```csharp
using DevExpress.DataAccess.ObjectBinding;
using Meditech.Core.Auth.Entities;
using Meditech.Core.Common.Repositories;
using System.ComponentModel;

namespace Meditech.App.CommonService.Services
{
    public interface IDonKSKData
    {
        public IEnumerable<DonKSK> GetDonHang(Guid id);
    }

    [DisplayName("DonKSKData")]
    [HighlightedClass]
    public class DonKSKData : IDonKSKData
    {
        private readonly IDonHangKhamSucKhoeRepository _donHangKhoeRepository;
        private readonly IPhuongAnKinhDoanhKskRepository _phuongAnKinhDoanhKskRepository;
        private readonly IDoanKhamSucKhoeRepository _doanKhamSucKhoeRepository;
        private readonly IDanhSachCongTyRepository _danhSachCongTyRepository;
        private readonly IDonHangKhamSucKhoeKhacRepository _donHangKhamSucKhoeKhacRepository;
        private readonly IUserRepository _userRepository;
        private readonly IDepartmentCategoryRepository _departmentCategoryRepository;
        private readonly ISignedDocumentRepository _signedDocumentRepository;

        [HighlightedMember]
        public DonKSKData(IDonHangKhamSucKhoeRepository donHangKhoeRepository,
            IPhuongAnKinhDoanhKskRepository phuongAnKinhDoanhKskRepository,
            IDoanKhamSucKhoeRepository doanKhamSucKhoeRepository,
            IDanhSachCongTyRepository danhSachCongTyRepository,
            IDonHangKhamSucKhoeKhacRepository donHangKhamSucKhoeKhacRepository,
            IEmployeeRepository employeeRepository,
            IUserRepository userRepository,
            IDepartmentCategoryRepository departmentCategoryRepository,
            ISignedDocumentRepository signedDocumentRepository)
        {
            _donHangKhoeRepository = donHangKhoeRepository;
            _phuongAnKinhDoanhKskRepository = phuongAnKinhDoanhKskRepository;
            _doanKhamSucKhoeRepository = doanKhamSucKhoeRepository;
            _danhSachCongTyRepository = danhSachCongTyRepository;
            _donHangKhamSucKhoeKhacRepository = donHangKhamSucKhoeKhacRepository;
            _userRepository = userRepository;
            _departmentCategoryRepository = departmentCategoryRepository;
            _signedDocumentRepository = signedDocumentRepository;
        }

        [HighlightedMember]
        public IEnumerable<DonKSK> GetDonHang(Guid id)//Guid id
        {
            var userLst = _userRepository.GetAll().ToList();
            var donVi = _departmentCategoryRepository.GetAll().ToList();
            var sign = _signedDocumentRepository.GetAll().Where(w => w.DocumentId == id && w.Required && w.Signed == 1).OrderByDescending(o => o.SigningRound).ToList(

            var don = from a in _donHangKhoeRepository.Table
                      join b in _phuongAnKinhDoanhKskRepository.Table on a.PhuongAnId equals b.Id
                      join c in _doanKhamSucKhoeRepository.Table on b.DoanKhamId equals c.Id
                      join d in _danhSachCongTyRepository.Table on c.CongTyId equals d.Id
                      where a.Id == id
                      select new
                      {
                          TenDonVi = "",
                          KhoaPhong = "",
                          SoPhieu = a.MaDonHang,
                          DiaChi = d.DiaChi,
                          NgayIn = $"Thanh Hóa, Ngày {DateTime.Now.Day} tháng {DateTime.Now.Month} năm {DateTime.Now.Year}",
                          NgayKhamThuc = $"từ {a.NgayKhamThucTeTu.Value.ToString("HH:mm")} đến {a.NgayKhamThucTe.Value.ToString("HH:mm dd/MM/yyyy")}",
                          NguoiLienHe = $"{d.NguoiLienHe} - {d.DienThoai}",
                          SoLuotKham = $"{b.SoLuotKham} người",
                          TenCongTy = d.TenCongTy,
                          NguoiLapPhieu = a.CreatedById,
                          NguoiDuyetPhieu = a.NguoiDuyetId,
                          ChiNhanhId = c.ChiNhanhId,
                          DaCoSo = b.CoSo ?? false ? "Đã có số" : "Chưa có số"
                      };

            var CUONG = don.AsEnumerable();

            Guid? nguoiDuyet = Guid.Empty;
            if (sign != null)
            {
                nguoiDuyet = sign.SignerId;
            };

            var _cuong = CUONG.Select(s => new DonKSKCuong()
            {
                TenDonVi = donVi.Find(f => f.Id == s.ChiNhanhId)?.CongTyChuQuan.ToUpper(),
                KhoaPhong = donVi.Find(f => f.Id == s.ChiNhanhId)?.Name.ToUpper(),
                SoPhieu =   $"/{donVi.Find(f => f.Id == s.ChiNhanhId)?.Code}-CL",
                DiaChi = s.DiaChi,
                NgayIn = s.NgayIn,
                NgayKhamThuc = s.NgayKhamThuc,
                NguoiLienHe = s.NguoiLienHe,
                SoLuotKham = s.SoLuotKham,
                TenCongTy = s.TenCongTy,
                NguoiLapPhieu = userLst.Find(f => f.Id == s.NguoiLapPhieu)?.FullName,
                UrlNguoiLapPhieu = userLst.Find(f => f.Id == s.NguoiLapPhieu)?.SignatureUrl,
                NguoiDuyetPhieu = userLst.Find(f => f.Id == nguoiDuyet)?.FullName,
                UrlNguoiDuyetPhieu = userLst.Find(f => f.Id == nguoiDuyet)?.SignatureUrl,
                DaCoSo = s.DaCoSo
            }).ToList();

            var donKhac = from a in _donHangKhamSucKhoeKhacRepository.Table
                          where a.DonHangId == id
                          select new
                          {
                              Stt = "",
                              NhanSuId = a.NhanSuId,
```

```csharp
                                TenNhanVien = "",
                                NhiemVu = a.ChuyenMon,
                                GhiChu = a.GhiChu,
                                KhoaPhong = a.KhoaPhong
                            };

                var CT = donKhac.AsEnumerable();
                var _chitiet = CT.Select(s => new DonKSKChiTiet()
                {
                    Stt = s.Stt,
                    TenNhanVien = s.NhanSuId != null ? userLst.Find(f => f.Id == s.NhanSuId)?.FullName : "",
                    KhoaPhong = s.KhoaPhong,
                    NhiemVu = s.NhiemVu,
                    GhiChu = s.GhiChu,
                }).ToList();

                int dem = 0;
                foreach (var item in _chitiet)
                {
                    dem++;
                    item.Stt = $"{dem}";
                }

                List<DonKSK> employees = new List<DonKSK>();
                employees.Add(new DonKSK()
                {
                    Cuong = _cuong.Find(f => 1 == 1),
                    ChiTiet = _chitiet
                });

                return employees;
        }
    }


    public class DonKSK
    {
        public DonKSKCuong Cuong { get; set; }
        public List<DonKSKChiTiet> ChiTiet { get; set; }
    }
    public class DonKSKCuong
    {
        public string TenDonVi { get; set; } = "";
        public string KhoaPhong { get; set; } = "";
        public string SoPhieu { get; set; } = "";
        public string NgayIn { get; set; } = "";
        public string TenCongTy { get; set; } = "";
        public string DiaChi { get; set; } = "";
        public string NgayKhamThuc { get; set; } = "";
        public string SoLuotKham { get; set; } = "";
        public string DaCoSo { get; set; } = "";
        public string NguoiLienHe { get; set; } = "";
        public string NguoiLapPhieu { get; set; } = "";
        public string UrlNguoiLapPhieu { get; set; } = "";
        public string NguoiDuyetPhieu { get; set; } = "";
        public string UrlNguoiDuyetPhieu { get; set; } = "";
    }
    public class DonKSKChiTiet
    {
        public string Stt { get; set; } = "";
        public string TenNhanVien { get; set; } = "";
        public string KhoaPhong { get; set; } = "";
        public string NhiemVu { get; set; } = "xx";
        public string GhiChu { get; set; } = "xx";
    }
}
```