ thực hiện các hành động sau:

- Hủy toàn bộ thay đổi gây ra bởi tất cả các lệnh SQL trong transaction bằng phản hồi trong tập undo.

- Giải phóng tất cả các khóa đang giữ bởi transaction
- Xóa tất cả savepoints trong transaction
- Kết thúc transaction.

Commit và kết thúc một transaction hiện tại và lưu tất cả sự thay đổi mà transaction ảnh hưởng. Khi transaction commit, các sự kiện sẽ xảy ra:

- Giải phóng tất cả các khóa mà transaction đang giữ trên dòng và bảng.
- Xóa các savepoints.
- Hoàn tất transaction.

Cấu trúc của một transaction: Gồm 1 hay nhiều câu truy vấn, có thể bao gồm DDL, DML.

Transaction bắt đầu khi câu lệnh SQL đầu tiên được thực thi bao gồm DDL, DML hoặc lệnh SET TRANSACTION. Khi một transaction bắt đầu, hệ quản trị sẽ gán nó vào undo data segment. Có thể kết thúc 1 transaction khi: gặp lệnh COMMIT, ROLLBACK, khi user ngắt kết nối đến hệ quản trị một cách đột ngột.

#### 4.2.3 Danh sách Stored Procedure

S T T	Tên	Tham số đầu vào	Tham số đầu ra	Ý nghĩa	Ghi chú
1	PRO_UPDATE_PHI EUDAT TRUOC	MAPHIEU_IN INT, MAPHONG_IN INT, MAKH_IN INT, MANV_IN INT, NGAYDEN_IN DATE, NGAYROI_IN DATE, SONGUOIDIKEM_IN INT, TINHTRANG_IN PHIEUDATTRUOC. TINHTRANG%TYPE		Cập nhật thông tin phiếu đặt trước	
2	PRO_UPDATE_PHI EUNHAN PHONG	MAPHIEUNHAN_IN INT, MAKH_IN INT, MAPHONG_IN INT, MANV_IN INT, SONGUOIDIKEM_IN INT, NGAYDEN_IN DATE, NGAYROI_IN DATE		Cập nhật phiếu nhân phòng	
3	PROD_HOADONTH ANHTOAN _TONGTIEN	MA_HD INT		Tính tổng tiền của hóa đơn	

				thanh toán	
4	PROD_BK_MONAN_THANHTIEN	MA_HD INT, MA_MON INT, NGAY_SD DATE		Tính thành tiền bảng kê món ăn	
5	SLEEP	IN_TIME NUMBER			
6	FUNC_CHECK_PHONGTRONG	MAPHONG_IN INT, NGAYDEN_IN DATE, NGAYROI_IN DATE, MAPHIEU_IN INT	CHECK_PHONGTRONG NUMBER	Kiểm tra phòng có trống không.	
7	FUNC_HOADON_TIENTHENPHONG	MA_HD HOADONTHANH TOAN.MAHD%TYPE		Tính tiền phòng	
8	FUNC_HOADON_DICHVU	MA_HD HOADONTHANHTOA N.MAHD%TYPE	HOADONT HANH TOAN.TIEN DICHVU%T YPE	Tính tiền dịch vụ	
9	FUNC_HOADON_MONAN	MA_HD HOADONTHANH TOAN.MAHD%TYPE	HOADONT HANHTOA N.TIENMO NAN%TYPE	Tính tiền món ăn	
10	FUNC_HOADON_DENBU	MAHD HOADONTHANH TOAN.MAHD%TYPE	HOADONT HANHTOA N.TIENDEN BU%TYPE	Tính tiền đèn bù CSVC	
11	FUNC_HOADON_TIENGIAM	MAHD HOADONTHANHTOA N.MAHD%TYPE	HOADONT HANHTOA N.TIENGIA M%TYPE	Tính tiền được giảm	

### 4.3 Mô tả một số Stored Procedure

#### 4.3.1 Cập nhật thông tin Phiếu đặt trước

Tên: PRO\_UPDATE\_PHIEUDATTRUOC

Nội dung: Cập nhật thông tin phiếu đặt trước

Tham số đầu vào: MAPHIEU\_IN, MAPHONG\_IN, MAKH\_IN, MANV\_IN, NGAYDEN\_IN, NGAYROI\_IN, SOLUONGDIKEM\_IN, TINHTRANG\_IN

Tham số đầu ra: Không có.

Các bước thực hiện:

[1]: Truy vấn mã phòng, ngày đến, ngày rời, tình trạng của phiếu đặt trước với mã phiếu đầu vào đã được lưu dưới CSDL.

[2]: So sánh tình trạng đầu vào với tình trạng đã được lưu dưới CSDL:

[2.1]: TH1: Nếu cả 2 tình trạng đều là “CHUA NHAN” thì:

So sánh tình trạng ngày đến đưa vào với ngày đến dưới CSDL và kiểm tra phòng đó có trống trong khoảng ngày đến và ngày rời nhập vào không?:

[2.1.1] TH1: Nếu muốn dời ngày đến trước ngày lúc đầu đã đặt và ngày đến đã đặt trước ngày hiện tại trong hệ thống và phòng trống trong khoảng ngày đến và ngày rời mới nhập vào thì cập nhật lại thông tin của phiếu đặt trước.

[2.1.2] TH2: Nếu muốn dời ngày đến trước ngày lúc đầu đã đặt và ngày đến đã đặt trước ngày hiện tại trong hệ thống và phòng bận trong khoảng ngày đến và ngày rời mới nhập vào thì hiện ra thông báo “Không thể cập nhật. Xung đột với một phiếu đặt trước/ phiếu nhận phòng”

[2.1.3] TH3: Nếu muốn dời ngày đến sau ngày lúc đầu đã đặt và ngày đến đã đặt trước ngày hiện tại trong hệ thống và phòng trống trong khoảng ngày đến và ngày rời mới nhập vào thì cập nhật lại thông tin phiếu đặt trước.

[2.1.4] TH4: Nếu muốn dời ngày đến sau ngày lúc đầu đã đặt và ngày đến đã đặt trước ngày hiện tại trong hệ thống và phòng bận trong khoảng thời ngày đến và ngày rời mới nhập vào thì hiện ra thông báo: “Không thể cập nhật. Xung đột với một phiếu đặt trước/ phiếu nhận phòng”.

[2.1.5] TH5: Nếu giữ nguyên ngày đến và thay đổi ngày rời và phòng trống trong khoảng ngày đến và ngày rời thì cập nhật lại thông tin phiếu đặt trước.

[2.1.6] TH6: Nếu giữ nguyên ngày đến và thay đổi ngày rời và phòng bận trong khoảng ngày đến và ngày rời thì hiện ra thông báo “Không thể cập nhật. Xung đột với một phiếu đặt trước/ phiếu nhận phòng”.

[2.1.7] TH7: Nếu giữ nguyên ngày đến và ngày rời chỉ thay đổi phòng muốn đặt, nếu phòng trống trong khoảng thời gian ngày đến và ngày rời thì cập nhật lại thông tin phiếu đặt trước.

[2.1.8] TH8: Nếu phòng bận thì hiện ra thông báo: “Không thể cập nhật. Xung đột với một phiếu đặt trước/ phiếu nhận phòng”.

[2.2] TH2: Nếu cập nhật tình trạng của phiếu đặt trước thành đã nhận và muốn đến trước ngày đã đặt. Kiểm tra tình trạng phòng trong khoảng thời gian ngày đến và ngày rời:

[2.2.1] TH1: Nếu phòng trống thì thực hiện cập nhật lại tình trạng phiếu đặt trước và thêm phiếu nhận phòng.

[2.2.2] TH2: Nếu phòng bận thì hiện ra thông báo “Không thể nhận phòng trước ngày đặt, vì phòng hiện đang ở trạng thái bận”.

[2.3] TH3: Nếu khách hàng muốn gia hạn ngày rời và nhận phòng vào đúng ngày đã đặt trước đó thì:

## Đồ án HQT \_ Quản Lý Khách Sạn

[2.3.1]: Nếu phòng đó trống trong khoảng thời gian mà khách hàng muốn thì cập nhật lại tình trạng của phiếu đặt trước thành đã nhận và tạo phiếu nhận phòng.

[2.3.2]: Nếu phòng đó bận thì in ra thông báo: “Không thể gia hạn ngày rời, vì đã có phiếu đặt trước”

[2.4] TH4: Nếu tình trạng của phiếu đặt trước đã nhận nhưng người dùng lại cập nhật lại tình trạng thành chưa nhận thì in ra thông báo “ Không thể cập nhật lại tình trạng khi phòng đã nhận”.

Mã PL/SQL:

```
CREATE OR REPLACE PROCEDURE PRO_UPDATE_PHIEUDATTRUOC(MAPHIEU_IN IN INT,
                                                       MAPHONG_IN IN INT,
                                                       MAKH_IN IN INT,
                                                       MANV_IN IN INT,
                                                       NGAYDEN_IN IN DATE,
                                                       NGAYROI_IN IN DATE,
                                                       SONGUOIDIKEM_IN IN INT,
                                                       TINHTRANG_IN IN
PHIEUDATTRUOC.TINHTRANG%TYPE)
AS
V_MAPHONG PHIEUDATTRUOC.MAPHONG%TYPE;
V_NGAYDEN PHIEUDATTRUOC.NGAYDEN%TYPE;
V_NGAYROI PHIEUDATTRUOC.NGAYROI%TYPE;
V_TINHTRANG PHIEUDATTRUOC.TINHTRANG%TYPE;
BEGIN
    SELECT MAPHONG, NGAYDEN, NGAYROI, TINHTRANG INTO V_MAPHONG, V_NGAYDEN,
V_NGAYROI,V_TINHTRANG FROM PHIEUDATTRUOC WHERE MAPHIEU = MAPHIEU_IN;
    IF TINHTRANG_IN = 'CHƯA NHẬN' AND V_TINHTRANG = 'ĐÃ NHẬN' THEN
        RAISE_APPLICATION_ERROR(-20013,'DA NHAN PHONG, KHONG THE CAP NHAT LAI
TINH TRANG');
    END IF;

    IF TINHTRANG_IN = 'CHƯA NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' THEN
        BEGIN
            IF NGAYDEN_IN < V_NGAYDEN AND V_NGAYDEN > SYSDATE AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 1
THEN
```

## Đồ án HQT \_ Quản Lý Khách Sạn

```
UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN = NGAYDEN_IN,
NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN
WHERE MAPHIEU = MAPHIEU_IN;

ELSIF NGAYDEN_IN < V_NGAYDEN AND V_NGAYDEN > SYSDATE AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 0
THEN

    RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT VOI MOT
PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');

ELSIF NGAYDEN_IN > V_NGAYDEN AND V_NGAYDEN >= SYSDATE AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 1
THEN

    UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN = NGAYDEN_IN,
NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN
WHERE MAPHIEU = MAPHIEU_IN;

ELSIF NGAYDEN_IN > V_NGAYDEN AND V_NGAYDEN >= SYSDATE AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 0
THEN

    RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT VOI MOT
PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');

ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN <> V_NGAYROI AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 1
THEN

    UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYROI = NGAYROI_IN,
SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN
WHERE MAPHIEU = MAPHIEU_IN;

ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN <> V_NGAYROI AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 0
THEN

    RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT VOI MOT
PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');

ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN = V_NGAYROI AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 1
THEN

    UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, SONGUOIDIKEM =
SONGUOIDIKEM_IN, MANV = MANV_IN, NGAYDEN = NGAYDEN_IN, NGAYROI = NGAYROI_IN
WHERE MAPHIEU = MAPHIEU_IN;

ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN = V_NGAYROI AND
FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) = 0
THEN
```

## Đồ án HQT \_ Quản Lý Khách Sạn

```
    RAISE_APPLICATION_ERROR(-20014, 'KHONG THE CAP NHAT, XUNG DOT VOI MOT  
PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');

    END IF;

    END;

END IF;

IF TINHTRANG_IN = 'ĐÃ NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' AND V_NGAYDEN >  
SYSDATE AND NGAYROI_IN = V_NGAYROI THEN

BEGIN

    IF FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN,  
MAPHIEU_IN) = 1 THEN

        BEGIN

            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN =  
NGAYDEN_IN, NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV =  
MANV_IN, TINHTRANG = TINHTRANG_IN

            WHERE MAPHIEU = MAPHIEU_IN;

            INSERT INTO PHIEUNHANPHONG(MAPHIEUNHAN, MAKH, MAPHONG, MANV,  
SONGUOIDIKEM, NGAYDEN, NGAYROI)

            VALUES(SEQ_PHIEUNHAN.NEXTVAL, MAKH_IN, MAPHONG_IN, MANV_IN,  
SONGUOIDIKEM_IN, NGAYDEN_IN, NGAYROI_IN);

        END;

    ELSE

        RAISE_APPLICATION_ERROR(-20015, 'KHONG THE NHAN PHONG TRUOC NGAY DAT,  
VI PHONG HIEN DANG O TRANG THAI BAN');

    END IF;

END;

ELSIF TINHTRANG_IN = 'ĐÃ NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' AND NGAYROI_IN  
> V_NGAYDEN AND V_NGAYDEN = SYSDATE THEN

BEGIN

    IF FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN)  
= 1 THEN

        BEGIN

            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYROI =  
NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN, TINHTRANG =  
TINHTRANG_IN

            WHERE MAPHIEU = MAPHIEU_IN;
```

```
        INSERT INTO PHIEUNHANPHONG(MAPHIEUNHAN, MAKH, MAPHONG, MANV,
SONGUOIDIKEM, NGAYDEN, NGAYROI)

            VALUES (SEQ_PHIEUNHAN.NEXTVAL, MAKH_IN, MAPHONG_IN, MANV_IN,
SONGUOIDIKEM_IN, NGAYDEN_IN, NGAYROI_IN);

        END;

        ELSE

            RAISE_APPLICATION_ERROR(-20016, 'KHONG THE GIA HAN NGAY ROI, VI CO
PHIEU DAT TRUOC CHO PHONG');

        END IF;

        END;

        END IF;

        SLEEP(10);

        COMMIT;

END;
```

#### 4.3.2 Thanh toán hóa đơn

Tên: PROD\_HOADONTHANHTOAN\_TONGTIEN

Nội dung: Thanh toán hóa đơn

Tham số đầu vào: MA\_HD

Tham số đầu ra: Không có

Các bước thực hiện:

[1]: Tính tiền phòng qua hàm FUNC\_HOADON\_TIENPHONG

[2]: Tính tiền dịch vụ qua hàm FUNC\_HOADON\_DICHVU

[3]: Tính tiền món ăn qua hàm FUNC\_HOADON\_MONAN

[4]: Tính tiền đèn bù qua hàm FUNC\_HOADON\_DENBU

[5]: Tính tiền khuyến mãi qua hàm FUNC\_HOADON\_TIENGIAM

[6]: Cập nhật cái thuộc tính của hóa đơn theo MA\_HD

[6.1]: Cập nhật TIENPHONG, TIENDICHVU, TIENMONAN, TIENDENBU, TIENGIAM theo số tiền đã được tính ở trên.

[6.2]: Cập nhật TONGTIEN = TIEN\_PHONG + TIEN\_DV + TIEN\_MA + DEN\_BU - TIEN\_GIAM và TRANGTHAI = 'ĐÃ THANH TOÁN'

Mã PL/SQL :

```

CREATE OR REPLACE PROCEDURE PROD_HOADONTHANHTOAN_TONGTIEN (MA_HD IN
HOADONTHANHTOAN.MAHD%TYPE)
AS
    TIEN_PHONG      HOADONTHANHTOAN.TIENPHONG%TYPE;
    TIEN_DV         HOADONTHANHTOAN.TIENDICHVU%TYPE;
    TIEN_MA         HOADONTHANHTOAN.TIENMONAN%TYPE;
    DEN_BU          HOADONTHANHTOAN.TIENDENBU%TYPE;
    TIEN_GIAM       HOADONTHANHTOAN.TIENGIAM%TYPE;
    PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    TIEN_PHONG := FUNC_HOADON_TIENPHONG(MA_HD);
    TIEN_DV := FUNC_HOADON_DICHVU(MA_HD);
    TIEN_MA := FUNC_HOADON_MONAN(MA_HD);
    DEN_BU := FUNC_HOADON_DENBU(MA_HD);
    TIEN_GIAM := FUNC_HOADON_TIENGIAM(MA_HD);
    UPDATE HOADONTHANHTOAN
    SET TIENPHONG = TIEN_PHONG,
        TIENDICHVU = TIEN_DV,
        TIENMONAN = TIEN_MA,
        TIENDENBU = DEN_BU,
        TIENGIAM = TIEN_GIAM,
        TONGTIEN = TIEN_PHONG + TIEN_DV + TIEN_MA + DEN_BU - TIEN_GIAM,
        TRANGTHAI = 'ĐÃ THANH TOÁN'
    WHERE MAHD = MA_HD;
    COMMIT;
END;

```

#### 4.3.3 Tính tiền bảng kê món ăn

Tên : PROD\_BK\_MONAN\_THANHTIEN

Nội dung: Tính tiền bảng kê món ăn

Tham số đầu vào: MA\_HD, MA\_MON, NGAY\_SD.

Tham số đầu ra: Không có

Các bước thực hiện :

[1] : Truy vấn giá món ăn theo mã món từ tham số truyền vào

[2] : Cập nhật THANHTIEN của bảng kê theo mã hóa đơn, mã món và ngày sử dụng từ tham số truyền vào với THANHTIEN = GIAMON \*SOLUONG

Mã PL/SQL

```

CREATE OR REPLACE PROCEDURE PROD_BK_MONAN_THANHTIEN
    (MA_HD IN BANGKE_MONAN.MAHD%TYPE, MA_MON IN BANGKE_MONAN.MAMON%TYPE,
    NGAY_SD IN BANGKE_MONAN.NGAY%TYPE)
AS
    GIA    MONAN.GIAMON%TYPE;
BEGIN
    SELECT GIAMON INTO GIA
    FROM MONAN

```

```
WHERE MAMON = MA_MON;

UPDATE BANGKE_MONAN
SET THANHTIEN = GIA * SOLUONG
WHERE MAHD = MA_HD AND NGAY = NGAY_SD AND MAMON = MA_MON;
END;
```

#### 4.3.4 Tính tiền phòng

Tên : FUNC\_HOADON\_TIENPHONG

Nội dung: Tính tiền phòng

Tham số đầu vào: MA\_HD

Tham số đầu ra: HOADONTANTHOAN.TIENPHONG%TYPE

Các bước thực hiện:

[1]: Truy vấn NGAYROI – NGAYDEN vào NGAY (INT) của hóa đơn từ tham số truyền vào

[2]: Truy vấn GIAPHONG từ bảng LOAIPHONG qua việc kết 4 bảng (HOADONTANTHOAN, PHIEUNHANPHONG, LOAIPHONG và PHONG) của mã hóa đơn từ tham số đầu vào.

[3]: Trả về giá tiền của phòng với GIA\_PHONG \* NGAY.

Mã PL/SQL

```
CREATE OR REPLACE FUNCTION FUNC_HOADON_TIENPHONG (MA_HD IN
HOADONTANTHOAN.MAHD%TYPE)
RETURN HOADONTANTHOAN.TIENPHONG%TYPE
AS
    NGAY           INT;
    GIA_PHONG     LOAIPHONG.GIAPHONG%TYPE;
BEGIN
    SELECT NGAYROI - NGAYDEN INTO NGAY
    FROM HOADONTANTHOAN
    WHERE MAHD = MA_HD;

    SELECT GIAPHONG INTO GIA_PHONG
    FROM HOADONTANTHOAN H
        JOIN PHIEUNHANPHONG N ON H.MAPHIEUNHAN = N.MAPHIEUNHAN
        JOIN PHONG P ON N.MAPHONG = P.MAPHONG
        JOIN LOAIPHONG L ON P.MALOAIPHONG = L.MALOAIPHONG
    WHERE MAHD = MA_HD;

    RETURN (GIA_PHONG * NGAY);

EXCEPTION
    WHEN NO_DATA_FOUND THEN RETURN 0;
END;
```

## CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI

### 5.1 Các mức cô lập trong Oracle

#### 5.1.1 Read Committed

- a. Trong Read Committed, mọi truy vấn được thực hiện bởi một transaction chỉ nhìn thấy dữ liệu đã được commit trước khi bắt đầu truy vấn – không phải là khi transaction bắt đầu. Mức cô lập này được Oracle chọn là mức cô lập mặc định. Nó phù hợp với môi trường cơ sở dữ liệu trong đó ít transaction có khả năng xung đột. Transaction thực hiện truy vấn dữ liệu trong mức cô lập read committed thì có tránh đọc dữ liệu mà chưa được commit. Tuy nhiên, bởi vì không ngăn cản các transaction khác cập nhật dữ liệu đọc bởi các truy vấn, hoặc các transaction có thể thay đổi dữ liệu giữa các lệnh truy vấn. Vì vậy, một transaction thực hiện cùng một lệnh truy vấn 2 lần có thể gây ra fuzzy reads and phantoms.

- b. Tính nhất quán ở mức cô lập Read Committed

Hệ quản trị cung cấp dữ liệu một cách nhất quán cho mỗi câu truy vấn, đảm bảo cho dữ liệu nhất quán.

Một (implicit query), như một lệnh truy vấn theo sau bởi mệnh đề WHERE trong lệnh UPDATE, được đảm bảo bởi một kết quả nhất quán. Tuy nhiên, implicit query sẽ không thấy được sự thay đổi, nhưng sẽ thấy dữ liệu đã tồn tại trước khi dữ liệu được thay đổi.

- c. Xung đột ghi ở mức cô lập Read Committed

Ở mức cô lập read committed, xung đột ghi xảy ra khi một transaction nỗ lực thay đổi một dòng dữ liệu được cập nhật bởi một transaction chưa được committed (uncommitted concurrent transaction). Transaction ngăn chặn sự thay đổi của dòng dữ liệu thường được gọi là blocking transaction. Mức cô lập Read Committed chờ cho blocking transaction kết thúc và giải phóng khóa trên dòng dữ liệu bị khóa.

Những lựa chọn mà mức cô lập Read Committed cung cấp:

- Nếu blocking transaction rolls back, sau đó transaction ghi thực hiện thay đổi dòng dữ liệu đã bị khóa trước đó như thể những transaction khác không tồn tại.
- Nếu blocking transaction committed và giải phóng khóa trên dòng dữ liệu đó, thì transaction đang chờ sẽ thực hiện cập nhật dữ liệu trên dòng dữ liệu đã được cập nhật.

#### 5.1.2 Serializable

Ở mức cô lập serializable isolation level, transaction chỉ thấy sự thay đổi đã được committed vào thời điểm transaction đã bắt đầu và sự thay đổi được thực hiện bởi chính transaction đó.

Mức cô lập serializable hoạt động trong môi trường chỉ một mình nó như thể không có người dùng nào khác đang sửa đổi dữ liệu. Mức cô lập serializable phù hợp với môi trường:

- Cơ sở dữ liệu với khối lượng lớn, transaction ngắn chỉ cập nhật vài dòng.
- Nơi mà cơ hội để 2 transaction cập nhật cùng những dòng dữ liệu là rất thấp.
- Nơi mà transaction có thời gian thực thi lâu chỉ được phép đọc.

Ở mức cô lập serializable, bất kì dữ liệu nào được đọc bởi transaction thì được đảm bảo rằng sẽ có cùng kết quả khi đọc lại dữ liệu cùng một lệnh truy vấn như trước. Sự thay đổi bởi những transaction

khác, câu truy vấn sẽ không thấy được sự thay đổi đó. Vì vậy ở mức độ serializable sẽ không xảy ra tình trạng dirty read, fuzzy read and phantom reads.

Hệ quản trị chỉ chấp nhận để serializable thay đổi dữ liệu chỉ khi sự thay đổi trên đơn vị dữ liệu đó được thay đổi bởi một transaction khác đã được committed trước khi transaction bắt đầu. Hệ quản trị sẽ in ra lỗi khi serializable cố gắng cập nhật hoặc xóa dữ liệu được thay đổi bởi transaction khác mà được commit sau khi serializable bắt đầu:

#### **ORA-08177: Cannot serialize access for this transaction**

Khi transaction serializable thực hiện thất bại và in ra lỗi ORA-08177, ứng dụng có thể đưa ra một số hành động, bao gồm như sau:

- Commit những công việc đã được thực thi cho tới điểm đó.
- Thực thi những câu lệnh bổ sung, sau khi roll back đến điểm savepoint đã được thiết lập trước khi thực hiện transaction
- Roll back những transaction đã được hoàn thành.

#### **5.1.3 Read-Only**

Mức độ Read-Only tương tự như mức độ Serializable, nhưng ở mức độ Read-Only transaction không chấp nhận dữ liệu thay đổi dữ liệu bởi các transaction trừ khi người dùng là SYS.

Read-Only transaction hữu ích cho việc xuất báo cáo .

Hệ quản trị Oracle thành công trong việc đọc dữ liệu một cách nhất quán bằng việc tái xây dựng dữ liệu nếu cần thiết từ tập undo.

### **5.2 Cơ chế khóa**

Hệ quản trị có thể thực thi một vài cơ chế khóa, tùy vào các thao tác khác nhau lên dữ liệu. Hệ quản trị sử dụng 2 loại chính là exclusive locks (khóa độc quyền) và share locks (khóa chia sẻ). Tại một thời điểm, chỉ có 1 khóa độc quyền thực thi trên 1 đơn vị dữ liệu nhưng sẽ có nhiều khóa chia sẻ cùng thực thi trên 1 đơn vị dữ liệu.

Một dòng dữ liệu chỉ bị khóa khi nó bị thay đổi bởi hành động ghi (chỉ khi cập nhật một dòng dữ liệu hệ quản trị mới thực thi câu lệnh UPDATE và yêu cầu khóa trên đơn vị dữ liệu đó). Cơ chế khóa sẽ ngăn các hành động ghi khác trên đơn vị dữ liệu đó. Hành động đọc dữ liệu sẽ không yêu cầu khóa trên đơn vị mà nó đang đọc. Chỉ khi thực hiện câu lệnh SELECT .... FOR UPDATE để yêu cầu khóa trên đơn vị dữ liệu mà nó đang đọc ngăn không cho các đơn vị dữ liệu khác yêu cầu được khóa ghi trên đơn vị dữ liệu đó. Khi một dữ liệu đang được thực hiện bởi một hành động ghi thì hệ quản trị sẽ trả lại giá trị trước đó đã được commit cho hành động đọc trên đơn vị dữ liệu đó.

Để đảm bảo được các tính chất của dữ liệu, hệ quản trị thực hiện cơ chế khóa. Cơ chế khóa diễn ra một cách tự động. Khi update trên một đơn vị dữ liệu hệ quản trị sẽ tự động row locking trên đơn vị dữ liệu đó và giải phóng khóa khi transaction commit.

Hệ quản trị cung cấp 3 chế độ cho cơ sở dữ liệu:

- Exclusive lock mode: Transaction nhận được exclusive lock khi nó có nhu cầu thay đổi dữ liệu. Các transaction khác sẽ không thể thao tác trên đơn vị dữ liệu khi transaction giải phóng khóa.
- Share lock mode: Chế độ này cho phép chia sẻ cùng 1 đơn vị dữ liệu. Nhiều thao tác độc có thể chia sẻ dữ liệu cho nhau, mỗi transaction giữ một share lock để ngăn cản hành động ghi vào cùng một thời điểm trên 1 đơn vị dữ liệu.

### 5.3 Một số vấn đề xảy ra khi nhiều giao tác truy xuất đồng thời

#### 5.3.1 Dirty reads

Xảy ra khi transaction thực hiện đọc dữ liệu được ghi bằng transaction khác mà hành động ghi đó chưa được commit.

#### 5.3.2 Non – repeatable reads

Xảy ra khi transaction thực hiện đọc lại dữ liệu đã được đọc lần trước và nhận thấy một transaction khác đã thực hiện cập nhật và xóa dữ liệu.

Ví dụ: Người dùng truy vấn một câu lệnh truy vấn và sau đó truy vấn dữ liệu bằng câu lệnh đã thực hiện trước đó nhưng phát hiện rằng dữ liệu bị thay đổi.

#### 5.3.3 Phantom reads

Xảy ra khi transaction thực hiện lại câu truy vấn và trả lại số dòng phù hợp với điều kiện tìm kiếm và nhận thấy một transaction khác đã thực hiện thêm một dòng dữ liệu bổ sung vào phù hợp với điều kiện.

Ví dụ: Transaction thực hiện truy vấn số lượng nhân viên. Năm phút sau, thực hiện lại cùng 1 câu truy vấn nhưng số lượng đã tăng thêm 1 bởi vì có một người dùng khác thêm một dòng dữ liệu cho nhân viên mới. Số lượng tăng thêm hơn trước nhưng khác với fuzzy reads thì dữ liệu trước không bị thay đổi.

#### 5.3.4 Lost update

Xảy ra khi một hành động ghi bị ghi đè bởi sự thay đổi bởi một hành động ghi khác trên cùng một đơn vị dữ liệu.

#### 5.3.5 Deadlock

Deadlock xảy ra khi một hay nhiều người dùng đang chờ yêu cầu khóa trên đơn vị dữ liệu mà đang bị người dùng khác kia giữ khóa trên đơn vị dữ liệu đó. Deadlock ngăn các transaction tiếp tục thực hiện.

Hệ quản trị tự động phát hiện và xử lý deadlock bằng cách roll back một transaction gây deadlock, giải phóng 1 dòng đang bị khóa và trả về thông báo cho transaction bị rollback.

### 5.4 Mô tả đồ án môn học

#### 5.4.1 Lost update

- a. Mô tả tình huống: Một nhân viên tiến hành cập nhật lại ngày đến và ngày rời cho một phiếu đặt trước của khách hàng. Vì một lý do nào đó phòng mà phiếu đặt trước của nhân viên đang cập nhật cần phải sửa chữa, người quản lý tiến hành cập nhật lại tình trạng phòng thành sửa

chữa và cập nhật lại phiếu đặt trước đó một phòng trống cùng loại. Khi nhân viên thực hiện cập nhật và commit dữ liệu thì người quản lí cũng thực hiện cập nhật và kết thúc, dẫn đến việc dữ liệu của quản lý cập nhật trước bị ghi đè bởi dữ liệu mà nhân viên cập nhật sau và dẫn tới sai sót.

b. Mã PL/SQL:

i. Procedure SLEEP

```
CREATE OR REPLACE PROCEDURE SLEEP(IN_TIME NUMBER)
AS
V_NOW DATE;
BEGIN
    SELECT SYSDATE INTO V_NOW FROM DUAL;
    LOOP
        EXIT WHEN V_NOW + (IN_TIME * (1/86400)) <= SYSDATE;
    END LOOP;
END;
```

ii. Procedure PRO\_UPDATE\_PHIEUDATTRUOC

```
CREATE OR REPLACE PROCEDURE PRO_UPDATE_PHIEUDATTRUOC(MAPHIEU_IN IN INT,
                                                       MAPHONG_IN IN INT,
                                                       MAKH_IN IN INT,
                                                       MANV_IN IN INT,
                                                       NGAYDEN_IN IN DATE,
                                                       NGAYROI_IN IN DATE,
                                                       SONGUOIDIKEM_IN IN
                                                       INT,
                                                       TINHTRANG_IN IN
                                                       PHIEUDATTRUOC.TINHTRANG%TYPE)
AS
V_MAPHONG PHIEUDATTRUOC.MAPHONG%TYPE;
V_NGAYDEN PHIEUDATTRUOC.NGAYDEN%TYPE;
V_NGAYROI PHIEUDATTRUOC.NGAYROI%TYPE;
V_TINHTRANG PHIEUDATTRUOC.TINHTRANG%TYPE;
BEGIN
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
    --set transaction isolation level serializable;
    SELECT MAPHONG, NGAYDEN, NGAYROI, TINHTRANG INTO V_MAPHONG,
    V_NGAYDEN, V_NGAYROI, V_TINHTRANG FROM PHIEUDATTRUOC WHERE MAPHIEU =
    MAPHIEU_IN;
    IF TINHTRANG_IN = 'CHƯA NHẬN' AND V_TINHTRANG = 'ĐÃ NHẬN' THEN
        RAISE_APPLICATION_ERROR(-20013,'DA NHAN PHONG, KHONG THE CAP NHAT
LAI TINH TRANG');
    END IF;

    IF TINHTRANG_IN = 'CHƯA NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' THEN
```

## Đồ án HQT \_ Quản Lý Khách Sạn

```
BEGIN
    IF NGAYDEN_IN < V_NGAYDEN AND V_NGAYDEN > SYSDATE AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    1 THEN
        UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN =
    NGAYDEN_IN, NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV
    = MANV_IN
            WHERE MAPHIEU = MAPHIEU_IN;
        ELSIF NGAYDEN_IN < V_NGAYDEN AND V_NGAYDEN > SYSDATE AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    0 THEN
            RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT
VOI MOT PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');
        ELSIF NGAYDEN_IN > V_NGAYDEN AND V_NGAYDEN >= SYSDATE AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    1 THEN
            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN =
    NGAYDEN_IN, NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV
    = MANV_IN
            WHERE MAPHIEU = MAPHIEU_IN;
        ELSIF NGAYDEN_IN > V_NGAYDEN AND V_NGAYDEN >= SYSDATE AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    0 THEN
            RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT
VOI MOT PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');
        ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN <> V_NGAYROI AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    1 THEN
            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYROI =
    NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN
            WHERE MAPHIEU = MAPHIEU_IN;
        ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN <> V_NGAYROI AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    0 THEN
            RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT
VOI MOT PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');
        ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN = V_NGAYROI AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    1 THEN
            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, SONGUOIDIKEM =
    SONGUOIDIKEM_IN, MANV = MANV_IN, NGAYDEN = NGAYDEN_IN, NGAYROI =
    NGAYROI_IN
            WHERE MAPHIEU = MAPHIEU_IN;
        ELSIF NGAYDEN_IN = V_NGAYDEN AND NGAYROI_IN = V_NGAYROI AND
    FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN, MAPHIEU_IN) =
    0 THEN
            RAISE_APPLICATION_ERROR(-20014,'KHONG THE CAP NHAT, XUNG DOT
VOI MOT PHIEU DAT TRUOC/ PHIEU NHAN PHONG KHAC');
        END IF;
    END;
```

## Đồ án HQT \_ Quản Lý Khách Sạn

```
END IF;

IF TINHTRANG_IN = 'ĐÃ NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' AND
V_NGAYDEN >= SYSDATE THEN
BEGIN
    IF FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN,
MAPHIEU_IN) = 1 THEN
        BEGIN
            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYDEN =
NGAYDEN_IN, NGAYROI = NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV =
MANV_IN, TINHTRANG = TINHTRANG_IN
                WHERE MAPHIEU = MAPHIEU_IN;
            INSERT INTO PHIEUNHANPHONG(MAPHIEUNHAN, MAKH, MAPHONG, MANV,
SONGUOIDIKEM, NGAYDEN, NGAYROI)
                VALUES (SEQ_PHIEUNHAN.NEXTVAL, MAKH_IN, MAPHONG_IN, MANV_IN,
SONGUOIDIKEM_IN, NGAYDEN_IN, NGAYROI_IN);
        END;
    ELSE
        RAISE_APPLICATION_ERROR(-20015, 'KHÔNG THỂ NHẬN PHÒNG TRƯỚC NGÀY
ĐAT, VI PHÒNG HIỆN ĐANG Ở TRẠNG THÁI BẢN');
    END IF;
END;

ELSIF TINHTRANG_IN = 'ĐÃ NHẬN' AND V_TINHTRANG = 'CHƯA NHẬN' AND
NGAYROI_IN > V_NGAYROI AND V_NGAYDEN = SYSDATE THEN
BEGIN
    IF FUNC_CHECK_PHONGTRONG(MAPHONG_IN, NGAYDEN_IN, NGAYROI_IN,
MAPHIEU_IN) = 1 THEN
        BEGIN
            UPDATE PHIEUDATTRUOC SET MAPHONG = MAPHONG_IN, NGAYROI =
NGAYROI_IN, SONGUOIDIKEM = SONGUOIDIKEM_IN, MANV = MANV_IN, TINHTRANG =
TINHTRANG_IN
                WHERE MAPHIEU = MAPHIEU_IN;
            INSERT INTO PHIEUNHANPHONG(MAPHIEUNHAN, MAKH, MAPHONG, MANV,
SONGUOIDIKEM, NGAYDEN, NGAYROI)
                VALUES (SEQ_PHIEUNHAN.NEXTVAL, MAKH_IN, MAPHONG_IN, MANV_IN,
SONGUOIDIKEM_IN, NGAYDEN_IN, NGAYROI_IN);
        END;
    ELSE
        RAISE_APPLICATION_ERROR(-20016, 'KHÔNG THỂ GIÀ HẠN NGÀY RƠI, VI
CO PHIEU DAT TRUOC CHO PHÒNG');
    END IF;
END;

SLEEP(10);
COMMIT;
END;
```

### iii. Trigger TRIG\_UPDATE\_TINHTRANGPHONG

```
CREATE OR REPLACE TRIGGER TRIG_UPDATE_TINHTRANGPHONG
AFTER UPDATE ON PHONG
FOR EACH ROW
```

## Đồ án HQT \_ Quản Lý Khách Sạn

```
DECLARE
V_PHIEUDATTRUOC PHIEUDATTRUOC%ROWTYPE;
CURSOR C IS SELECT * FROM PHIEUDATTRUOC WHERE MAPHONG = :NEW.MAPHONG
AND NGAYDEN <= SYSDATE + INTERVAL '4' DAY
    AND :NEW.TINHTRANG = 'SỬA CHỮA';
PHONG_TRONG INT := 0;
V_MALP INT;
pragma autonomous_transaction;
BEGIN
    OPEN C;
    LOOP
        FETCH C INTO V_PHIEUDATTRUOC;
        EXIT WHEN C%NOTFOUND;
        SELECT MALOAIPHONG INTO V_MALP FROM PHONG WHERE MAPHONG =
V_PHIEUDATTRUOC.MAPHONG;
        PHONG_TRONG := FUNC_RANDOM_PHONG(V_MALP,V_PHIEUDATTRUOC.NGAYDEN,
V_PHIEUDATTRUOC.NGAYROI);
        PRO_UPDATE_PHIEUDATTRUOC(V_PHIEUDATTRUOC.MAPHIEU,PHONG_TRONG,
V_PHIEUDATTRUOC.MAKH,
        1,
V_PHIEUDATTRUOC.NGAYDEN,V_PHIEUDATTRUOC.NGAYROI,V_PHIEUDATTRUOC.SONGUOI
DIKEM, V_PHIEUDATTRUOC.TINHTRANG);
        dbms_output.put_line(V_PHIEUDATTRUOC.maphieu);
    END LOOP;
    CLOSE C;
    COMMIT;
END;
```

### iv. FUNC\_RANDOM\_PHONG

```
create or replace FUNCTION FUNC_RANDOM_PHONG(MALOAIPHONG_IN IN INT,
                                              NGAYDEN_IN IN DATE,
                                              NGAYROI_IN IN DATE
)
RETURN NUMBER
IS
    PHONG_TRONG INT:= 0;
BEGIN
    SELECT DISTINCT MAPHONG INTO PHONG_TRONG FROM
        (SELECT MALOAIPHONG, MAPHONG FROM PHONG WHERE TINHTRANG <> 'SỬA
CHỮA'
        MINUS
        (SELECT DISTINCT P.MALOAIPHONG, PDT.MAPHONG FROM PHIEUDATTRUOC PDT
INNER JOIN PHONG P
        ON PDT.MAPHONG = P.MAPHONG WHERE (NGAYDEN_IN <= NGAYDEN AND
NGAYDEN <= NGAYROI_IN)
                                            OR (NGAYDEN_IN <= NGAYDEN AND
NGAYROI <= NGAYROI_IN)
                                            OR (NGAYDEN_IN <= NGAYROI AND
NGAYROI <= NGAYROI_IN)
        UNION
```

```

SELECT DISTINCT P1.MALOAIIPHONG, PNP.MAPHONG FROM PHIEUNHANPHONG
PNP INNER JOIN PHONG P1
    ON P1.MAPHONG = PNP.MAPHONG WHERE (NGAYDEN_IN <= NGAYDEN AND
NGAYDEN <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYDEN AND
NGAYROI <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYROI AND
NGAYROI <= NGAYROI_IN)
) ) A
WHERE MALOAIIPHONG = MALOAIIPHONG_IN
ORDER BY DBMS_RANDOM.VALUE
FETCH FIRST 1 ROW ONLY;
RETURN PHONG_TRONG;
END;

```

v. Bảng mô tả Lost Update

Session 1	Session 2	Explanation
EXEC PRO_UPDATE_PHIEUDATTRUOC (87,43,20,1,'09-JULY-2020','11- JULY-2020',1,'CHUA NHAN');	UPDATE PHONG SET TINHTRANG = 'SUA CHUA' WHERE MAPHONG = 43;	Trong khi sesion 1 thực hiện cập nhật cho phiếu đặt trước thì session 2 thực hiện cập nhật tình trạng cho mã phòng 43

c. Nguyên nhân và giải pháp:

- i. Vấn đề xảy ra: Dữ liệu mà nhân viên thực hiện cho phiếu đặt trước bị ghi đè bởi dữ liệu mà người quản lý cập nhật sau. Dẫn tới việc sai sót dữ liệu.
- ii. Nguyên nhân: Vì thiết lập mức độ read committed nên T2 không nhận biết được T1 đã cập nhật ngày đến của phiếu đặt trước. T2 lấy dữ liệu cũ và tiến hành cập nhật lại cho phiếu đặt trước.
- iii. Giải pháp: Sử dụng câu lệnh “SET TRANSACTION ISOLATION LEVEL  
SERIALIZABLE” thay cho câu lệnh “SET TRANSACTION ISOLATION LEVEL  
READ COMMITTED” ở Session 2 khi cập nhật phiếu đặt trước, câu lệnh sẽ đợi T1 thực hiện xong thì T2 mới lấy dữ liệu đó để cập nhật.

**5.4.2 Non - repeatable read:**

- a. Mô tả tình huống: Trong khi tiến hành thanh toán hóa đơn cho khách hàng, nhân viên tiếp tân thực hiện truy vấn giá của món ăn mà khách hàng đã sử dụng để tính thành tiền cho khách hàng, thì người quản lý thực hiện cập nhật giá món ăn. Khi kiểm tra hóa đơn thanh toán trước khi giao cho khách hàng thì thấy dữ liệu có sự thay đổi dẫn đến sai sót.
- b. Mã PL/SQL:

```
CREATE OR REPLACE PROCEDURE PROD_BK_MONAN_THANHTIEN
```

```

(MA_HD IN BANGKE_MONAN.MAHD%TYPE, MA_MON IN
BANGKE_MONAN.MAMON%TYPE, NGAY_SD IN BANGKE_MONAN.NGAY%TYPE)
AS
GIA MONAN.GIAMON%TYPE;
BEGIN
SET TRANSACTION ISOLATION LEVEL read committed;
SLEEP(10);
SELECT GIAMON INTO GIA
FROM MONAN
WHERE MAMON = MA_MON;

UPDATE BANGKE_MONAN
SET THANHTIEN = GIA * SOLUONG
WHERE MAHD = MA_HD AND NGAY = NGAY_SD AND MAMON = MA_MON;
COMMIT;
END;

```

- c. Bảng mô tả Non-repeatable read:

Session 1	Session 2	Explanation
SELECT GIAMON FROM MONAN WHERE MAMON = 1  GIAMON ----- 100000	No action.	Session 1 thực hiện truy vấn giá tiền của món ăn với mã món = 1.
DECLARE IN_THANHTIEN INT; BEGIN PROD_BK_MONAN_THANHTIEN (144,1,'07-JULY-20',2); SELECT THANHTIEN INTO IN_THANHTIEN FROM BANGKE_MONAN WHERE MAHD = 144 AND MAMON = 1 AND NGAY = '07-JULY-20'; DBMS_OUTPUT.PUT_LINE('THANH TIEN: '    IN_THANHTIEN); COMMIT; END;	UPDATE MONAN SET GIAMON = 120000 WHERE MAMON = 1;	Session 1 tính thành tiền cho bảng kê món ăn vừa thì Session 2 thực hiện cập nhật giá món của món ăn có MAMON = 4 từ 100000 → 120000.
No action.	COMMIT;	Transaction 2 thực hiện xong và kết thúc.
COMMIT; Kq: THANH TIEN:240000	No action	Transaction 1 thực hiện xong in ra kết quả thành tiền là 240000 và kết thúc.

		Ta nhận thấy kết quả không chính xác.
--	--	---------------------------------------

d. Nguyên nhân

- i. Vấn đề xảy ra: T1 thực hiện truy vấn số giá món của món ăn có mã món = 1, ngay lúc đó thì T2 thực hiện cập nhật giá món ăn cho mã món = 1, dẫn đến T1 lấy giá món mà T2 cập nhật để tính tiền cho bảng kê món ăn đã kê khai trước khi giá món được cập nhật, dẫn đến thành tiền của bảng kê bị sai lệch so với thực tế.
- ii. Nguyên nhân: T1 thiết lập mức độ read committed nên mỗi khi SELECT trong cùng 1 thao tác dữ liệu, nó sẽ đọc lại dữ liệu từ cơ sở dữ liệu (cơ sở dữ liệu lúc này có thể bị thay đổi) mặc dù những câu lệnh này đọc trên những đơn vị dữ liệu giống nhau.
- iii. Giải pháp: Sử dụng câu lệnh “SET TRANSACTION ISOLATION LEVEL SERIALIZABLE” thay cho câu lệnh “SET TRANSACTION ISOLATION LEVEL READ COMMITTED”.
- iv. Kết quả: Thành tiền của bảng kê món ăn sẽ được tính với giá món ban đầu mà không phải là giá món được cập nhật bởi T2.

#### 5.4.3 Phantom read

- a. Mô tả tình huống: Một khách hàng gọi điện đến khách sạn để đặt phòng trước, nhân viên tiếp tân thực hiện xem danh sách phòng trống với mã loại phòng = 4(Suite), vào lúc đó một nhân viên khác thêm phiếu nhận phòng của một khách hàng cũng có mã loại phòng = 4 có ngày đến và ngày rời ảnh hưởng tới ngày đến và ngày rời với khách hàng 1 muốn đặt. Sau khi nhân viên 1 và khách hàng 1 thảo luận đi đến thống nhất mã phòng để đặt trước, thì nhân viên 1 tiến hành tạo phiếu đặt trước và thực hiện truy vấn danh sách phòng trống ứng với mã loại phòng = 4 thì thấy số lượng phòng trống thay đổi.

b. Bảng mã PL/SQL

```

CREATE OR REPLACE PROCEDURE PRO_DANHSACH_PHONGTRONG(NGAYDEN_IN IN
DATE,
                                              NGAYROI_IN IN
DATE,
                                              MALOAIIPHONG_IN IN INT,
                                              CONTRO OUT
SYS_REFCURSOR)
IS
BEGIN
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
    --SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    SLEEP(10);
    OPEN CONTRO FOR SELECT DISTINCT MAPHONG FROM
        (SELECT MALOAIIPHONG, MAPHONG FROM PHONG WHERE TINHTRANG <>
        'SỬA CHỮA'
        MINUS
        (SELECT DISTINCT P.MALOAIIPHONG, PDT.MAPHONG FROM
        PHIEUDATTRUOC PDT INNER JOIN PHONG P
        WHERE P.TINHTRANG <> 'SỬA CHỮA' AND P.MALOAIIPHONG = PDT.MALOAIIPHONG));
END;
  
```

```

    ON PDT.MAPHONG = P.MAPHONG WHERE (NGAYDEN_IN <= NGAYDEN AND
    NGAYDEN <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYDEN AND
    NGAYROI <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYROI AND
    NGAYROI <= NGAYROI_IN)
        UNION
        SELECT DISTINCT P1.MALOAIPHONG, PNP.MAPHONG FROM
PHIEUNHANPHONG PNP INNER JOIN PHONG P1
        ON P1.MAPHONG = PNP.MAPHONG WHERE (NGAYDEN_IN <= NGAYDEN
AND NGAYDEN <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYDEN AND
    NGAYROI <= NGAYROI_IN)
                                OR (NGAYDEN_IN <= NGAYROI AND
    NGAYROI <= NGAYROI_IN)
    ))A
    WHERE MALOAIPHONG = MALOAIPHONG_IN;
    COMMIT;
END;

```

- Bảng mô tả Phantom Read

Session 1	Session 2	Explanation
<pre> DECLARE PHONGTRONG INT; C SYS_REFCURSOR; BEGIN PRO_DANHSACH_PHON GTRONG('08-JUL- 2020','10-JUL-2020',4,C); LOOP     FETCH C INTO PHONGTRONG;     EXIT WHEN C%NOTFOUND;  DBMS_OUTPUT.PUT_LIN E('PHONG TRONG: '    PHONGTRONG); END LOOP; CLOSE C; COMMIT; END; </pre>	No action.	<p>Session 1 thực hiện truy xuất danh sách phòng trống với ngày đến là '08/07/2020', ngày rời là '10/07/2020' với mã loại phòng là 4(Suite) để tiến hành thông báo số phòng trống cho khách hàng. Kết quả được in ra:</p> <p>PHONGTRONG:47 PHONGTRONG: 49</p>
No action.	No action	

<pre> DECLARE PHONGTRONG INT; C SYS_REFCURSOR; BEGIN PRO_DANHSACH_PHON GTRONG('08-JUL- 2020','10-JUL-2020',4,C); LOOP     FETCH C INTO PHONGTRONG;     EXIT WHEN C%NOTFOUND;  DBMS_OUTPUT.PUT_LIN E('PHONG TRONG: '    PHONGTRONG); END LOOP; CLOSE C; COMMIT; END; </pre>	<pre> INSERT INTO PHIEUNHANPHONG(MAPHIEU, MAPHONG, MAKH, MANV, NGAYDEN, NGAYROI, SONGUOIDIKEM) VALUES(SEQ_PHIEUDAT.NEXT VAL,49,68,2,'07-JUL-2020','08- JUL-2020', 0); </pre>	<p>Trong khi thực hiện đặt phòng cho khách hàng 1, nhân viên 1 chờ load danh sách phòng trống lên để chọn, thì nhân viên 2 tiến hành thêm phiếu nhận phòng cho khách hàng 2 và kết thúc.</p>
<p>PHONG TRONG: 47</p>	<p>COMMIT;</p>	<p>Session 1 thực hiện câu lệnh và hiện danh sách phòng còn trống hiện tại thì nhân viên 1 nhận thấy số lượng phòng trống đã bị thay đổi so với lúc trước.</p>

c. Nguyên nhân và giải pháp

- i. Vấn đề xảy ra: Khi procedure của transaction 1 thực hiện câu lệnh truy vấn danh sách phòng trống ứng với mã loại phòng mà khách hàng chọn thì thấy số lượng phòng trống là 2. Sau khi thảo luận và thực hiện việc tạo một phiếu đặt trước cho khách hàng. Trong lúc chờ dữ liệu danh sách phòng trống được load lên combobox để chọn thì transaction 2 thực hiện câu lệnh insert một phiếu nhận phòng cho một khách hàng khác với mã loại phòng giống với mã loại phòng mà khách hàng 1 đã chọn. Sau khi hoàn tất việc load dữ liệu lên

- combobox thì nhân viên nhận thấy phòng 49 không có trong dữ liệu như lúc nãy đã truy vấn (Giảm số lượng phòng trống từ 2 xuống 1).
- ii. Nguyên nhân: T1 thiết lập mức độ read committed nên lần đọc dữ liệu thứ 2 để load dữ liệu lên combobox sẽ loại đi phòng trống có mã phòng là 49 vì phòng 49 đã được nhận bởi 1 khách hàng khác.
  - iii. Giải pháp: Sử dụng câu lệnh “SET TRANSACTION ISOLATION LEVEL SERIALIZABLE” thay cho câu lệnh “SET TRANSACTION ISOLATION LEVEL READ COMMITTED”.
  - iv. Kết quả: Lần đọc thứ 2 lấy dữ liệu lên combobox danh sách phòng còn trống vẫn sẽ hiện phòng với mã phòng là 49

#### 5.4.4 Deadlock

##### a. Mô tả tình huống:

Nhân viên 1 thực hiện truy vấn thông tin các phòng và yêu cầu khóa trên bảng PHONG ngăn không cho transaction khác yêu cầu khóa trên bảng. Trong khi đó thì người quản lý thực hiện truy vấn danh sách phiếu nhận phòng và yêu cầu khóa trên bảng PHIEUNHANPHONG ngăn cho các transaction yêu cầu khóa. Sau khi thực hiện truy vấn thông tin phòng thì nhân viên thực hiện gia hạn ngày cho một phiếu nhận phòng, và người quản lý cũng yêu cầu khóa để cập nhật lại tình trạng cho 1 phòng. Dẫn đến tình trạng deadlock.

##### b. Bảng mã PL/SQL:

Session 1	Session 2	Explanation
<pre>SELECT P.MAPHONG, TINHTRANG, TENLOAIPHONG FROM PHONG P INNER JOIN LOAIPHONG LP ON P.MALOAIPHONG = LP.MALOAIPHONG FOR UPDATE;</pre>	No action.	Session 1 thực hiện truy vấn thông tin các phòng và yêu cầu khóa trên đơn vị dữ liệu bảng PHONG.
No action.	<pre>SELECT * FROM PHIEUNHANPHON G FOR UPDATE;</pre>	Session 2 truy vấn danh sách phiếu nhận phòng và yêu cầu khóa trên đơn vị dữ liệu bảng PHIEUNHANPHONG.
No action.	<pre>UPDATE PHONG SET TINHTRANG = 'SUA CHUA' WHERE MAPHONG = 49;</pre>	Session 2 yêu cầu khóa trên đơn vị dữ liệu MAPHONG = 49 và thực hiện cập nhật tình trạng của phòng.
<pre>EXEC PRO_UPDATE_PHIEUNHANPHO NG</pre>	No action.	Session 1 thực hiện yêu cầu khóa trên đơn vị dữ liệu có

(92,68,49,2,0,'07-JUL-2020','09-JULY-2020');		MAPHIEUNHAN = 92 và thực hiện gia hạn ngày rời cho phiếu nhận phòng.
--	--	--

c. Nguyên nhân và giải pháp

- i. Vấn đề xảy ra: Session 1 chờ Session 2 giải phóng khóa trên đơn vị dữ liệu PHIEUNHANPHONG và Session 2 chờ Session 1 giải phóng khóa trên đơn vị dữ liệu PHONG. Kết quả dẫn đến hệ quản trị báo lỗi và hủy T3
- ii. Nguyên nhân: T1 thực hiện truy vấn thông tin phòng và yêu cầu khóa trên đơn vị dữ liệu PHONG, T2 thực hiện truy vấn thông tin phiếu nhận phòng và yêu cầu khóa trên đơn vị dữ liệu PHIEUNHANPHONG. T3 thực hiện yêu cầu khóa trên đơn vị dữ liệu có MAPHONG = 49 mà T1 đang giữ. T4 thực hiện yêu cầu khóa trên đơn vị dữ liệu có MAPHIEUNHAN = 92 mà T2 đang giữ khóa. Session 1 chờ session 2 giải phóng khóa trên đơn vị dữ liệu PHIEUNHANPHONG. Session 2 chờ Session 1 giải phóng khóa trên đơn vị dữ liệu PHONG. Dẫn đến tình trạng Deadlock.
- iii. Giải pháp: Hệ quản trị sẽ tự động rollback transaction và trả về phản hồi cho transaction bị rollback. Thông thường thì hệ quản trị sẽ không khóa đơn vị dữ liệu khi đọc dữ liệu. Có thể sử dụng multitable deadlocks để tránh các transaction truy cập cùng table lock.

## Chương 6: THIẾT KẾ GIAO DIỆN

### 6.1 Danh sách các màn hình

#### 6.1.1 Đăng nhập

STT	Màn hình	Chức năng
1	Đăng nhập	Đăng nh

#### 6.1.2 Manager

STT	Màn hình	Chức năng
1	Quản lý khách hàng	Hiển thị những thông tin đã có trong hệ thống
2	Quản lý phòng	Hiển thị những thông tin đã có trong hệ thống
3	Quản lý nhân viên	Hiển thị những thông tin đã có trong hệ thống
4	Quản lý dịch vụ	Hiển thị những thông tin đã có trong hệ thống
5	Quản lý món ăn	Hiển thị những thông tin đã có trong hệ thống
6	Quản lý CSVC	Hiển thị những thông tin đã có trong hệ thống
7	Quản lý hóa đơn	Hiển thị những thông tin

Đồ án HQT \_ Quản Lý Khách Sạn

		đã có trong hệ thống
8	Thống kê doanh thu	Hiển thị những thông tin đã có trong hệ thống
9	Thêm khách hàng	Nhập thông tin để thêm khách hàng mới
10	Cập nhật khách hàng	Cập nhật thông tin khách hàng
11	Phiếu đặt trước	Quản lý phiếu đặt trước
12	Thêm phiếu đặt trước	Nhập thông tin để thêm phiếu đặt trước
13	Cập nhật phiếu đặt trước	Cập nhật thông tin phiếu đặt trước
14	Phiếu nhận phòng	Quản lý phiếu nhận phòng
15	Thêm phiếu nhận phòng	Nhập thông tin để thêm phiếu nhận phòng
16	Cập nhật phiếu nhận phòng	Cập nhật thông tin phiếu nhận phòng
17	Thêm phòng	Nhập thông tin để thêm phòng mới
18	Cập nhật phòng	Cập nhật thông tin phòng
19	Loại phòng	Tra cứu loại phòng
20	Thêm nhân viên	Nhập thông tin nhân viên để thêm mới
21	Thông tin nhân viên	Hiển thị thông tin nhân viên
22	Cập nhật nhân viên	Cập nhật thông tin nhân viên
23	Thêm dịch vụ	Nhập thông tin dịch vụ để thêm mới
24	Cập nhật dịch vụ	Cập nhật thông tin dịch vụ
25	Thêm món ăn	Nhập thông tin dịch vụ để thêm mới
26	Cập nhật món ăn	
27	Thêm CSVC	Nhập thông tin dịch vụ để thêm mới
28	Cập nhật CSVC	
29	Thêm bảng kê dịch vụ	Nhập thông tin dịch vụ để thêm mới
30	Thêm bảng kê món ăn	Nhập thông tin món ăn để thêm mới
31	Thêm hóa đơn CSVC	Nhập thông tin CSVC để thêm mới
32	Xóa bảng kê dịch vụ	Xóa bảng kê
33	Xóa bảng kê món ăn	Xóa bảng kê
34	Thống kê doanh thu	Tính tổng doanh thu

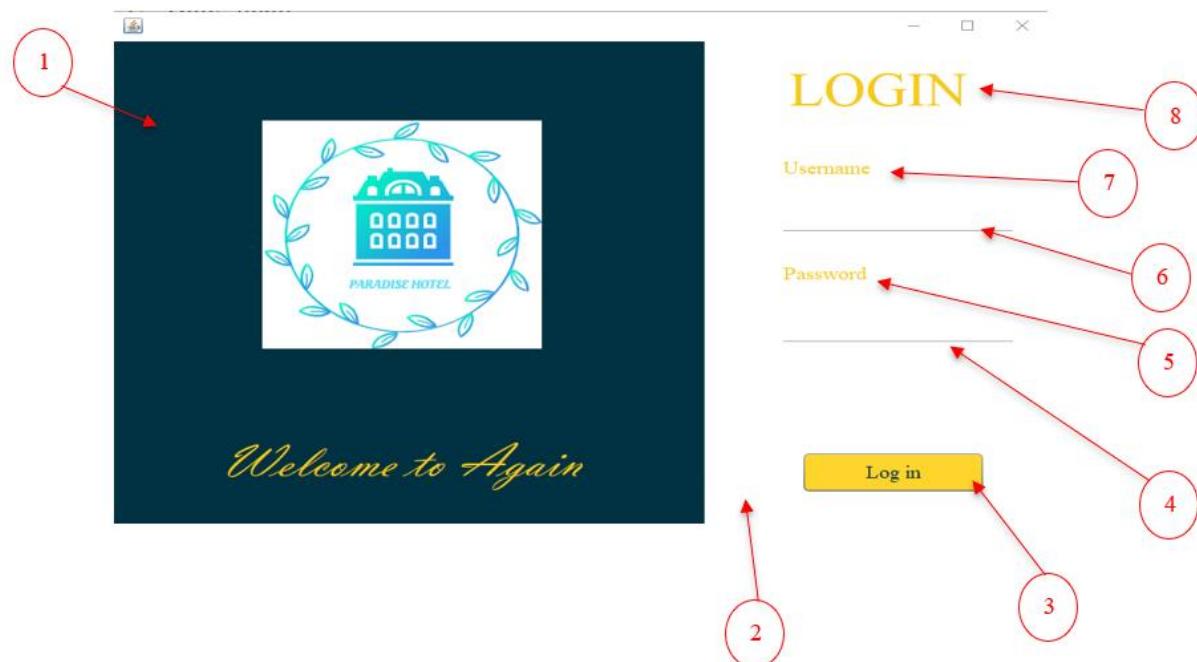
35	Thông kê số lượng đặt phòng theo loại phòng	Tính số lượng phòng theo từng loại
36	Top 3 món ăn được đặt nhiều nhất trong tháng	Thông kê 3 món ăn được đặt nhiều nhất trong 1 tháng

### 6.1.3 Receptionists

STT	Màn hình	Chức năng
1	Quản lý khách hàng	Hiển thị những thông tin đã có trong hệ thống
2	Danh sách phòng	Hiển thị danh sách phòng
4	Danh sách dịch vụ	Hiển thị danh sách dịch vụ
5	Danh sách món ăn	Hiển thị danh sách món ăn
6	Danh sách CSVC	Hiển thị danh sách CSVC
7	Hóa đơn thanh toán	Hiển thị những thông tin đã có trong hệ thống
8	Thông kê doanh thu	Hiển thị những thông tin đã có trong hệ thống
9	Thêm khách hàng	Nhập thông tin để thêm khách hàng mới
10	Cập nhật khách hàng	Cập nhật thông tin khách hàng
11	Phiếu đặt trước	Quản lý phiếu đặt trước
12	Thêm phiếu đặt trước	Nhập thông tin để thêm phiếu đặt trước
13	Cập nhật phiếu đặt trước	Cập nhật thông tin phiếu đặt trước
14	Phiếu nhận phòng	Quản lý phiếu nhận phòng
15	Thêm phiếu nhận phòng	Nhập thông tin để thêm phiếu nhận phòng
16	Cập nhật phiếu nhận phòng	Cập nhật thông tin phiếu nhận phòng
17	Loại phòng	Tra cứu loại phòng
18	Thêm bảng kê dịch vụ	Nhập thông tin dịch vụ để thêm mới
19	Thêm bảng kê món ăn	Nhập thông tin món ăn để thêm mới
20	Thêm hóa đơn CSVC	Nhập thông tin CSVC để thêm mới
21	Xóa bảng kê dịch vụ	Xóa bảng kê
22	Xóa bảng kê món ăn	Xóa bảng kê

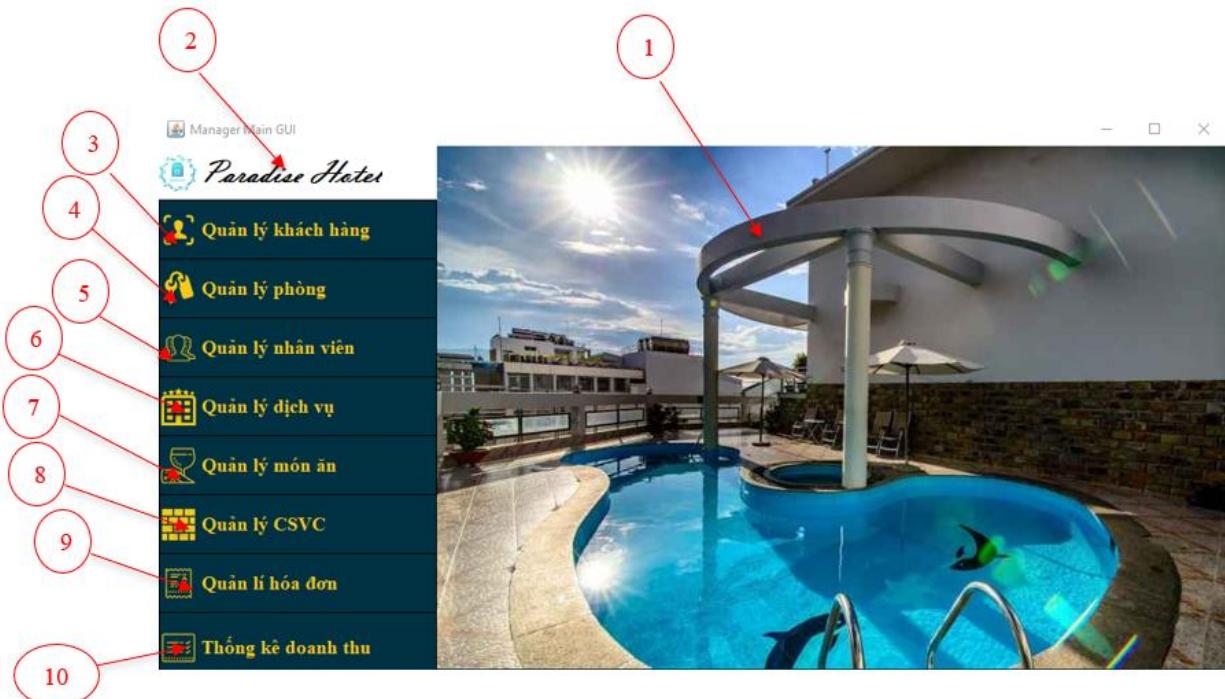
## 6.2 Mô hình các màn hình chính

### 6.2.1 Đăng nhập



STT	Tên	Kiểu	Chức năng
1	Khung hình	JPanel	Hiển thị thông tin
2	Khung đăng nhập	JPanel	Điền thông tin đăng nhập
3	Nút đăng nhập	JButton	Nhấn đăng nhập
4	Ô mật khẩu	JPasswordField	Điền mật khẩu
5	Nhãn mật khẩu	JLabel	Hiển thị thông tin
6	Ô tên tài khoản	Jtextfield	Điền tên đăng nhập
7	Nhãn tên đăng nhập	JLabel	Hiển thị thông tin
8	Nhãn login	JLabel	Hiển thị thông tin

### 6.2.2 Manager



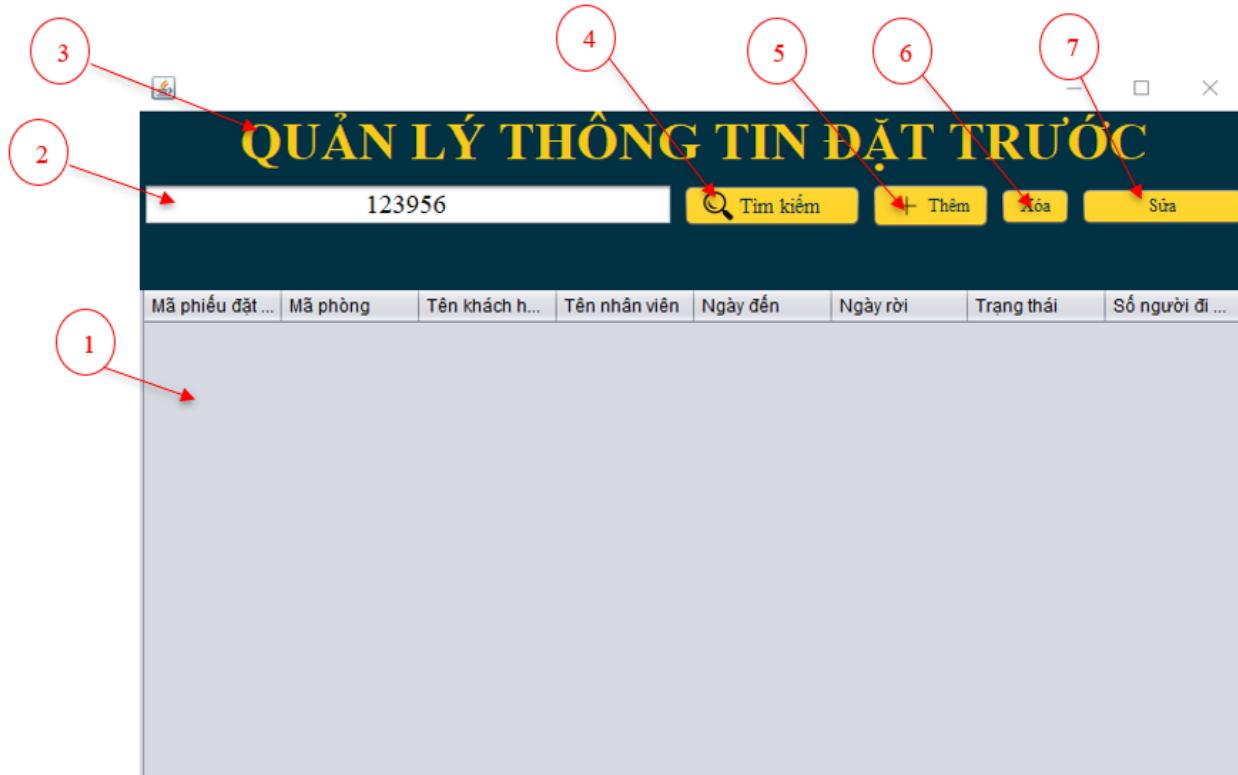
STT	Tên	Kiểu	Chức năng
1	Ảnh khách sạn	JLabel	Hiển thị thông tin
2	Logo và tên khách sạn	JLabel	Hiển thị thông tin
3	Quản lý khách hàng	JLabel	Nhấn đăng nhập
4	Quản lý phòng	JLabel	Điền mật khẩu
5	Quản lý nhân viên	JLabel	Hiển thị thông tin
6	Quản lý dịch vụ	JLabel	Điền tên đăng nhập
7	Quản lý món ăn	JLabel	Hiển thị thông tin
8	Quản lý CSVC	JLabel	Hiển thị thông tin
9	Quản lý hóa đơn	JLabel	Hiển thị thông tin
10	Thống kê doanh thu	JLabel	Hiển thị thông tin

### 6.2.3 Quản lý khách hàng



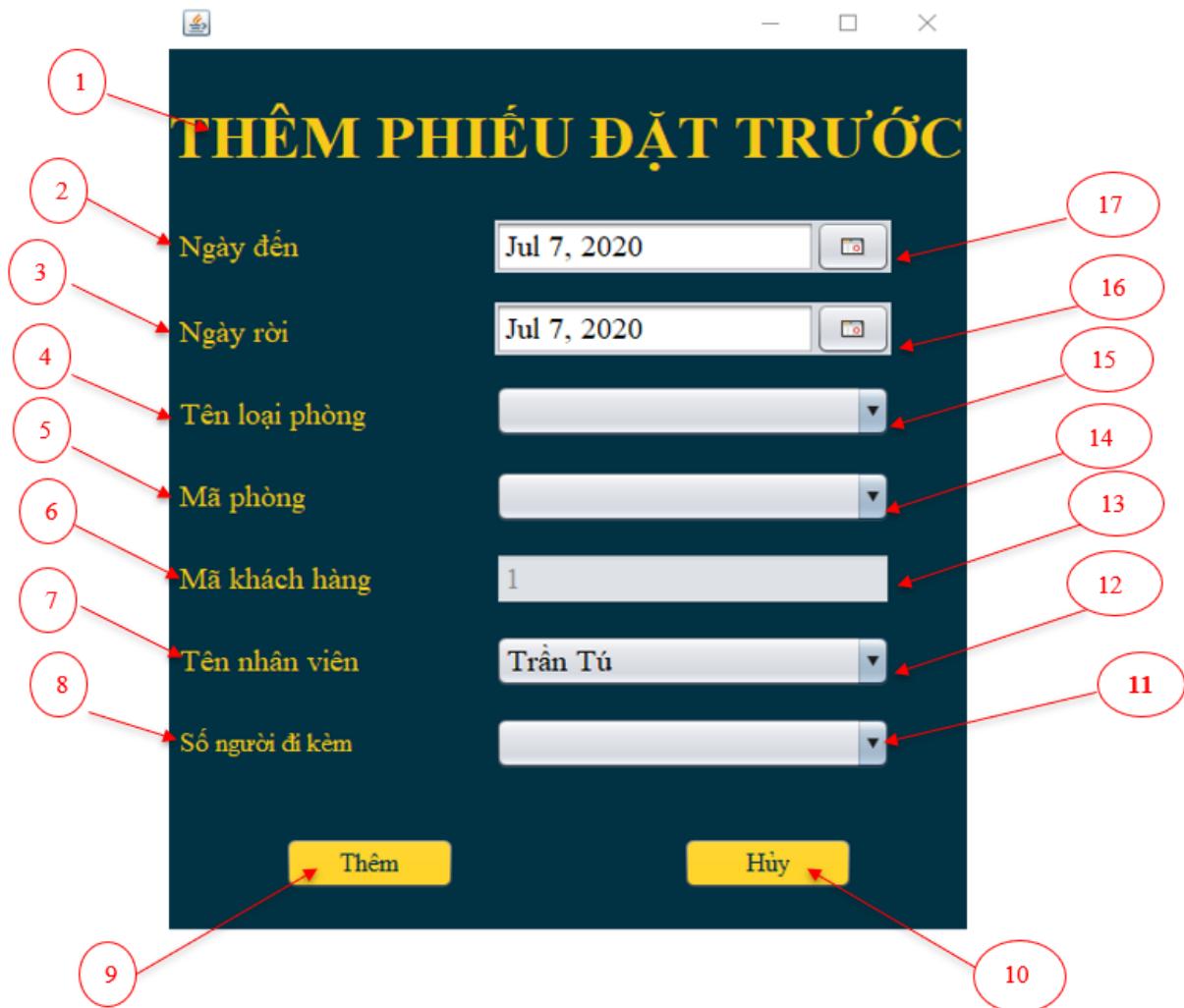
STT	Tên	Kiểu	Chức năng
1	Bảng thông tin khách hàng	JTable	Hiển thị thông tin khách hàng
2	Ô nhập chứng minh nhân dân	JTextfield	Nhập số chứng minh nhân dân
3	Nhân quản lý thông tin khách hàng	JLabel	Hiển thị thông tin
4	Nút thêm	JButton	Thực hiện chức năng thêm
5	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
6	Nút xóa	JButton	Thực hiện chức năng xóa
7	Nút sửa	JButton	Thực hiện chức năng sửa
8	Nút phiếu đặt trước	JButton	Hiển thị giao diện quản lý phiếu đặt trước
9	Nút phiếu nhận phòng	JButton	Hiển thị giao diện quản lý phiếu nhận phòng

#### 6.2.4 Quản lý phiếu đặt trước



STT	Tên	Kiểu	Chức năng
1	Bảng thông tin phiếu đặt trước	JTable	Hiển thị thông tin khách hàng
2	Ô nhập chứng minh nhân dân	JTextfield	Nhập số chứng minh nhân dân
3	Nhãn quản lý thông tin đặt trước	JLabel	Hiển thị thông tin
4	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
5	Nút thêm	JButton	Thực hiện chức năng thêm phiếu đặt
6	Nút xóa	JButton	Thực hiện chức năng xóa phiếu đặt
7	Nút sửa	JButton	Thực hiện chức năng sửa phiếu đặt

### 6.2.5 Thêm phiếu đặt trước



STT	Tên	Kiểu	Chức năng
1	Nhãn thêm phiếu đặt trước	JLabel	Hiển thị thông tin khách hàng
2	Nhãn ngày đến	JLabel	Hiển thị thông tin
3	Nhãn ngày rời	JLabel	Hiển thị thông tin
4	Nhãn tên loại phòng	JLabel	Hiển thị thông tin
5	Nhãn mã phòng	JLabel	Hiển thị thông tin
6	Nhãn mã khách hàng	JLabel	Hiển thị thông tin
7	Nhãn tên nhân viên	JLabel	Hiển thị thông tin
8	Nhãn số người đi kèm	JLabel	Hiển thị thông tin
9	Nút thêm	JButton	Thêm phiếu đặt
10	Nút hủy	Jbutton	Hủy thao tác
11	Khung số người đi kèm	Combobox	Hiển thị số lượng người đi kèm
12	Khung tên nhân viên	Combobox	Hiển thị tên nhân viên

13	Khung mã khách hàng	JTextField	Hiển thị mã khách hàng
14	Khung mã phòng	Combobox	Hiển thị mã phòng
15	Khung tên loại phòng	Combobox	Hiển thị loại phòng
16	Khung ngày rời	JDateChooser	Hiển thị ngày đến
17	Khung ngày đến	JDateChooser	Hiển thị ngày rời

#### 6.2.6 Cập nhật phiếu đặt trước

1	Mã phiếu đặt	21	21
2	Ngày đến	Jul 8, 2020	20
3	Ngày rời	Jul 10, 2020	19
4	Tên loại phòng	Superior	18
5	Mã phòng	29	17
6	Mã khách hàng	141	16
7	Tên nhân viên	Trần Tú	15
8	Tình trạng	<input checked="" type="checkbox"/> Đã nhận phòng	14
9	Số người đi kèm	1	13
10			
11	Cập nhật		
12	Hủy		

Đồ án HQT \_ Quản Lý Khách Sạn

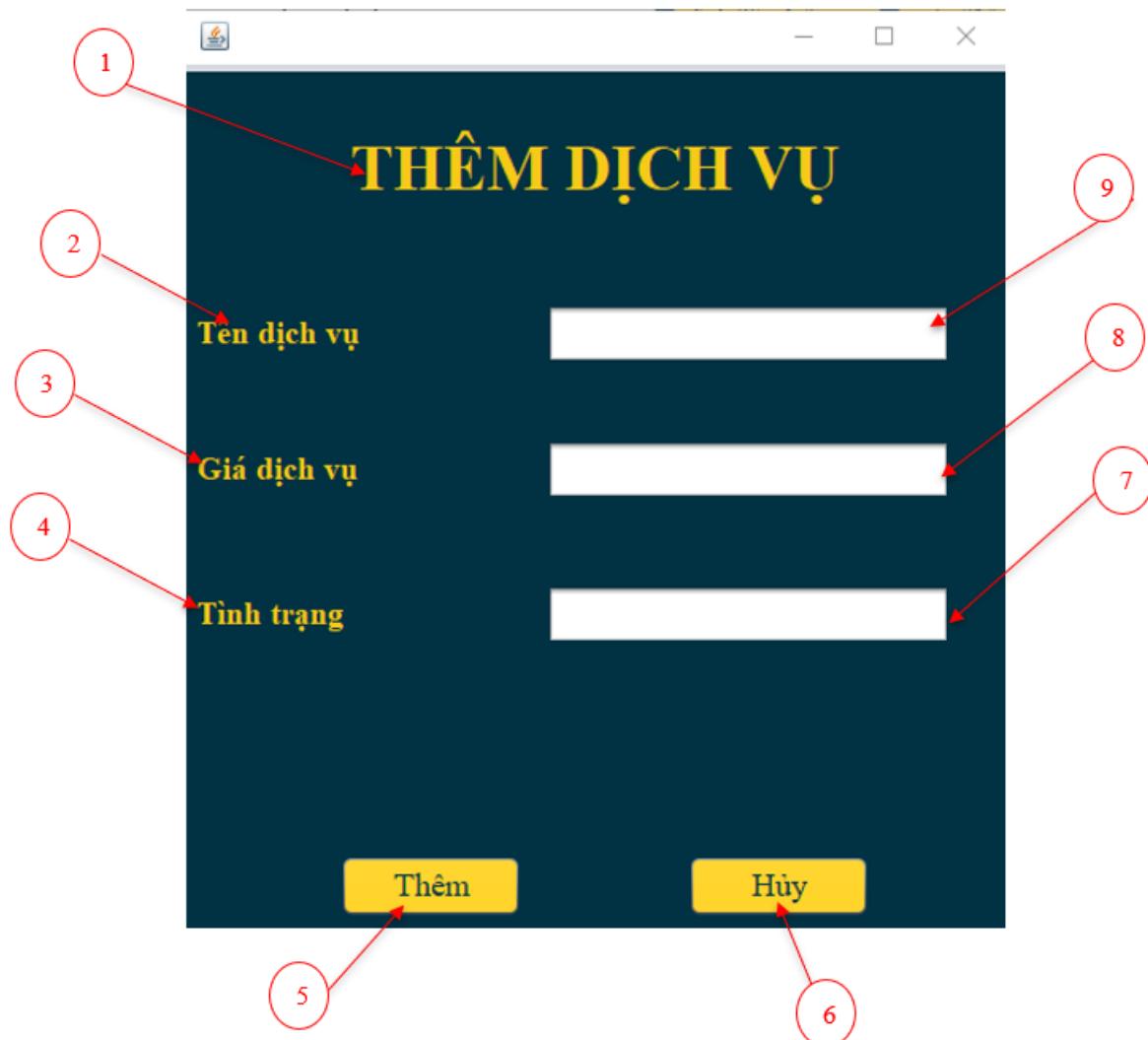
STT	Tên	Kiểu	Chức năng
1	Nhãn cập nhật phiếu đặt trước	JLabel	Hiển thị thông tin khách hàng
2	Nhãn mã phiếu đặt	JLabel	Hiển thị thông tin
3	Nhãn ngày đến	JLabel	Hiển thị thông tin
4	Nhãn ngày rời	JLabel	Hiển thị thông tin
5	Nhãn tên loại phòng	JLabel	Hiển thị thông tin
6	Nhãn mã phòng	JLabel	Hiển thị thông tin
7	Nhãn mã khách hàng	JLabel	Hiển thị thông tin
8	Nhãn tên nhân viên	JLabel	Hiển thị thông tin
9	Nhãn tình trạng	JLabel	Hiển thị thông tin
10	Nhãn số người đi kèm	JLabel	Hiển thị thông tin
11	Nút cập nhật	JButton	Cập nhật phiếu đặt
12	Nút hủy	Jbutton	Hủy thao tác
13	Khung số người đi kèm	Combobox	Hiển thị số lượng người đi kèm
14	Khung tình trạng	jCheckBox	Đánh dấu tình trạng “Đã nhận”
15	Khung tên nhân viên	Combobox	Hiển thị tên nhân viên
16	Khung mã khách hàng	JTextField	Hiển thị mã khách hàng
17	Khung mã phòng	Combobox	Hiển thị mã phòng
18	Khung tên loại phòng	Combobox	Hiển thị loại phòng
19	Khung ngày rời	JDateChooser	Hiển thị ngày đến
20	Khung ngày đến	JDateChooser	Hiển thị ngày rời
21	Khung mã phiếu đặt	JTextField	

### 6.2.7 Quản lý dịch vụ



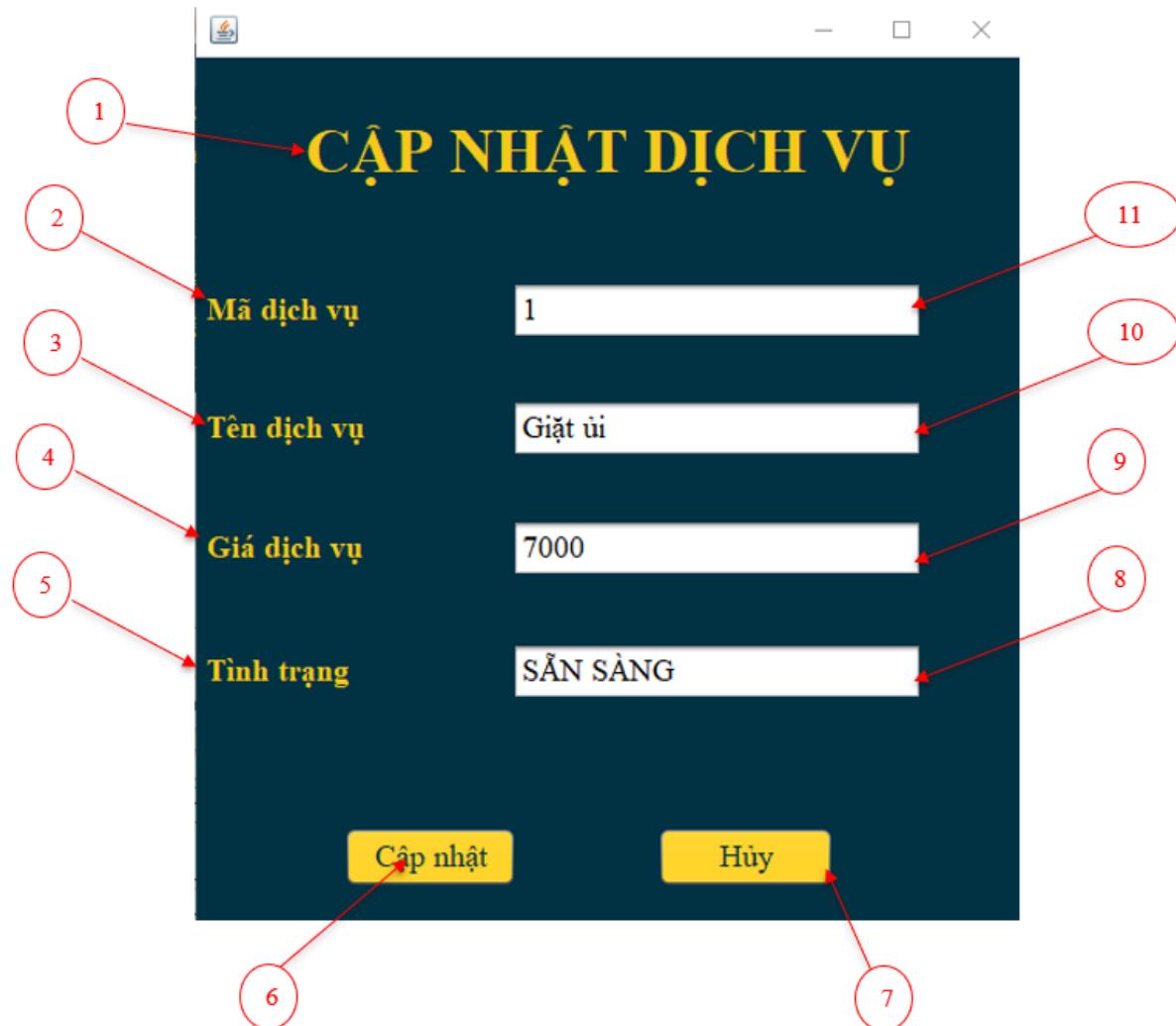
STT	Tên	Kiểu	Chức năng
1	Bảng thông tin dịch vụ	JTable	Hiển thị thông tin dịch vụ
2	Ô nhập dịch vụ	JTextfield	Nhập tên dịch vụ
3	Nhãn quản lí dịch vụ	JLabel	Hiển thị thông tin
4	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
5	Nút thêm	JButton	Thực hiện chức năng thêm
6	Nút xóa	JButton	Thực hiện chức năng xóa
7	Nút sửa	JButton	Thực hiện chức năng sửa

### 6.2.8 Thêm dịch vụ



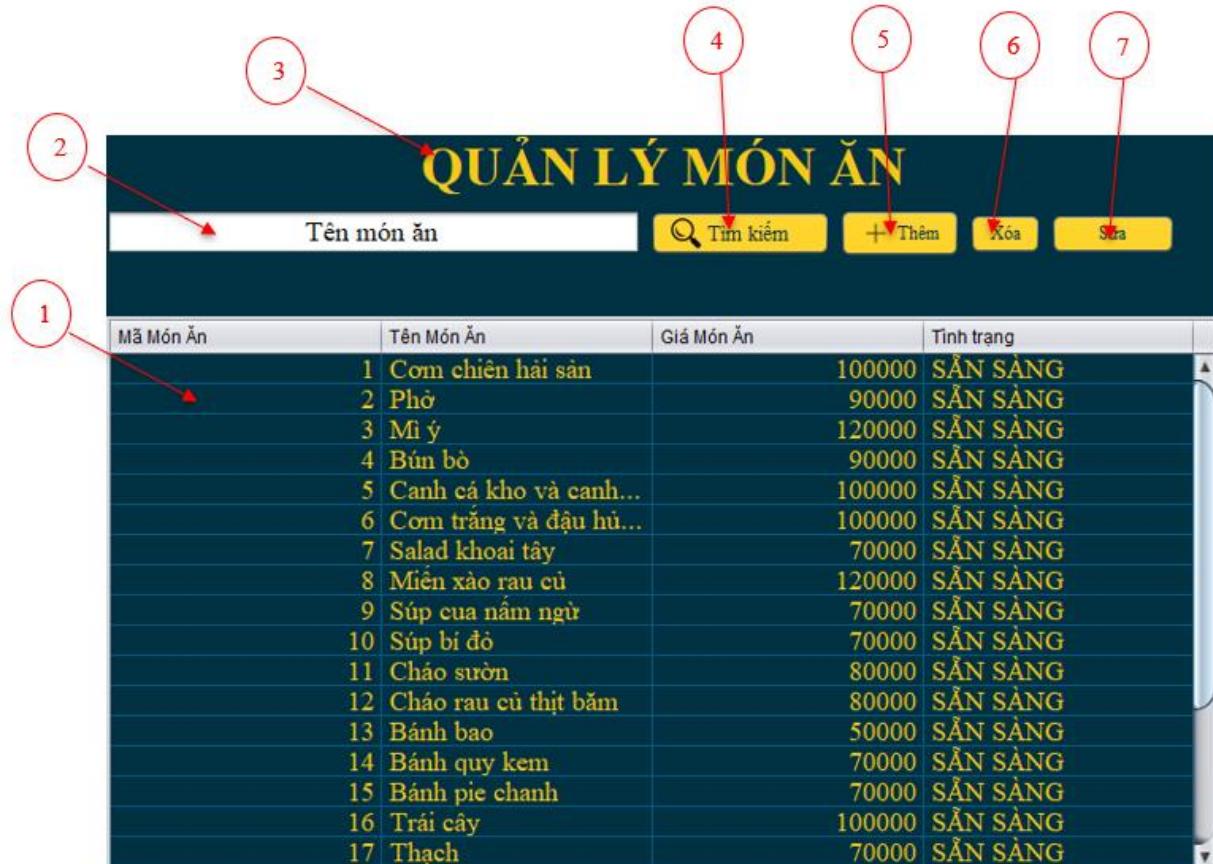
STT	Tên	Kiểu	Chức năng
1	Nhãn thêm dịch vụ	JLabel	Hiển thị thông tin dịch vụ
2	Nhãn tên dịch vụ	JLabel	Hiển thị thông tin
3	Nhãn giá dịch vụ	JLabel	Hiển thị thông tin
4	Nhãn tình trạng	JLabel	Hiển thị thông tin
5	Nút thêm	JButton	Thêm dịch vụ
6	Nút hủy	JButton	Hủy thao tác
7	Khung tình trạng	JTextfield	Nhận tình trạng dịch vụ
8	Khung giá dịch vụ	JTextfield	Nhập giá dịch vụ
9	Khung tên dịch vụ	JTextfield	Nhập tên dịch vụ

### 6.2.9 Cập nhật dịch vụ



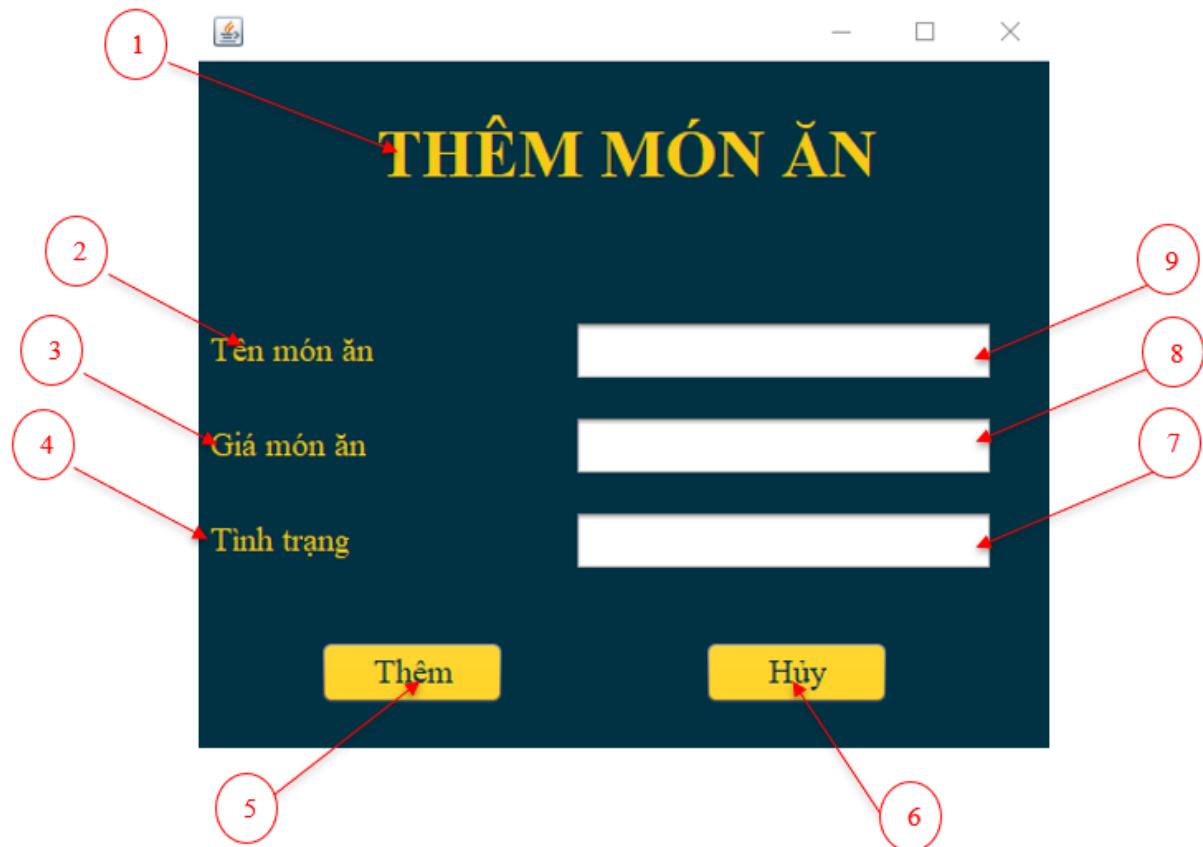
STT	Tên	Kiểu	Chức năng
1	Nhãn cập nhật dịch vụ	JLabel	Hiển thị thông tin
2	Nhãn mã dịch vụ	JLabel	Hiển thị thông tin
3	Nhãn tên dịch vụ	JLabel	Hiển thị thông tin
4	Nhãn giá dịch vụ	JLabel	Hiển thị thông tin
5	Nhãn tình trạng	JLabel	Hiển thị thông tin
6	Nút cập nhật	JButton	Cập nhật dịch vụ
7	Nút hủy	JButton	Hủy thao tác
8	Khung tình trạng	JTextfield	Nhận tình trạng dịch vụ
9	Khung giá dịch vụ	JTextfield	Nhập giá dịch vụ
10	Khung tên dịch vụ	JTextfield	Nhập tên dịch vụ
11	Khung mã dịch vụ	JTextfield	Hiển thị mã dịch vụ

### 6.2.10 Quản lý món ăn



STT	Tên	Kiểu	Chức năng
1	Bảng thông tin món ăn	JTable	Hiển thị thông tin món ăn
2	Ô nhập tên món ăn	JTextfield	Nhập tên món ăn
3	Nhãn quản lý thông tin món ăn	JLabel	Hiển thị thông tin
4	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
5	Nút thêm	JButton	Thực hiện chức năng thêm
6	Nút xóa	JButton	Thực hiện chức năng xóa
7	Nút sửa	JButton	Thực hiện chức năng sửa

### 6.2.11 Thêm món ăn



STT	Tên	Kiểu	Chức năng
1	Nhãn thêm món ăn	JLabel	Hiển thị thông tin
2	Nhãn tên món ăn	JLabel	Hiển thị thông tin
3	Nhãn giá món ăn	JLabel	Hiển thị thông tin
4	Nhãn tình trạng	JLabel	Hiển thị thông tin
5	Nút thêm	JButton	Thêm món ăn
6	Nút hủy	JButton	Hủy thao tác
7	Khung tình trạng	JTextField	Nhận tình trạng món ăn
8	Khung giá món ăn	JTextField	Nhập giá món ăn
9	Khung tên món ăn	JTextField	Nhập tên món ăn

### 6.2.12 Cập nhật món ăn



STT	Tên	Kiểu	Chức năng
1	Nhãn cập nhật món ăn	JLabel	Hiển thị thông tin
2	Nhãn mã món ăn	JLabel	Hiển thị thông tin
3	Nhãn tên món ăn	JLabel	Hiển thị thông tin
4	Nhãn giá món ăn	JLabel	Hiển thị thông tin
5	Nhãn tình trạng	JLabel	Hiển thị thông tin
6	Nút cập nhật	JButton	Cập nhật món ăn
7	Nút hủy	JButton	Hủy thao tác
8	Khung tình trạng	JTextfield	Nhận tình trạng món ăn
9	Khung giá món ăn	JTextfield	Nhập giá món ăn
10	Khung tên món ăn	JTextfield	Nhập tên món ăn
11	Khung mã món ăn	JTextfield	Hiển thị mã món ăn

### 6.2.13 Quản lý Cơ sở vật chất



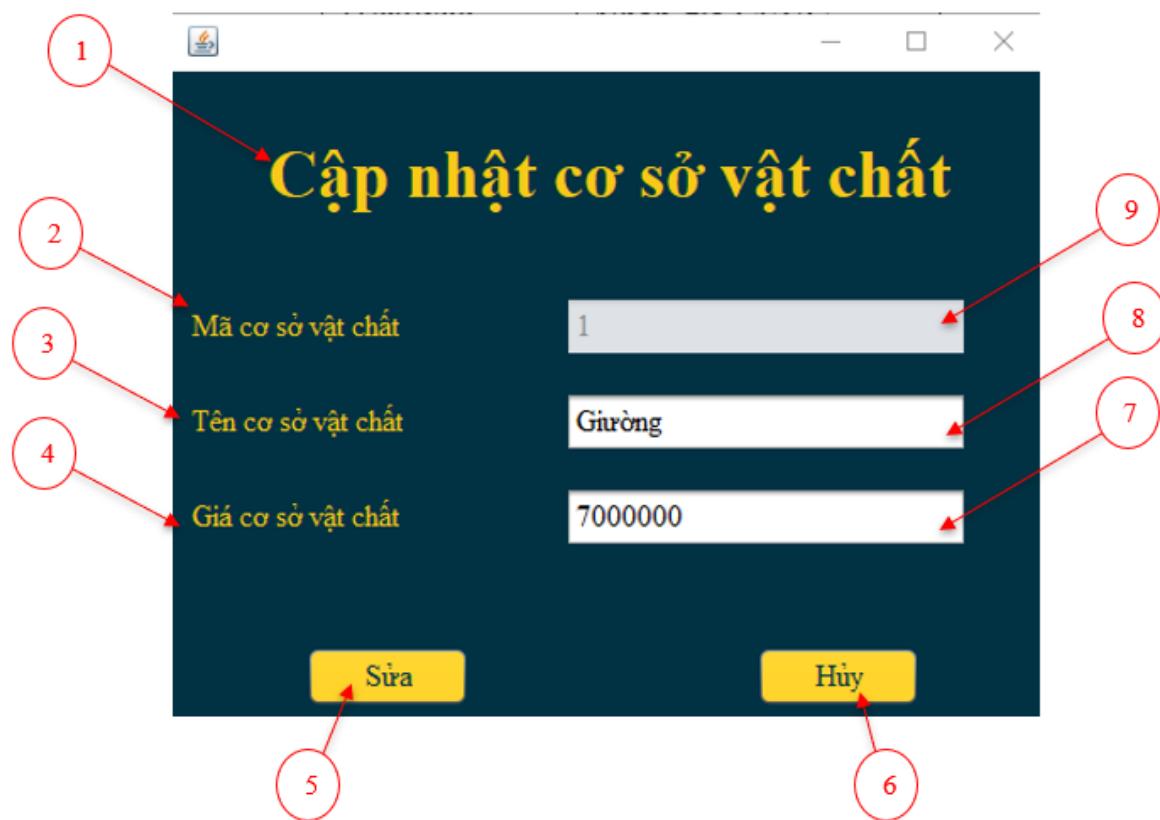
STT	Tên	Kiểu	Chức năng
1	Bảng thông tin CSVC	JTable	Hiển thị thông tin CSVC
2	Ô nhập tên CSVC	JTextfield	Nhập tên CSVC
3	Nhãn quản lí thông tin CSVC	JLabel	Hiển thị thông tin
4	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
5	Nút thêm	JButton	Thực hiện chức năng thêm
6	Nút xóa	JButton	Thực hiện chức năng xóa
7	Nút sửa	JButton	Thực hiện chức năng sửa

#### 6.2.14 Thêm Cơ sở vật chất



STT	Tên	Kiểu	Chức năng
1	Nhãn thêm CSVC	JLabel	Hiển thị thông tin
2	Nhãn tên CSVC	JLabel	Hiển thị thông tin
3	Nhãn giá món ăn	JLabel	Hiển thị thông tin
4	Nút thêm	JButton	Thêm CSVC
5	Nút hủy	JButton	Hủy thao tác
6	Khung giá CSVC	JTextfield	Nhập giá CSVC
7	Khung tên CSVC	JTextfield	Nhập tên CSVC

### 6.2.15 Cập nhật Cơ sở vật chất



STT	Tên	Kiểu	Chức năng
1	Nhãn cập nhật CSVC	JLabel	Hiển thị thông tin
2	Nhãn mã CSVC	JLabel	Hiển thị thông tin
3	Nhãn tên CSVC	JLabel	Hiển thị thông tin
4	Nhãn giá CSVC	JLabel	Hiển thị thông tin
6	Nút sửa	JButton	Cập nhật CSVC
7	Nút hủy	JButton	Hủy thao tác
9	Khung giá CSVC	JTextfield	Nhập giá CSVC
10	Khung tên CSVC	JTextfield	Nhập tên CSVC
11	Khung mã CSVC	JTextfield	Hiển thị mã CSVC

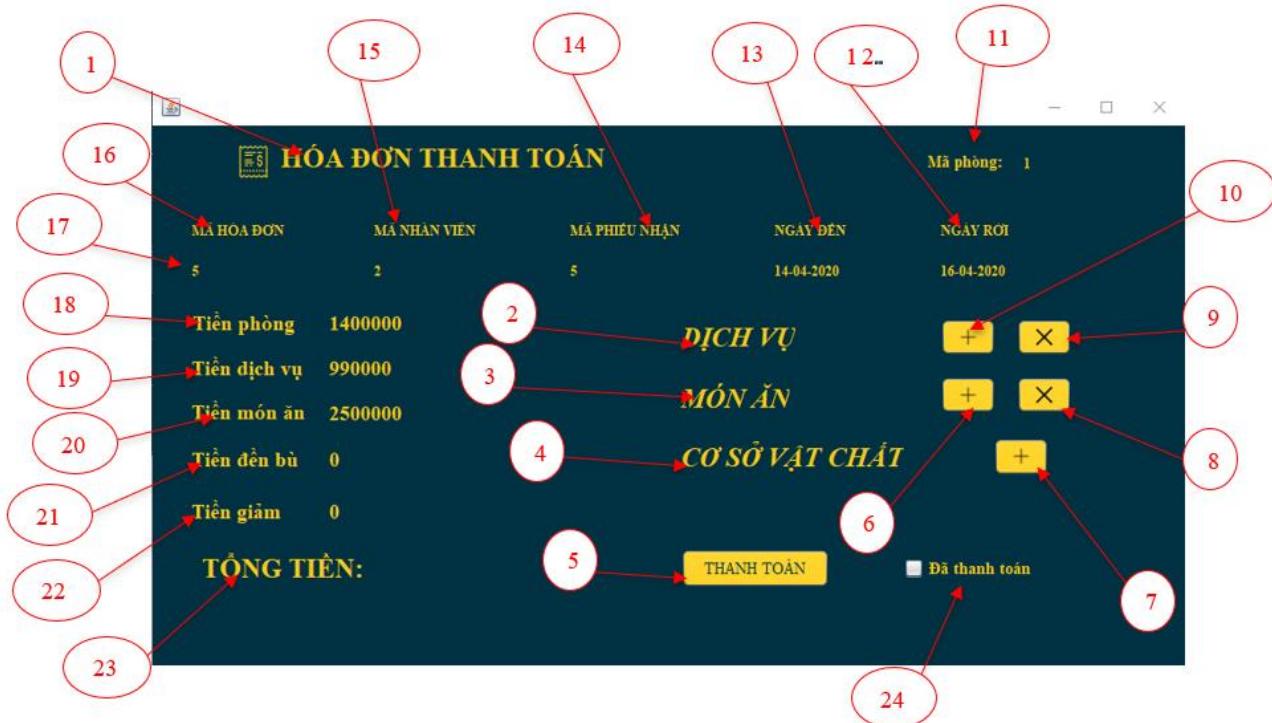
### 6.2.16 Quản lý hóa đơn thanh toán

**HÓA ĐƠN THANH TOÁN**

Mã hóa đơn	Mã nhân viên	Mã phiếu nhận	Mã phòng	Tổng tiền	Tình trạng	Ngày đến	Ngày rời
2	3	2	1	2410000	ĐÃ THANH TO...	12-04-2020	13-04-2020
56	3	2	1	700000	ĐÃ THANH TO...	12-04-2020	13-04-2020
3	3	3	2	3847000	ĐÃ THANH TO...	13-04-2020	15-04-2020
57	3	3	2	1400000	ĐÃ THANH TO...	13-04-2020	15-04-2020
4	4	4	10	3040000	ĐÃ THANH TO...	12-05-2020	15-05-2020
58	4	4	10	2100000	ĐÃ THANH TO...	12-05-2020	15-05-2020
5	2	5	1	0	CHUA THANH ...	14-04-2020	16-04-2020
59	2	5	1	1400000	ĐÃ THANH TO...	14-04-2020	16-04-2020
6	2	6	50	10037000	ĐÃ THANH TO...	14-04-2020	16-04-2020
60	2	6	50	5000000	ĐÃ THANH TO...	14-04-2020	16-04-2020
7	4	7	40	0	CHUA THANH...	15-05-2020	18-05-2020

STT	Tên	Kiểu	Chức năng
1	Bảng thông tin hóa đơn	JTable	Hiển thị thông tin hóa đơn
2	Ô nhập mã phòng	JTextfield	Nhập tên hóa đơn
3	Nhận hóa đơn thanh toán	JLabel	Hiển thị thông tin
4	Nút tìm kiếm	JButton	Thực hiện chức năng tìm kiếm
6	Nút xóa	JButton	Thực hiện chức năng xóa

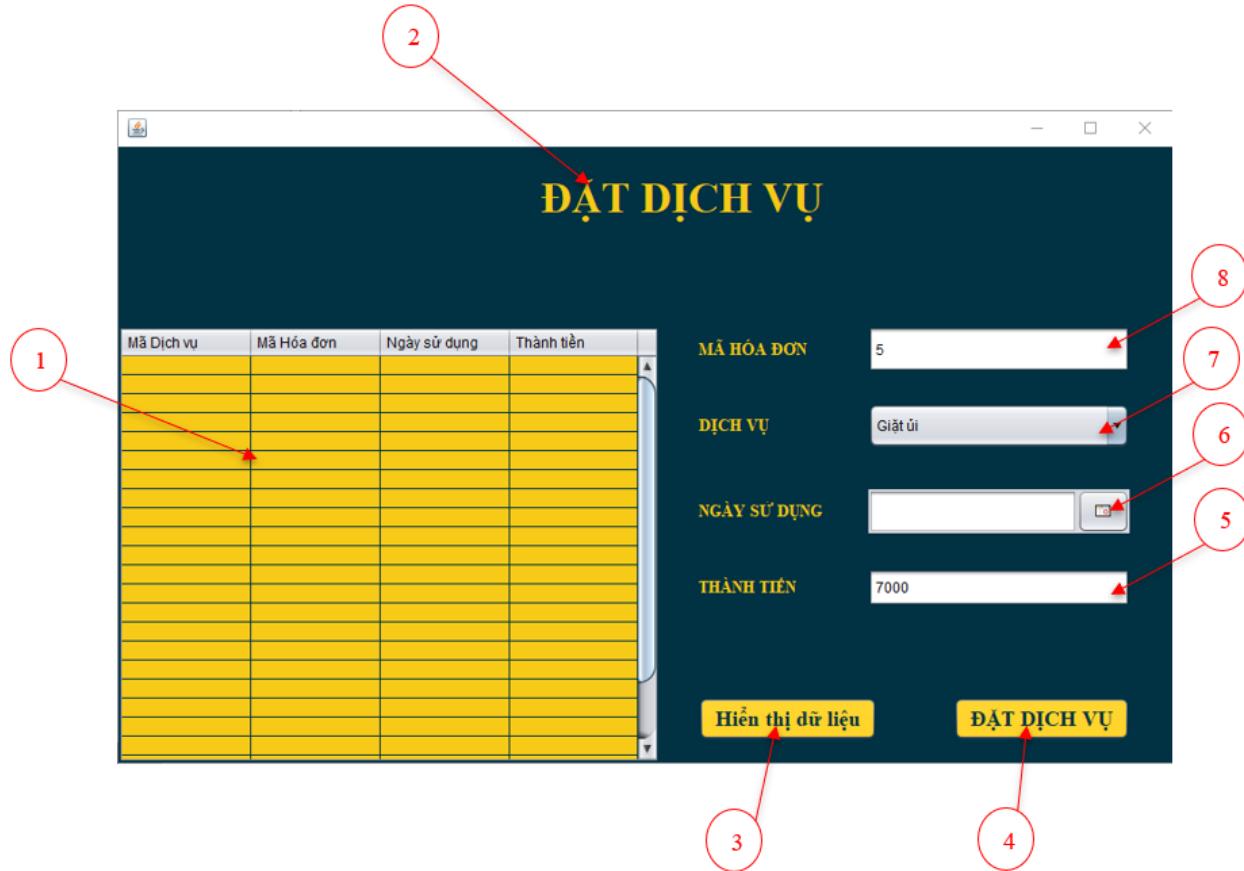
### 6.2.17 Hóa đơn thanh toán



STT	Tên	Kiểu	Chức năng
1	Nhãn hóa đơn thanh toán	JLabel	Hiển thị thông tin
2	Nhãn dịch vụ	JLabel	Hiển thị thông tin
3	Nhãn món ăn	JLabel	Hiển thị thông tin
4	Nhãn CSVC	JLabel	Hiển thị thông tin
5	Nhãn thanh toán	JLabel	Hiển thị thông tin
6	Nút thêm bảng kê món ăn	JButton	Thêm bảng kê món ăn
7	Nút thêm hóa đơn CSVC	JButton	Thêm hóa đơn CSVC
8	Nút xóa bảng kê món ăn	JButton	Xóa bảng kê món ăn
9	Nút xóa bảng kê dịch vụ	JButton	Xóa bảng kê dịch vụ
10	Nút thêm bảng kê dịch vụ	JButton	Thêm bảng kê dịch vụ
11	Nhãn mã phòng	JLabel	Hiển thị thông tin
12	Nhãn ngày rời	JLabel	Hiển thị thông tin
13	Nhãn ngày đến	JLabel	Hiển thị thông tin
14	Nhãn phiếu nhận	JLabel	Hiển thị thông tin
15	Nhãn mã nhân viên	JLabel	Hiển thị thông tin
16	Nhãn mã hóa đơn	JLabel	Hiển thị thông tin
17	Ô mã hóa đơn	JLabel	Hiển thị thông tin
18	Nhãn tiền phòng	JLabel	Hiển thị thông tin
19	Nhãn tiền dịch vụ	JLabel	Hiển thị thông tin
20	Nhãn tiền món ăn	JLabel	Hiển thị thông tin
21	Nhãn tiền đèn bù	JLabel	Hiển thị thông tin
22	Nhãn tiền giảm	JLabel	Hiển thị thông tin

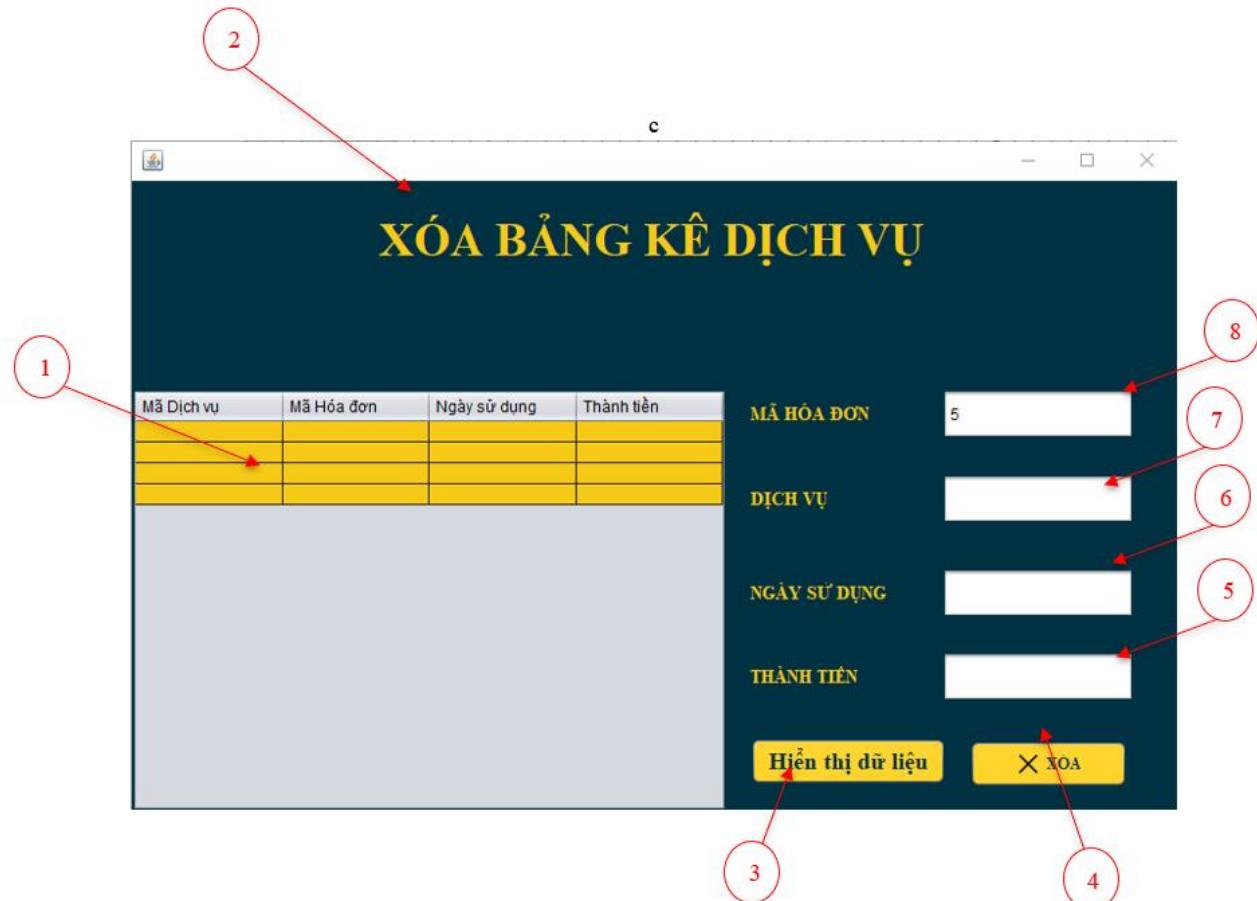
23	Nhãn tổng tiền	JLabel	Hiển thị thông tin
24	Check thanh toán	JCheckBox	Đánh dấu hóa đơn đã thanh toán

#### 6.2.18 Đặt dịch vụ



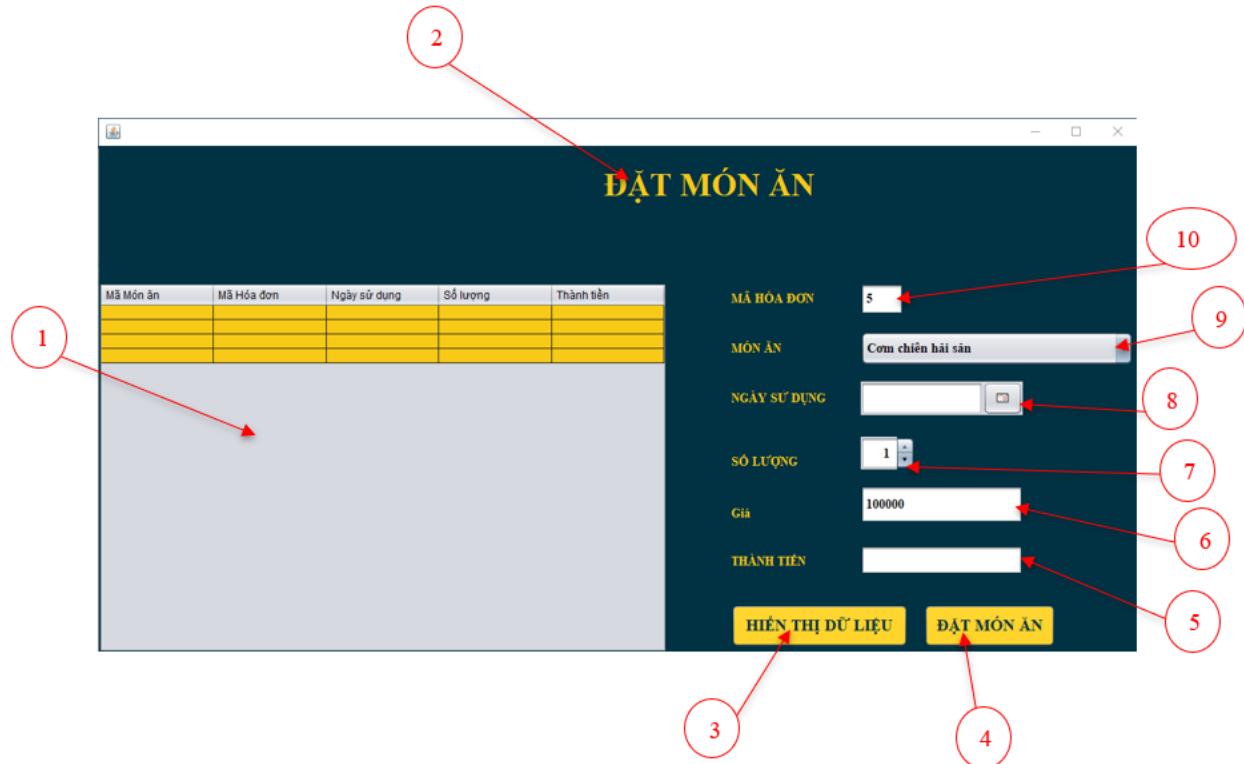
STT	Tên	Kiểu	Chức năng
1	Bảng danh sách cách bảng kê dịch vụ	JTable	Hiển thị thông tin hóa đơn
2	Nhãn đặt dịch vụ	JLabel	Hiển thị thông tin
3	Nút hiển thị	JButton	Hiển thị danh sách bảng kê dịch vụ
4	Nút đặt dịch vụ	JButton	Thực hiện chức năng lập bảng kê
5	Ô thành tiền	JTextfield	Hiển thị thành tiền
6	Ô ngày	JdateChooser	Hiển thị ngày
7	Ô dịch vụ	JCombobox	Hiển thị danh sách các dịch vụ
8	Ô mã hóa đơn	JTextfield	Hiển thị mã hóa đơn

### 6.2.19 Xóa bảng kê dịch vụ



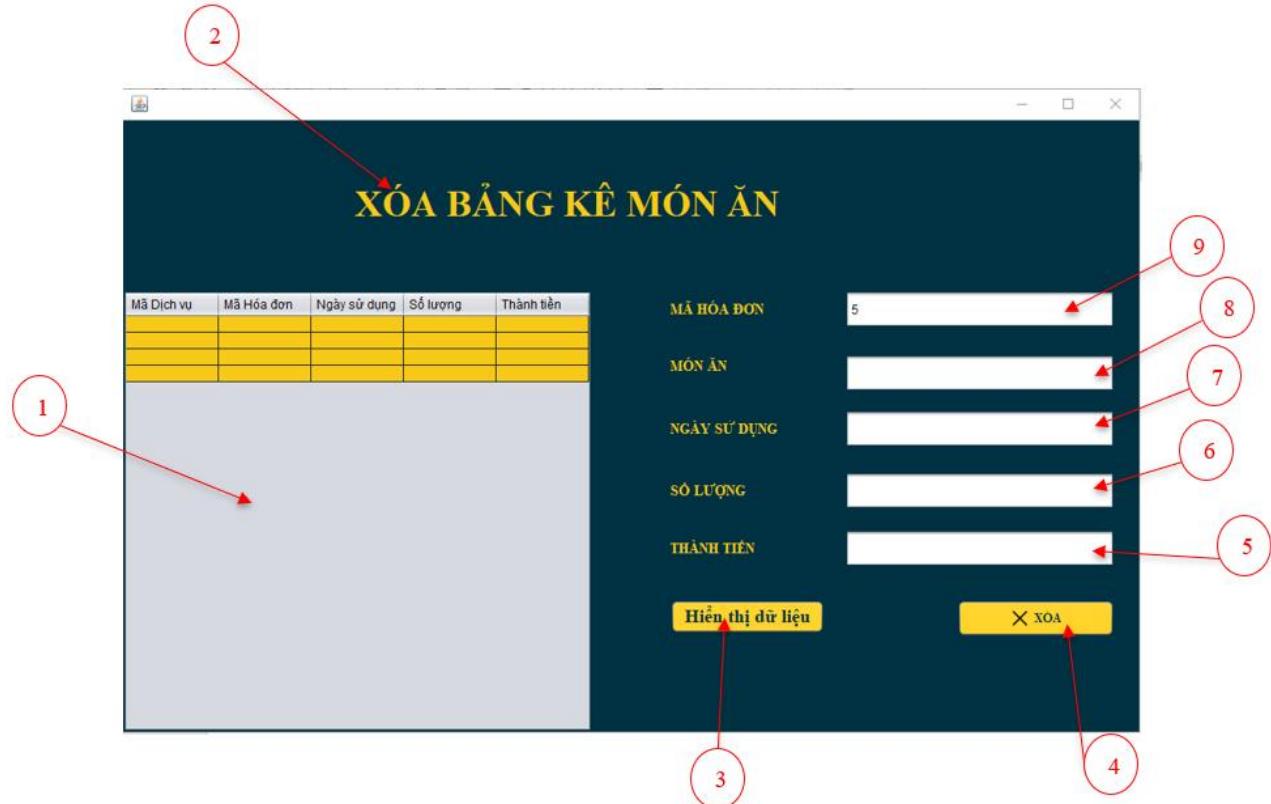
STT	Tên	Kiểu	Chức năng
1	Bảng danh sách cách bảng kê dịch vụ	JTable	Hiển thị thông tin hóa đơn
2	Nhãn đặt dịch vụ	JLabel	Hiển thị thông tin
3	Nút hiển thị	JButton	Hiển thị danh sách bảng kê dịch vụ
4	Nút xóa	JButton	Thực hiện chức năng xóa bảng kê
5	Ô thành tiền	JTextfield	Hiển thị thành tiền
6	Ô ngày	JTextfield	Hiển thị ngày
7	Ô dịch vụ	JTextfield	Hiển thị danh sách các dịch vụ
8	Ô mã hóa đơn	JTextfield	Hiển thị mã hóa đơn

### 6.2.20 Đặt món ăn



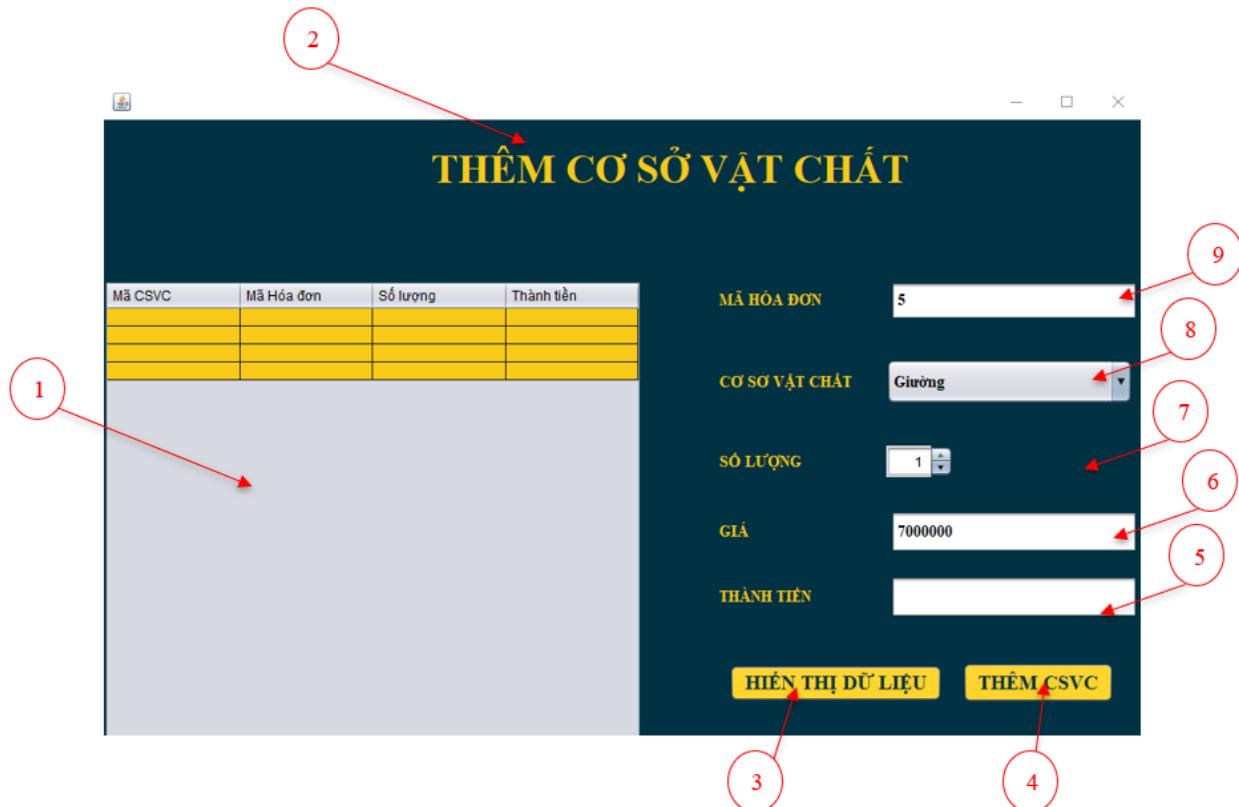
STT	Tên	Kiểu	Chức năng
1	Bảng danh sách cách bảng kê món ăn	JTable	Hiển thị thông tin
2	Nhãn đặt dịch vụ	JLabel	Hiển thị thông tin
3	Nút hiển thị	JButton	Hiển thị danh sách bảng kê dịch vụ
4	Nút đặt món ăn	JButton	Thực hiện chức năng xóa bảng kê
5	Ô thành tiền	JTextfield	Hiển thị thành tiền
6	Ô giá	JTextfield	Hiển thị giá
7	Ô số lượng	JCombobox	Hiển thị số lượng
8	Ô ngày	JdateChooser	Hiển thị ngày
9	Ô món ăn	JCombobox	Hiển thị danh sách các dịch vụ
10	Ô mã hóa đơn	JTextfield	Hiển thị mã hóa đơn

### 6.2.21 Xóa bảng kê món ăn



STT	Tên	Kiểu	Chức năng
1	Bảng danh sách cách bảng kê món ăn	JTable	Hiển thị thông tin
2	Nhãn xóa bảng kê món ăn	JLabel	Hiển thị thông tin
3	Nút hiển thị	JButton	Hiển thị danh sách bảng kê món ăn
4	Nút xóa món ăn	JButton	Thực hiện chức năng xóa bảng kê
5	Ô thành tiền	JTextfield	Hiển thị thành tiền
6	Ô số lượng	JCombobox	Hiển thị số lượng
7	Ô ngày	JdateChooser	Hiển thị ngày
8	Ô món ăn	JCombobox	Hiển thị món ăn
9	Ô mã hóa đơn	JTextfield	Hiển thị mã hóa đơn

### 6.2.22 Thêm hóa đơn CSVC



STT	Tên	Kiểu	Chức năng
1	Bảng danh sách cách bảng kê CSVC	JTable	Hiển thị thông tin
2	Nhãn thêm CSVC	JLabel	Hiển thị thông tin
3	Nút hiển thị	JButton	Hiển thị danh sách bảng kê dịch vụ
4	Nút thêm CSVC	JButton	Thực hiện chức năng xóa hóa đơn
5	Ô thành tiền	JTextfield	Hiển thị thành tiền
6	Ô giá	JTextfield	Hiển thị giá
7	Ô số lượng	JCombobox	Hiển thị số lượng
8	Ô CSVC	JCombobox	Hiển thị danh sách các CSVC
9	Ô mã hóa đơn	JTextfield	Hiển thị mã hóa đơn

## Chương 7: KẾT LUẬN

### 7.1 Kết quả đạt được

Trong suốt quá trình làm đồ án, nhóm đã đạt được những điều sau:

- Biết cách thiết kế, hoàn thiện một dự án nhỏ
- Nắm được các kỹ năng phân chia công việc, sắp xếp thời gian, làm việc nhóm, giải quyết sự cố.
- Hiểu và vận dụng được kiến thức của môn hệ quản trị cơ sở dữ liệu, cụ thể là làm việc trên Oracle, như:
  - Các cú pháp và cách suy nghĩ của ngôn ngữ PL/SQL
  - Cách thực hiện một Trigger, Function, Stored Procedure
  - Trang bị thêm kiến thức về khóa và các mức độ lập
  - Giải quyết được các trường hợp truy xuất đồng thời
- Xây dựng được phần mềm “Quản lý khách sạn” với các chức năng sau:
  - Quản lý nhân viên, khách hàng
  - Quản lý phòng, loại phòng, đặt phòng, nhận phòng
  - Quản lý dịch vụ, món ăn, CSVC
  - Quản lý thống kê

## 7.2 Hạn chế

Do thời gian và yêu cầu công việc ban đầu khá lớn nên trong quá trình thực hiện đồ án nhóm còn gặp một số vấn đề hạn chế:

- Giao diện quá đơn giản, chưa thẩm mỹ
- Phân chia công việc không đều
- Đề ra quá nhiều chức năng khó nhưng quá sức với nhóm
- Các tình huống truy xuất đồng thời còn đơn giản

## 7.3 Hướng phát triển

Để khắc phục các hạn chế còn tồn tại cũng như phát triển phần mềm nhóm có một số đề xuất:

- Dùng các công nghệ hiện đại hơn.
- Chú trọng hơn vào các chức năng chính
- Phần mềm hỗ trợ đa ngôn ngữ

## Phụ lục 1: BẢNG PHÂN CHIA CÔNG VIỆC

Công việc	Nguyễn Lê Mỹ Uyên	Lương Nguyễn Tường Vy	Nguyễn Vũ Tú Anh	Võ Thị Vân Anh
Phân tích bài toán, xác định chức năng	X	X	X	X
Thiết kế CSDL	X	X	X	X
Thiết kế giao diện	X	X		X
Xử lý giao diện	X	X		X
Xử lý các vấn đề liên quan truy				X

xuất đồng thời				
Test chức năng	X	X		X
Soạn và chỉnh báo cáo	X	X	X	X

## **Phụ lục 2: TÀI LIỆU THAM KHẢO**

1. File báo cáo đề tài: Quản lý học sinh của nhóm (Lê Phạm Huy Cường, Trần Minh Đức, Đỗ Quang Thiện, Dương Quốc Cường)
2. <https://docs.oracle.com/en/database/oracle/oracle-database/index.html> : tài liệu Oracle của hãng Oracle
3. Các slide của các môn học Phân tích thiết kế HTTT, Hệ quản trị CSDL, Lập trình Java