

BÁO CÁO ĐỒ ÁN THỰC HÀNH
MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
GVHD: Cô Hồ Thị Hoàng Vy

MỤC LỤC

THÔNG TIN NHÓM	5
BẢNG THIẾT KẾ THỰC THỂ KẾT HỢP	6
I. Lược đồ thực thể kết hợp	6
II. Ràng buộc dữ liệu bổ sung	6
III. Mô hình cơ sở dữ liệu quan hệ	9
IV. Xác định các loại người dùng	9
XÁC ĐỊNH CHỨC NĂNG HỆ THỐNG VÀ TÌNH HUỐNG TRANH CHẤP	12
I. Các chức năng của hệ thống	12
1. Chức năng cho DỪNG CHUNG	12
2. Phân hệ Khách hàng	12
3. Phân hệ Bác sĩ	13
4. Phân hệ Nhân viên	13
5. Phân hệ Quản trị viên	13
II. Xác định tình huống tranh chấp	15
III. Thiết kế giao diện chức năng	17
1. Dùng chung	17
1.1. Đăng nhập	17
2. Admin	17
2.1. Thông tin tài khoản	17
2.2. Quản lý tài khoản	18
2.3. Thêm tài khoản	19
2.4. Quản lý thuốc	19
2.5. Quản lý dịch vụ	20
3. Bác sĩ	20
3.1. Quản lý hồ sơ bệnh án	20
3.2. Thông tin cá nhân	21
3.3. Thêm hồ sơ	21
3.4. Cập nhật hồ sơ bệnh án	22
3.5. Quản lý lịch hẹn	22
4. Nhân viên	23
4.1. Đăng ký khách hàng	23
4.2. Đăng ký lịch hẹn khách hàng	23
4.3. Xem hồ sơ	24

4.4. Xem hóa đơn	24
5. Bệnh nhân	25
5.1. Sau khi đăng nhập role khách hàng	25
5.2. Đăng ký khách hàng mới (Sau khi click vào nút Sign up ở giao diện đăng nhập)	25
5.3. Đặt lịch hẹn khi click vào đặt lịch hẹn	26
5.4. Cửa sổ xem thông tin khách hàng sau khi click vào button xem thông tin cá nhân	26
CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP	27
I. Sinh viên thực hiện: Dương Hiền Lê Hoàng	27
1. Tình huống 1	27
2. Tình huống 2	30
II. Sinh viên thực hiện: Đinh Thành Danh	33
1. Tình huống 1	33
2. Tình huống 2	38
III. Sinh viên thực hiện: Nguyễn Hoài Mẫn	43
1. Tình huống 1	43
2. Tình huống 2	47
IV. Sinh viên thực hiện: Nguyễn Phụng Khanh	52
1. Tình huống 1	52
2. Tình huống 2	55
V. Giao tác một số chức năng:	58
XỬ LÝ TÌNH HUỐNG TRANH CHẤP	66
I. Sinh viên thực hiện: Dương Hiền Lê Hoàng	66
1. Tình huống 1	66
2. Tình huống 2	69
II. Sinh viên thực hiện: Nguyễn Hoài Mẫn	72
1. Tình huống 1	72
2. Tình huống 2	77
III. Sinh viên thực hiện: Nguyễn Phụng Khanh	82
1. Tình huống 1	82
2. Tình huống 2	85
IV. Sinh viên thực hiện: Đinh Thành Danh	89
1. Tình huống 1	89
2. Tình huống 2	92

THÔNG TIN NHÓM

STT	MSSV	Họ tên	Công việc	% Hoàn thành
1	20127561	Nguyễn Hoài Mẫn	<ul style="list-style-type: none">Thiết kế ER, viết ràng buộc dữ liệu, vẽ mô hình CSDLQH, xác định các loại người dùng.Xác định chức năng phân hệ bác sĩ.Thiết kế chức năng giao diện Đăng nhập, Quản lý hồ sơ bệnh án, Thông tin cá nhân, Thêm hồ sơ, Quản lý lịch hẹn.Cài đặt tình huống tranh chấp: LOST UPDATE, UNREPEATABLE READSửa lỗi tình huống tranh chấp: LOST UPDATE, UNREPEATABLE READ	100%
2	20127423	Đinh Thành Danh	<ul style="list-style-type: none">Thiết kế ER, viết ràng buộc dữ liệu, vẽ mô hình CSDLQH.Xác định chức năng phân hệ khách hàng.Xác định tình huống tranh chấp Dirty Read, Phantom.Thiết kế chức năng giao diện Đăng ký khách hàng, Đăng ký lịch hẹn, Xem thông tin của khách hàng (bệnh nhân)Cài đặt tình huống tranh chấp: DIRTY READ, PHANTOM READGiao tác chức năngSửa lỗi tình huống tranh chấp: DIRTY READ, PHANTOM READ	100%
3	20127503	Dương Hiền Lê Hoàng	<ul style="list-style-type: none">Thiết kế ER, tạo script, vẽ mô hình CSDLQH.Xác định chức năng phân hệ nhân viên.Thiết kế chức năng giao diện Xem hồ sơ, Xem hóa đơn.Cài đặt tình huống tranh chấp: DIRTY READ, UNREPEATABLE READSửa lỗi tình huống tranh chấp: DIRTY READ, UNREPEATABLE READ	100%
4	20127204	Nguyễn Phụng Khanh	<ul style="list-style-type: none">Thiết kế ER, viết ràng buộc dữ liệu, vẽ mô hình CSDLQH.Xác định chức năng dùng chung, phân hệ quản trị viên.	100%

- Thuộc tính *LoaiTK* chỉ nhận 1 trong 5 giá trị: -1 là tài khoản bị khóa, 0 là Admin, 1 là nha sĩ, 2 là nhân viên, 3 là bệnh nhân.

❖ **Bảng NhanVien:**

- Các thuộc tính *MaNV*, *HoTenNV*, *NgaySinhNV*, *DiaChiNV*, *SĐTNV*, *EmailNV* là NOTNULL.
- *SĐTNV*, *EmailNV* là duy nhất

❖ **Bảng QuanTri:**

- Các thuộc tính *MaQT*, *HoTenQT*, *NgaySinhQT*, *DiaChiQT*, *SĐTQT*, *EmailQT* là NOTNULL.
- *SĐTQT*, *EmailQT* là duy nhất

❖ **Bảng KháchHang**

- Các thuộc tính *MaKH*, *HoTenKH*, *NgaySinhKH*, *DiaChiKH*, *SDTKH*, *EmailKH* là NOTNULL
- Thuộc tính *MaKH* là duy nhất để phân biệt khách hàng

❖ **Bảng HoaDon**

- Các thuộc tính *MaHD*, *TenKH*, *NgayLap*, *PhiThanhToan* là NOTNULL
- *MaHD* là thuộc tính duy nhất phân biệt giữa các hợp đồng
- *PhiThanhToan* là thuộc tính phải > hoặc bằng 0
- *NgayLap* là ngày hiện tại lúc lập hóa đơn

❖ **Bảng HoSoBN**

- Các thuộc tính *MaHS*, *NgayKham*, *TenNhaSi* là NOTNULL

- Thuộc tính MaHS là thuộc tính dùng để phân biệt giữa các hồ sơ nên nó là duy nhất

❖ Bảng Thuoc

- Các thuộc tính : MaThuoc, TenThuoc, DonViTinh, ChiDinh, SLtonkho, NgayHetHan, Tien là NOTNULL
- Thuộc tính để phân biệt giữa các thuốc với nhau là MaThuoc vì nó là duy nhất
- SLtonkho phải có giá trị ≥ 0
- Tien phải có giá trị > 0

❖ Bảng LichHen

- Các thuộc tính : MaLichHen, NgayHen, ThoiGianHen, NguoiDatLich là NOTNULL
- Thuộc tính duy nhất để phân biệt giữa các lịch hẹn là MaLichHen

❖ Bảng NhaSi

- Các thuộc tính : MaNS, HoTenNS, NgaySinhNS, DiaChiNS, SDTNS, EmailNS là NOTNULL
- Thuộc tính phân biệt giữa các nha sĩ là MaNS

❖ Bảng LichCaNhanNS

- Các thuộc tính MaLichCaNhan, Ngay, ThoiGian là NOTNULL
- MaLichCaNhan là thuộc tính duy nhất

❖ Bảng CT_HoaDon

- Các thuộc tính : MaCTHD, MaDV, MaThuoc, TenDV, TenThuoc, SL, LoaiDV, ThanhTien là NOTNULL
- MaCTHD là thuộc tính dùng để phân biệt giữa các chi tiết hóa đơn
- SL và ThanhTien là hai thuộc tính có giá trị lớn hơn hoặc bằng không

- Khách hàng cũ: Người dùng này đã đăng ký tài khoản trước đó và có thể đăng nhập vào hệ thống để sử dụng các chức năng như đặt lịch hẹn, xem thông tin cá nhân, và xem hồ sơ bệnh án.

➤ Nha sĩ:

- Nha sĩ là người chuyên thực hiện khám và điều trị cho bệnh nhân. Họ có quyền thêm thông tin vào hồ sơ bệnh nhân, quản lý cuộc hẹn, và xem lịch hẹn của họ. Nha sĩ có thể cập nhật lịch cá nhân nhưng không thể cập nhật lịch hẹn do khách hàng đặt.

➤ Nhân viên:

- Nhân viên là người tiếp nhận bệnh nhân và hỗ trợ khách hàng. Họ có nhiệm vụ tiếp nhận thông tin từ khách hàng để đặt lịch khám hoặc đăng ký khám. Nhân viên cũng thực hiện thao tác thanh toán dựa trên thông tin khám bệnh của khách hàng. Họ in thông tin đăng ký khám và hóa đơn thanh toán cho khách hàng.

➤ Quản trị viên:

- Quản trị viên có quyền cao nhất trong hệ thống. Họ có khả năng quản lý danh mục thuốc, thêm/xóa/sửa thông tin thuốc, cập nhật số lượng tồn trong kho thuốc, và xem thông tin về thuốc. Quản trị viên cũng có quyền quản lý người dùng bằng cách thêm tài khoản nhân viên và nha sĩ, cũng như khóa tài khoản người dùng khi cần thiết.

XÁC ĐỊNH CHỨC NĂNG HỆ THỐNG VÀ TÌNH HUỐNG TRANH CHẤP

I. Các chức năng của hệ thống

1. Chức năng cho DÙNG CHUNG

STT	Chức năng	Mô tả hoạt động
ALL1	Đăng nhập vào tài khoản.	Đăng nhập vào trang web quản lý dựa vào tài khoản và mật khẩu.
ALL2	Quản lý thông tin cá nhân của mỗi tài khoản	Mỗi phân hệ người dùng đều tự xem và cập nhật được thông tin cá nhân của tài khoản mình.
ALL3	Xem danh sách lịch hẹn	Xem được danh sách lịch hẹn của nha sĩ dù là KH, NS và QTV.
ALL4	Chỉnh sửa mật khẩu tài khoản	Mỗi phân hệ người dùng được tự chỉnh sửa mật khẩu theo ý muốn (đúng quy tắc)

2. Phân hệ Khách hàng

STT	Chức năng	Mô tả hoạt động
KH1	Đăng ký tài khoản	<ul style="list-style-type: none"> - Người dùng truy cập chọn đăng ký tài khoản - Người dùng điền đầy đủ những thông tin như SĐT, Họ và tên, Email, mật khẩu v.v
KH2	Đặt lịch hẹn trực tuyến	<ul style="list-style-type: none"> - Sau khi đăng nhập thành công khách hàng chọn mục “ “ Đặt lịch hẹn “ để thực hiện đặt lịch hẹn - Tiếp đó chọn ngày giờ khám và nha sĩ sẽ khám cho mình
KH3	Xem thông tin cá nhân	<ul style="list-style-type: none"> - Khách hàng đăng nhập vào tài khoản của mình sau đó chọn tài khoản cá nhân chọn mục “ Xem thông tin cá nhân “ - Sau khi khách hàng chọn hệ thống sẽ hiển thị những thông tin như họ tên, ngày sinh, địa chỉ, sđt....
KH4	Xem hồ sơ bệnh án	<ul style="list-style-type: none"> - Sau khi đăng nhập chọn vào mục tài khoản cá nhân tiếp đó chọn mục “ Xem hồ sơ bệnh án “ - Hệ thống sẽ hiển thị thông tin về bệnh án của bệnh nhân
KH5	Xem danh sách dịch vụ	<ul style="list-style-type: none"> - Hệ thống sẽ hiển thị lên danh sách dịch vụ mà phòng khám đa khoa đang diễn ra
KH6	Xem danh sách nha sĩ	<ul style="list-style-type: none"> - Hệ thống sẽ hiển thị danh sách nha sĩ để khách hàng lựa chọn

3. Phân hệ **Bác sĩ**

STT	Chức năng	Mô tả hoạt động
BS1	Đăng nhập vào hệ thống	Đăng nhập vào hệ thống QUẢN LÝ PHÒNG KHÁM NHA KHOA dựa vào tài khoản và mật khẩu đã được quản trị viên tạo.
BS2	Quản lý hồ sơ bệnh nhân	<ul style="list-style-type: none"> - Bác sĩ sẽ thêm thông tin bệnh nhân vào hồ sơ bệnh nhân - Ghi nhận thông tin khám bệnh, bao gồm ngày khám, người thực hiện, dịch vụ sử dụng và danh sách thuốc kê cho mỗi lần khám.
BS3	Xem danh sách hồ sơ bệnh án thực hiện	Bác sĩ có quyền xem, xóa, sửa, cập nhật các hồ sơ bệnh án do chính mình thực hiện
BS4	Xem danh sách thuốc	Bác sĩ có quyền xem các danh sách thuốc của bệnh viện để kê cho mỗi lần khám
BS5	Xem danh sách dịch vụ	Bác sĩ có quyền xem danh sách dịch vụ của bệnh viện để kê cho mỗi lần khám
BS6	Quản lý cuộc hẹn	<ul style="list-style-type: none"> - Xem lịch hẹn cá nhân. - Thêm lịch hẹn cá nhân vào hệ thống. - Cập nhật lịch cá nhân (không được cập nhật lịch hẹn do khách hàng đặt).

4. Phân hệ **Nhân viên**

STT	Chức năng	Mô tả hoạt động
NV1	Đăng nhập vào hệ thống	Đăng nhập vào hệ thống QUẢN LÝ PHÒNG KHÁM NHA KHOA dựa vào tài khoản và mật khẩu đã được quản trị viên tạo
NV2	Đăng ký khám cho khách hàng	Nhân viên sẽ điền các thông tin của khách hàng vào trong hệ thống
NV3	Đăng ký lịch khám	Nhân viên sẽ điền thông tin lịch khám cho khách hàng
NV4	Tìm kiếm hồ sơ bệnh án	Nhân viên sẽ tìm hồ sơ bệnh án của bệnh nhân dựa trên số điện thoại của khách hàng đưa cho
NV5	Xem danh sách hóa đơn	Nhân viên xem danh sách hóa đơn của khách hàng và nhân viên có thể tìm hóa đơn của khách hàng dựa trên mã khách hàng

5. Phân hệ **Quản trị viên**

STT	Chức năng	Mô tả hoạt động
-----	-----------	-----------------

QTV1	Đăng nhập vào database	QTV đăng nhập vào database để truy xuất dữ liệu về thuốc và quản lý nhân viên, nha sĩ.
QTV2	Xem và chỉnh sửa thông tin thuốc	Quản trị viên có thể thực hiện xóa thuốc hết hạn, hoặc thêm thuốc mới khi thuốc được nhập vào kho. Quản trị viên cũng có thể cập nhật số lượng tồn của thuốc.
QTV3	Khóa và kích hoạt tài khoản	Xem xét các trường hợp khóa tài khoản khi có dấu hiệu xâm nhập hoặc nhận được báo cáo trợ giúp từ nhân viên và nha sĩ.
QTV4	Thêm-xóa-sửa tài khoản	THÊM trong trường hợp có NS, NV đăng ký tài khoản mới trên hệ thống. XOÁ với trường hợp NS, NV không còn nhu cầu sử dụng dịch vụ. SỬA khi người dùng yêu cầu cấp quyền thao tác trên dữ liệu người dùng.
QTV5	Cấp quyền thao tác trên dữ liệu	Cho phép cập nhật thông tin liên quan, khách hàng, nha sĩ chỉnh sửa thông tin cá nhân, chỉ khi có yêu cầu thao tác dữ liệu, và phải được bộ phận quản trị cấp quyền.
QTV6	Cập nhật quyền người dùng	Bổ sung các quy tắc dành cho các đối tượng người dùng Quyết định giới hạn truy cập của người dùng trên dữ liệu hệ thống.

II. Xác định tình huống tranh chấp

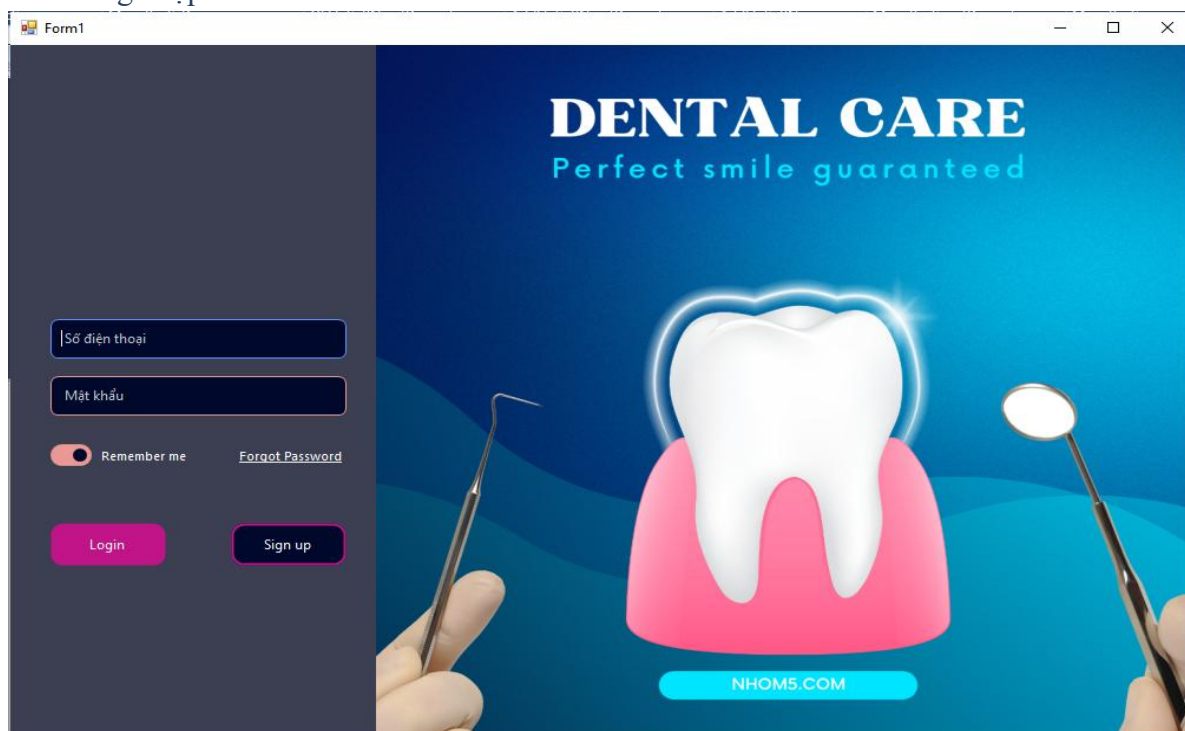
Chức năng 1	Người dùng	Chức năng 2	Người dùng	Lỗi tranh chấp
Giao tác 1: Đặt lịch hẹn trực tuyến khách hàng A đặt lịch hẹn với nha sĩ A vào 9h sáng ngày 11/11) Giao tác 1 ghi nhận lịch hẹn của KH A với NS A	Khách hàng	Giao tác 2: Đặt lịch hẹn trực tuyến (khách hàng B đặt lịch hẹn với nha sĩ A vào 9h sáng cùng ngày 11/11) => Giao tác 2 ghi nhận lịch hẹn của KH B với NS A	Khách hàng	2 giao tác xảy ra đồng thời, hệ thống hoàn thành hai giao tác, ghi nhận lịch hẹn thể ghi nhận lịch hẹn của KH B , tùy thuộc vào hiện của chúng. Điều này dẫn đến mất mát thông tin khi bị ghi đè bởi một trong hai giao tác, mà không có kiểm soát => LOST UPDATE
Giao tác 1: Cập nhật thông tin thuốc tồn kho (Loại thuốc A còn 30 vỉ trong kho, QTV A cập nhật SL tồn kho của loại thuốc A còn 20 vỉ) Giao tác 1 ghi nhận SL tồn kho còn lại của loại thuốc A là 20 vỉ	Quản trị viên	Giao tác 1: Cập nhật thông tin thuốc tồn kho (Loại thuốc A còn 30 vỉ trong kho, QTV B cập nhật SL tồn kho của loại thuốc A còn 15 vỉ) => Giao tác 2 ghi nhận SL tồn kho còn lại của loại thuốc A là 15 vỉ	Quản trị viên	2 giao tác xảy ra đồng thời, hệ thống hoàn thành hai giao tác, ghi nhận SL tồn kho của loại thuốc A là 20 hoặc 15 , tùy thuộc vào thứ tự thực hiện của hai giao tác này dẫn đến mất mát thông tin tồn kho có thể bị ghi đè trong hai giao tác, mà không có kiểm soát. => LOST UPDATE
• Giao tác 1 (đọc lần thứ nhất): Xem thông tin SL tồn kho của loại thuốc (QTV A muốn xem SL tồn kho của loại thuốc B ở thời điểm đó, đọc SL từ CSDL lên là 35 hộp) Giao tác 1 (đọc lần thứ hai, sau khi giao tác 2 đã thực hiện thay đổi): Đọc lại SL tồn kho của loại thuốc B, nhưng lần này là 25 hộp thay vì 35 như lần đầu	Quản trị viên	Giao tác 2: Cập nhật thông tin SL tồn kho của thuốc (QTV B cập nhật SL tồn kho của loại thuốc B cùng thời điểm đó, còn 25 hộp và lưu vào CSDL) => Giao tác 2 xảy ra sau giao tác 1 và khi giao tác 1 chưa hoàn thành	Quản trị viên	2 giao tác tạo ra mâu thuẫn, hệ thống không nhất quán khi đọc lại từ cơ sở dữ liệu, lưu lại từ csdl không còn đúng dữ liệu đã thay đổi khác, gây ra sự không nhất quán trong dữ liệu mà giao tác trước đã thay đổi => UNREPEATABLE READ
• Giao tác 1 (đọc lần thứ nhất): Xem thông tin cá nhân của nha sĩ A (Nhân viên muốn xem số điện thoại của nha sĩ để liên lạc việc cần thiết - 0912345678) Giao tác 2 (đọc lần thứ hai, sau	Nhân viên	Giao tác 2: Cập nhật số điện thoại của nha sĩ A (Nha sĩ A tự cập nhật thông tin cá nhân cùng thời điểm nhân viên xem số điện thoại nsA, thành 0987654321 và lưu vào CSDL) => Giao tác 2 xảy ra sau giao tác 1 và	Nha sĩ	2 giao tác tạo ra mâu thuẫn, hệ thống không nhất quán khi đọc lại từ cơ sở dữ liệu, lưu lại từ csdl không còn đúng dữ liệu đã thay đổi khác, gây ra sự không nhất quán trong dữ liệu mà giao tác trước đã thay đổi => UNREPEATABLE READ

<p>khí giao tác 2 đã thực hiện thay đổi): Đọc lại số điện thoại của nhà sĩ A, nhưng lần này là 0987654321 thay vì số như lần đọc đầu tiên</p>		<p>khí giao tác 1 chưa hoàn thành</p>		<p>khác, gây ra sự không nhất quán trong dữ liệu mà giao tác trước đã commit</p> <p>=> UNREPEATABLE READ</p>
<p>dùng 1 (Khách hàng): Khách hàng đặt lịch hẹn với nhà sĩ X vào thời điểm T1.</p>	Khách hàng	<p>Người dùng 2 (Nhà sĩ): Trong khoảng thời gian từ T1 đến T2, nhà sĩ X xem lịch hẹn và không thấy lịch hẹn của Khách hàng A. Tuy nhiên, sau đó, lịch hẹn của Khách hàng A được thêm vào và nhà sĩ X thấy lịch hẹn mới này (phantom).</p>	Nhân viên	<p>Lỗi phantom xảy ra do giao tác 1 (Khách hàng) đã commit lịch hẹn (T1 đến T2), có một bản sao của lịch hẹn của Khách hàng A được commit sau đó. Khi giao tác 2 (Nhà sĩ) xem lại lịch hẹn sau thời điểm T1, lịch hẹn mới này xuất hiện như "phantom" cho nhà sĩ X.</p> <p>=> PHANTOM</p>
<p>dùng 1 (Khách hàng): Khách hàng đặt lịch hẹn với nhà sĩ Z vào thời điểm T3.</p>	Khách hàng	<p>Người dùng 2 (Nhà sĩ): Trong khoảng thời gian từ T2 đến T4, nhà sĩ Z xem lịch hẹn và thấy lịch hẹn của Khách hàng B. Lưu ý rằng lịch hẹn của Khách hàng B không xuất hiện khi nhà sĩ Z xem lịch tại thời điểm T2 (phantom)</p>	Nhân viên	<p>Lỗi phantom xảy ra khi giao tác 1 (Khách hàng) đã commit lịch hẹn (T2 đến T4), có một bản sao của lịch hẹn của Khách hàng B được commit sau đó. Khi giao tác 2 (Nhà sĩ) xem lại lịch hẹn sau thời điểm T2, lịch hẹn mới này xuất hiện như "phantom" cho nhà sĩ Z.</p> <p>=> PHANTOM</p>
<p>dùng 1 (Khách hàng): Khách hàng đăng ký khám và tạo một hồ sơ khám.</p>	Khách hàng	<p>Người dùng 2 (Nhân viên tiếp nhận): Nhân viên B đọc thông tin đăng ký và bắt đầu tiếp nhận bệnh nhân tại thời điểm T1. Trước khi giao dịch của Khách hàng A được commit, nếu có lỗi và rollback xảy ra, Nhân viên B đã đọc thông tin không chính xác.</p>	Nhân viên	<p>Dirty read xảy ra khi giao tác 1 (Khách hàng) đã commit thông tin đăng ký nhưng chưa được commit. Nếu có lỗi và rollback xảy ra sau đó, thông tin đã được rollback thành không chính xác nhưng giao tác 2 (Nhân viên) đã đọc được commit.</p> <p>=> DIRTY READ</p>
<p>dùng 1 (Khách hàng): Khách hàng đăng ký khám và tạo một hồ sơ khám.</p>	Khách hàng	<p>Người dùng 2 (Nhân viên tiếp nhận): Nhân viên C đọc thông tin đăng ký và bắt đầu tiếp nhận bệnh nhân tại thời điểm T2. Trong trường hợp lỗi và rollback xảy ra trước khi giao dịch của khách hàng B được commit, nhân viên C đã đọc thông tin không chính xác.</p>	Nhân viên	<p>Dirty read xảy ra khi giao tác 1 (Khách hàng) đã commit thông tin đăng ký nhưng chưa được commit. Nếu có lỗi và rollback xảy ra sau đó, thông tin đã được rollback thành không chính xác nhưng giao tác 2 (Nhân viên) đã đọc được commit.</p> <p>=> DIRTY READ</p>

III. Thiết kế giao diện chức năng

1. Dùng chung

1.1. Đăng nhập



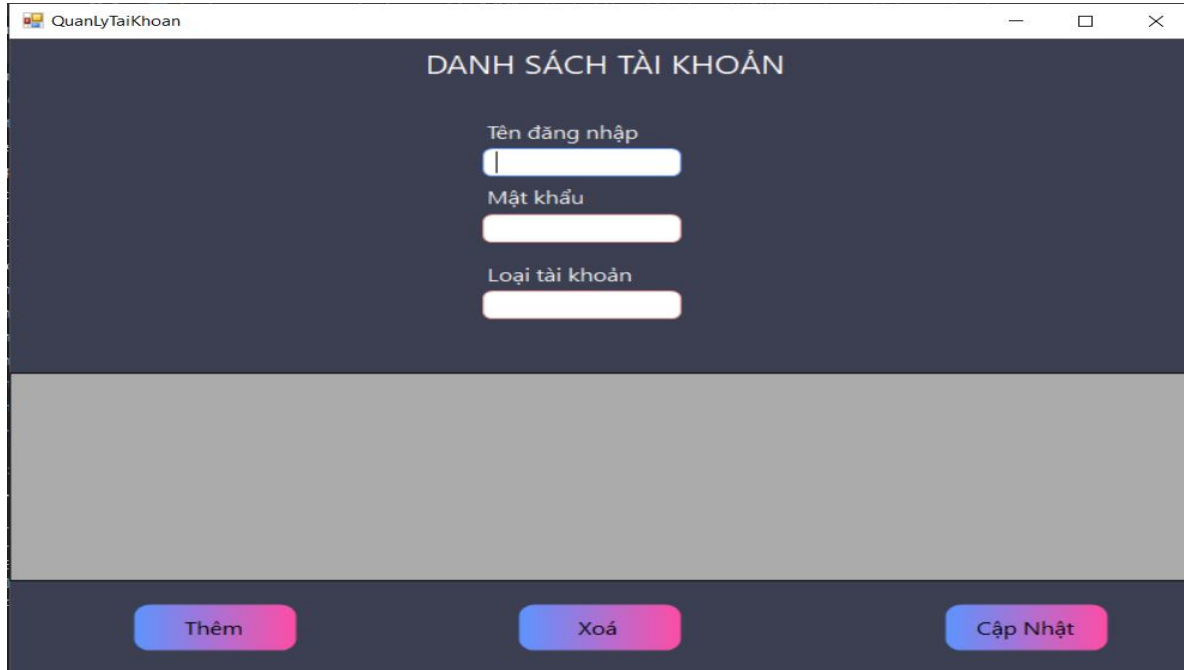
- Để sử dụng dịch vụ thì người dùng cần ghi vào các thông tin người dùng đã đăng ký rồi bấm đăng nhập

2. Admin

2.1. Thông tin tài khoản

- Quản trị viên có thể sửa thông tin cá nhân của chính bản thân mình
- Quản trị viên có thể đổi mật khẩu của tài khoản mình

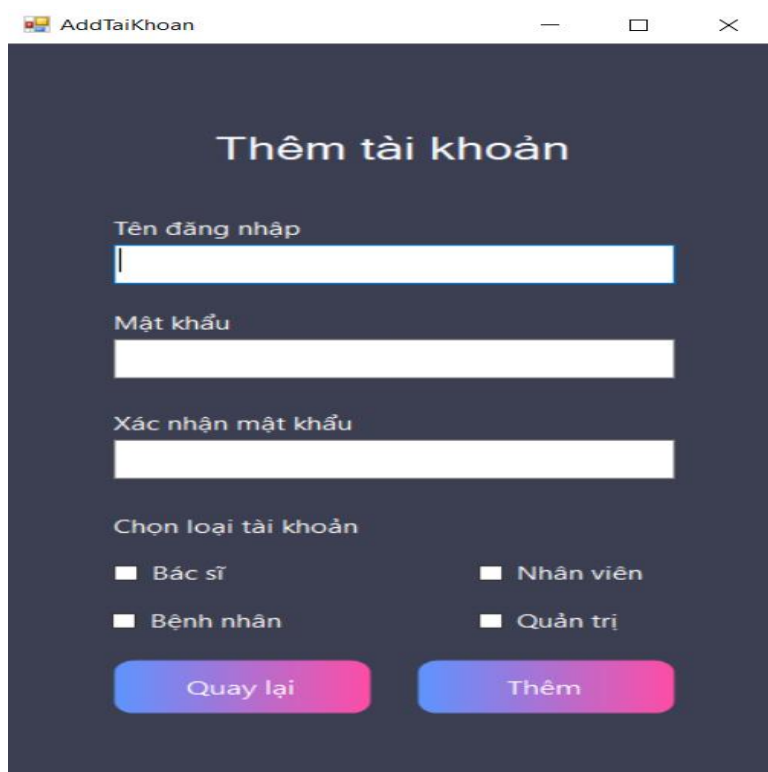
2.2. Quản lý tài khoản



The screenshot shows a web application window titled "QuanLyTaiKhoan". The main content area has a dark blue background with the title "DANH SÁCH TÀI KHOẢN" in white. Below the title, there are three input fields with labels: "Tên đăng nhập", "Mật khẩu", and "Loại tài khoản". Below these fields is a large gray rectangular area, likely a placeholder for a table or list of accounts. At the bottom of the interface, there are three buttons: "Thêm", "Xóa", and "Cập Nhật".

- Quản trị viên có thể thêm tài khoản của bất kỳ loại nhân viên nào
- Quản Trị viên có thể xóa tài khoản của bất kỳ thành viên nào
- Quản trị viên có thể cập nhật lại thông tin tài khoản của bất kỳ thành viên nào

2.3. Thêm tài khoản



- Quản trị viên có thể thêm bất kỳ loại tài khoản

2.4. Quản lý thuốc



- Quản trị có thể xem danh sách thuốc
- Quản trị có thể thêm bất kỳ thuốc nào
- Quản trị có thể cập nhật bất kỳ thuốc nào
- Quản trị có thể xóa bất kỳ thuốc nào

2.5. Quản lý dịch vụ

- Quản trị có thể thêm thông tin dịch vụ
- Quản trị có thể xóa các thông tin dịch vụ
- Quản trị có thể sửa lại thông tin dịch vụ
- Quản trị có thể xem danh sách dịch vụ

3. Bác sĩ

3.1. Quản lý hồ sơ bệnh án

	MaHS	NgayKham	NguoiKham	MaKH	MaNS
1	1/10/2023	KH001	2	2	
2	2/15/2023	KH002	2	5	
3	3/22/2023	KH003	3	1	
4	4/18/2023	KH004	4	4	
5	5/25/2023	KH005	5	3	

- Bác sĩ có thể cập nhật, xóa, thêm một hồ sơ bệnh án do chính mình quản lý

- Bác sĩ có thể tìm kiếm hồ sơ bệnh án bằng cách nhập mã hồ sơ để có thể tìm hồ sơ mà mình muốn tìm
- Khi bác sĩ bấm vào bảng thông tin sẽ hiện thông tin hồ sơ lên các textbox có sẵn

3.2. Thông tin cá nhân

The screenshot shows the 'MenuBacSi' application interface. On the left is a sidebar with icons for 'Quản lý hồ sơ' (Manage Records), 'Tài khoản' (Account), 'Thêm hồ sơ' (Add Record), and 'Quản lý lịch hẹn' (Manage Appointments). The main area is divided into two sections: 'Thông tin cá nhân' (Personal Information) and 'Đổi mật khẩu' (Change Password).

Thông tin cá nhân fields:

- Mã nha sĩ: 1
- Họ tên: Tran Van A
- Ngày sinh: 5/15/1990 12:00:00 AM
- Địa chỉ: Ha Noi
- Số điện thoại: 987654321
- Email: tva@gmail.com

Đổi mật khẩu fields:

- Mật khẩu cũ: (empty)
- Mật khẩu mới: (empty)
- Nhập lại mật khẩu: (empty)

Buttons: 'Sửa' (Edit) and 'Đổi mật khẩu' (Change Password).

- Bác sĩ có thể sửa thông tin cá nhân của chính bản thân mình
- Bác sĩ có thể đổi mật khẩu của tài khoản mình

3.3. Thêm hồ sơ

The screenshot shows the 'MenuBacSi' application interface for adding a new record. The sidebar is the same as in the previous section. The main area contains the 'Thêm hồ sơ' (Add Record) form.

Thêm hồ sơ fields:

- Mã khách hàng: (empty)
- Bệnh nhân: (empty)
- Ngày Khám: 08-29-2023

Table of Medications:

MaDV_Thuoc	TenDV	TenThuoc	LoaiDV	ThanhTien	MaHD	MaHS
DV001	Kham rang...	Khong su d...	Dich vu nh...	500000	2	3
T000	Lay tuy...	Khong su d...	Dich vu nh...	1200000	2	2
T000	Tram rang...	Khong su d...	Dich vu nh...	300000	3	3
DV010	Thuoc uong...	Paracetamol	Thuoc	30000	4	4
T002	Thuoc uong...	Ibuprofen	Thuoc	50000	5	5
DV005	Trong rang...	Khong su d...	Dich vu nh...	1500000	1	1

Table of Medications (Bottom):

Checkbox	MaThuoc	TenThuoc	DonViTin	Chu Dinh	SLTonKho	NgayHet	Tien
<input type="checkbox"/>	T000	Khong ...	Khong	Khong	0	1/1/2090	100
<input type="checkbox"/>	T001	Paracet...	Viên	U'ng	100	12/31/...	100
<input type="checkbox"/>	T002	Ibuprofen	Viên	U'ng	80	11/30/...	100
<input type="checkbox"/>	T003	Amoxicl...	Viên	U'ng	120	5/31/2...	100
<input type="checkbox"/>	T004	Aspirin	Viên	U'ng	90	10/15/...	100
<input type="checkbox"/>	T005	Loratad...	Viên	U'ng	60	3/20/2...	100
<input type="checkbox"/>	T006	Omepra...	Viên	U'ng	150	9/25/2...	100
<input type="checkbox"/>	T007	Cetizine	Viên	U'ng	110	1/5/2024	100

Table of Medications (Bottom Right):

Checkbox	MaDV	TenDV	Loai	Tien
<input type="checkbox"/>	DV000	Khong su dun...	Khong	100
<input type="checkbox"/>	DV001	Kham rang to...	Dich vu nhanh	100
<input type="checkbox"/>	DV002	Lay tuy	Dich vu nhanh	100
<input type="checkbox"/>	DV003	Nho rang khon	Dich vu nhanh	100
<input type="checkbox"/>	DV004	Tram rang	Dich vu nhanh	100
<input type="checkbox"/>	DV005	Trong rang su	Dich vu nhanh	100
<input type="checkbox"/>	DV006	Chup X-quang...	Dich vu nhanh	100
<input type="checkbox"/>	DV007	Nho rang thu...	Dich vu nhanh	100
<input type="checkbox"/>	DV008	Lay voi rang	Dich vu nhanh	100
<input type="checkbox"/>	DV009	Lo trinh lay tra...	Lo trinh	100
<input type="checkbox"/>	DV010	Thuoc uong	Thuoc	100

Buttons: 'Thêm' (Add) and 'Đăng xuất' (Logout).

- Bác sĩ có thể thêm hồ sơ bệnh án của khách hàng
- Bác sĩ xem được thông tin các loại thuốc và dịch vụ mà phòng khám cung cấp để sử dụng cho bệnh nhân
- Bác sĩ thêm những dịch vụ và thuốc của khách hàng sử dụng vào chi tiết hồ sơ của bệnh nhân

3.4. Cập nhật hồ sơ bệnh án

- Xem thông tin chi tiết hồ sơ và cập nhật thông tin của hồ sơ

3.5. Quản lý lịch hẹn

	MaLichCaNhan	Ngay	ThoiGian	MaNS
9		8/29/2023	03:00:00	7
10		7/31/2023	00:00:00	7

- Bác sĩ có thể thêm mới một lịch hẹn mà mình rảnh
- Bác sĩ có thể cập nhật, xóa một lịch hẹn mình đã tạo

4. Nhân viên

4.1. Đăng ký khách hàng

The screenshot shows a web form titled "Đăng ký khách hàng mới" (New Customer Registration). The form is divided into two columns. The left column contains fields for "Họ và tên:" (Last name and first name), "Ngày, tháng năm sinh:" (Date of birth) with a date picker showing "08-29-2023", and "Địa chỉ:" (Address). The right column contains fields for "Số ĐT:" (Phone number), "Mật khẩu:" (Password), and "Email:". At the bottom, there are two buttons: "Quay lại" (Go back) on the left and "Xác nhận" (Confirm) on the right.

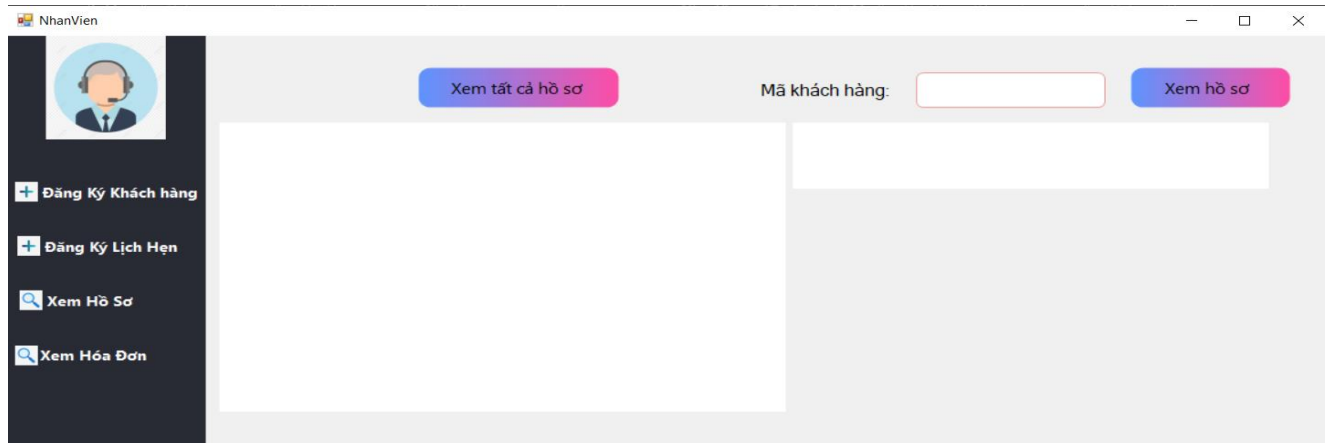
- Nhân viên có thể đăng ký thông tin cho khách hàng

4.2. Đăng ký lịch hẹn khách hàng

The screenshot shows a web form titled "Đăng ký lịch hẹn với nha sĩ" (Register appointment with dentist). The form has two main sections. The top section contains fields for "Ngày Khám:" (Date of examination) with a date picker showing "08-29-2023" and "Giờ Khám:" (Time of examination) with a time picker showing "12:00:00 AM". There are also "Tìm kiếm" (Search), "Quay lại" (Go back), and "Tạo lịch hẹn" (Create appointment) buttons. The bottom section contains two lists: "Danh sách nha sĩ phù hợp với thời gian:" (List of dentists suitable for the time) and "Danh sách dịch vụ" (List of services).

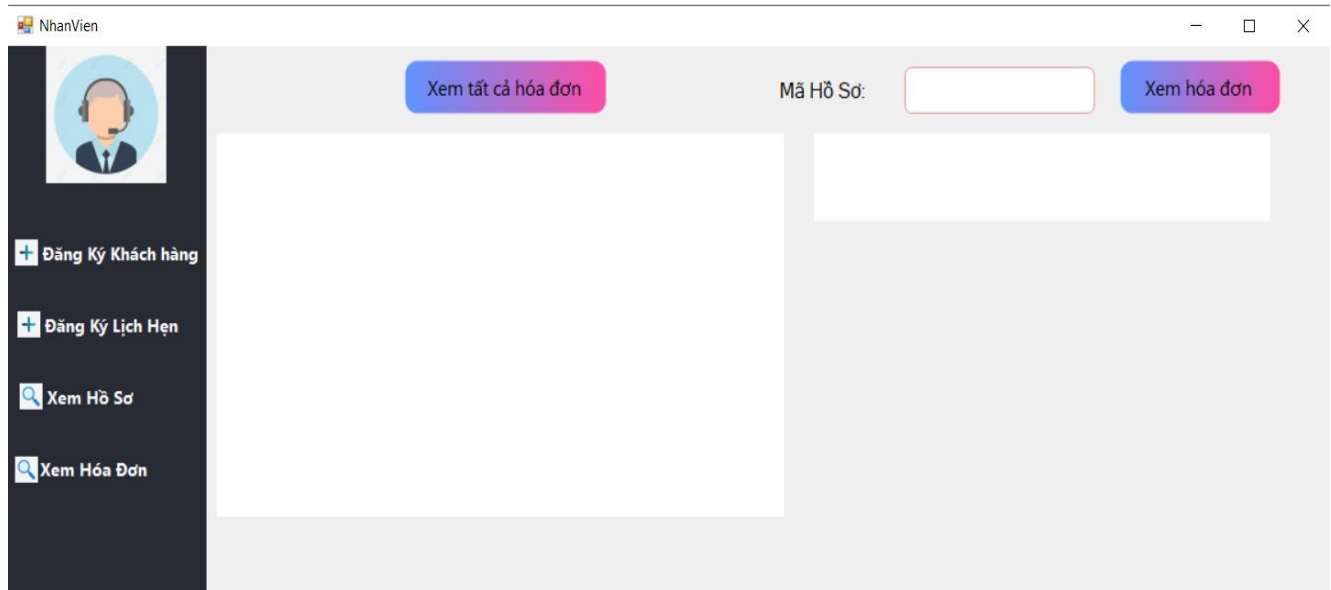
- Tìm kiếm lịch hẹn của nha sĩ phù hợp với khách hàng
- Xem danh sách các lịch hẹn của nha sĩ phù hợp với thời gian
- Xem danh sách các lịch hẹn
- Tạo một lịch hẹn mới

4.3. Xem hồ sơ



- Nhân viên có thể xem tất cả hồ sơ của khách hàng
- Nhân viên có thể tìm kiếm hồ sơ của khách hàng theo mã khách hàng

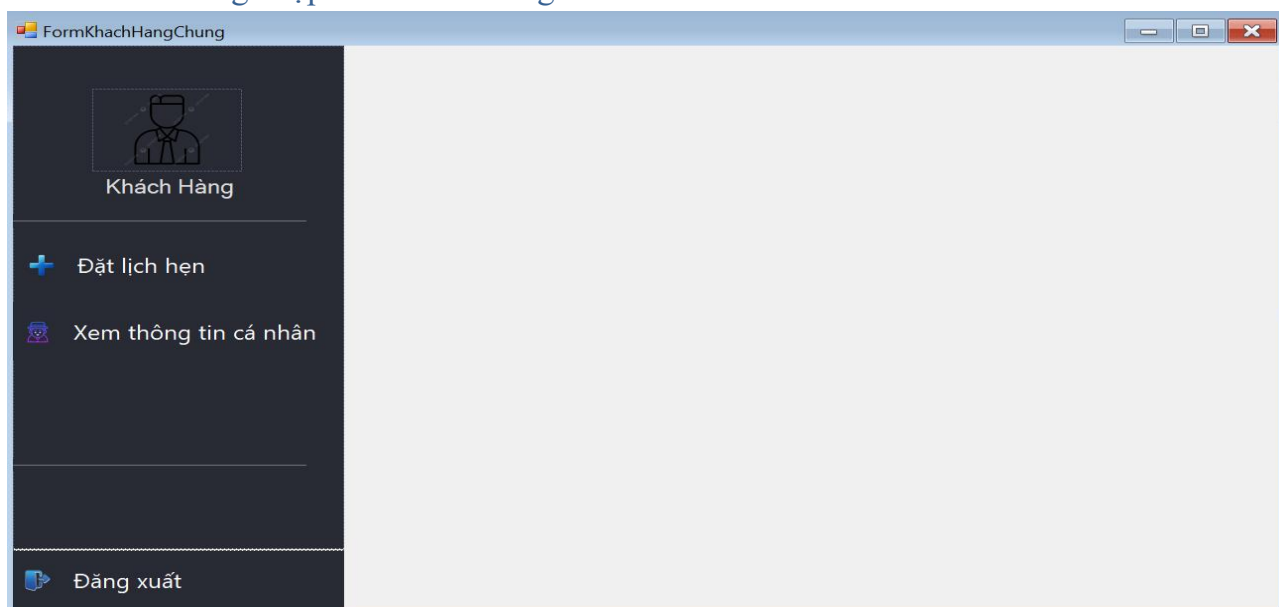
4.4. Xem hóa đơn



- Nhân viên có thể xem tất cả hóa đơn của khách hàng
- Nhân viên có thể tìm kiếm hóa đơn của khách hàng theo mã hồ sơ

5. Bệnh nhân

5.1. Sau khi đăng nhập role khách hàng



- Menu khi khách hàng đăng nhập vào ứng dụng

5.2. Đăng ký khách hàng mới (Sau khi click vào nút Sign up ở giao diện đăng nhập)

The screenshot shows a web application window titled 'DangKyKhachHang'. The form is titled 'Đăng ký khách hàng mới'. It contains several input fields: 'Họ và tên :', 'SĐT :', 'Ngày, tháng năm sinh :', 'Mật khẩu', 'Địa chỉ :', and 'Email :'. The 'Ngày, tháng năm sinh :' field has a date picker showing '08-29-2023'. At the bottom, there are two buttons: 'Quay lại' (Back) and 'Xác nhận' (Confirm).

- Khách hàng có thể đăng ký thông tin cá nhân của mình

5.3. Đặt lịch hẹn khi click vào đặt lịch hẹn

The screenshot shows a web application window titled "DatLichHen". The main heading is "Đăng ký lịch hẹn với nha sĩ" (Register appointment with dentist). Below this, there are two input fields: "Ngày Khám:" (Exam Date) with a date picker set to "08-29-2023" and "Giờ Khám:" (Exam Time) with a time picker set to "12:00:00 AM". To the right of these fields is a pink button labeled "Tìm kiếm" (Search). Below the search fields, there are two buttons: a pink "Quay lại" (Go back) button on the left and a pink "Tạo lịch hẹn" (Create appointment) button on the right. At the bottom, there are two sections: "Danh sách nha sĩ phù hợp với thời gian :" (List of dentists suitable for the time) and "Danh sách dịch vụ" (List of services), both of which are currently empty.

- Tìm kiếm lịch hẹn của nha sĩ phù hợp với khách hàng
- Xem danh sách các lịch hẹn của nha sĩ phù hợp với thời gian
- Tạo một lịch hẹn mới

5.4. Cửa sổ xem thông tin khách hàng sau khi click vào button xem thông tin cá nhân

The screenshot shows a web application window titled "XemThongTinCaNhan". The main heading is "Thông tin cá nhân khách hàng" (Patient personal information). Below this, there are several input fields: "Họ và tên :" (Last name and first name) with a text box, "Số ĐT" (Phone number) with a text box, "Ngày, tháng, năm sinh" (Date of birth, month, year) with a date picker set to "08-29-2023", "Email" with a text box, and "Địa chỉ" (Address) with a text box. At the bottom, there are three buttons: a pink "Quay lại" (Go back) button on the left, a pink "Cập nhật và lưu" (Update and save) button in the center, and a pink "Xem hồ sơ bệnh án" (View medical record) button on the right.

- Khách hàng có thể cập nhật thông tin cá nhân của chính bản thân mình
- Khách hàng có thể xem hồ sơ bệnh án của bản thân

CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Dương Hiền Lê Hoàng

1. Tình huống 1: Nhân viên A update thông tin khách hàng B thì trong lúc đó nhân viên B truy vấn thông tin khách hàng B để xem thông tin khách hàng B

ERR01: Dirtyread			
T1 (User = Nhân viên): thực hiện cập nhật 1 thông tin khách hàng B.			
T2 (User = Nhân viên): thực hiện xem thông tin khách hàng B			
UpdatedThôngTinKH	Khóa	LayThôngTinKH	Khóa
<u>Input:</u> @MaKH @SDTKH <u>Output:</u>		<u>Input:</u> @MaKH <u>Output:</u>	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) khách hàng IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với đk		

PRINT N'Không tồn tại khách hàng đó' ROLLBACK TRAN RETURN 0 END	MaKH = @MaKH.		
B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE KHACHHANG SET SDTKH = @SDTKH Where MaKH = @MaKH;	U,X(KHACHH ANG) //Xin khóa update trên bảng KHACHHAN G với dk MaKH = @MaKH		
WAITFOR DELAY '0:0:5';			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ UNCOMMITTED	Cho phép đọc dữ liệu đang được thực

			hiện bởi các giao dịch khác mà chưa được commit.
		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <pre> IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn tại khách hàng đó' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		<p>B2: Xem thông tin khách hàng</p> <pre> SELECT * FROM KHACHHANG Where MaKH = @MaKH; </pre>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>

		COMMIT	
<p>IF(@SDTKH > 999999999 OR @SDTKH < 100000000)</p> <p>BEGIN</p> <p> ROLLBACK TRAN</p> <p>END</p>			

2. Tình huống 2: Quản trị A cập nhật lại thông tin hồ sơ của khách hàng B thì trong lúc đó nhân viên C truy vấn thông tin khách hàng B

ERR01: UNREPEATABLE READ

T1 (User = Quản trị): thực hiện cập nhật 1 thông tin hồ sơ của khách hàng B.

T2 (User = Nhân viên): thực hiện xem thông tin hồ sơ của khách hàng B

UpdatedThongTinHoSo	Khóa	LayThongTinHoSo	Khóa
<p><u>Input:</u> @MaHS</p> <p>@NgayKham</p> <p>@MaKH</p> <p><u>Output:</u></p>		<p><u>Input:</u> @MaHS</p> <p><u>Output:</u></p>	

		BEGIN TRAN	
		B1: Kiểm tra thông tin khách hàng có tồn tại không IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS) BEGIN PRINT N'Không tồn tại hồ sơ đó' ROLLBACK TRAN RETURN 0 END	S(HoSoBN) //Xin khoá đọc trên bảng HoSoBN với đk MaHS = @MaHS.
		WAITFOR DELAY '0:0:5';	
BEGIN TRAN			
B1: Kiểm tra thông tin hồ sơ khách hàng IF EXISTS (SELECT * FROM HoSoBN WHERE	S(HoSoBN) //Xin khoá đọc trên bảng		

<p>MaHS = (@MaHS)</p> <p>BEGIN</p> <p>PRINT N'Không tồn tại hồ sơ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>HoSoBN với đk MaHS = @MaHS.</p>		
<p>B2: Cập nhật thông tin vào bảng KHACHHANG</p> <p>UPDATE HoSoBN SET NgayKham = @NgayKham, MaKH = @MaKH Where MaHS = @MaHS;</p>	<p>U,X(HoSoBN)</p> <p>//Xin khoá trên bảng HoSoBN với đk MaHS = @MaHS.</p>		
<p>WAITFOR DELAY '0:0:5';</p>			
		<p>B2: Xem thông tin hồ sơ khách hàng</p>	<p>S(HoSoBN)</p> <p>//Xin khoá đọc</p>

		SELECT * FROM HoSoKH Where MaKH = @MaKH;	trên bảng HoSoBN với đk MaKH = @MaKH.
COMMIT		COMMIT	

II. Sinh viên thực hiện: Đinh Thành Danh

- Tình huống 1: Khách hàng A đăng nhập vào hệ thống chọn việc đặt lịch hẹn, trong khi đó nha sĩ B cũng đăng nhập vào hệ thống hiển thị lịch hẹn cũ

ERR01: PHANTOM READ			
T1 (User = Khách hàng):Khách hàng 1 thực hiện việc đặt lịch hẹn			
T2 (User = Nha sĩ): Thực hiện xem danh sách cuộc hẹn			
Sp_ThemLichHen_1	Khóa	Sp_XemLichHen_2	Khóa

<p><u>Input:</u></p> <p><u>@NgayHen</u></p> <p><u>@ThoiGianHen</u></p> <p><u>@HoTenKH</u></p> <p><u>@MaKH</u></p> <p><u>@MaNS</u></p> <p><u>Output:</u> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</p>		<p><u>Input:</u></p>	
		<p>BEGIN TRAN</p>	

		<p>B1: Xem danh sách lịch hẹn:</p> <p>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</p> <p>return 1</p>	<p>S(LichCaNhanNS)</p> <p>//Xin khoá đọc trên bảng LichCaNhanNS</p>
		WAITFOR DELAY '0:0:10'	
BEGIN TRAN			

B1: Kiểm tra ngày và thời gian lịch hẹn xem có chưa ?

```
IF(EXISTS(SELECT * FROM LichHen WHERE  
Ngày = '01-10-2023' AND ThoiGian = '08:30:00')
```

```
RETURN -1
```

S(LichHen)

//Xin khoá đọc

trên bảng

LichHen với

điều kiện

Ngày = '01-10-2023'
và ThoiGian
='08:30:00'

<p>B2: Kiểm tra mã nha sĩ xem có chưa ?</p> <pre> IF(EXISTS(SELECT * FROM LichHen WHERE MaNS = '1') RETURN -1 </pre>	<p>S(LichHen)</p> <p>//Xin khoá đọc trên bảng</p> <p>LichHen với điều kiện MaNS = '1'</p>	
<p>B3 : Thêm lịch hen mới</p> <pre> INSERT INTO LichHen(NgayHen, ThoiGianHen, TenKH, MaKH, MaNS) VALUES ('01-10-2023', '08:30:00', N'Nguyen Van E', '3', '2') return 1 </pre>	<p>X(LichHen) //xin khoá ghi trên bảng LichHen để thêm lịch hen mới</p>	
<p>COMMIT</p>		

		<p>B1: Xem danh sách lịch hẹn:</p> <p>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</p> <p>return 1</p>	<p>S(LichCaNhanNS)</p> <p>//Xin khoá đọc trên bảng LichCaNhanNS</p>
		COMMIT	

2. Tình huống 2: Trong hệ thống, khách hàng đang cập nhật thông tin cá nhân thì nhân viên xem thông tin của khách hàng đó.

<p>ERR02: DIRTY READ</p> <p>T1 (User = Khách hàng): Khách hàng đang thực hiện cập nhật thông tin</p> <p>T2 (User = Nhân viên): Nhân viên đang xem thông tin cũ của khách hàng</p>			
Sp_XemThongTinCaNhan	Khóa	Sp_XemThongTinKhachHang	Khóa

<p><u>Input:</u></p> <p><u>@HoTenKH</u></p> <p><u>@NgaySinhKH</u></p> <p><u>@DiaChiKH</u></p> <p><u>@SDTKH</u></p> <p><u>@EmailKH</u></p> <p><u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>		<p><u>Input:</u></p>	
BEGIN TRAN			

<p>B1: Kiểm tra thông tin (1) khách hàng</p> <pre> IF EXISTS (SELECT * FROM KHACHHANG WHERE SDTKH = @SDTKH) BEGIN PRINT N'Không tồn tại tài khoản' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(KHACHHANG)</p> <pre> //Xin khoá đọc trên bảng KHACHHANG với đk SDTKH = @SDTKH </pre>		
<p>B2: Cập nhật thông tin vào bảng KHACHHANG</p> <pre> UPDATE KHACHHANG SET EmailKH = @EmailKH Where SDTKH = @SDTKH; </pre>	<p>U,X(KHACHHANG)</p> <pre> //Xin khóa update trên bảng KHACHHANG với đk SDTKH = @SDTKH </pre>		

WAITFOR DELAY '0:0:5';			
		BEGIN TRAN	
		SET TRAN ISOLATION LEVEL READ UNCOMMITTED	Cho phép đọc dữ liệu đang được thực hiện bởi các giao dịch khác mà chưa được commit

		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <pre> IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn tại khách hàng đó' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		<p>B2: Xem thông tin khách hàng</p> <pre> SELECT * FROM KHACHHANG Where MaKH = @MaKH; </pre>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.</p>
		COMMIT	

III. Sinh viên thực hiện: Nguyễn Hoài Mẫn

1. Tình huống 1: Quản trị viên A đăng nhập vào hệ thống để cập nhật thông tin của một dịch vụ, chẳng hạn như điều chỉnh giá hoặc thay đổi mô tả dịch vụ. Trong khi đó, Quản trị viên B cũng đăng nhập và chọn cùng một dịch vụ để cập nhật thông tin.

ERR01: LOST UPDATE			
T1 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 150			
T2 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 200			
Sp_UpdateDichVu_1	Khóa	Sp_UpdateDichVu_2	Khóa
<u>Input:</u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>		<u>Input:</u> <u>@MaDV = 'DV001'</u> <u>@TenDV = 'Kham rang tong quat'</u> <u>@Loai = 'Dich vu nhanh'</u> <u>@Tien = '150'</u> <u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>	

BEGIN TRAN			
<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <p>IF NOT EXISTS (SELECT *</p> <p>FROM DICHVU</p> <p>WHERE MADV = 'DV001')</p> <p>BEGIN</p> <p>PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(DICHVU)</p> <p>//Xin khoá đọc</p> <p>trên bảng</p> <p>DICHVU với</p> <p>điều kiện</p> <p>MADV = 'DV001'</p>		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	

		<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <pre>IF NOT EXISTS (SELECT * FROM DICHVU WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END</pre>	<p>S(DICHVU)</p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001'</p>
--	--	--	---

		<p>B2: Cập nhật thông tin thuộc tính TenDV, Loai, Tien trong bảng DICHVU thành 'Kham rang tong quat', 'Dich vu nhanh', 200</p> <p>UPDATE DichVu SET TenDV = @TenDV, Loai = @Loai, Tien = @Tien WHERE MaDV = @MaDV</p>	<p>U,X(DICHVU) //Xin khoá update trên bảng DICHVU với điều kiện MADV = 'DV001'</p>
		COMMIT	

<p>B2: Cập nhật thông tin thuộc tính</p> <p>TenDV, Loai, Tien trong bảng DICHVU</p> <p>thành 'Kham rang tong quat', 'Dich vu nhanh', 150</p> <p>UPDATE DichVu SET</p> <p>TenDV = @TenDV,</p> <p>Loai = @Loai,</p> <p>Tien = @Tien</p> <p>WHERE MaDV = @MaDV</p>	<p>U,X(DICHVU)</p> <p>//Xin khoá update</p> <p>trên bảng DICHVU</p> <p>với điều kiện</p> <p>MADV = 'DV001'</p>		
COMMIT			

2. Tình huống 2: Trong hệ thống, có một nha sĩ vừa đăng nhập xong thì admin tiến hành đổi mật khẩu của tài khoản bác sĩ đó. Dẫn đến việc sau khi đăng nhập, bác sĩ đó không thể xử lý lấy được thông tin tài khoản của mình.

ERR02: UNREPEATABLE READ

T1 (User = Nha sĩ): Nha sĩ tiến hành đăng nhập vào hệ thống với tên đăng nhập là : “11111112”,
mật khẩu là : “12345”

T2 (User = admin): admin tiến hành đổi mật khẩu của tài khoản nha sĩ có mã số 1 từ “12345” thành “54321”

Sp_NV_LayTongTinTK	Khóa	Sp_NV_DoiMK	Khóa
<p><u>Input:</u></p> <p><u>@SDT= '11111112'</u></p> <p><u>@MATKHAU = '12345'</u></p> <p><u>@LoaiTK = '1'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>		<p><u>Input:</u></p> <p><u>@SDT = '11111112',</u></p> <p><u>@MATKHAU = '54321'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>	
BEGIN TRAN			

B1: Xử lý lấy thông tin mã tài khoản nhà sĩ SELECT @Ma = NS.MaNS FROM TaiKhoan A INNER JOIN NhaSi NS ON A.SDT = NS.SDTNS WHERE A.SDT = <u>'111111112'</u> AND A.MatKhai = <u>'12345'</u> AND A.LoaiTK = <u>'1'</u>	S(TAIKHOAN) //Xin khoá đọc trên bảng TaiKhoan với điều kiện TENDANGNHAP = '111111112', MATKHAU = '12345', LoaiTK = '1'		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	

		<p>B1: Kiểm tra xem tài khoản có tồn tại hay không?</p> <pre> IF NOT EXISTS (SELECT * FROM TAIKHOAN WHERE SDT = '11111112') BEGIN PRINT 'Tài khoản Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(TAIKHOAN)</p> <p>//Xin khoá đọc trên bảng TAIKHOAN với điều kiện SDT = '11111112'</p>
		<p>B2: Xử lý cập nhật mật khẩu</p> <pre> UPDATE TAIKHOAN SET MATKHAU = '54321' WHERE SDT = '11111112' </pre>	<p>U,X(TAIKHOAN)</p> <p>//Xin khoá update trên bảng TAIKHOAN với điều kiện SDT = '11111112'</p>

<p>B2: Kiểm tra xem mã tài khoản có tồn tại hay không?</p> <pre> IF @Ma IS NULL BEGIN PRINT N'Nhà sĩ Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>			
<p>B3: Xử lý lấy thông tin của tài khoản?</p> <pre> SELECT * FROM NhaSi NS WHERE NS.MaNS = @Ma; </pre>	<p>S(ACCOUNT), S(NHANVIEN)</p> <p>//Xin khoá đọc trên bảng NHASI, với điều kiện NS.MaNS = @Ma</p>		
COMMIT		COMMIT	

IV. Sinh viên thực hiện: Nguyễn Phụng Khanh

1. Tình huống 1: Quản trị viên A cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ, trong lúc đó quản trị viên B cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ

ERR01: LOST UPDATE

T1 (User = Quản trị viên A): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ

T2 (User = Quản trị viên B): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ

UpdatedSoLuongTonKhoThuoc_1	Khóa	UpdatedSoLuongTonKhoThuoc_2	Khóa
<u>Input:</u> @SLTonKho		<u>Input:</u> @SLTonKho	
<u>Output:</u>		<u>Output:</u>	
BEGIN TRAN			
<p>B1: Kiểm tra thông tin thuốc có tồn tại hay không</p> <p>IF EXISTS (SELECT * FROM THUOC WHERE MaThuoc = @MaThuoc)</p> <p>BEGIN</p> <p>PRINT N 'Không tồn tại loại thuốc đó'</p> <p>ROLLBACK TRAN</p>	<p>S(THUOC)</p> <p>//Xin khoá đọc trên bảng THUOC với đk MaThuoc = @MaThuoc.</p>		

RETURN 0			
END			
<p>B2: Cập nhật thông tin SLTonKho vào bảng THUOC</p> <p>UPDATE THUOC</p> <p>SET @SLTonKho = 20</p> <p>where MaThuoc = @MaThuoc;</p>	<p>U,X(THUOC)</p> <p>//Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc</p>		
WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin thuốc có tồn tại hay không</p> <p>IF EXISTS (SELECT * FROM THUOC WHERE MaThuoc = @MaThuoc)</p> <p>BEGIN</p>	<p>S(THUOC)</p> <p>//Xin khoá đọc trên bảng THUOC với dk MaThuoc = @MaThuoc.</p>

		PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0 END	
		B2: Cập nhật thông tin SLTonKho vào bảng THUOC UPDATE THUOC SET @SLTonKho = 15 where MaThuoc = @MaThuoc;	U,X(THUOC) //Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc
		COMMIT	
COMMIT			

2. Tình huống 2: Nhân viên xem số điện thoại của nhà sĩ A để liên lạc, cùng lúc đó nhà sĩ A cập nhật số điện thoại cá nhân của mình từ 123456789 thành 987654321 và lưu thay đổi thành công. Sau đó nhân viên load lại thông tin, đọc được số điện thoại mới.

ERR01: UNREPEATEABLE READ

T1 (User = Nhân viên): thực hiện xem số điện thoại của nhà sĩ A để liên lạc.

T2 (User = Nhà sĩ): thực hiện cập nhật số điện thoại cá nhân từ 123456789 thành 987654321 và lưu thay đổi.

LaySoDienThoaiNS	Khóa	UpdatedSoDienThoaiNS	Khóa
<u>Input:</u> @SDTNS		<u>Input:</u> @SDTNS='987654321'	
<u>Output:</u> @SDTNS='123456789'		<u>Output:</u> @SDTNS='123456789'	
BEGIN TRAN			
B1: Kiểm tra thông tin nhà sĩ có tồn tại hay không IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS) BEGIN PRINT N 'Không tồn tại nhà sĩ đó' ROLLBACK TRAN	S(NhaSi) //Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.		

RETURN 0			
END			
<p>B2: Xem thông tin số điện thoại nha sĩ</p> <p>SELECT SDTNS</p> <p>FROM NhaSi</p> <p>WHERE MaNS = @MaNS;</p>	<p>S(NhaSi)</p> <p>//Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.</p>		
WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin nha sĩ có tồn tại hay không</p> <p>IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS)</p> <p>BEGIN</p> <p>PRINT N 'Không tồn tại nha sĩ đó'</p>	<p>S(NhaSi)</p> <p>//Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.</p>

		ROLLBACK TRAN RETURN 0 END	
		B2: Cập nhật số điện thoại nha sĩ UPDATE NhaSi SET @SDTNS = 123456789 where MaNS = @MaNS;	U,X(NhaSi) //Xin khóa update trên bảng Nha Si với dk MaNS = @MaNS
B3: Xem thông tin số điện thoại nha sĩ _ lần 2 SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	S(NhaSi) //Xin khoá đọc trên bảng NhaSi với dk MaNS = @MaNS.		
COMMIT			
		COMMIT	

V. Giao tác một số chức năng:

--Giao tác thêm khách hàng và hồ sơ bệnh nhân mới

BEGIN TRANSACTION;

-- Thêm khách hàng mới

INSERT INTO KHACHHANG (HoTenKH, NgaySinhKH, DiaChiKH, SDTKH, EmailKH, MaNV)

VALUES ('Nguyen Van X', '1990-01-01', 'TP.HCM', 123456789, 'nguyenvanx@example.com', 1);

-- Lấy mã khách hàng vừa thêm

DECLARE @MaKH INT;

SET @MaKH = SCOPE_IDENTITY();

-- Thêm hồ sơ bệnh nhân mới

INSERT INTO HoSoBN (NgayKham, NguoiKham, MaKH, MaNS)

VALUES ('2024-01-03', 'NV001', @MaKH, 2);

COMMIT;

--Giao tác cập nhật thông tin khách hàng:

BEGIN TRANSACTION;

-- Cập nhật thông tin khách hàng

UPDATE KHACHHANG

SET DiaChiKH = '135B Trần Hưng Đạo, 1, TPHCM'

WHERE MaKH = 1;

COMMIT;

--Giao tác xoá hồ sơ bệnh nhân và thông tin liên quan:

BEGIN TRANSACTION;

-- Lấy mã khách hàng và mã nhân sự từ hồ sơ bệnh nhân

DECLARE @MaKH INT, @MaNS INT;

SELECT @MaKH = MaKH, @MaNS = MaNS FROM HoSoBN WHERE MaHS = 1;

-- Xoá hồ sơ bệnh nhân

DELETE FROM HoSoBN WHERE MaHS = 1;

-- Xoá các liên quan trong các bảng khác

DELETE FROM LichHen WHERE MaKH = @MaKH OR MaNS = @MaNS;

-- Nếu cần, xoá khách hàng

DELETE FROM KHACHHANG WHERE MaKH = @MaKH;

COMMIT;

--Giao tác xoá dịch vụ và các liên kết trong chi tiết hóa đơn:

BEGIN TRANSACTION;

-- Lấy mã dịch vụ và xoá chi tiết hóa đơn liên quan

DECLARE @MaDV CHAR(5);

SELECT @MaDV = MaDV FROM DichVu WHERE MaDV = 'DV003';

DELETE FROM CT_HOADON WHERE MaDV_Thuoc = @MaDV;

-- Xoá dịch vụ

```
DELETE FROM DichVu WHERE MaDV = @MaDV;
```

```
COMMIT;
```

--Giao tác cập nhật số lượng tồn kho của một loại thuốc:

```
BEGIN TRANSACTION;
```

-- Cập nhật số lượng tồn kho của thuốc

```
UPDATE THUOC
```

```
SET SLTonKho = SLTonKho - 10
```

```
WHERE MaThuoc = 'T001';
```

```
COMMIT;
```

--Giao tác cập nhật thông tin một đơn hàng và chi tiết đơn hàng:

```
BEGIN TRANSACTION;
```

-- Cập nhật thông tin đơn hàng

UPDATE HOADON

SET PhiThanhToan = 1800000

WHERE MaHD = 1;

-- Cập nhật chi tiết đơn hàng

UPDATE CT_HOADON

SET SL = 5, ThanhTien = 900000

WHERE MaHD = 1 AND MaDV_Thuoc = 'DV001';

COMMIT;

--Giao tác cập nhật lịch hẹn và thông tin bệnh nhân:

BEGIN TRANSACTION;

-- Cập nhật lịch hẹn

UPDATE LichHen

SET ThoiGianHen = '15:00:00'

WHERE MaLichHen = 2;

-- Cập nhật thông tin bệnh nhân

UPDATE KHACHHANG

SET HoTenKH = 'Tran Van F', NgaySinhKH = '1988-11-20'

WHERE MaKH = 5;

COMMIT;

--Giao tác xoá một khách hàng và liên kết liên quan:

BEGIN TRANSACTION;

-- Lấy mã khách hàng và xoá liên kết liên quan

DECLARE @MaKHToremove INT;

SELECT @MaKHToremove = MaKH FROM KHACHHANG WHERE HoTenKH = 'Nguyen Van E';

DELETE FROM LichHen WHERE MaKH = @MaKHToremove;

```
DELETE FROM HoSoBN WHERE MaKH = @MaKHToremove;
```

```
DELETE FROM KHACHHANG WHERE MaKH = @MaKHToremove;
```

```
COMMIT;
```

```
--Giao tác thêm một nhân viên mới và liên kết với tài khoản:
```

```
BEGIN TRANSACTION;
```

```
-- Thêm nhân viên mới
```

```
INSERT INTO NhanVien (HoTenNV, NgaySinhNV, DiaChiNV, SDTNV, EmailNV)
```

```
VALUES ('Nguyen Thi G', '1995-03-10', 'Vung Tau', 555555556, 'ntg@gmail.com');
```

```
-- Lấy thông tin nhân viên vừa thêm
```

```
DECLARE @MaNVToAdd INT, @SDTNVToAdd INT;
```

```
SELECT @MaNVToAdd = MaNV, @SDTNVToAdd = SDTNV FROM NhanVien WHERE SDTNV = 555555556;
```

```
-- Thêm tài khoản cho nhân viên
```

```
INSERT INTO TAIKHOAN (SDT, MatKhau, LoaiTK)
```



```
VALUES (@SDTNVToAdd, 'newpassword', '0');
```

```
COMMIT;
```

XỬ LÝ TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Dương Hiền Lê Hoàng

1. Tình huống 1: Nhân viên A update thông tin khách hàng B thì trong lúc đó nhân viên B truy vấn thông tin khách hàng B để xem thông tin khách hàng B

ERR01: Dirty read(FIX)			
T1 (User = Nhân viên): thực hiện cập nhật 1 thông tin khách hàng B.			
T2 (User = Nhân viên): thực hiện xem thông tin khách hàng B			
UpdatedThôngTinKH	Khóa	LayThôngTinKH	Khóa
<u>Input:</u> @MaKH @SDTKH <u>Output:</u>		<u>Input:</u> @MaKH <u>Output:</u>	
		SET TRAN ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) khách hàng IF EXISTS (SELECT * FROM KHACHHANG	S(KHACHHANG) //Xin khoá đọc		

WHERE MaKH = @MaKH) BEGIN PRINT N'Không tồn khách hàng đó' ROLLBACK TRAN RETURN 0 END	trên bảng KHACHHAN G với dk MaKH = @MaKH.		
B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE KHACHHANG SET SDTKH = @SDTKH Where MaKH = @MaKH;	U,X(KHACHH ANG) //Xin khóa update trên bảng KHACHHAN G với dk MaKH = @MaKH		
WAITFOR DELAY '0:0:10';			
		BEGIN TRAN	

		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <p>IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH)</p> <p>BEGIN</p> <p>PRINT N'Không tồn khách hàng đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(KHACHHANG)</p> <p>//Không cấp khóa cho S do UpdatedThong TinKH đang giữ khóa X</p>
		<p>B2: Xem thông tin khách hàng</p> <p>SELECT * FROM KHACHHANG Where MaKH = @MaKH;</p>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHAN G với đk MaKH = @MaKH.</p>
		COMMIT	
IF(@SDTKH > 999999999 OR @SDTKH < 100000000)			

BEGIN			
ROLLBACK TRAN			
END			

2. Tình huống 2: Quản trị A cập nhật lại thông tin hồ sơ của khách hàng B thì trong lúc đó nhân viên C truy vấn thông tin khách hàng B

ERR01: UNREPEATEABLE READ

T1 (User = Quản trị): thực hiện cập nhật 1 thông tin hồ sơ của khách hàng B.

T2 (User = Nhân viên): thực hiện xem thông tin hồ sơ của khách hàng B

UpdatedThôngTinHoSo	Khóa	LayThôngTinHoSo	Khóa
<u>Input:</u> @MaHS @NgayKham @MaKH <u>Output:</u>		<u>Input:</u> @MaHS <u>Output:</u>	
		SET TRAN ISOLATION LEVEL	Thiết lập khóa S khi đọc và

		REPEATABLE READ	giữ đến khi hết giao tác
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <p>IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS)</p> <p>BEGIN</p> <p>PRINT N'Không tồn hồ sơ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(HoSoBN)</p> <p>//Xin khoá đọc trên bảng HoSoBN với đk MaHS = @MaHS.</p>
		WAITFOR DELAY '0:0:5';	
BEGIN TRAN			

B1: Kiểm tra thông tin hồ sơ khách hàng IF EXISTS (SELECT * FROM HoSoBN WHERE MaHS = @MaHS) BEGIN PRINT N'Không tồn hồ sơ đó' ROLLBACK TRAN RETURN 0 END			
B2: Cập nhật thông tin vào bảng KHACHHANG UPDATE HoSoBN SET NgayKham = @NgayKham, MaKH = @MaKH Where MaHS = @MaHS;	U,X(HoSoBN) //Không xin được khóa trên bảng HoSoBN vì LayThongTinHoSo đang giữ khóa		
WAITFOR DELAY '0:0:5';			

		B2: Xem thông tin hồ sơ khách hàng SELECT * FROM HoSoKH Where MaKH = @MaKH;	S(HoSoBN) //Xin khoá đọc trên bảng HoSoBN với đk MaKH = @MaKH.
COMMIT		COMMIT	

II. Sinh viên thực hiện: Nguyễn Hoài Mẫn

1. Tình huống 1: Quản trị viên A đăng nhập vào hệ thống để cập nhật thông tin của một dịch vụ, chẳng hạn như điều chỉnh giá hoặc thay đổi mô tả dịch vụ. Trong khi đó, Quản trị viên B cũng đăng nhập và chọn cùng một dịch vụ để cập nhật thông tin.

FIX01: LOST UPDATE

T1 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 150

T2 (User = Admin): Admin 1 tiến hành cập nhật thông tin dịch vụ là : 'DV001', 'Kham rang tong quat', 'Dich vu nhanh', 200

Sp_UpdateDichVu_1	Khóa	Sp_UpdateDichVu_2	Khóa
--------------------------	-------------	--------------------------	-------------

<p><u>Input:</u></p> <p><u>@MaDV = 'DV001'</u></p> <p><u>@TenDV = 'Kham rang tong quat'</u></p> <p><u>@Loai = 'Dich vu nhanh'</u></p> <p><u>@Tien = '150'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>		<p><u>Input:</u></p> <p><u>@MaDV = 'DV001'</u></p> <p><u>@TenDV = 'Kham rang tong quat'</u></p> <p><u>@Loai = 'Dich vu nhanh'</u></p> <p><u>@Tien = '150'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>	
BEGIN TRAN			

<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <p>IF NOT EXISTS (SELECT *</p> <p>FROM DICHVU WITH (XLOCK)</p> <p>WHERE MADV = 'DV001')</p> <p>BEGIN</p> <p>PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(DICHVU)</p> <p>//Xin khoá đọc</p> <p>trên bảng</p> <p>DICHVU với</p> <p>điều kiện</p> <p>MADV = 'DV001'</p>		
<p>WAITFOR DELAY '0:0:5'</p>			
		<p>BEGIN TRAN</p>	
		<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <p>IF NOT EXISTS (SELECT *</p> <p>FROM DICHVU WITH (XLOCK)</p>	<p>S(DICHVU)</p> <p>//Xin khoá đọc</p> <p>trên bảng</p> <p>DICHVU với</p>

		WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END	điều kiện MADV = 'DV001' nhưng do Sp_UpdateDichVu_1 đang giữ khóa S nên không xin được
B2: Cập nhật thông tin thuộc tính TenDV, Loai, Tien trong bảng DICHVU thành 'Kham rang tong quat', 'Dịch vụ nhanh', 150 UPDATE DichVu SET TenDV = @TenDV, Loai = @Loai, Tien = @Tien WHERE MaDV = @MaDV	U,X(DICHVU) //Xin khoá update trên bảng DICHVU với điều kiện MADV = 'DV001'		
COMMIT			

		<p>B1: Kiểm tra thông tin mã dịch vụ(MADV) có tồn tại hay không ?</p> <pre> IF NOT EXISTS (SELECT * FROM DICHVU WITH (XLOCK) WHERE MADV = 'DV001') BEGIN PRINT 'Dịch vụ ' + @MaDV + N' Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(DICHVU)</p> <p>//Xin khoá đọc trên bảng DICHVU với điều kiện MADV = 'DV001' do Sp_UpdateDichVu_1 đã nhả khóa S nên xin được</p>
--	--	--	--

		<p>B2: Cập nhật thông tin thuộc tính</p> <p>TenDV, Loai, Tien trong bảng DICHVU thành 'Kham rang tong quat', 'Dich vu nhanh', 200</p> <p>UPDATE DichVu</p> <p>SET TenDV = @TenDV,</p> <p>Loai = @Loai,</p> <p>Tien = @Tien</p> <p>WHERE MaDV = @MaDV</p>	<p>U,X(DICHVU)</p> <p>//Xin khoá update trên bảng DICHVU với điều kiện</p> <p>MADV = 'DV001'</p>
		COMMIT	

2. Tình huống 2: Trong hệ thống, có một nha sĩ vừa đăng nhập xong thì admin tiến hành đổi mật khẩu của tài khoản bác sĩ đó. Dẫn đến việc sau khi đăng nhập, bác sĩ đó không thể xử lý lấy được thông tin tài khoản của mình.

<p>FIX02: UNREPEATABLE READ</p> <p>T1 (User = Nha sĩ): Nha sĩ tiến hành đăng nhập vào hệ thống với tên đăng nhập là : “111111112”, mật khẩu là : “12345”</p> <p>T2 (User = admin): admin tiến hành đổi mật khẩu của tài khoản nha sĩ có mã số 1 từ “12345” thành “54321”</p>

Sp_NV_LayTongTinTK	Khóa	Sp_NV_DoiMK	Khóa
<p><u>Input:</u></p> <p><u>@SDT= '11111112'</u></p> <p><u>@MATKHAU = '12345'</u></p> <p><u>@LoaiTK = '1'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>		<p><u>Input:</u></p> <p><u>@SDT = '11111112',</u></p> <p><u>@MATKHAU = '54321'</u></p> <p><u>Output:</u> <u>1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u></p>	
BEGIN TRAN			
<p>SET TRAN ISOLATION LEVEL</p> <p>REPEATABLE READ</p>	<p>Thiết lập khóa S</p> <p>khi đọc và giữ đến</p> <p>khi hết giao tác</p>		

B1: Xử lý lấy thông tin mã tài khoản nhà sĩ SELECT @Ma = NS.MaNS FROM TaiKhoan A INNER JOIN NhaSi NS ON A.SDT = NS.SDTNS WHERE A.SDT = <u>'111111112'</u> AND A.MatKhai = <u>'12345'</u> AND A.LoaiTK = <u>'1'</u>	S(TAIKHOAN) //Xin khoá đọc trên bảng TaiKhoan với điều kiện TENDANGNHAP = '111111112', MATKHAU = '12345', LoaiTK = '1'		
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	

		<p>B1: Kiểm tra xem tài khoản có tồn tại hay không?</p> <pre> IF NOT EXISTS (SELECT * FROM TAIKHOAN WHERE SĐT = '111111112') BEGIN PRINT 'Tài khoản Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>	<p>S(TAIKHOAN)</p> <p>//Không cấp khóa cho S do Sp_NV_LayTongTin TK đang giữ khóa X</p>
		<p>B2: Xử lý cập nhật mật khẩu</p> <pre> UPDATE TAIKHOAN SET MATKHAU = '54321' WHERE SĐT = '111111112' </pre>	<p>U,X(TAIKHOAN)</p> <p>//Không cấp khóa cho S do Sp_NV_LayTongTin TK đang giữ khóa X</p>

<p>B2: Kiểm tra xem mã tài khoản có tồn tại hay không?</p> <pre> IF @Ma IS NULL BEGIN PRINT N'Nhà sĩ Không Tồn Tại' ROLLBACK TRAN RETURN 0 END </pre>			
<p>B3: Xử lý lấy thông tin của tài khoản?</p> <pre> SELECT * FROM NhaSi NS WHERE NS.MaNS = @Ma; </pre>	<p>S(ACCOUNT), S(NHANVIEN)</p> <p>//Xin khoá đọc trên bảng NHASI, với điều kiện NS.MaNS = @Ma</p>		
COMMIT		COMMIT	

III. Sinh viên thực hiện: Nguyễn Phương Khanh

- Tình huống 1: Quản trị viên A cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ, trong lúc đó quản trị viên B cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ

ERR01: LOST UPDATE			
T1 (User = Quản trị viên A): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 20 vỉ			
T2 (User = Quản trị viên B): thực hiện cập nhật số lượng tồn kho của loại thuốc A từ 30 vỉ còn 15 vỉ			
Updated_SLTonKhoThuoc_1	Khóa	UpdatedSoLuongTonKhoThuoc_2	Khóa
<u>Input:</u> @SLTonKho		<u>Input:</u> @SLTonKho	
<u>Output:</u>		<u>Output:</u>	
BEGIN TRAN			
B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WITH (XLOCK) WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó'	S(THUOC) //Xin khoá đọc trên bảng THUOC với đk MaThuoc = @MaThuoc.		

ROLLBACK TRAN RETURN 0 END			
WAITFOR DELAY '0:0:5'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin thuốc có tồn tại hay không IF EXISTS (SELECT * FROM THUOC WITH (XLOCK) WHERE MaThuoc = @MaThuoc) BEGIN PRINT N 'Không tồn tại loại thuốc đó' ROLLBACK TRAN RETURN 0	S(THUOC) //Xin khoá đọc trên bảng THUOC với đk MaThuoc = @MaThuoc. nhưng do Updated_SLT onKhoThuoc_ 1 đang giữ khoá S nên không xin được

		END	
<p>B2: Cập nhật thông tin SLTonKho vào bảng THUOC</p> <p>UPDATE THUOC</p> <p>SET @SLTonKho = 20</p> <p>where MaThuoc = @MaThuoc;</p>	<p>U,X(THUOC)</p> <p>//Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc</p>		
		<p>B1: Kiểm tra thông tin thuốc có tồn tại hay không</p> <p>IF EXISTS (SELECT *</p> <p>FROM THUOC WITH (XLOCK)</p> <p>WHERE MaThuoc = @MaThuoc)</p> <p>BEGIN</p> <p>PRINT N 'Không tồn tại loại thuốc đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p>	<p>S(THUOC)</p> <p>//Xin khoá đọc trên bảng THUOC với dk MaThuoc = @MaThuoc do Updated_SLTonKhoThuoc_1 đã nhả khóa S nên xin được</p>

		END	
		<p>B2: Cập nhật thông tin SLTonKho vào bảng THUOC</p> <p>UPDATE THUOC</p> <p>SET @SLTonKho = 15</p> <p>where MaThuoc = @MaThuoc;</p>	<p>U,X(THUOC)</p> <p>//Xin khóa update trên bảng THUOC với dk MaThuoc = @MaThuoc</p>
COMMIT		COMMIT	

2. Tình huống 2: Nhân viên xem số điện thoại của nha sĩ A để liên lạc, cùng lúc đó nha sĩ A cập nhật số điện thoại cá nhân của mình từ 123456789 thành 987654321 và lưu thay đổi thành công. Sau đó nhân viên load lại thông tin, đọc được số điện thoại mới.

ERR01: UNREPEATEABLE READ

T1 (User = Nhân viên): thực hiện xem số điện thoại của nha sĩ A để liên lạc.

T2 (User = Nha sĩ): thực hiện cập nhật số điện thoại cá nhân từ 123456789 thành 987654321 và lưu thay đổi.

LaySoDienThoaiNS	Khóa	UpdatedSoDienThoaiNS	Khóa
<p><u>Input:</u> @SDTNS</p> <p><u>Output:</u> @SDTNS='123456789'</p>		<p><u>Input:</u> @SDTNS='987654321'</p> <p><u>Output:</u> @SDTNS='123456789'</p>	

BEGIN TRAN			
SET TRAN ISOLATION LEVEL REPEATABLE READ	Thiết lập khóa S khi đọc và giữ đến hết giao tác		
B1: Kiểm tra thông tin nhà sĩ có tồn tại hay không IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS) BEGIN PRINT N 'Không tồn tại nhà sĩ đó' ROLLBACK TRAN RETURN 0 END	S(NhaSi) //Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.		
B2: Xem thông tin số điện thoại nhà sĩ	S(NhaSi) //Xin khoá đọc		

SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	trên bảng NhaSi với đk MaNS = @MaNS.		
WAITFOR DELAY '0:0:05';			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin nha sĩ có tồn tại hay không</p> <p>IF EXISTS (SELECT * FROM NhaSi WHERE MaNS = @MaNS)</p> <p>BEGIN</p> <p>PRINT N 'Không tồn tại nha sĩ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(NhaSi)</p> <p>//Không cấp khóa S do LaySoDienTh oiNS đang giữ khóa X</p>

		<p>B2: Cập nhật số điện thoại nhà sĩ</p> <p>UPDATE NhaSi</p> <p>SET @SDTNS = 123456789</p> <p>where MaNS = @MaNS;</p>	<p>U,X(NhaSi)</p> <p>//Không cấp khóa S do LaySoDienThoaiNS đang giữ khóa X</p>
<p>B2: Kiểm tra thông tin nhà sĩ có tồn tại hay không</p> <p>IF EXISTS (SELECT *</p> <p>FROM NhaSi</p> <p>WHERE MaNS = @MaNS)</p> <p>BEGIN</p> <p>PRINT N 'Không tồn tại nhà sĩ đó'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>			
B3: Xem thông tin số điện thoại nhà sĩ _ lần 2	S(NhaSi)		

SELECT SDTNS FROM NhaSi WHERE MaNS = @MaNS;	//Xin khoá đọc trên bảng NhaSi với đk MaNS = @MaNS.		
COMMIT		COMMIT	

IV. Sinh viên thực hiện: Đinh Thành Danh

1. Tình huống 1: Khách hàng A đăng nhập vào hệ thống chọn việc đặt lịch hẹn, trong khi đó nha sĩ B cũng đăng nhập vào hệ thống hiển thị lịch hẹn cũ

ERR01: PHANTOM READ

T1 (User = Khách hàng):Khách hàng 1 thực hiện việc đặt lịch hẹn

T2 (User = Nha sĩ): Thực hiện xem danh sách cuộc hẹn

Sp_ThemLichHen_1	Khóa	Sp_XemLichHen_2	Khóa
<u>Input:</u> <u>@NgayHen</u> <u>@ThoiGianHen</u> <u>@HoTenKH</u> <u>@MaKH</u>			

@MaNS			
<u>Output: 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</u>			
SET TRANSACTION ISOLATION LEVEL		SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	
		BEGIN TRAN	
		B1: Xem danh sách lịch hẹn: SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS return 1	S(LichCaNhanNS) //Xin khoá đọc trên bảng LichCaNhanNS
		WAITFOR DELAY '0:0:10'	
BEGIN TRAN			
B1: Kiểm tra ngày và thời gian lịch hẹn xem có chưa ? IF(EXISTS(SELECT * FROM LichHen WHERE Ngay = '01-10-2023' AND ThoiGian = '08:30:00'))	S(LichHen) //Xin khoá đọc trên bảng LichHen với		

RETURN -1	điều kiện Ngày = '01-10-2023' và ThoiGian ='08:30:00'		
B2: Kiểm tra mã nha sĩ xem có chưa ? IF(EXISTS(SELECT * FROM LichHen WHERE MaNS = '1') RETURN -1	S(LichHen) //Xin khoá đọc trên bảng LichHen với điều kiện MaNS = '1'		
B3 : Thêm lịch hẹn mới INSERT INTO LichHen(NgayHen, ThoiGianHen, TenKH, MaKH, MaNS) VALUES ('01-10-2023', '08:30:00', N'Nguyễn Văn E', '3', '2') return 1	X(LichHen) //xin khoá ghi trên bảng LichHen để thêm lịch hẹn mới		
B4: Cập nhật cuộc hẹn: UPDATE LichHen SET NgayHen = @NgayHen AND ThoiGianHen = @ThoiGianHen WHERE MAKH = @MaKH	UX(KHACHHANG) //xin khóa ghi trên bảng DOITAC		

		<p>B1: Xem danh sách lịch hẹn:</p> <p>SELECT Ngay,ThoiGian,MaNS FROM LichCaNhanNS</p> <p>return 1</p>	<p>S(LichCaNhanNS) //Xin khoá đọc trên bảng LichCaNhanNS</p>
		COMMIT	

2. Tình huống 2: Trong hệ thống, khách hàng đang cập nhật thông tin cá nhân thì nhân viên xem thông tin của khách hàng đó.

ERR02: DIRTY READ

T1 (User = Khách hàng): Khách hàng đang thực hiện cập nhật thông tin

T2 (User = Nhân viên): Nhân viên đang xem thông tin cũ của khách hàng

Sp_XemThongTinCaNhan	Khóa	Sp_XemThongTinKhachHang	Khóa
<p><u>Input:</u> <u>@HoTenKH</u></p> <p><u>@NgaySinhKH</u></p> <p><u>@DiaChiKH</u></p> <p><u>@SDTKH</u></p>			

<p><u>@EmailKH</u></p> <p><u>Output:</u> 1 biến int có giá trị 1 hoặc 0. 1 –Thành công ngược lại 0 – Thất bại</p>			
		SET TRAN ISOLATION LEVEL READ COMMITTED	
BEGIN TRAN			
<p>B1: Kiểm tra thông tin (1) khách hàng</p> <p>IF EXISTS (SELECT * FROM KHACHHANG WHERE SDTKH = @SDTKH)</p> <p>BEGIN</p> <p>PRINT N'Không tồn tài khoản'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHAN G với đk SDTKH = @SDTKH</p>		

<p>B2: Cập nhật thông tin vào bảng KHACHHANG</p> <p>UPDATE KHACHHANG SET EmailKH = @EmailKH Where SDTKH = @SDTKH;</p>	<p>U,X(KHACHHANG)</p> <p>//Xin khóa update trên bảng KHACHHANG với dk SDTKH = @SDTKH</p>		
<p>WAITFOR DELAY '0:0:05';</p>			
		<p>BEGIN TRAN</p>	
		<p>B1: Kiểm tra thông tin khách hàng có tồn tại không</p> <p>IF EXISTS (SELECT * FROM KHACHHANG WHERE MaKH = @MaKH)</p> <p>BEGIN</p> <p>PRINT N'Không tồn khách hàng đó'</p> <p>ROLLBACK TRAN</p>	<p>S(KHACHHANG)</p> <p>//Xin khoá đọc trên bảng KHACHHANG với dk MaKH = @MaKH.</p>

		RETURN 0 END	
		B2: Xem thông tin khách hàng SELECT * FROM KHACHHANG Where MaKH = @MaKH;	S(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với đk MaKH = @MaKH.
		COMMIT	