**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**QUẢN LÝ THUÊ TRẢ PHÒNG**

**NHÀ NGHỈ**

**Môn học:**

Hệ quản trị Cơ sở dữ liệu

(IS210.K21)

**Giảng viên phụ trách:** Ths.Đỗ Thị Minh Phụng

**Thành viên nhóm:**

|  |  |
| --- | --- |
| **Họ tên** | **MSSV** |
| Nguyễn Duy Hiệu (NT) | 18520746 |
| Hoàng Lê Nam | 18521120 |
| Bạch Thanh Long | 18521016 |
| Trịnh Ngọc Vĩnh | 18521660 |

**TP. HỒ CHÍ MINH – 04/2020**

Mục lục

[1 Phát biểu bài toán 1](#_Toc44402052)

[1.1 Đặt vấn đề 1](#_Toc44402053)

[1.2 Giải pháp 1](#_Toc44402054)

[1.3 Mục tiêu 1](#_Toc44402055)

[1.4 Phạm vi 2](#_Toc44402056)

[2 Xác định và phân tích yêu cầu 2](#_Toc44402057)

[2.1 Hiện trạng tổ chức 2](#_Toc44402058)

[2.2 Các quy trình nghiệp vụ 2](#_Toc44402059)

[**2.2.1** Chức năng của từng bộ phận 2](#_Toc44402060)

[**2.2.2** Các quy trình nghiệp vụ 3](#_Toc44402061)

[2.3 Phân tích yêu cầu 4](#_Toc44402062)

[**2.3.1** Yêu cầu chức năng 4](#_Toc44402063)

[**2.3.2** Yêu cầu phi chức năng 5](#_Toc44402064)

[3 Thiết kế mô hình quan hệ 6](#_Toc44402065)

[3.1 Mô hình ERD 6](#_Toc44402066)

[3.2 Mô hình quan hệ 6](#_Toc44402067)

[3.3 Bảng thuyết minh quan hệ và thuộc tính 7](#_Toc44402068)

[**3.3.1** Table: LOAIPHONG 7](#_Toc44402069)

[**3.3.2** Table: PHONG 8](#_Toc44402070)

[**3.3.3** Table:LOAIKHACH 8](#_Toc44402071)

[**3.3.4** Table: KHACHHANG 8](#_Toc44402072)

[**3.3.5** Table: PHIEUTHUE 8](#_Toc44402073)

[**3.3.6** Table: CHITIETPT 9](#_Toc44402074)

[**3.3.7** Table: DICHVU 9](#_Toc44402075)

[**3.3.8** Table: KHUYENMAI 9](#_Toc44402076)

[**3.3.9** Table:NGUOIDUNG 10](#_Toc44402077)

[**3.3.10** Table: DANGKYDV 10](#_Toc44402078)

[**3.3.11** Table: DANGKYKM 10](#_Toc44402079)

[**3.3.12** Table: QUANLI 11](#_Toc44402080)

[3.4 Mô tả ràng buộc toàn vẹn 11](#_Toc44402081)

[4 Công nghệ và công cụ hỗ trợ 18](#_Toc44402082)

[4.1 Oracle 18](#_Toc44402083)

[4.2 JDBC (Java Database Connectivity) 19](#_Toc44402084)

[4.3 JavaSwing 20](#_Toc44402085)

[4.4 GitHub 21](#_Toc44402086)

[4.5 NetBeans 22](#_Toc44402087)

[4.6 NetBeans IDE 22](#_Toc44402088)

[4.7 SQLDevloper 23](#_Toc44402089)

[5 Xây dựng và quản lý giao tác 23](#_Toc44402090)

[5.1 Trigger 23](#_Toc44402091)

[**5.1.1** Trigger trong Oracle 23](#_Toc44402092)

[**5.1.2** Danh sách Trigger 25](#_Toc44402093)

[**5.1.3** Mô tả một số trigger 25](#_Toc44402094)

[5.1.3.1 Kiểm tra ngày đặt thuê phòng 25](#_Toc44402095)

[5.1.3.2 Áp dụng khuyến mãi tự động 26](#_Toc44402096)

[5.2 Stored Procedure 28](#_Toc44402097)

[**5.2.1** Stored Procedure trong Oracle 28](#_Toc44402098)

[**5.2.2** Transaction trong Oracle 28](#_Toc44402099)

[5.2.2.1 Định nghĩa 28](#_Toc44402100)

[5.2.2.2 Cấu trúc transaction 29](#_Toc44402101)

[5.2.2.3 Transaction control 29](#_Toc44402102)

[5.2.2.4 Enqueues transaction 30](#_Toc44402103)

[5.2.2.5 Autonomous Transactions 31](#_Toc44402104)

[**5.2.3** Danh sách các Procedure 35](#_Toc44402105)

[**5.2.4** Mô tả một số Stored Procedure 36](#_Toc44402106)

[5.2.4.1 Thêm khách hàng vào phiếu thuê 36](#_Toc44402107)

[5.2.4.2 Thủ tục thanh toán phiếu thuê 38](#_Toc44402108)

[6 Xử lý truy xuất đồng thời 40](#_Toc44402109)

[6.1 Các mức cô lập trong Oracle 40](#_Toc44402110)

[**6.1.1** Read Committed 40](#_Toc44402111)

[**6.1.2** Serializable 40](#_Toc44402112)

[**6.1.3** Read-only 41](#_Toc44402113)

[6.2 Cơ chế khóa 41](#_Toc44402114)

[6.3 Deadlock 42](#_Toc44402115)

[6.4 Thiết lập mức cô lập 43](#_Toc44402116)

[6.5 Mô tả đồ án môn học 44](#_Toc44402117)

[**6.5.1** Lost update 44](#_Toc44402118)

[**6.5.2** Non-repeatable read 49](#_Toc44402119)

[**6.5.3** Phantom read 50](#_Toc44402120)

[7 Thiết kế giao diện 53](#_Toc44402121)

[7.1 Danh sách các màn hình 53](#_Toc44402122)

[7.2 Mô tả các màn hình 54](#_Toc44402123)

[8 Kết luận 68](#_Toc44402124)

[8.1 Kết quả đạt được 68](#_Toc44402125)

[8.2 Hạn chế 69](#_Toc44402126)

[8.3 Hướng phát triển 69](#_Toc44402127)

[9 Bảng phân công công việc 69](#_Toc44402128)

[10 Tài liệu tham khảo 70](#_Toc44402129)

# Phát biểu bài toán

## Đặt vấn đề

Hiện nay, với sự bùng nổ của thời đại công nghệ 4.0, thiết kế ứng dụng công nghệ đã rất thành công trong nhiều lĩnh vực, trong đó có cả kinh doanh khách sạn. Ứng dụng quản lý khách sạn giúp gia tăng sự hài lòng của khách hàng, kinh doanh hiệu quả hơn.

Với sự mở rộng quy mô cả về số lượng và chất lượng các dịch vụ khách sạn, đòi hỏi nhà quản lý cần thay đổi phương pháp, cách thức quản lý các hoạt động của khách sạn sao cho hiệu quả nhất, tiết kiệm chi phí nhất. Đây là lúc các nhà quản trị cần sự hỗ trợ đắc lực của phần mềm quản lý khách sạn.

Phần mềm quản lý dành cho khách sạn là sự tích hợp của nhiều chức năng khác nhau để phục vụ cho các khía cạnh công việc khác nhau của nhà quản trị như việc quản lý nhân viên, đặt phòng khách sạn, kế toán, thu – chi trong khách sạn,... giúp nhà quản trị có cái nhìn tổng quan nhất về tình hình hoạt động kinh doanh của khách sạn.

Và một lý do vô cùng quan trọng khác mà bạn có thể nhanh chóng nhận ra, đó là hàng ngày mọi người dành rất nhiều thời gian để truy cập, lướt feed, xem app và làm việc cá nhân trên máy tính, điện thoại. Chỉ cần ngồi ở nhà, khách hang cũng đã có thể hoàn toàn đặt vé, phòng ngay trên ứng dụng.

## Giải pháp

Nhằm giải quyết các vấn đề nêu trên, nhóm đưa ra giải pháp là xây dựng ứng dụng Quản lý khách sạn, để khách hàng và nhân viên lễ tân có thể tham khảo thông tin phòng, giá phòng và cuối cùng là đặt phòng của tất cả các khách sạn trong hệ thống. Đặc biệt, nhân viên lễ tân có thể dựa vào thông tin có từ ứng dụng để tư vấn cho khách hàng trực tiếp ngay tại quầy được dễ dàng hơn. Đồng thời, ứng dụng cũng làm cho việc quản lý và chăm sóc khách hàng được tiện lợi và nhanh chóng hơn.

## Mục tiêu

Xây dựng hệ thống quản lý thuê trả phòng và dịch vụ đáp ứng những yêu cầu sau:

* Quản lý đầy đủ thông tin về khách sạn hoặc nhà nghỉ (phòng, khách hàng,..)
* Quản lý quá trình nhập xuất thông tin của phòng, khách hàng, dịch vụ một cách nhanh chóng và dễ dàng.
* Quản lý quá trình thuê trả phòng đồng bộ
* Quản lý và kết xuất báo cáo.
* Quản lý thông tin và giao dịch của khách hàng.
* Quản lý thông tin các dịch vụ và khuyến mãi
* Quản lý các hóa đơn thanh toán
* Thông tin được lưu trữ trong cơ sở dữ liệu, giúp truy xuất, đồng bộ.

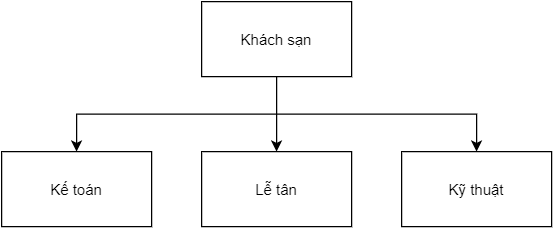
## Phạm vi

Với số lượng khách hàng hàng năm tăng lên không ngừng, khách sạn có nhu cầu cải tiến việc tự động lưu trữ, tìm kiếm và in ấn các báo cáo nhằm đáp ứng được một khối lượng lớn về xử lí thông tin và tính chính xác của thông tin. Sự can thiệp của hệ thống quản lí khách sạn sẽ mang đến hiệu quả hoạt động cao hơn trong công tác quản lí của khách sạn.

Do thời gian thực hiện đồ án ngắn nên việc xây dựng hệ thống chỉ tập trung vào quản lý việc thuê trả phòng, quản lí phòng, quản lí khách hàng, quản lý dịch vụ, khuyến mãi, quản lí việc báo cáo thống kê.

# Xác định và phân tích yêu cầu

## Hiện trạng tổ chức



## Các quy trình nghiệp vụ

### Chức năng của từng bộ phận

* Kế toán:
* Quản lý dịch vụ
* Quản lý khuyến mãi
* Quản lí việc báo cáo thống kê
* Lễ tân:
* Quản lý việc thuê trả phòng
* Kỹ thuật:
* Quản lý phòng

### Các quy trình nghiệp vụ

* Nghiệp vụ quản lí phòng và thuê trả phòng:

• Mỗi phòng cần quản lý thông tin: số phòng, loại phòng, giá phòng, ngày giờ đăng ký, ngày giờ nhận phòng, ngày giờ trả phòng. . .

* Nghiệp vụ quản lí khách hàng:

• Nắm được các thông tin cá nhân của từng khách thuê phòng : Họ tên, địa chỉ, số CMND. Và số lần thuê phòng của từng khách hàng

* Nghiệp vụ chăm sóc khách hàng:

• Khách thuê phòng làm thủ tục đăng ký đặt phòng, nhân viên khách sạn (lễ tân) căn cứ vào yêu cầu của khách về loại phòng, ngày bắt đầu thuê, số lượng phòng nếu đáp ứng được thì lưu vào trong máy tính. Phần này có thể truy tìm nhanh được các khách hàng đã từng lưu trú tại khách sạn

• Khách hàng thường xuyên lưu trú tại khách sạn sẽ được ưu tiên và áp dụng các khuyến mãi

* Nghiệp vụ quản lí dịch vụ, khuyến mãi:
* Các dịch vụ, khuyến mãi sẽ được linh động thay đổi cho phù hợp với nhu cầu của khách hàng và quy trình đăng ký dịch vụ sẽ tối ưu đơn giản và tiện lợi hơn đối với khách hàng
* Nghiệp vụ marketing khuyến mãi:
* Có các chương trình quảng cáo và hình thức quảng bá hình ảnh của khách sạn nhà nghỉ như online qua các mạng xã hội hoặc phát tờ rơi,…
* Viết bài PR cho nhà hàng, khách sạn.
* Phối hợp với các phòng ban khác để triển khai kế hoạch.
* Nghiên cứu và nắm bắt thị hiếu khách hàng cũng như biến chuyển thị trường ngành để đưa ra ý tưởng thực hiện phù hợp
* Nghiệp vụ nhập xuất hóa đơn:
  + Nhập hóa đơn khi khách hàng muốn thuê phòng bao gồm thông tin số phòng, CMND, thời gian bắt đầu thuê phòng
  + Xuất hóa đơn khi khách hàng trả phòng, kiểm tra số tiền thanh toán, in hóa đơn khi khách hàng thanh toán xong
* Nghiệp vụ báo cáo, thống kê:
* Hằng tháng hoặc hằng năm sẽ kết xuất các báo cáo về doanh thu chi tiêu, số lượng khách và loại khách ra vào khách sạn nhà nghỉ
* Để nắm bắt được nhu cầu của khách hàng và đưa ra các chiến lược hoặc hành động cụ thể để phát triển khách sạn

## Phân tích yêu cầu

### *Yêu cầu chức năng*

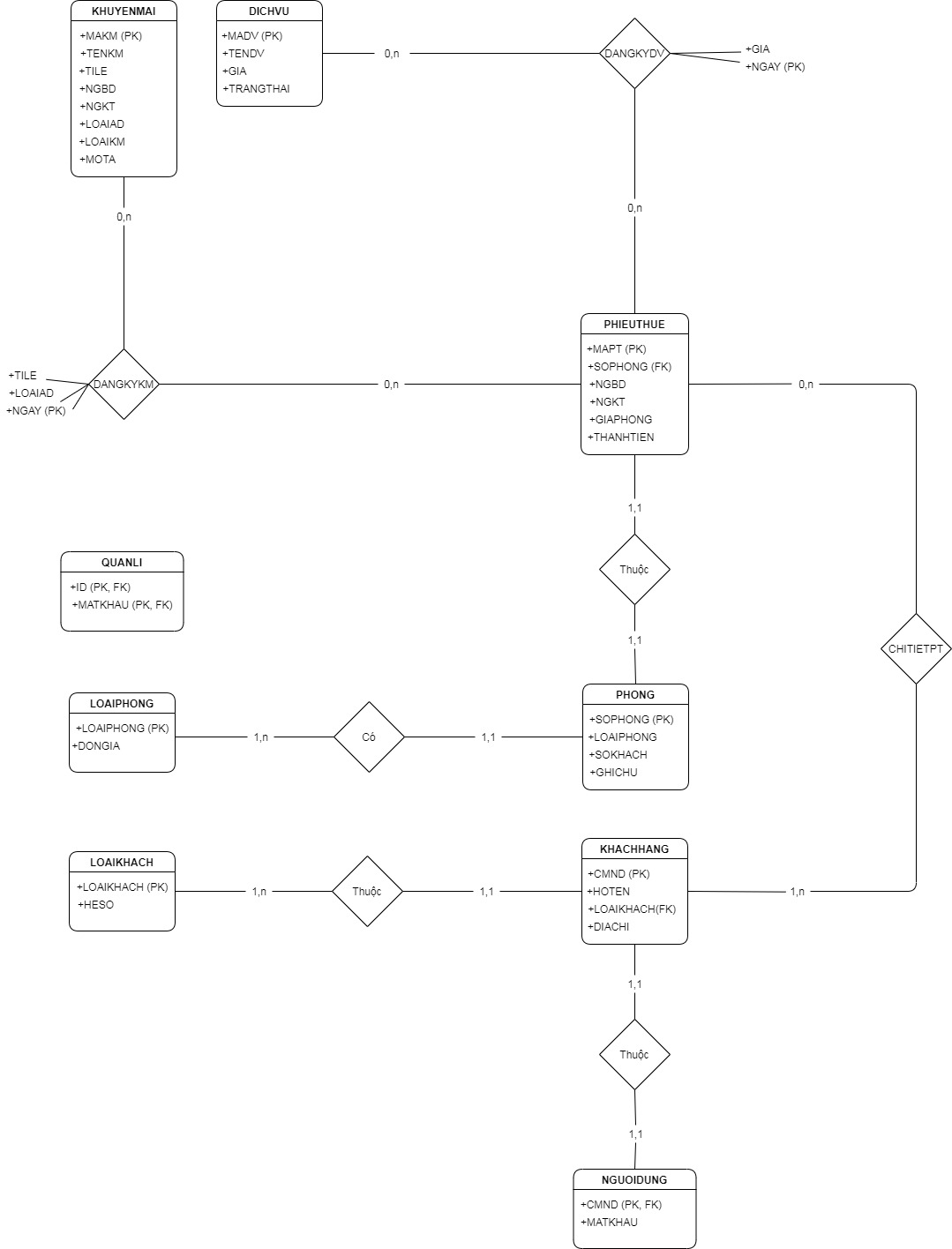
* Lưu trữ
* Thông tin Phòng: số phòng, loại phòng, đơn giá của loại phòng, số khách tối đa mỗi phòng, ghi chú
* Thông tin Khách hàng: CMND, họ tên khách hàng, địa chỉ, loại khách hàng (trong nước ngoài nước) và hệ số tương ứng
* Thông tin các dịch vụ: mã dịch vụ, tên dịch vụ, giá của dịch vụ đóThông tin các khuyến mãi: mã khuyến mãi, tên khuyến mãi, phần trăm khuyến mãi
* Thông tin về phiếu thuê phòng và hóa đơn: mã phiếu thuê (mã hóa đơn), số ngày thuê, thành tiền phiếu thuê, các dịch vụ sử dụng và các khuyến mãi được áp dụng
* Tìm kiếm
* Tra cứu thông tin phòng viên theo số phòng
* Tra cứu thông tin phòng còn trống
* Tra cứu thông tin khách hàng theo tên, cmnd
* Tra cứu thông tin phiếu thuê và hóa đơn
* Tính toán
* Tính toán đơn giá của phòng theo loại khách
* Tính thành tiền của hóa đơn gồm các khuyến mãi và dịch vụ
* Báo cáo thống kê
* Báo cáo doanh thu theo loại phòng trong tháng rồi
* Báo cáo doanh thu theo loại khách hàng trong tháng rồi

### *Yêu cầu phi chức năng*

* Yêu cầu về giao diện: sử dụng đa ngôn ngữ. Giao diện quản lý gần gũi, đơn giản. Giao diện dùng cho khách hàng thân thiện, sinh động, tương tác cao.
* Yêu cầu chất lượng:
* Tính tiến hóa:
* Dễ dàng trong lúc nâng cấp hoặc tăng các module tiện ích.
* Tính tiện dụng:
* Hệ thống có giao diện trực quan, thân thiện và dễ sử dụng.
* Các đặc tả và hướng dẫn sử dụng rõ ràng.
* Tính hiệu quả:
* Ứng dụng hoạt động ổn định và đáng tin cậy.
* Ứng dụng thể hiện tốt các loại phòng của khách sạn.
* Tính tương thích:
* Hoạt động tốt với các loại hệ điều hành phổ biến như (Android, iOS).

# Thiết kế mô hình quan hệ

## Mô hình ERD



## Mô hình quan hệ

A picture containing text, map

Description automatically generated

## Bảng thuyết minh quan hệ và thuộc tính

### *Table: LOAIPHONG*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | LOAIPHONG | Loại phòng | NvarChar2 | 20 |  | Khóa chính |
| 2 | DONGIA | Đơn giá của loại phòng | Number | (19,3) |  |  |

### *Table: PHONG*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | SOPHONG | Số phòng | Number | (4,0) |  | Khoá chính |
| 2 | LOẠIPHONG | Loại phòng | NvarChar2 | 20 |  | Khóa ngoại |
| 3 | SOKHACH | Số khách tối đa của phòng | Number | (1,0) |  |  |
| 4 | GHICHU | Ghi chú | NvarChar2 | 100 |  |  |

### *Table:**LOAIKHACH*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | LOAIKHACH | Loại khách | NvarChar2 | 20 |  | Khóa chính |
| 2 | HESO | Hệ số của khách ở dạng % | Number | (3,0) |  |  |

### *Table: KHACHHANG*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | CMND | Chứng minh nhân dân | NvarChar2 | 12 |  | Khóa chính |
| 2 | HOTEN | Họ và tên | Nvarchar2 | 50 |  |  |
| 3 | LOAIKHACH | Loại khách | Nvarchar2 | 20 |  | Khóa ngoại |
| 4 | DIACHI | Địa chỉ | Nvarchar2 | 50 |  |  |

### *Table: PHIEUTHUE*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MAPT | Mã phiếu thuê | Number | (7,0) |  | Khóa chính |
| 2 | SOPHONG | Số phòng | Number | (4,0) |  | Khóa ngoại |
| 4 | NGBD | Ngày bắt đầu thuê | Date |  |  |  |
| 5 | NGKT | Ngày trả phòng | Date |  |  |  |
| 6 | GIAPHONG | Giá thuê phòng tính theo hệ số cao nhất của khách | Number | (19,3) |  |  |
| 7 | THANHTIEN | Thành tiền | Number | (19,3) |  |  |

### *Table: CHITIETPT*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MAPT | Mã phiếu thuê | Nvarchar2 | (12) |  | Khóa chính, khóa ngoại |
| 2 | CMND | Chứng minh nhân dân | Nvarchar2 | 25 |  | Khóa chính, khóa ngoại |

### *Table: DICHVU*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MADV | Mã dịch vụ | Number | (3,0) |  | Khóa chính |
| 2 | TENDV | Tên dịch vụ | Nvarchar2 | 50 |  |  |
| 3 | GIA | Giá dịch vụ | Number | (19,4) |  |  |
| 4 | TRANGTHAI | Dịch vụ có sẵn hay không | Number | (1,0) |  |  |

### *Table: KHUYENMAI*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MAKM | Mã khuyến mãi | Number | (3,0) |  | Khóa chính |
| 2 | TENKM | Tên khuyến mãi | Nvarchar2 | 50 |  |  |
| 3 | TILE | Tỉ lệ khuyến mãi | Number | (2,0) |  |  |
| 4 | NGBD | Ngày bắt đầu khuyến mãi | Date |  |  |  |
| 5 | NGKT | Ngày kết thúc khuyến mãi | Date |  |  |  |
| 6 | LOAIAD | Áp dụng trên tổng HĐ, Áp dụng trên tiền thuê phòng | Number | (1,0) |  |  |
| 7 | LOAIKM | Áp dụng KM tự động, áp dụng thủ công | Number | (1,0) |  |  |
| 8 | MOTA | Mô tả thêm KM | Nvarchar2 | (300) |  |  |

### *Table:**NGUOIDUNG*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | CMND | Chứng minh nhân dân | Number | 20 |  | Khóa chính, khóa ngoại |
| 2 | MATKHAU | Mật khẩu | Nvarchar2 | 25 |  |  |

### *Table: DANGKYDV*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MAPT | Mã phiếu thuê | Number | (7,0) |  | Khóa chính, khóa ngoại |
| 2 | MADV | Mã dịch vụ | Number | (3,0) |  | Khóa chính, khóa ngoại |
| 3 | GIA | Giá của dịch vụ | Number | (19,3) |  |  |
| 4 | NGAY | Ngày dịch vụ | Date |  |  | Khóa chính |

### *Table: DANGKYKM*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MAPT | Mã phiếu thuê | Number | (7,0) |  | Khóa chính, khóa ngoại |
| 2 | MADV | Mã khuyến mãi | Number | (3,0) |  | Khóa chính, khóa ngoại |
| 3 | TILE | Tỉ lệ khyến mãi | Number | (2,0) |  |  |
| 4 | LOAIAD | Áp dụng trên tổng HĐ, Áp dụng trên tiền thuê phòng | Number | (1,0) |  |  |
| 5 | NGAY | Ngày sử dụng dịch vụ | Date |  |  | Khóa chính |

### *Table: QUANLI*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | ID | ID của quản lí | Nvarchar2 | (12) |  | Khóa chính, khóa ngoại |
| 2 | MATKHAU | Mật khẩu | Nvarchar2 | (25) |  | Khóa chính, khóa ngoại |

## Mô tả ràng buộc toàn vẹn

1. R1 : “Ngày đăng ký dịch vụ luôn luôn là ngày hiện tại”

RBTV miền giá trị

Bối cảnh : DANGKYDV

∀dkdv∈DANGKYDV: dkdv.NGAY = SYSDATE

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xoá | Sửa |
| DANGKYDV | + | - | +(NGAY) |

1. R2 : “Ngày áp dụng khuyến mãi luôn luôn là ngày hiện tại”

RBTV miền giá trị

Bối cảnh : DANGKYKM

∀ dkkm ∈DANGKYKM: dkkm.NGAY = SYSDATE

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xoá | Sửa |
| DANGKYKM | + | - | +(NGAY) |

1. R3 : “Tỉ lệ khuyến mãi thấp hơn 25 và lớn hơn 0”

RBTV miền giá trị

Bối cảnh : KHUYENMAI

∀km ∈KHUYENMAI : km.TILE < 25 ^ km.TILE > 0

|  |  |  |  |
| --- | --- | --- | --- |
| R3 | Thêm | Xóa | Sửa |
| KHUYENMAI | + | - | +(TILE) |

1. R4 : “Thành tiền tối thiểu là 200000

RBTV miền giá trị

Bối cảnh : PHIEUTHUE

∀pt ∈PHIEUTHUE : pt. THANHTIEN < 200000 🡪 pt. THANHTIEN = 200000

|  |  |  |  |
| --- | --- | --- | --- |
| R4 | Thêm | Xoá | Sửa |
| PHIEUTHUE | + | - | +(THANHTIEN) |

1. R5 : “Nếu phòng đang được thuê thì không được cho khách đăng kí phòng đó”

RBTV liên bộ

Bối cảnh: PHIEUTHUE

∀ pt1∈PHIEUTHUE, ∀ pt2∈PHIEUTHUE: (pt1.NGKT IS NULL)

🡪 ¬∃ pt2(pt2.SOPHONG=pt1.SOPHONG ^ pt2.NGKT IS NULL ^

pt2.NGBD > pt1.NGBD ^ pt2.NGBD < pt1.NGKTDK ^

pt2.NGKTDK >pt1.NGBD ^ pt2.NGKTDK < pt1.NGKTDK)

|  |  |  |  |
| --- | --- | --- | --- |
| R5 | Thêm | Xóa | Sửa |
| PHIEUTHUE | + | - | +(SOPHONG) |

1. R6 : “Ngày kết thúc phải sau ngày bắt đầu thuê phòng.”

RBTV miền giá trị

Bối cảnh: PHIEUTHUE

∀ pt ∈ PHIEUTHUE: pt.NGAYBD ≤ pt.NGAYKT

|  |  |  |  |
| --- | --- | --- | --- |
| R6 | Thêm | Xóa | Sửa |
| PHIEUTHUE | + | - | +(NGAYBD,NGAYKT) |

1. R7 : “Số khách tối đa trong phòng là các giá trị (2,4,6,8)”

RBTV miền giá trị

Bối cảnh: PHONG

∀ p ∈ PHONG: p.SOKHACH = 2 ˅ p.SOKHACH = 4 ˅ p.SOKHACH = 6 ˅ p.SOKHACH = 8

|  |  |  |  |
| --- | --- | --- | --- |
| R7 | Thêm | Xóa | Sửa |
| PHONG | + | - | +(SOKHACH) |

1. R8 : “Tập các giá trị LOAIPHONG trong quan hệ PHONG phải được tìm thấy trong tập các giá trị LOAIPHONG trong quan hệ LOAIPHONG”

RBTV tham chiếu

Bối cảnh: PHONG, LOAIPHONG

∀p ∈PHONG, ∃ lp ∈ LOAIPHONG: p.LOAIPHONG = lp.LOAIPHONG

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Xóa | Sửa |
| PHONG | + | - | +(LOAIPHONG) |
| LOAIPHONG | - | + | -(\*) |

1. R9 : “Tập các giá trị LOAIKHACH trong quan hệ KHACHHANG phải được tìm thấy trong tập các giá trị LOAIKHACH trong quan hệ LOAIKHACH”

RBTV tham chiếu

Bối cảnh: KHACHHANG, LOAIKHACH

∀kh ∈KHACHHANG, ∃ lk ∈ LOAIKHACH: kh.LOAIKHACH = lk.LOAIKHACH

|  |  |  |  |
| --- | --- | --- | --- |
| R9 | Thêm | Xóa | Sửa |
| KHACHHANG | + | - | +(LOAIKHACH) |
| LOAIKHACH | - | + | -(\*) |

1. R10 : “Tập các giá trị MAPT trong quan hệ CHITIETPT phải được tìm thấy trong tập các giá trị MAPT trong quan hệ PHIEUTHUE”

RBTV tham chiếu

Bối cảnh: CHITIETPT, PHIEUTHUE

∀ct ∈CHITIETPT, ∃ pt ∈ PHIEUTHUE: ct.MAPT = pt.MAPT

|  |  |  |  |
| --- | --- | --- | --- |
| R10 | Thêm | Xóa | Sửa |
| CHITIETPT | + | - | -(\*) |
| PHIEUTHUE | - | + | -(\*) |

1. R11 : “Tập các giá trị CMND trong quan hệ NGUOIDUNG phải được tìm thấy trong tập các giá trị CMND trong quan hệ KHACHHANG”

RBTV tham chiếu

Bối cảnh: NGUOIDUNG, KHACHHANG

∀ nd ∈NGUOIDUNG, ∃ kh ∈ KHACHHANG: nd.CMND = kh.CMND

|  |  |  |  |
| --- | --- | --- | --- |
| R11 | Thêm | Xóa | Sửa |
| NGUOIDUNG | + | - | -(\*) |
| KHACHHANG | - | + | -(\*) |

1. R12 : “Tập các giá trị MAPT trong quan hệ DANGKYKM phải được tìm thấy trong tập các giá trị MAPT trong quan hệ PHIEUTHUE”

RBTV tham chiếu

Bối cảnh: DANGKYKM, PHIEUTHUE

∀ dk ∈DANGKYKM, ∃ pt ∈ PHIEUTHUE: dk.MAPT = pt.MAPT

|  |  |  |  |
| --- | --- | --- | --- |
| R12 | Thêm | Xóa | Sửa |
| PHIEUTHUE | - | + | -(\*) |
| DANGKYKM | + | - | -(\*) |

1. R13 : “Tập các giá trị MAPT trong quan hệ DANGKYDV phải được tìm thấy trong tập các giá trị MAPT trong quan hệ PHIEUTHUE”

RBTV tham chiếu

Bối cảnh: DANGKYDV, PHIEUTHUE

∀ dk ∈ DANGKYDV, ∃ pt ∈ PHIEUTHUE: dk.MAPT = pt.MAPT

|  |  |  |  |
| --- | --- | --- | --- |
| R13 | Thêm | Xóa | Sửa |
| PHIEUTHUE | - | + | -(\*) |
| DANGKYDV | + | - | -(\*) |

1. R14 : “Tập các giá trị MAKM trong quan hệ DANGKYKM phải được tìm thấy trong tập các giá trị MAKM trong quan hệ KHUYENMAI”

RBTV tham chiếu

Bối cảnh: DANGKYKM, KHUYENMAI

∀ km ∈ KHUYENMAI, ∃ dk ∈ DANGKYKM: km.MAKM = dk.MAKM

|  |  |  |  |
| --- | --- | --- | --- |
| R14 | Thêm | Xóa | Sửa |
| KHUYENMAI | - | + | -(\*) |
| DANGKYKM | + | - | +(MAKM) |

1. R15 : “Tập các giá trị MADV trong quan hệ DANGKYDV phải được tìm thấy trong tập các giá trị MADV trong quan hệ DICHVU”

RBTV tham chiếu

Bối cảnh: DANGKYDV, DICHVU

∀ dv ∈ DICHVU, ∃ dk ∈ DANGKYDV: dv. MADV = dk. MADV

|  |  |  |  |
| --- | --- | --- | --- |
| R15 | Thêm | Xóa | Sửa |
| DICHVU | - | + | -(\*) |
| DANGKYDV | + | - | +MADV |

1. R16 : “Tập các giá trị CMND trong quan hệ CHITIETPT phải được tìm thấy trong tập các giá trị CMND trong quan hệ KHACHHANG”

RBTV tham chiếu

Bối cảnh: CHITIETPT, KHACHHANG

∀ kh ∈ KHACHHANG, ∃ ct ∈ CHITIETPT: kh.CMND = ct. CMND

|  |  |  |  |
| --- | --- | --- | --- |
| R16 | Thêm | Xóa | Sửa |
| CHITIETPT | + | - | -(\*) |
| KHACHHANG | - | + | -(\*) |

1. R17 : “Tập các giá trị SOPHONG trong quan hệ PHIEUTHUE phải được tìm thấy trong tập các giá trị SOPHONG trong quan hệ PHONG”

RBTV tham chiếu

Bối cảnh: PHIEUTHUE, PHONG

∀ pt ∈ PHIEUTHUE, ∃ p ∈ PHONG: pt.SOPHONG = p.SOPHONG

|  |  |  |  |
| --- | --- | --- | --- |
| R17 | Thêm | Xóa | Sửa |
| PHIEUTHUE | + | - | +(SOPHONG) |
| PHONG | - | + | -(\*) |

1. R18 : “Ngày sử dụng khuyến mãi phải sau ngày bắt đầu thuê phòng và trước ngày trả phòng, trường hợp ngày kết thúc chưa nhập thì ngày áp dụng khuyến mãi chỉ cần lớn hơn ngày bắt đầu”

RBTV liên thuộc tính – liên quan hệ

Bối cảnh: DANGKYKM,PHIEUTHUE

∀ km ∈ DANGKYKM, ∃ pt ∈PHIEUTHUE: km.MAPT = pt.MAPT ^

((km.NGAY > pt.NGAYBD km.NGAY ≤ pt.NGAYKT) ˅

(dv.NGAY > pt.NGAYBD ^ pt.NGAYKT IS NULL))

|  |  |  |  |
| --- | --- | --- | --- |
| R18 | Thêm | Xóa | Sửa |
| DANGKYKM | + | - | +(NGAY) |
| PHIEUTHUE | - | - | +(NGAYBD,NGAYKT) |

1. R19 : “Ngày sử dụng dịch vụ phải sau ngày bắt đầu thuê phòng và trước ngày trả phòng, trường hợp ngày kết thúc chưa nhập thì ngày đk dịch vụ chỉ cần lớn hơn ngày bắt đầu”

RBTV liên thuộc tính – liên quan hệ

Bối cảnh: DANGKYDV,PHIEUTHUE

∀ dv ∈ DANGKYDV, ∃ pt ∈ PHIEUTHUE: dv.MAPT = pt.MAPT ^

((dv.NGAY > pt.NGAYBD ^ dv.NGAY ≤ pt.NGAYKT)˅

(dv.NGAY > pt.NGAYBD ^ pt.NGAYKT IS NULL))

|  |  |  |  |
| --- | --- | --- | --- |
| R19 | Thêm | Xóa | Sửa |
| DANGKYDV | + | - | +(NGAY) |
| PHIEUTHUE | - | - | +(NGAYBD,NGAYKT) |

1. R20 : “Khi thêm, xóa khách hàng thì thêm, xóa người dùng với username là CMND và password là CMND”

RBTV tham chiếu

Bối cảnh: KHACHHANG, NGUOIDUNG

kh KHACHHANG nd NGUOIDUNG: nd.CMND = kh.CMND

nd.MATKHAU = kh.CMND

nd NGUOIDUNGkh KHACHHANG: nd.CMND = kh.CMND

|  |  |  |  |
| --- | --- | --- | --- |
| R20 | Thêm | Xóa | Sửa |
| KHACHHANG | + | + | -(\*) |
| NGUOIDUNG | - | - | +(MATKHAU) |

1. R21 : “Số khách đăng kí phòng không được vượt quá số khách tối đa của phòng”

RBTV do thuộc tính tổng hợp

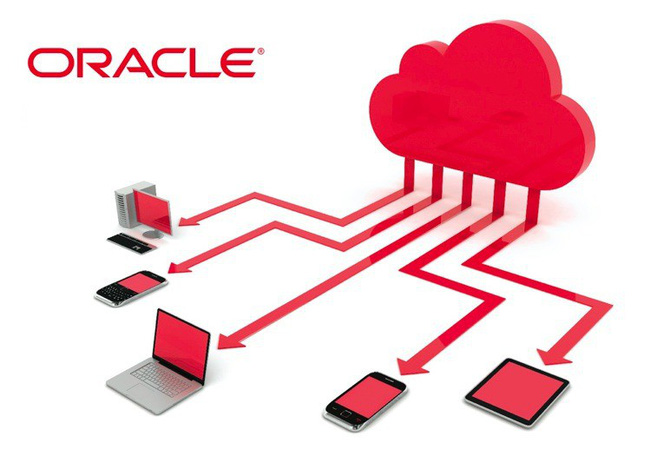
Bối cảnh: PHONG, PHIEUTHUE, CHITIETPT

∀p∈PHONG, p.SOKHACH  COUNT(sk ∈(CHITIETPT⋈(MAPT)PHIEUTHUE): sk.SOPHONG=pt.SOPHONG) (sk.CMND)

|  |  |  |  |
| --- | --- | --- | --- |
| R21 | Thêm | Xóa | Sửa |
| PHONG | - | - | +(SOKHACH) |
| PHIEUTHUE | + | - | +(SOPHONG) |
| CHITIETPT | + | - | -(\*) |

# Công nghệ và công cụ hỗ trợ

## Oracle

[](https://techvccloud.mediacdn.vn/2018/11/15/oraclecloudsolutions-1542271451547247956679.jpg)

Oracle là một trong những nhà cung cấp lớn nhất trên thị trường công nghệ hiện nay. Cái tên Oracle chính là tên viết tắt từ sản phẩm chủ lực của hãng, hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) có tên chính thức là Oracle Database. Phần mềm cơ sở dữ liệu thường giữ vị trí trung tâm trong mảng IT của công ty, hỗ trợ nhiều nhiệm vụ khác nhau gồm xử lý giao dịch, business intelligence (BI), và các ứng dụng phân tích.

Năm 1979, Oracle Corp. là công ty đầu tiên đưa nền tảng RDBMS thị trường, và công ty hiện vẫn là nhà cung cấp cơ sở dữ liệu hàng đầu với tỷ suất lợi nhuận khổng lồ. Dẫn đầu và chiếm phần lớn là từ doanh số bán hàng của Oracle Database, chiếm 40,4% doanh thu cho phần mềm cơ sở dữ liệu trên toàn thế giới trong năm 2016, theo Gartner; tuy đã giảm 2% so với năm 2015, nhưng vẫn gấp đôi số cổ phần so với Microsoft ở vị trí thứ hai.

Oracle Database được xây dựng dựa trên SQL, một ngôn ngữ lập trình chuẩn hóa để quản trị cơ sở dữ liệu, các nhà phân tích dữ liệu và các chuyên gia CNTT thường sử dụng công cụ này để quản lý cơ sở dữ liệu và truy vấn dữ liệu được lưu trữ trong đó. Phần mềm Oracle được gắn với PL/SQL, một phần mềm bổ trợ được Oracle phát triển nhằm bổ sung một số extension độc quyền cho SQL chuẩn – khá phổ biến trong các nhà cung cấp RDBMS. Cơ sở dữ liệu Oracle cũng hỗ trợ lập trình bằng Java và các chương trình được viết trên PL/SQL hoặc lập trình Java có thể được gọi từ ngôn ngữ khác.

Oracle sau nhiều thập kỷ phát triển đã mở rộng đáng kể các danh mục sản phẩm. Hiện tại, hãng cũng cung cấp một số cơ sở dạng dữ liệu khác, một số lượng lớn các dòng ứng dụng kinh doanh, phần mềm phân tích dữ liệu, phần mềm trung gian, hệ thống máy tính, thiết bị lưu trữ dữ liệu, công cụ phát triển và các công nghệ khác…

Ngoài ra, Oracle cũng đang dần dịch chuyển cơ cấu để hướng tới trở thành một nhà cung cấp các dịch vụ điện toán đám mây hàng đầu, sau khi những bước khởi điểm khá muộn màng trong nắm bắt công nghệ đám mây

## JDBC (Java Database Connectivity)

Java JDBC là một java API được sử dụng để kết nối và thực hiện truy vấn với cơ sở dữ liệu. JDBC API sử dụng trình điều khiển jdbc để kết nối với cơ sở dữ liệu.

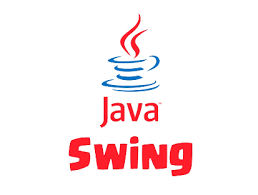


JDBC chỉ là một tập các interface, các định nghĩa, thông báo lỗi, đặc tả chứ không phải là thư viện. Với mỗi hệ quản trị cơ sở dữ liệu ta sẽ có một cài đặt JDBC riêng cho nó, ví dụ JDBC cho MySQL, JDBC cho MS SQL, JDBC cho OracleDatabase …

Các thành phần của JDBC:

* DriverManager: Dùng để quản lý danh sách các Driver (database drivers).
* Driver: Dùng để liên kết các kết nối tới cơ sở dữ liệu, điều khiển các liên kết.
* Connection: Biểu thị kết nối tới cơ sở dữ liệu. Dùng để tạo ra Statement, PreparedStatement và CallableStatement.
* Statement, PreparedStatement, CallableStatement: Chứa lệnh SQL gửi tới cơ sở dữ liệu để thực thi.
* ResultSet – biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là SELECT.
* SQLException – một lớp xử lý lỗi ngoại lệ chứa các lỗi truy cập cơ sở dữ liệu.

## JavaSwing



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

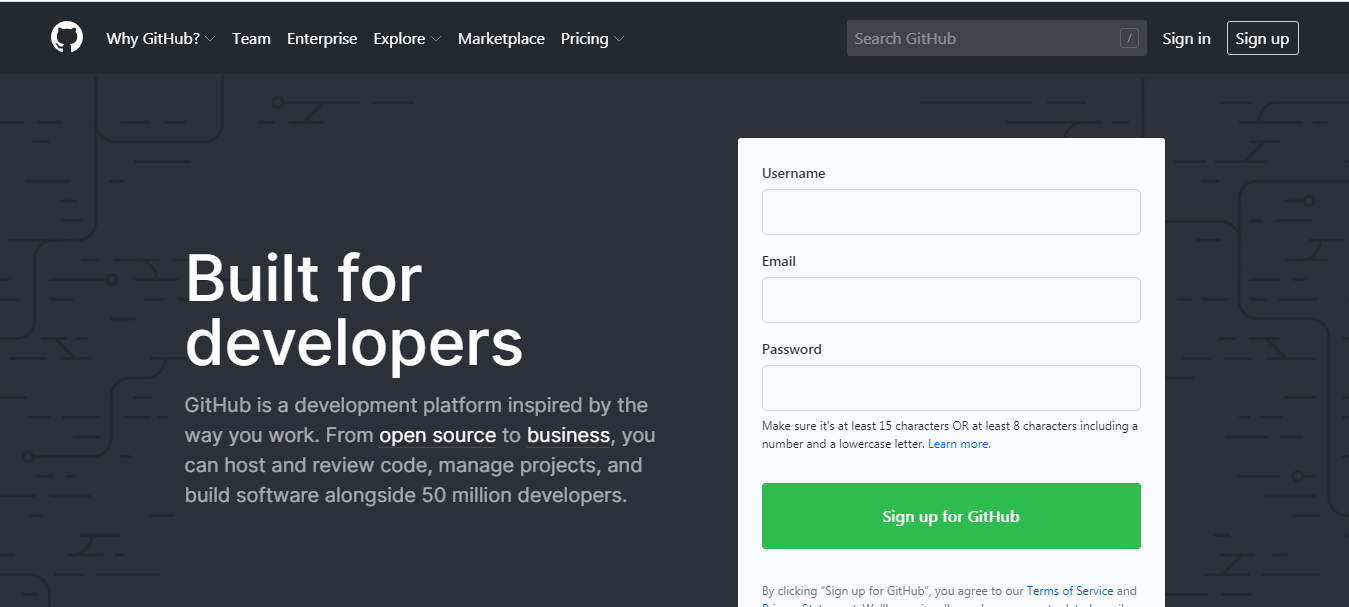
Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.

Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, v.v.

Ưu điểm của Swing so với AWT:

* Cung cấp thêm các đối tượng mới để xây dựng giao diện đồ họa.
* Có khả năng tùy biến cao.
* Look-and-feel: tùy biến để các thành phần giao diện của Swing nhìn giống như các thành phần giao diện của hệ điều hành.
* Có nhiều tài liệu hướng dẫn nghiên cứu

## GitHub

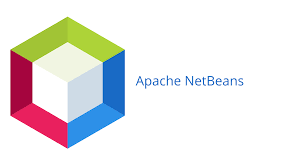


Đây là gã khổng lồ trong lĩnh vực này với nhiều code repository nổi tiếng, không những về nhúng mà còn về rất nhiều thứ khác. Nó hoàn toàn miễn phí cho tất cả mọi người sử dụng, hầu hết các dự án opensource đều được cung cấp mã nguồn tại đây. Tuy nhiên đúng với tính chất mở, nếu bạn có dự án yêu cầu tính bảo mật thì có thể mua phiên bản trả phí

Một số tính năng chính:

* Xem các thay đổi, các chú thích ở từng dòng code, bác cáo các lỗi
* Dễ dàng làm việc với người khác
* Có tích hợp với một số ứng dụng và công cụ
* Có 1 số công cụ để kiểm tra project
* Mỗi GitHub repository đều được xem như là Subversion repository
* Có thể dùng công cụ của SVN để checkout và commit lên GitHub
* Có phiên bản cho Windows, Linux và Mac OS

## 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. Ngoài việc phát triển Java, nó còn có các phần mở rộng cho các ngôn ngữ khác như PHP, C, C++, HTML5 và JavaScript. Các ứng dụng dựa trên NetBeans, bao gồm NetBeans IDE, có thể được mở rộng bởi các nhà phát triển bên thứ ba.

## NetBeans IDE

NetBeans IDE là một công cụ hỗ trợ lập trình viết mã code miễn phí được cho là tốt nhất hiện nay, được sử dụng chủ yếu cho các lập trình viên phát triển Java tuy nhiên phần mềm có dung lượng khá là nặng dành cho các máy cấu hình có RAM, CPU tương đối cao để vận hành.

NetBeans IDE là môi trường phát triển tích hợp và cực kỳ cần thiết cho các lập trình viên, công cụ này có thể hoạt động tốt với rất nhiều nền tảng hệ điều hành khác nhau như Linux, Windows, MacOS,... là một mã nguồn mở cung cấp các tính năng cần thiết nhất nhăm tại ra các ứng dụng web, thiết bị di động, desktop.

## SQLDevloper



Oracle SQL Developer là phần mềm do chính hãng Oracle phát triển. Bạn có thể download phần mềm này miễn phí.

Hoặc khi bạn cài đặt Oracle Client, một bản Oracle SQL Developer cũng đã được tích hợp sẵn

# Xây dựng và quản lý giao tác

## Trigger

### *Trigger trong Oracle*

Trigger là một chương trình lưu trữ nó sẽ tự động thực thi khi một vài sự kiện xảy ra. Thực tế Trigger được viết để phản hồi lại những sự kiện xảy ra khi thực hiện các lệnh dưới đây:

* Các câu lệnh thuộc nhóm lệnh kiểm soát dữ liệu (DELETE, INSERT, or UPDATE).
* Các câu lệnh thuộc nhóm lệnh định nghĩa dữ liệu DML (CREATE, ALTER, or DROP).
* Các câu lệnh hoạt động dữ liệu (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Lợi ích khi sử dụng Trigger:

* Ngăn ngừa những giao tác không hợp lệ.
* Tự động tạo ra những cột dữ liệu phụ.
* Nhân rộng bảng đồng bộ.

Cú pháp:

CREATE [OR REPLACE ] TRIGGER trigger\_name

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col\_name]

ON table\_name

[REFERENCING OLD AS o NEW AS n]

[FOR EACH ROW]

WHEN (condition)

DECLARE

Declaration-statements

BEGIN

Executable-statements

EXCEPTION

Exception-handling-statements

END;

Giải thích:

* CREATE [OR REPLACE ] TRIGGER trigger\_name: tạo hoặc thay thế một trigger đã tồn tại thành trigger\_name.
* {BEFORE | AFTER | INSTEAD OF }: cái này chỉ định khi nào trigger được thực thi. INSTEAD OF được sử dụng để tạo trigger trên view.
* {INSERT [OR] | UPDATE [OR] | DELETE}: chỉ định cụ thể lệnh nào thuộc loại lệnh DML được thực thi.
* [OF col\_name]: chỉ định cụ thể cột nào sẽ được cập nhật
* ON table\_name: chỉ định trigger sẽ được thực thi trên bảng nào.
* [REFERENCING OLD AS o NEW AS n]: cho phép dùng giá trị new hay old cho các lệnh DML, như là Insert, Update, Delete.
* [FOR EACH ROW]: cho biết cụ thể trigger sẽ thực thi trên từng dòng.
* WHEN (condition): cho biết dòng đó có điều kiện gì thì trigger mới thực thi được

### *Danh sách Trigger*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Thao tác | Bảng | Nội dung |
| 1 | phong\_dt | INSERT | phieuthue | Kiểm phòng đã được đạt trong khoảng thời gian đó hay chưa |
| 2 | sk\_toi\_da | INSERT | chitietpt | không được đăng ký thuê phòng quá số khách tối đa của phòng |
| 3 | khoa\_pt | UPDATE | phieuthue | Không được sửa khi phiếu thuê đã thanh toán |
| 4 | khoa\_dkdv\_ins | INSERT | dangkydv | Không được đăng ký dịch vụ vào phiếu thuê đã thanh toán |
| 5 | khoa\_dkdv\_del\_upd | DELETE or UPDATE | dangkydv | Không được sửa hay xóa các đăng ký dịch vụ của hóa phiếu thuê đã thanh toán |
| 6 | khoa\_dkkm\_ins | INSERT | dangkykm | Không được đăng ký khuyến mãi vào phiếu thuê đã thanh toán |
| 7 | khoa\_dkkm\_del\_upd | DELETE or UPDATE | dangkykm | Không được sửa hay xóa các đăng ký khuyến mãi của hóa phiếu thuê đã thanh toán |
| 8 | ngay\_dkdv | INSERT | dangkydv | Kiểm tra ngày đăng ký dịch vụ trong khoảng thời đang thuê phòng |
| 9 | ngay\_dkkm | INSERT | dangkykm | Kiểm tra ngày đăng ký khuyến mãi trong khoảng thời đang thuê phòng |
| 10 | gia\_dkdv | INSERT | dangkydv | Lấy giá của dịch vụ tại thời điểm đăng ký |
| 11 | tile\_dkkm | INSERT | dangkykm | Lấy giá của khuyến mãi tại thời điểm đăng ký |
| 12 | adkm\_thempt | INSERT | phieuthue | áp dụng các khuyến mãi tự động vào các phiếu thuê |
| 13 | khoa\_ctpt\_ins | INSERT | chitietpt | Không được sửa đổi chi tiết của phiếu thuê khi đã thanh toán |
| 14 | khoa\_ctpt\_del | DELETE | chitietpt | Không được sửa đổi chi tiết của phiếu thuê khi đã thanh toán |

### *Mô tả một số trigger*

#### Kiểm tra ngày đặt thuê phòng

Tên trigger: phong\_dt

Thao tác: insert

Trên bảng: phieuthue

Các bước:

1. Liệt kê các phòng có thể thuê trong khoảng thời gian ngbd và ngktdk
2. Kiểm tra phòng cần đặt có trong danh sách hay không (p.sophong = :new.sophong)
3. Đếm số dòng trong bảng lưu vào biến v\_rcount
4. Nếu biến v\_rcount = 0 thì phòng đó không đc thuê trong khoảng thời gian đấy và báo lỗi

Mã PL/SQL:

CREATE OR REPLACE TRIGGER phong\_dt BEFORE

INSERT ON phieuthue

FOR EACH ROW

DECLARE

v\_rcount number;

BEGIN

select count(\*) into v\_rcount

from do\_an.phong p

where p.sophong

not in (select distinct pt.sophong

from do\_an.phieuthue pt

where ngkt is null

and ((:new.ngbd between pt.ngbd and pt.ngktdk)

or (:new.ngktdk between pt.ngbd and pt.ngktdk)

or (pt.ngbd between :new.ngbd and :new.ngktdk)

or (pt.ngktdk between :new.ngbd and :new.ngktdk)))

and p.sophong = :new.sophong;

IF v\_rcount = 0 THEN

raise\_application\_error(-20011, 'Phòng đang được thuê');

END IF;

END;

#### Áp dụng khuyến mãi tự động

Tên trigger: adkm\_thempt

Trên bảng: phieuthue

Các bước:

1. Con trỏ lấy các khuyến mãi tự động
2. Dò từng khuyến mãi nếu phiếu thuê có ngày thuê (ngbd) trong khoảng thời gian khuyến mãi áp dụng khuyến mãi đó lên phiếu thuê (insert vào bảng dkkm)

Mã PL/SQL:

create or replace TRIGGER adkm\_thempt AFTER

INSERT ON phieuthue

FOR EACH ROW

DECLARE

CURSOR c\_dskm IS

SELECT

MAKM, NGBD, NGKT, TILE, LOAIAD

FROM

KHUYENMAI

WHERE

LOAIKM = 1;

r\_dskm c\_dskm%rowtype;

BEGIN

OPEN c\_dskm;

LOOP BEGIN

FETCH c\_dskm INTO r\_dskm;

EXIT WHEN c\_dskm%found = false;

IF :new.ngbd >= r\_dskm.ngbd and :new.ngbd <= r\_dskm.ngkt THEN

INSERT INTO dangkykm(mapt,makm,tile,loaiad)

VALUES (:new.mapt, r\_dskm.makm, r\_dskm.tile,r\_dskm.loaiad);

END IF;

END;

END LOOP;

CLOSE c\_dskm;

END;

## Stored Procedure

### *Stored Procedure trong Oracle*

Stored procedure là một hoặc một tập các lệnh SQL được biên dịch để thực hiện cùng lúc mỗi khi được gọi đến. 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, bởi vì thay vì gửi nhiều câu lệnh dài thì ta chỉ cần gọi tới một thủ tục và trong thủ tục này sẽ thực hiện nhiều câu lệnh SQL.

Cú pháp:

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

{IS | AS}

BEGIN

< procedure\_body >

END procedure\_name;

Giải thích:

* procedure\_name : tên của procedure
* [OR REPLACE]: thay thế những procedure đang tồn tại.
* IN: cho phép nhập tham số. OUT: giá trị trả về cụ thể của procedure.
* < procedure\_body >: chứa những câu lệnh của procedure.
* AS được sử dụng thay vì IS để tạo ra những procedure độc lập.

### *Transaction trong Oracle*

#### Định nghĩa

Transaction là một đơn vị làm việc có tính chất logic. Chứa một hoặc nhiều câu lệnh SQL

Và có tính chất ACID:

* Atomicity (tính nguyên tố)
* Consistency (tinh nhất quán)
* Isolation (tính cô lập)
* Durability (tính bền vững)

#### Cấu trúc transaction

Gồm một hay nhiều cấu lệnh DML (như là INSERT, DELETE, UPDATE) và chỉ một câu lệnh DDL (như là CREATE, DROP, ALTER,…)

Kết thúc transaction là 1 lệnh COMMIT hoặc là ROLLBACK không có mệnh đề SAVEPOINT (ROLLBACK TO SAVEPOINT <tên SAVEPOINT>)

Trong cơ sở dữ liệu oracle khi ta thực thi mỗi câu lệnh DDL thì csdl sẽ COMMIT các lệnh DML trước khi thực hiện câu lệnh DDL. Nếu có câu lệnh DDL trong 1 transaction gồm chuỗi các lệnh DML thì đầu tiền csdl Oracle COMMIT transaction sau đó chạy và COMMIT lệnh DDL như transaction mới gồm 1 câu lệnh DDL

User ngắt kết nối đến hệ quản trị đột ngột, transaction sẽ tự động commit.

Các ứng dụng đang kết nối đến hệ quản trị bị dừng đột ngột, transaction sẽ tự động rollback

#### Transaction control

Gồm các lệnh để quản lý sự thay đổi của DML lên database, gồm 1 số lệnh chính:

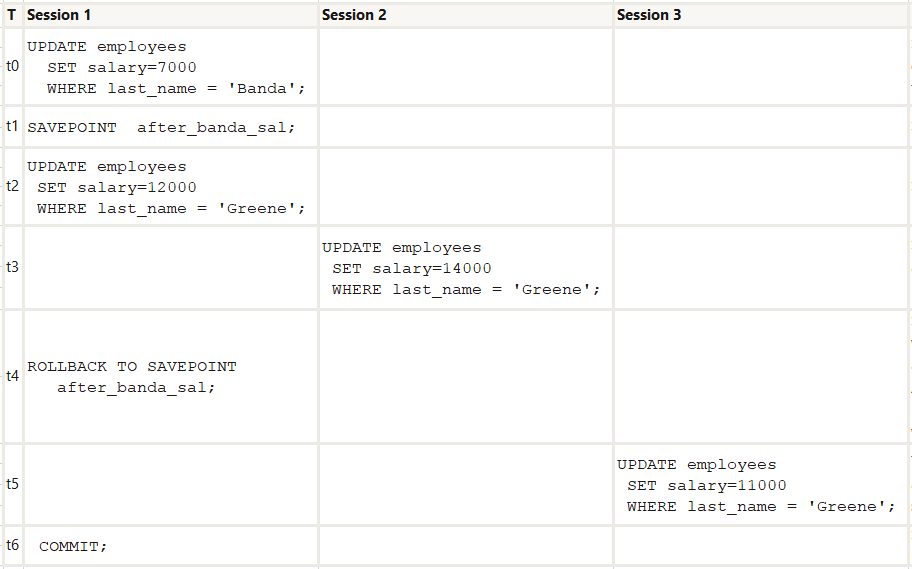
* SAVEPOINT: Xác định 1 điểm trong transaction để rollback về khi có sự cố.
* COMMIT: Kết thúc transaction, lưu thay đổi vĩnh viễn, xóa tất cả SAVEPOINT, mở transaction locks.
* ROLLBACK: phục hồi lại dữ liệu trước khi thay đổi.
* ROLLBACK:
  + Hoàn tác mọi thay đổi.
  + Mở tất cả khóa.
  + Xóa toàn bộ savepoints.
  + Kết thúc transaction
* ROLLBACK TO SAVEPOINT:
  + Đưa csdl về trạng thái trước điểm savepoint (rollback các câu lệnh sau điểm savepoint)
  + Mở các khóa dữ liệu sau điểm savepoint và giữ các khóa phía trước



*Hình 4.1: control transaction*

#### Enqueues transaction

Transaction đang đợi tài nguyên bị khóa sẽ bị block, nó sẽ xếp vào hàng đợi của transaction đang giữ tài nguyên đó. Transaction đang giữ tài nguyên phải commit hoặc rollback để transaction bị khóa tiếp tục thực thi



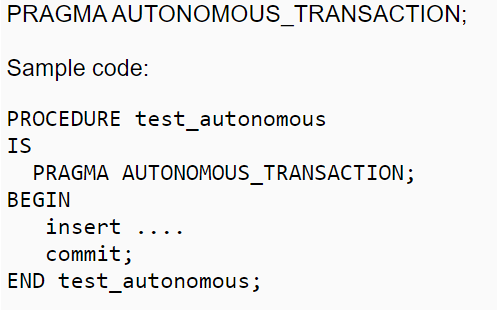
*Hình 4.2: Enqueues Transaction*

* Thời điểm t3: session 2 muốn update dòng dữ liệu của ‘Greene’ nhưng không được vì transaction đang giữ khóa trên dòng dữ liệu ‘Greene’
* Thời điểm t4: session 1 đã mở khóa dòng dữ liệu ‘Greene’ nhưng session 2 vẫn bị khóa do session 2 phải đợi session 1 kết thúc
* Thời điểm t5: dòng dữ liệu ‘Greene’ đã được mở khóa nên session đc update và khóa trên dòng dữ liệu ‘Greene’
* Thời điểm t6: session 1 commit và kết thúc, session 2 sẽ được thêm vào hàng đợi sau session 3 vì session 3 đang giữ khóa

#### Autonomous Transactions

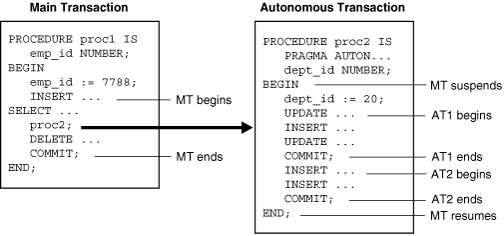
An autonomous transaction là một giao dịch độc lập có thể được gọi từ một main transaction khác

Ta có thể dừng một transaction đang chạy để thực thi các câu lệch SQL khác và commit hoặc rollback các câu lệch SQL, sau đó trở về với transaction đang được thực thi



Có thể là trigger, functrion,…

Hình 4.3: cấu trúc của một transaction autonomous



*Hình 4.4: Gọi transaction autonomous từ transaction khác (Main transaction)*

* Một transaction autonomous không thể thấy các kết quả chưa commit ở main transaction và không chia sẽ khóa và tài nguyên với main transaction
* Thay đổi trong transaction autonomous được nhìn thấy bởi các transaction khác chưa kết thúc. Do đó người dùng có thể truy cập vào các thông tin đã được cập nhật bởi trainsaction autonomous mà không phải chờ main transaction kết thúc
* Transaction autonomous có thể được gọi từ transaction autonomous khác. Không có giới hạn về số lần gọi transaction
* Nếu main transaction có kết thúc bằng lệnh rollback thì cũng không ảnh hưởng đến kết quả của transaction autonomous được gọi và ngược lại

PROCEDURE mainTran IS

v\_salary EMPLOYEES.salary%TYPE;

BEGIN

SELECT salary into v\_salary

FROM EMPLOYEES

WHERE employee\_id = 110;

UPDATE EMPLOYEES

SET salary = 11000

WHERE employee\_id = 110;

autonomousTran();

COMMIT;

END;

PROCEDURE autonomousTran IS

PRAGMA AUTONOMOUS\_TRANSACTION;

v\_salary EMPLOYEES.salary%TYPE;

BEGIN

SELECT salary into v\_salary

FROM EMPLOYEES

WHERE employee\_id = v\_emp\_id;

UPDATE EMPLOYEES

SET salary = 12000

WHERE employee\_id = 120;

COMMIT;

END;



*Hình 4.5: Mô tả autonomous transaction*

* Từ thời điểm t0 đến t2 ở session 1 đã gọi thủ tục mainTran tiến hành kiểm tra salary của nhân viên 110 kết quả là 8000, sau đó cập nhật salarycủa nhân viên 110 lên 11000.
* Thời điểm t3 đến t6 thì thủ tục mainTran đã gọi thủ tục autonomousTran (là một autonomous transaction) tiến hành kiểm tra salary nhân viên 110 được kết quả là 8000 (vì mainTran chưa commit nên giá trị salary ở đây sẽ là 8000), tiếp đó cập nhật salary của nhân viên 120 lên 12000 và commit kết thúc autonomousTran.
* Thời điểm t7 session kiểm tra salary của 2 nhân viên 110 và 120 cho kết quả 8000 và 12000 (vì giá trị salary của nhân viên 120 đã được thủ tục autonomousTran cập nhật và commit).
* Thời điểm t8 mainTran commit và kết thúc và ở t9 khi session kiểm tra salary của 2 nhân viên 110 và 120 sẽ cho ra kết quả 11000 và 12000.
* Nếu t9 là lệnh rollback thì khi kiểm tra lại salary của 2 nhân viên 110 và 120 sẽ cho ra kết quả 8000 và 12000 do mainTran có commit hay rolllback cũng sẽ không ảnh hưởng đến kết quả của autonomousTran

### *Danh sách các Procedure*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Tham số vào | Tham số ra | Ý nghĩa |
| 1 | xoa\_ctpt | v\_ma chitietpt.mapt%type,  v\_cm chitietpt.cmnd%type |  | Xóa khách hàng khỏi phiếu thuê |
| 2 | them\_ctpt | v\_ma chitietpt.mapt%type,  v\_cm chitietpt.cmnd%type |  | Thêm khách hàng vào phiếu thuê |
| 3 | tinh\_thanhtien | v\_mapt phieuthue.mapt%type, v\_ngkt phieuthue.ngkt%type |  | Tính thành tiền và kết thúc  hóa đơn |
| 4 | them\_khachhang | v\_cmnd khachhang.cmnd%type, v\_hoten khachhang.hoten%type, v\_loaikhach khachhang.loaikhach%type, v\_diachi khachhang.diachi%type |  | Thêm khách hàng, tạo tài khoản  mật khẩu cho khách hàng |
| 5 | xoa\_khachhang | v\_cmnd khachhang.cmnd%type |  | Xóa khách hàng khỏi hệ thống |
| 6 | doi\_mk\_khachhang | v\_cmnd nguoidung.cmnd%type, v\_mk nguoidung.matkhau%type |  | Đổi mật khẩu của khách hàng |
| 7 | doi\_mk\_quanli | v\_mk quanli.matkhau%type |  | Đổi mật khẩu của quản lí |
| 8 | huy\_pt | v\_mapt phieuthue.mapt%type |  | Hủy phiếu thuê |
| 9 | them\_pt | v\_sp phong.sophong%type, v\_cm khachhang.cmnd%type, v\_ngbd phieuthue.ngbd%type, v\_ngktdk phieuthue.ngbd%type | phieuthue.mapt%type | Thêm một phiếu thuê mới |
| 10 | xem\_thanhtien | v\_mapt phieuthue.mapt%type, v\_ngkt phieuthue.ngkt%type | phieuthue.thanhtien%type | Xem thành tiền của phiếu thuê |

### *Mô tả một số Stored Procedure*

#### Thêm khách hàng vào phiếu thuê

Tên: them\_ctpt

Nội dung: Thêm khách hàng vào phiếu thuê

Tham số vào: v\_ma, v\_cm

Tham số ra: không có

Các bước:

1. Thêm khách hàng có chứng minh (v\_cm) vào phiếu thuê có mã phiếu thuê (v\_ma)
2. Lấy đơn giá của phòng lưu vào biến v\_dg
3. Lấy hệ số cao nhất của các khách hàng đăng ký thuê phòng lưu vào biến v\_hs
4. Tính lại đơn giá của phòng (v\_dg := (v\_dg\*v\_hs)/100)
5. Cập nhật lại đơn giá phòng của phiếu thuê

Mã PL/SQL:

create or replace PROCEDURE THEM\_CTPT(

v\_ma chitietpt.mapt%type,

v\_cm chitietpt.cmnd%type)

AS

v\_dg phieuthue.giaphong%TYPE;

v\_hs loaikhach.heso%TYPE;

BEGIN

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

INSERT INTO chitietpt values (v\_ma, v\_cm);

SELECT lp.dongia INTO v\_dg

FROM (phong p

JOIN loaiphong lp on p.loaiphong = lp.loaiphong)

JOIN phieuthue pt on pt.sophong = p.sophong

where pt.mapt = v\_ma;

SELECT lk.heso INTO v\_hs

FROM (chitietpt c

JOIN khachhang k ON c.cmnd = k.cmnd )

JOIN loaikhach lk ON lk.loaikhach = k.loaikhach

WHERE c.mapt = v\_ma

ORDER BY lk.heso DESC

FETCH FIRST 1 ROW ONLY;

v\_dg := (v\_dg \* v\_hs) / 100;

UPDATE phieuthue

SET giaphong = v\_dg

WHERE mapt = v\_ma;

COMMIT;

END;

#### Thủ tục thanh toán phiếu thuê

Tên: tinh\_thanhtien

Nội dung: Tình thành tiền cập nhật ngày kết thúc, thành tiền của hóa đơn

Tham số vào: v\_mapt, v\_ngkt

Tham số ra: không có

Các bước:

1. Tính khoảng thời gian thuê phòng của làm tròn 1 chữ số lưu vào biến v\_ng
2. Tính tổng số tiền dịch vụ phải trả lưu vào biến v\_dv
3. Tính tổng tỉ lệ khuyến mãi trừ trên tổng hóa đơn lưu vào biến v\_kmhd
4. Tính tổng tỉ lệ khuyến mãi trừ trên tiền thuê phòng vào biến v\_kmtp
5. Lấy đơn giá của phòng lưu vào biến v\_dg
6. Tính tiền thuê phòng lưu vào biến v\_tt (ít nhất là 200000)
7. Tính lại thành tiền của phòng
8. Cập nhật thành tiền và ngày kết thúc của phiếu thuê

Mã PL/SQL:

create or replace PROCEDURE tinh\_thanhtien (

v\_mapt phieuthue.mapt%TYPE,

v\_ngkt phieuthue.ngkt%TYPE

) IS

v\_ng NUMBER := 0;

v\_tt phieuthue.thanhtien%TYPE := 0;

v\_dv dichvu.gia%TYPE := 0;

v\_kmhd khuyenmai.tile%TYPE := 0;

v\_kmtp khuyenmai.tile%TYPE := 0;

v\_dg phieuthue.giaphong%TYPE := 0;

BEGIN

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

SELECT round((months\_between(v\_ngkt, p.ngbd)) \* 31, 1)

INTO v\_ng

FROM phieuthue p

WHERE p.mapt = v\_mapt;

SELECT (CASE

WHEN SUM(gia) IS NULL THEN 0

ELSE SUM(gia)

END) INTO v\_dv

FROM dangkydv

WHERE mapt = v\_mapt;

SELECT (CASE

WHEN SUM(tile) IS NULL THEN 0

ELSE SUM(tile)

END) INTO v\_kmhd

FROM dangkykm

WHERE mapt = v\_mapt and loaiad = 0;

SELECT (CASE

WHEN SUM(tile) IS NULL THEN 0

ELSE SUM(tile)

END) INTO v\_kmtp

FROM dangkykm

WHERE mapt = v\_mapt and loaiad = 1;

SELECT giaphong INTO v\_dg

FROM phieuthue

WHERE mapt = v\_mapt;

v\_tt := v\_dg \* v\_ng;

IF v\_tt < 200000 THEN

v\_tt := 200000;

END IF;

v\_tt := v\_tt + v\_dv - ( v\_kmtp \* v\_tt / 100 );

v\_tt := v\_tt - ( v\_tt \* v\_kmhd / 100);

UPDATE phieuthue

SET thanhtien = v\_tt AND ngkt = v\_ngkt

WHERE mapt = v\_mapt;

COMMIT;

END;

# Xử lý truy xuất đồng thời

## Các mức cô lập trong Oracle

### *Read Committed*

Trong Read Committed , mọi truy vấn được thực hiện bởi một transaction chỉ nhìn thấy dữ liệu đã được commit. Mức cô lập này là mặc định. Nó phù hợp với môi trường cơ sở dữ liệu trong đó ít transaction có khả năng xung đột.

Xung đột ghi trong Read Committed:

* Trong Read Committed, một xung đột ghi xảy ra khi transaction cố gắng thay đổi một dòng đã được cập nhật bởi một transaction khác mà chưa được commit hay rollback.
* Transaction ngăn chặn sửa đổi dòng được gọi là blocking transaction. Transaction Read Committed chờ blocking transaction kết thúc và giải phóng khóa rồi transaction Read Committed mới lấy được khóa và thực hiện thay đổi.

### *Serializable*

Ở mức cô lập tuần tự (serializable isolation level), transaction chỉ thấy những thay đổi đã được commit trước thời điểm transaction có mức cô lập tuần tự này bắt đầu và các thay đổi được thực hiện bởi chính transaction này.

Một transaction tuần tự hoạt động trong một môi trường chỉ một mình nó như thể không có người dùng nào khác đang sửa đổi dữ liệu trong cơ sở dữ liệu. mức cô lập tuần tự phù hợp với các môi trường:

* Với cơ sở dữ liệu lớn và các transaction ngắn chỉ cập nhật một vài dòng.
* Trường hợp có 2 transaction đồng thời sửa đổi cùng 1 dòng là tương đối thấp.
* Trường hợp các transaction tương đối dài chủ yếu chỉ đọc.

Ở mức cô lập tuần tự, Bất kỳ dòng nào được đọc bởi transaction tuần tự được đảm bảo giống nhau khi đọc lại. Bất kỳ truy vấn nào cũng được đảm bảo trả về cùng kết quả trong suất thời gian transaction đó hoạt động, do đó, những thay đổi được thực hiện bởi transaction khác sẽ không hiển thị đối với truy vấn của transaction tuần tự bất kể những thay đổi của transaction khác đã chạy được bao lâu. Transaction tuần tự (serializable) ngăn ngừa được Dirty reads, Nonrepeatable reads, Phantom reads.

Trong cơ sở dữ liệu Oracle, nếu transaction Tuần tự thực hiện thay đổi 1 dòng nào đó mà dòng này đã được thay đổi bởi một transaction khác đã commit sau khi transaction tuần bắt đầu thực thi thì sẽ xuất hiện 1 lỗi đó là :

ORA-08177: Cannot serialize access for this transaction

### *Read-only*

Mức cô lập Read-only tương tự như mức cô lập Serializable, nhưng các transaction có mức cô lập Read-only chỉ được đọc dữ liệu, không được sửa đổi dữ liệu trong transaction (trừ trường hợp là người dùng là SYS)

## Cơ chế khóa

“Khóa” là một cơ chế giúp ngăn chặn các tương tác “có hại” đến hệ thống. Tương tác gọi là “có hại” khi nó cập nhật dữ liệu sai hoặc thay đổi sai cấu trúc dữ liệu, khiến dữ liệu không nhất quán giữa, thường xảy ra khi nhiều transaction cùng thao tác lên 1 đơn vị dữ liệu. Vì vậy, cơ chế khóa đóng vai trò cực kì quan trọng để đảm bảo cơ sở dữ liệu đồng thời và nhất quán

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

Cơ chế khóa tác động đến sự tương tác giữa việc đọc và ghi dữ liệu

* Một dòng dữ liệu chỉ bị khóa khi nó đang được thay đổi bởi việc hành động ghi. Khi có 1 câu truy vấn cập nhật dữ liệu của 1 dòng, hệ quản trị sẽ khóa dòng đó lại
* Một hành động ghi lên 1 dòng dữ liệu sẽ ngăn chặn các hành động khác cùng đồng thời ghi lên dòng dữ liệu đó
* Hành động đọc dữ liệu không ngăn cản hành động ghi: Vì hành động đọc không khóa đơn vị dữ liệu nó đang đọc, nên hành động ghi có thể thao tác lên đơn vị dữ liệu này. Chỉ có một trường hợp ngoại lệ là câu lệnh Select … for update sẽ lock đơn vị dữ liệu nó đang đọc
* Hành động ghi dữ liệu không ngăn cản hành động đọc: Khi một đơn vị dữ liệu đang được thao tác bởi hành động ghi, hệ quản trị sẽ sử dụng phiên bản trước đó của đơn vị dữ liệu để trả về cho hành động đọc

Cơ chế khóa phải thỏa các yêu cầu quan trọng của cơ sở dữ liệu: tính nhất quán và tính toàn vẹn.

Hệ quản trị Oracle đảm bảo dữ liệu đồng thời, nhất quán và toàn vẹn giữa các transactions thông qua cơ chế khóa. Cơ chế khóa diễn ra tự động và không cần sự can thiệp của người dùng.

Các chế độ khóa

* Hệ quản trị Oracle tự động sử dụng chế độ khóa ở cấp cần thiết thấp nhất để đảm bảo khả năng truy cập đồng thời của nhiều tiến trình lên cùng đơn vị dữ liệu
* Sử dụng chế độ khóa cấp càng thấp, dữ liệu có thể được truy cập bởi càng nhiều user. Ngược lại, nếu sử dụng mức khóa càng cao, dữ liệu càng hạn chế người truy cập
* Hệ quản trị Oracle sử dụng 2 chế độ khóa chính cho cơ sở dữ liệu nhiều người dùng
  + 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 lên đơn vị dữ liệu này cho đến khi transaction đang giữ exclusive lock nhả khóa
  + Share lock mode: Chế độ này cho phép chia sẻ cùng 1 đơn vị dữ liệu. Nhiều giao tác đọc có thể chia sẻ dữ liệu cho nhau, mỗi giao tác giữ một share lock để ngăn hành động ghi vào cùng thời điểm của các giao tác ghi. Nhiều giao tác có thể được cấp share lock trên cùng đơn vị dữ liệu

## Deadlock

Deadlock là tình huống khi một hay nhiều người dùng đang chờ một đơn vị dữ liệu mà đang bị các người dùng khác cũng đang chờ đơn vị dữ liệu khác block.

Hệ quản trị Oracle tự động phát hiện và xử lý deadlock bẳng cách rollback một transaction gây deadlock, giải phóng 1 dòng đang bị khóa. Cơ sở dữ liệu sẽ trả về thông báo cho transaction bị rollback

Mô tả trường hợp deadlock:

Dữ liệu ban đầu



Bảng mô tả trường hợp



Sau khi chờ nhau một khoảng thời gian thì một trong hai session sẽ xác định deadlock và rollback transaction, trong khi đó thì transaction còn lại sẽ thực thi thành công. Lỗi của deadlock sẽ được hiển thi như sau:

ERROR at line 1:

ORA-00060: deadlock detected while waiting for resource

ORA-06512: at line 16

## Thiết lập mức cô lập

Khi thực hiện các transaction có các mức cô lập khác nhau ta cần biết đến cách thiết lập mức cô lập ở trên transaction có thể gom thành thủ tục hoặc các transacion không phải thủ tục

Đối với transaction là 1 thủ tục (procedure, function, trigger) ta chỉ cần thêm câu lệnh: SET TRANSACTION ISOLATION LEVEL <type of isolation level> vào đầu transaction

Đối với transaction không là thủ tục ta cần thiết lập mức cô lập trên liên kết (connection) giữa ứng dụng với cơ sở dữ liệu và ứng dụng ở đây được viết bằng ngôn ngữ java

Trainsaction trong java:

* Các connection được tạo trong java sẽ mặc định một chế độ là auto\_commit. Điều đó có nghĩa khi thực thi từng câu lệnh DML, DDL java sẽ commit sau mỗi câu lệnh hay mỗi câu lệnh là 1 transaction. Vậy để thực thi nhiều câu lệnh và gôm thành 1 transaction t cần tắt chế độ auto\_commit này đi.
* Sau khi tắt chế độ auto\_commit thì làm sao để biết transaction đã kết thúc. Có 3 cách là thực thi câu lệnh (conn.close(); conn.commit(); conn = DriverManager.getConnection – biến conn là Connection) nếu không thực hiện 3 câu lệnh trên thì những thay đổi sẽ không được lưu vào cơ sở dữ liệu.

Thiết đặt mức cô lập trong java:

* Trong Oracle chỉ hỗ trợ 2 mức cô lập là Read Committed và Serializable. Vì vậy trên Java ta chỉ được thiết lập 2 mức cô lập đó, nếu thiết lập mức cô lập khác sẽ bị báo lỗi.
* Ta có 2 câu lệnh để thiết lập mức cô lập trên java là:

conn.setTransactionIsolation(Connection.TRANSACTION\_READ\_COMMITTED);

conn.setTransactionIsolation(Connection.TRANSACTION\_SERIALIZABLE);

## Mô tả đồ án môn học

### *Lost update*

Mô tả tình huống: khi thêm khách hàng vào phiếu thuê hoặc xóa khách hàng khỏi phiếu thuê ta sài thủ tục, và thủ tục đó sẽ tính lại đơn giá phòng của phiếu thuê theo hệ số cao nhất của các khách hàng đăng ký thuê phòng. Khi vừa thêm và vừa xóa khách hàng thì xảy ra trường hợp đơn giá phòng của phiếu thuê được tính và cập nhật vào csdl sai.

Mã PL/SQL:

* Thủ tục them\_ctpt

create or replace PROCEDURE THEM\_CTPT(

v\_ma chitietpt.mapt%type,

v\_cm chitietpt.cmnd%type)

AS

v\_dg phieuthue.giaphong%TYPE;

v\_hs loaikhach.heso%TYPE;

BEGIN

INSERT INTO chitietpt values(v\_ma, v\_cm);

SELECT lp.dongia INTO v\_dg

FROM (phong p

JOIN loaiphong lp on p.loaiphong = lp.loaiphong)

JOIN phieuthue pt on pt.sophong = p.sophong

where pt.mapt = v\_ma;

SELECT lk.heso INTO v\_hs

FROM (chitietpt c

JOIN khachhang k ON c.cmnd = k.cmnd )

JOIN loaikhach lk ON lk.loaikhach = k.loaikhach

WHERE c.mapt = v\_ma

ORDER BY lk.heso DESC

FETCH FIRST 1 ROW ONLY;

v\_dg := (v\_dg \* v\_hs) / 100;

UPDATE phieuthue

SET giaphong = v\_dg

WHERE mapt = v\_ma;

COMMIT;

END;

* Thủ tục xoa\_ctpt

create or replace PROCEDURE XOA\_CTPT(

v\_ma chitietpt.mapt%type,

v\_cm chitietpt.cmnd%type)

AS

v\_dg phieuthue.giaphong%TYPE;

v\_hs loaikhach.heso%TYPE;

v\_sk number;

BEGIN

SELECT COUNT(cmnd) INTO v\_sk

FROM chitietpt

WHERE mapt = v\_ma;

IF v\_sk <= 1 THEN

raise\_application\_error(-20018, 'Không thể xóa khách hàng');

END IF;

DELETE chitietpt where mapt = v\_ma and cmnd = v\_cm;

SELECT lp.dongia INTO v\_dg

FROM (phong p

JOIN loaiphong lp on p.loaiphong = lp.loaiphong)

JOIN phieuthue pt on pt.sophong = p.sophong

where pt.mapt = v\_ma;

SELECT lk.heso INTO v\_hs

FROM (chitietpt c

JOIN khachhang k ON c.cmnd = k.cmnd )

JOIN loaikhach lk ON lk.loaikhach = k.loaikhach

WHERE c.mapt = v\_ma

ORDER BY lk.heso DESC

FETCH FIRST 1 ROW ONLY;

v\_dg := (v\_dg \* v\_hs) / 100;

UPDATE phieuthue

SET giaphong = v\_dg

WHERE mapt = v\_ma;

COMMIT;

END;

Dữ liệu ban đầu:



Mô tả chi tiết:

* Do không nhìn thấy được dữ liệu do transaction khác chưa commit, cụ thể ở đây khi session 2 thực hiện câu select ở thời điểm t7 chỉ thấy được dữ liệu mà session đó thay đổi là chỉ thấy được 1 dòng dữ liệu của khách hàng 01231 có mapt là 1010 trong bảng ctpt và gây ra sai sót dữ liệu của biến v\_hs
* Ở thời điểm t14 sau khi session 2 commit gây mất giá trị commit của session 1 (lost update) và gây sai sót dữ liệu

Bảng mô tả trường hợp



Phương pháp giải quyết: thiết lập mức cô lập cho cả 2 thủ tục them\_ctpt và xoa\_ctpt là SERIALIZABLE

### *Non-repeatable read*

Mô tả tình huống: khi đăng ký dịch vụ cho phiếu thuê sẽ có trigger lấy giá tại thời điểm đó và thêm vào bảng đăng ký dịch vụ (dkdv), nếu có ai update giá của dịch vụ trước khi trigger được thực thi sẽ gây ra sai lệch dữ liệu

Mã PL/SQL:

create or replace TRIGGER gia\_dkdv BEFORE

INSERT ON DANGKYDV

FOR EACH ROW

DECLARE

v\_gia dichvu.gia%TYPE;

BEGIN

SELECT gia INTO v\_gia

FROM dichvu

WHERE :new.madv = madv;

:new.gia := v\_gia;

END;

Mô tả chi tiết:

* Do session có mức cô lập mặc định là read commited nên sẽ đọc được dữ liệu cập nhật từ session 2 gây ra sai lệch dữ liệu (Non-repeatable read)

Bảng mô tả trường hợp:



Phương pháp giải quyết: thiết lập mức cô lập là SERIALIZABLE thay cho mức cô lập mặc định là READ COMMITED. Và thiết mức cô lập này ở trên java.

con.setTransactionIsolation(Connection.TRANSACTION\_SERIALIZABLE);

### *Phantom read*

Mô tả tình huống: khi xóa khách hàng ra khỏi phiếu thuê ta phải kiểm tra trong phiếu thuê có bao nhiêu khách đã đăng ký lớn hơn 1 thì khách hàng sẽ được xóa và cập nhật lại đơn giá phòng. Nếu ta thực thi xóa 2 khách hàng cùng lúc thì sẽ xảy ra lỗi khi 1 trong 2 transaction commit trước khi transaction kia kiểm tra lại hệ số khách hàng.

Mã PL/SQL:

create or replace PROCEDURE XOA\_CTPT(

v\_ma chitietpt.mapt%type,

v\_cm chitietpt.cmnd%type)

AS

v\_dg phieuthue.giaphong%TYPE;

v\_hs loaikhach.heso%TYPE;

v\_sk number;

BEGIN

SELECT COUNT(cmnd) INTO v\_sk

FROM chitietpt

WHERE mapt = v\_ma;

IF v\_sk <= 1 THEN

raise\_application\_error(-20018, 'Không thể xóa khách hàng');

END IF;

DELETE chitietpt where mapt = v\_ma and cmnd = v\_cm;

SELECT lp.dongia INTO v\_dg

FROM (phong p

JOIN loaiphong lp on p.loaiphong = lp.loaiphong)

JOIN phieuthue pt on pt.sophong = p.sophong

where pt.mapt = v\_ma;

SELECT lk.heso INTO v\_hs

FROM (chitietpt c

JOIN khachhang k ON c.cmnd = k.cmnd )

JOIN loaikhach lk ON lk.loaikhach = k.loaikhach

WHERE c.mapt = v\_ma

ORDER BY lk.heso DESC

FETCH FIRST 1 ROW ONLY;

v\_dg := (v\_dg \* v\_hs) / 100;

UPDATE phieuthue

SET giaphong = v\_dg

WHERE mapt = v\_ma;

COMMIT;

END;

Dữ liệu ban đầu:



Bảng mô tả tình huống:



Phương pháp giải quyết: thiết lập mức cô lập cho thủ tục xoa\_ctpt là SERIALIZABLE

# Thiết kế giao diện

## Danh sách các màn hình

1. Đăng nhập

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Đăng nhập | Đăng nhập |

1. Thuê phòng

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Thuê | Thuê phòng và tìm phòng trống |

1. Phòng

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Tìm kiếm | Tìm kiếm phòng |
| 2. | Thêm, sửa | Thêm phòng và sửa thông tin phòng |
| 3. | Quản lí loại phòng | Thêm loại phòng |

1. Hóa đơn

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Hóa đơn | Tra cứu thông tin theo mã phiếu thuê |
| 2. | Xuất HĐ | Xuất hóa đơn |

1. Khách hàng

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Tìm kiếm | Tìm kiếm khách hàng |
| 2. | Thêm, sửa | Thêm khách hàng và sửa thông tin khách hàng |
| 3. | Quản lý loại khách | Thêm loại khách |

1. DV, KM

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Tìm kiếm | Tìm kiếm dịch vụ hoặc khuyến mãi |
| 2. | Thêm, sửa dv | Thêm dịch vụ và sửa thông tin dịch vụ |
| 3. | Thêm, sửa km | Thêm khuyến mãi và sửa thông tin khuyến mãi |

1. Báo cáo

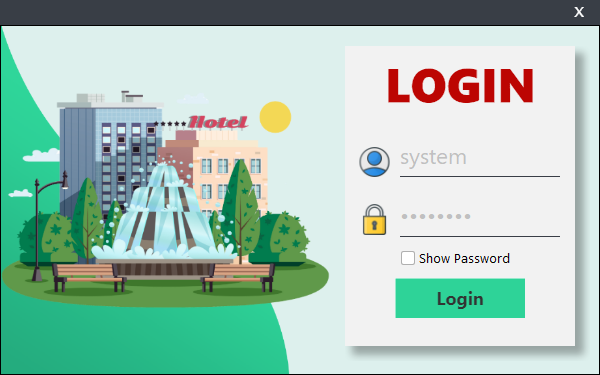
|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Báo cáo | Xem báo cáo |
| 2. | Xuất | Xuất báo cáo |

1. Cài đặt

|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1. | Cài đặt | Đổi mật khẩu |

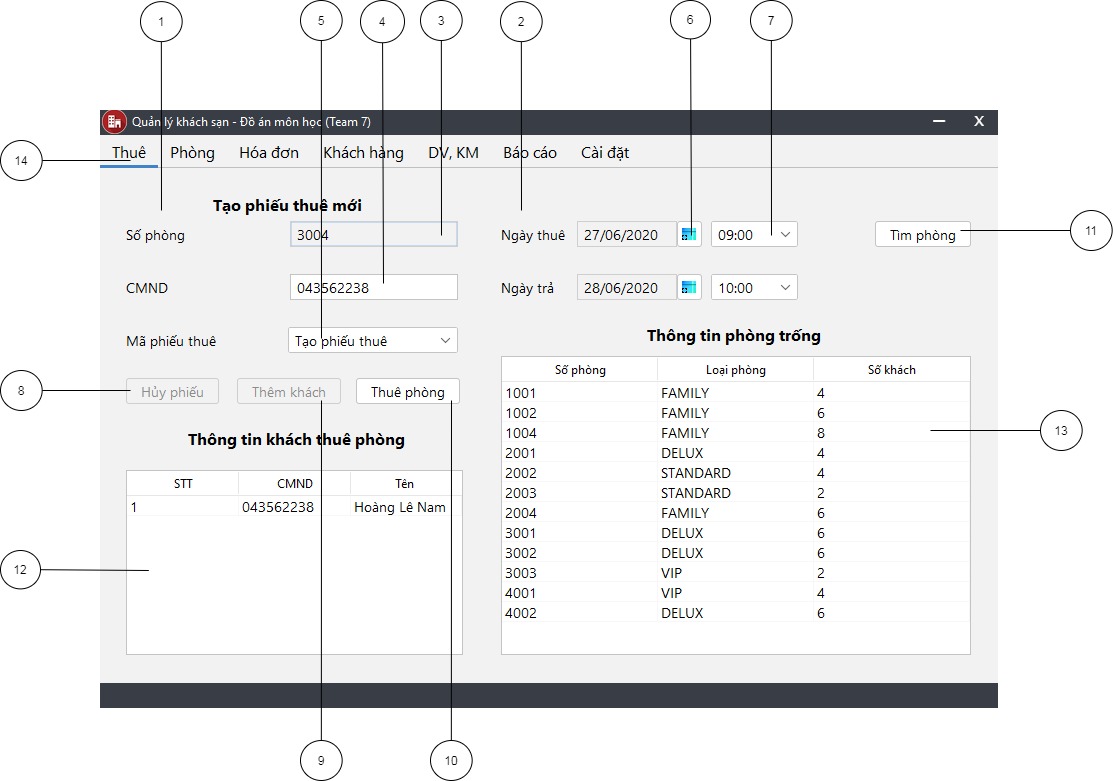
## Mô tả các màn hình

* 1. Đăng nhập



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Hình ảnh | Label | Tạo sự sinh động cho phần mềm |
| 2 | Hình ảnh | Label | Giúp người dùng biết đây là màn hình đăng nhập |
| 3 | Tên tài khoản | TextField | Cho biết thông tin cần nhập và được phép nhập. |
| 4 | Mật khẩu | TextField | Cho biết thông tin cần nhập và được phép nhập. |
| 4 | Lựa chọn | CheckBox | Chọn thì sẽ hiển thị mật khẩu.  Không chọn thì mật khẩu sẽ bị ẩn. |
| 4 | Login | Button | Thực hiện đăng nhập vào phần mềm |

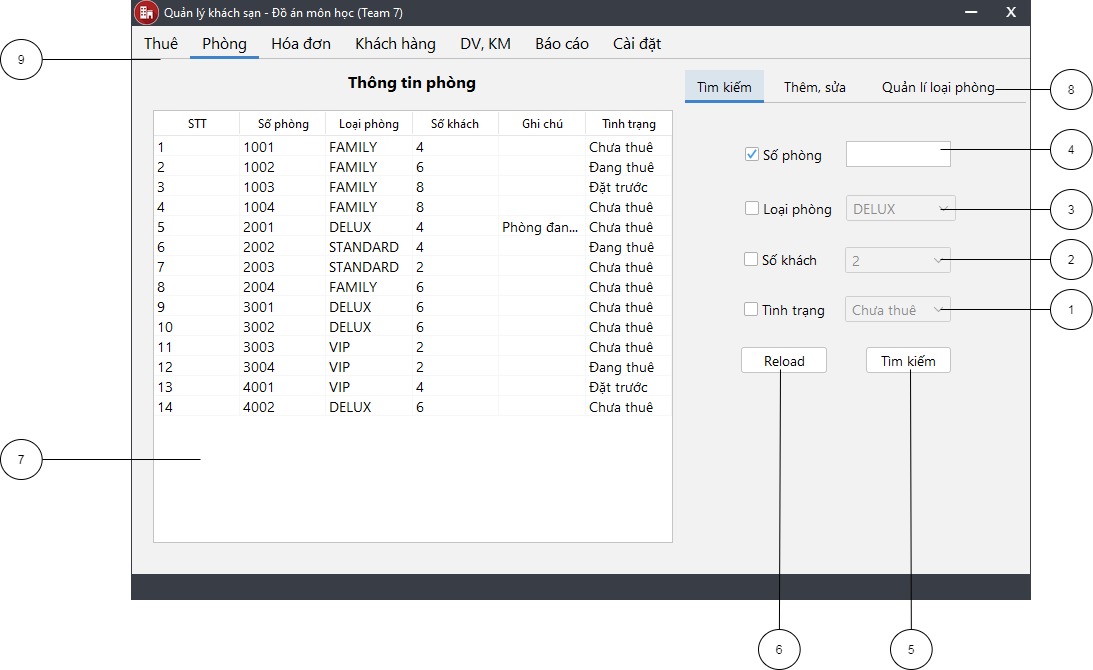
* 1. Thuê phòng



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Số phòng  CMND  Mã phiếu thuê | Label | Hiển thị thông tin cho biết khung nhập kế bên là khung nhập thông tin gì |
| 2 | Ngày thuê  Ngày trả | Label | Hiển thị thông tin cho biết khung nhập kế bên là khung nhập thông tin gì |
| 3 | Khung trống | Textfield | Nhập thông tin mà label kế bên cho biết |
| 4 | Khung trống | Textfield | Nhập thông tin mà label kế bên cho biết |
| 5 | Mã phiếu thuê | Combobox | Chọn mã phiếu thuê |
| 6 | Ngày tháng | Jcalendar | Chọn ngày tháng mà label kế bên cho biết |
| 7 | Thời gian | Combobox | Chọn thời gian mà label kế bên cho biết |
| 8 | Hủy phiếu | Button | Hủy phiếu thuê |
| 9 | Thêm khách | Button | Thực hiện thao tác thêm khách hàng |
| 10 | Thuê phòng | Button | Thực hiện thao tác xác nhận thuê phòng |
| 11 | Tìm phòng | Button | Thực hiện thao tác tìm phòng trống theo ngày giờ đã chọn |
| 12 | Thông tin khách thuê phòng | Table viewer | Hiển thị khách hàng thuê phòng |
| 13 | Thông tin phòng trống | Table viewer | Hiển thị các phòng trống theo ngày giờ đã chọn |
| 14 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

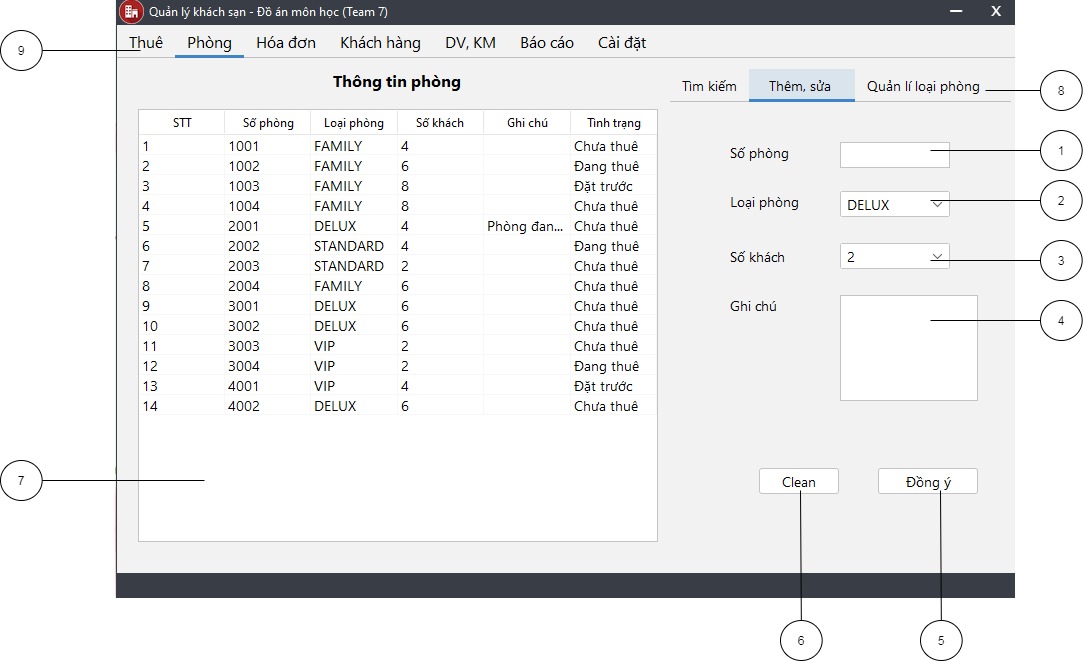
* 1. Phòng

1. Tìm kiếm



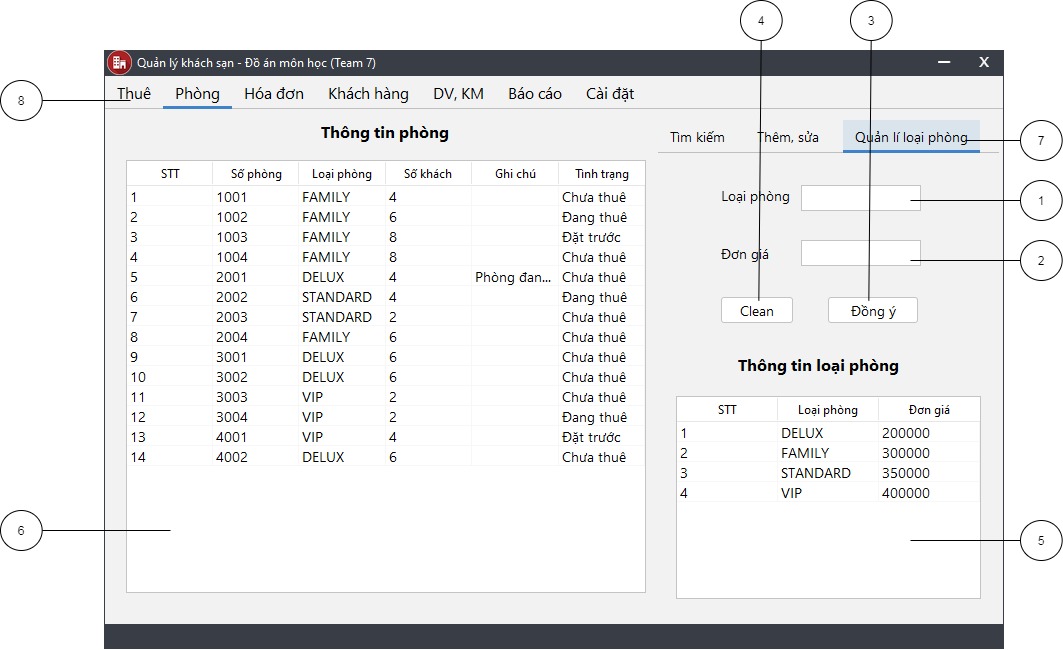
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Tình trạng | Combobox | Lựa chọn tình trạng của phòng |
| 2 | Số khách | Combobox | Lựa chọn số khách tối đa của phòng |
| 3 | Loại phòng | Combobox | Lựa chọn loại phòng |
| 4 | Số phòng | Textfield | Nhập vào số phòng |
| 5 | Tìm kiếm | Button | Thực hiện thao tác tìm kiếm phòng |
| 6 | Reload | Button | Thực hiện thao tác làm mới các thông tin tìm kiếm |
| 7 | Thông tin phòng | Table viewer | Hiển thị thông tin phòng |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

1. Thêm sửa



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Số phòng | Textfield | Nhập vào số phòng |
| 2 | Loại phòng | Combobox | Lựa chọn loại phòng |
| 3 | Số khách | Combobox | Lựa chọn số khách tối đa của phòng |
| 4 | Ghi chú | Textfield | Nhập vào ghi chú cần thiết |
| 5 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm hoặc sửa thông tin phòng |
| 6 | Clean | Button | Thực hiện thao tác làm mới các thông tin thêm sửa thông tin phòng |
| 7 | Thông tin phòng | Table viewer | Hiển thị thông tin phòng |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

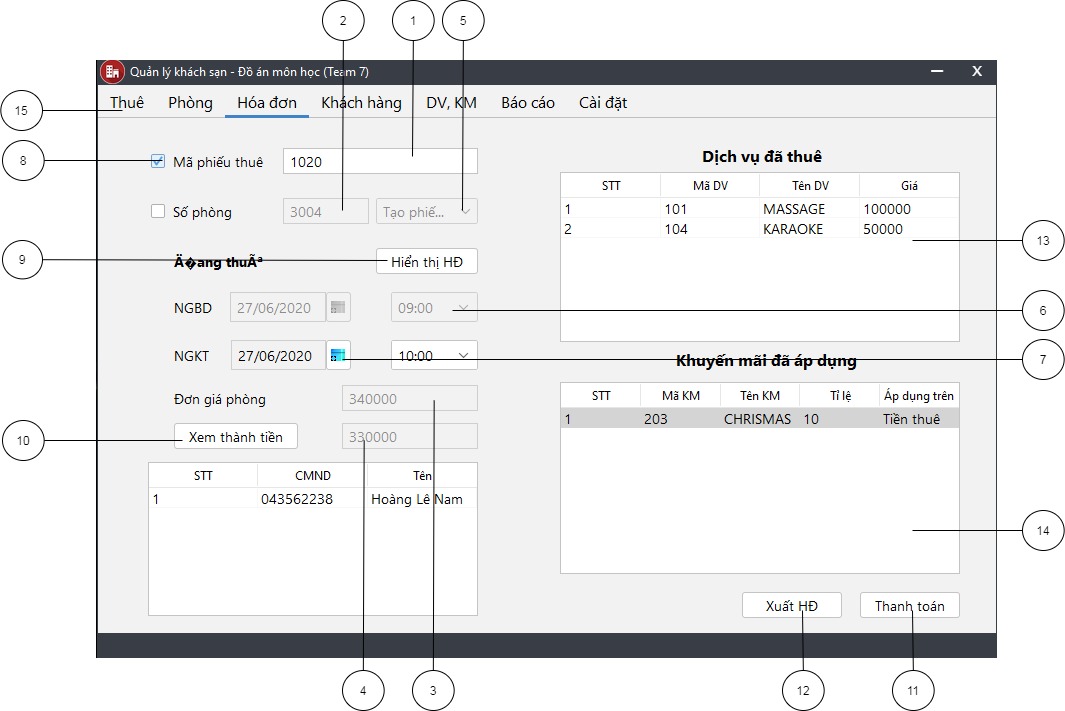
1. Quản lí loại phòng



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Loại phòng | Textfield | Nhập vào tên loại phòng |
| 2 | Đơn giá | Textfield | Nhập vào đơn giá phòng |
| 3 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm loại phòng |
| 4 | Clean | Button | Thực hiện thao tác làm mới các thông tin thêm loại phòng |
| 5 | Thông tin loại phòng | Table viewer | Hiển thị thông tin các loại phòng đang có |
| 6 | Thông tin phòng | Table viewer | Hiển thị thông tin phòng |
| 7 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

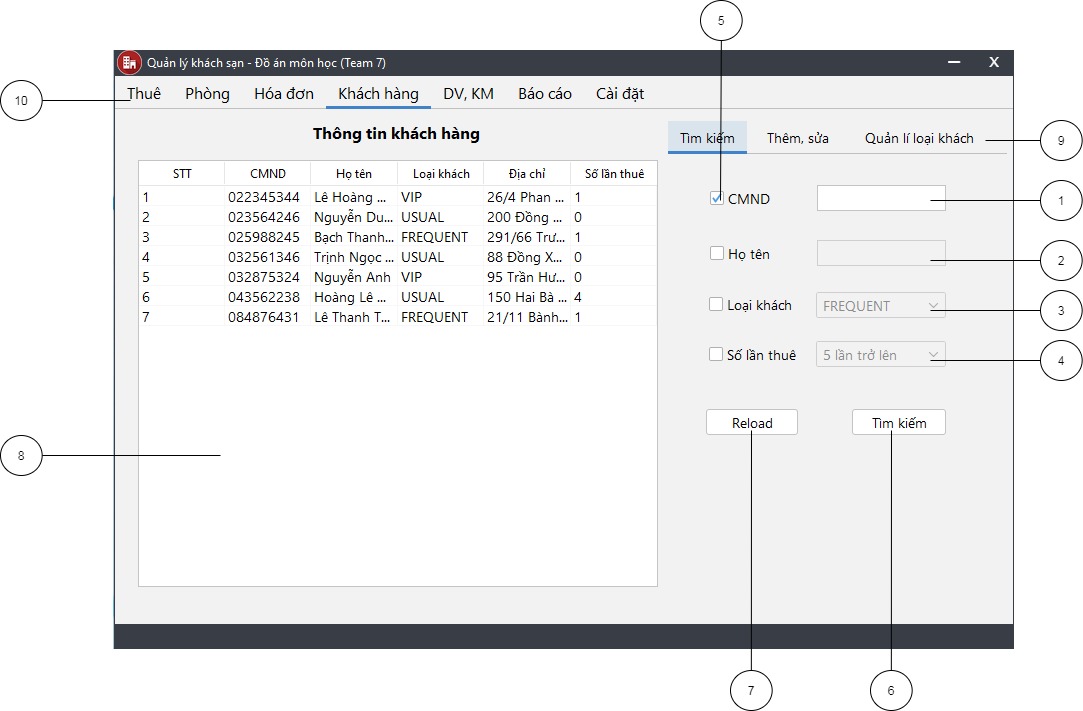
* 1. Hóa đơn

1. Hóa đơn



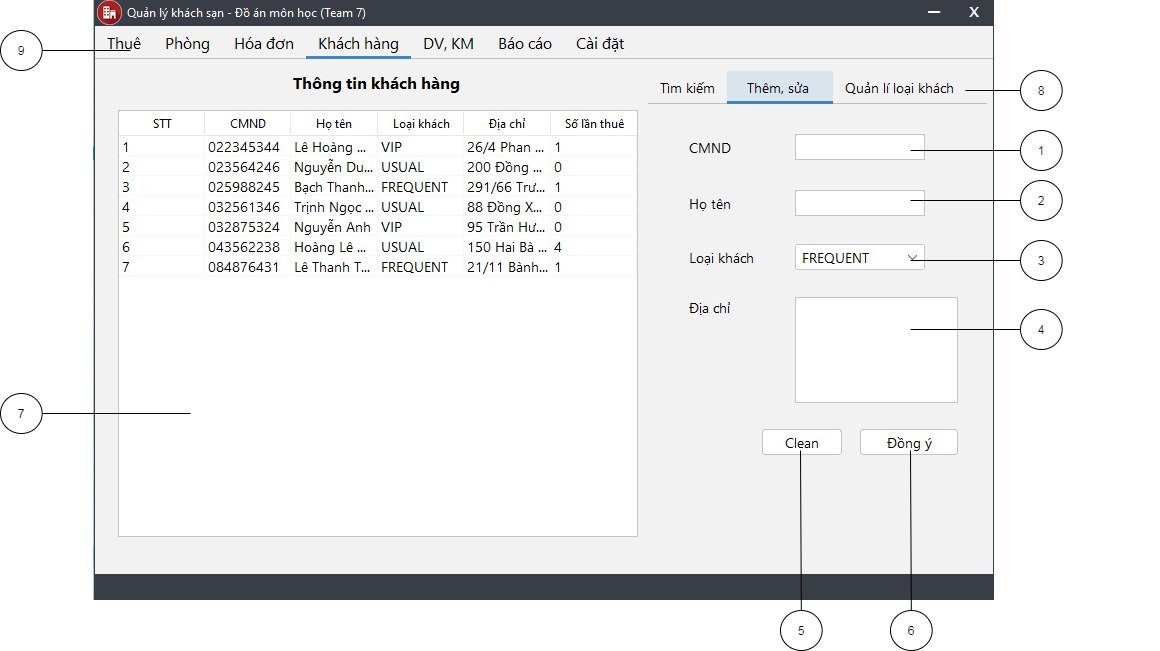
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã phiếu thuê | Textfield | Nhập vào mã phiếu thuê |
| 2 | Số phòng | Textfield | Nhập vào số phòng |
| 3 | Đơn giá phòng | Textfield | Hiển thị đơn giá phòng |
| 4 | Thành tiền | Textfield | Hiển thị thành tiền |
| 5 | Khung chọn | Combobox | Chọn mã phiếu thuê |
| 6 | Thời gian | Combobox | Lựa chọn thời gian |
| 7 | Ngày tháng | Jcalendar | Lựa chọn ngày tháng |
| 8 | Ô chọn | Radiobutton | Lựa chọn nhập số phòng hoặc mã phiếu thuê |
| 9 | Hiển thị HĐ | Button | Thực hiện thao tác hiển thị ra các thông tin khách hàng, ngày thuê, dịch vụ và khuyến mãi |
| 10 | Xem thành tiền | Button | Thực hiên thao tác tính tổng tiền phòng, dịch vụ và khấu hao phần khuyến mãi |
| 11 | Thanh toán | Button | Thực hiện thao tác thanh toán |
| 12 | Xuất HĐ | Button | Thực hiện thao tác xuất hóa đơn |
| 13 | Dịch vụ đã thuê | Table viewer | Hiển thị những dịch vụ mà khách hàng đã sử dụng |
| 14 | Khuyến mãi đã áp dụng | Table viewer | Hiển thị những khuyến mãi mà đã được áp dụng cho phiếu thuê |
| 15 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

1. Xuất hóa đơn
   1. Khách hàng
2. Tìm kiếm



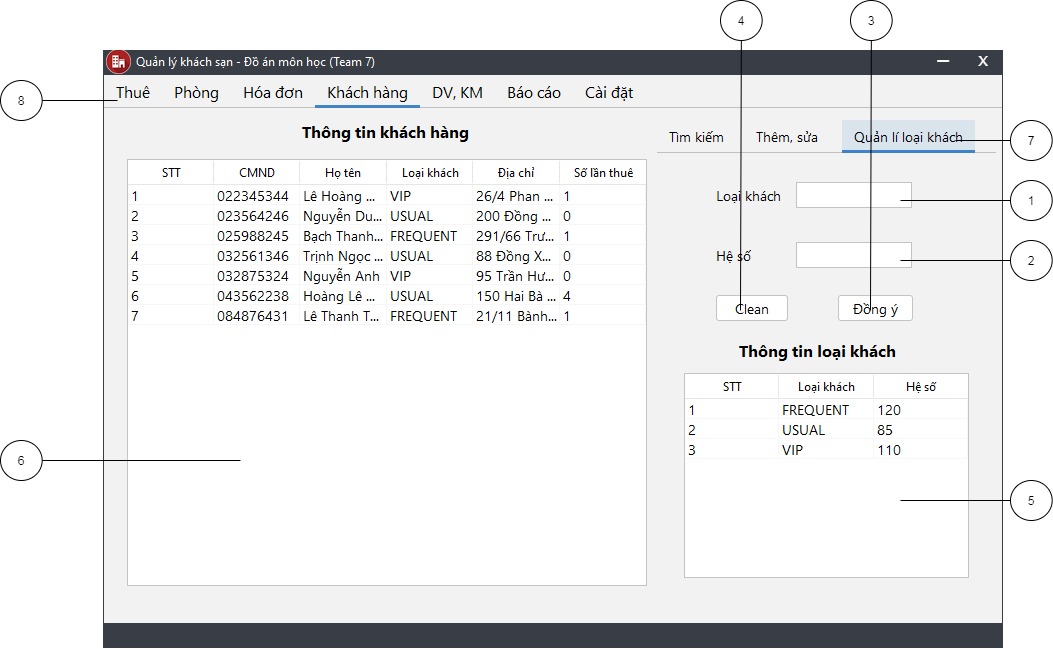
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | CMND | Textfield | Nhập vào chứng minh nhân dân |
| 2 | Họ tên | Textfield | Nhập vào họ tên |
| 3 | Loại khách | Combobox | Lựa chọn loại khác |
| 4 | Số lần thuê | Combobox | Lựa chọn số lần khách đã từng thuê phòng |
| 5 | Ô chọn | Radiobutton | Lựa chọn loại thông tin để tìm kiếm |
| 6 | Tìm kiếm | Button | Thực hiện thao tác tìm kiếm theo thông tin đã nhập |
| 7 | Reload | Button | Thực hiện thao tác làm mới các thông tin tìm kiếm đã nhập |
| 8 | Thông tin khách hàng | Table viewer | Hiển thị danh sách thông tin khách hàng |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 10 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

1. Thêm, sửa



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | CMND | Textfield | Nhập vào chứng minh nhân dân |
| 2 | Họ tên | Textfield | Nhập vào họ tên |
| 3 | Loại khách | Combobox | Lựa chọn loại khách |
| 4 | Địa chỉ | Textfield | Nhập vào ghi chú cần thiết |
| 5 | Clean | Button | Thực hiện thao tác làm mới các thông tin vừa nhập |
| 6 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm hoặc sửa thông tin khách hàng |
| 7 | Thông tin khách hàng | Table viewer | Hiển thị danh sách thông tin khách hàng |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

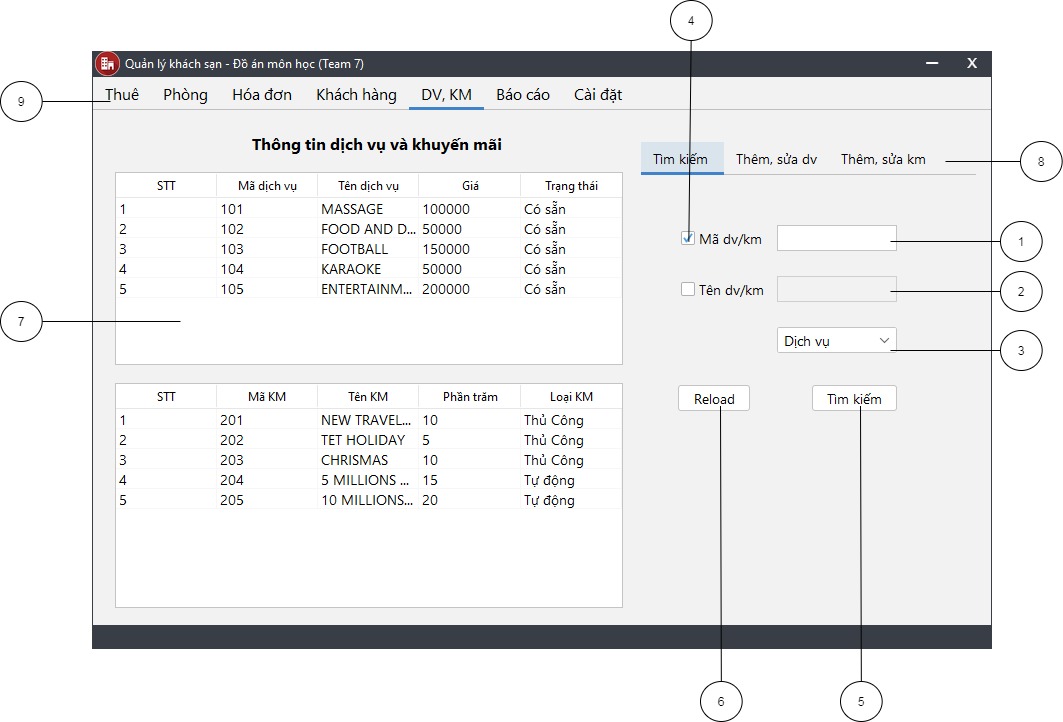
1. Quản lí loại khách



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Loại khách | Textfield | Nhập vào loai khách mới |
| 2 | Hệ số | Textfield | Nhập vào hệ số |
| 3 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm loại khách |
| 4 | Clean | Button | Thực hiện thao tác làm mới các thông tin vừa nhập |
| 5 | Thông tin loại khách | Table viewer | Hiển thị các loại khách |
| 6 | Thông tin khách hàng | Table viewer | Hiển thị danh sách khách hàng |
| 7 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

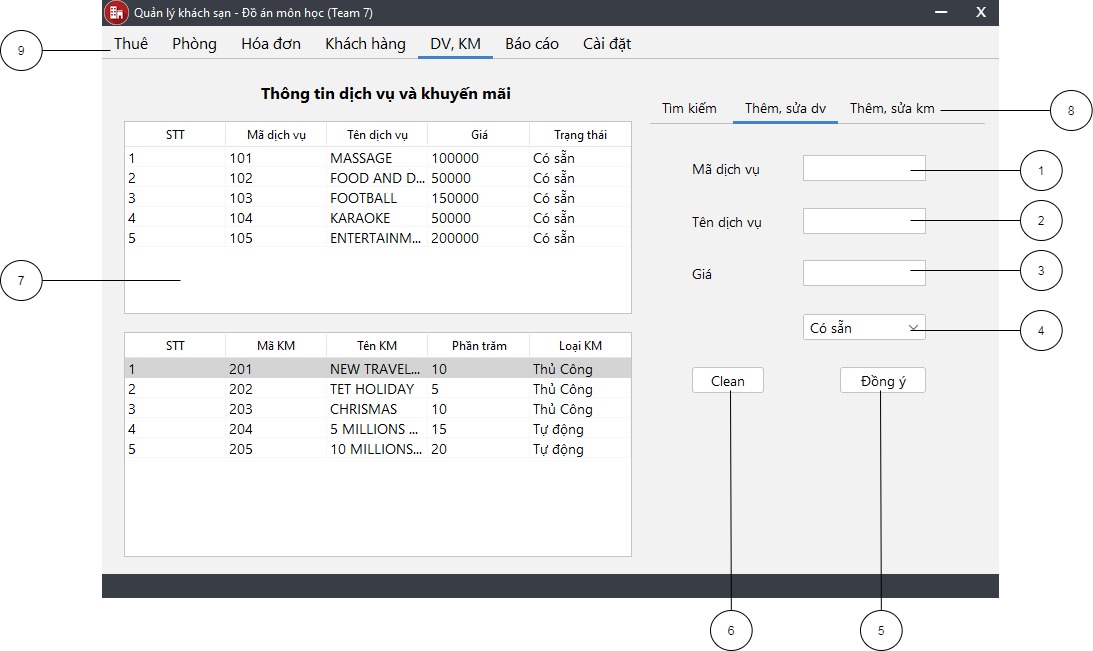
* 1. DV, KM

1. Tìm kiếm



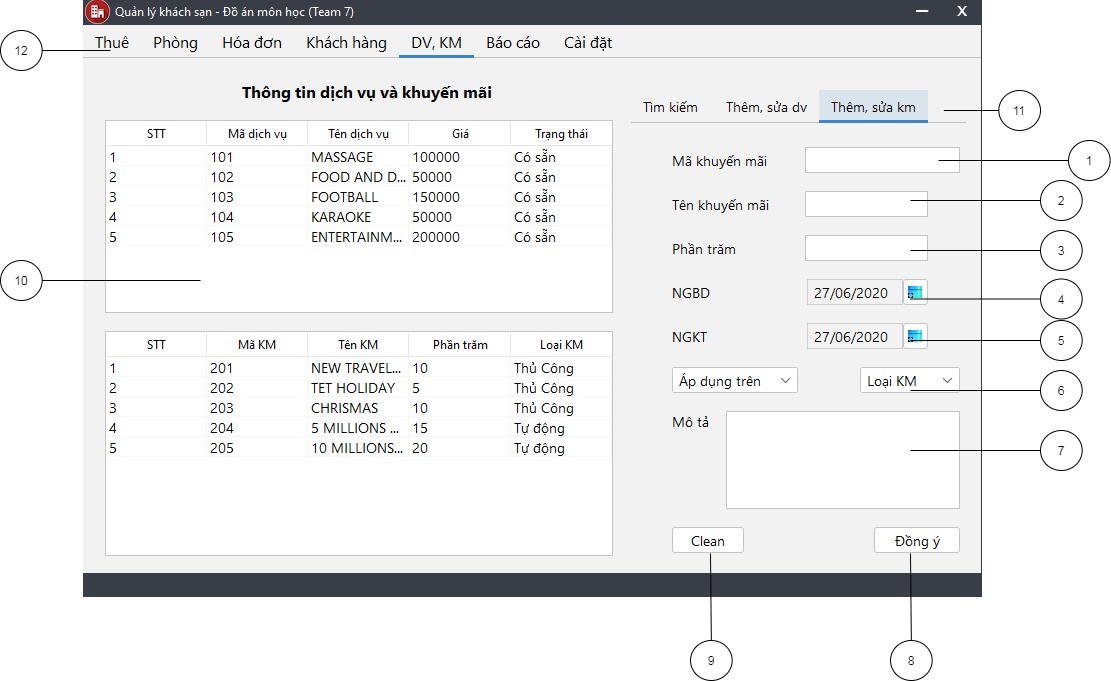
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã dv/km | Textfield | Nhập vào mã dịch vụ hoặc mã khuyến mãi |
| 2 | Tên dv/km | Textfield | Nhập vào tên dịch vụ hoặc tên khuyến mãi |
| 3 | Khung chọn | Combobox | Lựa chọn tìm kiếm dịch vụ hoặc khuyến mãi |
| 4 | Ô chọn | Radiobutton | Lựa chọn tìm kiếm theo mã hoạc theo tên |
| 5 | Tìm kiếm | Button | Thực hiện thao tác tìm kiếm |
| 6 | Reload | Button | Thực hiện thao tác làm mới thông tin vừa nhập |
| 7 | Thông tin dịch vụ khuyến mãi | Table viewer | Hiển thị dach sách các dịch vụ và khuyến mãi |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

1. Thêm, sửa dv



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã dịch vụ | Textfield | Nhập vào mã dịch vụ |
| 2 | Tên dịch vụ | Textfield | Nhập vào tên dịch vụ |
| 3 | Giá | Textfield | Nhập vào mức giá của dịch vụ |
| 4 | Khung chọn | Combobox | Lựa chọn trạng thái dịch vụ |
| 5 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm hoặc sửa dịch vụ |
| 6 | Clean | Button | Thực hiện thao tác làm mới thông tin vừa nhập |
| 7 | Thông tin dịch vụ và khuyến mãi | Table viewer | Hiển thị danh sách thông tin dịch vụ và khuyến mãi |
| 8 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 9 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

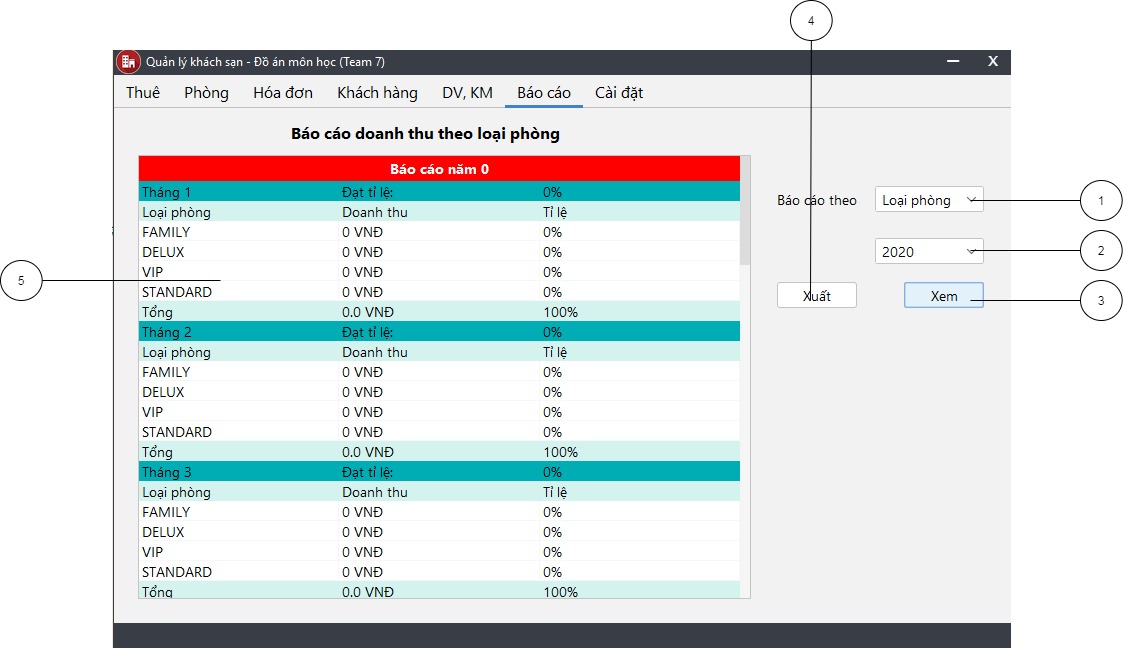
1. Thêm, sửa km



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã khuyến mãi | Textfield | Nhập vào mã khuyến mãi |
| 2 | Tên khuyến mãi | Textfield | Nhập vào tên khuyến mãi |
| 3 | Phần trăm | Textfield | Nhập vào phần trăm khuyến mãi |
| 4 | Ngày bắt đầu | Jcalendar | Lựa chọn ngày bắt đầu khuyến mãi |
| 5 | Ngày kết thức | Jcalendar | Lựa chọn ngày kết thúc khuyến mãi |
| 6 | Áp dụng trên  Loại KM | Combobox | Lựa chọn áp dụng trên tiền phòng hoặc tổng hóa đơn  Lựa chọn loại khuyến mãi (thủ công hoặc tự động) |
| 7 | Mô tả | Textfield | Nhập vào mô tả chi tiết khuyến mãi |
| 8 | Đồng ý | Button | Thực hiện thao tác xác nhận thêm hoặc sửa thông tin khuyến mãi |
| 9 | Clean | Button | Thực hiện thao tác làm mới thông tin vừa nhập |
| 10 | Thông tin dịch vụ và khuyến mãi | Table viewer | Hiển thị danh sách thông tin dịch vụ và khuyến mãi |
| 11 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 12 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |

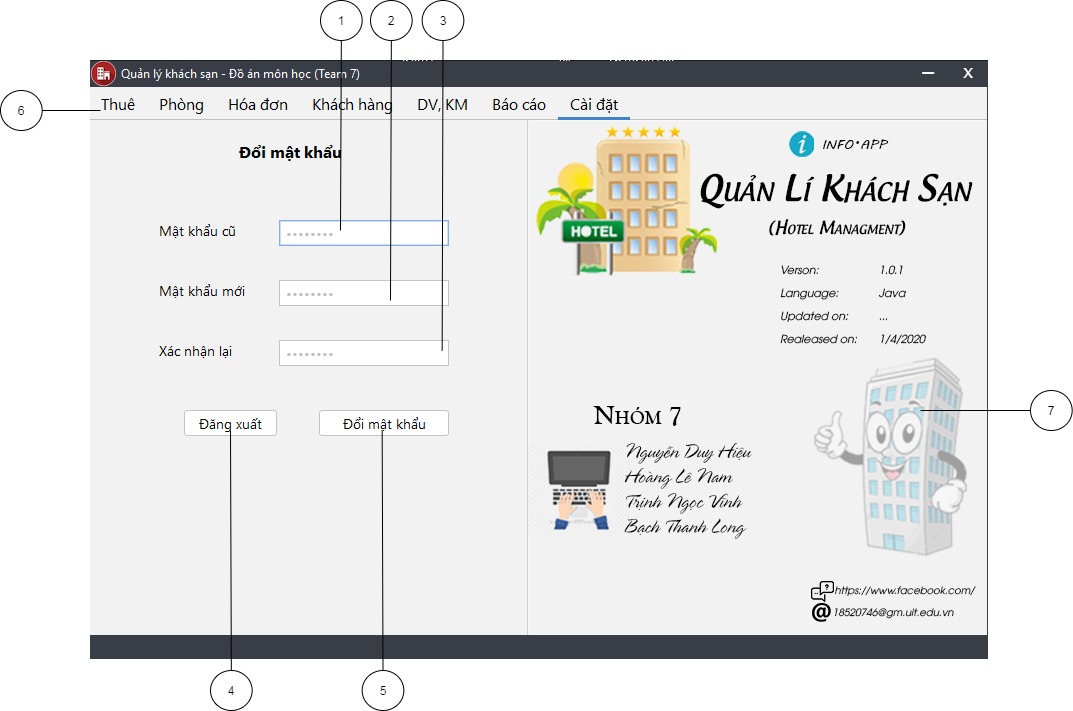
* 1. Báo cáo

1. Báo cáo



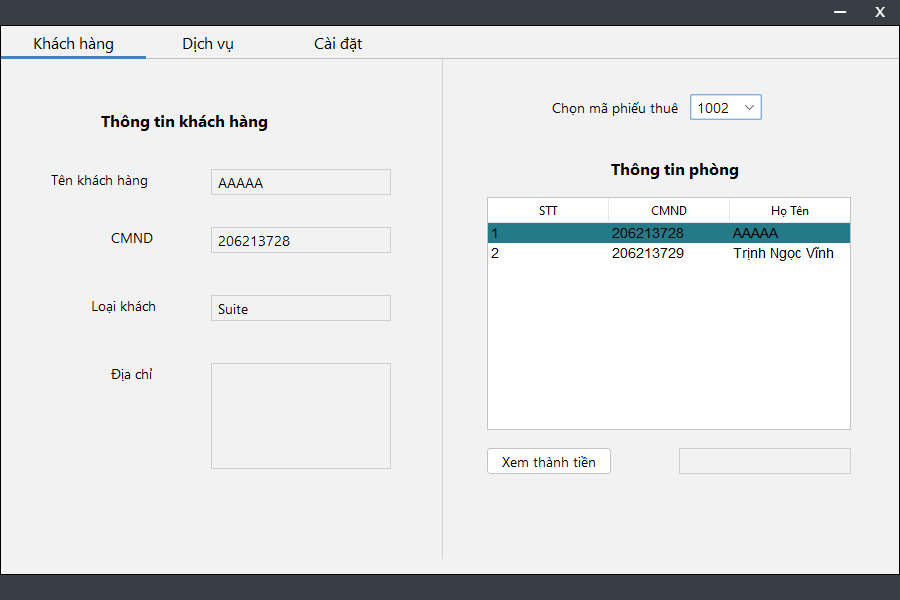
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Báo cáo theo | Combobox | Lựa chọn các thức báo cáo |
| 2 | Báo cáo theo | Combobox | Lựa chọn năm |
| 3 | Xem | Button | Thực hiện thao tác hiển thị bản báo cáo |
| 4 | Xuất | Button | Xuất báo cáo |
| 5 | Báo cáo doanh thu | Table viewer | Hiển thị bản báo cáo |

1. Xuất
   1. Cài đặt



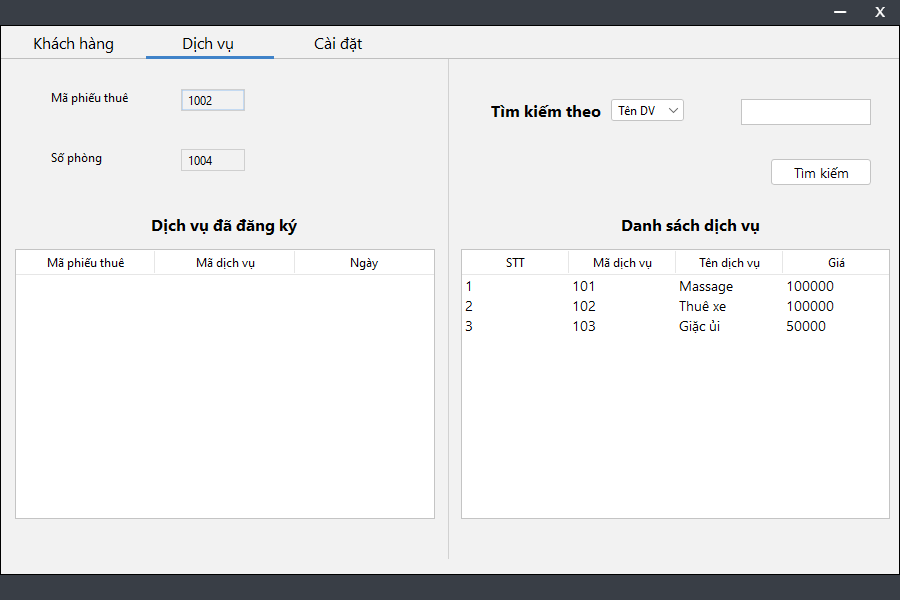
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mật khẩu cũ | Textfield | Nhập mật khẩu cũ |
| 2 | Mật khẩu mới | Textfield | Nhập mật khẩu mới |
| 3 | Xác nhập lại | Textfield | Nhập lại mật khẩu mới một lần nữa |
| 4 | Đăng xuất | Button | Thực hiên thao tác đăng xuất |
| 5 | Đổi mật khẩu | Button | Thực hiện thao tác đổi mật khẩu |
| 6 | Thanh lựa chọn | TabbedPane | Lựa chọn màn hình muốn xem |
| 7 | Logo | Label | Hiển thị thông tin |

1. Khách hàng – from khách hàng



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Tên khách hàng | TextField | Hiển thị tên khách hàng. |
| 2 | CMND | TextField | Hiển thị số CMND của khách hàng. |
| 3 | Loại khách | TextField | Hiển thị loại khách hàng. |
| 4 | Địa chỉ | TextField | Hiển thị địa chỉ của khách hàng |
| 5 | Chọn mã phiếu thuê | ComboBox | Hiển thị toàn bộ mã phiếu thuê của khách hàng và cho phép chọn. |
| 6 | Thông tin phòng | Table | Hiển thị người thuê chung phòng theo mã hóa đơn đã chọn. |
| 7 | Xem thành tiền | Button | Thực hiện tính thành tiền của phòng theo mã phiếu thuê đã chọn. |
| 8 | Thành tiền | TextField | Hiển thị số tiền cần thành toán |

1. Dịch vụ - form khách hàng



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | Mã phiếu thuê | TextField | Hiển thị mã phiếu thuê đã chọn ở Tab khách hàng |
| 2 | Số phòng | TextField | Hiển thị số phòng theo mã phiếu thuê đã chọn ở Tab khách hàng |
| 3 | Dịch vụ đã đăng kí | Table | Hiển thị thông tin các dịch vụ đã đăng kí của mã phiếu thuê đã chọn ở Tab khách hàng |
| 4 | Danh sách dịch vụ | Table | Hiển thị thông tin các dịch vụ hiện có theo yêu cầu tìm kiếm. |
| 5 | Tìm kiếm theo | ComboBox | Hiển thị và cho phép chọn các kiểu loại (theo tên dịch vụ, theo mã dịch vụ hoặc tất cả) muốn tìm kiếm |
| 6 | Tìm kiếm | TextField | Cho phép nhập thông tin dịch vụ muốn tìm kiếm |
| 7 | Tìm kiếm | Button | Thực hiện tìm kiếm |

# Kết luận

## Kết quả đạt được

Trong thời gian thực hiện dự án, nhóm đạt được những kết quả sau:

* Thực hiện được các chức năng cơ bản cho phần mềm “Quản lí thuê trả phòng”:
* Thuê, trả phòng
* Quản lý phòng và loại phòng
* Quản lý khách hàng và loại khách hàng
* Quản lý hóa đơn, thanh toán và xuất hóa đơn
* Quản lý Dịch vụ, khuyến mãi
* Quản lý việc đăng ký dịch vụ, khuyến mãi
* Xem và xuất báo cáo doanh thu theo từng tháng, từng năm
* Qua việc thực hiện dự án, các thành viên tham gia dự án rút ra được nhiều kinh nghiệm hơn cho bản thân như: kinh nghiệm làm việc nhóm, có được nhiều kiến thức mới trong ngành, nâng cao khả năng tự học hỏi của bản thân.
* Ngoài ra, đã hiểu và vận dụng được các kiến thức của môn hệ quản trị cơ sở dữ liệu, cụ thể là làm việc trên Oracle Database, như:
* Các cú pháp và cách tư duy về ngôn ngữ PL/SQL.
* Cách viết và thực thi các Trigger, Function, Stored Procedure.
* Cách tối ưu hóa các câu truy vấn dữ liệu.
* Kiến thức về khóa và các mức cô lập.
* Khôi phục dữ liệu khi xảy ra sự cố.
* Giải quyết được các trường hợp truy xuất đồng thời.

## Hạn chế

Thời gian thực hiện dự án ngắn, lượng công việc phải hoàn thành lớn nên nhóm còn nhiều hạn chế về nhiều mặt:

* + Nhóm mới làm dự án lần đầu còn chưa có nhiều kinh nghiệm.
  + Các tình huống truy xuất đồng thời có thể không hợp lí.
  + Một số thành viên bận công việc hoặc học tập khác.

## Hướng phát triển

* + Mở rộng khả năng tương tác của phần mềm.
  + Tối ưu hóa bằng thuật toán giúp phần mềm hoạt động nhanh và hiệu quả hơn.
  + Mở rộng thêm nhiều chức năng khác cho phần mềm: khách hàng có thể thanh toán online hoặc qua thẻ ATM, xây dựng kênh hỗ trợ trực tuyến, khách hàng có thể sử dụng phần mềm trên nền tảng mobile, …

# Bảng phân công công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Nguyễn Duy Hiệu | Bạch Thanh Long | Hoàng Lê Nam | Trịnh Ngọc Vĩnh |
| Phân tích bài toán, xác định chức năng | 🗸 | 🗸 | 🗸 | 🗸 |
| Thiết kế CSDL | 🗸 | 🗸 | 🗸 | 🗸 |
| Thiết kế giao diện |  | 🗸 | 🗸 |  |
| Xử lí giao diện |  | 🗸 | 🗸 |  |
| Xử lí các vấn đề liên quan đến truy xuất đồng thời | 🗸 |  |  | 🗸 |
| Test chức năng | 🗸 |  |  | 🗸 |
| Viết Báo cáo | 🗸 | 🗸 | 🗸 | 🗸 |

Bảng tỉ lệ phân chia công việc:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | Nguyễn Duy Hiệu | Bạch Thanh Long | Hoàng Lê Nam | Trịnh Ngọc Vĩnh |
| Tỉ lệ | 25% | 25% | 25% | 25% |
| Tỉ lệ hoàn thành | 90% | 80% | 80% | 85% |

# Tài liệu tham khảo

1. Các slide môn Hệ quản trị cơ sở dữ liệu và Lập trình Java của khoa Hệ thống thông tin
2. Chương 9,10 (<https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/transactions.html>)