Maintain Categoy

這資料表比較簡單，可以用以下方式處理

UI	BLL	DAL
Add new Category		
● 收集表單資訊,繫結到 CategoryVm ● 針對 CategoryVm 進行欄位驗證 ● 將 CategoryVm 轉型為 CategoryDto ● 將 CategoryDto 傳送給 CategoryService	CategoryService.Create ● 接收 CategoryDto ● 將 CategoryDto 轉型為 CategoryEntity ● 由於 CategoryEntity 會保持不變性，所以若必填欄位居然沒值,在此會錯誤 ● 檢查 CategoryName 是否唯一 ● 將 CategoryEntity 傳送給 CategoryRepository	CategoryRepository.Create ● 接收 CategoryEntity ● 組合出 sql statememt / SqlParameter[] ● 執行 sql statement ,建立一筆新記錄
由於上述功能簡單，可以簡化如下		
● 收集表單資訊,繫結到 CategoryVm ● 針對 CategoryVm 進行欄位驗證 ● 將 CategoryVm 轉型為 CategoryEntity 將 CategoryEntity 傳送給 CategoryService	CategoryService.Create ● 接收 CategoryEntity ● 檢查 CategoryName 是否唯一 ● 將 CategoryEntity 傳送給 CategoryRepository	CategoryRepository.Create 與上相同

Edit Category		
● 叫用 CategoryService.Get(id) ● 收到 CategoryDto ● 將 CategoryDto 轉型為 CategoryVm ● 繫結到表單	CategoryService.Get(id) ● 叫用 CategoryRepository(id) ● 收到 CategoryEntity ● 轉型為 CategoryDto 並傳回	CategoryRepository.Get(id) ● 組合出 sql statememt / SqlParameter[] ● 傳回 CategoryEntity
由於上述功能簡單, 可以簡化如下		
● 叫用 CategoryService.Get(id) ● 收到 CategoryEntity ● 將 CategoryEntity 繫結到表單	CategoryService.Get(id) ● 叫用 CategoryRepository(id) ● 收到 CategoryEntity 並傳回	CategoryRepository.Get(id) 與上相同
Update Category		
處理方式與 Add Category 相似	CategoryService.Update 處理方式與 Add Category 相似	CategoryRepository.Create 處理方式與 Add Category 相似

Display Categories		
● 叫用 CategoryService.Search() ● 收到 IEnumerable<CategoryDto> ● 轉型為 IEnumerable<CategoryVm> ● 繫結到 DataGridView	CategoryService.Search() ● 叫用 CategoryRepository.Search() ● 收到 IEnumerable<CategoryEntity> ● 轉型為 Ienumerable<CategoryDto>並傳回	CategoryRepository.Search() ● 傳回 IEnumerable<CategoryEntity>
由於上述功能簡單，可以簡化如下		
● 叫用 CategoryService.Search() ● 收到 IEnumerable<CategoryVm> 繫結到 DataGridView	CategoryService.Search() ● 叫用 CategoryRepository.Search() ● 收到 IEnumerable<CategoryEntity>並傳回	CategoryRepository.Search() 與上相同

Maintain User

如果加入新會員與編輯會員的表單欄位不同,在註冊新會員會有 Password,Confirm Password,在編輯時,通常不會容許編輯 Account, Password(修改密碼會另外做一頁)，可以用以下方式處理

UI	BLL	DAL
Register new User		
● 收集表單資訊,繫結到 UserCreateVm ● 它包含 password,confirmPassword ● 針對 UserCreateVm 進行欄位驗證 ● 將 UserCreateVm 轉型為 UserCreateDto ● 它不包含 ConfirmPassword ● 將 UserCreateDto 傳送給 UserService	UserService.Create ● 接收 UserCreateDto ● 將 UserCreateDto 轉型為 UserEntity ● 轉換時,視需要將 Password 加密 ● 由於 UserEntity 會保持不變性, 所以若必填欄位居然沒值,在此會錯誤 ● 檢查 Account 是否唯一 ● 將 UserEntity 傳送給 UserRepository	UserRepository.Create ● 接收 UserEntity ● 組合出 sql statememt / SqlParameter[] ● 執行 sql statement ,建立一筆新記錄
Edit User		
● 叫用 UserService.Get(id) ● 收到 UserEditDto ● 將 UserEditDto 轉型為 UserEditVm ● 繫結到表單	UserService.Get(id) ● 叫用 UserRepository.Get(id) ● 收到 UserEntity ● 轉型為 UserEditDto(不含 password)並傳回	UserRepository.Get(id) ● 組合出 sql statememt / SqlParameter[] ● 傳回 UserEntity
Update User		
● 收集表單資訊,繫結到 UserEditVm ● 它不包含 password,confirmPassword ● 針對 UserEditVm 進行欄位驗證 ● 將 UserEditVm 轉型為 UserUpdateDto ● 將 UserUpdateDto 傳送給 UserService	UserService.Update ● 接收 UserUpdateDto ● 將 UserUpdateDto 轉型為 UserEntity ● 由於 UserEntity 會保持不變性, 所以需要從 db 取得原本的密碼填進 UserEntity 裡 ● 檢查 Account 是否唯一(如果允許修改) ● 將 UserEntity 傳送給 UserRepository	UserRepository.Update ● 接收 UserEntity ● 組合出 sql statememt / SqlParameter[] 執行 sql statement ,更新記錄

各個 view model, dto, entity 要放在哪一個專案裡?		
● UserCreateVm ● UserEditVm	● UserEntity ● UserCreateDto ● UserUpdateDto	
這三個專案,要如何參考?		
● 參考 BLL ● 參考 SqlDataLayer	不必參考另二個專案,它是軟體的核心 ● UserService ● IUserRepository ● UserEntity ● UserCreateDto ● UserUpdateDto	● 參考 BLL ● UserRepository : IUserRepository

呈現商品清單

它需要 join Categories, Products

UI	BLL	DAL
● 叫用 ProductsView01.Search () ● 收到 IEnumerable<Products01Dto> ● 繫結到 DataGridView 由於沒有業務邏輯，可以考慮直接叫用 Repository class		● ProductsDto01 class ProductsView01.Search() ● 傳回 IEnumerable<Products01Dto> 此功能不是寫在 ProductRepository 裡
Create / Edit Product 功能與 Cateogry 的相似，就不重覆寫		