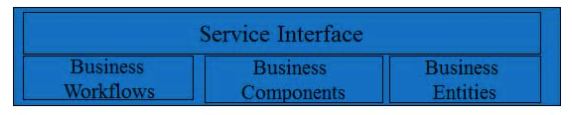
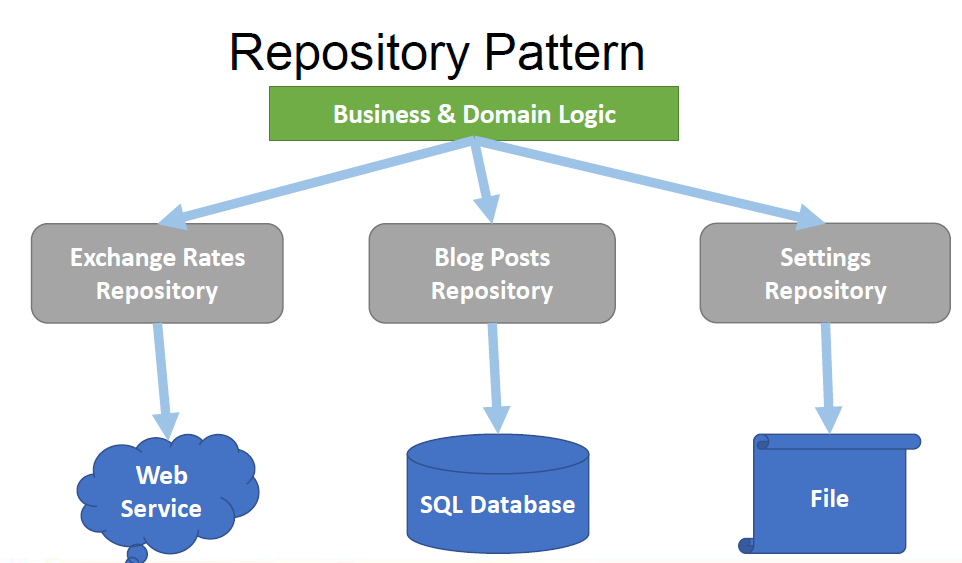
**REPOSITORY PATTERN**

1. Web API: Chỉ làm việc trực tiếp với tầng Service. (Web API là gì, các thành phần… tham khảo tài liệu có sẵn)
2. Service Layer: xử lý nghiệp vụ
   * Đảm bảo việc lấy và xử lý dữ liệu theo yêu cầu nghiệp vụ
   * Tính toán, đảm bảo tính hợp lệ của dữ liệu. Nơi kiểm tra các ràng buộc dữ liệu được chuyển lên từ client.
   * Lớp trung gian giữa Lớp repository và Web API (Controller)

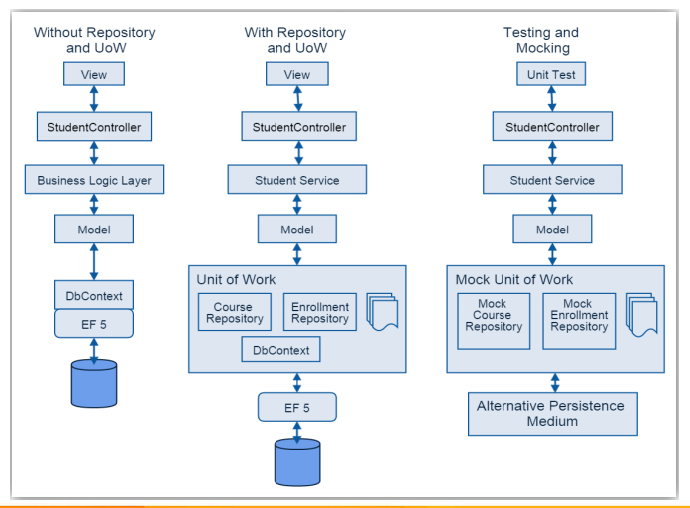


1. Repository Layer: là lớp chứa các class trừu tượng, nơi sẽ tập trung các đoạn mã logic có thể sử dụng chung để tối ưu việc sử dụng code. Unit Of Work được sử dụng chung với tầng này.
   * Giảm thiểu sự trùng lặp code
   * Đảm bảo các code sử dụng chung 1 kiến trúc
   * Đóng gói Code thao tác cơ sở dữ liệu
   * Giả thiểu lỗi
   * Dễ bảo trì
   * Bỏ sự phụ thuộc sử tầng Data (Code thao tác cơ sở dữ liệu) và Tầng Service (xử lý nghiệp vụ)
   * Lớp trung gian giữa Database Access và Service
   * Làm cho chương trình có khả năng Automation Test được



1. Unit of Work: đảm bảo việc thao tác với cơ sở dữ liệu luôn toàn vẹn.
   * Quản lý và kiểm soát danh sách các đối tượng bị thay đổi bởi Logic nghiệp vụ tập trung
   * Đảm bảo các việc như Insert, Update, Delete dữ liệu được thồng nhất và thực thi trong cùng một transaction.
   * Ngăn chặn việc thao tác trực tiếp với database gây lỗi không mong muốn. Xử lý các vấn đề tương tranh dữ liệu.
   * Giúp code nghiệp vụ không cần quan tâm kiểm soát thay đổi trên Entity
2. Data Access Layer: thao tác với cơ sở dữ liệu, truy vấn dữ liệu
3. Models Layer: chứa các class mô tả các thực thể: kiểm dữ liệu, phương thức….
4. Common Layer: nơi chứa các hàm, phương thức, class, enum… có thể sử dụng chung cho tất các các lớp
5. Test Layer: sử dụng phục vụ cho việc kiểm thử code

Kiến trúc áp dụng:



Tham khảo thêm tại: https://techbrij.com/generic-repository-unit-of-work-entity-framework-unit-testing-asp-net-mvc