# metin içeren bir resim Açıklama otomatik olarak oluşturuldu

# **1. Özet**

Nesne tabanlı programlama dersi kapsamında yapmış olduğumuz proje online kitap, CD ve dergi sipariş sistemidir. Proje kapsamında c# dili kullanılmıştır. Nesne tabanlı programlama mantığı ile sınıflar, miras, kalıtım vb. yapılar kullanılmıştır. Sistem kayıtlı bir müşterinin giriş yaparak almak istediği ürünleri tercih ettiği ödeme sistemi ile siparişini tamamladığı bir sistemdir.

# **2. Tasarım**

## **2.1. UML Diyagram**

UML(Unfield Modelling Language) iş sistemlerinin modellenmesi amacıyla ortaya çıkmış bir dildir. Birçok farklı modeli bulunmaktadır. Projemiz nesneye dayalı bir proje olduğundan ve içerisinden birden çok sınıf (class) barındırdığından class diagram modeli kullanılmıştır. UML diyagram ile sınıfların birbirleri arasındaki ilişkileri gösterilmiştir. (türetilme, içerme, vb.)

Şekil UML Diagram

## **2.2. Projeden Ekran Görüntüleri**ekran görüntüsü içeren bir resim Açıklama otomatik olarak oluşturuldu

Şekil 2Giriş sayfası

Şekil 2’de giriş sayfası görüntülenmektedir. Kullanıcı tipine göre admin veya üye girişi yapılabilir. Eğer kullanıcı üye değil ise kayıt ol butonu ile üye olur.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3Admin girişi

Şekil 3’te admin girişi gösterilmektedir. Projemizde bir tane admin bulunmaktadır. Kullanıcı adı ‘admin’ ve şifre ‘admin’ olarak ayarlanmıştır.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 4Admin sayfası

Şekil 4’te admin sayfası görüntülenmektedir. Admin bu sayfada ürünlerle ve kullanıcılarla ilgili işlemler yapmaktadır.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 5 Ürün güncelleme sayfası

Şekil 5’te admin dergi kategorisindeki bir ürünün güncellemesini yapmaktadır.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 6 Üye girişi

Şekil 6’da üye girişi gösterilmektedir.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 7 Üye sayfası

Şekil 7’de üye sayfası gösterilmektedir. Ürünler kategorisinden istediği ürünü seçerek alışverişine devam etmektedir.

ekran görüntüsü, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 8 Sipariş veme sayfası

Şekil 8’de alışveriş sayfası gösterilmektedir. Üye burada kitap siparişini vermektedir.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 9 Sipariş ödeme sayfası

Şekil 9’da ödeme sayfası gösterilmektedir. Üye bu sayfada istediği ödeme yöntemi ile siparişini tamamlar.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 10 Sipariş tamamlama sayfası

Şekil 10’da sipariş detayı ve verilen sipariş gösterilmektedir.

# **3. Tasarım Kalıpları**

## **3.1. Singleton Design Pattern**

Projemizde tasarım kalıplarından singleton design pattern (tek nesne) kullanılmıştır. İsminden de anlaşılacağı üzere **singleton**tasarım deseni, hazırlayacağımız sınıftan sadece bir örneğinin oluşturulmasını sağlar. Bu sayede nesnenin kopyalanmasını ya da yeni bir tane oluşturmasını engeller ve nesneye ihtiyaç duyulduğunda o nesnenin daha önceden oluşturulan örneği çağırır. Projemizde ‘Login’ sınıfı singleton design pattern kullanılarak oluşturulmuştur. Kod parçacığı aşağıdaki gibidir.

|  |
| --- |
| public sealed class Login  {  private static Login logInstance = null;  private static readonly object Instancelock = new object();  private Customer userLoggedIn = new Customer();//giriş yapan kullanıcı  DateTime logInTime, logOutTime;//giriş-çıkış zamanları için  StringBuilder logFile=new StringBuilder();//log tutan strinbuilder  private Login() { }  //obje döndürme  public static Login GetInstance  {  get  {  lock (Instancelock)  {  if (logInstance == null)  {  logInstance = new Login();  }  return logInstance;  }  }  } |

## **3.2. Factory Method Pattern**

Aynı interface'i veya abstract sınıfı implement etmiş etmiş factory nesnelerinin üretiminden sorumlu pattern dir. Projemizde ProductFactory class’ında bu metod kullanılmıştır. Kod parçacığı aşağıdaki gibidir.

|  |
| --- |
| static public Product FactoryMethod(string choice,string name,double price, int stock,bool bl)  {  long ID = (bl)?(int.Parse(GetID())):0;//unique ID  Product objChosen = null;  if (choice == "Book")  {  objChosen = new Book(name,ID,price,stock);  }  if (choice == "Magazine")  {  objChosen = new Magazine(name, ID, price,stock);  }  if (choice == "MusicCD")  {  objChosen = new MusicCD(name, ID, price, stock);  }  return objChosen;  } |

Customer Class, Login class, Product class

Book class, magazine class, Music class , ShoppingCart class

ItemToPurchase class, ProductFactory class,