using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static BO.Tools;

namespace BlImplemementation;

internal class ProductImplementatiton :BlApi.IProduct

{

private DalApi.IDal \_dal = DalApi.Factory.Get;

public int Create(BO.Product item)

{

try

{

return \_dal.Product.Create(item.convertToDoProduct());

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public void Delete(int id)

{

try

{

\_dal.Product.Delete(id);

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public BO.Product? Read(int id)

{

try

{

return \_dal.Product.Read(id).convertToBoProduct();

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public BO.Product? Read(Func<BO.Product, bool> filter)

{

try

{

return \_dal.Product.Read(doProduct => filter(doProduct.convertToBoProduct())).convertToBoProduct();

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public List<BO.Product?> ReadAll(Func<BO.Product, bool>? filter = null)

{

try

{

return \_dal.Product.ReadAll(doProduct => filter(doProduct.convertToBoProduct())).Select(p => p.convertToBoProduct()).ToList();

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public List<BO.SaleInProduct> SalesInProduct(int productId, bool IsPreferredCustomer)

{

try

{

return \_dal.Sale.ReadAll(s => s.productId == productId && (s.isForEveryOne??false || IsPreferredCustomer) && DateTime.Now >= s.startDateForSale && DateTime.Now <= s.endDateForSale).Select(s => s.convertToBOSaleInProduct()).ToList();

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

public void Update(BO.Product item)

{

try

{

\_dal.Product.Update(item.convertToDoProduct());

}

catch (Exception ex)

{

throw new NotImplementedException();

}

}

}