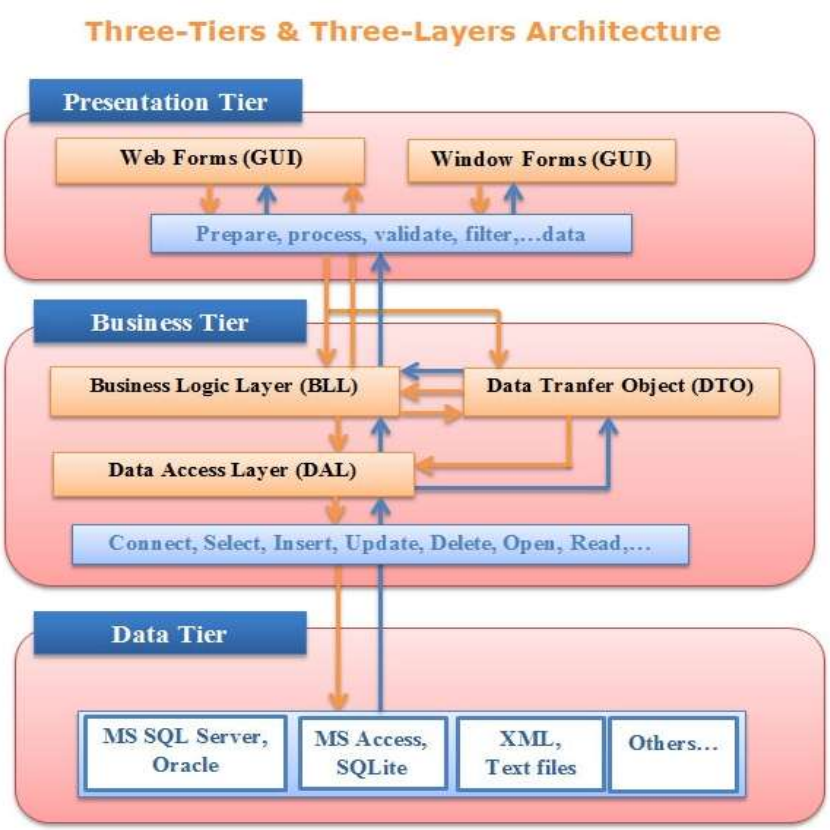
TỔNG QUAN VỀ MÔ HÌNH 3 LỚP

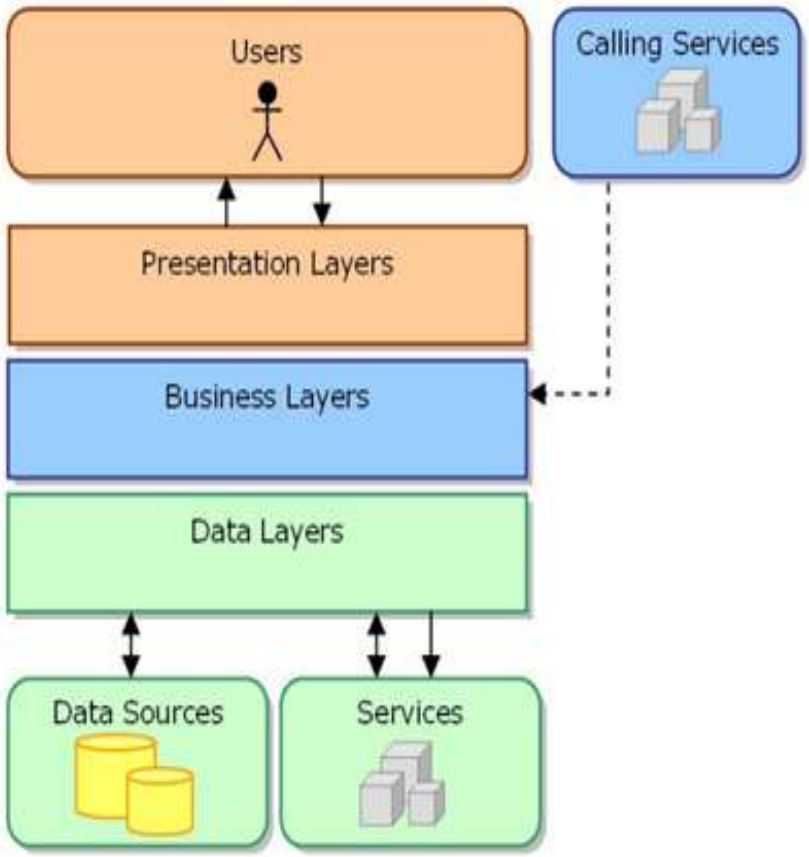
Người soạn: ThS. Lê Trí Thành

Đơn vị: Bộ môn Hệ thống thông tin- Khoa Công nghệ thông tin

Tier (Tầng) và Layer (Lớp)



Three Tiers



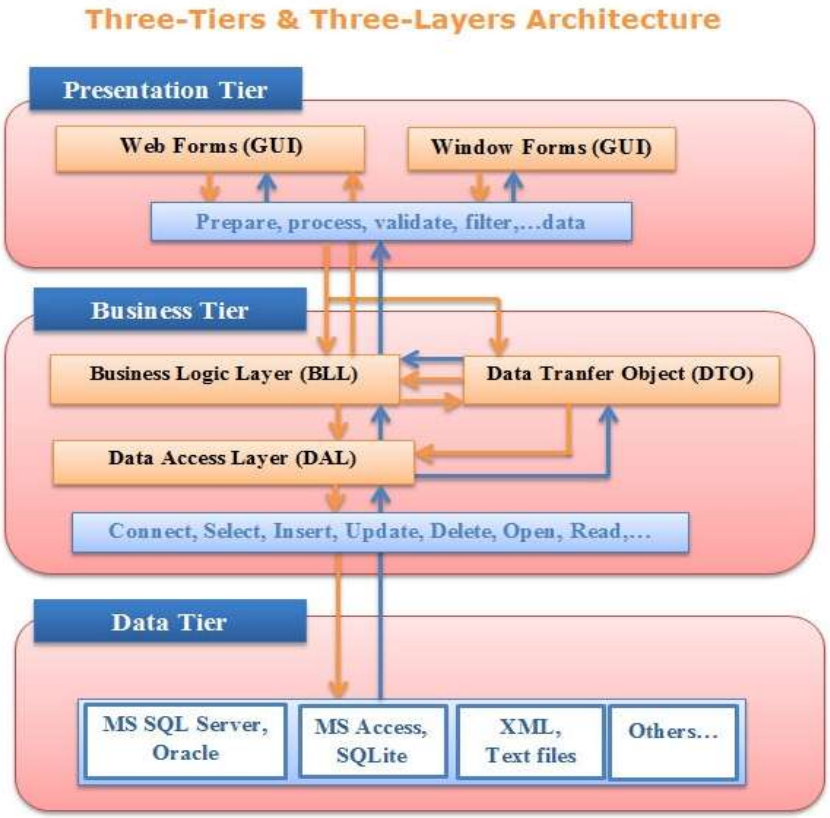
Three Layers

Tier (Tầng) và Layer (Lớp)

- Tier (Tầng) cho chúng ta thấy sự tách biệt vật lý với nhau, những tầng này có thể nằm cùng một nơi hay các nơi khác nhau. Trên thực tế, các ứng dụng lớn thì Database sẽ nằm ở một Server, các API hay Web Service nằm một Server khác và ứng dụng thì chạy ở Client.

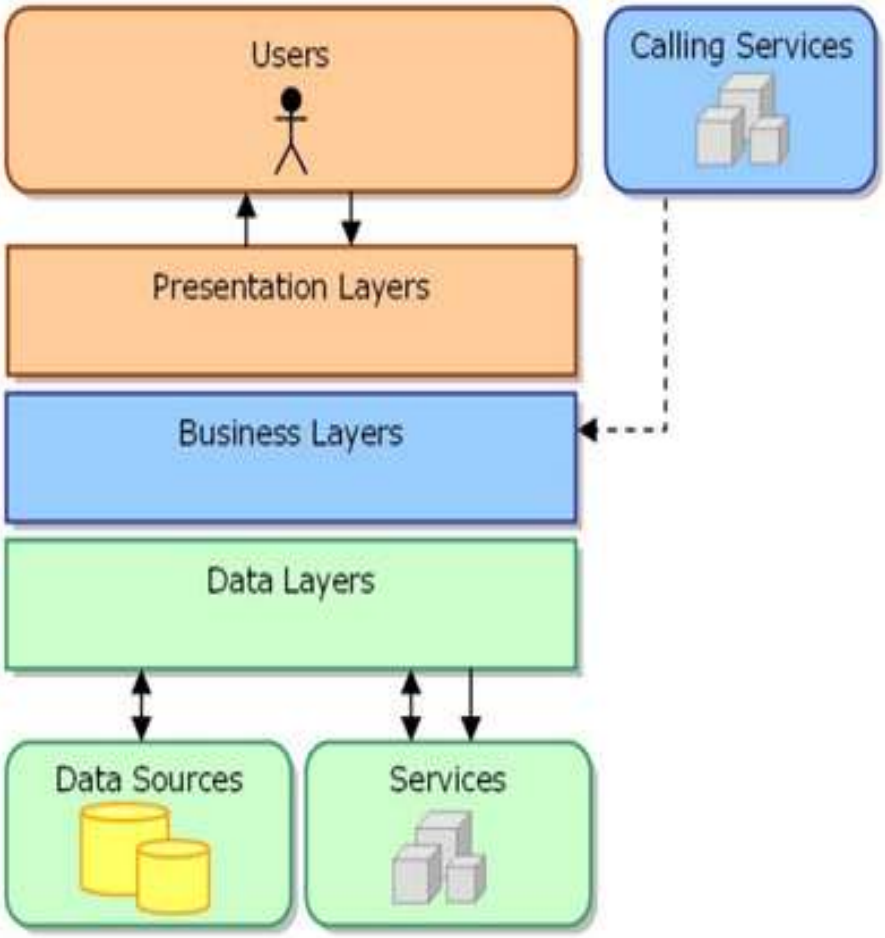
- Layer (Lớp) thể hiện tính logic và là một thành phần của Tier, chúng thường nằm chung một nơi nhưng ở các namespace khác nhau.

Mô hình 3 tầng (Three Tiers)



+ Presentation tier bao gồm các thành phần phần xử lý giao diện Graphic User Interface (GUI)
+ Business tier gồm các thành phần Business Logic Layer (BLL), Data Access Layer (DAL) và Data Tranfer Object (DTO)
+ Data tier: lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,...

Mô hình 3 lớp (Three Layers)



+ Presentation Layer (GUI): Thành phần giao diện, là các form của chương trình tương tác với người sử dụng.
+ Business Logic Layer (BLL): Xử lý các nghiệp vụ của chương trình như tính toán, xử lý hợp lệ và toàn vẹn về mặt dữ liệu.
+ Data Access Layer (DAL): Tầng giao tiếp với các hệ quản trị CSDL

Presentation Layers (GUI)

Presentation Layers: Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng.

- Trong.NET ta có thể dùng Windows Forms, ASP.NET hay Mobile Forms để hiện thực lớp này.

- Lớp này không sử dụng trực tiếp các dịch vụ của lớp Data Access mà nên sử dụng thông qua các service của lớp Business Logic

Presentation Layers (GUI)

Lớp này có 2 thành phần chính là User Interface Components và User Interface Process Components.

- UI Components: là những phần tử chịu trách nhiệm thu thập và hiển thị thông tin cho người dùng cuối. Trong ASP.NET thì những thành phần này có thể là các TextBox, các Button, DataGrid…

- UI Process Components: là thành phần chịu trách nhiệm quản lý các qui trình chuyển đổi giữa các UI Components. Ví dụ chịu trách nhiệm quản lý các màn hình nhập dữ liệu trong một loạt các thao tác định trước như các bước trong một Wizard…

Business Logic Layer (BLL/ BUS)

Business Logic Layer: Đây là lớp xử lý chính các dữ liệu trước khi được đưa lên hiển thị trên màn hình hoặc xử lý các dữ liệu trước khi chuyển xuống Data Access Layer để lưu vào cơ sở dữ liệu.

- Lớp này có chức năng kiểm tra ràng buộc, các yêu cầu nghiệp vụ, tính toán, xử lý các yêu cầu và lựa chọn kết quả trả về cho Presentation Layers.

Business Logic Layer (BLL/ BUS)

Lớp này có các thành phần chính là Business Components, Business Entities và Service Interface.

- **Service Interface:** là giao diện lập trình mà lớp này cung cấp cho lớp **Presentation** sử dụng. Lớp **Presentation** chỉ cần biết các dịch vụ thông qua giao diện này mà không cần phải quan tâm đến bên trong lớp này được hiện thực như thế nào.
- **Business Entities:** là những thực thể mô tả những đối tượng thông tin mà hệ thống xử lý. Các Business Entities này cũng được dùng để trao đổi thông tin giữa lớp **Presentation** và lớp **Data Layers**.
- **Business Components:** là những thành phần chính thực hiện các dịch vụ mà **Service Interface** cung cấp, chịu trách nhiệm kiểm tra các ràng buộc logic (constraints), các qui tắc nghiệp vụ (Business Rules), sử dụng các dịch vụ bên ngoài khác để thực hiện các yêu cầu của ứng dụng.

Data Access Layer (DAL/ DAO)

Data Access Layer: Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng như đọc, lưu, cập nhật cơ sở dữ liệu.

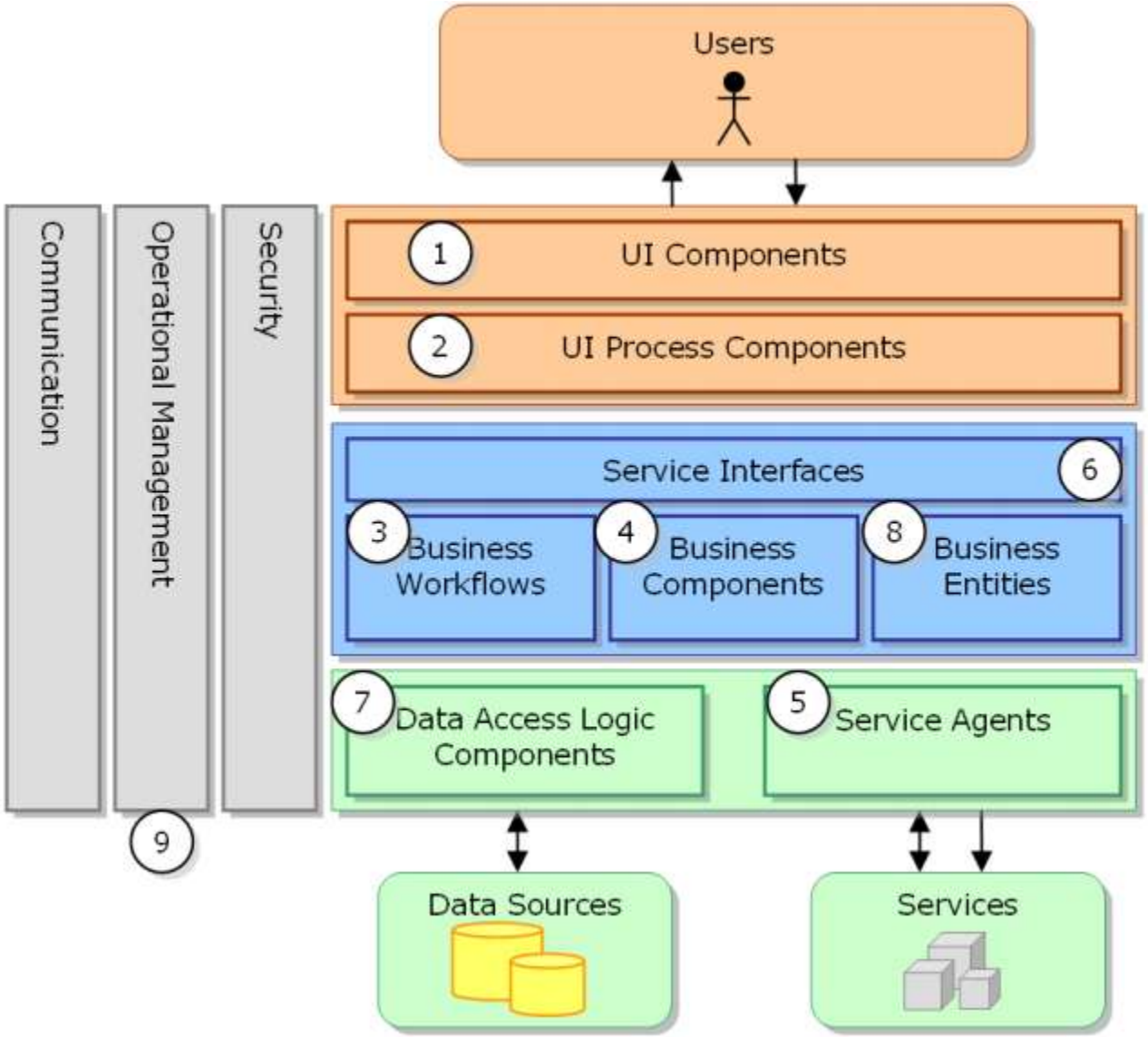
Data Access Layer (DAL/ DAO)

Lớp này có các thành phần chính là Data Access Logic, Data Sources, Servive Agents).

- Data Access Logic Components (DAL) là thành phần chính chịu trách nhiệm lưu trữ vào và truy xuất dữ liệu từ các nguồn dữ liệu – Data Sources như RDMBS, XML, File systems…. Trong .NET Các **DAL** này thường được hiện thực bằng cách sử dụng thư viện ADO.NET để giao tiếp với các hệ cơ sở dữ liệu hoặc sử dụng các O/R Mapping Frameworks để thực hiện việc ánh xạ các đối tượng trong bộ nhớ thành dữ liệu lưu trữ trong CSDL.

- Service Agents: là những thành phần trợ giúp việc truy xuất các dịch vụ bên ngoài một cách dễ dàng và đơn giản như truy xuất các dịch vụ nội tại.

Mô hình 3 lớp đầy đủ

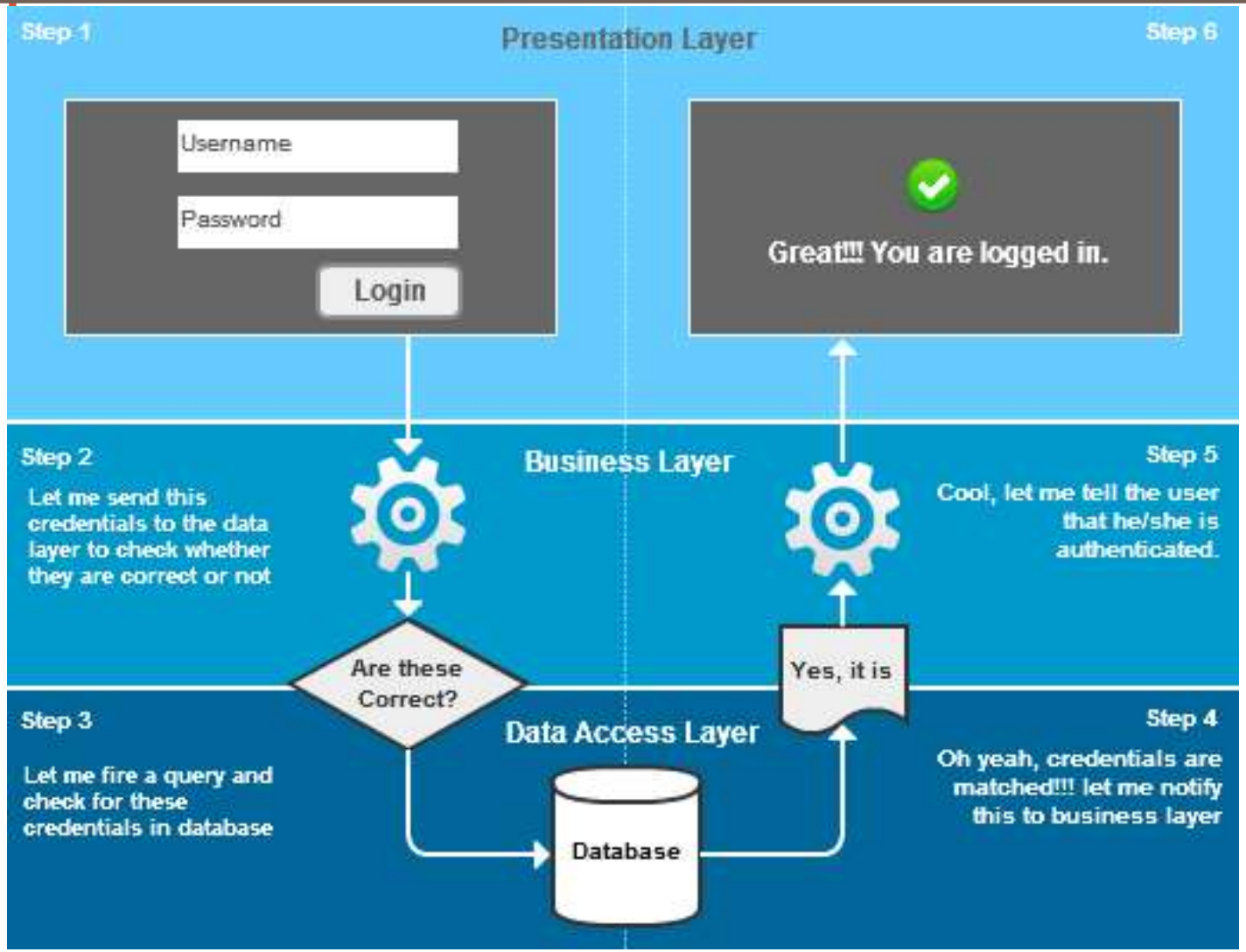


Hoạt động của mô hình

Đối với 3-Layer, yêu cầu được xử lý tuần tự qua các layer:

1) User giao tiếp với Presentation Layers (GUI) để gửi đi thông tin và yêu cầu. Tại layer này, các thông tin sẽ được kiểm tra, nếu được chấp nhận chúng sẽ được chuyển xuống Business Logic Layer (BLL).
2) Tại BLL, các thông tin sẽ được tính toán theo đúng yêu cầu đã gửi, nếu không cần đến Database thì BLL sẽ gửi trả kết quả về GUI, ngược lại nó sẽ đẩy dữ liệu (thông tin đã xử lý) xuống Data Access Layer (DAL).
3) DAL sẽ thao tác với Database và trả kết quả về cho BLL, BLL kiểm tra và gửi nó lên GUI để hiển thị cho người dùng.
4) Một khi gặp lỗi (các trường hợp không đúng dữ liệu) thì đang ở layer nào thì sẽ thông báo lên trên layer cao hơn nó 1 bậc cho tới GUI thì sẽ thông báo cho người dùng biết
5) Các dữ liệu được trung chuyển giữa các Layer thông qua một đối tượng gọi là Data Transfer Object (DTO), đơn giản đây chỉ là các Class đại diện cho các đối tượng được lưu trữ trong Database.

Ví dụ: Hoạt động của mô hình khi đăng nhập



Tổ chức mô hình

Có rất nhiều cách tổ chức cho các thành phần của 3 lớp, thông thường sẽ có các cách sau:

- Cách 1: GUI, BUS, DAL

- Cách 2: GUI, BLL, DAO, DTO

- Cách 3: Presentation, BLL, DAL

Tổ chức mô hình trong lập trình

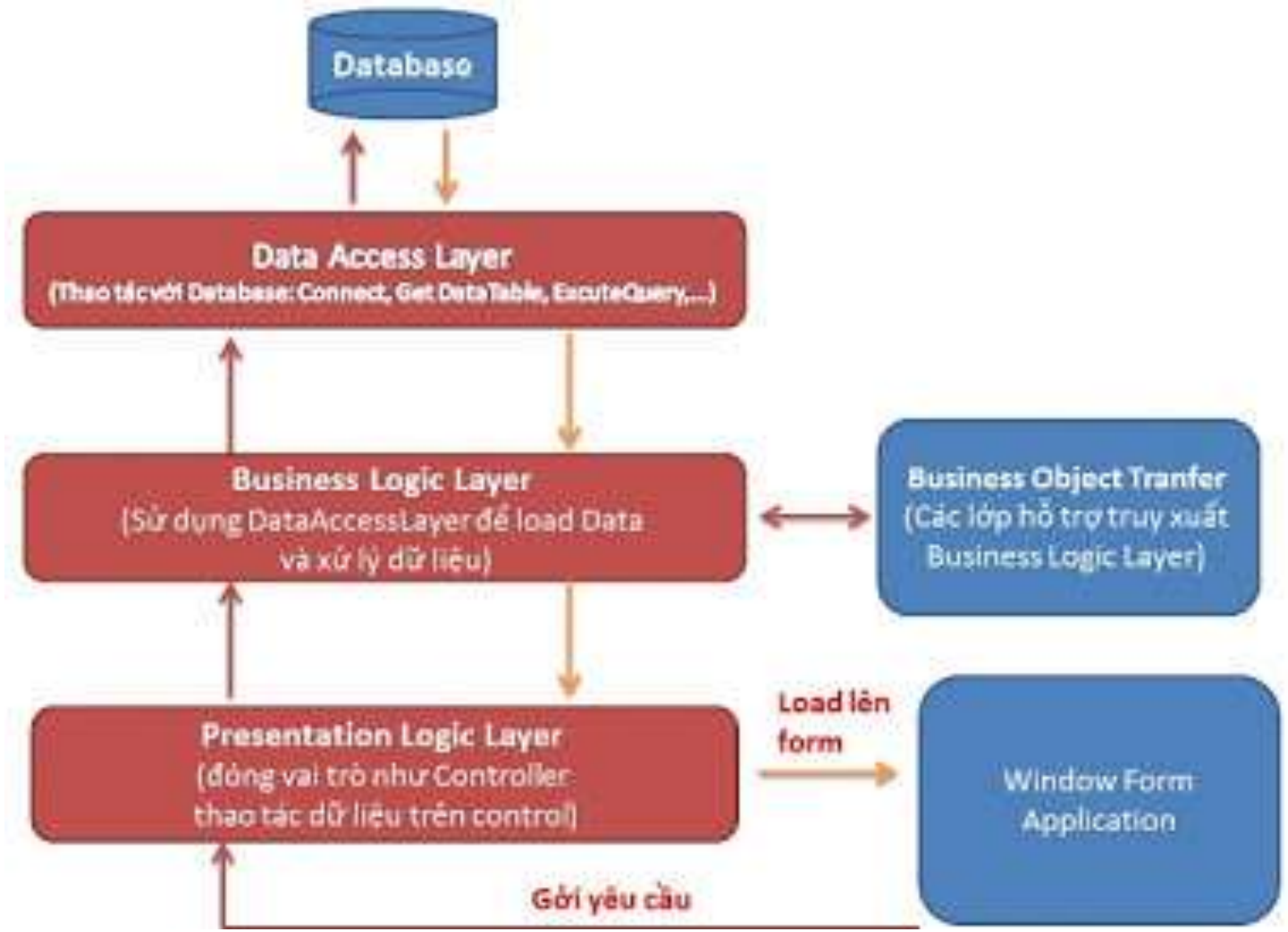
Với mỗi lớp (DAL,BLL) ta tạo 1 project mới kiểu Class Library, sau khi buil ra các dll như: BUS.dll, DAL.dll. Khi đó:

- Tầng GUI là project chính chương trình, vì đặc điểm GUI chỉ thấy BLL nên ta sẽ add references BLL.dll từ tab project vào GUI

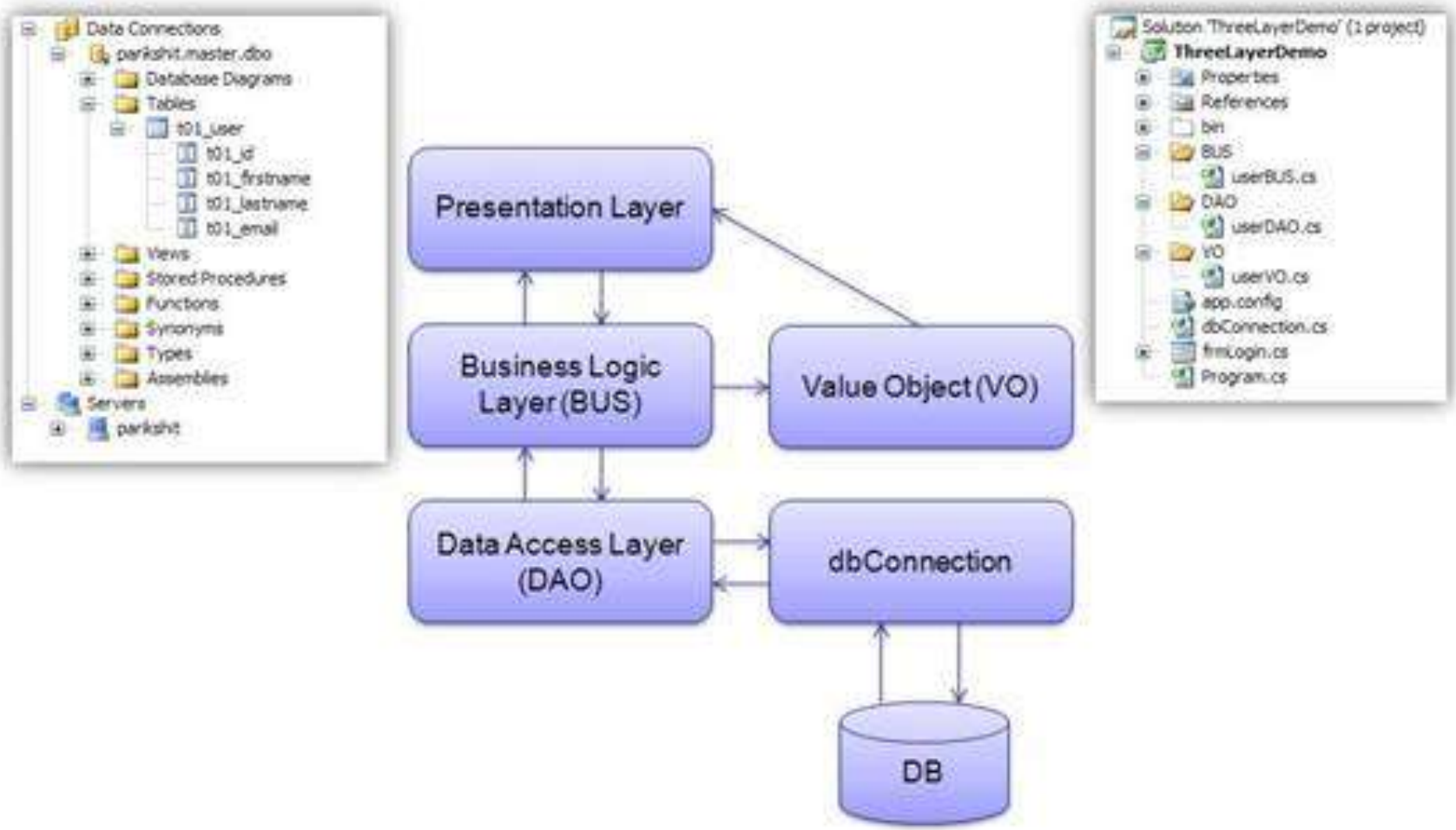
- Tầng BUS chỉ thấy được DAL, ta tiếp tục add references DAL.dll vào BUS

- Tầng DAL giao tiếp được với database nên ta chỉ sử dụng các namespace data provider để tương tác với hệ quản trị CSDL

Tổ chức mô hình trong lập trình



Tổ chức mô hình trong lập trình



Đối tượng SQLCLIENT

+ SqlConnection(<Chuỗi kết nối>): kết nối đến CSDL.
+ Open/ Close thuốc lớp SqlConnection để đóng/ mở kết nối. Lưu ý tại 1 thời điểm chỉ có 1 kết nối đến CSDL.
+ Cấu trúc chuỗi kết nối sử dụng người dùng SQL:
Data Source=<Tên Server>;Initial Catalog=<Tên CSDL>;uid=<Tên người dùng trên SQL>;pwd=<Mật khẩu người dùng SQL>
+ Cấu trúc chuỗi kết nối sử dụng người dùng Windows:
Data Source=<Tên Server>;Initial Catalog=<Tên CSDL>;

Đối tượng SQLCLIENT

VD: Hàm kiểm tra kết nối đến CSDL

CnStr =“Data Source=.;Initial Catalog=ViDu;uid=sa;pwd=123”

```
Public Cn As SqlClient.SqlConnection
Public Function KetNoi(ByVal CnStr As String) As Boolean
        Try
            Cn = New SqlClient.SqlConnection(CnStr)
            Cn.Open()
            KetNoi = True
            Cn.Close()
        Catch ex As Exception
            KetNoi = False
        End Try
    End Function
```

Đối tượng SQLCLIENT

+ SqlDataAdapter (<Câu lệnh SQL>, <Chuỗi kết nối>): tham chiếu đến 1 đối tượng trong CSDL thông qua câu lệnh SQL
+ Fill: hàm thuộc lớp SqlDataAdapter để gán giá trị kiểu Bảng dữ liệu cho 1 biến
Ví dụ: Lấy dữ liệu dạng bảng qua câu lệnh SQL

```
Public Function LayDuLieu(ByVal str As String) As DataTable
        Try
            Dim tbl As New DataTable
            Dim cad As SqlClient.SqlDataAdapter
            Cn.Open()
            cad = New SqlClient.SqlDataAdapter(str, Cn)
            cad.Fill(tbl)
            LayDuLieu = tbl
            Cn.Close()
        Catch ex As Exception
            Cn.Close()
        End Try
    End Function
```

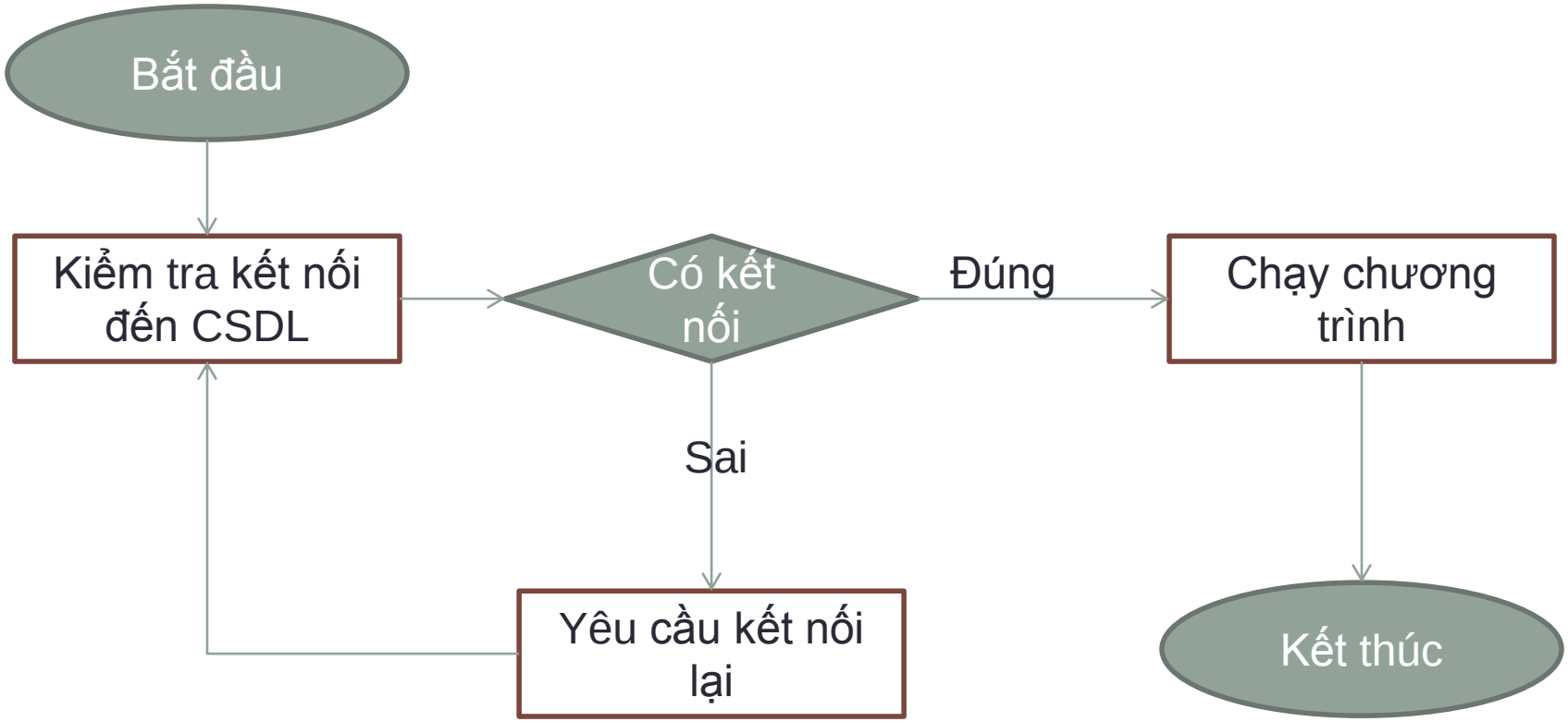
Đối tượng SQLCLIENT

+ SqlCommand(<Câu lệnh SQL>, <Chuỗi kết nối>): tham chiếu đến trình thực thi câu lệnh SQL trong CSDL thông qua câu lệnh SQL
+ ExecuteNonQuery: hàm thuộc lớp SqlCommand để thực thi câu lệnh và trả về số dòng dữ liệu tác động lên
+ ExecuteReader: hàm thuộc lớp SqlCommand để thực thi câu lệnh và trả về đối tượng quản lý dữ liệu chỉ đọc

VD: Thực thi câu lệnh SQL bất kỳ

```
Public Function ThucHiencauLenhSQL(ByVal str As String) As Boolean
        Try
            Cn.Open()
            Dim cmd As New SqlClient.SqlCommand(str, Cn)
            cmd.ExecuteNonQuery()
            ThucHiencauLenhSQL = True
            Cn.Close()
        Catch ex As Exception
            Cn.Close()
            ThucHiencauLenhSQL = False
        End Try
    End Function
```

Kết nối đến CSDL từ ứng dụng



Một số lưu ý

+ Viết thư viện thao tác CSDL dành cho việc kết nối CSDL, thực thi câu lệnh SQL và các thao tác khác liên quan đến dữ liệu.
+ Viết form cấu hình dữ riêng và chỉ gọi chi không kết nối được hoặc khi cần thiết
+ Nên sử dụng 1 biến SqlConnection dùng chung cho toàn chương trình, khi cần thao tác thì mở ra, sau khi thao tác xong thì đóng lại để tránh việc duy trì kết nối thường xuyên đến CSQL và khởi tạo, kết nối liên tục khi thao tác dữ liệu.
+ Viết thư viện kiểm tra sự đúng đắn của dữ liệu riêng: kiểm tra số nguyên, số thực, chuẩn tên gọi, chuỗi,...