



MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU



Sinh viên thực hiện:

19127040 – Trần Ngọc Lam

19127088 – Nguyễn Phương Vy

19127219 – Nguyễn Kim Thị Tố Nga

GV phụ trách:

Hồ Thị Hoàng Vy

Tiết Gia Hồng

ĐỒ ÁN/BÀI TẬP MÔN HỌC - HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
HỌC KỲ I – NĂM HỌC 2021-2022



NỘI DUNG

A. Thông tin chung	3
I. Thông tin chi tiết nhóm.....	3
II. Bảng phân công công việc.....	3
B. Nội dung báo cáo.....	4
I. Mô tả hệ thống	4
1. Đặc tả hệ thống đặt hàng và chuyển hàng online	4
2. Mô tả bảng và thuộc tính	5
3. Thiết kế cơ sở dữ liệu của hệ thống	8
4. Phân quyền	9
5. Các hàm chức năng của hệ thống	9
II. Tình huống tranh chấp	13
1. Xác định tình huống tranh chấp	13
1.1. Test Case 1:.....	13
1.2. Test Case 2:.....	17
1.3. Test Case 3:.....	20
1.4. Test Case 4:.....	23
1.5. Test Case 5:.....	25
1.6. Test Case 6:.....	28
1.7. Test Case 7:.....	30
1.8. Test Case 8:.....	32
1.9. Test Case 9:.....	35
2. Giải quyết tình huống tranh chấp.....	36
2.1. Test Case 1:.....	36
2.2. Test Case 2:.....	40
2.3. Test Case 3:.....	44
2.4. Test Case 4:.....	47
2.5. Test Case 5:.....	49
2.6. Test Case 6:.....	51
2.7. Test Case 7:.....	53
2.8. Test Case 8:.....	56



2.9. Test Case 9:	58
III. Thiết kế giao diện chức năng.....	60
1. Khách hàng	60
2. Đối tác	66
3. Tài xế	71
IV. Link Google Drive	73



A. Thông tin chung

I. Thông tin chi tiết nhóm

Mã nhóm:	Nhóm 17		
Số lượng:	3 thành viên		
MSSV	Họ tên	Email	Điện thoại
19127040	Trần Ngọc Lam	19127040@student.hcmus.edu.vn	0794202507
19127219	Nguyễn Kim Thị Tô Nga	19127219@student.hcmus.edu.vn	0933821001
19127088	Nguyễn Phương Vy	19127088@student.hcmus.edu.vn	0931457030

II. Bảng phân công công việc

Người thực hiện	Công việc	Mức độ hoàn thành của công việc được giao	Mức độ hoàn thành trên đồ án
Trần Ngọc Lam	<ul style="list-style-type: none"> - Vẽ sơ đồ ER thiết kế cơ sở dữ liệu. - Thực hiện phân quyền cho phân hệ đối tác. - Phân tích tranh chấp và tạo các store procedure liên quan cho TestCase 5,6 - Giải quyết tranh chấp cho TestCase 5,6 - Code giao diện và cài đặt hệ thống cho đăng ký , đăng nhập, trang chủ, trang sản phẩm, tài xế đặt hàng,cập nhật tình trạng - Làm báo cáo 	100%	100%
Nguyễn Phương Vy	<ul style="list-style-type: none"> - Cài đặt cơ sở dữ liệu. - Thực hiện phân quyền cho phân hệ khách hàng và phân hệ tài xế. - Phân tích tranh chấp và tạo các store procedure liên quan cho TestCase 3,4,8 - Giải quyết tranh chấp cho TestCase 3,4,8 - Code giao diện và cài đặt hệ thống cho phần đặt hàng, kiểm tra sản phẩm, thêm sản phẩm, cập nhật sản phẩm - Làm báo cáo 	100%	100%
Nguyễn Kim Thị Tô Nga	<ul style="list-style-type: none"> - Cài đặt cơ sở dữ liệu. - Thực hiện phân quyền cho phân hệ nhân viên và phân hệ quản lý. - Phân tích tranh chấp và tạo các store procedure liên quan cho TestCase 1,2,7,9 - Giải quyết tranh chấp cho TestCase 1,2,7,9 - Code giao diện và cài đặt hệ thống cho danh sách đơn hàng, chi tiết đơn hàng - Làm báo cáo 	100%	100%



B. Nội dung báo cáo

I. Mô tả hệ thống

1. Đặc tả hệ thống đặt hàng và chuyển hàng online

Đối tác hay còn gọi là người bán hàng sẽ có một mã đối tác để quản lý, tên đối tác, mã số thuế, tên người đại diện, địa chỉ, số chi nhánh, số lượng đơn hàng, loại hàng vận chuyển, địa chỉ kinh doanh, số điện thoại, email.

Mỗi đối tác sẽ có **hợp đồng**, hợp đồng phân biệt bằng mã hợp đồng và các thông tin như người đại diện, số chi nhánh đăng ký, thời gian hiệu lực và phần trăm hoa hồng. Mỗi đối tác có thể có nhiều hợp đồng khác nhau.

Mỗi **chi nhánh** của từng đối tác sẽ có một mã chi nhánh để quản lý, tên quản lý, địa chỉ và số điện thoại. Mỗi đối tác sẽ có nhiều chi nhánh khác nhau.

Các **sản phẩm** sẽ được quản lý bởi mã sản phẩm, tên sản phẩm, số lượng tồn, giá bán. Các sản phẩm được cung cấp bởi các chi nhánh của đối tác.

Mỗi **khách hàng** sẽ có một mã khách hàng khác nhau, họ tên, số điện thoại, địa chỉ và email.

Mỗi **đơn hàng** sẽ có mã đơn hàng khác nhau, địa chỉ giao hàng, tên người nhận, số điện thoại người nhận, ngày lập, phí vận chuyển, tổng tiền hàng, tổng tiền, hình thức thanh toán.

Mỗi đơn hàng sẽ có **chi tiết hóa đơn** bao gồm mã đơn hàng, sản phẩm, số lượng, giá bán và thành tiền của từng loại sản phẩm.

Mỗi đơn hàng sẽ có nhiều **tình trạng đơn hàng** được lưu bởi ngày cập nhật và **chi tiết tình trạng đơn hàng** sẽ bao gồm mã tình trạng và mô tả tình trạng tương ứng như sau:

- 0: Đơn hàng đã hủy.
- 1: Đơn hàng được tiếp nhận.
- 2: Đơn hàng đang đóng gói.
- 3: Đang chờ giao hàng
- 4: Đơn hàng đang giao.
- 5: Đơn hàng giao thành công.

Thông tin của **tài xế** bao gồm số CMND, họ tên, số điện thoại, địa chỉ, biển số xe, khu vực hoạt động, email, số tài khoản, ngân hàng và chi nhánh đơn hàng.

Thu nhập của tài xế sẽ là phí vận chuyển của đơn hàng mà tài xế tiếp nhận.

a. Đặt hàng



Khách hàng chọn đối tác từ danh sách đối tác, chi nhánh của hệ thống. Khi khách hàng chọn đối tác, hệ thống sẽ hiển thị danh sách sản phẩm của đối tác. Khách hàng chọn sản phẩm, số lượng tương ứng và cung cấp các thông tin cần thiết cho quá trình giao hàng. Hệ thống sẽ tiến hành tính toán tổng tiền cho đơn hàng và phí vận chuyển tương ứng (phí vận chuyển được tính bằng 5% của tổng tiền hàng). Sau khi khách hàng xác nhận thì đơn hàng sẽ được chuyển giao cho đối tác và tài xế.

b. Tài xế tiếp nhận đơn hàng

Sau khi đơn hàng đã được đóng gói bởi đối tác, hệ thống sẽ hiển thị danh sách các đơn hàng. Tài xế xác nhận tiếp nhận đơn hàng sẽ phục vụ và cập nhật tình trạng đơn hàng tương ứng.

c. Thu nhập tài xế

Với mỗi đơn hàng giao hàng thành công, tài xế sẽ được nhận khoản thu nhập tương ứng với phí vận chuyển của đơn hàng đó.

d. Các trigger được cài đặt bao gồm:

Thành tiền của từng sản phẩm (chi tiết đơn hàng) = số lượng * giá bán.

Gía bán trong chi tiết đơn hàng phải bằng giá bán của sản phẩm.

Tổng tiền của đơn hàng = tổng thành tiền (chi tiết đơn hàng) + phí vận chuyển.

Số lượng chi nhánh (đối tác) sẽ được cập nhật mỗi khi đối tác thêm mới một chi nhánh.

Số lượng tồn (sản phẩm) sẽ được cập nhật lại mỗi khi có đơn hàng được đặt, khi sửa số lượng (chi tiết hóa đơn) và khi đơn hàng bị hủy.

2. Mô tả bảng và thuộc tính

Tên thuộc tính	Kiểu dữ liệu	Mô tả thuộc tính
<i>DoiTac</i>		
MaDT (PK)	varchar(10)	Mã đối tác cho từng đối tác riêng biệt
pword	varchar(20)	Password để truy cập vào tài khoản của đối tác
MSThue	varchar(10)	Mã số thuế của từng đối tác
TenDT	nvarchar(100)	Tên gọi của từng đối tác
TenNgDaiDien	nvarchar(100)	Tên người đại diện của đối tác tương ứng
ThanhPho	nvarchar(50)	Thành phố của đối tác
Quan	nvarchar(30)	Quận của đối tác
SoChiNhanh	int	Tổng số chi nhánh trên cả nước của đối tác
LoaiHang	nvarchar(100)	Loại hàng mà đối tác cung cấp
DiaChi	nvarchar(200)	Địa chỉ cụ thể của đối tác
Email	varchar(30)	Email liên lạc của đối tác

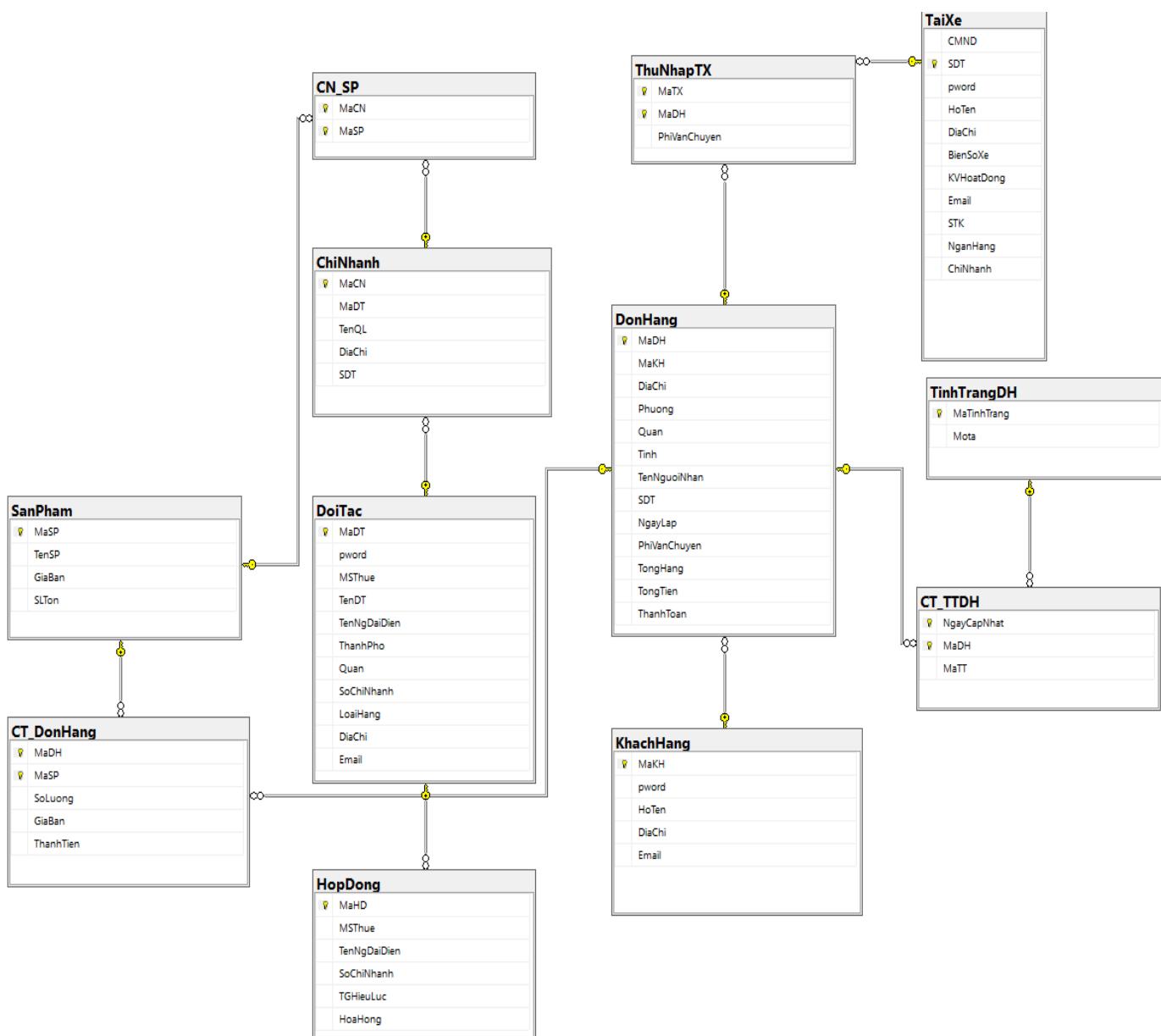


Tên thuộc tính	Kiểu dữ liệu	Mô tả thuộc tính
HopDong		
MaHD (PK)	Varchar(10)	Mã hợp đồng cho từng hợp đồng riêng biệt
MSThue	Varchar(10)	Mã số thuế của đối tác (Khóa ngoại vào DoiTac)
TenNgDaiDien	Nvarchar(100)	Tên người đại diện hợp đồng
SoChiNhanh	Int	Số chi nhánh của cửa hàng
TGHieuLuc	Date	Thời gian hiệu lực của hợp đồng
HoaHong	Int	Phần trăm hoa hồng
ChiNhanh		
MaCN (PK)	varchar(10)	Mã chi nhánh cho từng chi nhánh riêng biệt
MaDT	varchar(10)	Mã đối tác thể hiện chi nhánh thuộc đối tác nào
TenQL	nvarchar(100)	Tên của người quản lý chi nhánh đó
DiaChi	nvarchar(100)	Địa chỉ cụ thể của chi nhánh
SDT	varchar(10)	Số điện thoại liên hệ của chi nhánh
SanPham		
MaSP (PK)	varchar(6)	Mã sản phẩm cho từng sản phẩm riêng biệt
TenSP	nvarchar(50)	Tên gọi cho từng sản phẩm
GiaBan	float	Xem giá bán của từng sản phẩm
SLTon	int	Số lượng tồn của sản phẩm ở trong cửa hàng
CN_SP		
MaCN (PK)	varchar(10)	Mã chi nhánh mà sản phẩm thuộc về
MaSP (PK)	varchar(6)	Mã sản phẩm được cung cấp bởi chi nhánh tương ứng
KhachHang		
MaKH	varchar(10)	Mã khách hàng cho từng khách hàng riêng biệt
pword	varchar(20)	Password để truy cập vào tài khoản của khách hàng
HoTen	nvarchar(100)	Họ tên của khách hàng
DiaChi	nvarchar(100)	Địa chỉ liên lạc của khách hàng
Email	varchar(30)	Email liên lạc của khách hàng
DonHang		
MaDH (PK)	varchar(10)	Mã đơn hàng cho từng đơn hàng riêng biệt
MaKH	varchar(10)	Mã khách hàng đã mua hàng tương ứng với đơn hàng
DiaChi	nvarchar(30)	Số nhà và tên đường giao hàng
Phuong	nvarchar(30)	Địa chỉ phường giao hàng
Quan	nvarchar(30)	Địa chỉ quận giao hàng
Tinh	nvarchar(30)	Địa chỉ thành phố giao hàng
TenNguoiNhan	nvarchar(100)	Tên người nhận hàng
SDT	varchar(10)	Số điện thoại người nhận hàng
NgayLap	datetime	Ngày lập đơn hàng
PhiVanChuyen	float	Phí vận chuyển của đơn hàng
TongHang	float	Tổng tiền của của hàng hóa trong đơn hàng
TongTien	float	Tổng tiền của đơn hàng bằng tổng của phí vận chuyển và tổng tiền hàng hóa
ThanhToan	nvarchar(100)	Hình thức thanh toán của khách hàng khi mua hàng



Tên thuộc tính	Kiểu dữ liệu	Mô tả thuộc tính
CT_DonHang		
MaDH (PK)	varchar(10)	Mã đơn hàng cho từng đơn hàng riêng biệt
MaSP (PK)	varchar(6)	Mã sản phẩm được mua trong đơn hàng tương ứng
SoLuong	int	Số lượng được mua của sản phẩm ứng với mã sản phẩm
GiaBan	float	Giá bán của sản phẩm ứng với những sản phẩm
ThanhTien	int	Thành tiền của sản phẩm bằng số lượng x giá bán
TinhTrangDH		
MaTinhTrang (PK)	int	Mã tình trạng của từng tình trạng riêng biệt
Mota	nvarchar(100)	Mô tả cho mã tình trạng: - 0: Đơn hàng đã hủy. - 1: Đơn hàng được tiếp nhận. - 2: Đơn hàng đang đóng gói. - 3: Đang chờ giao hàng - 4: Đơn hàng đang giao. - 5: Đơn hàng giao thành công.
CT_TTDH		
NgayCapNhat (PK)	datetime	Ngày giờ cập nhật tình trạng đơn hàng
MaDH (PK)	varchar(10)	Mã đơn hàng được cập nhật
MaTT	int	Mã tình trạng tương ứng với tình trạng đơn hàng
TaiXe		
CMND	varchar(12)	Chứng minh nhân dân của từng tài xế riêng biệt
SDT (PK)	varchar(10)	Số điện thoại liên lạc của tài xế
pword	varchar(20)	Password để đăng nhập vào tài khoản của tài xế
HoTen	nvarchar(100)	Họ tên của tài xế
DiaChi	nvarchar(100)	Địa chỉ liên lạc của tài xế
BienSoXe	nvarchar(12)	Biển số xe của tài xế để cho khách hàng có thể nhận biết
KVHoatDong	nvarchar(30)	Khu vực hoạt động giao hàng của tài xế
Email	varchar(30)	Email liên lạc của tài xế
STK	varchar(15)	Số tài khoản ngân hàng của tài xế
NganHang	nvarchar(30)	Ngân hàng mà tài xế lập tài khoản
ChiNhanh	nvarchar(30)	Chi nhánh mà tài xế hoạt động
ThuNhapTX		
MaTX (PK)	varchar(10)	Mã tài xế của từng tài xế riêng biệt (SDT)
MaDH (PK)	varchar(10)	Mã đơn hàng mà tài xế đã nhận
PhiVanChuyen	int	Phí vận chuyển của đơn hàng tài xế đã nhận

3. Thiết kế cơ sở dữ liệu của hệ thống





4. Phân quyền

Nhóm người dùng	Các quyền trên cơ sở dữ liệu
Phân hệ đối tác	<ul style="list-style-type: none"> - Xem, sửa các thông tin cá nhân. - Xem, thêm, xóa, sửa thông tin sản phẩm. - Xem, thêm, xóa, sửa thông tin chi nhánh. - Xem thông tin và chi tiết đơn hàng. - Cập nhật tình trạng đơn hàng.
Phân hệ khách hàng	<ul style="list-style-type: none"> - Đăng ký, xem, sửa các thông tin cá nhân. - Xem danh sách các đối tác, chi nhánh trực thuộc và danh sách sản phẩm được cung cấp bởi từng chi nhánh của đối tác. - Thêm mới một đơn hàng. - Xem thông tin, chi tiết đơn hàng và tình trạng đơn hàng.
Phân hệ tài xế	<ul style="list-style-type: none"> - Đăng ký, xem, sửa các thông tin cá nhân. - Xem danh sách đơn hàng. - Cập nhật tình trạng của đơn hàng - Xem danh sách thu nhập cá nhân
Phân hệ nhân viên	<ul style="list-style-type: none"> - Xem danh sách đối tác. - Cập nhật hợp đồng.
Phân hệ quản trị	<ul style="list-style-type: none"> - Cập nhật thông tin tài khoản. - Xem, thêm, xóa, sửa tài khoản của quản trị viên, nhân viên. - Khóa và kích hoạt tài khoản. - Cấp quyền thao tác trên cơ sở dữ liệu và giao diện.

5. Các hàm chức năng của hệ thống

STT	STORE PROCEDURE	CHỨC NĂNG	THAM SỐ	CÁC BƯỚC THỰC HIỆN
1	sp_CreateAccount_DT	Tạo account cho đối tác khi đăng ký và lưu thông tin đối tác vào Table DoiTac.	<code>@MSThue varchar(10), @TenDT nvarchar(100), @TenNgDaiDien nvarchar(100), @ThanhPho nvarchar(50), @Quan nvarchar(30), @SoChiNhanh int, @LoaiHang nvarchar(100), @DiaChi nvarchar(200), @email varchar(30), @SDT varchar(10)</code>	Insert thông tin của đối tác vào Table đối tác với MaDT vừa được tạo.



2	sp_CreateAccount_TX	Tạo account cho tài xế khi đăng ký và lưu thông tin tài xế vào table TaiXe.	<pre>@CMND varchar(12), @SDT varchar(10), @pword varchar(20), @HoTen nvarchar(100), @DiaChi nvarchar(100), @BienSoXe nvarchar(12), @KVHoatDong nvarchar(30), @Email varchar(30), @STK varchar(15), @NganHang nvarchar(30), @ChiNhanh nvarchar(30)</pre>	Insert thông tin của tài xế vào Table TaiXe.
3	sp_CreateAccount_KH	Tạo account cho khách hàng khi đăng ký và lưu thông tin KhachHang vào bảng KhachHang.	<pre>@MaKH varchar(10), @pword varchar(20), @HoTen nvarchar(100), @DiaChi nvarchar(100), @Email varchar(30)</pre>	Insert thông tin của Khách hàng vào Table KhachHang.
4	sp_ThemDonHang	Tạo mã đơn hàng và lưu thông tin cho đơn hàng mới vào Table DonHang.	<pre>@MaKH varchar(10), @DiaChi nvarchar(30), @Phuong nvarchar(30), @Quan nvarchar(30), @Tinh nvarchar(30), @TenNguoiNhan nvarchar(100), @SDT varchar(10), @ThanhToan nvarchar(100), @MaDonHang varchar (10) output</pre>	<ol style="list-style-type: none">Lấy ra MaDH cuối cùng lớn nhất trong bảng DonHang.Kiểm tra nếu bảng chưa có thông tin nào thì MaDH mới bằng 1, ngược lại thì bằng MaDH cũ cộng thêm 1.Insert thông tin của đơn hàng vào Table DonHang với MaDH vừa được tạo.
5	sp_ThemSanPham	Tạo mã sản phẩm và lưu thông tin cho sản phẩm mới vào Table SanPham.	<pre>@TenSP nvarchar(50), @GiaBan float, @SLTon int</pre>	<ol style="list-style-type: none">Lấy ra MaSP cuối cùng lớn nhất trong bảng SanPham.Kiểm tra nếu bảng chưa có thông tin nào thì MaSP mới bằng 1, ngược lại thì bằng MaSP cũ cộng thêm 1.

				<p>3. Insert thông tin của đơn hàng vào Table SanPham với MaSP vừa được tạo.</p>
6	sp_ThemChiTietDonHang	Thêm chi tiết đơn hàng vào đơn hàng.	<pre>@MaDH varchar(10), @MaSP varchar(6), @SoLuong int, @error int output</pre>	<p>1. Kiểm tra đơn hàng có tồn tại hay không.</p> <p>2. Đọc số lượng tồn của sản phẩm được thêm.</p> <p>3. Xét xem nếu số lượng tồn nhỏ hơn số lượng sản phẩm cần thêm vô thì raiserror.</p> <p>4. Ngược lại thì xét xem nếu đã tồn tại sản phẩm trong đơn hàng rồi thì cộng thêm số lượng. Còn nếu đơn hàng chưa có sản phẩm đó thì thêm sản phẩm vào đơn hàng (thêm vào Table CT_DonHang).</p>
7	sp_TaiXeNhanDonHang	Tài xế nhận giao đơn hàng, lưu thông tin vào thu nhập tài xế, cập nhật tình trạng đơn hàng	<pre>@MaTX VARCHAR(12), @MaDH VARCHAR(10), @msg nvarchar(100) output</pre>	<p>1. Kiểm tra đơn hàng tương ứng có ngày cập nhật mới nhất có đang được chờ giao hay không (MaTT = 3) trong bảng TinhTrangDH.</p> <p>2. Đọc phí vận chuyển từ bảng đơn hàng của đơn hàng tương ứng.</p> <p>3. Insert thông tin của đơn hàng vừa nhận vào Table ThuNhapTX.</p> <p>4. Insert tình trạng đơn hàng = 4 ("Đang giao hàng") ở thời điểm hiện tại vào bảng TinhTrangDH.</p>
8	sp_CapNhatSanPham	Cập nhật thông tin của sản phẩm	<pre>@MaSP varchar(6), @TenSP nvarchar(50), @GiaBan float, @SLTon int</pre>	<p>1. Kiểm tra sản phẩm có tồn tại hay không.</p> <p>2. Cập nhật thông tin sản phẩm.</p>
9	sp_CapNhatTinhTrangDoHang	Cập nhật tình trạng đơn hàng	<pre>@MaDH varchar(10), @MaTT int</pre>	<p>1. Kiểm tra xem đơn hàng có tồn tại hay không.</p> <p>2. Nếu đơn hàng có tồn tại thì thêm một dòng vào bảng TinhTrangDH với</p>



				với ngày cập nhật là thời điểm thực hiện thao tác.
10	sp_XoaDonHang	Xóa đơn hàng	@MaDH varchar(10)	<ol style="list-style-type: none">Kiểm tra đơn hàng có tồn tại hay không.Nếu đơn hàng có tồn tại thì thực hiện xóa chi tiết có mã đơn hàng tương ứng. Sau đó xóa đơn hàng.
11	sp_KiemTraSP	Kiểm tra sản phẩm dựa theo MaSP và TenSP	@MaSP varchar(6), @TenSP nvarchar(50)	<ol style="list-style-type: none">Kiểm tra sản phẩm có tồn tại hay không dựa trên MaSP và TenSP.Nếu sản phẩm tồn tại thì in ra sản phẩm đó.
12	sp_KiemTraSLTon	In ra các sản phẩm có số lượng tồn <@slt	@slt int, @Tong int output	<ol style="list-style-type: none">Gán tổng các sản phẩm thỏa điều kiện vào biến @TongIn ra các sản phẩm thỏa điều kiện
13	sp_KiemTraGiaBan	In ra các sản phẩm có giá bán <@gb	@gb float, @Tong int output	<ol style="list-style-type: none">Gán tổng các sản phẩm thỏa điều kiện vào biến @TongIn ra các sản phẩm thỏa điều kiện
14	sp_getBillForDeliver	Lấy ra danh sách các đơn hàng đang trong tình trạng chờ vận chuyển	@start int, @num int, @tong int output	<ol style="list-style-type: none">Chọn ra các mã đơn hàng có MaTT = 3.@start và @num là hai giá trị phục vụ cho việc hiển thị ra giao diện.@tong là tổng số đơn hàng đang trong tình trạng chờ vận chuyển.



II. Tình huống tranh chấp

1. Xác định tình huống tranh chấp

1.1. Test Case 1:

Tình huống: Ở tình huống này trigger sẽ update mỗi khi một chi tiết đơn hàng được insert. Hai khách hàng đặt hàng cùng 1 lúc, nhưng số lượng tồn của 2 người đọc được lại ở thời điểm ban đầu khiến cho lưu trữ thông tin số lượng tồn bị sai lệch.

T1: Khách hàng 1 thêm sản phẩm vào chi tiết đơn hàng (Mã HD: 0000000001)

T2: Khách hàng 2 thêm sản phẩm vào chi tiết đơn hàng (Mã HD: 0000000002)

Xem số lượng tồn của sản phẩm '000001' = 50

T1	T2
<pre>exec sp_ThemChiTietDonHang_TC '0000000001', '000001', 20</pre>	<pre>exec sp_ThemChiTietDonHang_TC '0000000002', '000001', 15</pre>
<pre>EXEC('DISABLE TRIGGER sLT_dathang ON CT_DonHang'); BEGIN TRAN</pre>	
	<pre>EXEC('DISABLE TRIGGER sLT_dathang ON CT_DonHang'); BEGIN TRAN</pre>
<p>B1: Kiểm tra đơn hàng có tồn tại hay không</p> <pre>if not exists (select * from DonHang where @MaDH= MaDH) begin print('1') raiserror(N'Không tồn tại đơn hàng',15,1) end</pre>	
	<p>B1: Kiểm tra đơn hàng có tồn tại hay không</p> <pre>if not exists (select * from DonHang where @MaDH= MaDH) begin print('1') raiserror(N'Không tồn tại đơn hàng',15,1) end</pre>
<p>B2: Xem số lượng tồn ban đầu (@sl) và in ra @sl</p> <pre>declare @sl int set @sl = (select SLTon from SanPham where MaSP = @MaSP) print(@sl) ⇒ SLT = 50.</pre>	

	<p><i>B2: Xem số lượng tồn ban đầu và in ra</i></p> <pre>declare @sl int set @sl = (select SLTon from SanPham where MaSP = @MaSP) print(@sl) ⇒ SLT = 50.</pre>
<p><i>B3: Kiểm tra có vượt quá số lượng tồn trong kho hay không, nếu không vượt quá → kiểm tra chi tiết đơn hàng có tồn tại thì cộng thêm số lượng, ngược lại tạo một chi tiết đơn hàng mới</i></p> <pre>if (@sl-@SoLuong >= 0) begin if EXISTS (select * from CT_DonHang where MaDH = @MaDH and MaSP = @MaSP) begin waitfor delay '00:00:02' update CT_DonHang set SoLuong = (SoLuong + @SoLuong) where MaDH = @MaDH and MaSP = @MaSP end else begin waitfor delay '00:00:02' INSERT INTO CT_DonHang (MaDH, MaSP, SoLuong) VALUES (@MaDH, @MaSP, @SoLuong) UPDATE SanPham SET SLTon = @sl - @SoLuong WHERE MaSP = @MaSP end end ⇒ Ở đây thỏa điều kiện if vì @sl - @SoLuong = 50- 20 >= 0.</pre>	
	<p><i>B3: Kiểm tra có vượt quá số lượng tồn trong kho hay không, nếu không vượt quá → kiểm tra chi tiết đơn hàng có tồn tại thì cộng thêm số lượng, ngược lại tạo một chi tiết đơn hàng mới</i></p> <pre>if (@sl-@SoLuong >= 0) begin if EXISTS (select * from CT_DonHang where MaDH = @MaDH and MaSP = @MaSP) begin waitfor delay '00:00:02' update CT_DonHang</pre>

	<pre> set SoLuong = (SoLuong + @SoLuong) where MaDH = @MaDH and MaSP = @MaSP else begin waitfor delay '00:00:02' INSERT INTO CT_DonHang (MaDH, MaSP, SoLuong) VALUES (@MaDH, @MaSP, @SoLuong) UPDATE SanPham SET SLTon = @sl - @SoLuong WHERE MaSP = @MaSP end end ⇒ <i>Ở đây thỏa điều kiện if vì @sl - @SoLuong = 50-15 >= 0</i> </pre>
<p>⇒ <i>insert thành công CTĐH '0000000001', '000001', 20 → update số lượng tồn bằng 50-20=30.</i></p> <p>COMMIT TRAN</p> <p>EXEC('ENABLE TRIGGER slt_dathang ON CT_DonHang');</p>	
	<p>⇒ <i>insert thành công CTĐH '0000000002', '000001', 15 → update số lượng tồn bằng 50-15=35 → MẤT DỮ LIỆU SỐ LUỢNG TỒN CỦA T1.</i></p> <p>COMMIT TRAN</p> <p>EXEC('ENABLE TRIGGER slt_dathang ON CT_DonHang');</p>

⇒ Số lượng tồn của sau khi T1 (=30) đã update bị ghi đè bởi T2 (=35) update sau.



Hình ảnh demo:

1 Demo trên SQL SERVER

The screenshot shows two side-by-side SQL Server Management Studio windows. The left window, titled 'Tran_1_TestCase01....E (NGA\Admin (60))', contains the following script:

```

1 --số lượng tồn sản phẩm '000001' đang là 50
2 USE HT_DHCH_ONLINE
3 GO
4 SELECT * FROM SanPham
5 GO
6 DECLARE @error int
7 exec sp_ThemChiTietDonHang_TC '0000000001', >
     '000001', 20, @error = @error output
9 GO
10 SELECT * FROM SanPham
11 GO

```

The right window, titled 'Tran_2_TestCase01....E (NGA\Admin (61))', contains the following script:

```

1 --số lượng tồn sản phẩm '000001' đang là 50
2 USE HT_DHCH_ONLINE
3 GO
4 SELECT * FROM SanPham
5 GO
6 DECLARE @error int
7 exec sp_ThemChiTietDonHang_TC '0000000002', >
     '000001', 15, @error = @error output
8 GO
9 --sau khi chạy tran 2 sẽ in ra số lượng tồn >
     --đang ra nên có trong SanPham
10 --số lượng tồn được cập nhật trong SanPham
11 SELECT * FROM SanPham
12 select * from CT_DonHang

```

Both windows show results tables below the scripts. The left window's results table has one row: MaSP 000001, TenSP Áo thun Mickey, GiaBan 50000, SLTon 50. The right window's first results table also has one row: MaSP 000001, TenSP Áo thun Mickey, GiaBan 50000, SLTon 50. Its second results table has one row: MaSP 000001, TenSP Áo thun Mickey, GiaBan 50000, SLTon 35. Its third results table has two rows:

MaDH	MaSP	SoLuong	GiaBan	ThanhTien
0000000001	000001	20	50000	1000000
0000000002	000001	15	50000	750000

2 Demo trên Giao diện: Hai khách hàng cùng thêm một sản phẩm

The screenshot shows a web application interface with two separate browser windows. Both windows display a 'My Cart' section. In the left window, a user adds a product with ID 000001 (Áo thun Mickey) into their cart, setting the quantity to 20. In the right window, another user adds the same product with ID 000001 (Áo thun Mickey) into their cart, setting the quantity to 15. Both windows show a total value of \$10,000,000.00.

Số lượng tồn bị sai vì chỉ được cập nhật theo khách hàng thứ 2

The screenshot shows a 'Danh Sách Sản phẩm' (Product List) page. A table displays a single product entry:

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000001	Áo thun Mickey	50000	35	<button>+ ↎</button>

The 'Số lượng tồn' (Stock Quantity) field is highlighted with a red box, indicating an error. The current value is 35, which is incorrect because the system only updated the stock for the second customer who added 15 units.



1.2. Test Case 2:

Tình huống: Hai tài xế cùng đọc 1 đơn hàng có tình trạng là đang chờ giao hàng để bắt đầu nhận đơn, tuy nhiên vì đọc trước khi cả hai nhận đơn nên sẽ xảy ra tình trạng cả 2 tài xế cùng nhận 1 đơn hàng.

T1: Tài xế 1 đọc và nhận đơn hàng ‘0000000001’ T2: Tài xế 2 đọc và nhận đơn hàng ‘0000000001’ * Đơn hàng ‘0000000001’ đang có tình trạng là 3 (Đang chờ giao hàng)	
EXEC sp_TaiXeNhanDonHang_TC '012456781', '0000000001'	EXEC sp_TaiXeNhanDonHang_TC '012456782', '0000000001'
BEGIN TRAN	
	BEGIN TRAN
B1: B1: Kiểm tra tình trạng đơn hàng <pre>DECLARE @ttdh int, @mota nvarchar(100) Set @ttdh = (SELECT MaTT FROM CT_TTDH ttd1 WHERE ttd1.MaDH = @MaDH AND ttd1.NgayCapNhat >= ALL(SELECT ttd2.NgayCapNhat FROM CT_TTDH ttd2 WHERE ttd2.MaDH = ttd1.MaDH)) SET @mota = (select Mota from TinhTrangDH where @ttdh = MaTinhTrang) print(@mota) ⇒ Tình trạng đơn hàng = 3</pre>	
	B1: Kiểm tra tình trạng đơn hàng <pre>DECLARE @ttdh int, @mota nvarchar(100) Set @ttdh = (SELECT MaTT FROM CT_TTDH ttd1 WHERE ttd1.MaDH = @MaDH AND ttd1.NgayCapNhat >= ALL(SELECT ttd2.NgayCapNhat FROM CT_TTDH ttd2 WHERE ttd2.MaDH = ttd1.MaDH)) SET @mota = (select Mota from TinhTrangDH where @ttdh = MaTinhTrang) print(@mota) ⇒ Tình trạng đơn hàng = 3</pre>
B2: Kiểm tra đơn hàng có đang được chờ giao hàng hay không <pre>IF @ttdh < 3 BEGIN</pre>	



```
PRINT (N'Dơn hàng chưa sẵn sàng để
giao')
RAISERROR ('1',15,1)
END
ELSE IF @ttdh = 3
BEGIN
DECLARE @PhiVanChuyen int
SET @PhiVanChuyen = (SELECT
PhiVanChuyen FROM DonHang WHERE MaDH =
@MaDH)
INSERT INTO ThuNhapTX VALUES (@MaTX,
@MaDH, @PhiVanChuyen)

INSERT INTO CT_TTDH VALUES (GETDATE(),
@MaDH, 4)
END
ELSE
begin
print (N'Dơn hàng đã có người nhận')
raiserror ('2',15,1);
end
⇒ Thỏa điều kiện tình trạng lần cập nhật mới nhất
của đơn hàng = 3. Thực hiện insert
```

B2: Kiểm tra đơn hàng có đang được chờ giao hàng hay không

```
IF @ttdh < 3
BEGIN
PRINT (N'Dơn hàng chưa sẵn sàng
để giao')
RAISERROR ('1',15,1)
END
ELSE IF @ttdh = 3
BEGIN
DECLARE @PhiVanChuyen int
SET @PhiVanChuyen = (SELECT
PhiVanChuyen FROM DonHang WHERE MaDH =
@MaDH)
INSERT INTO ThuNhapTX VALUES
(@MaTX, @MaDH, @PhiVanChuyen)

INSERT INTO CT_TTDH VALUES
(GETDATE (), @MaDH, 4)
END
ELSE
begin
print (N'Dơn hàng đã có người nhận')
raiserror ('2',15,1)
end
```

	⇒ Thỏa điều kiện tình trạng lần cập nhật mới nhất của đơn hàng = 3. Thực hiện insert
COMMIT TRAN	
	COMMIT TRAN

⇒ **T1 và T2 cùng một lúc nhận cùng một đơn hàng**

Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two separate SQL Server query windows side-by-side. Both windows are running the same stored procedure: sp_TaiXeNhanDonHang_TC. The parameters passed are '012456781', '0000000001', and @msg = @msg output. The results from both queries are identical, showing a message 'Đang chờ giao hàng.' and a value '3' in the TTDH column.

```

Tran_1_TestCase02...phy01@gmail.com)  Tran_2_TestCase02...phy01@gmail.com)
Use HT_DHCH_ONLINE
GO
declare @msg nvarchar(100)
EXEC sp_TaiXeNhanDonHang_TC '012456781', '0000000001',@msg=@msg output
SELECT @msg
Results Messages
MoTa
1 | Đang chờ giao hàng.

TTDH
1 | 3

(No column name)
1 | Nhận đơn thành công
Click to select the whole column

Tran_2_TestCase02...phy01@gmail.com)  Tran_1_TestCase02...phy01@gmail.com)
Use HT_DHCH_ONLINE
GO
declare @msg nvarchar(100)
EXEC sp_TaiXeNhanDonHang_TC '012456782', '0000000001',@msg=@msg output
SELECT @msg
Results Messages
MoTa
1 | Đang chờ giao hàng.

TTDH
1 | 3

(No column name)
1 | Nhận đơn thành công

```

Cả hai tài xế cùng đặt và cả hai đều nhận được đơn hàng

The screenshot displays a web-based application interface. At the top, there is a header for 'Thông tin Tài Xe'. Below it, there is a section for driver details: 'Mã Tài Xe: 047733459124', 'Họ và tên: Tô Huy Thành', and 'Số điện thoại: 012456782'. A red box highlights the phone number. Below this, there is a section titled 'Danh sách thu nhập' (Income list) with a sub-section 'Cập nhật đơn hàng'. It includes input fields for 'Mã Đơn hàng' and 'Mã Tình Trạng' with a 'Cập nhật' button. The main content area shows a table of delivery orders:

#	Mã Đơn hàng	Phí Vận chuyển
1	0000000001	75000



Thông tin Tài Xe

Mã Tài Xe: 012317983262

Họ và tên: Huỳnh Bá Vỹ

Số điện thoại: 012456781

Danh sách thu nhập

Cập nhật đơn hàng

#	Mã Đơn hàng	Phí Vận chuyển
1	000000001	75000

1.3. Test Case 3:

Tình huống: Hai đối tác cùng thêm sản phẩm mới vào bảng SanPham, mã sản phẩm phải được tăng lên theo đúng thứ tự thêm vào. Nhưng do cả hai transaction T1 và T2 cùng đọc một lúc dẫn đến tình trạng MaSP tăng lên không theo đúng thứ tự nên T2 sẽ không thêm được do bị trùng mã.

Tình huống tương tự có thể xảy ra khi tạo thêm đối tác, thêm đơn hàng

T1	T2
sp_ThemSanPham_TC N'Nước rửa chén',500,6	sp_ThemSanPham_TC N'Xà bông tắm',500,6
BEGIN TRAN	
	BEGIN TRAN
<i>B1: Đọc mã sản phẩm lớn nhất từ bảng SanPham</i> declare @ma_count bigint, @MaSP varchar(6) set @MaSP = (select TOP 1 (MaSP) from SanPham order by MaSP DESC) $\Rightarrow @MaSP=000001$	
	<i>B1: Đọc mã sản phẩm lớn nhất từ bảng SanPham</i> declare @ma_count bigint, @MaSP varchar(6) set @MaSP = (select TOP 1 (MaSP) from SanPham order by MaSP DESC) $\Rightarrow @MaSP=000001$



<p>B2: Kiểm tra bảng SanPham trước đó đã có sản phẩm nào chưa, nếu có thì tăng MaSP mới lên 1 đơn vị.</p> <pre>if (isnull(@MaSP, 'false') <> 'false') begin set @ma_count=cast (@MaSP as bigint) + 1 end else begin set @ma_count=1 end set @MaSP = RIGHT ('00000' + CAST (@ma_count AS VARCHAR (6)), 6) ⇒ @MaSP=000002</pre>	
---	--

	<p>B2: Kiểm tra bảng SanPham trước đó đã có sản phẩm nào chưa, nếu có thì tăng MaSP mới lên 1 đơn vị.</p> <pre>if (isnull(@MaSP, 'false') <> 'false') begin set @ma_count=cast (@MaSP as bigint) + 1 end else begin set @ma_count=1 end set @MaSP = RIGHT ('00000' + CAST (@ma_count AS VARCHAR (6)), 6) ⇒ @MaSP=000002.</pre>
--	--

<p>B3: Thêm sản phẩm vào Table</p> <pre>INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) ⇒ Insert thành công vào bảng SanPham (@MaSP=000002).</pre>	
--	--

	<p>B3: Thêm sản phẩm vào Table</p> <pre>INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) ⇒ @MaSP bị trùng, không thể insert được.</pre>
--	--

COMMIT TRAN

ROLLBACK TRANSACTION

Hình ảnh demo:

2 Demo trên SQL SERVER

The screenshot shows two side-by-side SSMS windows. Both windows have tabs labeled 'Tran_1_TestCase04...E (NGA\Admin (52))' and 'Tran_2_TestCase04...E (NGA\Admin (55))'. The left window contains the following script:

```

1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase04
4 --Transaction 1
5 EXEC sp_ThemSanPham_TC N'Nước rửa chén', 500, 6
    
```

The right window contains:

```

1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase04
4 --Transaction 2
5 EXEC sp_ThemSanPham_TC N'Xà bông tắm', 500, 6
    
```

Both windows show a 'Messages' pane at the bottom with the message '000002' and '(1 row affected)' or '(0 rows affected)'.

Demo trên Giao diện: Cả 2 đối tác cùng thêm sản phẩm

The screenshot shows two identical-looking product addition forms. Both forms have a pink header 'Vngalam' and a title 'Thêm Sản phẩm'. The fields are as follows:

- Tên sản phẩm: Nước rửa chén
- Giá bán: 500
- Số lượng tồn: 6
-

The right form's fields are highlighted in blue, indicating they are being edited.

Chỉ 1 đối tác thêm sản phẩm thành công: MaSP=000002

The page title is 'Danh Sách Sản phẩm Của Đối Tác'. At the top right are buttons for 'Cập nhật sản phẩm' (Update product) and 'Thêm sản phẩm' (Add product). The main content is a table:

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1		000001	Nước hoa	50000	32
2		000002	Nước rửa chén	500	6

1.4. Test Case 4:

Tình huống: Đối tác đang thêm sản phẩm mới, cùng lúc đó người dùng khác đọc sản phẩm, nhưng sau đó sản phẩm do có số lượng tồn nhỏ hơn 0 nên không được chấp nhận thêm vào bảng dẫn đến Rollback. Khi đó, sản phẩm người dùng đọc được không tồn tại nữa, gây ra vấn đề DirtyRead.

- T1: Thêm sản phẩm mới
- T2: Đọc danh sách sản phẩm
- T1: Rollback → T2 đọc sai thông tin

T1	T2
sp_ThemSanPham_TC N'Óp điện thoại', 200, -1	
BEGIN TRAN	
	BEGIN TRAN
<p><i>B1: Lấy ra MaSP cao nhất trong bảng SanPham để tăng thêm 1 đơn vị vào MaSP mới</i></p> <pre>declare @ma_count bigint, @MaSP varchar(6) set @MaSP = (select TOP 1 (MaSP) from SanPham order by MaSP DESC) ⇒ @MaSP=000003. if (isnull(@MaSP, 'false') <> 'false') begin set @ma_count=cast (@MaSP as bigint)+1 end else begin set @ma_count=1 end set @MaSP = RIGHT ('00000' + CAST (@ma_count AS VARCHAR(6)), 6) ⇒ @MaSP=000004.</pre>	
<p><i>B2: Thêm sản phẩm vào Table</i></p> <pre>INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) ⇒ Insert thành công vào bảng SanPham (@MaSP=000004).</pre>	
	<p><i>Đọc bảng SanPham</i></p> <pre>select * from SanPham with (nolock) ⇒ Kết quả trả về có thông tin của (000004, Óp điện thoại, 200, -1) trong danh sách Sản Phẩm.</pre>
	<p>COMMIT TRAN</p> <p>⇒ Đã đọc xong và COMMIT.</p>



B3: Kiểm tra số lượng tồn <0

```
if (@SLTon < 0)
begin
    print ('1')
    raiserror (N'không được nhỏ hơn
0', 15, 1)
end
⇒ @SLTon=-1 nên ROLLBACK.
```

ROLLBACK TRANSACTION

Hình ảnh demo:

Demo trên SQL SERVER

The screenshot shows two SQL Server Management Studio windows side-by-side. The left window, titled 'Tran_1_TestCase04.s...Administrator (62)', contains the following script:

```
1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase04
4 --Transaction 1
5 exec sp_ThemSanPham_TC N'Ốp
điện thoại', 200, -1
```

The right window, titled 'Tran_2_TestCase04.s...Administrator (63)', contains the following script:

```
1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase04
4 --Transaction 2
5 begin tran
6 waitfor delay '00:00:09'
7 select * from SanPham with
(nolock)
8 COMMIT
```

Both windows show the results pane with the following data:

Mã SP	Tên SP	Giá Ban	Số Lượng
000001	Áo thun Mickey	50000	32
000002	Áo thun Minnie	45000	4
000003	Áo thun Donald	46000	7
000004	Ốp điện thoại	200	-1

Demo trên Giao diện: Đối tác thêm sản phẩm nhưng chưa Commit

The screenshot shows a web-based application interface for adding a product. The top navigation bar includes links for 'Đơn hàng', 'Sản Phẩm', and 'Quản lý'. The main form is titled 'Thêm Sản phẩm' and contains the following fields:

- Tên sản phẩm:
- Giá bán:
- Số lượng tồn:
- Lưu



Demo trên Giao diện: Cùng lúc đó thì khách hàng lại đọc được sản phẩm chưa commit

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000001	Áo thun Mickey	50000	32	<button>+ </button>
2		000002	Áo thun Minnie	45000	4	<button>+ </button>
3		000003	Áo thun Donald	46000	7	<button>+ </button>
4		000004	Óp điện thoại	200	-1	<button>+ </button>

1.5. Test Case 5:

Tình huống: Tài xế cập nhật tình trạng đơn hàng và ngay sau đó khách hàng truy cập và xem tình trạng đơn hàng nhưng quá trình cập nhật của tài xế xảy ra lỗi nên phải rollback điều này khiến cho dữ liệu tình trạng đơn hàng mà khách hàng xem được trở thành dữ liệu rác.

T1: Tài xế cập nhật tình trạng đơn hàng.

T2: Khách hàng xem tình trạng đơn hàng.

T1	T2
exec sp_CapNhatTinhTrangDonHang_TC '0000000001', 4;	
BEGIN TRAN	
	BEGIN TRAN
-- Tài xế thêm một dòng vào bảng tình trạng đơn hàng với MaDH là 'DH0000' và mã tình trạng là 4 (Đơn hàng đang giao) insert into CT_TTDH values (GETDATE (), @MaDH, @MaTT)	-- Khách hàng xem tình trạng của đơn hàng có MaDH là 'DH0000', tình trạng đơn hàng mới nhất sẽ là dòng dữ liệu có ngày cập nhật mới nhất. Đặt isolation level là READ UNCOMMITTED để giao tác có thể đọc giá trị chưa được commit.

	<pre>SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED; SELECT * FROM CT_TTDH WHERE MaDH= '0000000001' ⇒ @MaTT mới nhất = 4</pre>
-- Vì một lỗi nào đó khiến cho quá trình thêm tình trạng đơn hàng của tài xế không thành công nên T1 rollback. ROLLBACK TRANSACTION	
	COMMIT TRAN

⇒ Sau khi giao tác T2 commit, khách hàng xem được tình trạng đơn hàng là 'Đơn hàng đang giao' nhưng giao tác T1 thêm tình trạng đơn hàng do tài xế thực hiện đã xảy ra lỗi và rollback nên tình trạng đơn hàng đúng ra nên là 'Đang chờ giao hàng' (mã tình trạng là 3).

Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two separate SQL Server sessions side-by-side.

Session 1 (Tran_1):

```

1 USE HT_DHCH_ONLINE
2 Go
3
4 exec sp_CapNhatTinhTrangDonHang_TC
    '0000000001', 4;
5
6 select * from CT_TTDH where MaDH =
    '0000000001';
7 select * from TinhTrangDH
8
9

```

Session 2 (Tran_2):

```

1 USE HT_DHCH_ONLINE
2 Go
3
4 begin tran
5 SET TRANSACTION ISOLATION LEVEL
    READ UNCOMMITTED;
6 select * from CT_TTDH where
    MaDH='0000000001';
7
8
9
10

```

Results Grid (Tran_1):

NgayCapNhat	MaDH	MaTT
2021-11-11 00:00:00.000	0000000001	1
2021-11-13 00:00:00.000	0000000001	2
2021-11-14 00:00:00.000	0000000001	3

Results Grid (Tran_2):

NgayCapNhat	MaDH	MaTT
2021-11-11 00:00:00.000	0000000001	1
2021-11-13 00:00:00.000	0000000001	2
2021-11-14 00:00:00.000	0000000001	3
2021-12-19 00:58:12.680	0000000001	4

Lookup Table (TinhTrangDH):

MaTinhTrang	Mota
0	Đơn hàng đã hủy.
1	Đơn hàng được tiếp nhận.
2	Đơn hàng đang đóng gói.
3	Đang chờ giao hàng.
4	Đơn hàng đang giao.
5	Đơn hàng giao thành công.



Demo trên Giao diện: Khách hàng đọc được tình trạng đơn hàng chưa được Commit



Mã khách hàng: 0909123450
Tên người nhận: Trần Văn A
Số điện thoại: 0909123450

Địa chỉ:
Phường:
Quận:
Tỉnh:

Đơn hàng
Mã đơn hàng: 0000000001
Ngày lập: 10 Nov 2021 00:00:00:000

#	Mã sản phẩm	Tên sản phẩm	Giá Bán	Số Lượng	Thành Tiền
					Tổng tiền hàng
					Phi vận chuyển
					Tổng số tiền
					Phương thức thanh toán

Thanks for choosing us.

Tình trạng đơn hàng

Ngày	Mã tình trạng	Mã tình trạng	Tình trạng
11 Nov 2021 00:00:00:000	1	0	Đơn hàng đã hủy
13 Nov 2021 00:00:00:000	2	1	Đơn hàng được tiếp nhận
14 Nov 2021 00:00:00:000	3	2	Đơn hàng đang đóng gói
18 Dec 2021 00:31:37:823	4	3	Đang chờ giao hàng
		4	Đơn hàng đang giao
		5	Đơn hàng giao thành công

Tài xé cập nhật đơn hàng

Danh sách thu nhập

Cập nhật đơn hàng

0000000001	4	Cập nhật
------------	---	----------

#	Mã Đơn hàng	Phi Vận chuyển

TRANG TRƯỚC
TRANG KẾ

1.6. Test Case 6:

Tình huống: Khách hàng muốn xem các sản phẩm có giá dưới 100000, sau đó đổi tác thực hiện thêm một sản phẩm có giá dưới 100000 nên khi khách hàng xem lại các sản phẩm có giá dưới 100000 sẽ nhận được số bộ dữ liệu khác nhau giữa hai lần đọc.

- T1: Khách hàng xem các sản phẩm có giá dưới 100000.
 T2: Đổi tác thêm một sản phẩm mới tên ‘Bút bi’ có giá là 7000.

T1	T2
<pre>declare @sum int exec sp_KiemTraGiaBan_TC 100000, @Tong = @sum ouput; select @sum as TongSoSanPham</pre>	<pre>exec sp_ThemSanPham N'Bút bi', 7000, 80</pre>
BEGIN TRAN	
-- Khách hàng xem các sản phẩm có giá dưới 100000. <pre>select @Tong = count (MaSP) from SanPham where SanPham.GiaBan < @gb ⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 3.</pre>	
	-- Đổi tác thực hiện việc thêm một sản phẩm có tên là ‘Bút bi’, giá bán là 7000 và số lượng tồn là 80. <pre>exec sp_ThemSanPham N'Bút bi', 7000, 80</pre>
-- Khách hàng xem lại các sản phẩm có giá dưới 100000. <pre>SELECT ROW_NUMBER () OVER (ORDER BY MaSP) AS ROWNUMBER, MaSP, TenSP, GiaBan, SLTon FROM SanPham WHERE GiaBan < @gb ⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 4.</pre>	
COMMIT TRAN	
<p>⇒ Cùng trong giao tác T1 nhưng hai thời điểm đọc khác nhau lại cho ra số bộ dữ liệu thỏa điều kiện khác nhau.</p>	



Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two side-by-side SQL Server Management Studio windows. Both windows have the title bar 'Tran_1_TestCase06....E (NGA\Admin (55))' and 'Tran_2_TestCase06....E (NGA\Admin (56))'. The left window contains the following script:

```
1 Use HT_DHCH_ONLINE
2 go
3
4 --TestCase06
5   --Transaction 1
6   declare @sum int
7   exec sp_KiemTraGiaBan_TC 100000,
     @Tong = @sum output
8   select @sum as TongSoSanPham
```

The right window contains the following script:

```
1 Use HT_DHCH_ONLINE
2 go
3
4 --TestCase06
5   --Transaction 2
6   exec sp_ThemSanPham_TC N'Bút bi',
     7000, 80
7
```

Below the scripts, both windows show the 'Results' tab with the following data:

ROWNUMBER	Masp	TenSP	GiaBan	SLTon
1	000002	Áo thun	89000	450
2	000003	Bông cài	20000	28
3	000005	Dép	65000	1100
4	000007	Bút bi	7000	80

The right window also shows the 'Messages' tab with the following output:

```
000007
(1 row affected)
Completion time: 2021-12-19T00:43:13.0319253+07:
```

Demo Giao diện: Đối tác 2 thêm sản phẩm trong lúc đối tác 1 đang đọc sản phẩm

The screenshot shows a web-based application interface. At the top, there is a navigation bar with tabs: 'Vingroup' (highlighted in red), 'Đơn hàng', 'Sản Phẩm', and 'Quản lý'. Below the navigation bar, there is a sub-navigation bar with tabs: 'Thống kê', 'Nhập kho', 'Xuất kho', 'Nhập xuất', 'Tìm kiếm', and 'Nhập xuất'. The main content area has a title 'Thêm Sản phẩm'. It contains the following form fields:

- Tên sản phẩm:
- Gía bán:
- Số lượng tồn:
-



Đối tác 1 kiểm tra sản phẩm thì tổng số sản phẩm khác với số sản phẩm được in ra

Vyngalam

Đơn hàng Sản Phẩm Quản lý

Kiểm tra Sản phẩm

Số lượng tồn

Tổng Sản Phẩm: 3

#	Mã sản phẩm	Tên sản phẩm	Gía bán	Số lượng tồn
1	000002	Áo thun	89000	450
2	000003	Bông cài	20000	28
3	000005	Dép	65000	1100
4	000007	Bút bi	7000	80

1.7. Test Case 7:

Tình huống: Đối tác xem sản phẩm có số lượng tồn < 10 để nhập thêm hàng, tuy nhiên người dùng lại thêm 1 sản phẩm vào đơn đặt hàng khiến cho số lượng sản phẩm có slt < 10 tăng lên. Đến đây đối tác đọc được 2 bộ dữ liệu khác nhau.

T1: Đối tác đọc số lượng tồn
T2: Khách hàng đặt hàng
***Cho số lượng tồn của sản phẩm '000001' là 32**

T1	T2
<pre>declare @sum int exec sp_KiemTraSLTon_TC 10, @Tong = @sum output; select @sum as NumberOfProduct</pre>	<pre>DECLARE @error int exec sp_ThemChiTietDonHang_TC '0000000001', '000001', 25, @error = @error output</pre>
BEGIN TRAN	
	BEGIN TRAN
--Đối tác xem các sản phẩm có số lượng tồn dưới 10 <pre>select @Tong = count (MaSP) from SanPham where SanPham.SLTon < @slt waitfor delay '00:00:12'</pre>	
⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 2.	

	-- Khách hàng thực hiện thêm sản phẩm '000001' vào đơn hàng '0000000001' với số lượng là 25 <code>INSERT INTO CT_DonHang(MaDH, MaSP, SoLuong) VALUES(@MaDH, @MaSP, @SoLuong)</code>
--Đối tác in ra các sản phẩm có số lượng tồn dưới 10 <code>SELECT ROW_NUMBER() OVER (ORDER BY MaSP) AS ROWNUMBER, MaSP, TenSP, GiaBan, SLTon FROM SanPham WHERE SLTon < @slt</code> ⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 3.	
COMMIT TRAN	
	COMMIT TRAN

⇒ *T1 đọc được số bộ dữ liệu khác nhau ở hai thời điểm.*

Hình ảnh demo:

Demo trên SQL Server

```

Tran_1_TestCase07.s...Administrator (52))  ↗ X
1  use HT_DHCH_ONLINE
2  GO
3
4  declare @sum int
5  exec sp_KiemTraSLTon_TC 10,
    @Tong = @sum output;
6  select @sum as
    NumberofProduct

Tran_2_TestCase07.s...Administrator (54))  ↗ X
1  use HT_DHCH_ONLINE
2  GO
3  DECLARE @error int
4
5  exec sp_ThemChiTietDonHang_TC
    '0000000001', '000001',
    25, @error = @error output
6  GO

```

ROWNUMBER	MaSP	TenSP	GiaBan	SLTon
1	000001	Áo thun Mickey	50000	7
2	000002	Áo thun Minnie	45000	4
3	000003	Áo thun Donald	46000	7

MaSP	TenSP	GiaBan	SLTon
000001	Áo thun Mickey	50000	7
000002	Áo thun Minnie	45000	4
000003	Áo thun Donald	46000	7



Demo trên Giao diện: Khách hàng mua sản phẩm

#	Image	Mã sản phẩm	Tên sản phẩm	Gía bán	Số lượng tồn
1		000001	Áo thun Mickey	\$50000.00	25
2		000002	Áo thun Minnie	45000	4
3		000003	Áo thun Donald	46000	7

Trước đó đối tác đang kiểm tra số lượng tồn <10 thì tổng số với số sản phẩm in ra khác nhau

#	Mã sản phẩm	Tên sản phẩm	Gía bán	Số lượng tồn
1	000001	Áo thun Mickey	50000	7
2	000002	Áo thun Minnie	45000	4
3	000003	Áo thun Donald	46000	7

1.8. Test Case 8:

Tình huống: Người dùng tìm sản phẩm có mã là ‘000001’ và tên là ‘Nước hoa’. Hệ thống xác nhận có tồn tại sản phẩm, nhưng sau đó đối tác lại cập nhật tên sản phẩm thành ‘Áo hai dây’. Khiến cho lần đọc tiếp theo của người dùng không đọc được thông tin sản phẩm dù sản phẩm có tồn tại.

T1: Kiểm tra thông tin sản phẩm với @MaSP và @TenSP

T2: Cập nhật sản phẩm với tên mới

T1: Xem thông tin sản phẩm với @MaSP và @TenSP ban đầu → không hiện thông tin.

T1	T2
<pre>exec sp_KiemTraSP_TC '000001', N'Nước hoa', @result = @result output</pre>	<pre>EXEC sp_CapNhatSanPham_TC '000001', N'Áo hai dây', 300, 3</pre>
BEGIN TRAN	
	BEGIN TRAN
<p><i>B1: Kiểm tra sản phẩm có mã '000001' và tên là 'Nước hoa' có tồn tại không</i></p> <pre>if not exists (select * from SanPham where @MaSP=MaSP and @TenSP= TenSP) raiserror (N'Không tồn tại sản phẩm',15,1) set @result=1 waitfor delay '00:00:08' ⇒ @MaSP và @TenSP có tồn tại, @result=1</pre>	
	<p><i>Kiểm tra sản phẩm có tồn tại không, sau đó cập nhật lại tên sản phẩm</i></p> <pre>if not exists (select * from SanPham where @MaSP= MaSP) begin print ('1') raiserror (N'Không tồn tại sản phẩm',15,1) end waitfor delay '00:00:05' update SanPham set TenSP= @TenSP, GiaBan= @GiaBan, SLTon= @SLTon where MaSP= @MaSP ⇒ Sản phẩm '000001' được cập nhật tên thành công từ 'Nước hoa' sang 'Áo hai dây'</pre>
<p><i>B2: Đọc thông tin của sản phẩm có mã '000001' và tên là 'Nước hoa'</i></p> <pre>select * from SanPham where MaSP= @MaSP and TenSP= @TenSP ⇒ Không có gì xuất hiện vì tên sản phẩm không trùng khớp</pre>	
	COMMIT TRAN
COMMIT TRAN	



Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two side-by-side SQL Server Management Studio windows. Both windows have tabs for 'Results' and 'Messages'.

Tran_1_TestCase11...E (NGA\Admin (56))*

```
1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase11
4 --Transaction 1
5 declare @result int
6 exec sp_KiemTraSP_TC '000001',N'Nước hoa',@result = @result output
```

Tran_2_TestCase11...E (NGA\Admin (57))*

```
1 USE HT_DHCH_ONLINE
2 GO
3 --TestCase11
4 --Transaction 2
5 EXEC sp_CapNhatSanPham_TC '000001',N'Áo hai dây',300,3
```

Results tab content:

Mã SP	Tên SP	Giá Ban	Số Lượng
1	Áo hai dây	300	3
2	Áo thun Minnie	45000	4
3	Áo thun Donald	46000	7

Messages tab content:

Output from Tran_1_TestCase11: 0 rows

Output from Tran_2_TestCase11: 3 rows

Demo trên Giao diện: Người dùng kiểm tra sản phẩm nhưng đổi tác lại cập nhật sản phẩm đó thành tensp khác nên lúc người dùng in sản phẩm với masp và tensp cũ thì không in dc mặc dù tình trạng hiện là tồn tại

The screenshot shows a web application interface with two main panels: 'Kiểm tra Sản phẩm' (Search Product) and 'Cập nhật Sản phẩm' (Update Product).

Kiểm tra Sản phẩm

Form fields:
Mã sản phẩm: 000001
Tên sản phẩm: Nước hoa
Tình trạng: 1

Table:

#	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
	000001	Nước hoa		1

Cập nhật Sản phẩm

Form fields:
Mã sản phẩm: 000001
Tên sản phẩm: Áo hai dây
Giá bán: 10000
Số lượng tồn: 5

Lưu button

1.9. Test Case 9:

Tình huống: Giả sử T1 và T2 thay phiên nhau update 2 sản phẩm, lúc này sẽ xảy ra tình trạng circle deadlock do cả hai phải chờ nhau.

T1	T2
BEGIN TRAN	
	BEGIN TRAN
<pre>UPDATE SanPham set SLTon = 100 where MaSP = '000001'; (1)</pre> <p>--Thành công. Lúc này table SanPham đang lock row vừa update của (1)</p>	
	<pre>UPDATE SanPham set SLTon = 200 where MaSP = '000002'; (2)</pre> <p>--Thành công. Lúc này table SanPham đang lock row vừa update của (2)</p>
<pre>UPDATE SanPham set SLTon = 100 where MaSP = '000002'; (3)</pre> <p>--Không thể thực thi được đang ở trạng thái wait do row có MaSP='000002' đang bị lock</p>	
	<pre>UPDATE SanPham set SLTon = 200 where MaSP = '000001'; (4)</pre> <p>--Tương tự (3) do row có MaSP='000001' đang bị lock \Rightarrow DEADLOCK</p>
--Bắt buộc phải kill 1 transaction => Kill tran 2 COMMIT TRAN	
	Bị KILL và báo EXCEPTION

\Rightarrow (4) của transaction 2 bị kill và báo lỗi.

Hình ảnh demo:

```

Tran_1_TestCase09...E (NGA\Admin (61))  TestCase09.sql - N...E (NGA\Admin (60))*  Tran_2_TestCase09...E (NGA\Admin (62))  TestCase09.sql - N...E (NGA\Admin (62))*  Messages  Messages  Messages  Messages
1 USE HT_DHCH_ONLINE 1 USE HT_DHCH_ONLINE
2 GO 2 GO
3 BEGIN TRAN 3 BEGIN TRAN
4 UPDATE SanPham set SLTon = 100 where MaSP = 4 UPDATE SanPham set SLTon = 200 where MaSP =
   '000001';  '000002';
5 WAITFOR DELAY '00:00:10' 5 WAITFOR DELAY '00:00:10'
6 UPDATE SanPham set SLTon = 100 where MaSP = 6 UPDATE SanPham set SLTon = 200 where MaSP =
   '000002';  '000001';
7 COMMIT TRAN 7 COMMIT TRAN
8

```

(1 row affected)
(1 row affected)

Completion time: 2021-12-19T19:29:44.6715784+07:00

(1 row affected)
Msg 1205, Level 13, State 51, Line 6
Transaction (Process ID 62) was deadlocked on lock resources wi...

Completion time: 2021-12-19T19:29:44.6097431+07:00

Query executed successfully | NGA\SQLEXPRESS (15.0 RTM) | NGA\Admin (61) | HT_DHCH_ONLINE | 00:00:13 | 0 rows

Query completed with error | NGA\SQLEXPRESS (15.0 RTM) | NGA\Admin (62) | HT_DHCH_ONLINE | 00:00:12 | 0 rows

2. Giải quyết tình huống tranh chấp

2.1. Test Case 1:

Tình huống: Ở tình huống này trigger sẽ update mỗi khi một chi tiết đơn hàng được insert. Hai khách hàng đặt hàng cùng 1 lúc, nhưng số lượng tồn của 2 người đọc được lại ở thời điểm ban đầu khiến cho lưu trữ thông tin số lượng tồn bị sai lệch đi.

T1: Khách hàng 1 thêm sản phẩm vào chi tiết đơn hàng (Mã HD: 0000000001)

T2: Khách hàng 2 thêm sản phẩm vào chi tiết đơn hàng (Mã HD: 0000000002)

Xem số lượng tồn của sản phẩm '000001' = 50

T1	T2
exec sp_ThemChiTietDonHang '0000000001', '000001', 20	exec sp_ThemChiTietDonHang '0000000002', '000001', 15
BEGIN TRAN	
	EXEC('DISABLE TRIGGER slt_dathang ON CT_DonHang'); BEGIN TRAN
B1: Kiểm tra đơn hàng có tồn tại hay không if not exists (select * from DonHang where @MaDH= MaDH) begin	



<pre> print ('1') raiserror (N'Không tồn tại đơn hàng',15,1) end </pre>	
	<p>B1: Kiểm tra đơn hàng có tồn tại hay không</p> <pre> if not exists (select * from DonHang where @MaDH= MaDH) begin print ('1') raiserror (N'Không tồn tại đơn hàng',15,1) end </pre>
<p>B2: Xem số lượng tồn ban đầu (@sl) và in ra @sl</p> <pre> declare @sl int set @sl = (select SLTon from SanPham with (XLOCK, ROWLOCK) where MaSP = @MaSP) print(@sl) ⇒ SLT = 50. </pre>	
<p>B3: Kiểm tra có vượt quá số lượng tồn trong kho hay không, nếu không vượt quá → kiểm tra chi tiết đơn hàng có tồn tại thì cộng thêm số lượng, ngược lại tạo một chi tiết đơn hàng mới</p> <pre> if (@sl-@SoLuong>=0) begin if EXISTS (select * from CT_DonHang where MaDH = @MaDH and MaSP = @MaSP) begin waitfor delay '00:00:02' update CT_DonHang set SoLuong = (SoLuong + @SoLuong) where MaDH = @MaDH and MaSP = @MaSP end else begin waitfor delay '00:00:02' INSERT INTO CT_DonHang (MaDH, MaSP, SoLuong) VALUES (@MaDH, @MaSP, @SoLuong) end end ⇒ Ở đây thỏa điều kiện if vì @sl - @SoLuong = 50-20 ≥ 0. </pre>	

	Vì tran 1 giữ XLOCK nên tran 2 chờ tran 1 commit mới có thể thực hiện bước 2
$\Rightarrow \text{insert thành công CTDH '000000001', '000001', 20 \rightarrow \text{update số lượng tồn bằng } 50 - 20 = 30.$ COMMIT TRAN	
	<p>B2: Xem số lượng tồn ban đầu và in ra</p> <pre>declare @sl int set @sl = (select SLTon from SanPham with (XLOCK, ROWLOCK) where MaSP = @MaSP) print(@sl) ⇒ SLT = 30.</pre> <p>B3: Kiểm tra có vượt quá số lượng tồn trong kho hay không, nếu không vượt quá → kiểm tra chi tiết đơn hàng có tồn tại thì cộng thêm số lượng, ngược lại tạo một chi tiết đơn hàng mới</p> <pre>if (@sl - @SoLuong >= 0) begin if EXISTS (select * from CT_DonHang where MaDH = @MaDH and MaSP = @MaSP) begin waitfor delay '00:00:02' update CT_DonHang set SoLuong = (SoLuong + @SoLuong) where MaDH = @MaDH and MaSP = @MaSP end else begin waitfor delay '00:00:02' INSERT INTO CT_DonHang (MaDH, MaSP, SoLuong) VALUES (@MaDH, @MaSP, @SoLuong) UPDATE SanPham SET SLTon = @sl - @SoLuong WHERE MaSP = @MaSP end end ⇒ Ở đây thỏa điều kiện if vì @sl - @SoLuong = 30 - 15 >= 0</pre>

	<p>⇒ <i>insert thành công CTDH '0000000002', '000001', 15 → update số lượng tồn bằng 50-15=15 → GIẢI QUYẾT TRANH CHẤP.</i> COMMIT TRAN</p>
--	--

⇒ Đề xử lý tranh chấp thực hiện:

Khi đọc số lượng sản phẩm để tiến hành đặt sản phẩm, đặt rowlock ngay trên dòng sản phẩm đó.

`select SLTon from SanPham with (XLOCK, ROWLOCK) where MaSP=@MaSP`

XLOCK kết hợp với ROWLOCK để ngăn chặn một session khác đọc trên dòng cho đến hết transaction.

Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two separate SQL Server query windows. Both are running under the 'Administrator' account.

Tran_1_TestCase01.sql:

```

1 --số lượng tồn sản phẩm
  '000001' đang là 50
2 USE HT_DHCH_ONLINE
3 GO
4 DECLARE @error int
5 exec sp_ThemChiTietDonHang
  '0000000001', '000001', 20,
  @error = @error output
6 GO

```

Tran_2_TestCase01.sql:

```

1 --số lượng tồn sản phẩm
  '000001' đang là 50
2 USE HT_DHCH_ONLINE
3 GO
4 DECLARE @error int
5 exec sp_ThemChiTietDonHang
  '0000000002', '000001', 15,
  @error = @error output
6 GO
7 -----sau khi chạy tran 2 sẽ in
      --số lượng tồn đã bị thay đổi

```

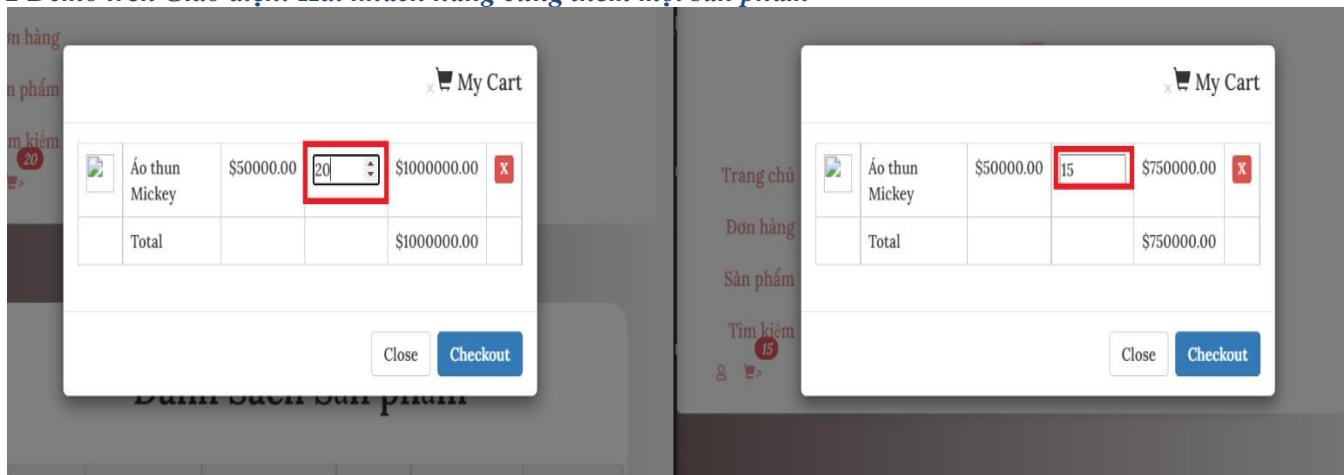
Both windows show results tables:

MaSP	TenSP	GiaBan	SLTon
1 000001	Áo thun Mickey	50000	30

MaSP	TenSP	GiaBan	SLTon
1 000001	Áo thun Mickey	50000	15

MaDH	MaSP	Soluong	GiaBan	ThanhTien
1 0000000001	000001	20	50000	1000000
2 0000000002	000001	15	50000	750000

2 Demo trên Giao diện: Hai khách hàng cùng thêm một sản phẩm





Số lượng tồn được cập nhật đúng

Danh Sách Sản phẩm						
#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000001	Áo thun Mickey	50000	15	

2.2. Test Case 2:

Tình huống: Hai tài xế cùng đọc 1 đơn hàng có tình trạng là đang chờ giao hàng để bắt đầu nhận đơn, tuy nhiên vì đọc trước khi cả hai nhận đơn nên sẽ xảy ra tình trạng cả 2 tài xế cùng nhận 1 đơn hàng.

T1: Tài xế 1 đọc và nhận đơn hàng '0000000001' T2: Tài xế 2 đọc và nhận đơn hàng '0000000001' * Đơn hàng '0000000001' đang có tình trạng là 3 (Đang chờ giao hàng)	
EXEC sp_TaiXeNhanDonHang_TC '012456781', '0000000001'	EXEC sp_TaiXeNhanDonHang_TC '012456782', '0000000001'
BEGIN TRAN	
	BEGIN TRAN
<p><i>B1: Kiểm tra tình trạng đơn hàng</i></p> <pre>DECLARE @ttdh int, @mota nvarchar(100) Set @ttdh = (SELECT MaTT FROM CT_TTDH ttd1 with (XLOCK, ROWLOCK) WHERE ttd1.MaDH = @MaDH AND ttd1.NgayCapNhat >= ALL(SELECT ttd2.NgayCapNhat FROM CT_TTDH ttd2 WHERE ttd2.MaDH = ttd1.MaDH)) SET @mota = (select Mota from TinhTrangDH where @ttdh = MaTinhTrang) print(@mota)</pre> <p>⇒ Tình trạng đơn hàng = 3</p>	
	Vì tran 1 giữ xlock nên tran 2 sẽ chờ tran 1 bỏ lock để tiếp tục bước 1
<p><i>B2: Kiểm tra đơn hàng có đang được chờ giao hàng hay không</i></p> <pre>IF @ttdh < 3</pre>	



```

BEGIN
    PRINT (N'Dơn hàng chưa sẵn
sàng để giao')
    RAISERROR ('1',15,1)
END
ELSE IF @ttdh = 3
BEGIN
    DECLARE @PhiVanChuyen int
    SET @PhiVanChuyen = (SELECT
PhiVanChuyen FROM DonHang WHERE MaDH =
@MaDH)
    INSERT INTO ThuNhapTX VALUES
(@MaTX, @MaDH, @PhiVanChuyen)

    INSERT INTO CT_TTDH VALUES
(GETDATE(), @MaDH, 4)
END
ELSE
begin
    print (N'Dơn hàng đã
có người nhận')
    raiserror ('2',15,1);
end

```

**Thỏa điều kiện tình trạng lần cập nhật mới nhất của
đơn hàng = 3. Thực hiện insert**

B1: Kiểm tra tình trạng đơn hàng

```

DECLARE @ttdh int, @mota nvarchar(100)
Set @ttdh = (SELECT MaTT FROM
CT_TTDH ttd1 with (XLOCK, ROWLOCK)
WHERE ttd1.MaDH = @MaDH
AND ttd1.NgayCapNhat >=
ALL(SELECT ttd2.NgayCapNhat
FROM CT_TTDH ttd2 WHERE ttd2.MaDH =
ttd1.MaDH))
SET @mota = (select Mota
from TinhTrangDH where @ttdh =
MaTinhTrang)
print(@mota)

```

Tình trạng đơn hàng = 4

*B2: Kiểm tra đơn hàng có đang được chờ giao hàng
hay không*

```

IF @ttdh < 3
BEGIN
    PRINT (N'Dơn hàng chưa sẵn
sàng để giao')
    RAISERROR ('1',15,1)

```

	<pre> END ELSE IF @ttdh = 3 BEGIN DECLARE @PhiVanChuyen int SET @PhiVanChuyen = (SELECT PhiVanChuyen FROM DonHang WHERE MaDH = @MaDH) INSERT INTO ThuNhapTX VALUES (@MaTX, @MaDH, @PhiVanChuyen) INSERT INTO CT_TTDH VALUES (GETDATE (), @MaDH, 4) END ELSE begin print (N'Dơn hàng đã có người nhận') raiserror ('2',15,1) end ⇒ Không thỏa điều kiện =3 vì tình trạng =4 nên thông báo đã có người nhận → raiserror </pre>
COMMIT TRAN	
	ROLLBACK TRANSACTION

⇒ Để xử lý tranh chấp thực hiện:

Thêm XLOCK, ROWLOCK để ngăn tình trạng tran 2 đọc sai.

`SELECT MaTT FROM CT_TTDH ttd1 with (XLOCK, ROWLOCK)`

Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two separate SQL Server query windows side-by-side. Both windows have the same code:

```

Tran_1_TestCase02...phvy01@gmail.com  Tran_2_TestCase02...phvy01@gmail.com
Use HT_DHCH_ONLINE
GO
declare @msg nvarchar(100)
EXEC sp_TaiXeNhanDonHang '012456781', '0000000001',@msg=@msg output
SELECT @msg

```

Window 1 (Left):

- Results tab: MoTa: Dang chờ giao hàng.
- TTDH tab: 1 | 3
- Output tab: (No column name) | Nhận đơn thành công.

Window 2 (Right):

- Results tab: MoTa: Đơn hàng đang giao.
- TTDH tab: 1 | 4
- Output tab: (No column name) | Đơn hàng đã có người nhận.



Demo trên giao diện Cả hai tài xế cùng nhận đơn hàng

Vyngalam

Đơn hàng Thông tin cá nhân

Đăng ký vận chuyển

#	Mã Đơn hàng	Mã khách hàng	Địa chỉ	Phường	Quận	Tỉnh	Tên người nhận	Số điện thoại	Ngày lập	Phi vận chuyển	Tổng tiền	Thanh toán	
1	0000000001	0930123450								75000	1575000		+

TRANG TRƯỚC

TRANG KẾ

0/0.01

Nhưng chỉ một tài xế nhận được, tài xế 2 bị báo lỗi

Thông tin Tài Xe

Mã Tài Xe: 012317983262

Họ và tên: Huỳnh Bá Vy

Số điện thoại: 012456781

Danh sách thu nhập

Cập nhật đơn hàng

Mã Đơn hàng Mã Tình Trạng

<#> Mã Đơn hàng

Phi Vận chuyển

1 0000000001 75000

Thông tin Tài Xe

Mã Tài Xe: 047733459124

Họ và tên: Tô Huy Thành

Số điện thoại: 012456782

Danh sách thu nhập

Cập nhật đơn hàng

Mã Đơn hàng Mã Tình Trạng

<#> Mã Đơn hàng

Phi Vận chuyển

TRANG TRƯỚC

TRANG KẾ

0/0

2.3. Test Case 3:

Tình huống: Hai đối tác cùng thêm sản phẩm mới vào bảng SanPham, mã sản phẩm phải được tăng lên theo đúng thứ tự thêm vào. Nhưng do cả hai transaction T1 và T2 cùng đọc một lúc dẫn đến tình trạng MaSP tăng lên không theo đúng thứ tự nên T2 sẽ không thêm được do bị trùng mã.

Tình huống tương tự có thể xảy ra khi tạo thêm đối tác, thêm đơn hàng, thêm chi nhánh và thêm hợp đồng vào hệ thống.

T1: Đối tác 1 thêm sản phẩm mới T2: Đối tác 2 thêm sản phẩm mới	
T1	T2
sp_ThemSanPham N'Nước rửa chén',500,6	sp_ThemSanPham N'Xà bông tắm',500,6
BEGIN TRAN	BEGIN TRAN
<i>B1: Đọc mã sản phẩm lớn nhất từ bảng SanPham</i> declare @ma_count bigint, @MaSP varchar(6) set @MaSP = (select TOP 1 (MaSP) from SanPham with (XLOCK, ROWLOCK) order by MaSP DESC) $\Rightarrow @MaSP=000001$	
<i>B2: Kiểm tra bảng SanPham trước đó đã có sản phẩm nào chưa, nếu có thì tăng MaSP mới lên 1 đơn vị.</i> if (isnull(@MaSP,'false') <> 'false') begin set @ma_count = cast (@MaSP as bigint) + 1 end else begin set @ma_count = 1 end set @MaSP = RIGHT ('00000' + CAST (@ma_count AS VARCHAR (6)), 6) $\Rightarrow @MaSP=000002$	
	<i>Vì tran 1 giữ XLOCK để insert nên tran 2 phải chờ tran 1 commit</i>
<i>B3: Thêm sản phẩm vào Table</i> INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) $\Rightarrow Insert thành công vào bảng SanPham$ $(@MaSP=000002).$	



COMMIT TRAN	
	<p>B1: <i>Đọc mã sản phẩm lớn nhất từ bảng SanPham</i></p> <pre>declare @ma_count bigint, @MaSP varchar(6) set @MaSP = (select TOP 1 (MaSP) from SanPham with (XLOCK, ROWLOCK) order by MaSP DESC) ⇒ @MaSP=000002</pre>
	<p>B2: <i>Kiểm tra bảng SanPham trước đó đã có sản phẩm nào chưa, nếu có thì tăng MaSP mới lên 1 đơn vị.</i></p> <pre>if (isnull(@MaSP, 'false') <> 'false') begin set @ma_count = cast (@MaSP as bigint) + 1 end else begin set @ma_count = 1 end set @MaSP = RIGHT ('00000' + CAST (@ma_count AS VARCHAR (6)), 6) ⇒ @MaSP=000003.</pre>
	<p>B3: <i>Thêm sản phẩm vào Table</i></p> <pre>INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) ⇒ Insert thành công vào bảng SanPham (@MaSP=000003).</pre>
COMMIT TRAN	

⇒ **Để xử lý tranh chấp thực hiện:**

Khi đọc sản phẩm có mã sản phẩm lớn nhất để tiến hành đặt sản phẩm, đặt rowlock ngay trên dòng sản phẩm đó.

`select TOP 1 (MaSP) from SanPham with (XLOCK, ROWLOCK) order by MaSP DESC`
XLOCK kết hợp với ROWLOCK để ngăn chặn một session khác đọc trên dòng cho đến hết transaction.



Hình ảnh demo:

Demo trên SQL Server

```

Tran_1_TestCase03...phy01@gmail.com) USE HT_DHCH_ONLINE GO
--TestCase03
--Transaction 1
EXEC sp_ThemSanPham N'Nước rửa chén',500,6
Completion time: 2021-12-18T21:35:23.0818932+07:00

Tran_2_TestCase03...phy01@gmail.com) USE HT_DHCH_ONLINE GO
--TestCase03
--Transaction 2
EXEC sp_ThemSanPham N'Xà bông tắm',500,6
Completion time: 2021-12-18T21:35:29.1023090+07:00
  
```

2 Demo trên Giao diện : Cả 2 đối tác cùng thêm sản phẩm

Vijayalam

Thêm Sản phẩm

Tên sản phẩm

Giá bán

Số lượng tồn

Lưu

Vijayalam

Thêm Sản phẩm

Tên sản phẩm

Giá bán

Số lượng tồn

Lưu

Hai đối tác thêm sản phẩm thành công

Danh Sách Sản phẩm Của Đối Tác

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1		000001	Nước hoa	50000	32
2		000002	Nước rửa chén	500	6
3		000003	Xà bông tắm	500	6



2.4. Test Case 4:

Tình huống: Đối tác đang thêm sản phẩm mới, cùng lúc đó người dùng khác đọc sản phẩm, nhưng sau đó sản phẩm do có số lượng tồn nhỏ hơn 0 nên không được chấp nhận thêm vào bảng dẫn đến Rollback. Khi đó, sản phẩm người dùng đọc được không tồn tại nữa, gây ra vấn đề DirtyRead.

T1: Thêm sản phẩm mới T2: Đọc danh sách sản phẩm T1: Rollback → T2 đọc sai thông tin	
T1	T2
sp_ThemSanPham N'Óp điện thoại', 200, -1	
BEGIN TRAN	
	BEGIN TRAN
<i>B1: Lấy ra MaSP cao nhất trong bảng SanPham để tăng thêm 1 đơn vị vào MaSP mới</i> declare @ma_count bigint, @MaSP varchar(6) set @MaSP=(select TOP 1 (MaSP) from SanPham order by MaSP DESC) ⇒ @MaSP=000003. if (isnull(@MaSP, 'false')<>'false') begin set @ma_count=cast (@MaSP as bigint)+1 end else begin set @ma_count=1 end set @MaSP = RIGHT('00000'+CAST(@ma_count AS VARCHAR(6)), 6) ⇒ @MaSP=000004.	
<i>B2: Thêm sản phẩm vào Table</i> INSERT INTO SanPham values (@MaSP, @TenSP, @GiaBan, @SLTon) ⇒ Insert thành công vào bảng SanPham (@MaSP=000004).	
	<i>Đọc bảng SanPham vì khi insert SQL Server đã lock trên dòng nên sẽ không đọc được nữa</i> select * from SanPham ⇒ Kết quả trả về KHÔNG có thông tin của (000004, Óp điện thoại, 200, -1) trong danh sách Sản Phẩm.

	COMMIT TRAN ⇒ Đã đọc xong và COMMIT.
<i>B3: Kiểm tra số lượng tồn <0</i> <pre>if (@SLTon <0) begin print('1') raiserror(N'không được nhỏ hơn 0',15,1) end ⇒ @SLTon=-1 nên ROLLBACK.</pre>	
ROLLBACK TRANSACTION	

⇒ Để xử lý tranh chấp thực hiện:

Loại bỏ nolock vì sql server đã có hỗ trợ lock trên dòng khi insert ngăn ngừa tình trạng dirty read.

Hình ảnh demo:

Demo trên SQL Server

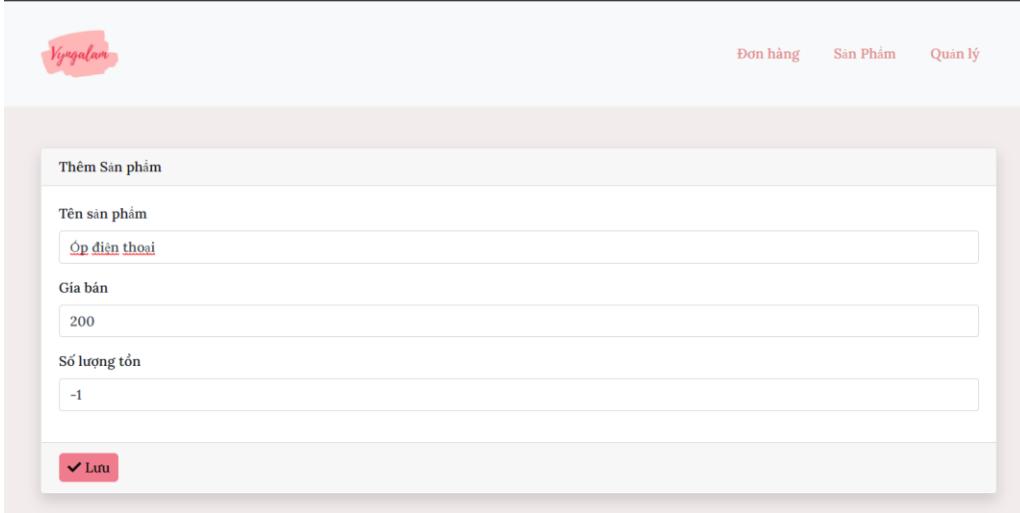
```
Tran_1_TestCase04...phvy01@gmail.com  USE HT_DHCH_ONLINE GO
--TestCase04
--Transaction 1
exec sp_ThemSanPham N'Ốp điện thoại',200,-1

108 %  Messages
000004
(1 row affected)
1
loi
Completion time: 2021-12-18T21:49:03.0714809+07:00
```

```
Tran_2_TestCase04...phvy01@gmail.com  USE HT_DHCH_ONLINE GO
--TestCase04
--Transaction 2
begin tran
waitfor delay '00:00:09'
select * from SanPham
COMMIT
```

MãSP	TênSP	GiaBan	SLTon
000001	Áo thun Mickey	50000	32
000002	Áo thun Minnie	45000	4
000003	Áo thun Donald	46000	7

Demo trên Giao diện: Đối tác thêm sản phẩm nhưng chưa Commit





Demo trên Giao diện: Cùng lúc đó thì khách hàng lại đọc nhưng kết quả vẫn đúng

Danh Sách Sản phẩm

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000001	Áo thun Mickey	50000	32	+ 眼界
2		000002	Áo thun Minnie	45000	4	+ 眼界
3		000003	Áo thun Donald	46000	7	+ 眼界

2.5. Test Case 5:

Tình huống: Tài xế cập nhật tình trạng đơn hàng và ngay sau đó khách hàng truy cập và xem tình trạng đơn hàng nhưng quá trình cập nhật của tài xế xảy ra lỗi nên phải rollback điều này khiến cho dữ liệu tình trạng đơn hàng mà khách hàng xem được trở thành dữ liệu rác.

T1: Tài xế cập nhật tình trạng đơn hàng. T2: Khách hàng xem tình trạng đơn hàng.	
T1	T2
exec sp_CapNhatTinhTrangDonHang '0000000001', 4;	
BEGIN TRAN	
	BEGIN TRAN
-- Tài xế thêm một dòng vào bảng tình trạng đơn hàng với MaDH là 'DH0000' và mã tình trạng là 4 (Đơn hàng đang giao) insert into CT_TTDH values (GETDATE (), @MaDH, @MaTT)	
	-- Khách hàng xem tình trạng của đơn hàng có MaDH là 'DH0000', tình trạng đơn hàng mới nhất sẽ là dòng dữ liệu có ngày cập nhật mới nhất. SELECT * FROM CT_TTDH WHERE MaDH= '0000000001' ⇒ @MaTT mới nhất = 3
-- Vì một lỗi nào đó khiến cho quá trình thêm tình trạng đơn hàng của tài xế không thành công nên T1 rollback. ROLLBACK TRANSACTION	

COMMIT TRAN

⇒ Để xử lý tranh chấp thực hiện:

Sau khi xử lý tran 2 không còn ở tình trạng read uncommitted nữa. Xuất ra được bộ dữ liệu đúng.

Hình ảnh demo:

Demo trên SQL Server

```

Tran_1_TestCase05...phvy01@gmail.com)  USE HT_DHCH_ONLINE
                                         Go
                                         exec sp_CapNhatTinhTrangDonHang '0000000001', 4;
                                         select * from CT_TTDH where MaDH = '0000000001'
                                         select * from TinhTrangDH
                                         108 %

Tran_2_TestCase05...phvy01@gmail.com)  USE HT_DHCH_ONLINE
                                         Go
                                         begin tran
                                         select * from CT_TTDH where MaDH= '0000000001'
                                         commit tran
                                         108 %
    
```

Tran_1_TestCase05...phvy01@gmail.com

```

USE HT_DHCH_ONLINE
Go
exec sp_CapNhatTinhTrangDonHang '0000000001', 4;
select * from CT_TTDH where MaDH = '0000000001'
select * from TinhTrangDH
    
```

Tran_2_TestCase05...phvy01@gmail.com

```

USE HT_DHCH_ONLINE
Go
begin tran
select * from CT_TTDH where MaDH= '0000000001'
commit tran
    
```

Results

NgayCapNhat	MaDH	MaTT
2021-11-11 00:00:00.000	0000000001	1
2021-11-13 00:00:00.000	0000000001	2
2021-11-14 00:00:00.000	0000000001	3

MaTinhTrang	Mota
0	Đơn hàng đã hủy.
1	Đơn hàng được tiếp nhận.
2	Đơn hàng đang đóng gói.
3	Đang chờ giao hàng.
4	Đơn hàng đang giao.
5	Đơn hàng giao thành công.

Demo trên Giao diện: tài xế cập nhật đơn hàng

Danh sách thu nhập

Cập nhật đơn hàng

0000000001	4	Cập nhật
------------	---	----------

#	Mã Đơn hàng	Phí Vận chuyển

TRANG TRƯỚC **TRANG KẾ**



Khách hàng đọc tình trạng đơn hàng vẫn đúng



Mã khách hàng: 0909123450
Tên người nhận: Trần Văn A
Số điện thoại: 0909123450

Địa chỉ:
Phường:
Quận:
Tỉnh:

Đơn hàng
Mã đơn hàng: 0000000001
Ngày lập: 10 Nov 2021 00:00:00:000

#	Mã sản phẩm	Tên sản phẩm	Giá Bán	Số Lượng	Thành Tiền
					Tổng tiền hàng
					Phi vận chuyển
					Tổng số tiền
					Phương thức thanh toán

Thanks for choosing us.

Tình trạng đơn hàng

Ngày	Mã tình trạng
11 Nov 2021 00:00:00:000	1
13 Nov 2021 00:00:00:000	2
14 Nov 2021 00:00:00:000	3

Mã tình trạng	Tình trạng
0	Đơn hàng đã hủy
1	Đơn hàng được tiếp nhận
2	Đơn hàng đang đóng gói

2.6. Test Case 6:

Tình huống: Khách hàng muốn xem các sản phẩm có giá dưới 100000, sau đó đối tác thực hiện thêm một sản phẩm có giá dưới 100000 nên khi khách hàng xem lại các sản phẩm có giá dưới 100000 sẽ nhận được số bộ dữ liệu khác nhau giữa hai lần đọc.

T1: Khách hàng xem các sản phẩm có giá dưới 100000. T2: Đối tác thêm một sản phẩm mới tên 'Bút bi' có giá là 7000.	
T1	T2
declare @sum int exec sp_KiemTraGiaBan 100000, @Tong = @sum output; select @sum as TongSoSanPham	exec sp_ThemSanPham N'Bút bi', 7000, 80
BEGIN TRAN	
-- Khách hàng xem các sản phẩm có giá dưới 100000.	

<pre>select @Tong = count (MaSP) from SanPham WITH (TABLOCK, HOLDLOCK) where SanPham.GiaBan < @gb ⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 3.</pre>	
<pre>-- Khách hàng xem lại các sản phẩm có giá dưới 100000. SELECT ROW_NUMBER () OVER (ORDER BY MaSP) AS ROWNUMBER, MaSP, TenSP, GiaBan, SLTon FROM SanPham WHERE GiaBan < @gb ⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 3.</pre>	<pre>-- Vì tran1 giữ table lock nên tran2 phải đợi đến khi tran 1 bỏ lock mới có thể thêm sản phẩm</pre>
<pre>COMMIT TRAN</pre>	
	<pre>-- Đổi tác thực hiện việc thêm một sản phẩm có tên là 'Bút bi', giá bán là 7000 và số lượng tồn là 80. exec sp_ThemSanPham N'Bút bi', 7000, 80</pre>

⇒ **Để xử lý tranh chấp thực hiện:**

Khi đọc các sản phẩm có giá dưới 100000 thêm table lock để ngăn chặn phantom read.

```
select @Tong = count (MaSP) from SanPham WITH (TABLOCK, HOLDLOCK) where
SanPham.GiaBan < @gb
```

Hình ảnh demo:

Demo trên SQL Server

The screenshot shows two SSMS windows side-by-side. Both windows have the same connection string: Tran_1_TestCase06...phy01@gmail.com. The left window (Session 1) contains the following code:

```
Tran_1_TestCase06...phy01@gmail.com) ➔ X
Use HT_DHCH_ONLINE
go

--TestCase06
--Transaction 1
declare @sum int
exec sp_KiemTraGiaBan 100000, @Tong = @sum output
select @sum as NumberOfColumn
```

The right window (Session 2) contains the following code:

```
Tran_2_TestCase06...phy01@gmail.com) ➔ X
Use HT_DHCH_ONLINE
go

--TestCase06
--Transaction 2
exec sp_ThemSanPham N'Bút bi', 7000, 80
```

The results pane for Session 1 shows a table with three rows:

RowNumber	MaSP	TenSP	GiaBan	SLTon
1	000002	Áo thun	89000	450
2	000003	Bông cài	20000	28
3	000005	Dép	65000	1100

The results pane for Session 2 shows the following output:

```
000007
(1 row affected)
Completion time: 2021-12-18T22:38:58.6565841+07:00
```



Đối tác 2 thêm sản phẩm trong lúc đối tác 1 đang đọc sản phẩm

Vngalan

Đơn hàng Sản Phẩm Quản lý

Thêm Sản phẩm

Tên sản phẩm

Giá bán

Số lượng tồn

Lưu

Đối tác 1 vẫn đọc đúng số lượng sản phẩm cần đọc

Kiểm tra Sản phẩm

Tổng Sản Phẩm: 3

#	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1	000002	Áo thun	89000	450
2	000003	Bông cài	20000	28
3	000005	Dép	65000	1100

2.7. Test Case 7:

Tình huống: Đối tác xem sản phẩm có số lượng tồn < 10 để nhập thêm hàng, tuy nhiên người dùng lại thêm 1 sản phẩm vào đơn đặt hàng khiến cho số lượng sản phẩm có slt < 10 tăng lên. Dẫn đến đối tác đọc được 2 bộ dữ liệu khác nhau.

T1: Đối tác đọc số lượng tồn T2: Khách hàng đặt hàng *Cho số lượng tồn của sản phẩm '0000001' là 32	
T1 <pre>declare @sum int exec sp_KiemTraSLTon 10, @Tong = @sum output; select @sum as NumberOfProduct</pre>	T2 <pre>DECLARE @error int exec sp_ThemChiTietDonHang '0000000001', '000001', 25, @error = @error output</pre>

BEGIN TRAN	
	BEGIN TRAN
--Đối tác xem các sản phẩm có số lượng tồn dưới 10 <pre>select @Tong = count (MaSP) from SanPham WITH (TABLOCK, HOLDLOCK) where SanPham.SLTon < @slt waitfor delay '00:00:12'</pre> <p>⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 2.</p>	
--Đối tác in ra các sản phẩm có số lượng tồn dưới 10 <pre>SELECT ROW_NUMBER () OVER (ORDER BY MaSP) AS ROWNUMBER, MaSP, TenSP, GiaBan, SLTon FROM SanPham WHERE SLTon < @slt</pre> <p>⇒ Số bộ dữ liệu của bảng SanPham thỏa điều kiện là 2.</p>	
	-- Vì tran 1 giữ lock nên tran 2 phải chờ tran 1 bỏ lock mới có thể insert vào CT_DonHang
COMMIT TRAN	
	-- Khách hàng thực hiện thêm sản phẩm '000001' vào đơn hàng '0000000001' với số lượng là 25 <pre>INSERT INTO CT_DonHang(MaDH, MaSP, SoLuong) VALUES(@MaDH, @MaSP, @SoLuong)</pre>
	COMMIT TRAN

⇒ Để xử lý tranh chấp thực hiện:

Khi đọc số lượng tồn các sản phẩm để xuất bảng thêm table lock để ngăn chặn phantom read.

```
select @Tong = count (MaSP) from SanPham WITH (TABLOCK, HOLDLOCK) where
SanPham.SLTon < @slt
```

Hình ảnh demo:

```

Tran_1_TestCase07...phy01@gmail.com)  Trn_2_TestCase07...phy01@gmail.com)
use HT_DHCH_ONLINE
GO

declare @sum int
exec sp_KiemTraSLTon 10, @Tong = @sum output;
select @sum as NumberOfProduct

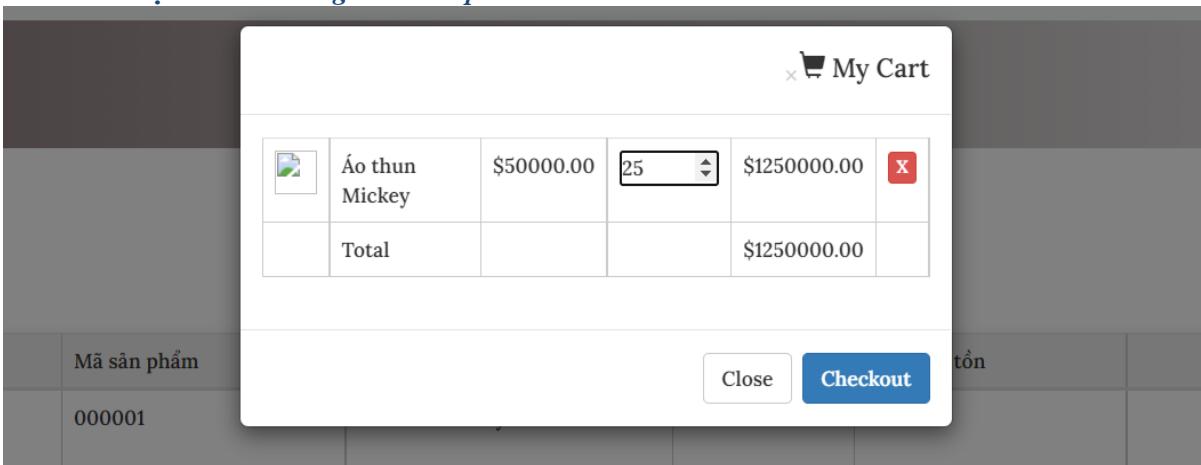
-- Tran_2
DECLARE @error int
exec sp_ThemChiTietDonHang '0000000001', '000001', 25, @error = @error output
GO

```

ROWNUMBER	MaSP	TenSP	GiaBan	SLTon
1	000002	Áo thun Minnie	45000	4
2	000003	Áo thun Donald	46000	7

	NumberOfProduct
1	2

Demo trên Giao diện: Khách hàng mua sản phẩm



Trước đó đối tác đang kiểm tra số lượng tồn <10 thì tổng số với số sản phẩm in ra giống nhau

Kiểm tra Sản phẩm				
10	Xem			
Gía bán	Xem			
Tổng Sản Phẩm: 2				
#	Mã sản phẩm	Tên sản phẩm	Gía bán	Số lượng tồn
1	000002	Áo thun Minnie	45000	4
2	000003	Áo thun Donald	46000	7



2.8. Test Case 8:

Tình huống: Người dùng tìm sản phẩm có mã là ‘000001’ và tên là ‘Nước hoa’. Hệ thống xác nhận có tồn tại sản phẩm, nhưng sau đó đổi tên lại cập nhật tên sản phẩm thành ‘Bông cài’. Khiến cho lần đọc tiếp theo của người dùng không đọc được thông tin sản phẩm dù sản phẩm có tồn tại.

T1: Kiểm tra thông tin sản phẩm với @MaSP và @TenSP

T2: Cập nhật sản phẩm với tên mới

T1: Xem thông tin sản phẩm với @MaSP và @TenSP ban đầu → không hiện thông tin.

T1	T2
<pre>exec sp_KiemTraSP '000001', N'Nước hoa', @result = @result output</pre>	<pre>EXEC sp_CapNhatSanPham '000001', N'Áo hai dây', 300, 3</pre>
BEGIN TRAN	
	BEGIN TRAN
<p><i>B1: Kiểm tra sản phẩm có mã ‘000001’ và tên là ‘Nước hoa’ có tồn tại không</i></p> <pre>if not exists (select * from SanPham with (XLOCK, ROWLOCK) where @MaSP=MaSP and @TenSP= TenSP) raiserror (N'Không tồn tại sản phẩm',15,1) set @result=1 waitfor delay '00:00:08' ⇒ @MaSP và @TenSP có tồn tại, @result=1</pre>	
<p><i>B2: Đọc thông tin của sản phẩm có mã ‘000001’ và tên là ‘Nước hoa’</i></p> <pre>select * from SanPham where MaSP= @MaSP and TenSP= @TenSP ⇒ In ra được sản phẩm có MaSP=000001 và TenSP= Nước hoa</pre>	<p><i>Vì tran 1 giữ XLOCK nên tran 2 chờ tran 1 commit mới có thể thực hiện bước 1</i></p>
COMMIT TRAN	
	<p><i>Kiểm tra sản phẩm có tồn tại không, sau đó cập nhật lại tên sản phẩm</i></p> <pre>if not exists (select * from SanPham with (XLOCK, ROWLOCK) where @MaSP= MaSP) begin print ('1') raiserror (N'Không tồn tại sản phẩm',15,1) end waitfor delay '00:00:05'</pre>

	<pre>update SanPham set TenSP= @TenSP, GiaBan= @GiaBan, SLTon= @SLTon where MaSP= @MaSP</pre> <p>⇒ Sản phẩm '000001' được cập nhật tên thành công từ 'Nước hoa' sang 'Áo hai dây'</p>
	COMMIT TRAN

⇒ Để xử lý tranh chấp thực hiện:

Khi đọc sản phẩm thêm XLOCK, ROWLOCK để tránh tình trạng phantom read.

```
select * from SanPham with (XLOCK, ROWLOCK) where @MaSP=MaSP and @TenSP= TenSP
```

Hình ảnh demo:

Demo trên SQL Server

Tran_1_TestCase08...phvy01@gmail.com) USE HT_DHCH_ONLINE GO
--TestCase08
--Transaction 1
declare @result int
exec sp_KiemTraSP '000001',N'Nước hoa',@result = @result output

Tran_2_TestCase08...phvy01@gmail.com)* USE HT_DHCH_ONLINE GO
--TestCase08
--Transaction 2
EXEC sp_CapNhatSanPham '000001',N'Áo hai dây',300,3
SELECT * from SanPham

	MaSP	TenSP	GiaBan	SLTon
1	000001	Nước hoa	50000	32

	MaSP	TenSP	GiaBan	SLTon
1	000001	Áo hai dây	300	3
2	000002	Áo thun Minnie	45000	4
3	000003	Áo thun Donald	46000	7

Demo trên Giao diện: Người dùng kiểm tra sản phẩm nhưng đổi tác lại cập nhật sản phẩm đó thành tensp khác
nên lúc người dùng in sản phẩm với masp và tensp cũ thì vẫn in được đúng.

Kiểm tra Sản phẩm

#	Mã sản phẩm	Tên sản phẩm	Gía bán	Số lượng tồn
	000001	Nước hoa	50000	32

Cập nhật Sản phẩm

Mã sản phẩm

Tên sản phẩm

Gía bán

Số lượng tồn

Lưu

2.9. Test Case 9:

Tình huống: Giả sử T1 và T2 thay phiên nhau update 2 sản phẩm, lúc này sẽ xảy ra tình trạng circle deadlock do cả hai phải chờ nhau.

T1	T2
BEGIN TRAN	
	BEGIN TRAN
<pre>UPDATE SanPham set SLTon = 100 WITH (TABLOCK,HOLDLOCK) where MaSP = '000001'; (1)</pre> <p>--Thành công. Lúc này table SanPham đang lock vừa update của (1)</p>	
	--Tran 1 đang giữ table lock nên tran 2 sẽ chờ cho đến khi tran 1 commit
<pre>UPDATE SanPham set SLTon = 100 where MaSP = '000002'; (3)</pre> <p>--Thành công. Update sản phẩm 000002</p>	
COMMIT TRAN	
	<pre>UPDATE SanPham set SLTon = 200 where MaSP = '000002'; (2)</pre> <p>--Thành công. Lúc này table SanPham đang lock row vừa update của (2)</p>
	<pre>UPDATE SanPham set SLTon = 200 where MaSP = '000001'; (4)</pre> <p>-- Thành công. Update sản phẩm 000001</p>
	COMMIT TRAN

⇒ **Để xử lý tranh chấp thực hiện:**

Thực hiện table lock để ngăn không cho xảy ra tình trạng deadlock.

Hình ảnh demo:

The screenshot shows two parallel transactions running in separate windows of SQL Server Management Studio.

Tran_1_TestCase09...E (NGA\Admin (54))

```
1 USE HT_DHCH_ONLINE
2 GO
3 BEGIN TRAN
4 UPDATE SanPham WITH (TABLOCK,HOLDLOCK) set SLTon = 100 where MaSP = '000001';
5 WAITFOR DELAY '00:00:10'
6 UPDATE SanPham set SLTon = 100 where MaSP = '000002';
7 COMMIT TRAN
```

Tran_2_TestCase09...E (NGA\Admin (58))

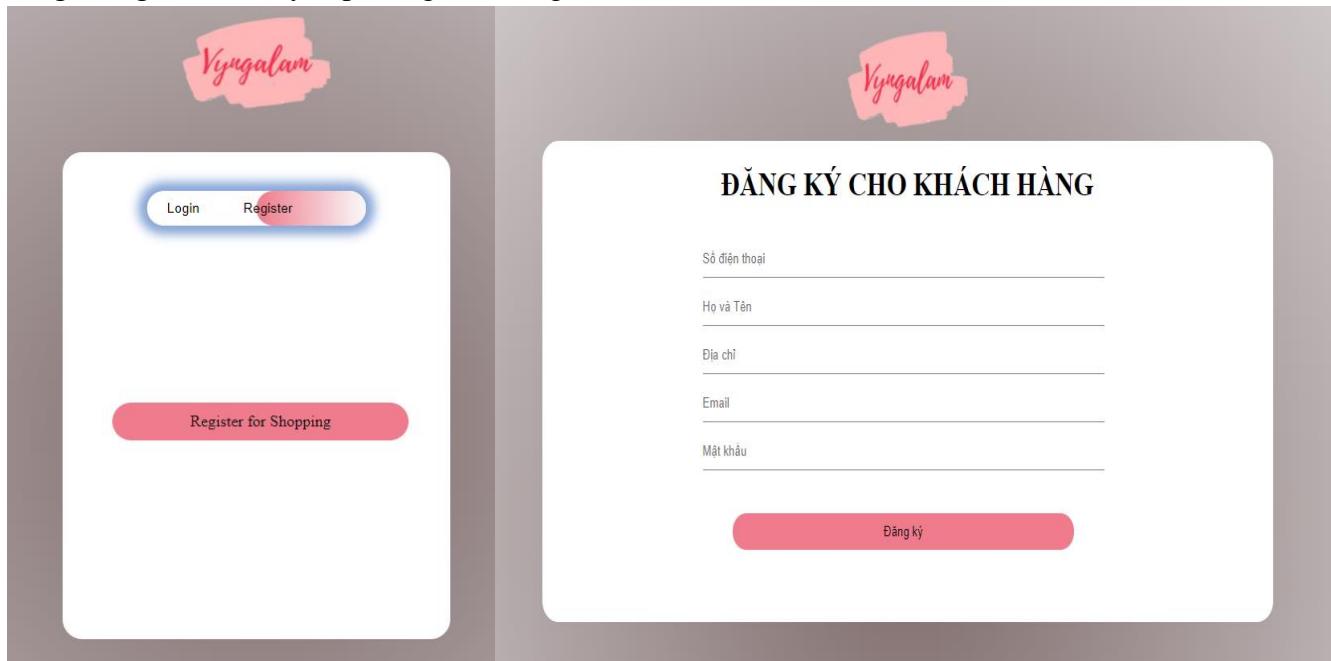
```
1 USE HT_DHCH_ONLINE
2 GO
3 BEGIN TRAN
4 UPDATE SanPham set SLTon = 200 where MaSP = '000002';
5 WAITFOR DELAY '00:00:10'
6 UPDATE SanPham set SLTon = 200 where MaSP = '000001';
7 COMMIT TRAN
8
```

Both transactions report 1 row affected in their respective messages panes. The completion times are also identical, indicating they were executed simultaneously.

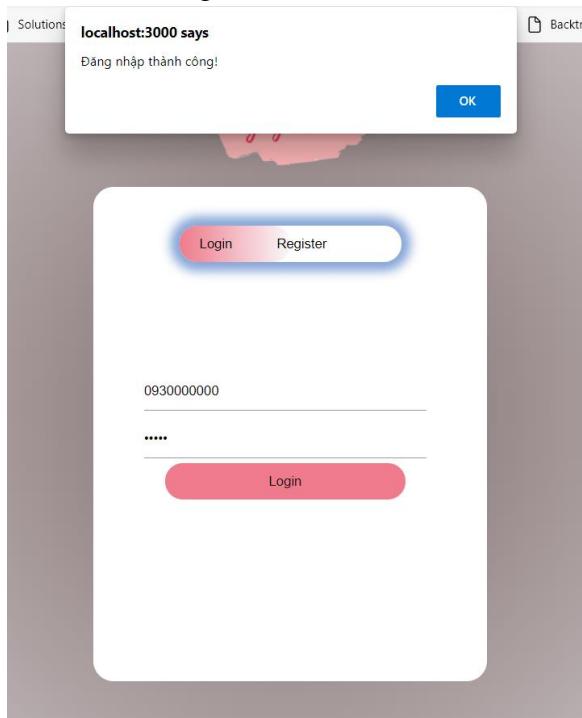
III. Thiết kế giao diện chức năng

1. Khách hàng

Khách hàng chưa có tài khoản sẽ đăng ký tài khoản và thông tin chi tiết để sử dụng. Sau khi đăng ký thành công, trang web sẽ truy cập thẳng vào trang chủ với tài khoản được tạo.



Nếu khách hàng đã có tài khoản thì thực hiện Login để vào trang chủ.





Trang chủ

Vyngalam

Trang chủ Sản phẩm Đơn hàng Tim kiếm

Tim Kiếm Sản Phẩm

BlackFriday

SALE



Đặc trưng

Tại đây bạn có thể xem sản phẩm mới của chúng tôi với giá hợp lý tại Vyngalam



Bút lông màu
100.000đ



Nhẫn nữ
200.000đ



Đầm nữ dài
300.000đ



Giày thể thao nam
1.300.000đ



Chọn tab Sản phẩm để xem danh sách sản phẩm

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000000	Giày cao gót	331000	40	+ <input type="button" value=""/> <input type="button" value="-"/>
2		000001	Áo	480000	22	+ <input type="button" value=""/> <input type="button" value="-"/>
3		000002	Túi ví	271000	43	+ <input type="button" value=""/> <input type="button" value="-"/>
4		000003	Kính mắt	142000	39	+ <input type="button" value=""/> <input type="button" value="-"/>
5		000004	Đầm/Váy	179000	24	+ <input type="button" value=""/> <input type="button" value="-"/>

Khi khách hàng chọn sản phẩm (biểu tượng +), sản phẩm sẽ được đưa vào giỏ hàng theo số lượng đã chọn.

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	
1		000000	Giày cao gót	331000	40	+ <input type="button" value=""/> <input type="button" value="-"/>
2		000001	Áo	480000	22	+ <input type="button" value=""/> <input type="button" value="-"/>



Nhập đầy đủ thông tin để đặt hàng

localhost:3000 says

Total Price: 893000
Total Quantity: 3

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	Đặt hàng
1		000000	Giày cao gót	331000	18	+ ⏪
2		000001	Áo	480000	2	+ ⏪
3		000002	Túi vi	271000	31	+ ⏪
4		000003	Kính mắt	142000	38	+ ⏪

Khi đặt hàng thành công trang web sẽ hiển thị thông báo đã đặt hàng thành công.

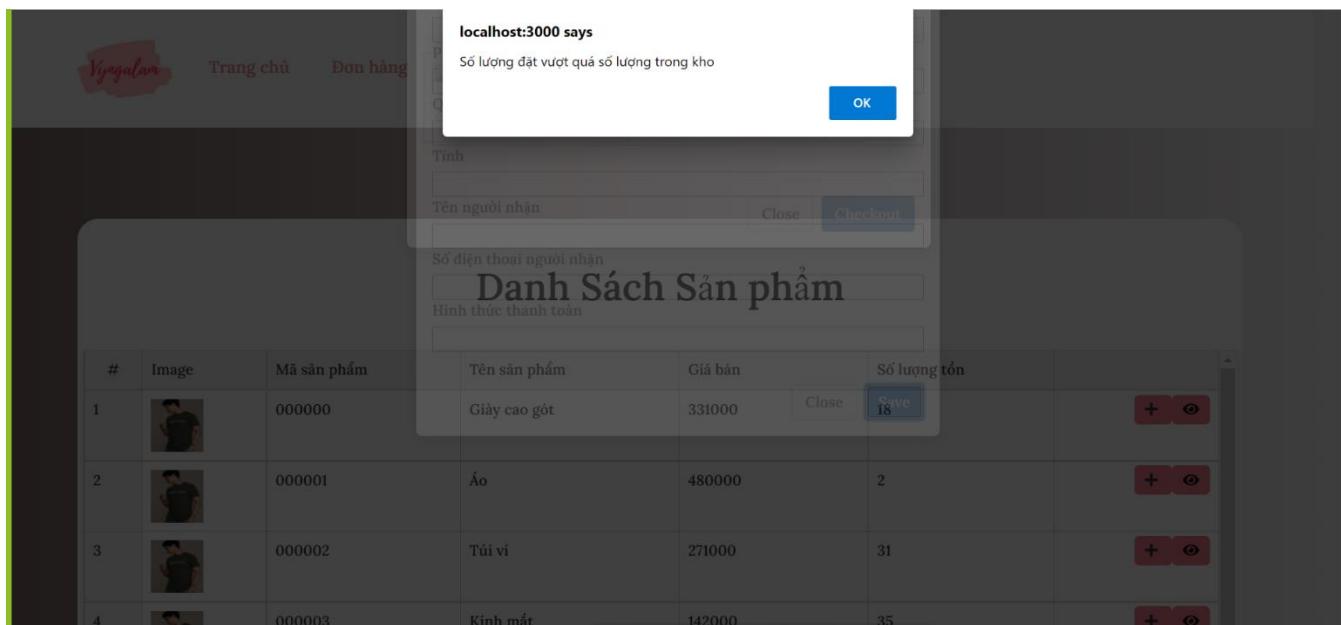
localhost:3000 says

Đặt hàng thành công.

Danh Sách Sản phẩm

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn	Đặt hàng
1		000000	Giày cao gót	331000	18	+ ⏪
2		000001	Áo	480000	2	+ ⏪
3		000002	Túi vi	271000	31	+ ⏪
4		000003	Kính mắt	142000	38	+ ⏪

Khi đặt hàng xảy ra lỗi thì hệ thống sẽ thông báo lỗi ra ví dụ như việc người dùng đặt hàng sản phẩm có số lượng vượt quá số lượng sản phẩm trong kho.



Khi người dùng chọn vào Đơn hàng sẽ xem được thông tin cá nhân của khách hàng và danh sách các đơn hàng mà khách hàng đã đặt.

The screenshot shows two main sections. The top section displays customer information: Ma khach hang: 0930000000, Họ và tên: Đoàn Thanh Bảo, Địa chỉ: 15 Vườn Lài, Phường 17, Quận 4, TP.HCM, Email: BarneyRosado@nowhere.com. The bottom section displays a table of orders:

#	Mã Đơn hàng	Mã khách hàng	Ngày lập	Phí Vận chuyển	Tổng hàng	Tổng tiền	X
	0000000000	0930000000	01 Jan 2020 00:00:01:000				

Navigation buttons at the bottom left say 'TRANG TRƯỚC' (Previous Page) and 'TRANG KẾ' (Next Page), and a page number '0/01'.

Khi bấm vào kí hiệu xem để xem chi tiết đơn hàng và tình trạng đơn hàng.





Trang chủ Đơn hàng Sản Phẩm Tìm kiếm

Đơn hàng
Mã đơn hàng: 0000000000
Ngày lập: 01 Jan 2020 00:00:01:000

Mã khách hàng: 0930000000
Tên người nhận: Nguyễn Khắc Hoài Hào
Số điện thoại: 0521242672

Địa chỉ: 07 Hoa Hảo
Phường: Phường 3
Quận: Quận 8
Tỉnh: TP.HCM

#	Mã sản phẩm	Tên sản phẩm	Giá Bán	Số Lượng	Thành Tiền
					Tổng tiền hàng
					Phi vận chuyển
					Tổng số tiền
					Phương thức thanh toán
					Momo

Thanks for choosing us.

Tình trạng đơn hàng

Ngày	Mã tình trạng	Mã tình trạng	Tình trạng
04 Jan 2021 14:56:42:000	1	0	Đơn hàng đã hủy
27 Apr 2021 06:49:27:000	2	1	Đơn hàng được tiếp nhận
26 Jul 2021 21:51:55:000	3	2	Đơn hàng đang đóng gói
19 Dec 2021 13:34:02:077	4	3	Đang chờ giao hàng
		4	Đơn hàng đang giao
		5	Đơn hàng giao thành công

Khi bấm vào mục Tìm kiếm người dùng có thể tìm kiếm sản phẩm theo mã sản phẩm hoặc tên sản phẩm.



Vijayalakshmi

Trang chủ Đơn hàng Sản Phẩm Tìm kiếm

Kiểm tra Sản phẩm

000001	Áo	Xem
--------	----	-----

Tình trạng:

#	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
	000001	Áo	480000	22



2. Đối tác

Bảng danh sách đơn hàng hiển thị dành riêng cho đối tác để đối tác có thể cập nhật tình trạng đơn hàng lúc nhận đơn và đóng gói.

VnGalaxy
Đơn hàng
Sản Phẩm
Quản lý

#	Mã Đơn hàng	Mã khách hàng	Ngày lập	Phi Vận chuyển	Tổng hàng	Tổng tiền	
1	0000000000	0930000000	01 Jan 2020 00:00:01:000				
2	0000000001	0930123450		11550	231000	242550	
3	0000000002	0930000002	03 Jan 2020 00:00:01:730				
4	0000000003	0930000003	04 Jan 2020 00:00:01:830	33100	662000	695100	
5	0000000004	0930000004	05 Jan 2020 00:00:01:540				
6	0000000005	0930000005	06 Jan 2020 00:00:01:510	82750	1655000	1737750	
7	0000000006	0930000006	07 Jan 2020 00:00:01:860	284700	5694000	5978700	
8	0000000007	0930000007	08 Jan 2020 00:00:01:270				
9	0000000008	0930000008	09 Jan 2020 00:00:01:320				
10	0000000009	0930000009	10 Jan 2020 00:00:01:750				
11	0000000010	0930000010	11 Jan 2020 00:00:01:650	64800	1296000	1360800	
12	0000000011	0930000011	12 Jan 2020 00:00:01:810	161700	3234000	3395700	
13	0000000012	0930000012	13 Jan 2020 00:00:01:530	19200	384000	403200	

TRANG TRƯỚC
TRANG KẾ
0/0.2



Chọn biểu tượng xem chi tiết đơn hàng để hiển thị bảng chi tiết đơn hàng. Ở đây đối tác có thể xem được danh sách các món hàng cần đóng gói và cập nhật tình trạng đơn hàng.



Mã khách hàng: 0930000003
Tên người nhận: Đào Phước Trần Thúy
Số điện thoại: 0551421731

Địa chỉ: 873 Lê Văn Sỹ
Phường: Phường 6
Quận: Long Biên
Tỉnh: Hà Nội

Đơn hàng
Mã đơn hàng: 0000000003
Ngày lập: 04 Jan 2020 00:00:01:830

#	Mã sản phẩm	Tên sản phẩm	Giá Bản	Số Lượng	Thành Tiền
1	000000	Giày cao gót	331000	2	662000

Tổng tiền hàng	662000
Phi vận chuyển	33100
Tổng số tiền	695100
Phương thức thanh toán	ZaloPay

Thanks for choosing us.

Tình trạng đơn hàng

Ngày	Mã tình trạng
01 Jan 2020 00:08:49:000	3
12 Mar 2020 00:08:50:000	2
10 Dec 2021 23:59:59:000	0

Cập nhật tình trạng

Mã đơn hàng

Mã tình trạng

Mã tình trạng	Tình trạng
0	Đơn hàng đã hủy
1	Đơn hàng được tiếp nhận
2	Đơn hàng đang đóng gói
3	Đang chờ giao hàng
4	Đơn hàng đang giao
5	Đơn hàng giao thành công



Đối tác còn có thêm một bảng sản phẩm dành riêng cho đối tác. Trong bảng sản phẩm này, đối tác có thể thêm sản phẩm hoặc cập nhật sản phẩm.

Vygalam

Đơn hàng Sản Phẩm Quản lý

Danh Sách Sản phẩm Của Đối Tác

* Cập nhật sản phẩm + Thêm sản phẩm

#	Image	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1		000000	Giày cao gót	331000	40
2		000001	Áo	480000	22
3		000002	Túi ví	271000	43
4		000003	Kính mắt	142000	39
5		000004	Đầm/Váy	179000	24
6		000005	Quần short	159000	50
7		000006	Suit	462000	31
8		000007	Áo khoác	128000	24



Ở bảng sản phẩm của đối tác, chọn nút Cập nhật sản phẩm → đối tác sẽ được đưa đến trang Cập nhật sản phẩm. Ở đây đối tác có thể Thêm sản phẩm vào chi nhánh phân phối hoặc Cập nhật sản phẩm. Sau khi nhập đầy đủ thông tin, chọn Lưu để thực hiện Thêm hoặc Cập nhật.

Wagalam

Đơn hàng Sản Phẩm Quản lý

Thêm Sản phẩm vào Chi nhánh

Mã sản phẩm

Mã chi nhánh

Lưu

Cập nhật Sản phẩm

Mã sản phẩm

Tên sản phẩm

Giá bán

Số lượng tồn

Lưu



Ở bảng sản phẩm của đối tác, chọn nút Thêm sản phẩm → đối tác sẽ được đưa đến trang Thêm sản phẩm. Ở đây đối tác sẽ nhập Tên sản phẩm, Giá bán, Số lượng tồn. Sau khi nhập xong chọn Lưu để thêm sản phẩm.

Thêm Sản phẩm

Tên sản phẩm

Giá bán

Số lượng tồn

Lưu

Trên thanh Menu chọn tab Quản lý, đối tác sẽ được đưa đến trang kiểm tra Sản phẩm, đối tác có thể kiểm tra sản phẩm có số lượng tồn < n hoặc giá bán < n. Chọn nút Xem để hiển thị bảng danh sách như hai hình sau.

#	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1	000001	Áo	480000	22
2	000003	Kính mắt	142000	39
3	000004	Đầm/Váy	179000	24
4	000006	Suit	462000	31
5	000007	Áo khoác	128000	24
6	000008	Áo vest	162000	36
7	000009	Chân váy	420000	37
8	000010	Quần jeans	120000	13
9	000012	Đồ tập	113000	18
10	000015	Quần dài	263000	14
11	000016	Blazer	212000	19



Vyngalam

[Đơn hàng](#) [Sản Phẩm](#) [Quản lý](#)

Kiểm tra Sản phẩm

Số lượng tồn	Xem
420000	Xem

Tổng Sản Phẩm: 15

#	Mã sản phẩm	Tên sản phẩm	Giá bán	Số lượng tồn
1	000000	Giày cao gót	331000	40
2	000002	Túi ví	271000	43
3	000003	Kính mắt	142000	39
4	000004	Đầm/Váy	179000	24
5	000005	Quần short	159000	50
6	000007	Áo khoác	128000	24

3. Tài xế

Tài xế chưa có tài khoản sẽ đăng ký tài khoản và thông tin chi tiết để sử dụng. Sau khi đăng ký thành công, trang web sẽ truy cập thẳng vào trang thông tin tài xế với tài khoản được tạo.

Vyngalam

Đăng nhập
Đăng ký

Đăng ký để bắt đầu

Vyngalam

ĐĂNG KÝ CHO TÀI XẾ

CMND _____

Họ và Tên _____

Số điện thoại _____

Địa chỉ _____

Biển số xe _____

Khu vực hoạt động _____

Email _____

Số tài khoản _____

Ngân hàng _____

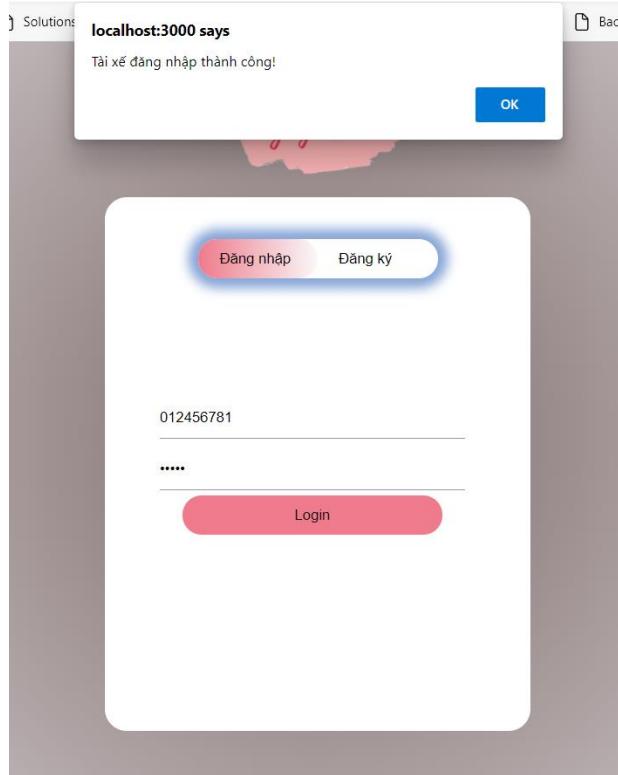
Chi nhánh ngân hàng _____

Mật khẩu _____

Đăng ký



Tài xế đã có tài khoản sẽ đăng nhập để vào trang web.



Sau khi đăng nhập hoặc đăng ký, trang thông tin tài xế sẽ hiển thị thông tin cá nhân của tài xế đó và danh sách các đơn hàng trong thu nhập tài xế. Dựa vào danh sách thu nhập để cập nhật tình trạng đơn hàng.

#	Mã Đơn hàng	Phi Vận chuyển
1	0000000001	75000



Truy cập vào tab đơn hàng để xem các đơn hàng cần vận chuyển. Sau đó click vào biểu tượng + để nhận đơn hàng.

Vietgatel

Đơn hàng
Thông tin cá nhân

#	Mã Đơn hàng	Mã khách hàng	Địa chỉ	Phường	Quận	Tỉnh	Tên người nhận	Số điện thoại	Ngày lập	Phi vận chuyển	Tổng tiền	Thanh toán	
1	0000000000	0930000000	97 Hoà Hảo	Phường 3	Quận 8	TP.HCM	Nguyễn Khắc Hoài Hảo	0521242672	01 Jan 2020 00:00:01:000			Momo	+
2	0000000002	0930000002	91 Hồng Lạc	Phường 1	Quận 12	TP.HCM	Phan Thanh Khắc Anh	0332262152	03 Jan 2020 00:00:01:730			ZaloPay	+
3	0000000003	0930000003	873 Lê Văn Sỹ	Phường 6	Long Biên	Hà Nội	Đào Phước Trần Thúy	0551421731	04 Jan 2020 00:00:01:830	33100	695100	ZaloPay	+
4	0000000008	0930000008	04 Công Chúa Ngọc Hân	Phường 4	Hoàn Kiếm	Hà Nội	Trương Minh Hoài Long	0469029403	09 Jan 2020 00:00:01:320			ZaloPay	+
5	0000000013	0930000013	69 Nguyễn Văn Tạo	Phường 6	Quận 4	TP.HCM	Trương Huy Du	0105421035	14 Jan 2020 00:00:01:240			Bank	+
6	0000000017	0930000017	753 Phùng Văn Cung	Phường 8	Quận 9	TP.HCM	Đào Phước Trang	0979693021	18 Jan 2020 00:00:01:900			Momo	+

TRANG TRƯỚC
TRANG KẾ

Màn hình thông báo nhận đơn hàng thành công

Solutions

localhost:3000 says

Nhận đơn thành công

Backtra

Đơn hàng

IV. Link Google Drive

<https://drive.google.com/drive/folders/1F3gxQ5vabZUYEi1ZNtlstZ9Z2WQZPOEI?usp=sharing>