**DataAccessLayer**

1. Infrastructure
   1. DbFactory: collegamento con il DB. Prende in input un DataContext (che rappresenta la tipologia DB utilizzatoche sia EF, NHibernate, ADO.NET
   2. IRepository: espone operazioni CRUD per il DB
   3. UnitOfWork: funge tipo da proxy. Il servizio al posto di parlare con il Repository, committa alla UnitOfWork
   4. RepositoryBase: Tutti i repo specifici erediteranno da tale classe e dunque implementeranno IRepository. I metodi contenuti in esso sono virtual. Dunque tutti i repo possono fare override di una specifica operazione
2. -Repository
   1. CategoryRepository : RepositoryBase<Category>, ICategoryRepository ..Supporta sia le operazioni base sia implementa ICategoryRepository che include operazioni specifiche per quel tipo di entità
3. –StoreEntities e StoreSeedData : rispettivamente il DataContext e l’inizializzatore del DB

**Service Layer**

In questo layer vengono esposte le operazioni ai nostri MVC Controllers. E' esattamente qui che risiede l'implementazione della logica di business.

Un buon pattern per modellare il Service Layer è il Facade

**PresentetionLayer**

1. **Global.sax** si occupa dell'inizializazione del DB e di tutte le operazioni che devono essere effettuate la prima volta. **Autofac** e BundleConfig per bootstrap
2. **Web.config** contiene la connection string
3. **Autofac**: tutti i servizi hanno nel costruttore l'interfaccia del Repository che deve essere inniettata. I servizi stessi in seguito saranno inniettati nei controllers. Tale DependencyInjection viene realizzata con Autofac.
4. Le cartelle css,font, e JS costituiscono il framework Bootstrap
5. **BundleCOnfig** serve per referenziare le risorse di bootstrap
6. **ViewModel:** serve per mappare effettivamente le informazioni delle Entity del DB che vogliamo realmente visualizzare nel browser
7. **Automapper:** serve per mappare le property del ViewModel automaticamente a quelle del Model. Devono avere lo stesso identico nome. Definiremo due mapping uno dal ViewModelTODomain e l'altro DomainToViewModel
8. **Bootstrapper:** invoca il SetAutofacContainer (registra dinamicamente le componenti) e l'Automapper
9. **Controller**: i servizi sono iniettati all'interno del controller per ogni richiesti e i dati ritornati saranno mappati al ViewModel. Qui ci sono implemtate le funzioni che vediamo nella pagina web