

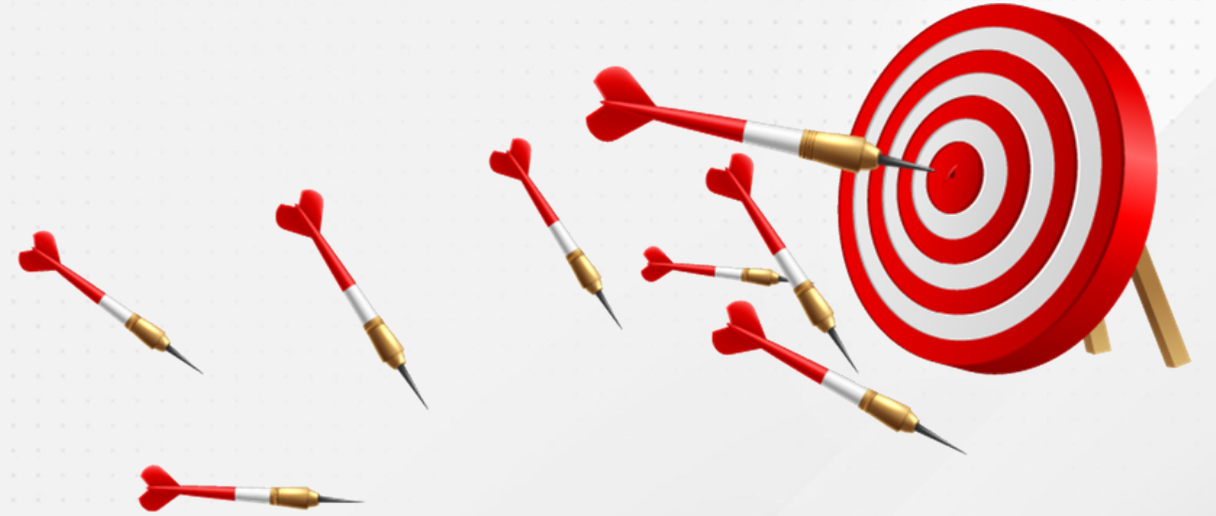


ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
**UZAKTAN EĞİTİM UYGULAMA VE
ARAŞTIRMA MERKEZİ**

İNTERNET PROGRAMCILIĞI

Dersin Genel Hedefleri

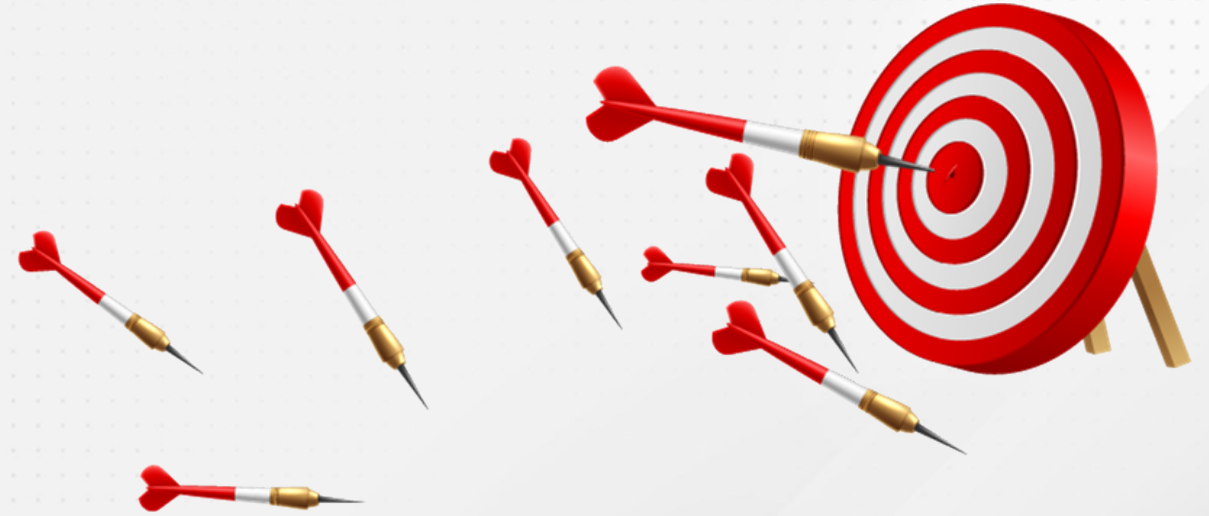
- ASP.NET ile Web Uygulama projesi geliřtirebilme
- Business katmanının oluřturulması



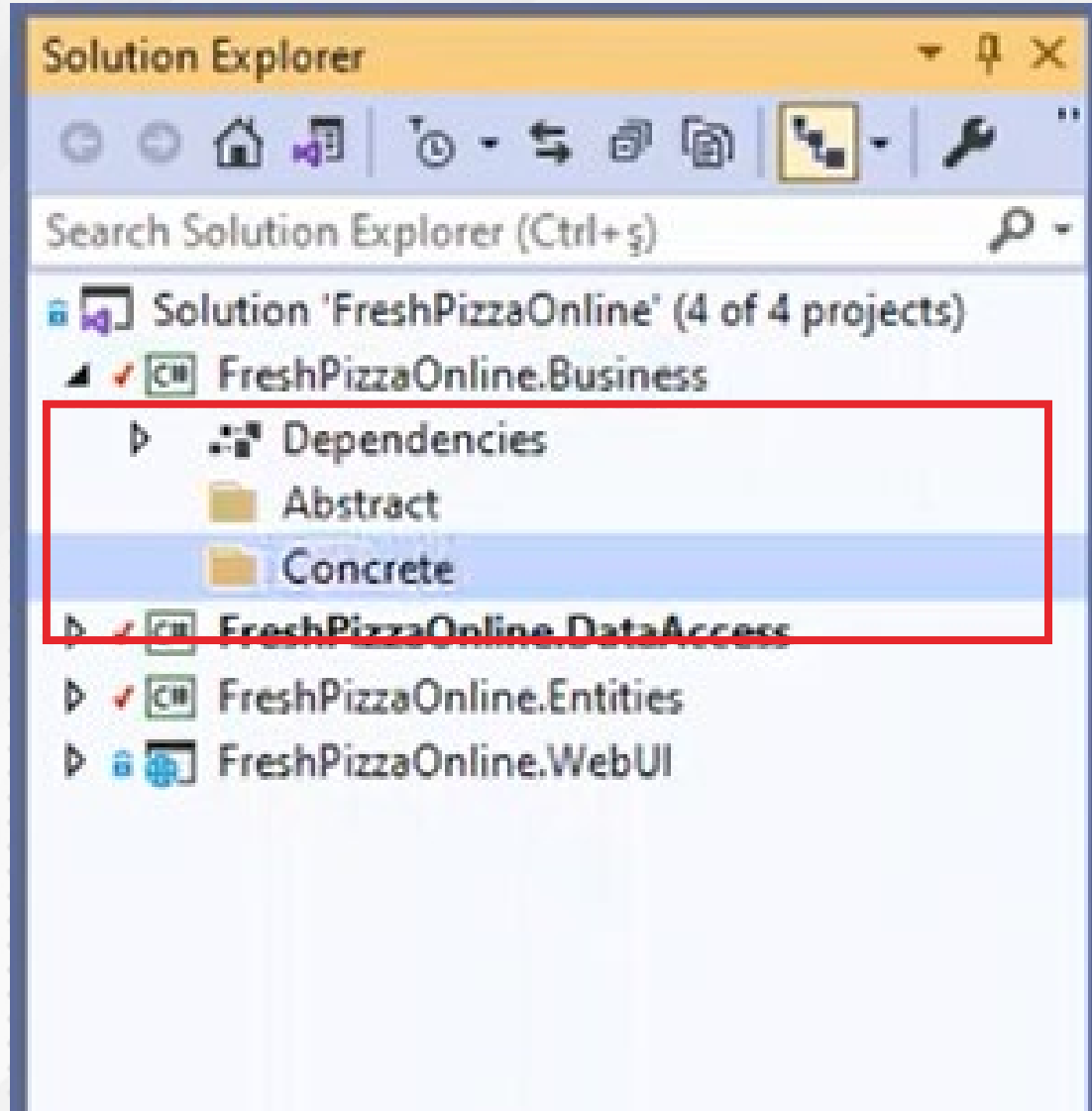
HAFTA 12. ASP.NET İLE WEB UYGULAMA PROJESİ GELİŞTİREBİLME

Bölüm Hedefleri

- Katmanlı Mimari Projesi Oluşturma
- Proje Mimarisine Uygun Business Katmanının Oluşturulması

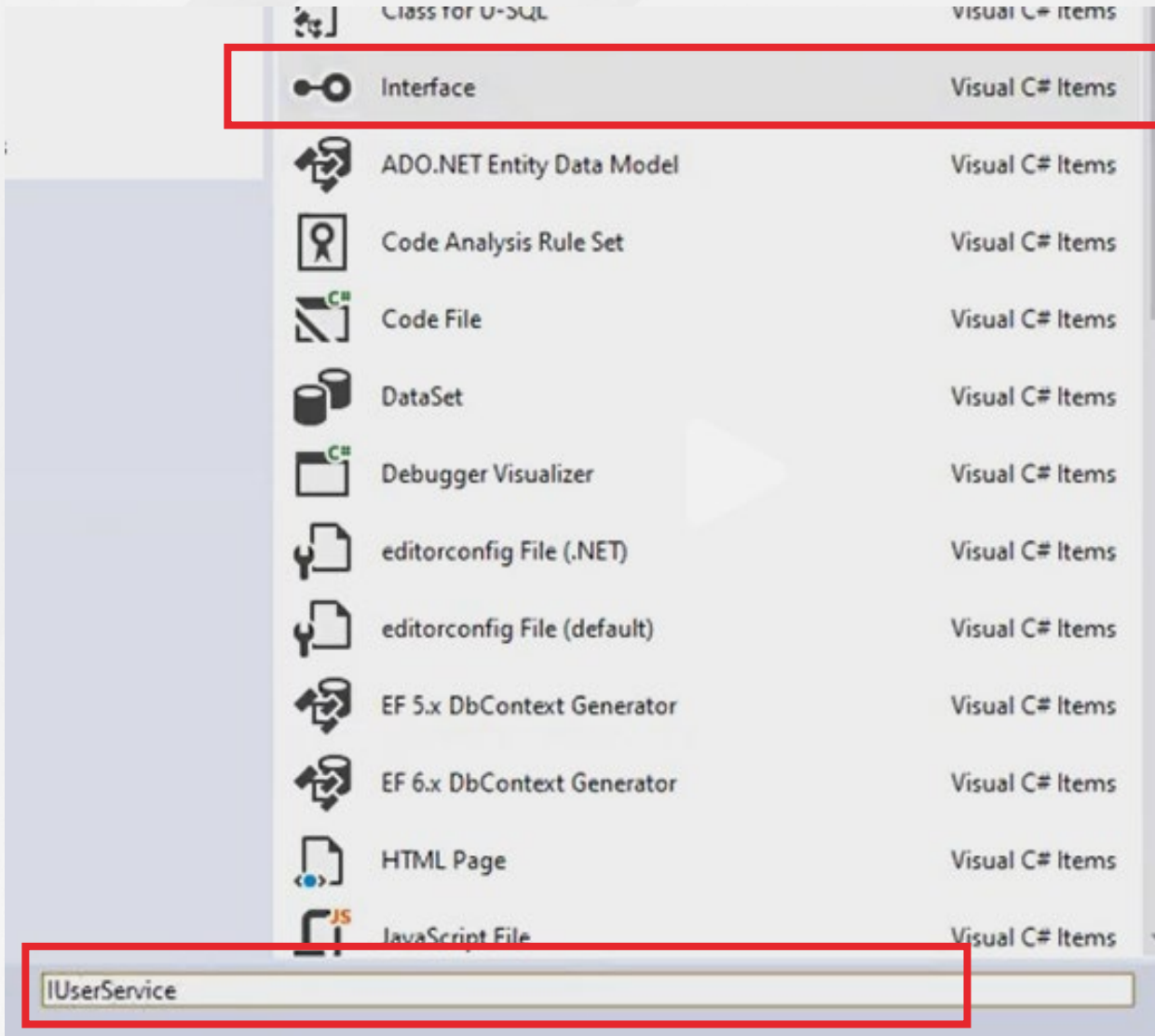


1.1. Business Katmanının Oluşturulması



Business katmanında Abstract ve Concrete isimli iki yeni klasör oluşturulur.

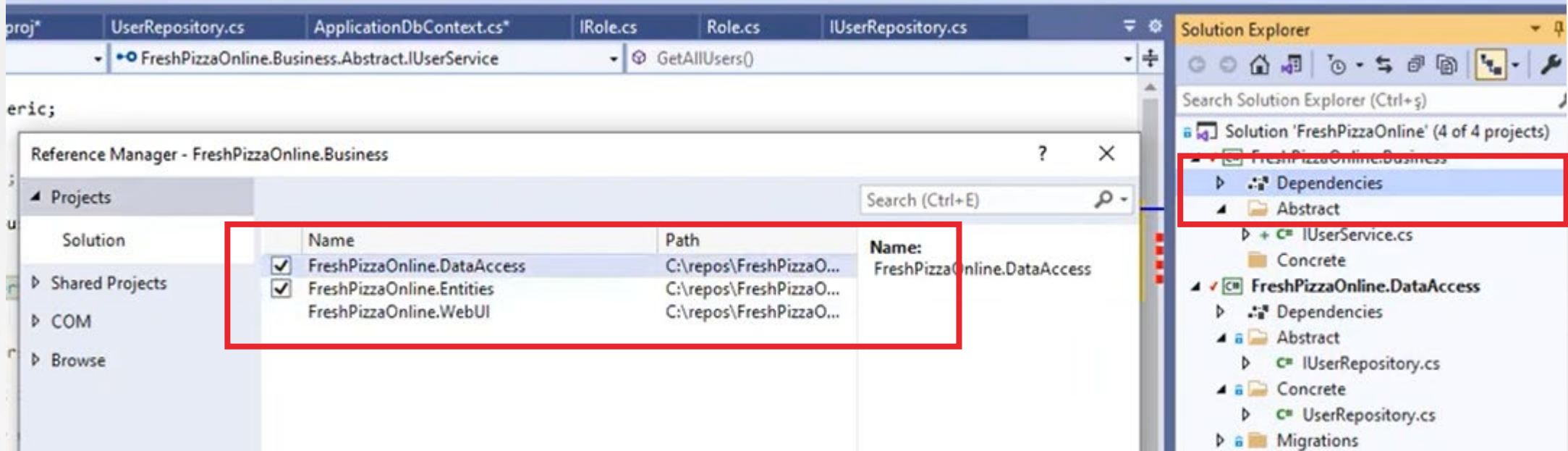
1.1. Business Katmanının Oluşturulması



Abstract klasörü içinde Interface tipinde yeni bir sınıf oluşturulur ve IUserService olarak isimlendirilir.

1.1. Business Katmanının Oluşturulması

Dependencies klasörüne sağ tıklanarak gerekli seçimler işaretlenir.

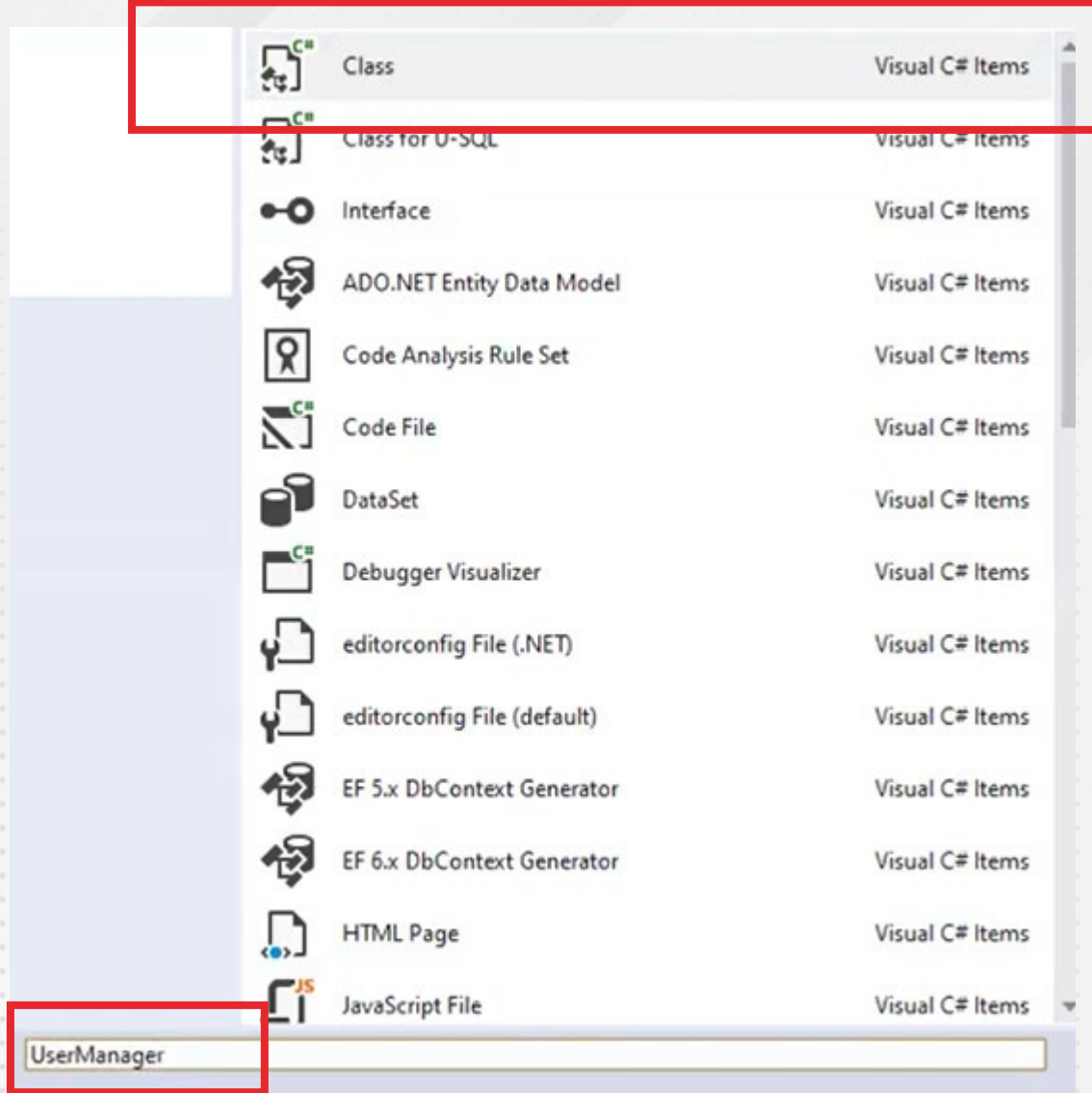


1.1. Business Katmanının Oluşturulması

```
IUserService.cs* x FreshPizzaOnline.Business.csproj* UserRepository.cs
FreshPizzaOnline.Business
1 using FreshPizzaOnline.Entities.Concrete;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace FreshPizzaOnline.Business.Abstract
9 {
10     public interface IUserService
11     {
12         List<User> GetAllUsers();
13         User GetUserById(int id);
14         User CreateUser(User user);
15         User UpdateUser(User user);
16         void DeleteUser(int id);
17     }
18 }
19
```

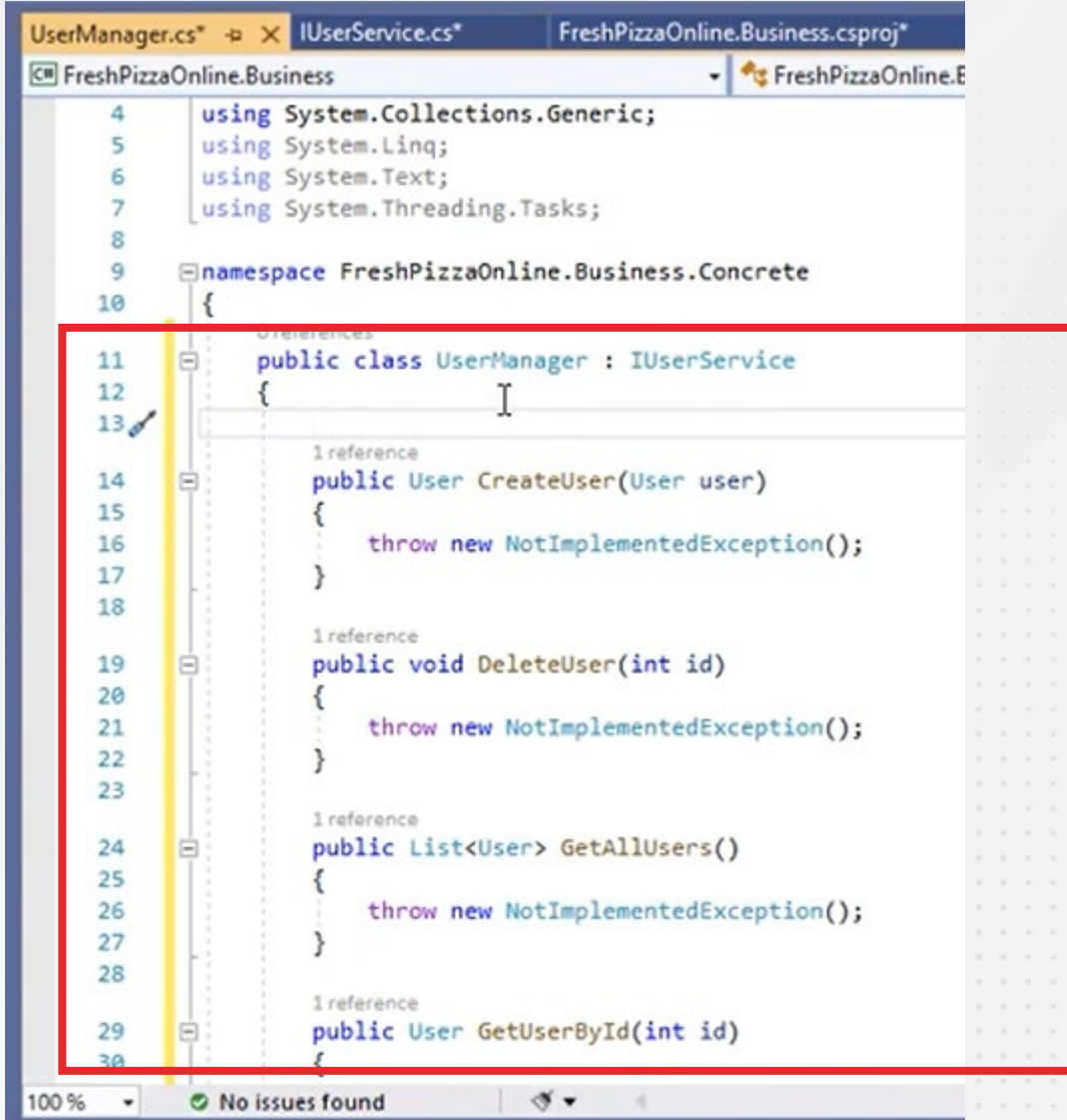
IUserService içinde gerekli metodlar görseldeki gibi tanımlanır

1.1. Business Katmanının Oluşturulması



Concrete klasörü içine bir yeni sınıf eklenir ve UserManager olarak isimlendirilir.

1.1. Business Katmanının Oluşturulması



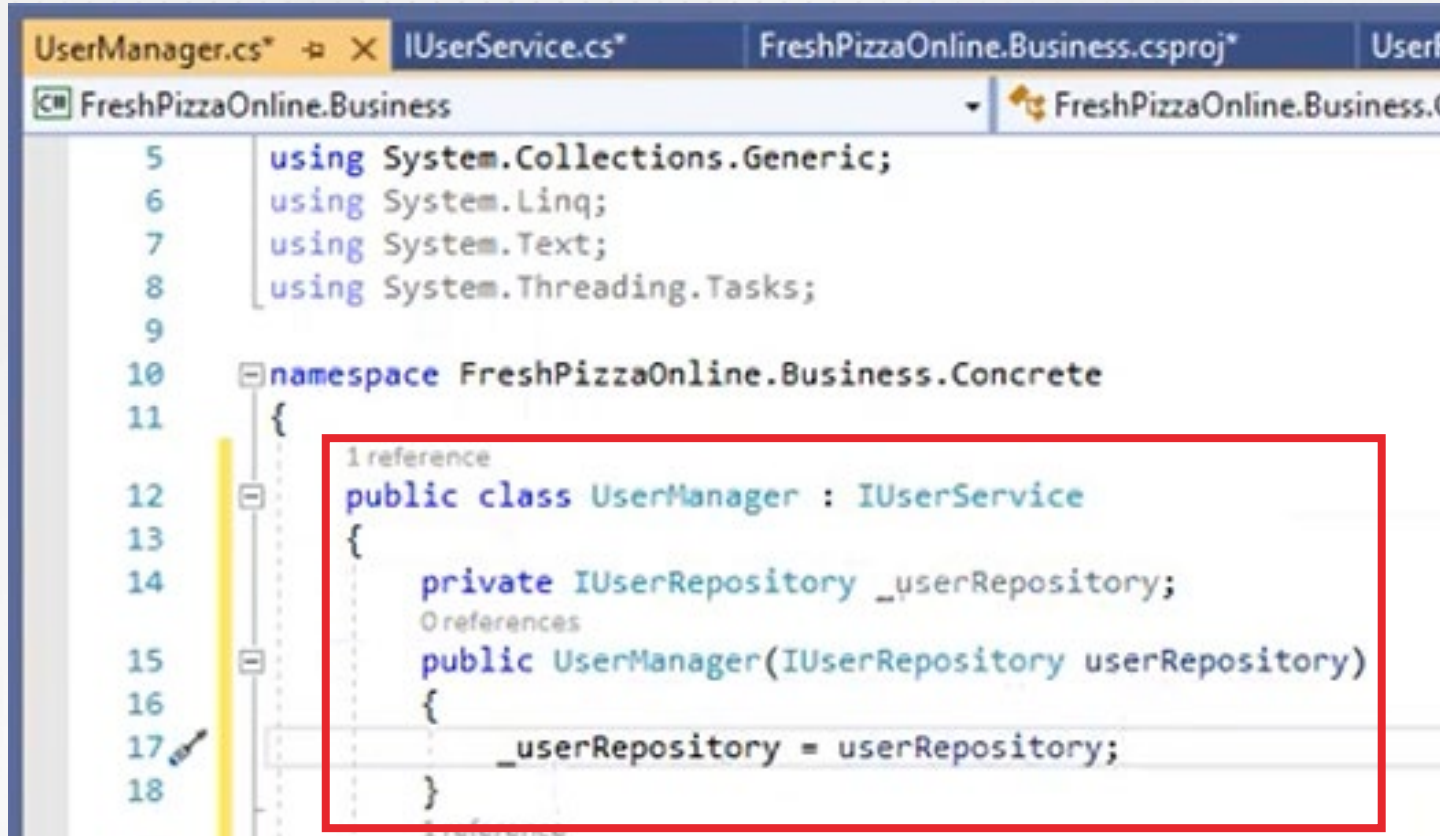
```

UserManager.cs* x IUserService.cs* FreshPizzaOnline.Business.csproj*
FreshPizzaOnline.Business FreshPizzaOnline.E
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace FreshPizzaOnline.Business.Concrete
10 {
11     public class UserManager : IUserService
12     {
13
14         1 reference
15         public User CreateUser(User user)
16         {
17             throw new NotImplementedException();
18         }
19
20         1 reference
21         public void DeleteUser(int id)
22         {
23             throw new NotImplementedException();
24         }
25
26         1 reference
27         public List<User> GetAllUsers()
28         {
29             throw new NotImplementedException();
30         }
31
32         1 reference
33         public User GetUserById(int id)
34         {
35             throw new NotImplementedException();
36         }
37     }
38 }
100 % No issues found
```

Metodlara ilişkin kod blokları daha önce Data Access katmanında olduğu gibi ilgili alana eklenir.

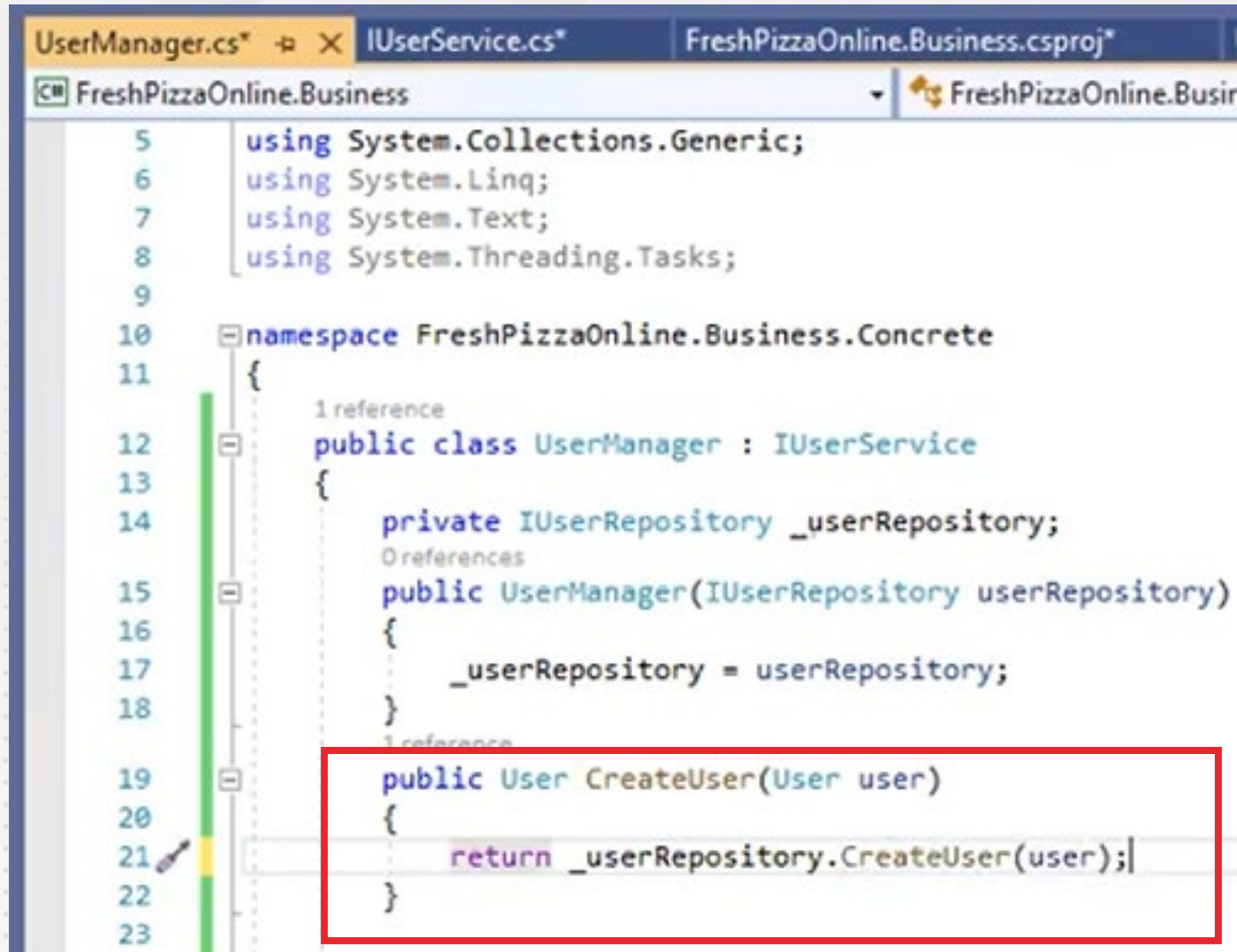
1.1. Business Katmanının Oluşturulması

Data Access katmanındaki IUserRepository de bulunan kodlara erişmek için gereken kod bloğu görselde olduğu gibi ilgili alana eklenir.



```
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace FreshPizzaOnline.Business.Concrete
11 {
12     1 reference
13     public class UserManager : IUserService
14     {
15         private IUserRepository _userRepository;
16         0 references
17         public UserManager(IUserRepository userRepository)
18         {
19             _userRepository = userRepository;
20         }
21     }
22 }
```

1.1. Business Katmanının Oluşturulması



```
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace FreshPizzaOnline.Business.Concrete
11 {
12     1 reference
13     public class UserManager : IUserService
14     {
15         private IUserRepository _userRepository;
16         References
17         public UserManager(IUserRepository userRepository)
18         {
19             _userRepository = userRepository;
20         }
21         1 reference
22         public User CreateUser(User user)
23         {
24             return _userRepository.CreateUser(user);
25         }
26     }
27 }
```

CreateUser Metodunun kullanılabilmesi için gerekli kod satırı eklenir.

1.1. Business Katmanının Oluşturulması

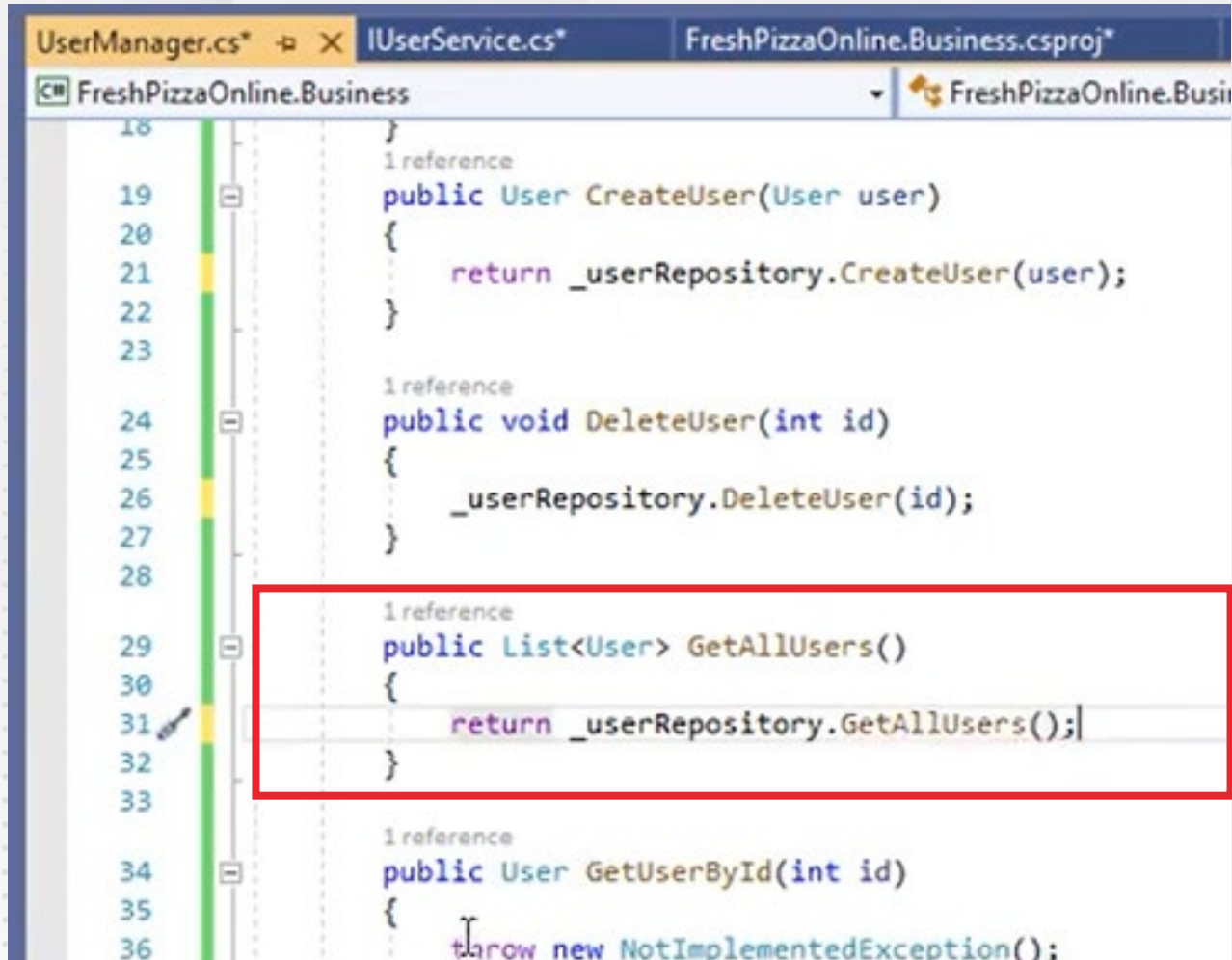
DeleteUser Metodunun kullanılabilmesi için gerekli kod satırı eklenir.



The screenshot shows a Visual Studio IDE with the 'FreshPizzaOnline.Business' project selected. The 'IUserService.cs' file is open, showing the implementation of the 'DeleteUser' method. The method is highlighted with a red rectangle. The code is as follows:

```
16 }  
17  
18 1 reference  
19 public User CreateUser(User user)  
20 {  
21     return _userRepository.CreateUser(user);  
22 }  
23  
24 1 reference  
25 public void DeleteUser(int id)  
26 {  
27     _userRepository.DeleteUser(id);  
28 }
```

1.1. Business Katmanının Oluşturulması



```
18 }
19 1 reference
20 public User CreateUser(User user)
21 {
22     return _userRepository.CreateUser(user);
23 }
24 1 reference
25 public void DeleteUser(int id)
26 {
27     _userRepository.DeleteUser(id);
28 }
29 1 reference
30 public List<User> GetAllUsers()
31 {
32     return _userRepository.GetAllUsers();
33 }
34 1 reference
35 public User GetUserById(int id)
36 {
37     throw new NotImplementedException();
38 }
```

GetAllUsers Metodunun kullanılabilmesi için gerekli kod satırı eklenir.

1.1. Business Katmanının Oluşturulması

```
UserManager.cs* x IUserService.cs* FreshPizzaOnline.Business.csproj* UserRepository.cs
FreshPizzaOnline.Business
FreshPizzaOnline.Business.Concrete.UserM

1 reference
24 public void DeleteUser(int id)
25 {
26     _userRepository.DeleteUser(id);
27 }
28
1 reference
29 public List<User> GetAllUsers()
30 {
31     return _userRepository.GetAllUsers();
32 }
33
1 reference
34 public User GetUserById(int id)
35 {
36     if (id > 0)
37     {
38         return _userRepository.GetUserById(id);
39     }
40     throw new Exception("id değeri 1'den küçük olamaz.");
41 }
42
1 reference
43 public User UpdateUser(User user)
44 {
45     throw new NotImplementedException();
46 }
47 }
```

GetUserById Metodunun kullanılabilmesi için gerekli kod satırı eklenir. Ayrıca yapılmak istenen kontroller Business katmanında if kod blokları ile gerçekleştirilebilir.

1.1. Business Katmanının Oluşturulması

UpdateUser Metodunun kullanılabilmesi için gerekli kod satırı eklenir.

```
1 reference  
public User UpdateUser(User user)  
{  
    return _userRepository.UpdateUser(user);  
}
```

1.1. Business Katmanının Oluşturulması

UpdateUser Metodunun kullanılabilmesi için gerekli kod satırı eklenir.

```
1 reference  
public User UpdateUser(User user)  
{  
    return _userRepository.UpdateUser(user);  
}
```

Bölüm Özeti

- Bu bölüm Business katmanı oluşturulma adımlarını içerir.

Değerlendirme Soruları

BÖLÜM SORULARI

- Business katmanı hangi amaçla kullanılır açıklayınız?