**N-TIER ÖĞRENME ÖRNEK PROJESİ V1.0**

**Konu:** Bir kitapçı dükkanın ihtiyaç duyabileceği belirli işleri yapabilen, katmanlı yapıda, database ve orm bağımsız, farklı clientlara hizmet edebilen, test edilebilirliği ve okunabilirliği yüksek bir proje.

**Proje Adı:** Kitapcim v1.0

**Gereksinim Analizi:** -- ÖRNEK SENARYO ---

Yeni açılan bir kitapçı, bazı işlerini bilgisayar üzerinden takip etmek istiyor. Zaman ve maliyet bakımından tüm işlerini bilgisayara taşımanın zorluğunun farkında olduğu için belirli işlevleri yerine getiren bir uygulama istiyor. Bu uygulamanın zamanla olgunlaşmasıyla diğer isteklerini de karşılayabilecek bir modülerlikte olmasını, ayrıca şuan için windows üzerine yüklenen bir uygulama olmasını, ancak ilerde webclient, tablet ve ya akıllı telefonlar üzerinden mobil olarak erişim sağlanabilmesinin sağlanmasını istiyor.

Bu amaçla ilk olarak stok ve satış modülü istemekte. Uygulamda olmasını istediği işlevler şunlar:

* Kitapların yazarlarına ait bilgileri girebileceği/değiştirebileceği bir ekran
* Kitapların yayın evlerini girebileceği/değiştirebileceği bir ekran
* Kitaplara ait fiyat, tür, yazar, yayıncı v.b. bilgilerinin girileceği/değiştirileceği/arama- listeleme yapan bir ekran
* Müşterilerin bilgilerini girebileceği bir ekran
* Müşterilerin siparişlerini girebileceği bir ekran

Yukarıda belirtilen ana işlevler müşterinin beklentisidir.

**Yazılım Gereksinimleri**

**Geliştirme Ortamı ve Database:**  Visual Studio, C# .net framework 3.5 ve üstü. Database Sql Server 2008 express edition R2.

**Kod Yönetimi:** Github üzerinden <https://github.com/ersinokur/Kitapcim/>

* Bir kitabın birden fazla yazarı olabileceği için many-to-many ilişkiyi tutan bir tablo
* Kitap birden fazla türe ait olabileceği için kitaptür ara tablosu many-to many
* Kitapların tür bilgisini tutan bir tablo
* Yazarların bilgisinin depolandığı bir tablo
* Kitap tablosu
* Yayıncı tablosu
* Müşteri tablosu
* Adres tablosu
* Sipariş parent ve child tabloları

**Tablolar**

GitHub üzerinde olulturulmuş veritabanı ve bu db içindeki tablolar ile ilişkileri bulunmaktadır.

**Yazılım Mimarisi:**

Kodlamanın, Model View Presenter tasarım deseninde olması planlanmıştır. Test aşamaları amaç mimarinin öğrenilmesi olduğundan dikkate alınmamıştır.

Katmanlar:

Data Katmanı (DAO): En alt katman olup, DB ve DAO erişim yöntemleri bağımsız olarak oluşturulması düşünülmüştür.

Domain Katmanı: Tüm katmanlarla dikey haberşen bu katmanda poco sınıfları bulunmaktadır.

Business Layer: Uygulamanın iş mantığı bu katmanda bulunmaktadır.

Bu üç ana katman hangi client yöntemi seçilirse seçilsin ek bir çaba sarf etmeden aynı kalabilmeli,

Yazılımcılar presenter katmanına yoğunlaşmalıdır.

İlk etapta windows forms yapısı kullanılacağından presenter buna göre şekillendirelecektir.