**1. Chức năng đặt bàn:**

Các Actor:

* Tiếp tân
* Quản lý nhà hàng

Các quan hệ:

* DatBan:

Các hành động:

* Thêm đặt bàn:
  + B1: Tìm bàn có trạng thái trống trong bảng BanAn: tìm trong bảng DatBan theo MaBan và GioDen
  + B2: Cập nhật thông tin vào bảng BanAn
  + Cập nhận trạng thái cho bàn vừa đặt: nếu bàn đó đang ở trạng thái **sử dụng** hoặc **đang đặt** thì giữ nguyên trạng thái, ngược lại thì cập nhật trạng thái thành **đang đặt**.
* Sửa đặt bàn

PHÂN TÍCH HÀNH ĐỘNG:

1. Đặt bàn:

* Tìm bàn có trạng thái trống:
  + Tìm được một mã bàn MB phù hợp yêu cầu khách hàng, giờ đến là GD
  + Kiểm tra mã bàn trong bảng DatBan
    - SELECT \* FROM DatBan WHERE MaBan = MB AND GioDen = GD AND (TinhTrang = 1 OR TinhTrang = 2)
    - Trường hợp 1: câu select trả về NULL. Tức là có thể đặt bàn này, nhưng khi lưu xuống Database thì không thành công vì một giao tác khác đã đặt bàn theo cùng MB và GD trước đó.
      * Đây là tranh chấp Phantom.
    - Trường hợp 2: câu select trả về khác NULL. Tức là bàn này đã có người đặt trước. Có thể phát sinh Unrepeatable Read, nhưng không nghiệp vụ thực tế nào làm phát sinh tình huống này.
* Cập nhật trạng thái trong bảng BanAn: đúng lúc giao tác T1 cập nhật trạng thái cho bàn là **sử dụng** thì giao tác T2 cập nhật trạng thái là **đang đặt**, trước đó giao tác 2 đã đọc được bàn ăn đang tranh chấp còn ở trạng thái **trống**. Như thế bàn bị cập nhật là **đang đặt**. Tức là sai kết quả nghiệp vụ.
  + Đây là tranh chấp Lost Update.

**2. Chức năng quản lý doanh thu:**

Các Actor:

* Tiếp tân
* Quản lý nhà hàng

Các quan hệ:

* HoaDon, ChiTietHoaDon
* DatHang, ChiTietDatHang
* NhanVien, LuongNhanVien

Các hành động:

* Đọc thông tin lương của nhân viên trong: NhanVien và LuongNhanVien
* Đọc thông tin đặt hàng các nguyên liệu trong: DatHang, ChiTietDatHang
* Đọc thông tin số tiền thu được từ các món ăn: HoaDon, ChiTietHoaDon

Do đây là các hành động đọc nên không phát sinh tranh chấp Lost Update.

PHÂN TÍCH HÀNH ĐỘNG:

1. Đọc thông tin lương của nhân viên trong: NhanVien và LuongNhanVien

* T2 thay đổi lương của nhân viên A, T1 đọc thông tin lương của nhân viên A để xử lý, T2 rollback, **vậy T1 bị Dirty Read**.
* T1 đọc được thông tin một loại nhân viên để xử lý, ngay sau đó thì T2 xoá loại nhân viên đó. Vậy T1 không thể đọc lại, hoặc cập nhật thông tin cho loại nhân viên vừa đọc được. **Đây là tranh chấp Unrepeatable Read.**
* T1 đọc được thông tin các nhân viên, ngay sau đó T2 thêm một (hoặc nhiều) nhân viên mới làm cho khi T1 đọc lại thì thấy có nhiều nhân viên hơn so với lần đọc trước. **Đây là tranh chấp Phantom.**

1. Đọc thông tin đặt hàng các nguyên liệu trong: DatHang, ChiTietDatHang

* T1 đọc được thông tin của đơn đặt hàng A trong bảng DatHang theo điều kiện ThoiGianDat, ngay sau đó thì T2 thay đổi thông tin thời gian đặt của đơn đặt hàng A làm cho điều kiện WHERE của T1 không còn đúng với đơn đặt hàng A nữa, hoặc T2 xoá đơn đặt hàng A. Vậy T1 không thể đọc lại thông tin của đơn đặt hàng A. **Đây là tranh chấp Unrepeatable Read.**
* T1 đọc được các thông tin của các đơn đặt hàng thoả điều kiện ThoiGianDat = TGD. Ngay sau đó T2 thêm vào thông tin một đơn đặt hàng mới A cũng có ThoiGianDat = TGD. Như vậy khi T1 tính tổng tiền của các đơn đặt hàng có ThoiGianDat = TGD thì đã đọc được thông tin của A trong khi trước đó không đọc được A. **Đây là tranh chấp Phantom.**
* Có thể phát sinh Dirty Read, nhưng không nghiệp vụ thực tế nào làm phát sinh tình huống này.

1. Đọc thông tin số tiền thu được từ các món ăn: HoaDon, ChiTietHoaDon

* T1 đọc được thông tin các hoá đơn có ThoiGianLap = TGL, T2 thay đổi ThoiGianLap của một hoặc nhiều hoá đơn (hoặc xoá hoá đơn) mà T1 đã đọc trước đó làm cho T1 không thể đọc lại được thông tin các hoá đơn đã bị thay đổi. Đây là tranh chấp **Unrepeatable Read.**
* T1 đọc được thông tin các hoá đơn có ThoiGianLap = TGL, T2 thêm một (hoặc nhiều) hoá đơn A có ThoiGianLap = TGL làm cho khi T1 đọc lại để tính tổng tiền thì sẽ đọc được thông tin của A trong khi trước đó không đọc được. Đây là tranh chấp **Phantom.**