

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**Khoa Công Nghệ Thông Tin**

# BÁO CÁO CUỐI KỲ

## RESTAURANT SYSTEM

**Giảng viên hướng dẫn:**

**Hoàng Anh Tú**

**Các thành viên trong nhóm 40:**

**Nguyễn Bá Ngọc – MSSV : 1141085**

**Bùi Trung Tân – MSSV : 1141129**

**Nguyễn Minh Nghĩa – MSSV : 1141083**

**Nguyễn Minh Vũ – MSSV : 1141187**

# Phụ Lục

<b>Phụ Lục.....</b>	<b>2</b>
<b>1. Thông tin nhóm .....</b>	<b>1</b>
<b>1.1 Thông tin thành viên: .....</b>	<b>1</b>
<b>2. Các tranh chấp đồng thời đã phát hiện.....</b>	<b>1</b>
<b>2.1 1141085 – Nguyễn Bá Ngọc.....</b>	<b>1</b>
<b>2.1.1 Các trường hợp tranh chấp .....</b>	<b>1</b>
<b>2.2 1141187 – Nguyễn Minh Vũ.....</b>	<b>7</b>
<b>2.2.1 Các trường hợp tranh chấp .....</b>	<b>7</b>
<b>2.3 1141083 – Nguyễn Minh Nghĩa .....</b>	<b>24</b>
<b>2.3.1 Các trường hợp tranh chấp .....</b>	<b>24</b>
<b>2.4 1141129 – Bùi Trung Tân .....</b>	<b>28</b>
<b>2.4.1 Các trường hợp tranh chấp .....</b>	<b>28</b>
<b>3. Các chức năng trong đè tài.....</b>	<b>36</b>
<b>3.1 Nhân Viên Tiếp Tân .....</b>	<b>36</b>
<b>3.1.1 Xếp bàn ăn cho khách .....</b>	<b>36</b>
<b>3.1.2 Đặt bàn.....</b>	<b>37</b>
<b>3.1.3 Danh sách đặt bàn .....</b>	<b>39</b>
<b>3.2 Nhân Viên Thu Ngân.....</b>	<b>40</b>
<b>3.2.1 Thêm món ăn .....</b>	<b>40</b>
<b>3.2.2 Xoá món ăn, cập nhật số lượng món ăn.....</b>	<b>40</b>
<b>3.2.3 Chuyển bàn.....</b>	<b>41</b>
<b>3.2.4 Ghép bàn .....</b>	<b>41</b>
<b>3.2.5 Kết thúc sử dụng bàn.....</b>	<b>41</b>
<b>3.3 Nhân Viên Quản Lý Kho .....</b>	<b>43</b>
<b>3.3.1.Quản Lý Danh Sách Nguyên Liệu .....</b>	<b>44</b>
<b>3.3.2.Quản Lý Danh Sách Nhà Cung Cấp .....</b>	<b>46</b>
<b>3.3.3.Quản Lý Thông Tin Đặt Hàng.....</b>	<b>49</b>
<b>3.3.4.Quản Lý Thông Tin Nhập Kho.....</b>	<b>51</b>
<b>3.4 Quản Lý Nhà Hàng.....</b>	<b>52</b>

<b>3.4.1.Quản lý Thực đơn.....</b>	<b>53</b>
<b>3.4.2.Quản Lý Danh Sách Nhân Viên.....</b>	<b>56</b>
<b>3.4.3.Quản Lý Danh Sách Bàn Ăn .....</b>	<b>60</b>

## 1. Thông tin nhóm

### 1.1 Thông tin thành viên:

STT	MSSV	Họ tên	Nhóm trưởng	Đánh giá
1	1141085	Nguyễn Bá Ngọc	X	100%
2	1141187	Nguyễn Minh Vũ		
3	1141083	Nguyễn Minh Nghĩa		
4	1141129	Bùi Trung Tân		

## 2. Các tranh chấp đồng thời đã phát hiện

### 2.1 1141085 - Nguyễn Bá Ngọc

#### 2.1.1 Các trường hợp tranh chấp

##### a. Drity read: Nghiệp vụ Đặt Bàn Ăn.

- Kích bản:

T1	T2
» Thực hiện thao tác Thêm bàn đặt	
	» Xem danh sách đặt bàn
» Hủy thao tác Thêm bàn đặt	

⚡ Khi T1 đang thực hiện thao tác Thêm 1 bàn đặt mà chưa Commit, cùng lúc đó T2 giữ khóa Read Uncommitted (đọc không cần xin khóa) đọc danh sách các bàn đã được đặt trong đó có bàn mà T1 mới đặt. Lỗi Drity Read xảy ra khi T1 hủy bản vừa mới đặt.

- Sửa lỗi:

T1	T2
<pre> » begin tran » insert into DatBan   (maNH,MaBan,MaKhachHang,   ThoiGianDat,TrangThai,ThoiGianDen,sdt)   values(1,3,123654789,'05/12/2012',0,'05/12/   2012',0) </pre>	
	<pre> » SET TRANSACTION ISOLATION   LEVEL READ UNCOMMITTED » begin tran » select * from datban with   (rowlock,readcommitted) » commit tran </pre>
<pre> » waitfor delay '0:0:5' » declare @sdt int » select @sdt=sdt from DatBan where maban=3 » if @@error&gt;0 or @sdt=0   rollback tran » else commit tran </pre>	

⚡ Khi T1 đang thực hiện thao tác Thêm 1 bàn đặt mà chưa Commit, cùng lúc đó T2 đọc danh sách bàn đã được đặt “With (rowlock,readcommitted)” (xin khóa đọc những dòng dữ liệu cần đọc trong 1 bảng và nhả khóa sau khi đọc xong). Như vậy: T2 đọc dữ liệu, nếu trong dữ liệu đó có dòng của T1 đã thêm thì T2 sẽ phải chờ T1 thực hiện xong mới thực hiện được việc đọc dữ liệu.

### b. Lost update: Nghệp vụ Đặt bàn ăn.

- Kịch bản:

T1	T2
» Đọc danh sách đặt bàn	» Đọc danh sách đặt bàn.
» Cập nhật tên khách hàng đã đặt bàn.	» Cập nhật tên khách hàng đã đặt bàn.
» Thực hiện xong	» Thực hiện xong

↳ T1 đọc danh sách bàn đã được đặt, và cập nhật tên khách hàng của bàn số 3, cùng lúc đó T2 đọc danh sách bàn đã được đặt và cập nhật tên khách hàng khác cũng trên cùng bàn số 3, cùng bàn mà T1 đã sửa. Như vậy sau khi thực hiện xong, dữ liệu của T1 không giống như dữ liệu mà mình đã thay đổi trước đó. Lý do: vì T2 đã ghi đè lên dữ liệu của T1, nên dữ liệu của T1 bị mất khi cập nhật.

- Sửa lỗi:

T1	T2
<pre> » begin tran » select * from DatBan with(rowlock,xlock) where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101) and TrangThai=0 and maBan=3 » waitfor delay '0:0:05' </pre>	<pre> » begin tran </pre>
	<pre> » select * from DatBan where MaNH=1 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101) and TrangThai=0 and maBan=3 » waitfor delay '0:0:5' </pre>
<pre> » update Datban set maKhachHang=123654789 where MaBan=3 </pre>	
	<pre> » update Datban set maKhachHang=234234 where MaBan=3 </pre>
<b>commit tran</b>	<b>commit tran</b>

⚡ T1 đọc danh sách bàn đã được đặt với khóa (*rowlock,xlock*) nghĩa là đặt khóa ghi cho các dòng mà T1 muốn đọc, và cập nhật tên khách hàng của bàn số 3, cùng lúc đó T2 đọc bàn mà T1 đang đọc, T2 sẽ không được đọc, sẽ phải chờ cho đến khi T1 hoàn thành xong thì T2 mới thực hiện.

### c. Unrepeatable read: Nghiệp vụ Đặt bàn ăn.

- Kích bàn:

T1	T2
» Đọc danh sách đặt bàn	» Sửa tình trạng bàn đã đặt bàn thành bàn đang sử dụng
» Đọc danh sách đặt bàn	

⚡ T1 đọc danh sách đặt bàn lần 1, T2 cập nhật tình trạng bàn từ được đặt thành đang sử dụng, tiếp đó T1 đọc lại danh sách đặt bàn lần 2. Khi ta thấy rằng đọc lần 1 của T1 sẽ có kết quả khác với lần đọc thứ 2. Lần đọc 1 có số dòng nhiều hơn lần đọc 2.

- Sửa lỗi:

T1	T2
<pre>» begin tran » select MaBan from DatBan with   (rowlock,repeatableread) where MaNH=1 and ThoiGianDen=CONVERT(nvarchar, GETDATE(),101) and TrangThai=0 » waitfor delay '0:0:05'</pre>	
	<pre>» begin tran » update DatBan set TrangThai=1 where MaNH=1 and MaBan=4 and ThoiGianDen=CONVERT(nvarchar,GETDATE(),101) » commit tran</pre>
<pre>» select MaBan from DatBan where MaNH=1 and ThoiGianDen=CONVERT(nvarchar, GETDATE(),101) and TrangThai=0 » commit tran</pre>	

↳ T1 đọc danh sách đặt bàn lần 1 với khóa (*rowlock,repeatable read*) nghĩa là khi T1 đọc sẽ giữ khóa đọc những dòng T1 cần đọc cho đến hết transaction. Lúc này T2 thực hiện việc cập nhật dòng dữ liệu mà T1 đọc, T1 đang giữ khóa đọc nên T2 sẽ không cấp được khóa Ghi. T2 muốn thực hiện phải chờ T1 thực hiện xong thì T2 mới thực hiện được.

#### *d. Phantom: Nghiệp vụ Đặt bàn ăn.*

- Kích bản:

T1	T2
» Đọc danh sách đặt bàn	
	» Thêm bàn đặt
» Đọc danh sách đặt bàn	

↳ T1 đọc danh sách đặt bàn lần 1, T2 thực hiện thao tác thêm 1 bàn đặt, tiếp đó T1 đọc lại danh sách đặt bàn lần 2. Khi ta thấy rằng đọc lần 1 của T1 sẽ có kết quả khác với lần đọc thứ 2. Lần đọc 1 có số dòng ít hơn lần đọc 2.

- Sửa lỗi:

T1	T2
<pre>» begin tran » select MaBan from DatBan with   (rowlock,serializable) where MaNH=1 and ThoiGianDen=CONVERT(nvarchar, GETDATE(),101) and TrangThai=0 » waitfor delay '0:0:05'</pre>	
	<pre>» begin tran » insert into DatBan (maNH,MaBan,MaKhachHang,ThoiGianDat, TrangThai,ThoiGianDen,sdt) values(1,2,123456756,'05/10/2012',0,CONVERT( nvarchar,GETDATE(),101),234234) » commit tran</pre>
» select MaBan from DatBan	

```

where MaNH=1 and
ThoiGianDen=CONVERT(nvarchar,
GETDATE(),101) and TrangThai=0
» commit tran

```

↳ T1 đọc danh sách đặt bàn lần 1 với khóa (*rowlock, serializable*) nghĩa là khi T1 đọc sẽ giữ khóa đọc những dòng T1 cần đọc cho đến hết transaction. Lúc này T2 thực hiện việc Thêm dữ liệu, T1 đang giữ khóa đọc nên T2 sẽ không cấp được khóa Ghi. T2 muốn thực hiện phải chờ T1 thực hiện xong thì T2 mới thực hiện được.

#### e. Deadlock: Nghiệp vụ Xếp bàn ăn cho khách.

- **Kích bản:**

T1	T2
» Đọc danh sách bàn ăn	
	» Đọc danh sách bàn ăn
» Cập nhật tình trạng bàn ăn	
	» Cập nhật tình trạng bàn ăn

↳ T1 đọc danh sách bàn ăn và giữ khóa đọc đến hết giao tác, T2 cũng đọc danh sách bàn ăn và giữ khóa đọc đến hết giao tác. T1 cập nhật tình trạng bàn ăn nên phải chờ cho T2 nhả khóa đọc trên bảng bàn ăn, cùng lúc đó T2 cũng cập nhật tình trạng bàn ăn và cũng phải chờ cho đến khi T1 nhả khóa đọc trong bảng bàn ăn. Nên cả hai cùng chờ nhau gây lỗi conversion deadlock

- **Sửa lỗi:**

T1	T2
<pre> » SET TRANSACTION ISOLATION LEVEL repeatable read » begin tran » select * from DatBan with   (rowlock,xlock) where MaNH=1 and ThoiGianDen=CONVERT(nvarchar, GETDATE(),101) and TrangThai=0 » waitfor delay '0:0:05' </pre>	

	<pre> » SET TRANSACTION ISOLATION LEVEL     repeatable read  » begin tran  » select * from DatBan with (rowlock,xlock)     where MaNH=1 and         ThoiGianDen=CONVERT(nvarchar,             GETDATE(),101) and TrangThai=0     » waitfor delay '0:0:5' </pre>
» update Datban set TinhTrang=1 where MaBan=2	
» commit tran	<pre> » update Datban set TinhTrang=1 where     MaBan=2 </pre>
	» Commit tran

↳ T1 đọc danh sách bàn ăn và giữ khóa ghi đến hết giao tác, T2 cũng đọc danh sách bàn ăn nhưng T2 phải chờ cho đến khi T1 hoàn thành giao tác, nên sẽ không xảy ra hiện tượng conversion Deadlock.

## 2.2 1141187 - Nguyễn Minh Vũ

### 2.2.1 Các trường hợp tranh chấp

#### a. Dirty read

Trường hợp 1 :

Kịch bản :

T1	T2
- Thêm vào một nguyên liệu mới	- Lấy danh sách nguyên liệu để hiển thị lên màn hình
- Kiểm tra tên nguyên liệu mới có là duy nhất hay không, nếu có 2 nguyên liệu trùng tên và đơn vị =>Hủy bỏ việc thêm nguyên liệu này	

⇒ Giao tác 2 sẽ đọc dữ liệu mà Giao tác T1 vừa thêm vào , nhưng sau đó Giao tác T1 hủy việc thêm vào này nên Giao tác T2 đọc dữ liệu sai

Store procedure :

T1	T2
<pre> create proc InsertNguyenLieu     @Flag int out,     @MaNH nchar(10),     @TenNL nvarchar(50),     @DonVi nvarchar(20),     @SoLuongTon int As set @Flag = 0 begin tran set transaction isolation level read uncommitted insert into NguyenLieu values (     @MaNH     ,@TenNL     ,@DonVi     ,@SoLuongTon ) waitfor delay '00:00:05' </pre>	
	<pre> create proc SelectNguyenLieu     @MaNH nchar(10) as begin     begin tran     set transaction isolation level     read uncommitted     select *     from NguyenLieu nl     where nl.MaNH = @MaNH     order by         nl.TenNL      commit tran end go </pre>
<pre> if ((select count(*)       from NguyenLieu       where TenNL = @TenNL             and DonVi = @DonVi             and MaNH = @MaNH)&lt;&gt;1) begin     rollback tran     return end  if (@@error&lt;&gt;0) begin     rollback tran     return end  set @Flag = 1 commit end </pre>	

[go](#)

### Nguyên nhân tranh chấp:

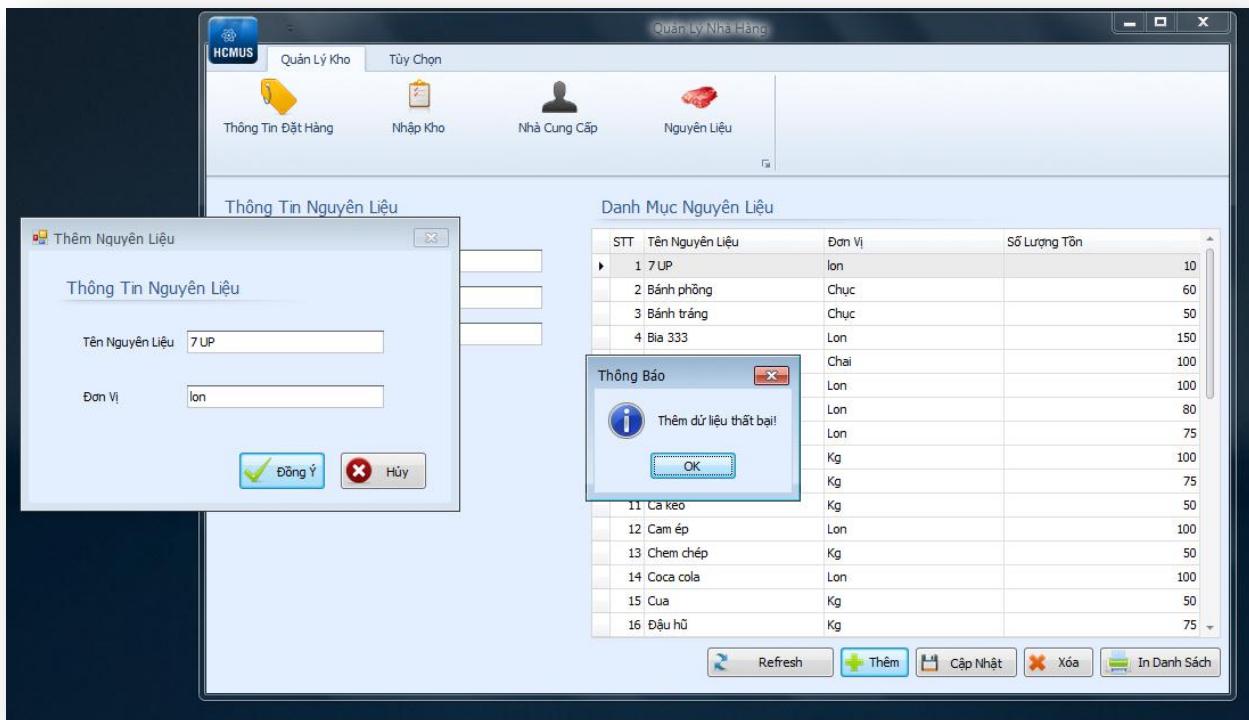
Với mức cô lập **read uncommitted** Giao tác T2 không xin khóa đọc khi lấy dữ liệu, vì vậy khi T1 đang ghi dữ liệu, nó vẫn có thể đọc dữ liệu, dẫn đến tranh chấp

### Cách khắc phục:

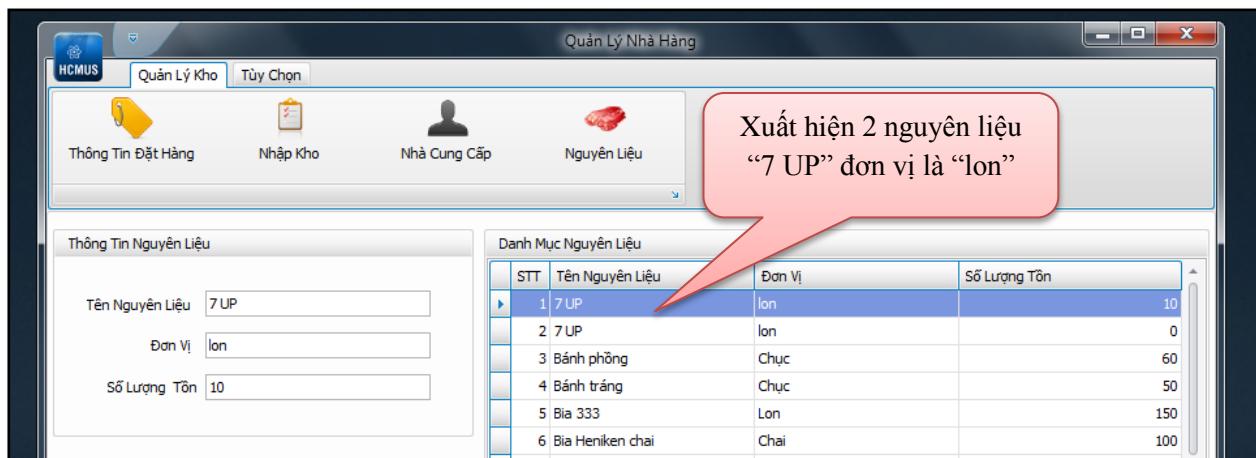
Sử dụng mức cô lập **read committed** :với mức cô lập này khi T2 cần đọc dữ liệu nó phải xin khóa đọc,nhưng do T1 đang ghi, nên nó không xin được khóa đọc mà phải đợi đến khi T1 commit T2 mới có thể đọc dữ liệu => Khắc phục được lỗi đọc dữ liệu rác

### Giao diện chương trình:

#### Giao tác T1:



#### Giao tác T2:



### b. Lost update

Trường hợp 1 :

Kịch bản :

T1	T2
-	
	-
-	

Store procedure :

T1	T2

Nguyên nhân tranh chấp:

Cách khắc phục:

Giao diện chương trình:

Giao tác T1:

Giao tác T2:

### c. Unrepeatable tread

Trường hợp 1 :

Kịch bản :

T1	T2
<ul style="list-style-type: none"> <li>- Kiểm tra danh sách nguyên liệu của nhà cung cấp =&gt; Lấy danh sách nguyên liệu có trong nhà cung cấp này</li> </ul>	
<ul style="list-style-type: none"> <li>- Kiểm tra danh sách nguyên liệu của nhà cung cấp =&gt; Lấy danh sách nguyên liệu không có trong nhà cung cấp này (dùng để cập nhật thêm nguyên liệu cho chi tiết nhà cung cấp chưa có)</li> </ul>	<ul style="list-style-type: none"> <li>- Cập nhật lại danh sách nguyên liệu của nhà cung cấp</li> </ul>

⇒ Hiện tại trong kho quản lý 10 danh mục nguyên liệu .Nhà cung cấp A cung cấp 4 nguyên liệu

1/Giao tác T1 lấy danh sách nguyên liệu của nhà cung cấp => có 4 nguyên liệu

2/Giao tác T2 xóa 2 nguyên liệu từ nhà cung cấp => Nhà cung cấp bây giờ có 2 nguyên liệu

3/Giao tác T1 kiểm tra danh sách nguyên liệu của nhà cung cấp => có 2 nguyên liệu

Và lấy ra danh sách 8 nguyên liệu còn lại (danh sách nguyên liệu còn lại phải là 10-4 =6)

⇒ Giữa lần đọc dữ liệu thứ nhất và lần đọc dữ liệu thứ 2 của giao tác T1 có sự thay đổi, lần thứ 2 bị mất 2 nguyên liệu

*Store procedure :*

T1	T2
<pre> create proc SelectNguyenLieu_NCC     @MaNCC int,     @MaNH nchar(10) as begin     begin tran     set transaction isolation level     read uncommitted      select nl.*         ,ct.Gia     from   NhaCungCap ncc             ,ChiTietNCC ct             ,NguyenLieu nl     where  ct.MaNCc = @MaNCC             and nl.MaNH = @MaNH             and ct.MaNL = nl.MaNL             and ct.MaNCc = ncc.MaNCc     order by nl.TenNL     waitfor delay '00:00:05' </pre>	<pre> create proc UpdateNhaCungCap     @Flag int out,     @MaNCC int,     @TenNCC nvarchar(50),     @sdt nvarchar(50),     @DiaChi nvarchar(50),     @DiemUuTien int,     @ChiTiet TableType_ChiTietNCC readonly as begin     set @Flag = 0     begin tran     set transaction isolation level     read uncommitted      if(( select count(*) from NhaCungCap         where MaNCc = @MaNCC )&lt;&gt;0)     begin         update NhaCungCap         set TenNCC = @TenNCC             ,sdt = @sdt             ,DiaChi = @DiaChi             ,DiemUuTien = @DiemUuTien     end </pre>

```

        where MaNCC = @MaNCC
        if(@@ERROR<>0)
begin
    set @Flag = 0
    rollback
    return
end

--Xoa nguyen lieu da bi mat trong
danh sach
delete ChiTietNCC
where MaNCC = @MaNCC
and MaNL in
(select ctn.MaNL
from ChiTietNCC ctn
where ctn.MaNCC = @MaNCC
and ctn.MaNL not in
(select MaNL
from @ChiTiet))
if(@@ERROR<>0)
begin
    set @Flag = 0
    rollback
    return
end
--Them cac nguyen lieu moi cap nhat
insert into ChiTietNCC
select ct.MaNL
,ct.MaNCC
,ct.Gia
from @ChiTiet ct
where ct.MaNL not in
(select ctn.MaNL
from ChiTietNCC ctn
where ctn.MaNCC = @MaNCC)

if(@@ERROR<>0)
begin
    set @Flag = 0
    rollback
    return
end
end
set @Flag = 1
commit tran
end

select *
from NguyenLieu nl1
where nl1.MaNH = @MaNH
and nl1.MaNL not in
(select nl1.MaNL
from NguyenLieu nl1
,ChiTietNCC ct
where ct.MaNCC = @MaNCC
and nl1.MaNH = @MaNH
and nl1.MaNL = ct.MaNL)
order by nl1.TenNL

```

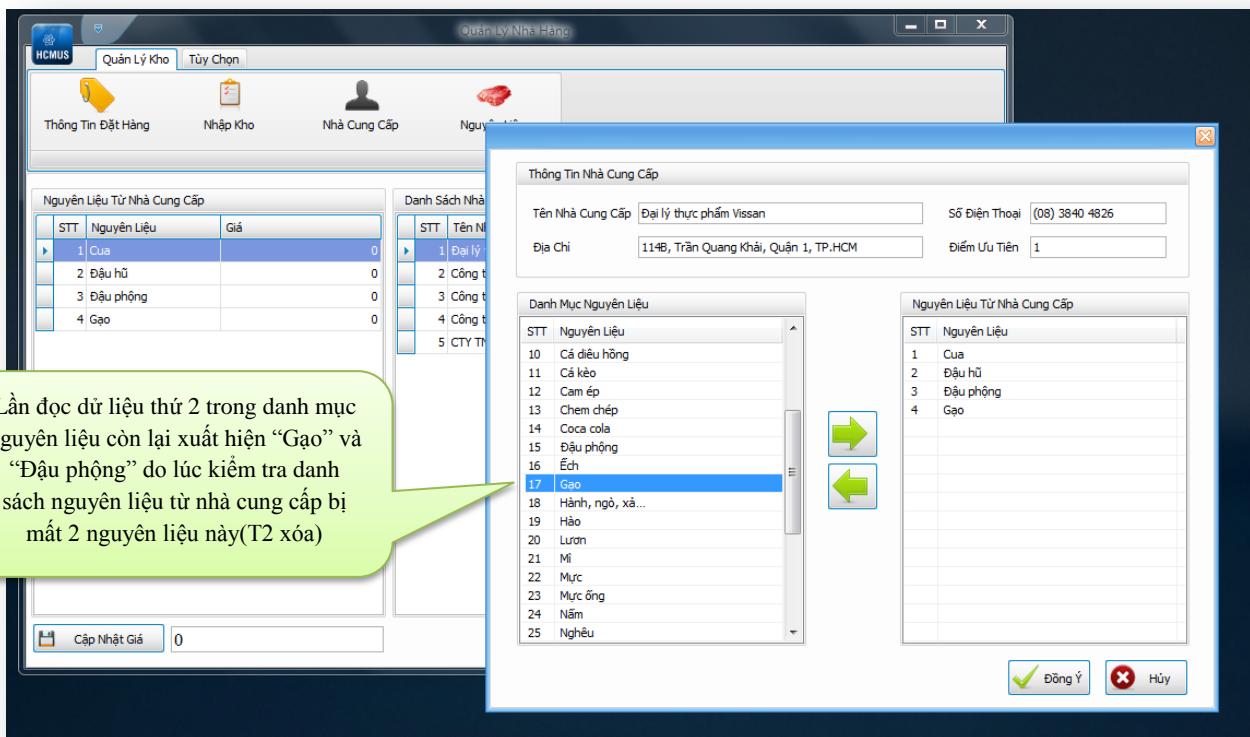
*Nguyên nhân tranh chấp:*

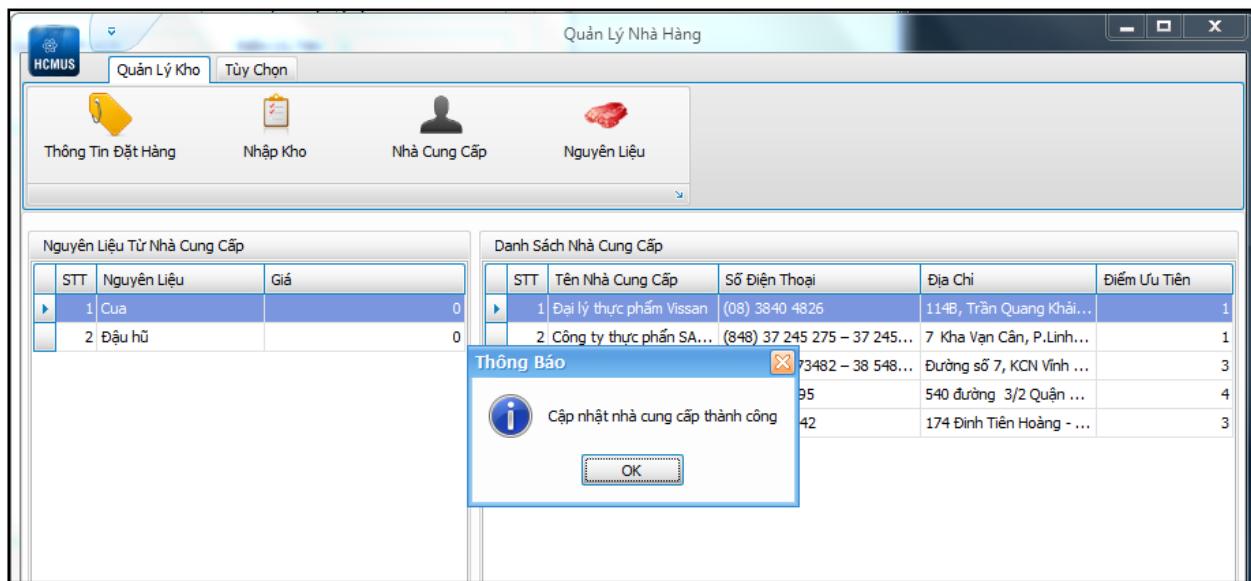
Với mức cõ lập **read uncommitted** không thiết lập **Shared Lock** trên đơn vị dữ liệu được đọc, vì vậy khi T1 đọc dữ liệu lần 1 T2 có quyền ghi dữ liệu(update,delete), dẫn đến lần đọc dữ liệu thứ 2 của T1 có sự thay đổi.

*Cách khắc phục:*

Mức cõ lập : **read committed** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và **Shared Lock** được giải phóng ngay sau khi đọc xong dữ liệu, vì vậy khi T1 đọc dữ liệu xong T2 vẫn có thể update hoặc delete dữ liệu này => không giải quyết được tranh chấp.

Mức cõ lập : **repeatable read** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và giữ **Shared Lock** này đến hết giao tác . Vì vậy ghi T1 đọc dữ liệu T2 không thể update hoặc delete những đơn vị dữ liệu mà T1 đang giữ **Shared Lock** => Khắc phục được tranh chấp dữ liệu

*Giao diện chương trình:**Giao tác T1:**Giao tác T2:*



### d. Phantom

Trường hợp 1 :

Kịch bản :

T1	T2
<ul style="list-style-type: none"> <li>- Kiểm tra danh sách nguyên liệu của nhà cung cấp =&gt; Lấy danh sách nguyên liệu có trong nhà cung cấp này</li> </ul>	
<ul style="list-style-type: none"> <li>- Kiểm tra danh sách nguyên liệu của nhà cung cấp =&gt; Lấy danh sách nguyên liệu không có trong nhà cung cấp này (dùng để cập nhật thêm nguyên liệu cho chi tiết nhà cung cấp chưa có)</li> </ul>	<ul style="list-style-type: none"> <li>- Cập nhật lại danh sách nguyên liệu của nhà cung cấp</li> </ul>

⇒ Hiện tại trong kho quản lý 10 danh mục nguyên liệu .Nhà cung cấp A cung cấp 4 nguyên liệu

1/Giao tác T1 lấy danh sách nguyên liệu của nhà cung cấp => có 4 nguyên liệu

2/Giao tác T2 thêm 2 nguyên liệu từ nhà cung cấp => Nhà cung cấp bây giờ có 6 nguyên liệu

3/Giao tác T1 kiểm tra danh sách nguyên liệu của nhà cung cấp => có 6 nguyên liệu

Và lấy ra danh sách 4 nguyên liệu còn lại (danh sách nguyên liệu còn lại phải là  $10 - 4 = 6$ )

⇒ Giữa lần đọc dữ liệu thứ nhất và lần đọc dữ liệu thứ 2 của giao tác T1 có sự thay đổi, lần thứ 2 xuất hiện thêm 2 nguyên liệu

Store procedure :

T1	T2
<pre>create proc SelectNguyenLieu_NCC     @MaNCC int,</pre>	

<pre> @MaNH nchar(10) as begin     begin tran     set transaction isolation level     read uncommitted      select nl.*         ,ct.Gia     from   NhaCungCap ncc         ,ChiTietNCC ct         ,NguyenLieu nl     where  ct.MaNC = @MaNC         and nl.MaNH = @MaNH         and ct.MaNL = nl.MaNL         and ct.MaNC = ncc.MaNC     order by nl.TenNL     waitfor delay '00:00:05' </pre>	
	<pre> create proc UpdateNhaCungCap     @Flag int out,     @MaNC int,     @TenNC nvarchar(50),     @sdt nvarchar(50),     @DiaChi nvarchar(50),     @DiemUUtien int,     @ChiTiet TableType_ChiTietNCC readonly as begin     set @Flag = 0     begin tran     set transaction isolation level     read uncommitted      if(( select count(*) from NhaCungCap         where MaNC = @MaNC )&lt;&gt;0)     begin         update NhaCungCap         set TenNC = @TenNC             ,sdt = @sdt             ,DiaChi = @DiaChi             ,DiemUUtien = @DiemUUtien         where MaNC = @MaNC         if(@@ERROR&lt;&gt;0)         begin             set @Flag = 0             rollback             return         end          --Xoa nguyen lieu da bi mat trong         danh sach         delete ChiTietNCC         where MaNC = @MaNC             and MaNL in             (select ctn.MaNL             from ChiTietNCC ctn             where ctn.MaNC = @MaNC             and ctn.MaNL not in </pre>

	<pre> (select MaNL from @ChiTiet))  if(@@ERROR&lt;&gt;0) begin     set @Flag = 0     rollback     return end --Them cac nguyen lieu moi cap nhat insert into ChiTietNCC select ct.MaNL ,ct.MaNCC ,ct.Gia from @ChiTiet ct where ct.MaNL not in (select ctn.MaNL from ChiTietNCC ctn where ctn.MaNCC = @MaNCC)  if(@@ERROR&lt;&gt;0) begin     set @Flag = 0     rollback     return end set @Flag = 1 commit tran end </pre>
<pre> select * from NguyenLieu nl1 where nl1.MaNH = @MaNH and nl1.MaNL not in (select nl1.MaNL from NguyenLieu nl1 ,ChiTietNCC ct where ct.MaNCC = @MaNCC and nl1.MaNH = @MaNH and nl1.MaNL = ct.MaNL) order by nl1.TenNL </pre>	

*Nguyên nhân tranh chấp:*

Với mức cô lập **read uncommitted** không thiết lập **Shared Lock** trên đơn vị dữ liệu được đọc, vì vậy khi T1 đọc dữ liệu lần 1 T2 có quyền ghi dữ liệu(update,delete), dẫn đến lần đọc dữ liệu thứ 2 của T1 có sự thay đổi.

*Cách khắc phục:*

Mức cô lập : **read committed** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và **Shared Lock** được giải phóng ngay sau khi đọc xong dữ liệu, vì vậy khi T1 đọc dữ liệu xong T2 vẫn có thể insert thêm dữ liệu

⇒ Kkhông giải quyết được tranh chấp

Mức cô lập : **repeatable read** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và giữ **Shared Lock** này đến hết giao tác . Vì vậy ghi T1 đọc dữ liệu T2 vẫn có thể insert dữ liệu => không giải quyết được tranh chấp

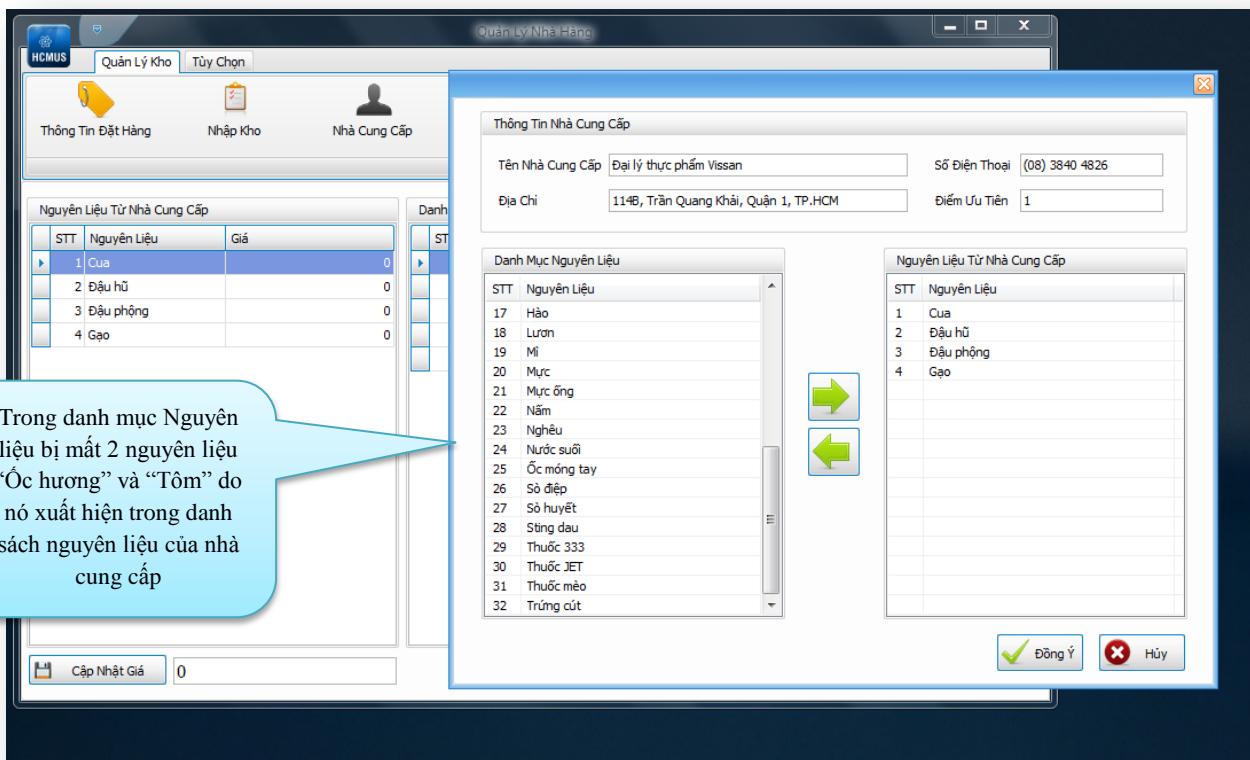
Mức cô lập : **Serializable** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và giữ **Shared Lock** này đến hết giao tác .Đồng thời không cho phép insert những dòng dữ liệu thỏa mãn điều kiện thiết lập **Shared Lock**

Do đó khi T1 đọc dữ liệu trên bảng ChiTietNCC thì T2 không được insert dữ liệu lên bảng này

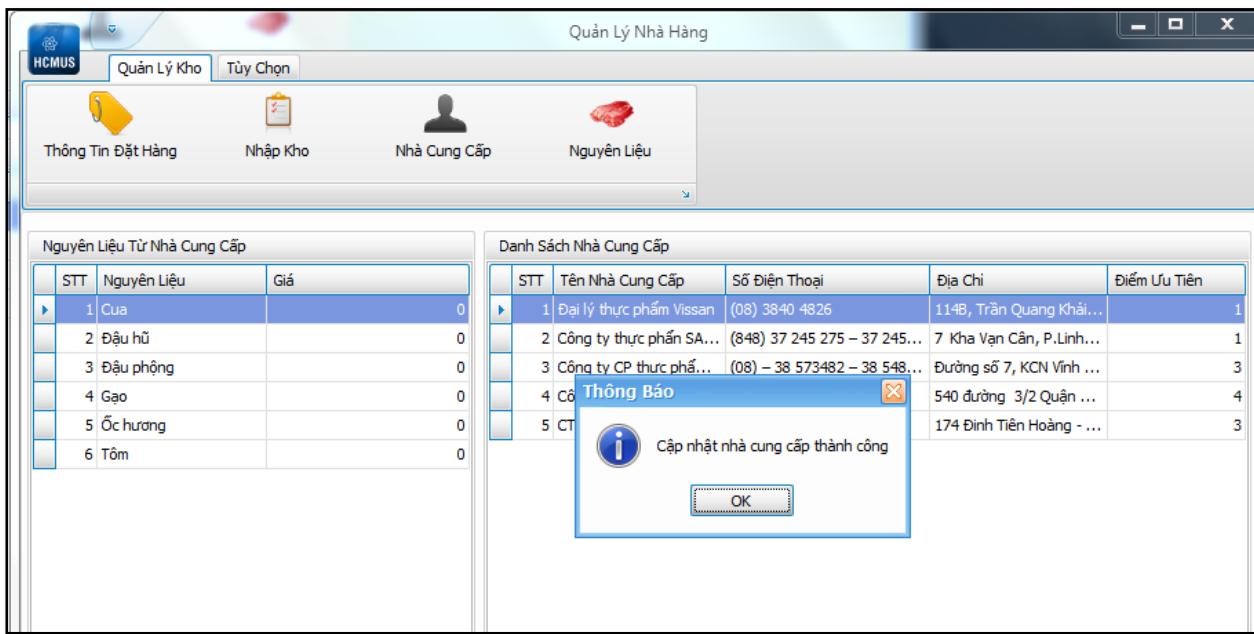
⇒ Khắc phục được tranh chấp

*Giao diện chương trình:*

*Giao tác T1:*



Giao tác T2:



### e. deadlock

Trường hợp 1 :

Kịch bản :

T1	T2
- thêm hóa đơn nhập kho, cập nhật lại tình trạng đơn đặt hàng là “Đã giao”	
- Lấy số lượng tồn của nguyên liệu	- thêm hóa đơn nhập kho, cập nhật lại tình trạng đơn đặt hàng là “Đã giao” - Lấy số lượng tồn của nguyên liệu
- Cập nhật lại số lượng tồn của nguyên liệu = số lượng tồn cũ + với số lượng nguyên liệu có trong chi tiết hóa đơn	
	- Cập nhật lại số lượng tồn của nguyên liệu = số lượng tồn cũ + với số lượng nguyên liệu có trong chi tiết hóa đơn

- ⇒ Xét trường hợp 1 nguyên liệu có số lượng tồn = 50,
- ⇒ Người dùng 1 tiến hành nhập kho vào số lượng nguyên liệu này là 5 và cập nhật số lượng tồn của nguyên liệu =  $50 + 5 = 55$
- ⇒ Người dùng 2 cũng tiến hành nhập kho với số lượng nguyên liệu là 10 và cập nhật số lượng tồn =  $50 + 10 = 60$

⇒ Trong khi đó số lượng nguyên liệu thực tế trong kho là  $50 + 5 + 10 = 65$

*Store procedure :*

T1	T2
<pre> create proc UpdateTinhTrangDatHang     @Flag int out,     @MaHoaDon int,     @TinhTrang nvarchar(50) as begin     set @Flag = 0     begin tran     set transaction isolation level     read uncommitted     if(@TinhTrang &lt;&gt; N'Dã Giao')     begin         update DatHang         set TinhTrang = @TinhTrang         where MaHoaDon = @MaHoaDon          if(@@ERROR&lt;&gt;0)         begin             set @Flag = 0             rollback             return         end     end     else     begin         update DatHang         set TinhTrang = @TinhTrang         where MaHoaDon = @MaHoaDon         if(@@ERROR&lt;&gt;0)         begin             set @Flag = 0             rollback             return         end         --Cap nhat so luong nguyen lieu ton         declare @soluongton int,                 @soluongthem int,                 @MaNL int         declare @cur cursor         set @cur = cursor for         select MaNL         from ChiTietDatHang ctdh         where ctdh.MaHoaDon = @MaHoaDon          open @cur         fetch next from @cur into @MaNL         while @@FETCH_STATUS = 0         begin             set @soluongthem = (                 select ctdh.SoLuong                 from ChiTietDatHang ctdh                 where ctdh.MaHoaDon = @MaHoaDon                 and MaNL = @MaNL)             set @soluongton =                 (select SoLuongTon                 from NguyenLieu                 where MaNL = @MaNL)             waitfor delay '00:00:06'         end     end end </pre>	
	<pre> create proc UpdateTinhTrangDatHang     @Flag int out,     @MaHoaDon int,     @TinhTrang nvarchar(50) as begin </pre>

	<pre> set @Flag = 0 begin tran set transaction isolation level read uncommitted if(@TinhTrang &lt;&gt; N'Đã Giao') begin     update DatHang     set TinhTrang = @TinhTrang     where MaHoaDon = @MaHoaDon      if(@@ERROR&lt;&gt;0)     begin         set @Flag = 0         rollback         return     end else begin     update DatHang     set TinhTrang = @TinhTrang     where MaHoaDon = @MaHoaDon     if(@@ERROR&lt;&gt;0)     begin         set @Flag = 0         rollback         return     end --Cap nhat so luong nguyen lieu ton declare @soluongton int,         @soluongthem int,         @MaNL int declare @cur cursor set @cur = cursor for select MaNL from ChiTietDatHang ctdh where ctdh.MaHoaDon = @MaHoaDon  open @cur fetch next from @cur into @MaNL while @@FETCH_STATUS = 0 begin     set @soluongthem = (         select ctdh.SoLuong         from ChiTietDatHang ctdh         where ctdh.MaHoaDon = @MaHoaDon         and MaNL = @MaNL)     set @soluongton = (select SoLuongTon from NguyenLieu where MaNL = @MaNL)     waitfor delay '00:00:06' </pre>
<pre> update NguyenLieu set SoLuongTon = @soluongthem+@soluongton where MaNL = @MaNL  if(@@ERROR&lt;&gt;0) begin     set @Flag = 0     rollback     return end fetch next from @cur into @MaNL end end set @Flag = 1 commit tran end </pre>	
	<pre> update NguyenLieu set SoLuongTon </pre>

```

= @soluongthem+@soluongton
where MaNL = @MaNL

if (@@ERROR<>0)
begin
    set @Flag = 0
    rollback
    return
end
fetch next from @cur into @MaNL
end
set @Flag = 1
commit tran
end

```

### *Nguyên nhân tranh chấp:*

Với mức cõ lập **read uncommitted** không thiết lập **Shared Lock** trên đơn vị dữ liệu được đọc, vì vậy khi T1 đọc dữ liệu A thì T2 vẫn được đọc dữ A này, nên khi T2 cập nhật dữ liệu với thông tin đọc được là A thì trên dài T1 đã cập nhật lại dữ liệu A này thành giá trị khác rồi => dẫn đến T2 cập nhật dữ liệu với thông tin sai

### *Cách khắc phục:*

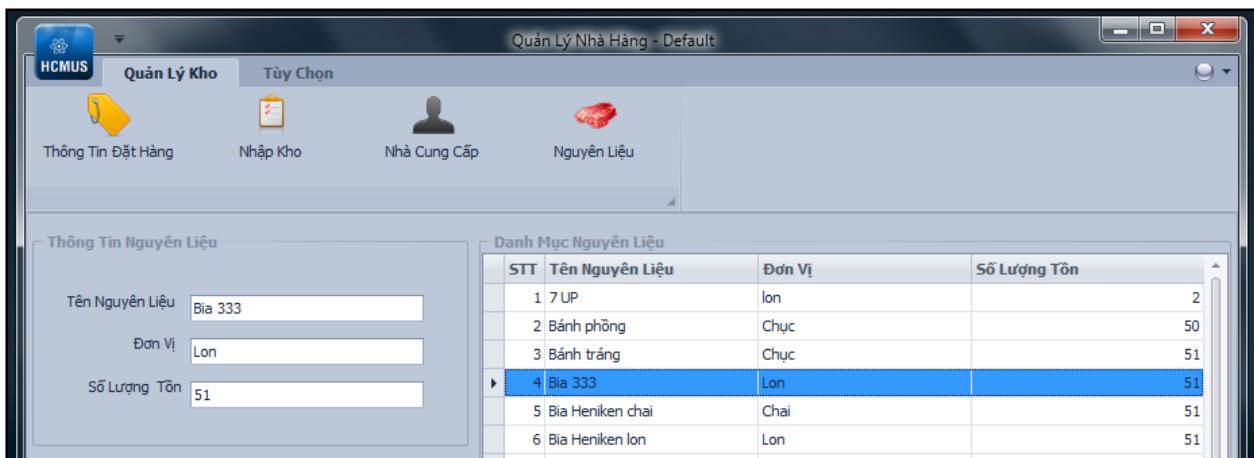
Mức cõ lập : **read committed** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và **Shared Lock** được giải phóng ngay sau khi đọc xong dữ liệu, vì vậy khi T1 đọc dữ liệu xong T2 vẫn được đọc dữ liệu này => không giải quyết được tranh chấp

Mức cõ lập : **repeatable read** Tạo **Shared Lock** trên đơn vị dữ liệu được đọc và giữ **Shared Lock** này đến hết giao tác . với mức cõ lập này T1 đọc dữ liệu T2 vẫn được đọc => Khi T2 phát khóa **Shared Lock** thì T1 không được ghi do T2 đang giữ khóa đọc nên bị deadlock, trong khi đó T2 sau khi đọc xong lại xin khóa ghi trong khi T1 lại đang giữ khóa **Shared Lock** nên cũng bị deadlock => chương trình bị treo

⇒ Để giải quyết tình trạng này ta phải kết hợp Mức cõ lập : **repeatable read** và Khóa **Updlock** để khóa nhưng dòng dữ liệu khi 1 giao tác đang đọc

### *Giao diện chương trình:*

Xét Nguyên Liệu “Bia 333” số lượng tồn hiện tại =51



Người dùng 1: Nhập kho đơn đặt hàng từ công ty SAFACO với số lượng là 1=>cập nhật số lượng "Bia 333" tăng lên 1

Chi Tiết Nhập Hàng			
STT	Nguyên Liệu	Số Lượng	Thành Tiền
►	1 Bia 333	1	12000

Danh Sách Đặt Hàng Chưa Giao			
STT	Nhà Cung Cấp	TongTien	Thời Gian Đặt
1	Công ty CP thực phẩm Cholimex	116000	11/05/2012
2	CTY TNHH Kinh doanh thực phẩm trực tuyến	129000	11/05/2012
3	Công ty TNHH Yến Sào Khánh Hòa	48000	12/05/2012
4	Công ty TNHH Yến Sào Khánh Hòa	51500	19/05/2012
► 5	Công ty thực phẩm SAFACO	12000	20/05/2012
6	Công ty TNHH Yến Sào Khánh Hòa	10000	20/05/2012

Nhập Hàng
 Hủy

Người dùng 2: Nhập kho đơn đặt hàng từ công ty Yến Sào Khánh Hòa với số lượng là 1=>cập nhật số lượng "Bia 333" tăng lên 1

Chi Tiết Nhập Hàng			
STT	Nguyên Liệu	Số Lượng	Thành Tiền
►	1 Bia 333	1	10000

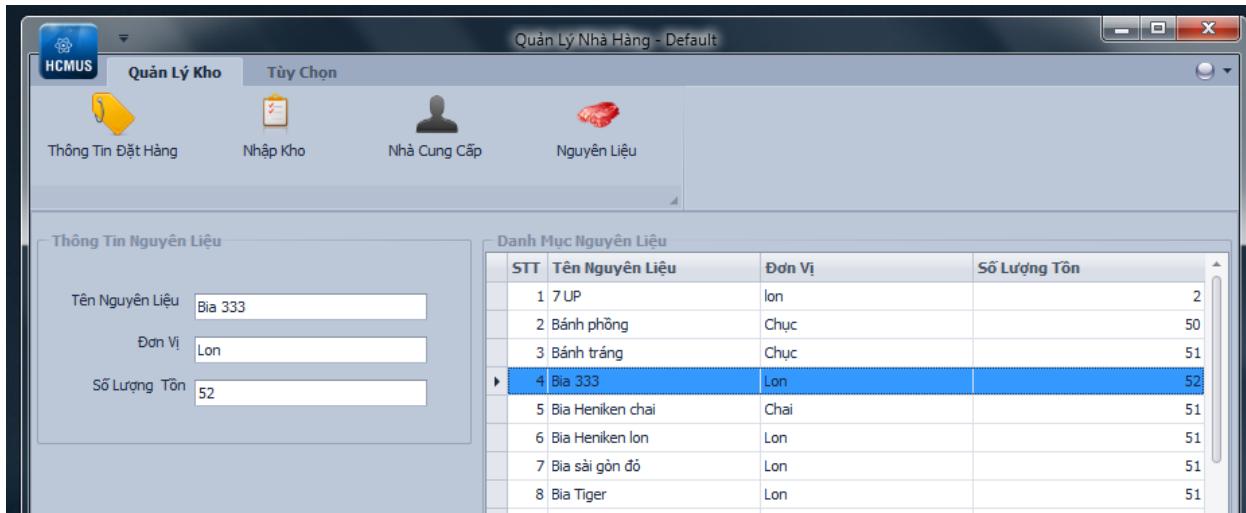
  

Danh Sách Đặt Hàng Chưa Giao			
STT	Nhà Cung Cấp	TongTien	Thời Gian Đặt
1	Công ty CP thực phẩm Cholimex	116000	11/05/2012
2	CTY TNHH Kinh doanh thực phẩm trực tuyến	129000	11/05/2012
3	Công ty TNHH Yến Sào Khánh Hòa	48000	12/05/2012
4	Công ty TNHH Yến Sào Khánh Hòa	51500	19/05/2012
► 5	Công ty thực phẩm SAFACO	12000	20/05/2012
6	Công ty TNHH Yến Sào Khánh Hòa	10000	20/05/2012

Nhập Hàng
 Hủy

Kết quả : sau 2 lần tăng số lượng tồn kết quả = 52 . Nhưng thực tế phải là 53



## 2.3 1141083 – Nguyễn Minh Nghĩa

### 2.3.1 Các trường hợp tranh chấp

#### a. Dirty read

Tranh chấp giữa giao tác Đọc danh sách Món Ăn (T2) và giao tác Thêm Món Ăn(T1)

#	T1	T2
1	<pre> begin tran     insert into         MonAn (TenMon, LoaiMon, Gia, DonViTinh, MaNH)     values         (@TenMon, @LoaiMon, @Gia, @DonViTinh, @MaNH)          if ((select count(*) from MonAn             where TenMon=@TenMon)&lt;&gt;1)             begin                 waitfor delay '00:00:05' </pre>	
2		<pre> set tran isolation level read committed begin tran select     MaMon, TenMon, TenLoai, Gia, DonViTinh     from MonAn m, LoaiMonAn     lm, NhaHang nh     where m.LoaiMon=lm.MaLoai and     m.MaNH=nh.MaNH and nh.MaNH =@MaNH      Commit go </pre>
3	<pre> rollback tran end </pre>	

	<pre> if(@@error&lt;&gt;0) begin     rollback tran     commit end else     commit tran go </pre>	
--	--	--

### b. Lost update

Tranh chấp giữa giao tác Cập nhật Món Ăn (T2) và giao tác Cập nhật Món Ăn (T1)

#	T1	T2
1	<pre> set tran isolation level repeatable read begin tran if((select count(*) from MonAn with(updlock) where MaMon=@MaMon)=1) begin     waitfor delay '00:00:05' </pre>	
2		<pre> set tran isolation level repeatable read begin tran if((select count(*) from MonAn with(updlock) where MaMon=@MaMon)=1) begin     waitfor delay '00:00:05' </pre>
3	<pre> update MonAn set TenMon=@TenMon, Gia=@gia, DonViTinh=@DVTinh where MaMon=@MaMon waitfor delay '00:00:05' </pre>	
		<pre> update MonAn set TenMon=@TenMon, Gia=@gia, DonViTinh=@DVTinh where MaMon=@MaMon waitfor delay '00:00:05' </pre>
4	<pre> if((select count(*) from MonAn where TenMon=@TenMon)&lt;&gt;1) begin     rollback tran     return end end if(@@error &lt;&gt; 0) begin     rollback tran     return end commit tran </pre>	
5		<pre> if((select count(*) from MonAn where TenMon=@TenMon)&lt;&gt;1) begin </pre>

	<pre> rollback tran return  end end  if(@@error &lt;&gt; 0) begin     rollback tran     return end  commit tran </pre>
--	--

### c. Unrepeatable read

Tranh chấp giữa giao tác Xóa Món Ăn (T2) và giao tác Tìm kiếm Món Ăn theo loại(T1)

#	T1	T2
1	<pre> set tran isolation level repeatable read begin tran  select @dem=count(*) from MonAn m,LoaiMonAn lm,NhaHang nh where m.LoaiMon=lm.MaLoai and m.MaNH=nh.MaNH and nh.MaNH =@maNH and lm.MaLoai=@maLoai  waitfor delay '00:00:05' </pre>	
2		<pre> set tran isolation level read committed  begin tran delete MonAn where MaMon=@maMon  commit go </pre>
3	<pre> select MaMon,TenMon,TenLoai,Gia,DonViTinh from MonAn m,LoaiMonAn lm,NhaHang nh where m.LoaiMon=lm.MaLoai and m.MaNH=nh.MaNH and nh.MaNH =@maNH and lm.MaLoai=@maLoai  commit tran </pre>	

*d. Phantom*

**Tranh chấp giữa giao tác Thêm Món Ăn (T2) và giao tác Tìm kiếm Món Ăn theo loại(T1)**

#	T1	T2
1	<pre>set tran isolation level serializable begin tran  select @dem=count(*) from MonAn m,LoaiMonAn lm,NhaHang nh where m.LoaiMon=lm.MaLoai and m.MaNH=nh.MaNH and nh.MaNH =@maNH and lm.MaLoai=@maLoai  waitfor delay '00:00:05'</pre>	
2		<pre>set tran isolation level read committed begin tran insert into MonAn(TenMon,LoaiMon,Gia,DonViTinh,MaNH) values (@TenMon,@LoaiMon,@Gia,@DonViTinh,@MaNH)  commit go</pre>
3	<pre>select MaMon,TenMon,TenLoai,Gia,DonViTinh from MonAn m,LoaiMonAn lm,NhaHang nh where m.LoaiMon=lm.MaLoai and m.MaNH=nh.MaNH and nh.MaNH =@maNH and lm.MaLoai=@maLoai  commit tran</pre>	

## 2.4 1141129 - Bùi Trung Tân

### 2.4.1 Các trường hợp tranh chấp

#### a. *Dirty read*

**Kịch bản :**

T1: thực hiện chuyển bàn ăn A sang B	T2: ghép bàn
Cập nhật trạng thái bàn A thành 'trống'	Lấy danh sách bàn trống để chuyển bàn. (hoặc ghép bàn, xếp bàn...)
Kiểm tra bàn B có đang trống hay không: - Nếu không còn trống (do bị giao tác khác đã thay đổi trạng thái bàn ăn này) =>Rollback: trả lại trạng thái ban 'đang sử dụng' cho bàn A - Ngược lại thì cập nhật trạng thái bàn B thành 'dang sử dụng'. =>Hoàn tất nghiệp vụ	

- ⇒ Tại thời điểm T2 đã đọc dữ liệu, được trạng thái bàn A và B đều là 'trống' và được đọc lên.
- Nếu T1 bị rollback, trạng thái bàn A bị gán lại thành 'đang sử dụng', T2 đã đọc sai dữ liệu khi đọc được bàn A.
  - Nếu T1 thành công, trạng thái bàn B bị gán thành 'đang sử dụng', vậy T2 đã đọc sai dữ liệu khi đọc được bàn B.

**Stored Procedure :**

T1	T2
<pre>create proc sp_ChuyenBan @kq int out, @mabanOLD int, @mabanNEW int, @mahd int as begin  begin tran t1 SET @kq = 1  UPDATE BanAn SET TrangThai = 0 WHERE MaBan = @mabanOLD  waitfor delay '0:0:10'</pre>	
	<pre>create proc sp_selectBanAn @trangthai int, @manh nchar(10) as begin tran IF (@trangthai = -1)     SELECT * FROM BanAn     WHERE MaNH = @manh ELSE     SELECT * FROM BanAn</pre>

	<pre> WHERE TrangThai = @trangthai AND MaNH = @manh commit tran </pre>
<pre> IF( (SELECT TrangThai FROM BanAn WHERE MaBan = @mabanNEW) = 0 ) begin EXEC dbo.sp_UpdateTrangThaiBanAn @mabanNEW, 1 UPDATE HoaDon SET MaBan = @mabanNEW WHERE MaHD = @mahd commit tran t1 end ELSE begin SET @kq = 0 rollback tran t1 end end </pre>	

### ***Nguyên nhân tranh chấp :***

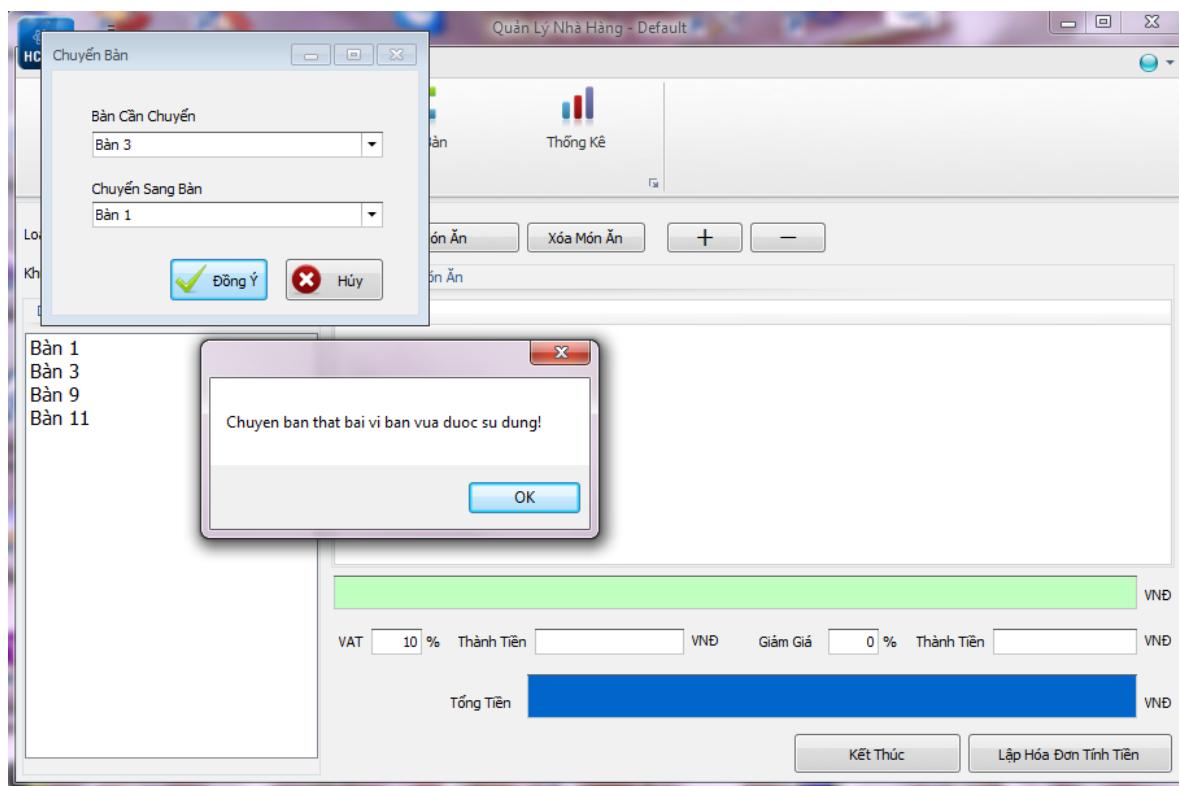
Với mức cõ lập Read Uncommitted, T2 đọc mà không xin khoá đọc. Như vậy bàn ăn A đang có khoá ghi (T1 update trạng thái) và T2 vẫn có thể đọc được dữ liệu, nên T2 có thể bị đọc sai.

### ***Cách khắc phục:***

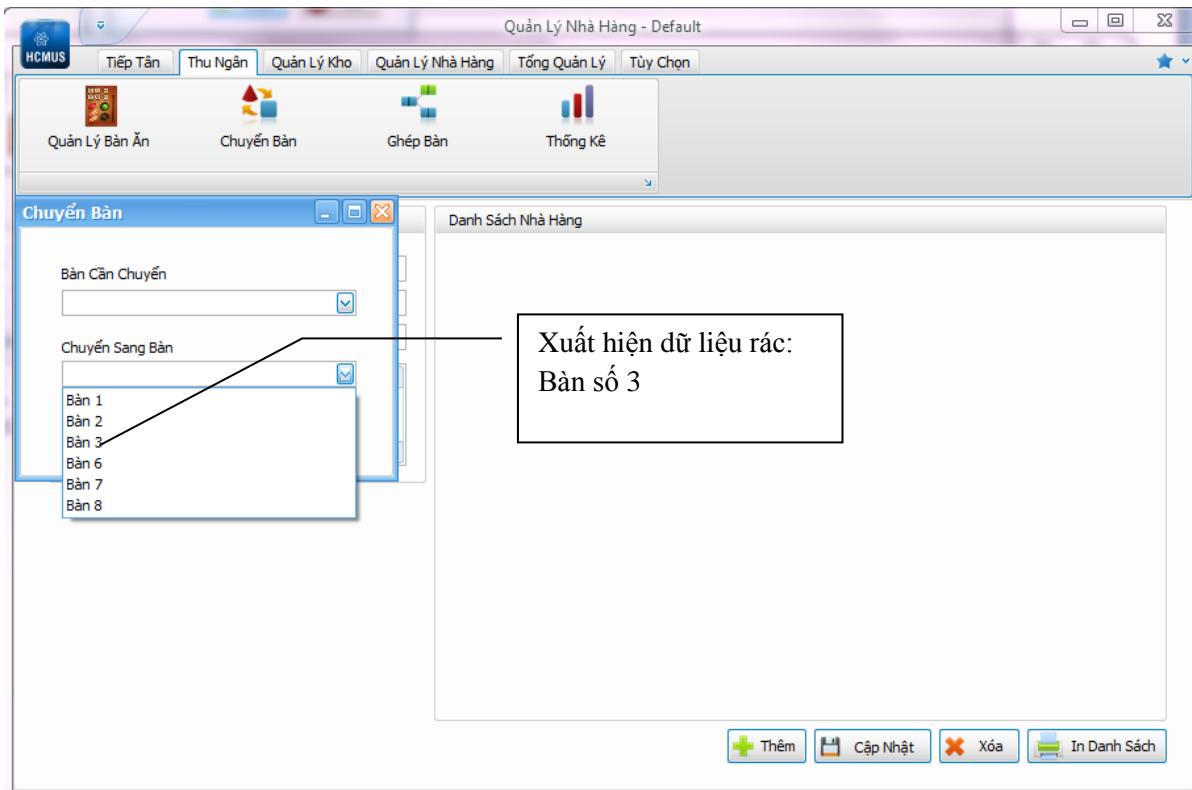
Thiết lập mức cõ lập Read Committed cho T1 và T2. T2 phải xin khoá đọc trên bàn A và bàn B, như vậy T2 phải đợi T1 hoàn tất.

**Giao diện chương trình bị Dirty Read:**

Giao tác T1



Giao tác T2



**b. Lost update**

**c. Unrepeatable read**

**Kịch bản :**

T1: thực hiện thêm một món ăn A vào hoá đơn	T2: xoá món ăn A
Kiểm tra xem món ăn A có tồn tại hay không (đọc lần 1)	
	Xoá món ăn A
Nếu món ăn A có tồn tại (kiến quả của đọc lần 1) thì <ul style="list-style-type: none"> <li>- Đọc lần 2 để lấy ra giá của món ăn A</li> <li>- Thêm thông tin món ăn vào bảng chi tiết hoá đơn</li> </ul>	

- ⇒ Tại thời điểm T1 kiểm tra xem món ăn có tồn tại hay không (đọc lần 1).
  - Giả sử lần đọc đầu tiên trả ra kết quả là có, các câu lệnh bên trong hàm IF được thực hiện
  - Trước khi T1 đọc lần thứ 2, T2 đã xoá món ăn đó, làm cho T1 không thể đọc lại món ăn đó để lấy ra Giá của món ăn.

**Stored Procedure :**

T1	T2
<pre> create proc sp_insertCTHD @mahd int, @mamon int, @soluong int as begin tran IF( (select COUNT(*) from MonAn where MaMon = @mamon) = 1 ) begin waitfor delay '0:0:10' </pre>	
	DELETE MonAn WHERE MaMon = @mamon

**Nguyên nhân tranh chấp:**

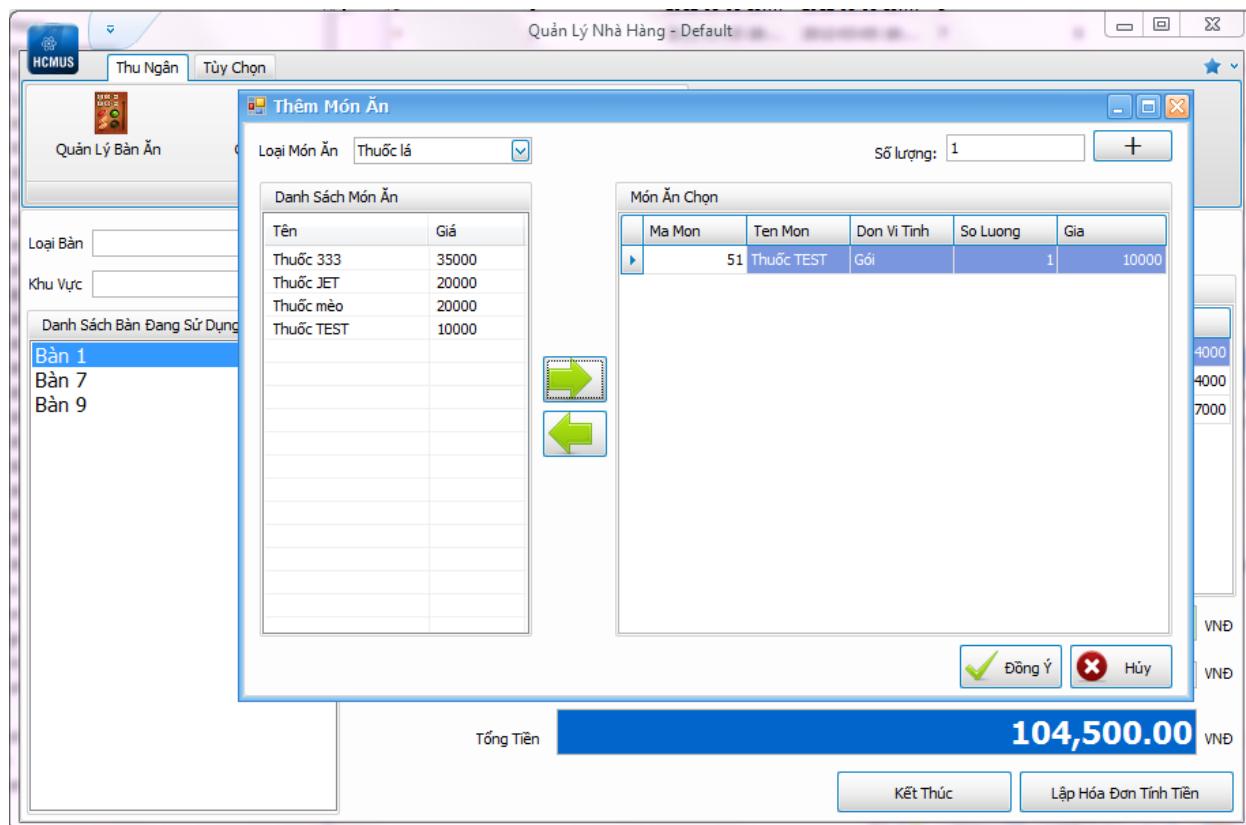
Khi T1 được thiết lập mức cô lập là Read Uncommitted hoặc Read Committed, sau lần đọc thứ nhất, khoá đọc được giải phóng trên món ăn A ngay sau khi đọc xong. Như vậy giao tác T2 có thể thay đổi thông tin món ăn A (xoá, thay đổi điều kiện lọc), làm cho T1 không thể đọc lại món ăn A ở lần đọc thứ hai.

**Cách khắc phục:**

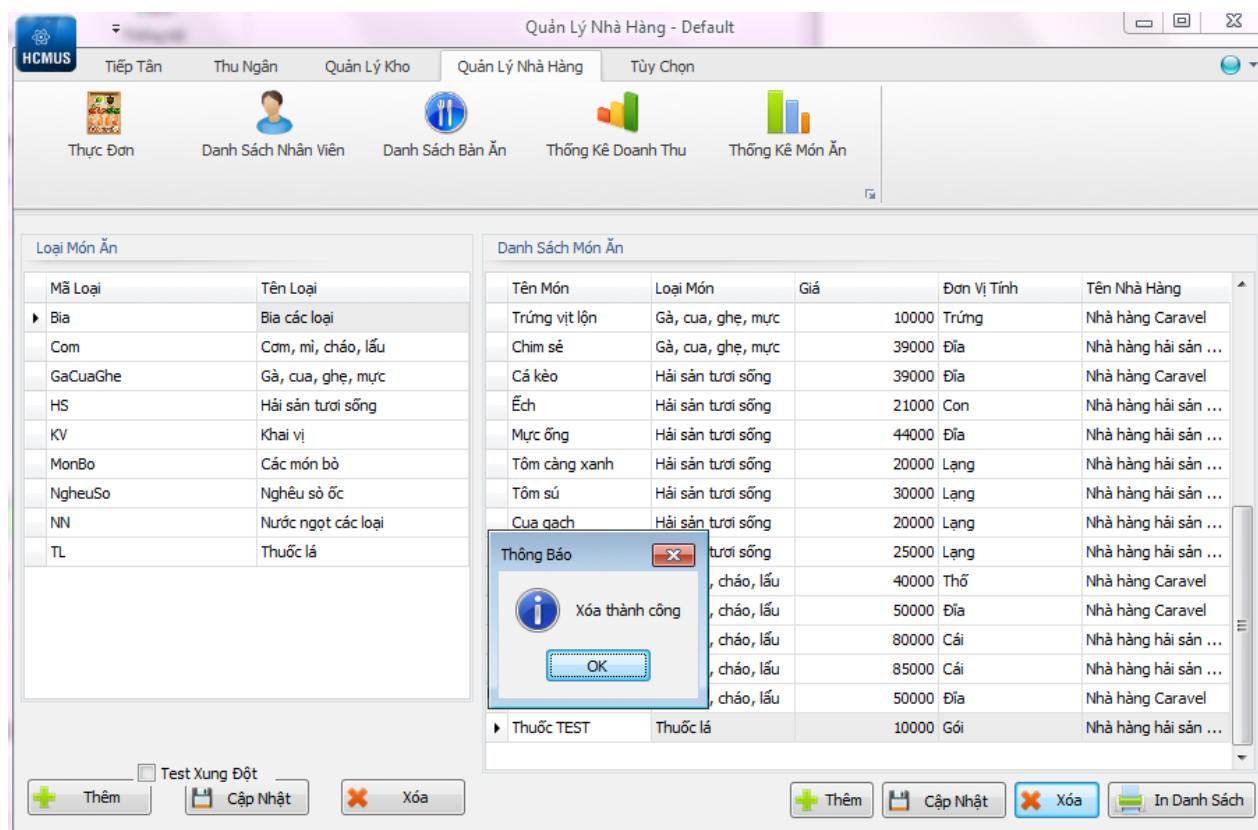
Thiết lập mức cô lập Repeatable Read cho T1. Như vậy ở lần đọc thứ nhất, khoá đọc được giữ trên món ăn A đến khi T1 kết thúc, T2 phải đợi T1 hoàn tất thì mới có thể thực hiện câu lệnh xoá món ăn.

**Giao diện chương trình bị Unrepeatable Read:**

Giao tác T1



Giao tác T2



*d. Phantom**e. Deadlock***Kịch bản :**

T1: cập nhật số lượng cho chi tiết hoá đơn	T2: cập nhật số lượng cho chi tiết hoá đơn
Kiểm tra xem món ăn A cần cập nhật số lượng đã có trong chi tiết hoá đơn chưa.	
	Kiểm tra xem món ăn A cần cập nhật số lượng đã có trong chi tiết hoá đơn chưa.
Nếu có rồi thì cập nhật lại số lượng cho món ăn A	
	Nếu có rồi thì cập nhật lại số lượng cho món ăn A

⇒ Để không bị Unrepeatable Read, mức độ lặp được sử dụng ở T1 và T2 là Repeatable Read. Như vậy xuất hiện Conversion Deadlock khi shared lock được cấp và giữ đến cuối giao tác.

**Stored Procedure:**

T1	T2
<pre>create proc sp_UpdateSoLuongCTHD @mahd int, @mamon int, @soluong int as begin tran SET TRAN ISOLATION LEVEL REPEATABLE READ  IF( (SELECT COUNT(*) FROM ChiTietHoaDon       WHERE MaHD = @mahd       AND MaMon = @mamon) = 1 ) begin     waitfor delay '0:0:7'</pre>	
	<pre>create proc sp_UpdateSoLuongCTHD @mahd int, @mamon int, @soluong int as begin tran SET TRAN ISOLATION LEVEL REPEATABLE READ  IF( (SELECT COUNT(*) FROM ChiTietHoaDon       WHERE MaHD = @mahd       AND MaMon = @mamon) = 1 ) begin     waitfor delay '0:0:7'</pre>
<pre>UPDATE ChiTietHoaDon SET SoLuong = @soluong WHERE MaHD = @mahd AND MaMon = @mamon End commit tran</pre>	
	<pre>UPDATE ChiTietHoaDon SET SoLuong = @soluong WHERE MaHD = @mahd AND MaMon =</pre>

```
@mamon
End
commit tran
```

### **Nguyên nhân tranh chấp:**

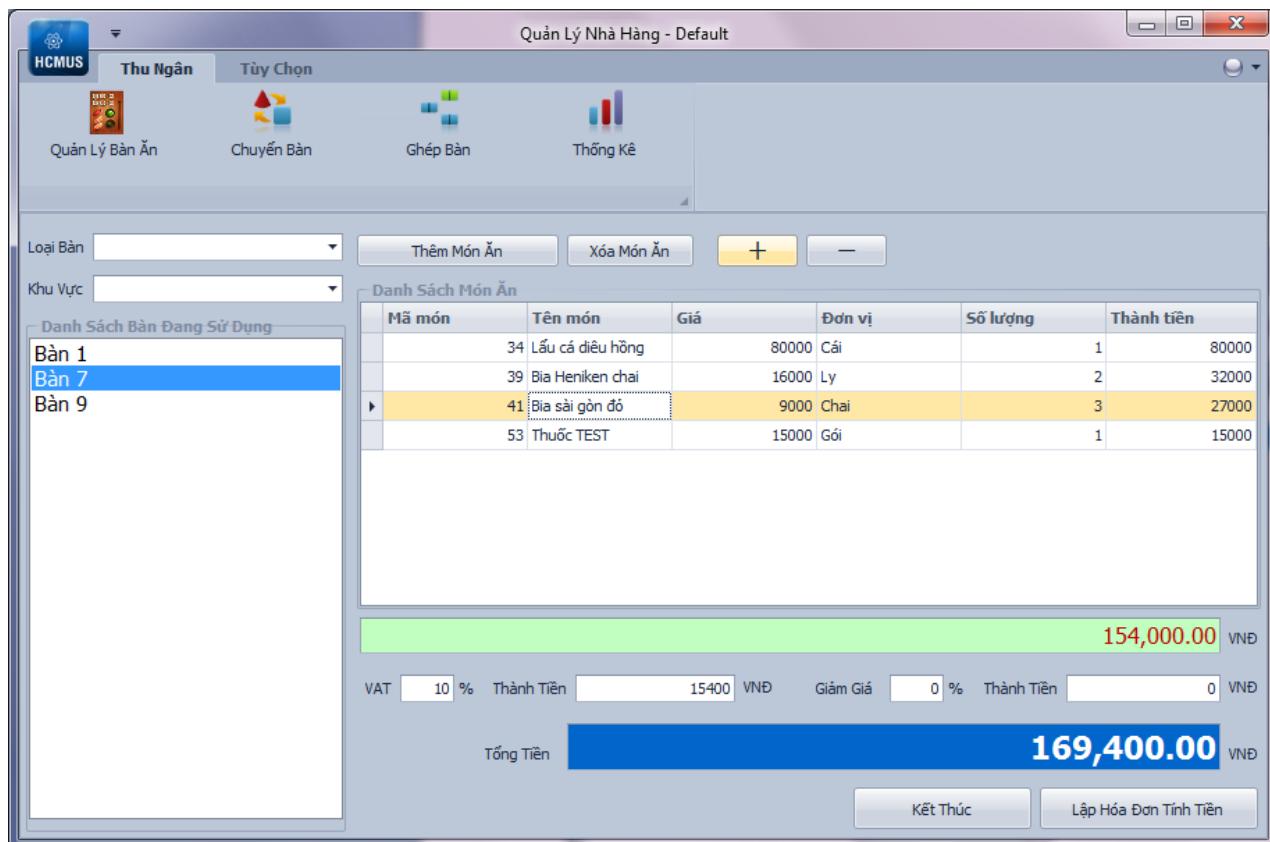
- ⇒ Để cập nhật số lượng món ăn trong chi tiết hoá đơn, cần đảm bảo trong chi tiết hoá đơn có tồn tại món ăn đó, nên cần thực hiện kiểm tra
- ⇒ Để đảm bảo sau khi kiểm tra, món ăn không bị thay đổi thông tin (ví dụ: xoá) thì cần thiết lập mức độ Repeatable Read. Như vậy ở khoá đọc được cấp và giữ trên món ăn A đến hết giao dịch.
- ⇒ T1 cấp shared lock trên món ăn A và giữ đến hết giao dịch. T2 cấp shared lock trên món ăn A và giữ đến hết giao dịch. T1 cần cập nhật giữ liệu trên món ăn A, nên phải chờ T2 kết thúc. T2 cũng cần cập nhật giữ liệu trên món ăn A và phải chờ T1 kết thúc.
- ⇒ Xuất hiện Conversion Deadlock.

### **Cách khắc phục:**

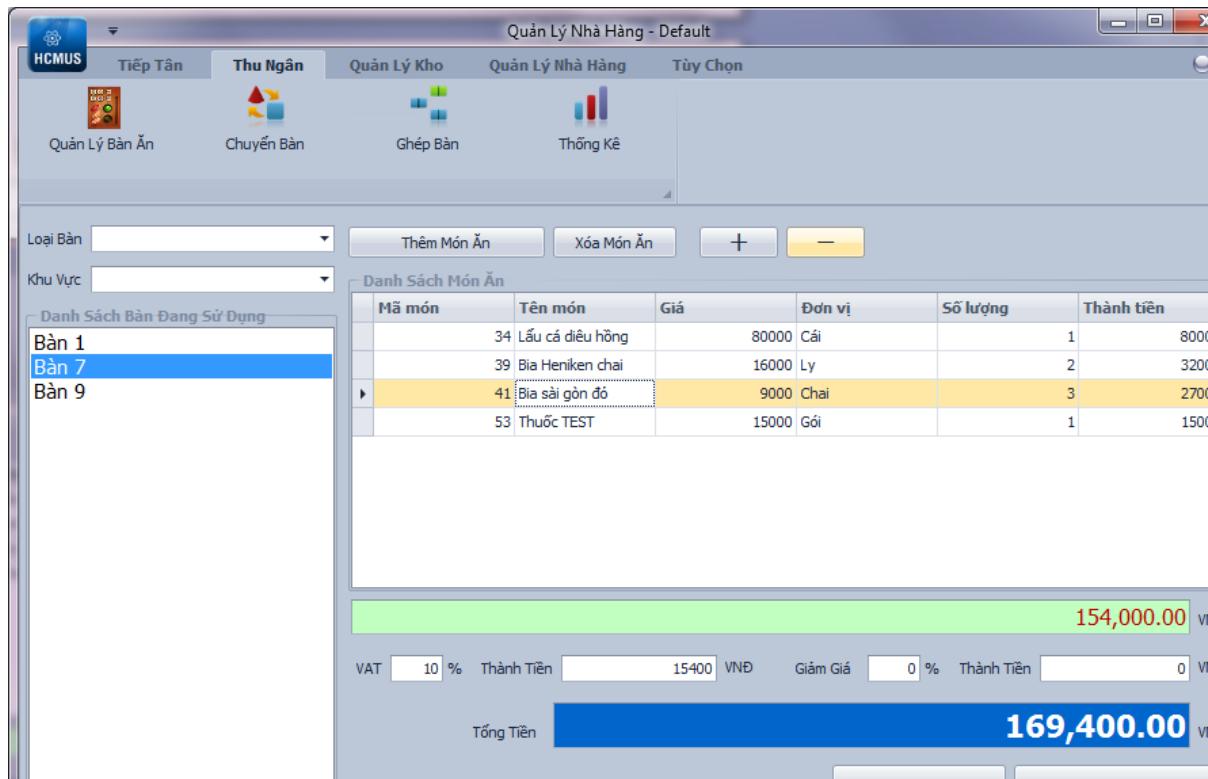
Thiết lập timeout (SET LOCK\_TIMEOUT 20000) để chỉ có một giao dịch được thực thi, giao dịch còn lại bị huỷ.

### **Giao diện chương trình bị Conversion Deadlock:**

Giao dịch T1: (tăng số lượng món ăn lên 1)



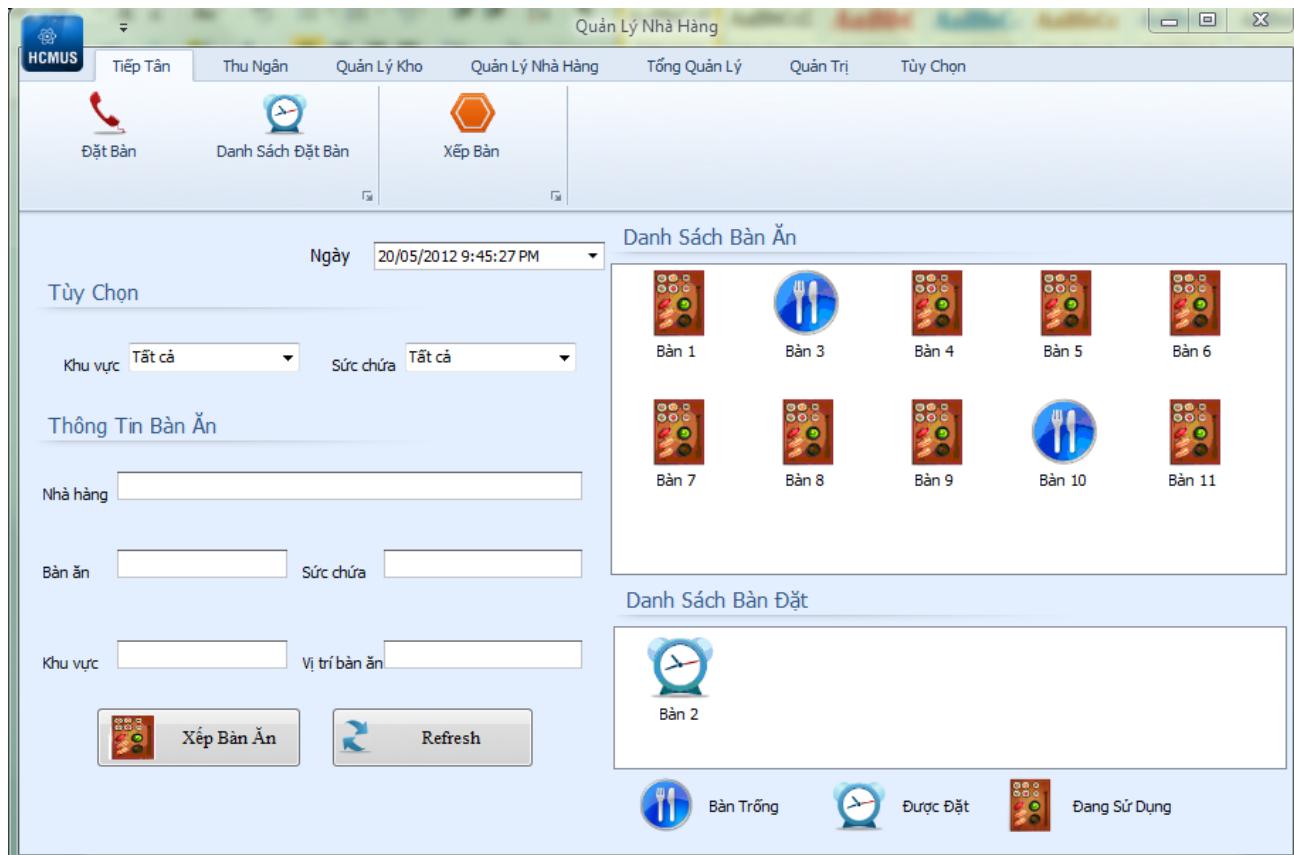
Giao dịch T2: (giảm số lượng món ăn đi 1)



### 3. Các chức năng trong đề tài

#### 3.1 Nhân Viên Tiếp Tân

##### 3.1.1 Xếp bàn ăn cho khách



### 3.1.2 Đặt bàn

**Quản Lý Nhà Hàng**

HCMUS Tiệp Tân Thu Ngân Quản Lý Kho Quản Lý Nhà Hàng Tổng Quản Lý Quản Trị Tùy Chọn

**Đặt Bàn** **Danh Sách Đặt Bàn** **Xếp Bàn**

**Danh Sách Bàn**

Bàn	Khu Vực	Vị Trí	Sức Chứa
1	Sảnh – tầng trệt	Giữa đại sảnh của nh...	20
2	Góc – tầng trệt	Nằm ở góc	15
3	Sân thượng	Sân thượng	10
4	Lầu 1	Lầu 1	10
5	Sảnh – tầng trệt	Giữa đại sảnh của nh...	10
6	Sảnh – tầng trệt	Giữa đại sảnh của nh...	15
7	Sát tường - tầng trệt	Sát tường, nhìn ra bê...	20
8	Lầu 2	Lầu 2	30
9	Góc – tầng trệt	Nằm ở góc	15
10	Lầu 1	Lầu 1	20
11	Sảnh – tầng trệt	Giữa đại sảnh của nh...	30

**Thông Tin Đặt Bàn**

Tên Khách Hàng: Ngoc

Số Điện Thoại:

Thời Gian Đến:

**Danh Sách Bàn Đặt**

Bàn	Tên Khách H...	Sdt	Thời Gian Đến	Vị Trí

**Đặt Bàn** **Hủy**

### 3.1.3 Danh sách đặt bàn

Quản Lý Nhà Hàng

Tùy Chọn

Mã Bàn	All	Tình Trạng	All
Khu Vực	All	Sức Chứa	All

Danh Sách Đặt Bàn

Bàn	Tên Khách Hàng	Tên Khu Vực	Vị Trí	Sức Chứa	Thời Gian Đặt	Thời Gian Đến	Tình Trạng
2	Vũ	Góc – tăng trệt	Nằm ở góc	15	20/05/2012	20/05/2012	Chưa đến ăn

Đặt Bàn Thay Đổi Xóa Refresh

FrmCapNhatBanDat

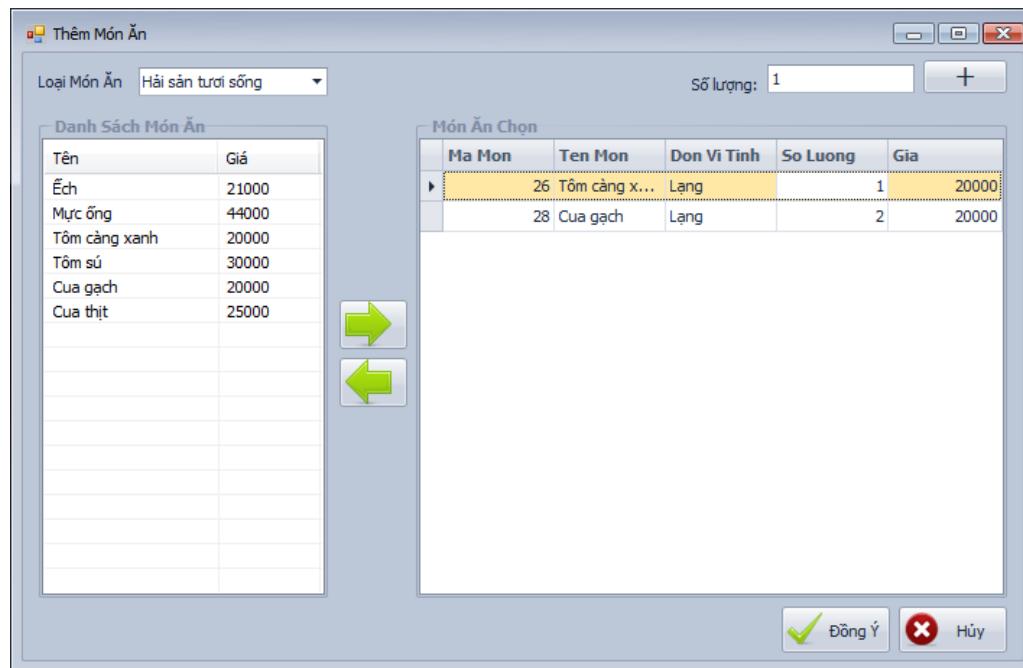
Bàn Ăn	Khách Hàng
2	Vũ
Hiện Trạng	Thời Gian Đến
Chưa đến ăn	20/05/2012

Đồng Ý Hủy

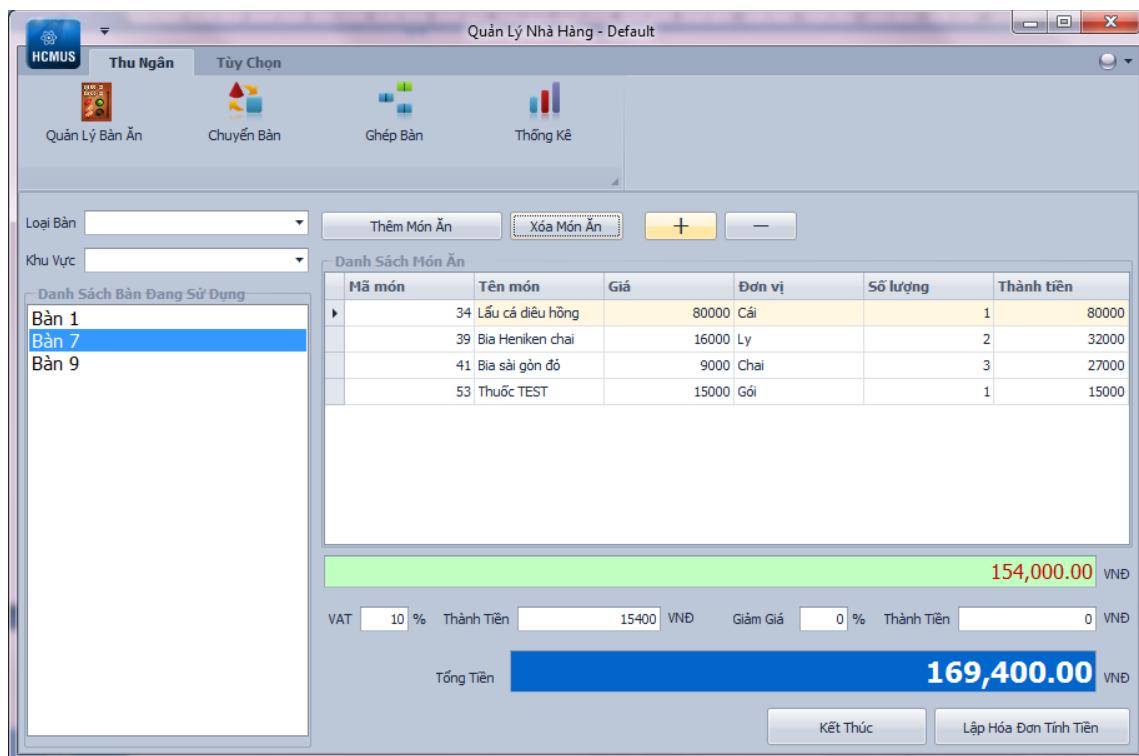
## 3.2 Nhân Viên Thu Ngân

Đăng nhập với loại tài khoản là Thu ngân hoặc Quản lý nhà hàng, sử dụng Ribbon Thu Ngân

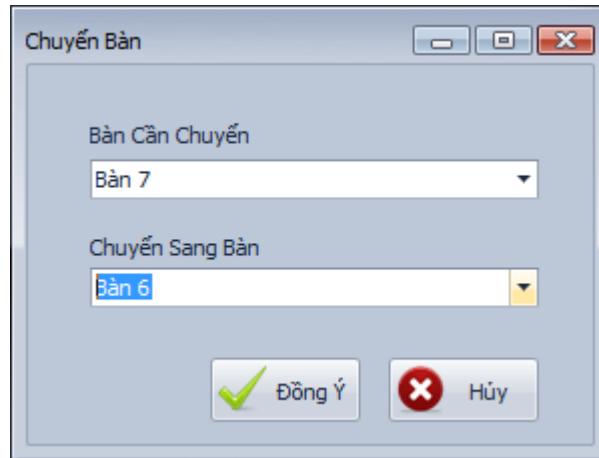
### 3.2.1 Thêm món ăn



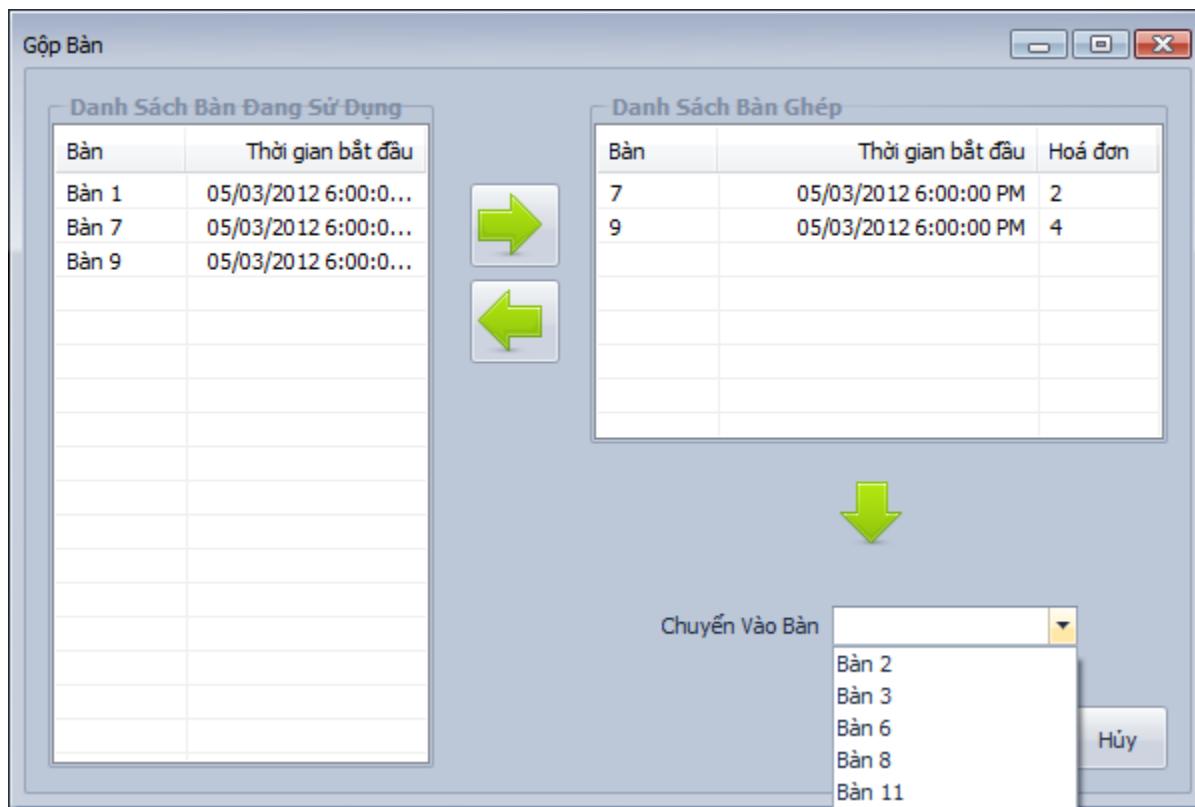
### 3.2.2 Xoá món ăn, cập nhật số lượng món ăn



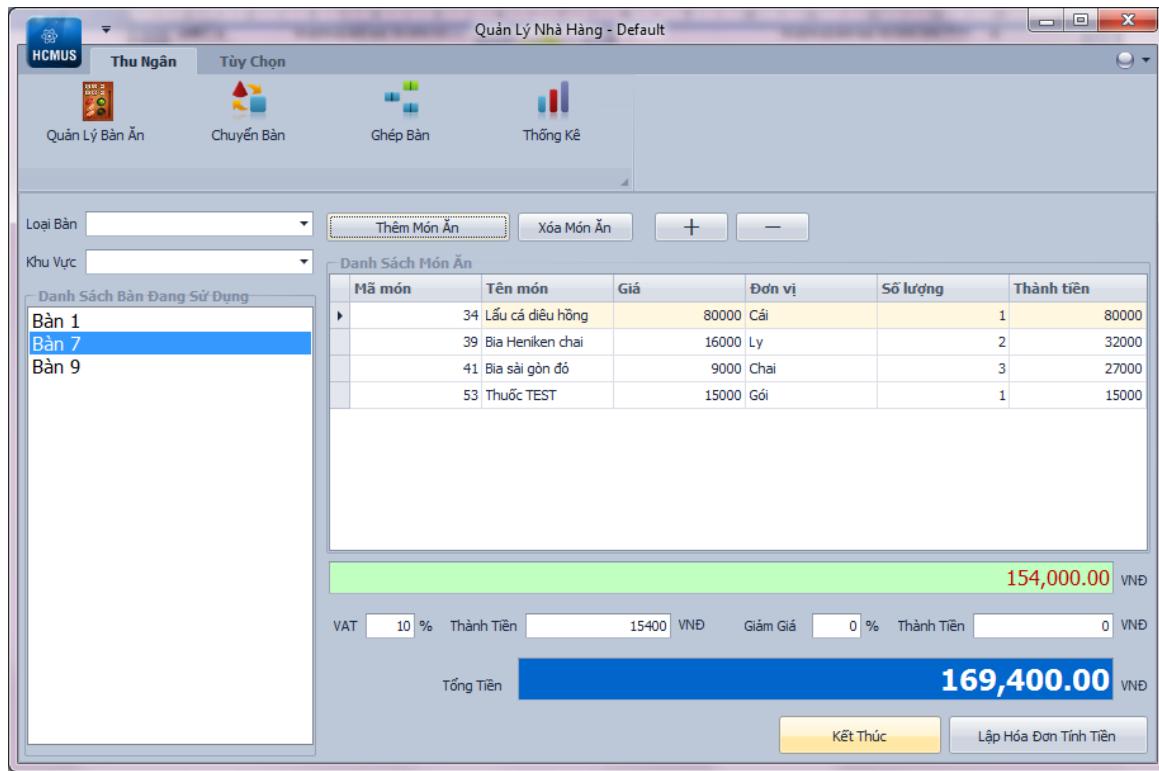
### 3.2.3 Chuyển bàn



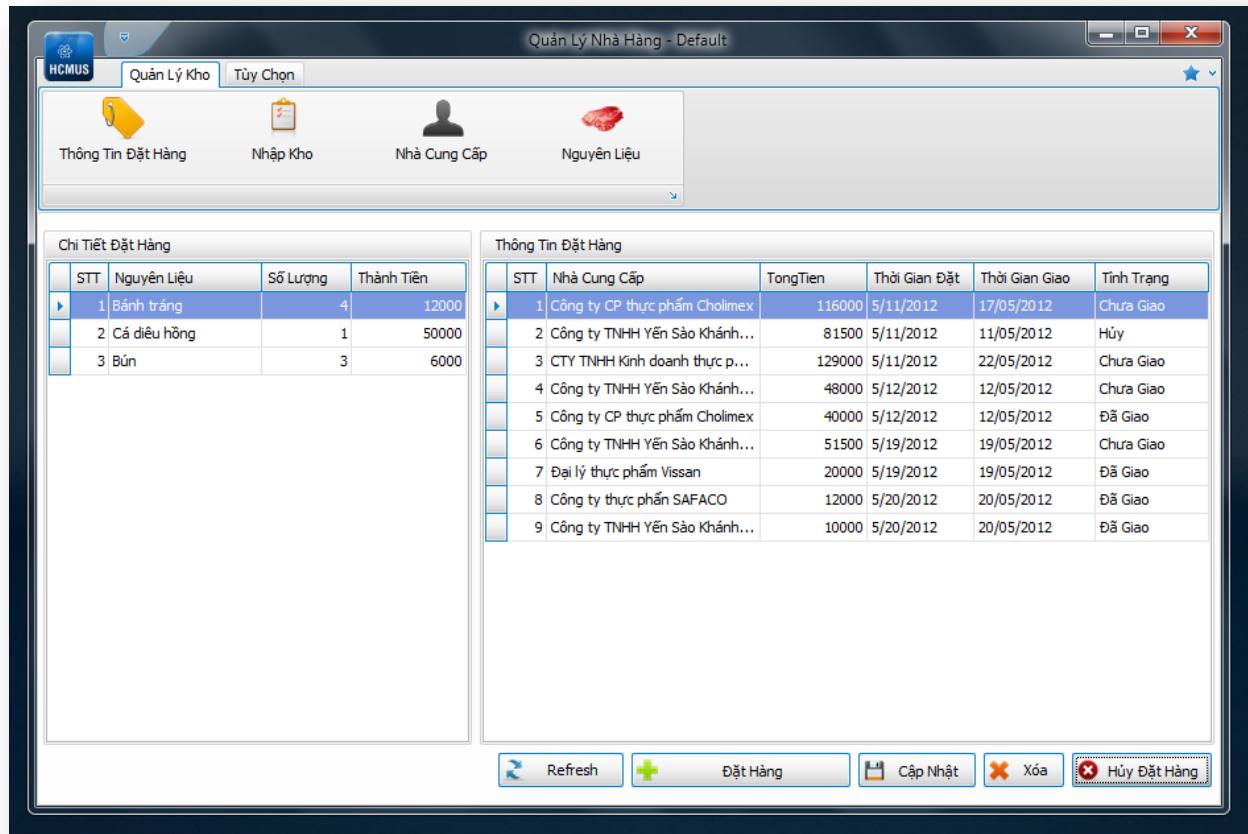
### 3.2.4 Ghép bàn



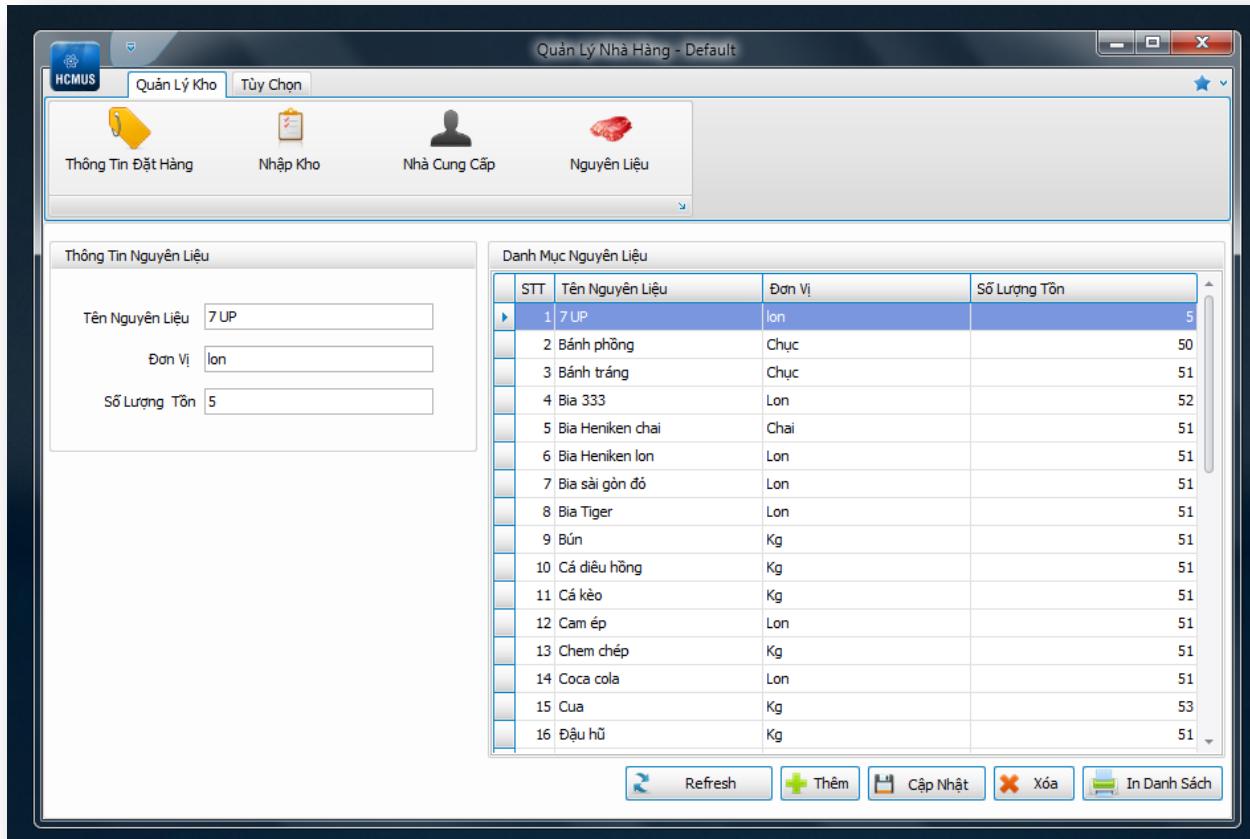
### 3.2.5 Kết thúc sử dụng bàn



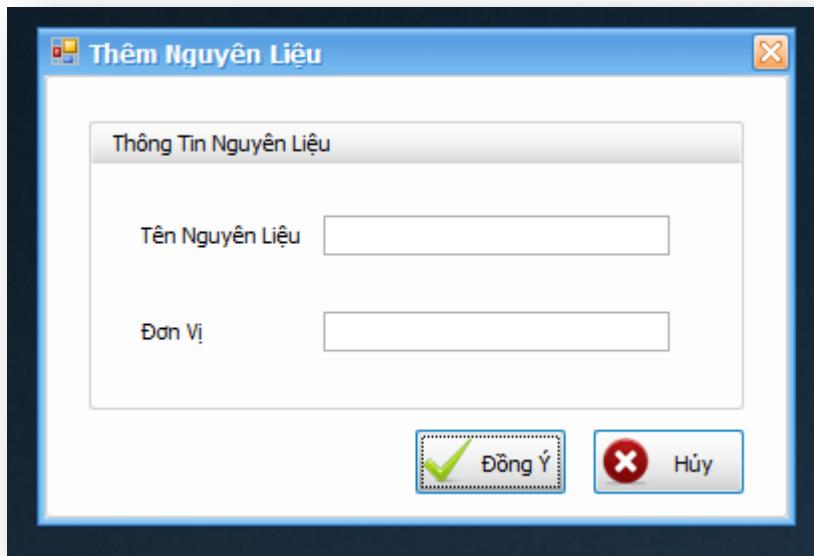
### 3.3 Nhân Viên Quản Lý Kho



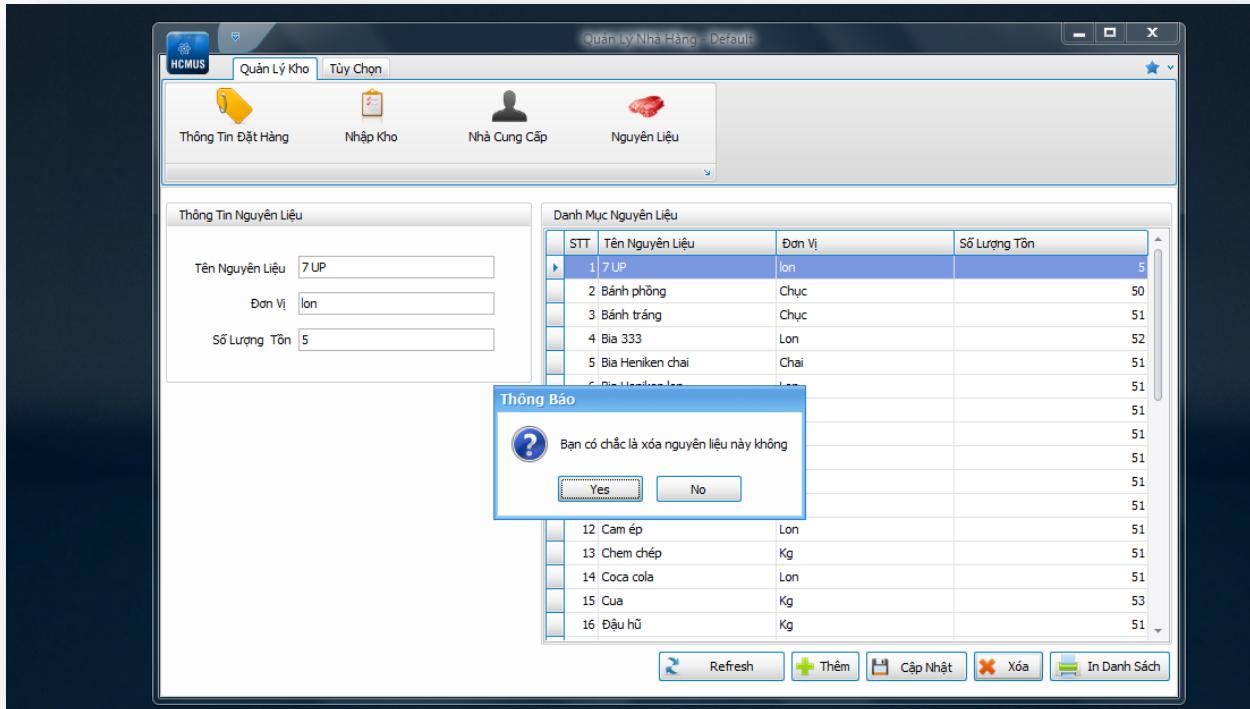
### 3.3.1. Quản Lý Danh Sách Nguyên Liệu



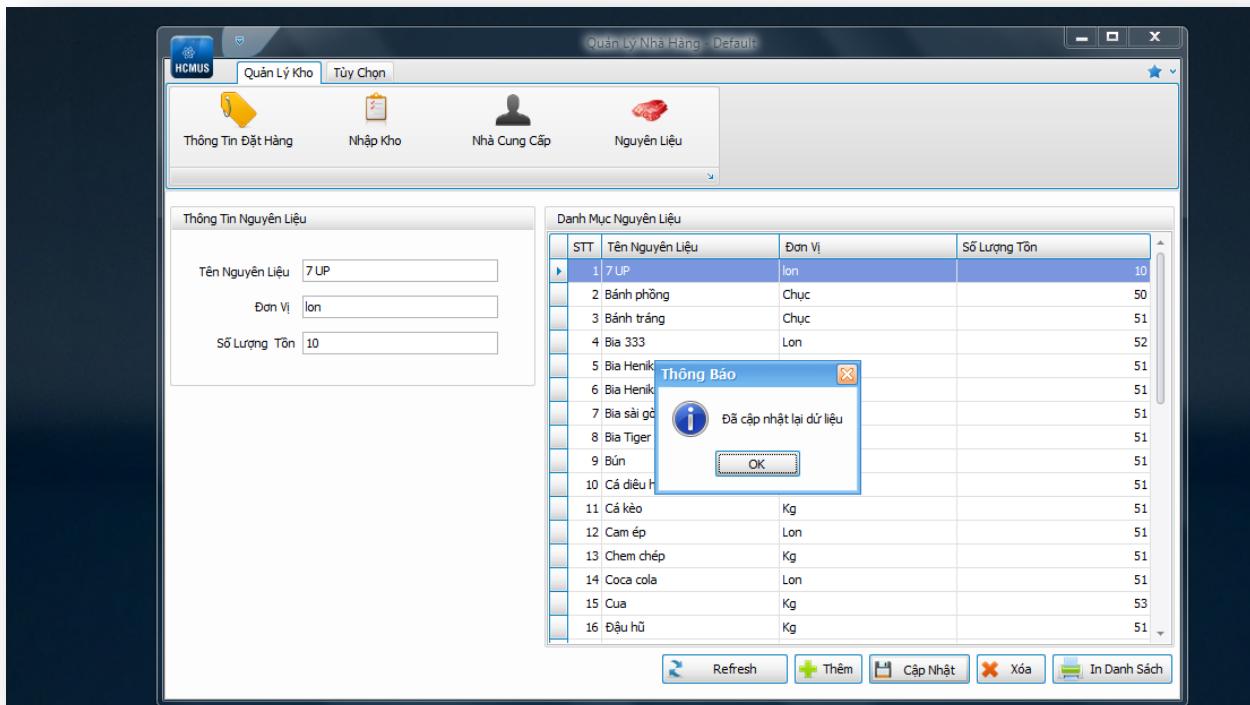
#### 3.3.1.1. Thêm Nguyên Liệu



### 3.3.1.2. Xóa Nguyên Liệu



### 3.3.1.3. Cập Nhật Nguyên Liệu



### 3.3.1.4.In Danh Sách Nguyên Liệu

STT	Tên Nguyên Liệu	Đơn Vị	Số Lượng Tồn
1	7 UP	lon	5
2	Bánh phồng	Chục	50
3	Bánh tráng	Chục	51
4	Bia 333	Lon	52
5	Bia Heniken chai	Chai	51
6	Bia Heniken lon	Lon	51
7	Bia sài gòn đỏ	Lon	51
8	Bia Tiger	Lon	51
9	Bún	Kg	51
10	Cá diêu hồng	Kg	51
11	Cá kèo	Kg	51
12	Cam ép	Lon	51
13	Chem chép	Kg	51
14	Coca cola	Lon	51
15	Cua	Kg	53
16	Đậu hũ	Kg	51
17	Đậu phộng	Kg	51
18	Éch	Con	20
19	Gạo	Kg	51

Page 1 of 1 | 100% | Zoom: + -

### 3.3.2.Quản Lý Danh Sách Nhà Cung Cấp

#### 3.3.2.1.Thêm Nhà Cung Cấp Mới

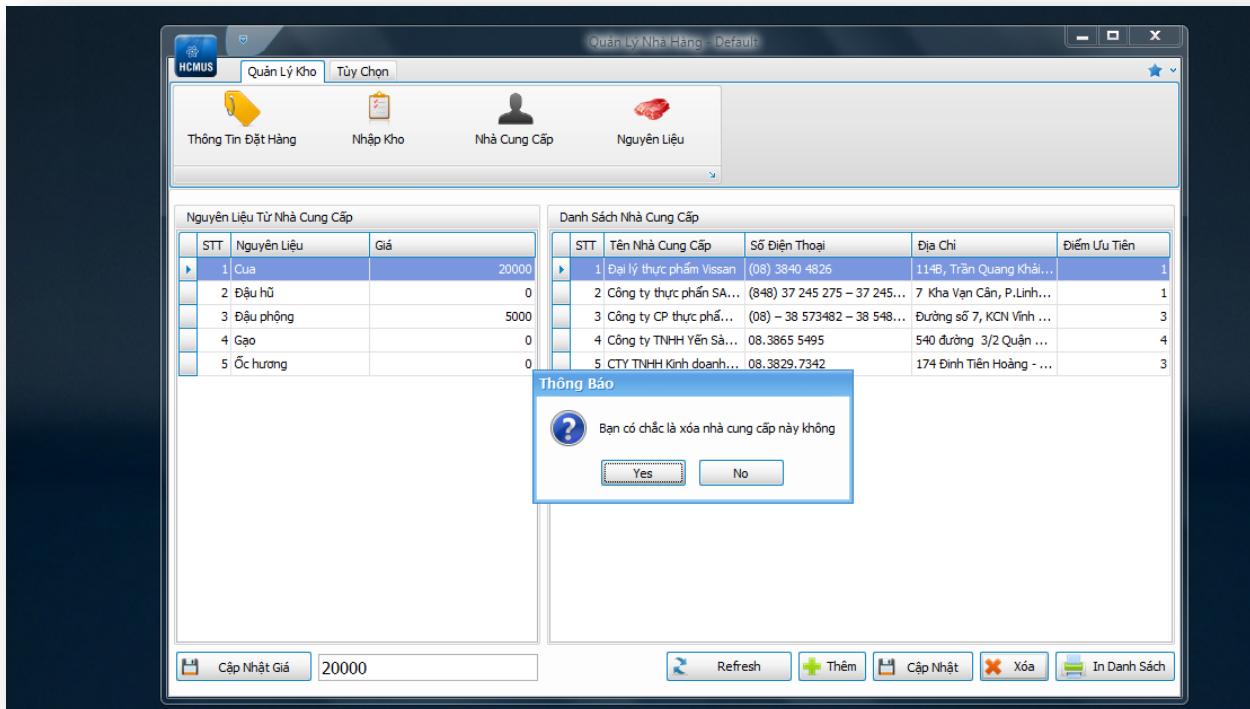
STT	Nguyên Liệu
1	7 UP
2	Bánh phồng
3	Bia 333
4	Bia Heniken chai
5	Bún
6	Cá diêu hồng
7	Cá kèo
8	Cam ép
9	Chem chép
10	Coca cola
11	Cua
12	Đậu hũ
13	Đậu phộng
14	Éch
15	Gạo
16	Hành, ngò, xả...

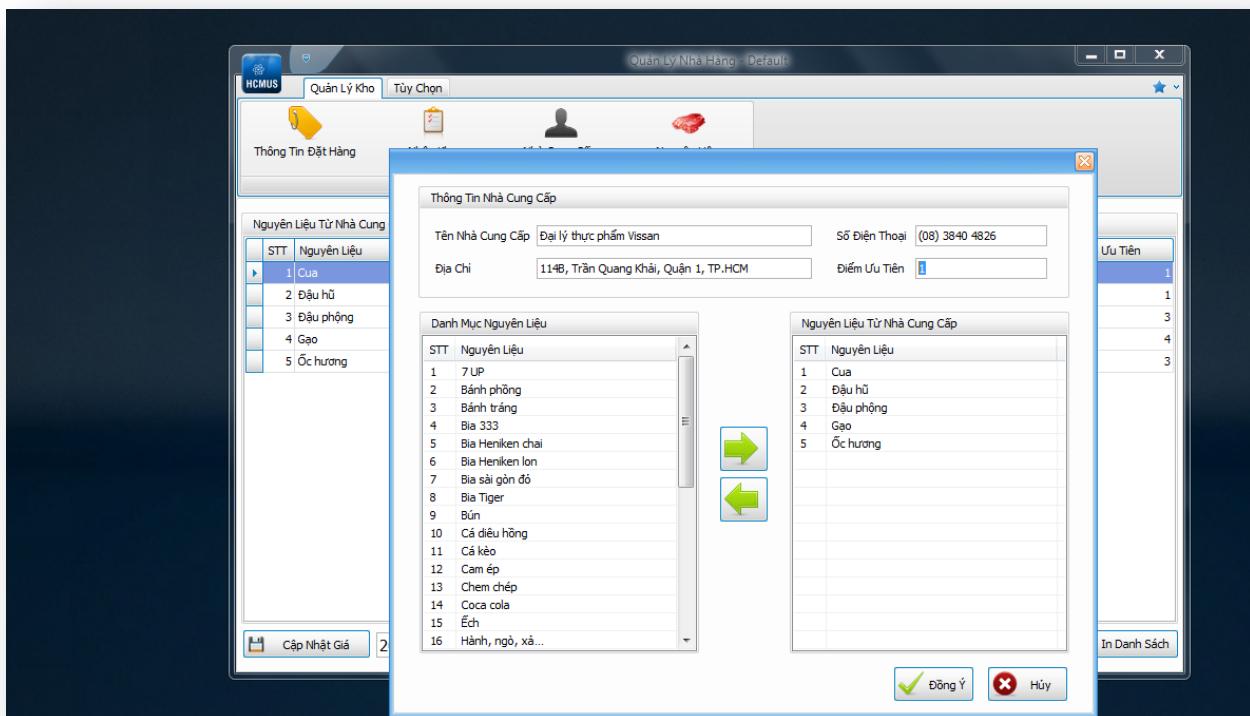
STT	Nguyên Liệu
1	Bánh tráng
2	Bia Heniken lon
3	Bia sài gòn đỏ
4	Bia Tiger

Đồng Ý     Hủy  
 Cập Nhật Giá    20000     Refresh     Thêm     Cập Nhật     Xóa     In Danh Sách

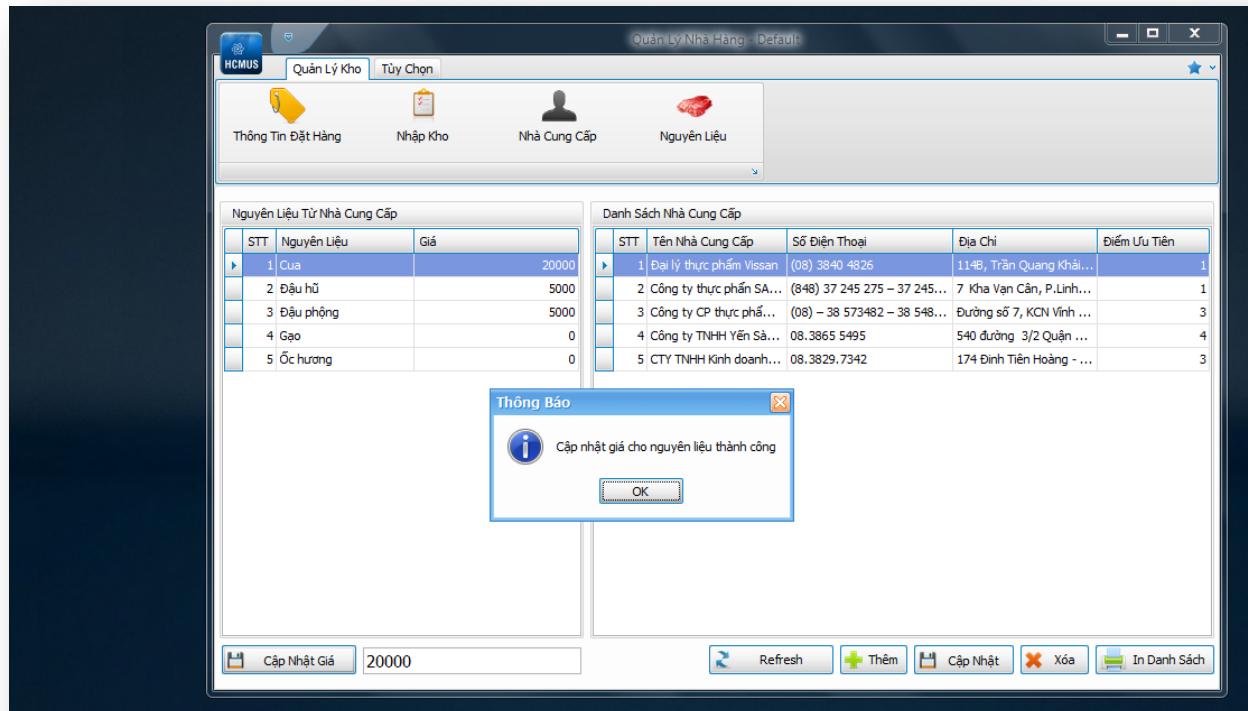
### 3.3.2.2.Xóa Nhà Cung Cấp



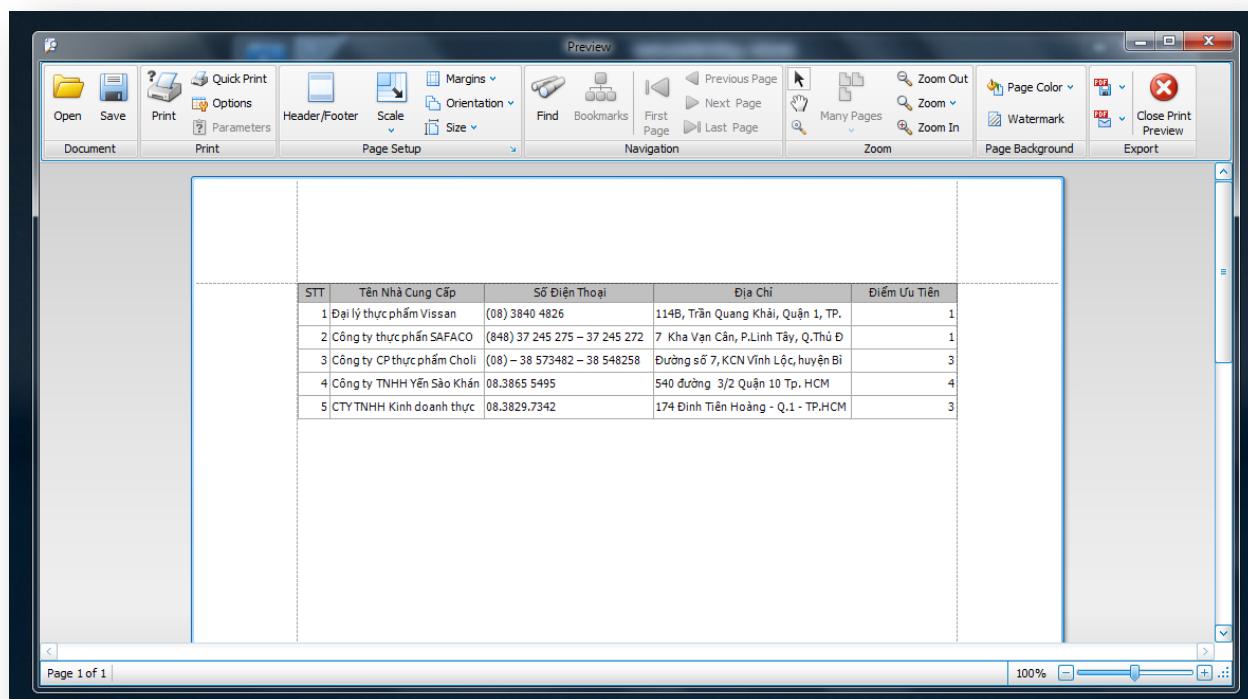
### 3.3.2.3.Cập Nhật Nhà Cung Cấp



### 3.3.2.4.Cập Nhật Giá Nguyên Liệu Nhà Cung Cấp



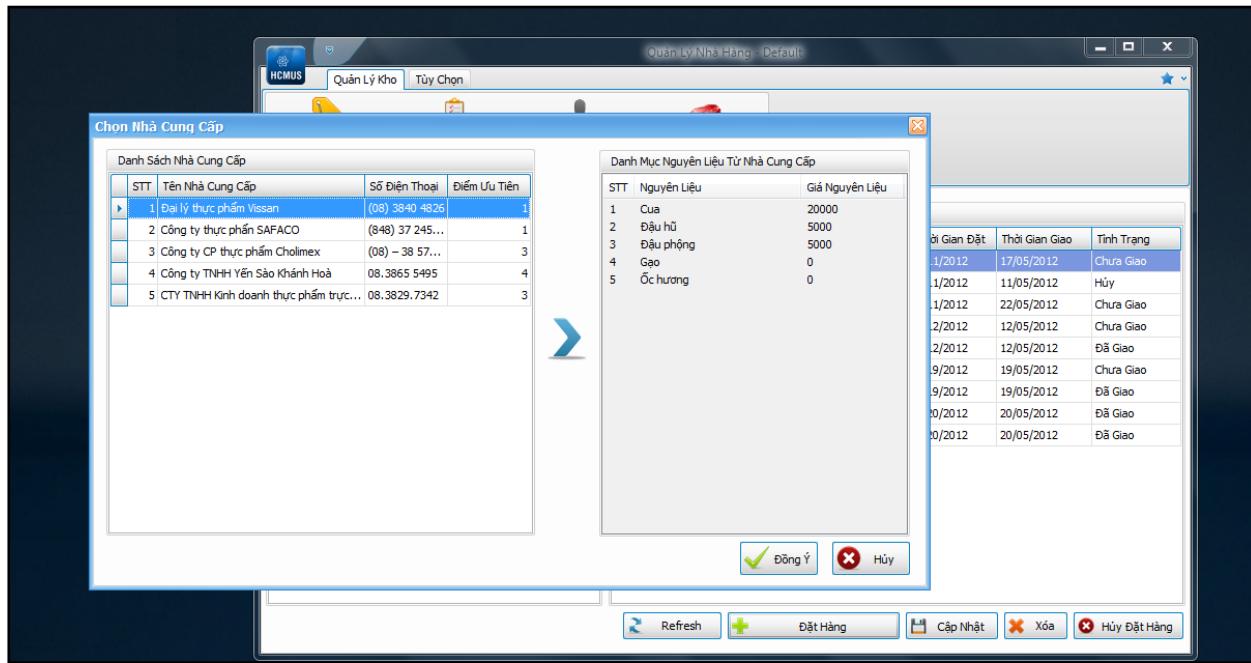
### 3.3.2.5.In Danh Sách Nhà Cung Cấp



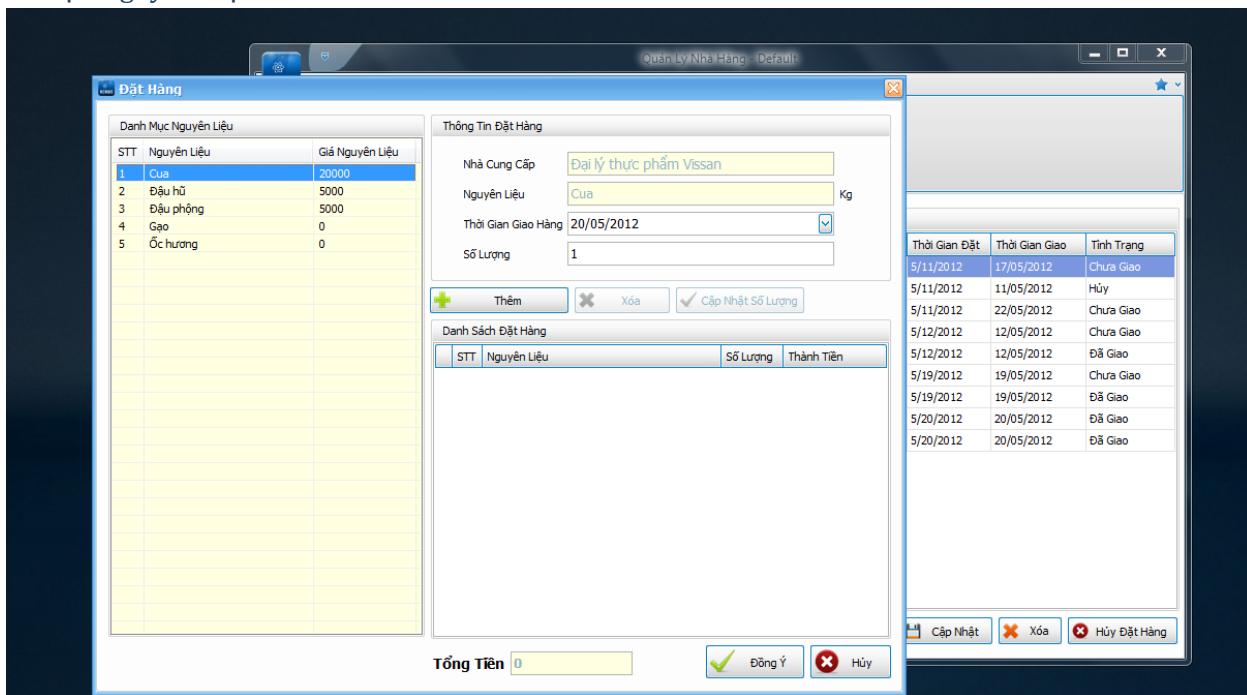
### 3.3.3. Quản Lý Thông Tin Đặt Hàng

#### 3.3.3.1. Thêm Đơn Đặt Hàng Mới

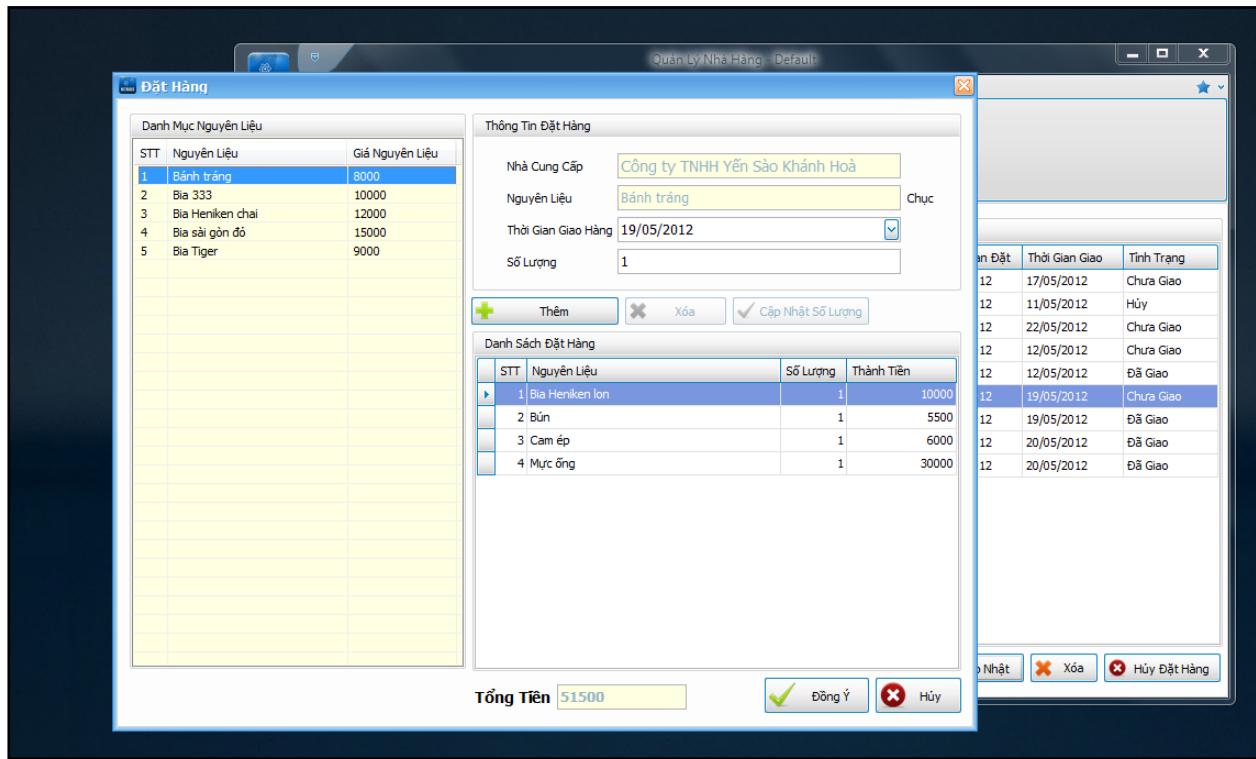
a. Chọn Nhà Cung Cấp



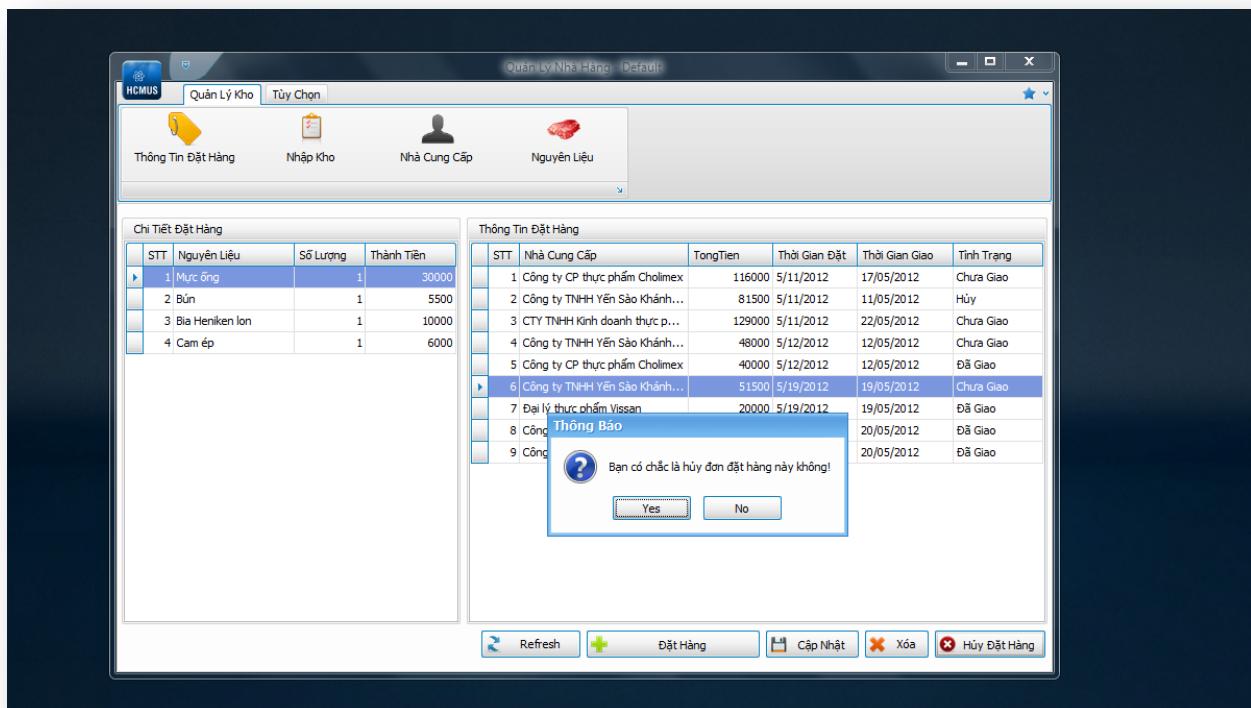
b. Chọn Nguyên Liệu



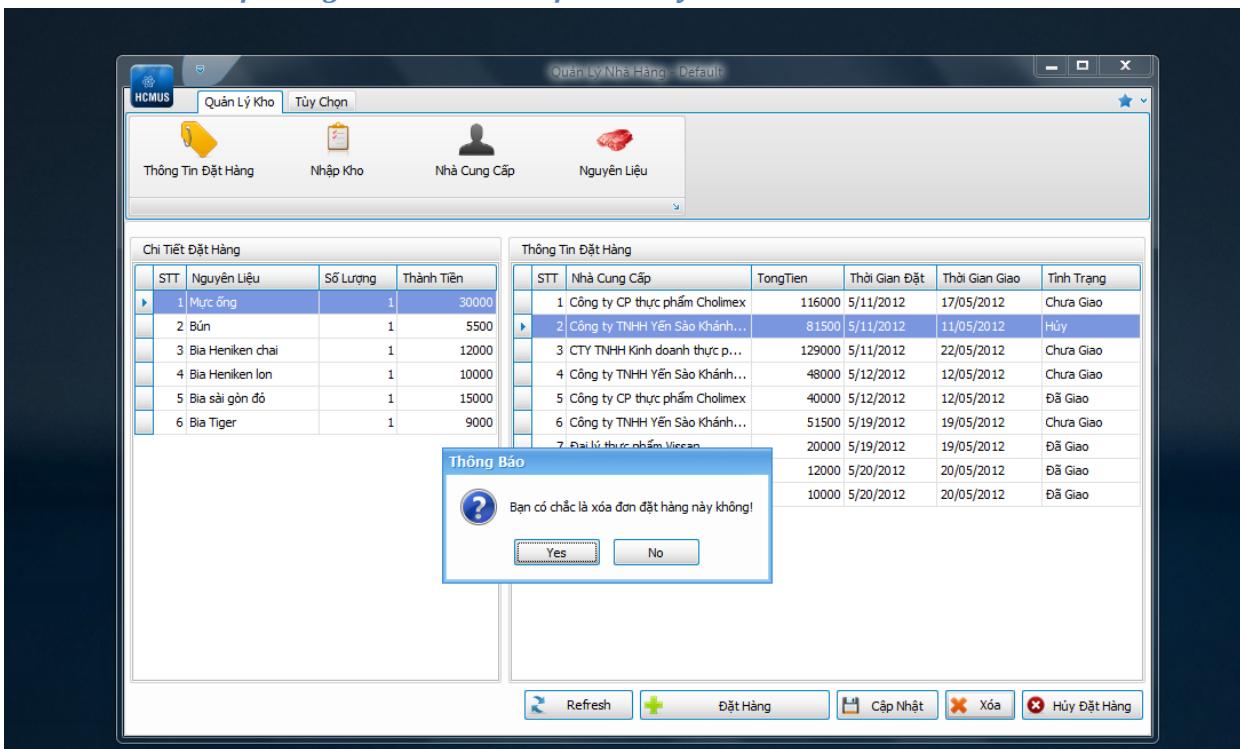
### 3.3.3.2.Cập Nhật Thông Tin Đơn Đặt Hàng Chưa Giao



### 3.3.3.4.Hủy Đơn Đặt Hàng Chưa Giao

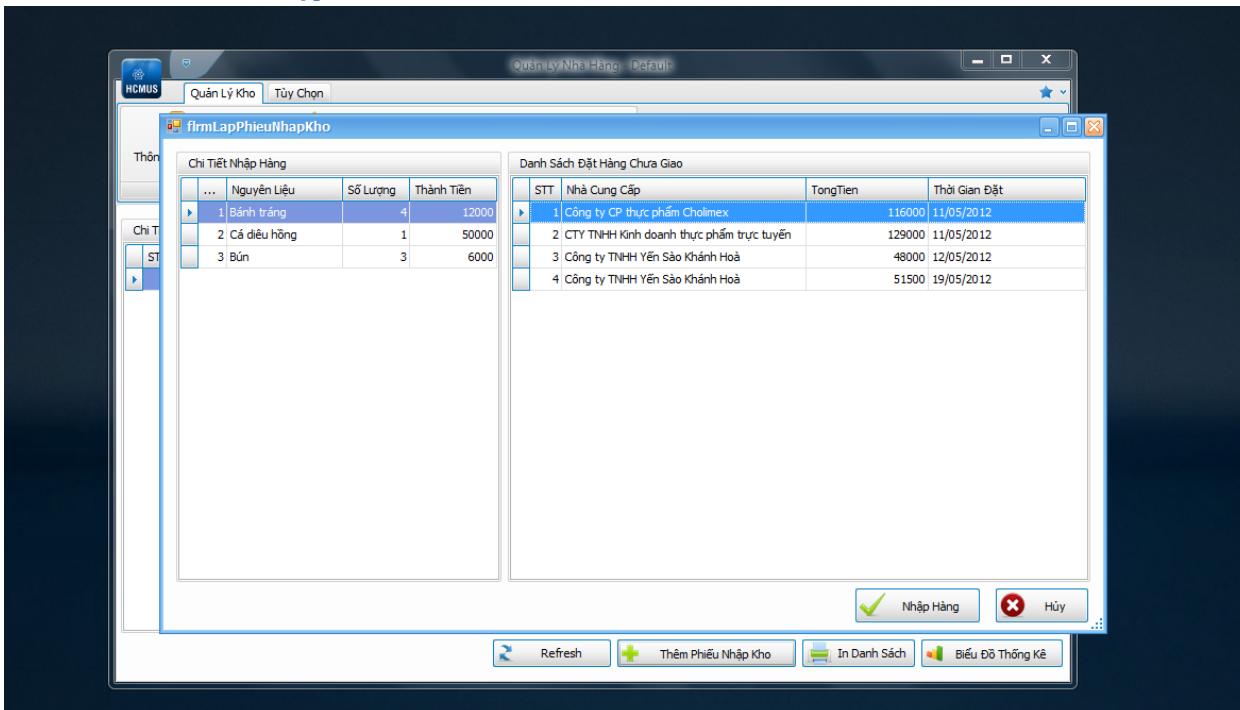


### 3.3.3.5.Xóa Đơn Đặt Hàng “Chưa Giao” Hoặc Đã “Hủy”

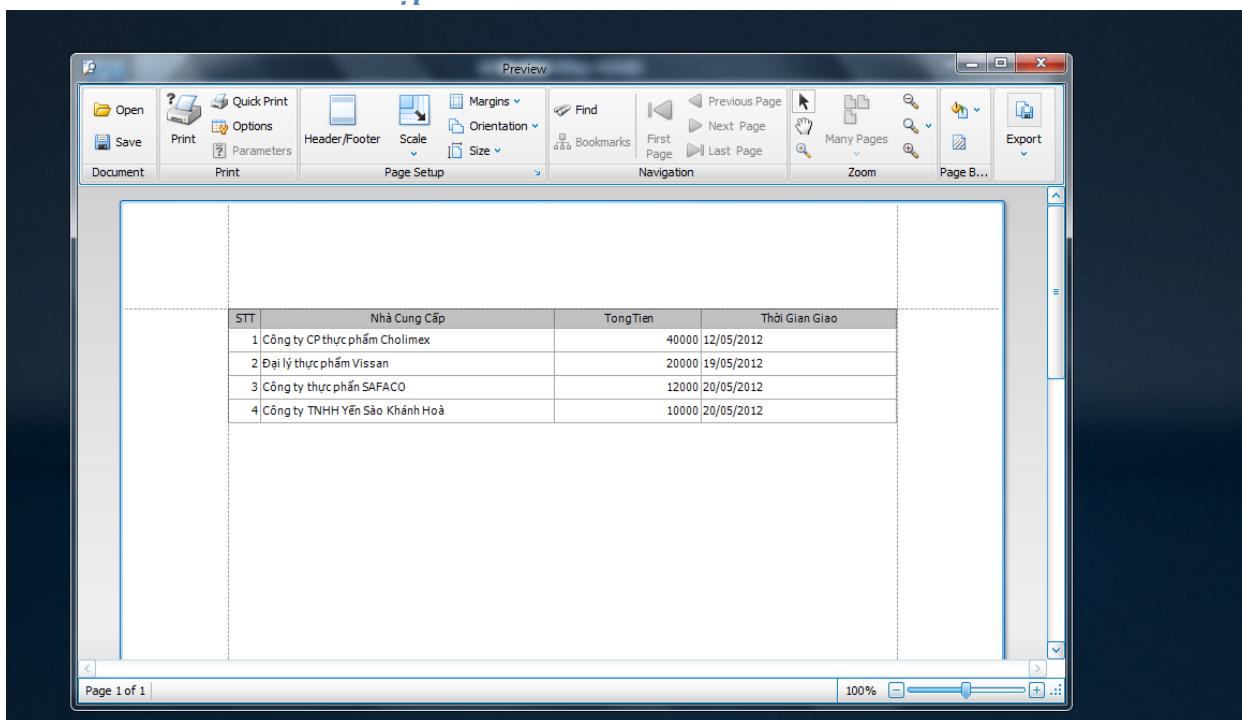


### 3.3.4.Quản Lý Thông Tin Nhập Kho

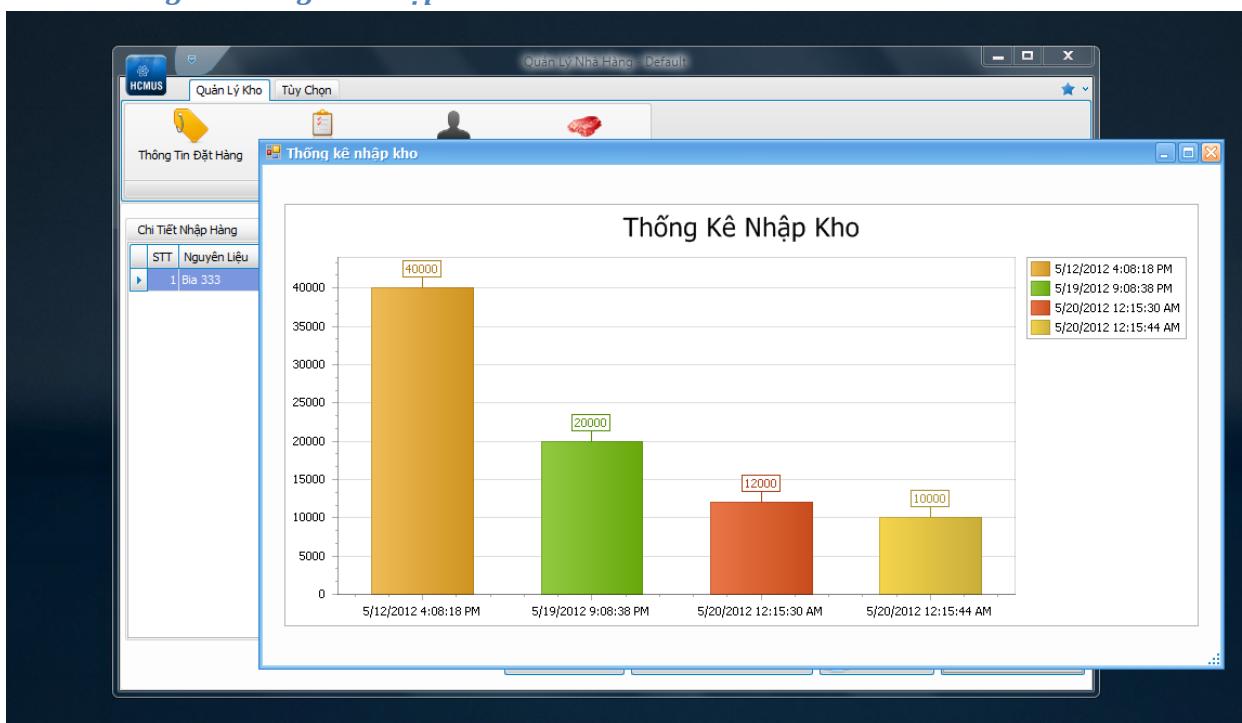
#### 3.3.4.1.Thêm Phiếu Nhập Kho



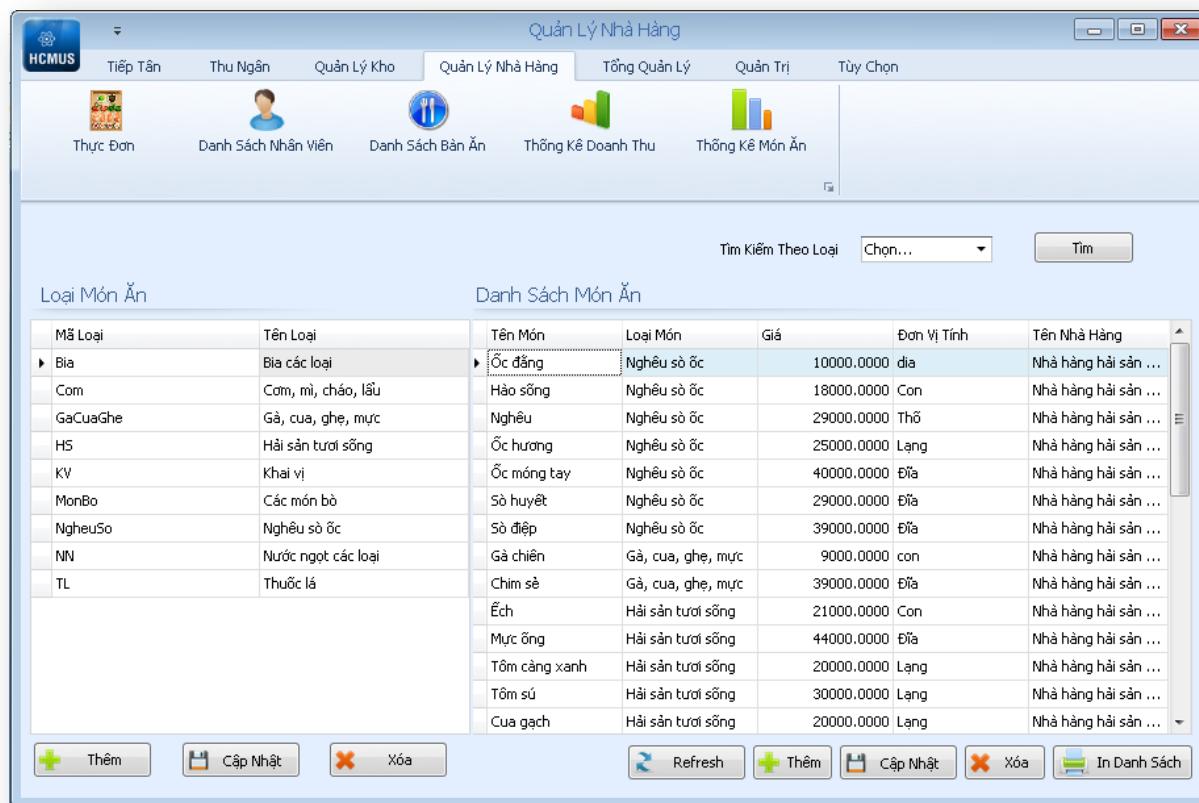
### 3.3.4.2.In Danh Sách Phiếu Nhập Kho



### 3.3.4.3.Thống Kê Thông Tin Nhập Kho



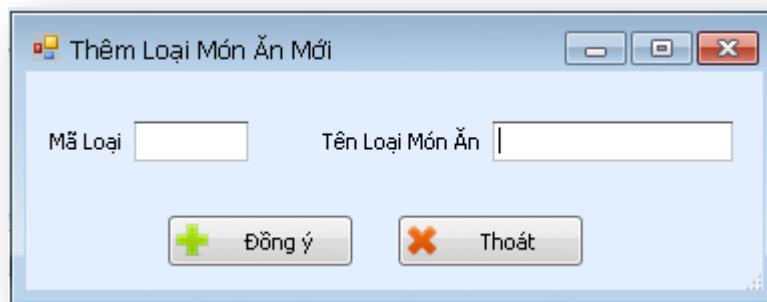
## 3.4 Quản Lý Nhà Hàng



### 3.4.1. Quản lý Thực đơn

*Quản lý Loại Món Ăn*

+ Thêm Loại Món Ăn



+ Cập nhật Loại Món Ăn

**Quản Lý Nhà Hàng**

Tiếp Tân Thu Ngân Quản Lý Kho Quản Lý Nhà Hàng Tổng Quản Lý Quản Trị Tùy Chọn

Thực Đơn Danh Sách Nhân Viên Danh Sách Bàn Ăn Thông Kê Doanh Thu Thông Kê Món Ăn

Tìm Kiếm Theo Loại: Chọn... Tìm

**Loại Món Ăn**

Mã Loại	Tên Loại
Bia	Bia các loại
Com	Com, mì, cháo, lẩu
GaCuaGhe	Gà, cua, ghe, mực
HS	Hải sản tươi sống
KV	Khai vị
MonBo	Các món bò
NgheuSo	Nghêu sò ốc
NN	Nước ngọt các loại
TL	Thuốc lá

**Danh Sách Món Ăn**

Thông Báo	Giá	Đơn Vị Tính	Tên Nhà Hàng
Cập nhật thành công	10000.0000	đia	Nhà hàng hải sản ...
OK	18000.0000	Con	Nhà hàng hải sản ...
	29000.0000	Thỗ	Nhà hàng hải sản ...
	25000.0000	Lạng	Nhà hàng hải sản ...
	40000.0000	Đĩa	Nhà hàng hải sản ...
Sò huyết	29000.0000	Đĩa	Nhà hàng hải sản ...
Sò điệp	39000.0000	Đĩa	Nhà hàng hải sản ...
Gà chiên	9000.0000	con	Nhà hàng hải sản ...
Chim sẻ	39000.0000	Đĩa	Nhà hàng hải sản ...
Ếch	21000.0000	Con	Nhà hàng hải sản ...
Mực ống	44000.0000	Đĩa	Nhà hàng hải sản ...
Tôm cảng xanh	20000.0000	Lạng	Nhà hàng hải sản ...
Tôm sú	30000.0000	Lạng	Nhà hàng hải sản ...
Cua gạch	20000.0000	Lạng	Nhà hàng hải sản ...

Thêm Cập Nhật Xóa Refresh Thêm Cập Nhật Xóa In Danh Sách

**+Xóa Loại Món Ăn**

**Quản Lý Nhà Hàng**

Tiếp Tân Thu Ngân Quản Lý Kho Quản Lý Nhà Hàng Tổng Quản Lý Quản Trị Tùy Chọn

Thực Đơn Danh Sách Nhân Viên Danh Sách Bàn Ăn Thông Kê Doanh Thu Thông Kê Món Ăn

Tìm Kiếm Theo Loại: Chọn... Tìm

**Loại Món Ăn**

Mã Loại	Tên Loại
AN	Thức ăn nhanh
Bia	
Com	
GaCuaGhe	
HS	
KV	
MonBo	
NgheuSo	Nghêu sò ốc
NN	Nước ngọt các loại
TL	Thuốc lá

**Danh Sách Món Ăn**

Thông Báo	Tên Món	Loại Món	Giá	Đơn Vị Tính	Tên Nhà Hàng
Bạn có muốn xóa dòng thứ 1	Ốc đắng	Nghêu sò ốc	10000.0000	đia	Nhà hàng hải sản ...
Yes No	Nghêu sò ốc	Nghêu sò ốc	18000.0000	Con	Nhà hàng hải sản ...
	Nghêu sò ốc	Nghêu sò ốc	29000.0000	Thỗ	Nhà hàng hải sản ...
	Nghêu sò ốc	Nghêu sò ốc	25000.0000	Lang	Nhà hàng hải sản ...
	Nghêu sò ốc	Nghêu sò ốc	40000.0000	Đĩa	Nhà hàng hải sản ...
	Nghêu sò ốc	Nghêu sò ốc	29000.0000	Đĩa	Nhà hàng hải sản ...
	Nghêu sò ốc	Nghêu sò ốc	39000.0000	Đĩa	Nhà hàng hải sản ...
	Gà, cua, ghe, mực	Gà, cua, ghe, mực	9000.0000	con	Nhà hàng hải sản ...
	Gà, cua, ghe, mực	Gà, cua, ghe, mực	39000.0000	Đĩa	Nhà hàng hải sản ...
	Chim sẻ	Gà, cua, ghe, mực	39000.0000	Đĩa	Nhà hàng hải sản ...
	Ếch	Hải sản tươi sống	21000.0000	Con	Nhà hàng hải sản ...
	Mực ống	Hải sản tươi sống	44000.0000	Đĩa	Nhà hàng hải sản ...
	Tôm cảng xanh	Hải sản tươi sống	20000.0000	Lạng	Nhà hàng hải sản ...
	Tôm sú	Hải sản tươi sống	30000.0000	Lạng	Nhà hàng hải sản ...
	Cua gạch	Hải sản tươi sống	20000.0000	Lạng	Nhà hàng hải sản ...

Thêm Cập Nhật Xóa Refresh Thêm Cập Nhật Xóa In Danh Sách

**Quản lý Món Ăn****+ Thêm Món Ăn**

**Form\_ThemMonAn**

Tên Món Ăn	<input type="text"/>
Loại Món Ăn	<input type="text"/>
Giá	<input type="text"/>
Đơn Vị Tính	<input type="text"/>
<input style="width: 30px; height: 30px; margin-right: 10px;" type="button" value="+"/> <b>Đồng Ý</b> <input style="width: 30px; height: 30px; margin-left: 10px;" type="button" value="X"/> <b>Thoát</b>	

**+ Cập nhật Món Ăn**

**Quản Lý Nhà Hàng**

Thực Đơn      Danh Sách Nhân Viên      Danh Sách Bàn Ăn      Thống Kê Doanh Thu      Thống Kê Món Ăn

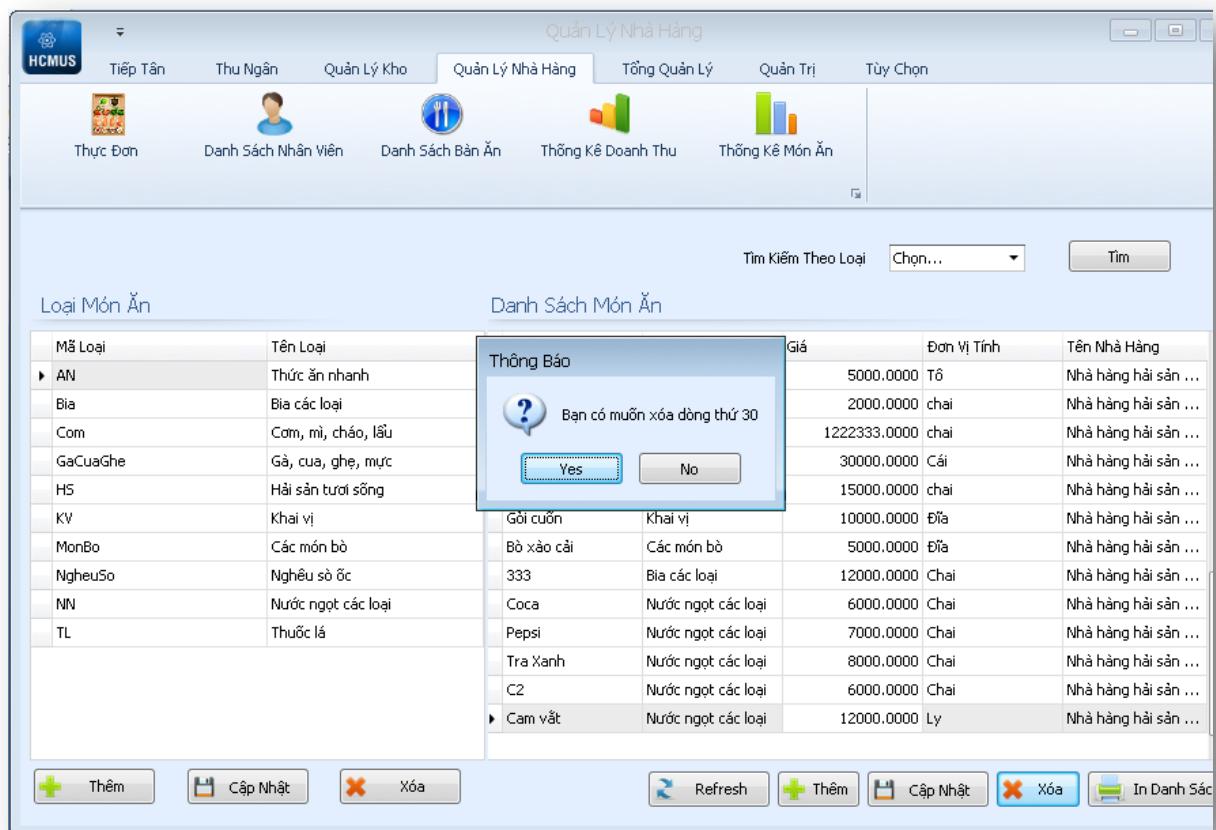
Tìm Kiếm Theo Loại:  Chọn...      Tim

Loại Món Ăn		Danh Sách Món Ăn			
Mã Loại	Tên Loại	Giá	Đơn Vị Tính	Tên Nhà Hàng	
AN	Thức ăn nhanh	10000.0000	đia	Nhà hàng hải sản ...	
Bia	Bia các loại	18000.0000	Con	Nhà hàng hải sản ...	
Com	Com, mì, cháo, lẩu	29000.0000	Thổ	Nhà hàng hải sản ...	
GaCuaGhe	Gà, cua, ghẹ, mực	25000.0000	Lạng	Nhà hàng hải sản ...	
H5	Hải sản tươi sống	40000.0000	Đĩa	Nhà hàng hải sản ...	
KV	Khai vị	29000.0000	Đĩa	Nhà hàng hải sản ...	
MonBo	Các món bò	39000.0000	Đĩa	Nhà hàng hải sản ...	
NgheuSo	Nghêu sò ốc	9000.0000	con	Nhà hàng hải sản ...	
NN	Nước ngọt các loại	39000.0000	Đĩa	Nhà hàng hải sản ...	
TL	Thuốc lá	21000.0000	Con	Nhà hàng hải sản ...	
		44000.0000	Đĩa	Nhà hàng hải sản ...	
	Sò huyết	39000.0000	Đĩa	Nhà hàng hải sản ...	
	Sò điệp	9000.0000	con	Nhà hàng hải sản ...	
	Gà chiên	39000.0000	Đĩa	Nhà hàng hải sản ...	
	Chim sè	21000.0000	Con	Nhà hàng hải sản ...	
	Ếch	44000.0000	Đĩa	Nhà hàng hải sản ...	
	Mực ống	20000.0000	Lạng	Nhà hàng hải sản ...	
	Tôm cảng xanh	30000.0000	Lạng	Nhà hàng hải sản ...	
	Tôm sú	20000.0000	Lạng	Nhà hàng hải sản ...	
	Cua gạch	20000.0000	Lạng	Nhà hàng hải sản ...	

**Thông Báo**  
Cập nhật thành công  
**OK**

**Thao Tác:** Thêm      Cập Nhật      Xóa      Refresh      Thêm      Cập Nhật      Xóa      In Danh Sách

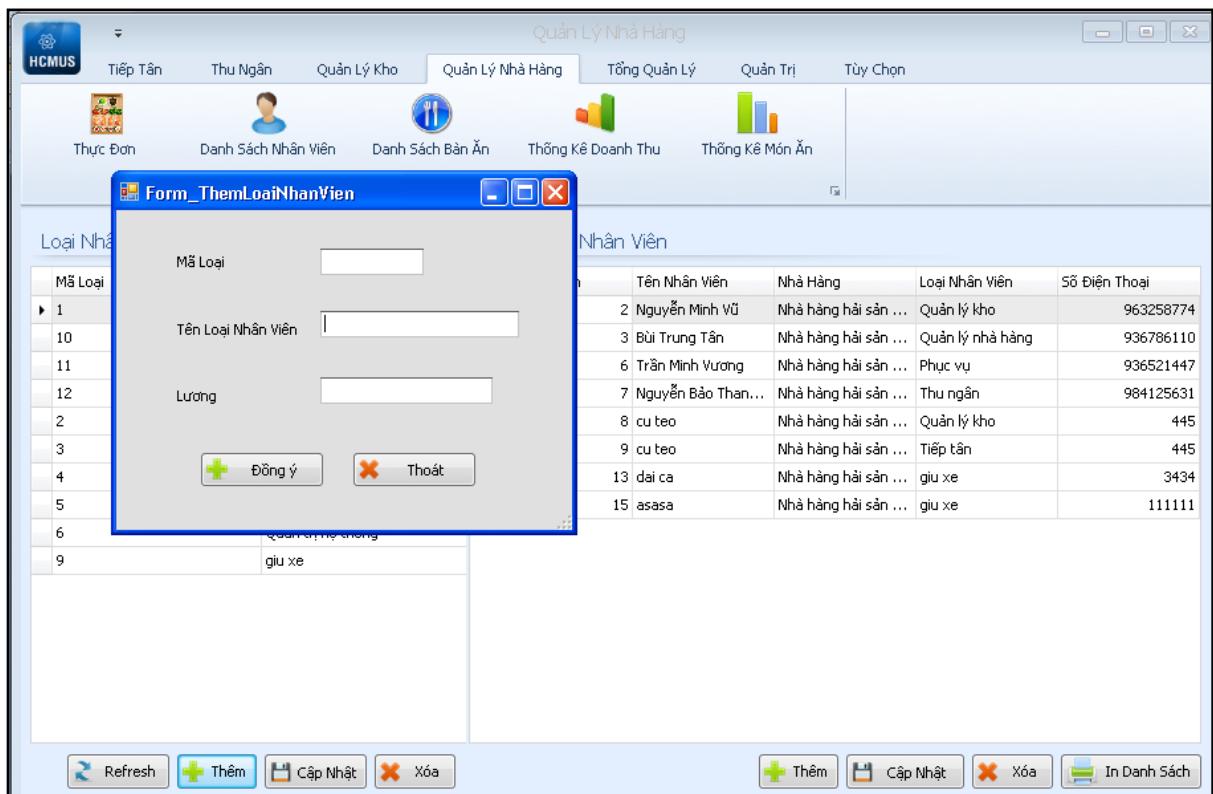
+ Xóa Món Ăn



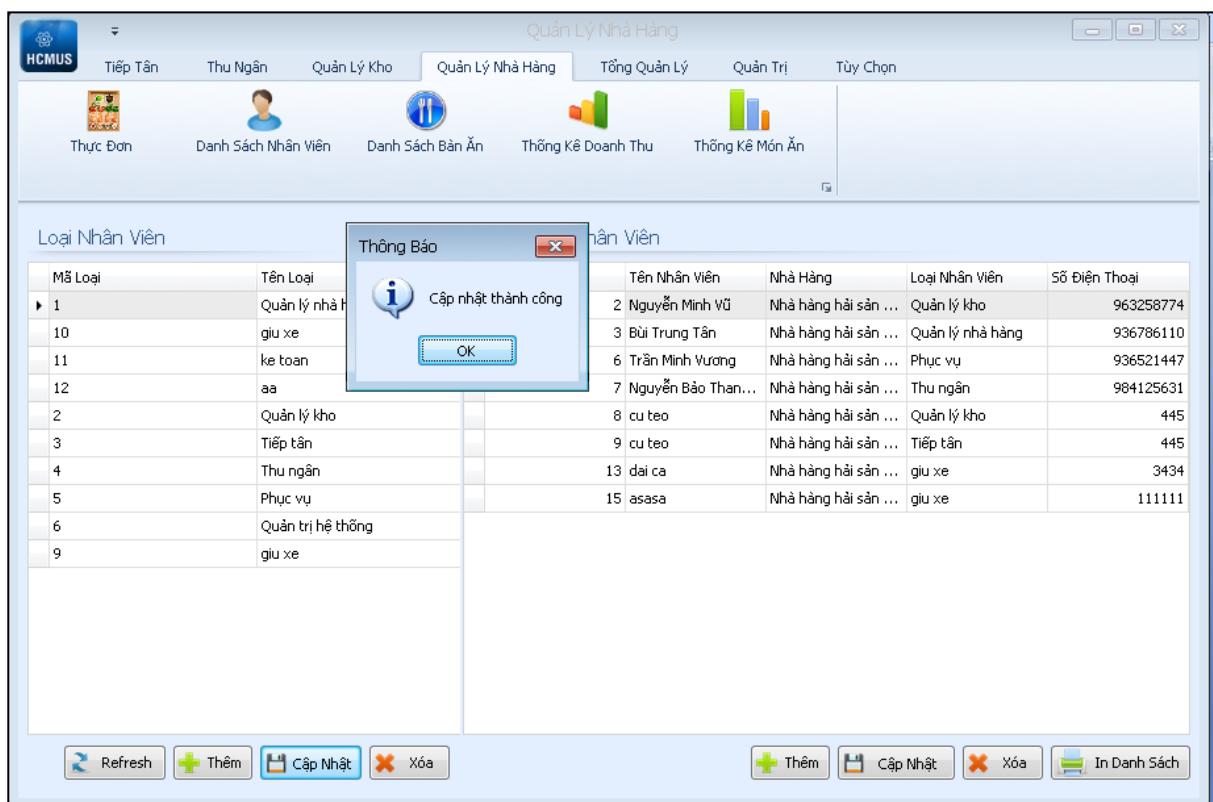
### 3.4.2. Quản Lý Danh Sách Nhân Viên

#### Quản lý Loại Nhân Viên

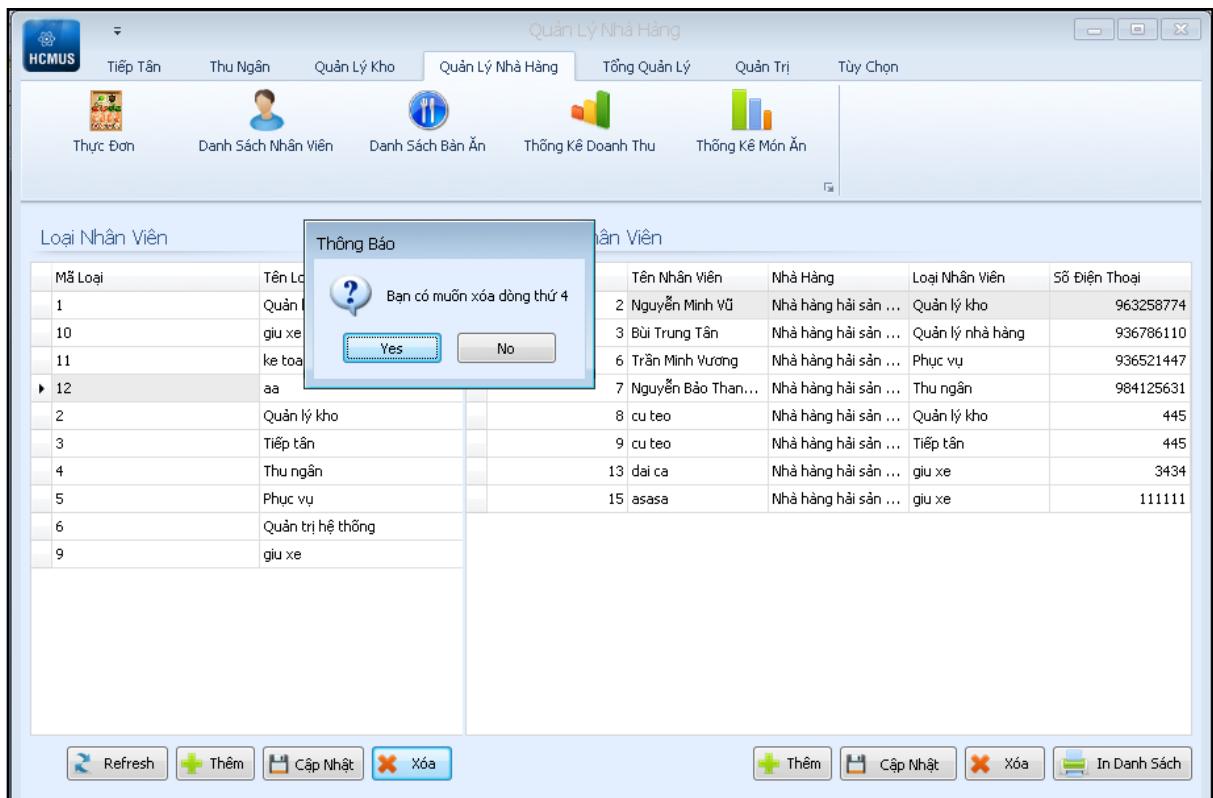
+ Thêm Loại Nhân Viên



### + Cập nhật Loại Nhân Viên



## + Xóa Loại Nhân Viên

*Quản lý Nhân Viên**+ Thêm Nhân viên*

Quản Lý Nhà Hàng

Tiếp Tân Thu Ngân Quản Lý Kho Quản Lý Nhà Hàng Tổng Quản Lý Quản Trị Tùy Chọn

**Form\_ThemNhanVien**

Tên Nhân Viên	<input type="text"/>
Loại Nhân Viên	<input type="button" value="▼"/>
CMND	<input type="text"/>
Số ĐT	<input type="text"/>
<input style="width: 30px; height: 20px; margin-right: 10px;" type="button" value="+"/> Đồng Ý <input style="width: 30px; height: 20px;" type="button" value="X"/>	

Danh Sách Nhân Viên

Mã Nhân Viên	Tên Nhân Viên	Nhà Hàng	Loại Nhân Viên	Số Điện Thoại
2	Nguyễn Minh Vũ	Nhà hàng hải sản ...	Quản lý kho	963258774
3	Bùi Trung Tân	Nhà hàng hải sản ...	Quản lý nhà hàng	936786110
6	Trần Minh Vương	Nhà hàng hải sản ...	Phục vụ	936521447
7	Nguyễn Bảo Thành	Nhà hàng hải sản ...	Thư ngân	984125631
8	cu teo	Nhà hàng hải sản ...	Quản lý kho	445
9	cu teo	Nhà hàng hải sản ...	giú xe	445
13	dai ca	Nhà hàng hải sản ...	giú xe	3434
15	asasa	Nhà hàng hải sản ...	giú xe	111111
17	nghia	Nhà hàng hải sản ...	Quản lý nhà hàng	123345

Refresh  Thêm  Xóa  Cập Nhật  Xóa

### +Cập nhật Nhân viên

Quản Lý Nhà Hàng

Tiếp Tân Thu Ngân Quản Lý Kho Quản Lý Nhà Hàng Tổng Quản Lý Quản Trị Tùy Chọn

Thực Đơn   Thông Kê Doanh Thu Thông Kê Món Ăn

Loại Nhân Viên

Mã Loại	Tên Loại
1	Quản lý nhà hàng
10	giú xe
11	ke toán
12	aa
2	Quản lý kho
3	Tiếp tân
4	Thư ngân
5	Phục vụ
6	Quản trị hệ thống
9	giú xe

Danh Sách Nhân Viên

Mã Nhân Viên	Tên Nhân Viên	Nhà Hàng	Loại Nhân Viên	Số Điện Thoại
2	Nguyễn Minh Vũ	Nhà hàng hải sản ...	Quản lý kho	963258774
3	Bùi Trung Tân	Nhà hàng hải sản ...	Quản lý nhà hàng	936786110
6	Trần Minh Vương	Nhà hàng hải sản ...	Phục vụ	936521447
7	Nguyễn Bảo Thành	Nhà hàng hải sản ...	Thư ngân	984125631
8	cu teo	Nhà hàng hải sản ...	giú xe	445
9	cu teo	Nhà hàng hải sản ...	giú xe	445
13	dai ca	Nhà hàng hải sản ...	giú xe	3434
15	asasa	Nhà hàng hải sản ...	giú xe	111111
17	nghia	Nhà hàng hải sản ...	Quản lý nhà hàng	123345

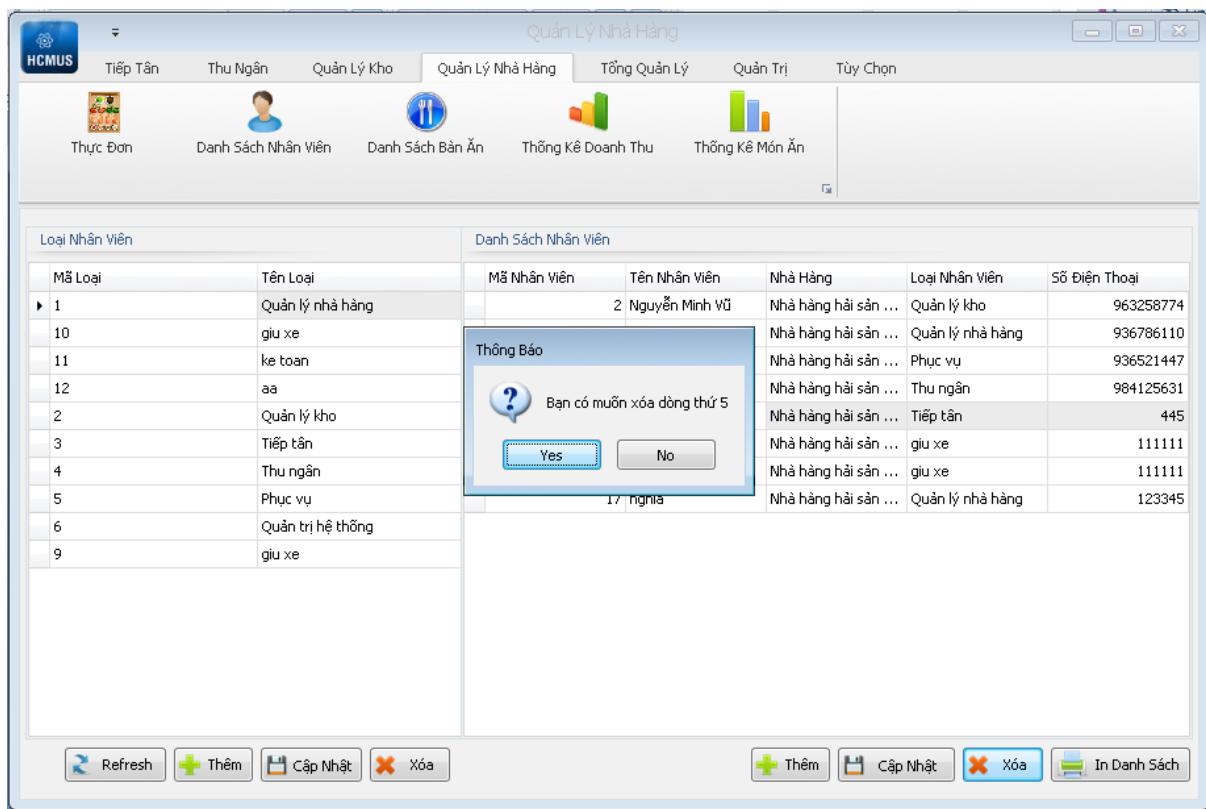
Thông Báo

Cập nhật thành công

OK

Refresh  Thêm  Xóa  Cập Nhật  Xóa

+Xóa Nhân Viên



### 3.4.3. Quản Lý Danh Sách Bàn Ăn

*Quản Lý Loại Bàn Ăn*

+ Thêm Loại Bàn Ăn

Quản Lý Nhà Hàng

Form\_ThemLoaiBanAn

Mã Loại	Sức Chứa

**Đồng Ý** **Thoát**

**Thêm** **Cập Nhật** **Xóa**

**Khu Vực**

Mã Khu Vực	Tên Khu Vực	Vị Trí
1	Sảnh – tầng trệt	Giữa đại sảnh c...
2	Góc – tầng trệt	Nằm ở góc
3	Sát tường - tần...	Sát tường, nhìn ...

**Thêm** **Cập Nhật** **Xóa** **In Danh Sách**

Thống Kê Món Ăn

Mã	Khu Vực	Sức Chứa	Nhà Hàng	Trạng Thái
1	Sảnh – tầng trệt	20	Nhà hàng hải sản ...	Chưa đến ăn
2	Góc – tầng trệt	15	Nhà hàng hải sản ...	Chưa đến ăn
3	Sân thượng	10	Nhà hàng hải sản ...	Đã đến ăn
9	Sân thượng	20	Nhà hàng hải sản ...	Chưa đến ăn
10	Sân thượng	15	Nhà hàng hải sản ...	Chưa đến ăn

### +Cập nhật Loại Bàn Ăn

Quản Lý Nhà Hàng

Danh Sách Nhân Viên

Danh Sách Bàn Ăn

Thông Báo

Cập nhật thành công

OK

**Thêm** **Cập Nhật** **Xóa**

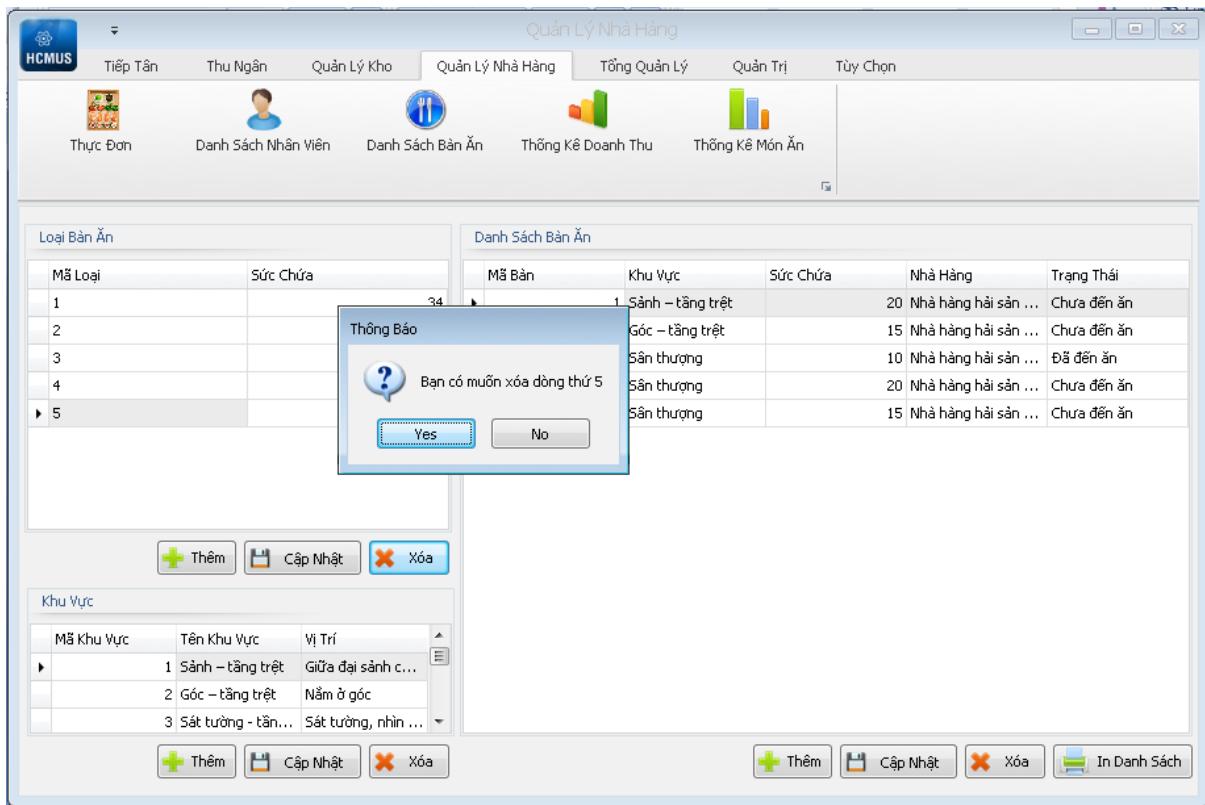
**Khu Vực**

Mã Khu Vực	Tên Khu Vực	Vị Trí
1	Sảnh – tầng trệt	Giữa đại sảnh c...
2	Góc – tầng trệt	Nằm ở góc
3	Sát tường - tần...	Sát tường, nhìn ...

**Thêm** **Cập Nhật** **Xóa** **In Danh Sách**

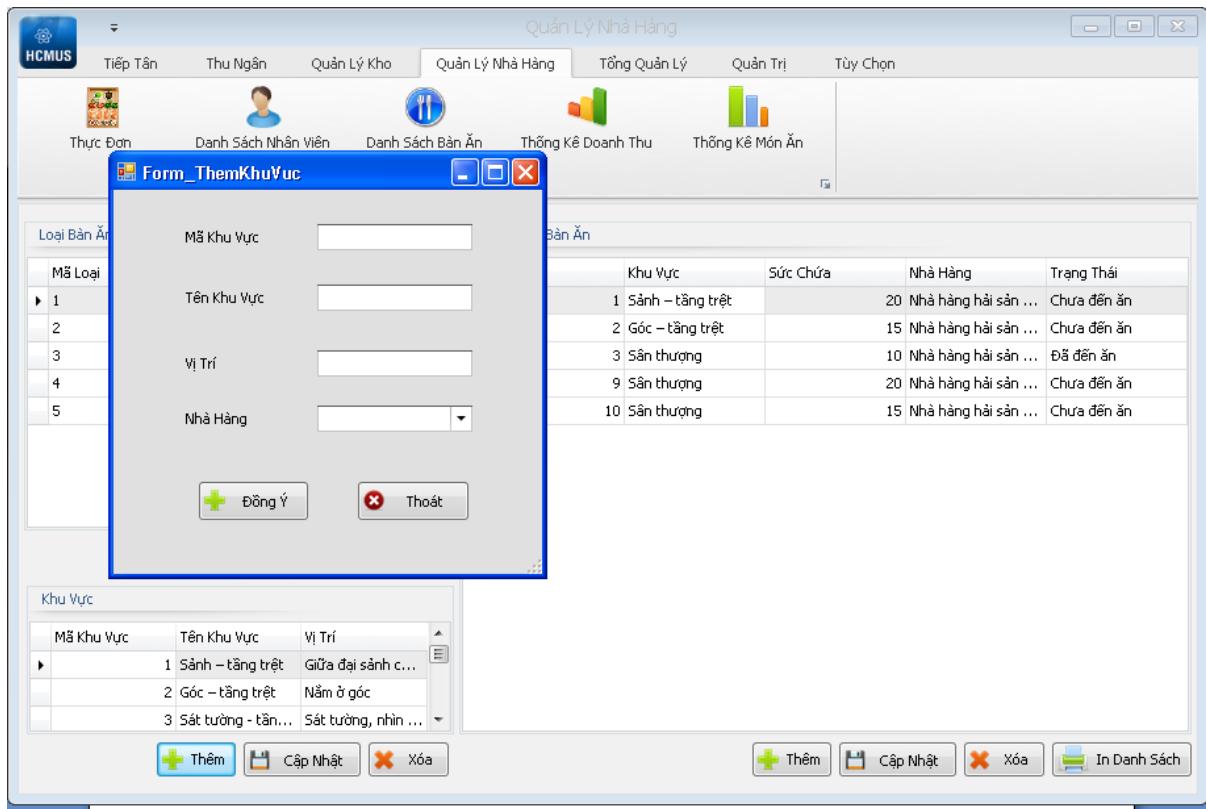
Mã	Khu Vực	Sức Chứa	Nhà Hàng	Trạng Thái
1	Sảnh – tầng trệt	20	Nhà hàng hải sản ...	Chưa đến ăn
2	Góc – tầng trệt	15	Nhà hàng hải sản ...	Chưa đến ăn
3	Sân thượng	10	Nhà hàng hải sản ...	Đã đến ăn
9	Sân thượng	20	Nhà hàng hải sản ...	Chưa đến ăn
10	Sân thượng	15	Nhà hàng hải sản ...	Chưa đến ăn

## +Xóa Loại Bàn Ăn

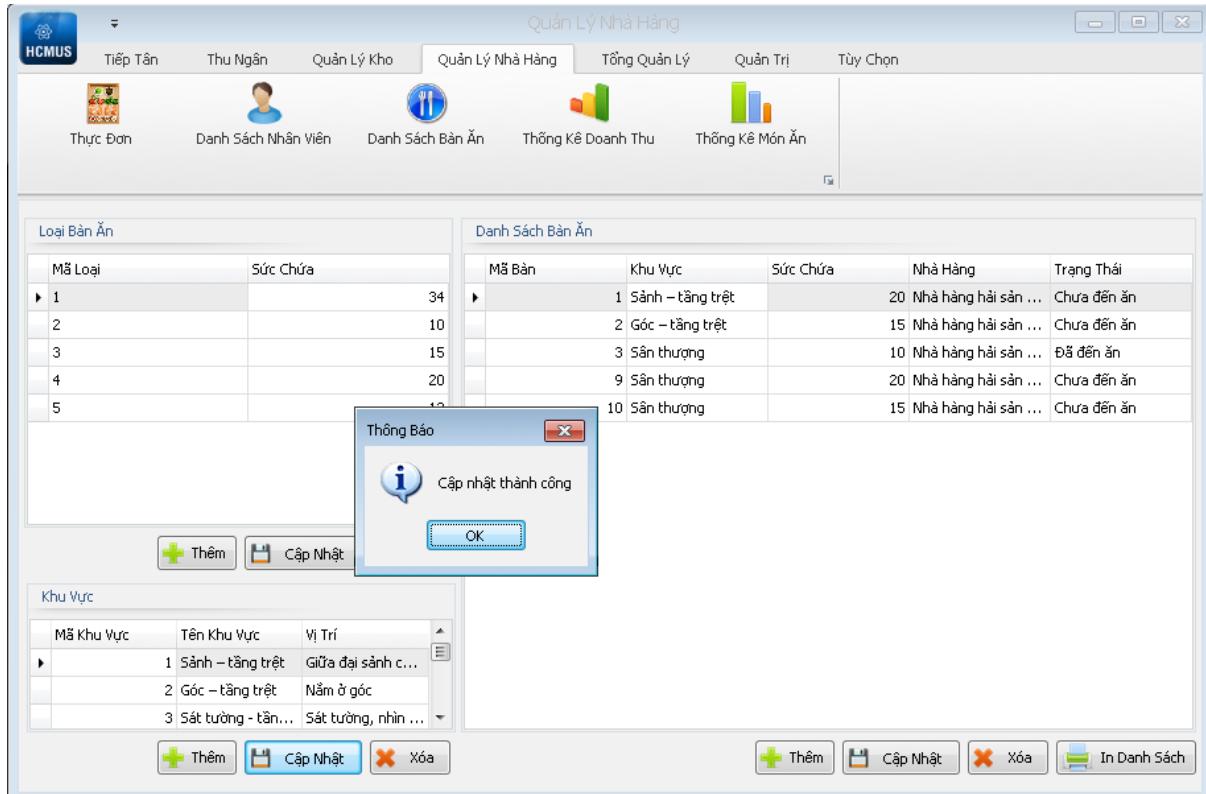


## Quản Lý Khu Vực

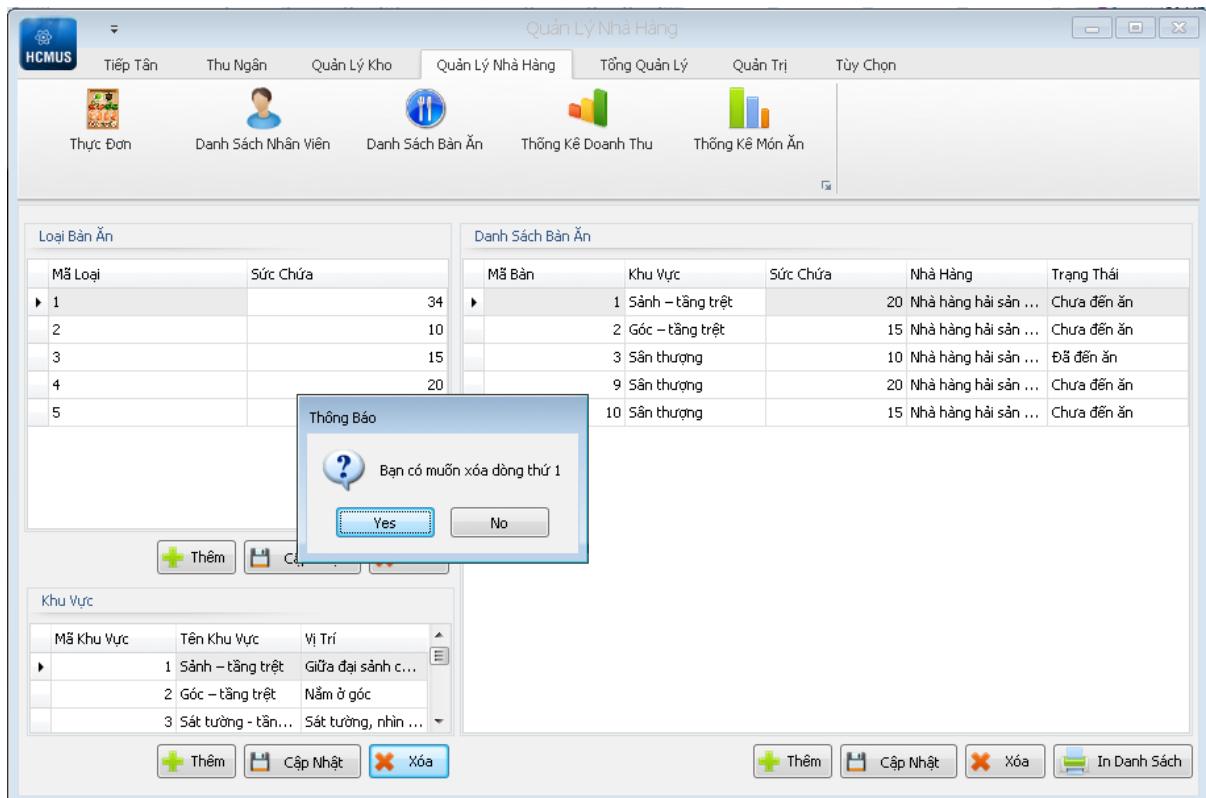
+ Thêm Khu Vực



### + Cập nhật Khu vực



### + Xóa Khu Vực



### Quản Lý Danh sách Bàn Ăn

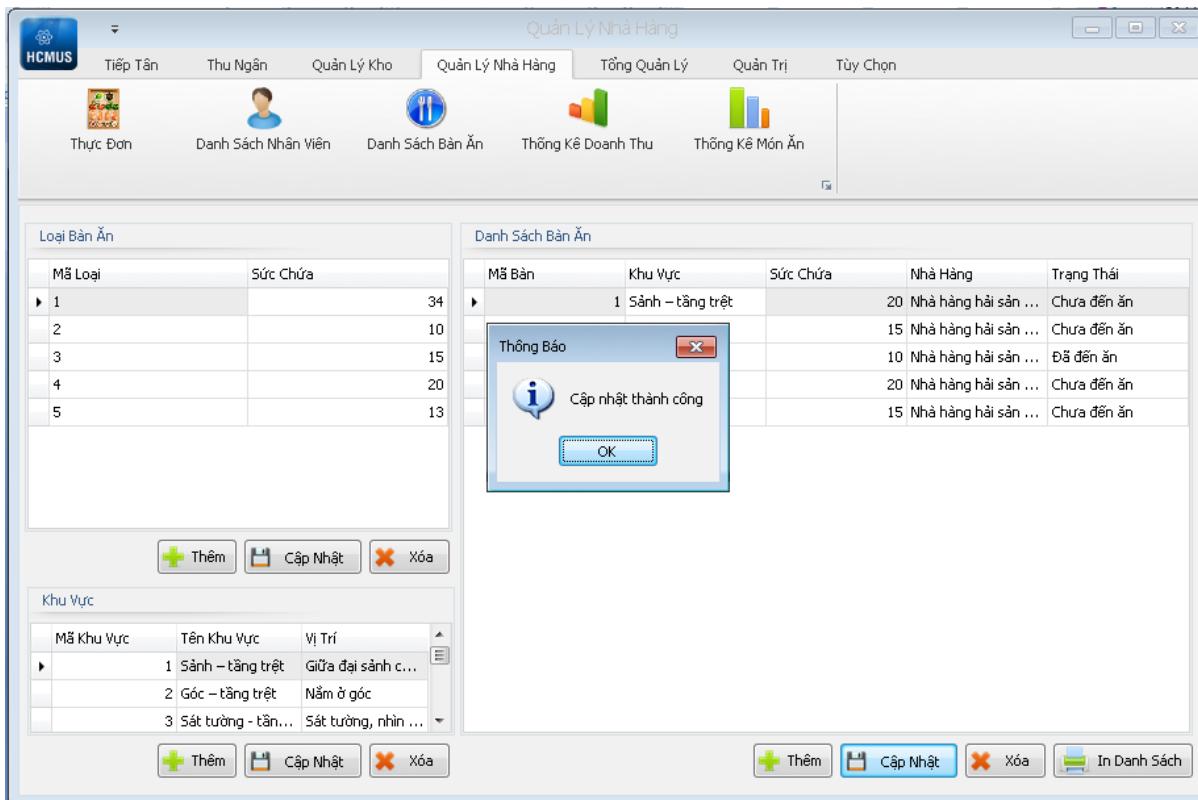
+ Thêm Bàn Ăn

The screenshot shows the 'Quản Lý Nhà Hàng' (Restaurant Management) application interface. The main window has a menu bar with 'HCMUS', 'Tiếp Tân', 'Thu Ngân', 'Quản Lý Kho', 'Quản Lý Nhà Hàng' (selected), 'Tổng Quản Lý', 'Quản Trị', and 'Tùy Chọn'. Below the menu is a toolbar with icons for 'Thực Đơn', 'Danh Sách Nhân Viên', 'Danh Sách Bàn Ăn' (highlighted), 'Thông Kê Doanh Thu', and 'Thông Kê Món Ăn'. The central area displays a list of tables with columns 'Sức chứa' (Capacity), 'Nhà Hàng' (Restaurant), and 'Trạng Thái' (Status). A modal dialog box titled 'Form\_ThemBanAn' is open, containing fields for 'Khu Vực' (Area) and 'Sức chứa' (Capacity), with 'Đồng Ý' (Accept) and 'Thoát' (Exit) buttons. At the bottom of the main window, there are buttons for 'Thêm' (Add), 'Cập Nhật' (Update), 'Xóa' (Delete), and 'In Danh Sách' (Print List).

Mã Loại	Sức chứa	Nhà Hàng	Trạng Thái
1		20	Nhà hàng hải sản ...
2		15	Nhà hàng hải sản ...
3		10	Nhà hàng hải sản ...
4		20	Nhà hàng hải sản ...
5		15	Nhà hàng hải sản ...

Mã Khu Vực	Tên Khu Vực	Vị Trí
1	Sảnh - tầng trệt	Giữa đại sảnh c...
2	Góc - tầng trệt	Nằm ở góc
3	Sát tường - tần...	Sát tường, nhìn ...

+ Cập nhật Bàn Ăn



### + Xóa Bàn Ăn

