**LAB 7:**

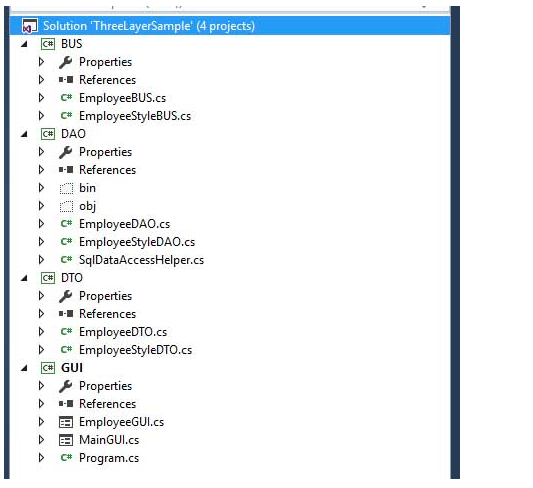
**7.1 Chọn lựa các lớp**

“Biết địch biết ta, trăm trận trăm thắng “. Vậy muốn biết có gì hay thì phải hiểu rõ về nó. Được rồi, nhân dịp đang học nhập môn [công nghệ phần mềm](https://topdev.vn/viec-lam-it/gia-cong-phan-mem-kt3954) ở trường, kiến thức còn nóng hổi nên mình sẽ phân tích về mô hình 3 lớp (3-layer).

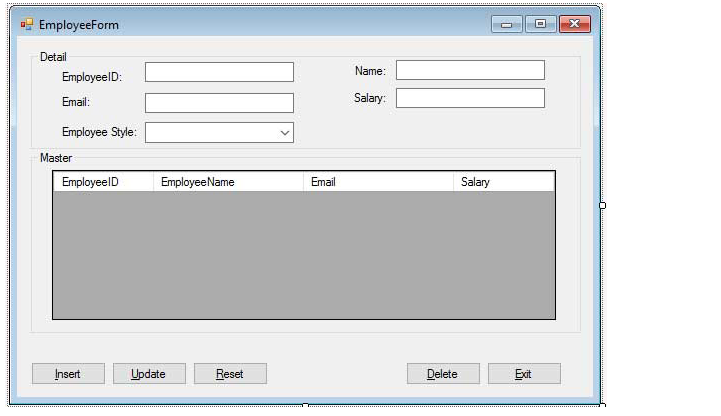
* **Data Access Logic Components :** chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu ([Data Sources](https://topdev.vn/blog/dung-gi-de-luu-tru-data-thay-the-local-storage/)) như XML, file system,… Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.
* **Service Agents :** giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

**Cấu trúc mô hình 3 lớp**

Để hiểu rõ hơn về cấu trúc và cách xây dựng của mô hình 3 lớp, chúng ta cùng tham khảo một ví dụ về mô hình quản lí công nhân gồm các lớp BUS, DAO, GUI. (Các đoạn code sẽ bị lược bỏ bớt )



Đầu tiên là GUI gồm các button insert, update, reset ,delete ,exit .Người dùng sẽ giao tiếp với màn hình giao diện này



Lớp DTO, đây không phải là layer, đây chỉ là 1 gói dữ liệu đươc trao đổi giữa các lớp. Gói dữ liệu này được xây dựng dưới dạng lớp đối tượng. Mỗi một công nhân sẽ mang những thuộc tính sau:

namespace DTO

{

public class EmployeeDTO

{

#region Atrributes

private String \_employeeID;

private String \_name;

private String \_email;

private float \_salary;

private int \_employeeStyle;

#endregion

//.....

Xem tiếp...

Các nghiệp vụ xử lý chính sẽ được đặt ở lớp BUS (hay là BLL)  gồm các nghiệp vụ insert, update, delete, retrieve

namespace BUS

{

public class EmployeeBUS

{

#region 1. Inserting

public static bool InsertEmployee(EmployeeDTO emp)

{

if (EmployeeDAO.CheckEmployeeByID(emp.EmployeeID)==true

&amp;&amp;EmployeeStyleDAO.CheckEmployeeStyleByID(emp.EmployeeStyle)==false)

{

return false;

}

return EmployeeDAO.InsertEmployee(emp);

}

#endregionhttps://techtalk.vn/wp-admin/post-new.php#

//2. Updating

//3. Deleting

//4. Retrieving

}

}

Xem tiếp...

Và cuối cùng là lớp DAO ( hay là DAL ). Truy  vấn đến cơ sở dữ liệu

{

public class EmployeeDAO

{

#region 1. Inserting

public static bool InsertEmployee(EmployeeDTO emp)

{

bool result=false;

try

{

// Create List Sql Parameter

List sqlParams = new List();

sqlParams.Add(new SqlParameter("@EmployeeID", emp.EmployeeID));

sqlParams.Add(new SqlParameter("@Name", emp.Name));

sqlParams.Add(new SqlParameter("@Email", emp.Email));

sqlParams.Add(new SqlParameter("@Salary", emp.Salary));

sqlParams.Add(new SqlParameter("@EmployeeStyle", emp.EmployeeStyle));

// Call Store Procedure

int n = SqlDataAccessHelper.ExecuteNoneQuery("spInsertEmployee", sqlParams);

if (n == 1)

result = true;

}

catch (Exception ex)

{

throw ex;

}

return result;

}

#endregion

// 2. Updating

// 3. Deleting

// 4. Retrieving

Xem tiếp...

Vì đây là ví dụ mô phỏng nên tôi chỉ chú trọng đến cách cấu trúc một chương trình sử dụng mô hình 3 lớp. Cảm ơn đã đọc hết.

**Đừng bỏ lỡ những bài viết hay về lập trình hướng đối tượng:**

* [Mô hình quan hệ – thực thể (Entity – Relationship Model) là gì?](https://topdev.vn/blog/mo-hinh-quan-he-thuc-the-la-gi/)
* [Lập trình hướng chức năng đang thống trị mảng UI với Pure Views](https://topdev.vn/blog/lap-trinh-huong-chuc-nang-dang-thong-tri-mang-ui-voi-pure-views/)
* [Cấu trúc 3 lớp trong C# .Net](https://www.codeproject.com/Articles/36847/Three-Layer-Architecture-in-C-NET)