

**Bu kitaba sığmayan
daha neler var!**



Karekodu okutun, bu kitapla
ilgili EBA içeriklerine ulaşın!

ÖDS

**ÖĞRENCİ/ÖĞRETMEN
DESTEK SİSTEMİ**

<https://ods.eba.gov.tr>

- Konu Anlatımlı
Ders Videoları
- Soru Çözüm
Videoları
- Ders Anlatım
Videoları
- Çoktan Seçmeli
Sorular



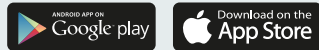
Kişiselleştirilmiş
Öğrenme ve
Raporlama

Animasyonlar,
3B Modeller,
Simülasyon ve Oyunlar

Paylaşım ve
İş birliği

Ortak / Özel
Takvim

eBa
www.eba.gov.tr



**BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA
ÜCRETSİZ OLARAK VERİLMİŞTİR.
PARA İLE SATILAMAZ.**

ISBN: 978-975-11-7115-3

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmelik'in 5'inci Maddesinin
İkinci Fıkrası Çerçevesinde Bandrol Taşınması Zorunlu Değildir.

BİLİŞİM TEKNOLOJİLERİ ALANI

YAZILIM PROJESİ

11-12

DERS MATERYALI

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

YAZILIM PROJESİ

11-12 DERS
MATERYALI



**MESLEKİ VE TEKNİK
ANADOLU LİSESİ**

BİLİŞİM TEKNOLOJİLERİ ALANI

YAZILIM PROJESİ

11-12

DERS MATERYALİ

YAZARLAR

Atılım ÇİFTÇİ

Hamza BALCI

Mahmut ATİK



MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI: 8723
YARDIMCI VE KAYNAK KİTAPLAR DİZİSİ: 2608

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir. Ders materyalinin metin, soru ve şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

HAZIRLAYANLAR

Dil Uzmanı

Melek DEMİR

Program Geliştirme Uzmanı

Fulya ÖLKEN

Ölçme ve Değerlendirme Uzmanı

Fatma YILMAZ

Rehberlik Uzmanı

Gülşen YALIN

Görsel Tasarım Uzmanı

Tuğba SANCI

ISBN: 978-975-11-7115-3

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ile Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.



İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?
Sana olmaz dökülen kanlarımız sonra helâl.
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
Medeniyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın;
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı:
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?
Şüheda fışkıracak toprağı sıksan, şüheda!
Cânı, cânânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlahî, şudur ancak emeli:
Değmesin mabedimin göğsüne nâmahrem eli.
Bu ezanlar -ki şehadetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,
Her cerâhamdan İlahî, boşanıp kanlı yaşım,
Fışkırır ruh-ı mücerret gibi yerden na'sım;
O zaman yükselerek arşa değer belki başım.

Dalgalar sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyyen sana yok, ırkıma yok izmihlâl;
Hakkıdır hür yaşamış bayrağımın hürriyyet;
Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

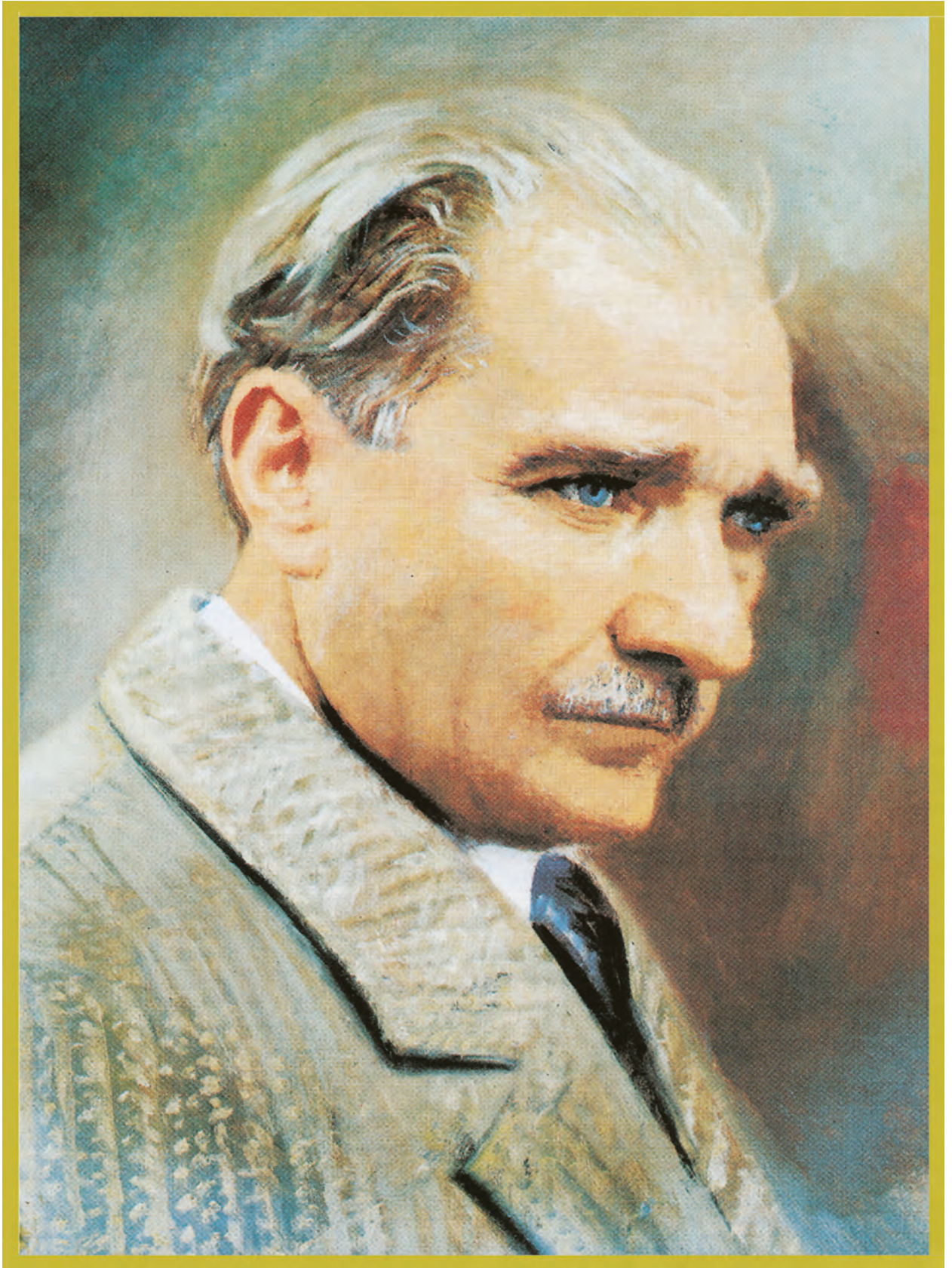
GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsaî bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK

İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI	12
----------------------------------	----

1. ÖĞRENME BİRİMİ : PROJE HAZIRLIK

1. PROJE HAZIRLIK	14
1.1. PROJE HAZIRLIK AŞAMALARI	16
1.2. PROJE UYGULAMA SÜRECİ	16
1.2.1. Planlama	17
1.2.2. Tasarım	21
1.2.3. Kodlama	27
1.2.4. Test	42
1.2.5. Bakım	45
1.3. PROJE SONUÇ RAPORU	46
ÖLÇME VE DEĞERLENDİRME	51

2. ÖĞRENME BİRİMİ : NESNE TABANLI PROGRAMLAMA PROJESİ HAZIRLAMA

2. NESNE TABANLI PROGRAMLAMA PROJESİ HAZIRLAMA	52
2.1. STOK TAKİP PROGRAMININ PLANLAMA AŞAMASI	54
2.1.1. Müşterinin Temel İhtiyaçları	54
2.1.2. Projenin Sınırlarının Belirlenmesi	54
2.1.3. Proje Uygulama Takviminin Oluşturulması	54
2.1.4. Risk Yönetimi ve Analizinin Yapılması	54
2.2. STOK TAKİP PROGRAMININ TASARIM AŞAMASI	55
2.2.1. Gereksinim Analizi	55
2.2.2. Varlık İlişki Şemalarının Oluşturulması	56
2.2.3. Veri Akış Diyagramlarının Hazırlanması	56
2.2.4. Kullanıcı Arayüzleri Tasarımı	57
2.2.5. Veri Tabanı Tasarımı	58
2.2.6. Kullanılacak Teknik ve Yöntemlerin Belirlenmesi	59
2.3. STOK TAKİP PROGRAMI KODLAMA AŞAMASI	59
2.3.1. Veri Tabanının Oluşturulması	59
2.3.2. Form Uygulamasının Oluşturulması ve Veri Tabanı Bağlantısının Kurulması	60
2.3.3. Kategori İşlemleri	63
2.3.4. Tedarikçi İşlemleri	66
2.3.5. Ürün İşlemleri	70
2.3.6. Müşteri İşlemleri	74
2.3.7. Personel İşlemleri	78
2.3.8. Giriş Sayfası	83
2.3.9. Proje Anasayfası	85
2.3.10. Satış Detayları	91
2.4. STOK TAKİP PROGRAMI TEST AŞAMASI	94
2.5. STOK TAKİP PROGRAMI BAKIM AŞAMASI	100
2.6. STOK TAKİP PROGRAMI RAPOR AŞAMASI	103
ÖLÇME VE DEĞERLENDİRME	107

3. ÖĞRENME BİRİMİ : MOBİL UYGULAMA PROJESİ HAZIRLAMA

3. MOBİL UYGULAMA PROJESİ HAZIRLAMA	108
3.1. MOBİL UYGULAMANIN PLANLAMASI	110
3.2. MOBİL UYGULAMANIN TASARIMI	115
3.2.1. Mantıksal Tasarım	115
3.2.1.1. Sql Olarak İlişkisel Tablo Diyagramı	115
3.2.1.2. Firebase Olarak Hazırlanan Yapı	116
3.3. MOBİL UYGULAMANIN KODLAMASI	124
3.3.1. activity_gezi_rehberi.xml	125
3.3.2. activity_kayit.xml	126
3.3.3. activity_logo.xml	126
3.3.4. activity_main.xml	127
3.3.5. fragment_foto_kayit.xml	128
3.3.6. fragmnet_tr_bilgi.xml	128
3.3.7. fragment_eng_bilgi.xml	129
3.3.8. gezi_menu.xml	129
3.3.9. dilsecimi_menu.xml	129
3.3.10. build.gradle (Project:GeziTakimi)	130
3.3.11. build.gradle (Modüle:GeziTakimi.app)	130
3.3.12. AndroidManifest.xml	131
3.3.13. LogoActivity.java	132
3.3.14. GeziRehberBilgileri.java	132
3.3.15. FotoKayitFragment.java	133
3.3.16. TrBilgiFragment.java	135
3.3.17. EngBilgiFragment.java	136
3.3.18. KayitActivity.java	137
3.3.19. MainActivity.java	143
3.3.20. GeziRehberiActivity.java	146
3.4. MOBİL UYGULAMANIN TEST VE BAKIMI	148
3.5. MOBİL UYGULAMANIN RAPORU	150
ÖLÇME VE DEĞERLENDİRME	151

4. ÖĞRENME BİRİMİ : WEB TABANLI UYGULAMA PROJESİ HAZIRLAMA

4. WEB TABANLI UYGULAMA PROJESİ HAZIRLAMA	152
4.1. WEB TABANLI UYGULAMA PROJESİNİN PLANLANMASI	154
4.1.1. Yazılım Geliştirme Planı	154
4.1.2. Yazılım Geliştirme Planına Duyulan İhtiyaç	154
4.1.3. Proje İş Akışının Tanımlanması	155
4.1.4. Proje Planının Kapsamına Karar Verilmesi	155
4.1.5. Proje Sürecinin ve Çıktılarının Netleştirilmesi	155
4.1.6. Projenin İş Yükünün Tahmin Edilmesi	155
4.1.7. Projenin İş Yükü Tahmininin Sorgulanması	155
4.1.8. Projenin Kilometre Taşlarının Oluşturulması	155
4.1.9. Projenin Kaynaklarının Tahmin Edilmesi	156
4.1.10. Projede Değişikliklere Yer Bırakılması	156
4.1.11. Projenin Hizmete Geçişinin Planlanması	156
4.1.12. Proje Sonrası İnceleme Yapılması	156

4. ÖĞRENME BİRİMİ : WEB TABANLI UYGULAMA PROJESİ HAZIRLAMA

4.1.13. Risk Yönetimi	156
4.1.13.1. Riskin Tanımlanması	157
4.1.13.2. Risk Analizinin Yapılması	157
4.1.13.3. Riskin Planlanması	157
4.1.13.4. Riskin İzlenmesi	157
4.2. WEB TABANLI UYGULAMA PROJESİNİN TASARLANMASI	158
4.2.1. Gereksinim Analiz Aşamaları	158
4.2.2. Sistemin Modellenmesi	159
4.3. WEB TABANLI UYGULAMA PROJESİNİN KODLANMASI	159
4.4. WEB TABANLI UYGULAMA PROJESİNİN TEST VE BAKIMININ YAPILMASI ..	183
4.5. WEB TABANLI UYGULAMA PROJESİNİN RAPORUNUN OLUŞTURULMASI ..	184
ÖLÇME VE DEĞERLENDİRME	187

KAYNAKÇA	188
GÖRSEL KAYNAKÇASI	188
CEVAP ANAHTARI	189

DERS MATERYALİNİN TANITIMI

Öğrenme biriminin numarasını gösterir.

Öğrenme biriminin görselini gösterir.

Öğrenme biriminin konularını gösterir.

Öğrenme biriminde neler öğrenileceğini gösterir.

1. ÖĞRENME BİRİMİ

PROJE HAZIRLIK



KONULAR

- 1.1. PROJE HAZIRLIK AŞAMALARI
- 1.2. PROJE UYGULAMA SÜRECİ
- 1.3. PROJE SONUÇ RAPORU

NELER ÖĞRENECEKSİNİZ?

- Proje hazırlık aşamaları
- Proje uygulama süreci adımları
- Proje sonuç raporunu hazırlama aşamaları

TEMEL KAVRAMLAR

Bakım, gereksinim analizi, kodlama, planlama, proje sonuç raporu, proje uygulama süreci, risk yönetimi, tasarım, test, yazılım projesi

HAZIRLIK ÇALIŞMALARI

1. Proje kelimesinin size çağrıştırdıklarını arkadaşlarınızla paylaşınız.
2. Yazılım projelerinin günlük projelerden farkları neler olabilir? Arkadaşlarınızla tartışınız.

Öğrenme biriminin adını gösterir.

Öğrenme biriminin karekodunu gösterir.

Öğrenme biriminin temel kavramlarını gösterir.

Öğrencilerin öğrenme birimiyle ilgili ön bilgilerini harekete geçirmek ve konuya ilgilerini çekmek için yapılacak çalışmaları gösterir.

DERS MATERYALİNİN TANITIMI

Konuyla ilgili görselleri gösterir.

Öğrenme biriminin adını ve numarasını gösterir.

Konu başlıklarını gösterir.

Konuyla ilgili notları gösterir.

Araştırma etkinliğini gösterir.

Sıra sizde etkinliğini gösterir.



1. ÖĞRENME BİRİMİ: PROJE HAZIRLIK

1.1. PROJE HAZIRLIK AŞAMALARI

Proje, Türk Dil Kurumu Güncel Türkçe Sözlüğü'ne göre "Değişik alanlarda önceden plan ve programa alınmış, maliyeti hesaplanmış, kurum ve kuruluşların yönetim organları tarafından onaylanmış, kısa ve uzun vadeye bağlanarak özel kurum veya devlet adına gerçekleştirilmesi kabul edilmiş bilimsel çalışma tasarısı." veya "Gerçekleştirilmesi istenen tasarım" olarak tanımlanır. Bu tanımlara göre proje için bir tasarım gereklidir. Herhangi bir problemi yazılım kullanarak çözmek için bir tasarım hazırlamak önemlidir.

1.2.1. Projenin Kaynaklarını Tahmin Etmek

Yazılım projesi geliştirme sürecinin ilk aşaması olan planlama aşamasında talep alma (müşteri ihtiyaçları), uygulanabilirlik, maliyet hesabı, kaynak ve zaman planlaması, proje takvimi oluşturulması gibi işlemler yapılır (Görsel 1.2). Planlamanın yapılabilmesi için proje ile ilgili temel ihtiyaçların önceden belirlenmesi gerekir. Projenin neleri içereceği (kapsam), hangi probleme çözüm üreteceği veya nasıl bir kolaylık sağlayacağı genel hatlarıyla belirlenmeli ve buna göre planlama yapılmalıdır.



Görsel 1.2: Planlama



ÖRNEK

Telefon Rehberi projesi için test aşaması şu şekilde gerçekleştirilir:

- Projenin grup sayfasında ekleme, silme, güncelleme ve listeleme fonksiyonlarının doğru çalışıp çalışmadığı test edilir (Görsel 1.36).
- Projenin telefon rehberi sayfasında ekleme, silme, güncelleme, listeleme ve arama fonksiyonlarının çalışıp çalışmadığı test edilir (Görsel 1.37, Görsel 1.38).
- Bilgi girişi kontrollerinin yapıp yapılmadığı test edilir.
- Projede oluşan dışı durum çözümleme işlemlerinin yapıp yapılmadığı test edilir.



1. UYGULAMA

İşlem adımlarına göre web tabanlı e-tercih yazılım projesi veri tabanını oluşturunuz.

1. Adım: Veri tabanı oluşturma uygulamasını açınız.

Veri tabanını oluşturmak için Microsoft SQL Server Management Studio uygulaması kullanılacaktır. Uygulama bilgisayarınızda kurulu ise uygulamayı çalıştırınız. Uygulama bilgisayarınızda kurulu değilse internetten indirilerek öncelikle uygulamayı kurunuz. Uygulamayı çalıştırdığınızda Görsel 4.1'deki ekranla karşılaşacaksınız.

2

YAZILIM PROJESİ

Uygulama etkinliğini gösterir.

Kitap adını gösterir.

Konuyla ilgili örnekleri gösterir.



1. ÖĞRENME BİRİMİ: PROJE HAZIRLIK

ARAŞTIRMA

Yazılım yaşam döngüsü kavramını araştırınız.



NOT

Özellikle birden fazla kişi tarafından geliştirilen orta ve büyük ölçekli projelerde versiyon kontrol sistemi kullanılması avantaj sağlayacaktır.



SIRA SİZDE

Personel Takip Programı projesinin risk yönetimini tablo halinde oluşturunuz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje risklerini belirledi.		
2. Risklerin gerçekleşme olasılığını hesapladı.		
3. Risklerin projeye etkisini hesapladı.		
4. Risklere karşı alınacak önlemleri belirledi.		

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Bir yazılım projesi hazırlanırken ilk olarak aşağıdaki işlemlerden hangisi yapılmalıdır?

- A) Fizibilite çalışması
- B) Risk analizi
- C) Varlık ilişki şemaları
- D) Veri akış diyagramı
- E) Veri tabanı tasarımı

2. Aşağıdakilerden hangisi yazılım projesi hazırlama süreci adımlarından biri değildir?

- A) Kodlama
- B) Planlama
- C) Sunum
- D) Tasarım
- E) Test

3

YAZILIM PROJESİ

Ölçme ve değerlendirme sorularını gösterir.

Sayfa numarasını gösterir.



1. ÖĞRENME BİRİMİ

PROJE HAZIRLIK



KONULAR

- 1.1. PROJE HAZIRLIK AŞAMALARI
- 1.2. PROJE UYGULAMA SÜRECİ
- 1.3. PROJE SONUÇ RAPORU

NELER ÖĞRENECEKSİNİZ?

- Proje hazırlık aşamaları
- Proje uygulama süreci adımları
- Proje sonuç raporunu hazırlama aşamaları

TEMEL KAVRAMLAR

Bakım, gereksinim analizi, kodlama, planlama, proje sonuç raporu, proje uygulama süreci, risk yönetimi, tasarım, test, yazılım projesi

HAZIRLIK ÇALIŞMALARI

1. Proje kelimesinin size çağrıştırdıklarını arkadaşlarınızla paylaşınız.
2. Yazılım projelerinin günlük projelerden farkları neler olabilir? Arkadaşlarınızla tartışınız.



1.1. PROJE HAZIRLIK AŞAMALARI

Proje, Türk Dil Kurumu Güncel Türkçe Sözlük'e göre “Değişik alanlarda önceden plan ve programa alınmış, maliyeti hesaplanmış, kurum ve kuruluşların yönetim organları tarafından onaylanmış, kısa ve uzun vadeye bağlanarak özel kurum veya devlet adına gerçekleştirilmesi kabul edilmiş bilimsel çalışma tasarısı.” veya “Gerçekleştirilmesi istenen tasarı.” olarak tanımlanır. Bu tanımlara göre proje için bir tasarı gereklidir. Herhangi bir problemi yazılım kullanarak çözmek için bir tasarı hazırlamak önemlidir.

Bir yazılım projesi, diğer projelerden farklı olarak daha karmaşık, görünmez ve esnek bir yapıya sahiptir. Bu nedenle yazılım projesi geliştirmek, diğer alanlardaki projelere göre daha zordur. Diğer alanlardaki projelerde bir projenin başlangıç ve bitiş süresi, yapılacak işlemler, ortaya çıkarılacak ürün ve proje maliyeti daha başlangıç aşamasında tahmin edilebilir. Değişiklik yapılacak olsa bile bunu projeye uygulamak daha kolaydır. Yazılım projesinde ise yapılacak en küçük değişiklik, projenin bir veya birden fazla bölümünün olumsuz etkilenmesine ve tekrar kodlanmasına, dolayısıyla maliyet artışına ve zaman kaybına neden olabilir.

Proje hazırlık aşamasında **fizibilite** çalışması yapılır. Bir başka deyişle ihtiyaçlar, maliyet, kullanılabilirlik, yöntem veya teknik, yasalara uygunluğu gibi bazı ölçütlere göre projenin uygulanabilirliği araştırılır. Örneğin telif haklarını gözetmeyen bir projeyi hayata geçirmek doğru değildir veya teknik açıdan hayata geçirilmesi mümkün olmayan bir projeye başlamak mantıklı değildir. Bunlara benzer olumsuz durumlar, proje hazırlık aşamasında yapılacak fizibilite çalışmasında ortaya çıkarılarak önlenir.

Örneğin yapılması planlanan bir telefon rehberi projesi; hazırlık aşamasında ihtiyaçlar, maliyet, kullanılabilirlik, yöntem, teknik ve yasalara uygunluk açısından incelendiğinde projenin uygulanmasında bir problem olmadığı görülür. Bu durumda projeye başlanabilir.



SIRA SİZDE

Yapılacak “Personel Takip Programı” projesinin hazırlık aşamasını gerçekleştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

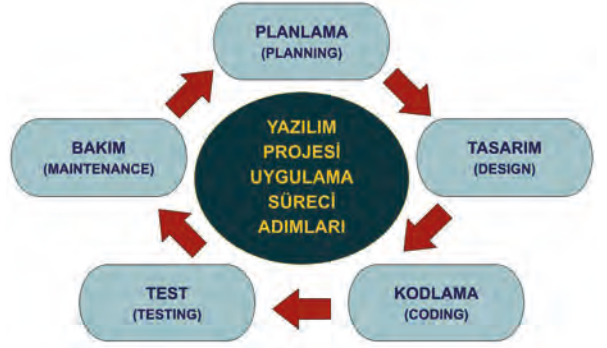
Değerlendirme Ölçütleri	Evet	Hayır
1. Projenin kullanıcı ihtiyaçları ve maliyet açısından yapılabilirliğini kontrol etti.		
2. Projenin kullanılabilirliğini kontrol etti.		
3. Projenin yöntem bakımı ve teknik açıdan yapılabilirliğini inceledi.		
4. Projenin yasalara uygunluğunu kontrol etti.		

1.2. PROJE UYGULAMA SÜRECİ

Yazılım projesi uygulama süreci; bir yazılım projesi hazırlanırken kullanıcı ihtiyaçlarının belirlenmesinden projenin hazır hâle getirilmesine, hatta projenin test ve bakımının yapılmasına kadar bütün işlemleri kapsayan bir süreç olarak tanımlanabilir (Görsel 1.1). Proje sürecinin başarılı bir şekilde tamamlanabilmesi için bu süreçteki her işlemin eksiksiz bir şekilde yapılması gerekir. Doğru bir şekilde tamamlanmayan her aşama, o adımı tekrar dönülmesine yol açar ve projenin tamamlanma süresini uzatır. Ayrıca herhangi bir aşamada yapılan hata, sürecin tamamını etkileyerek ortaya eksik veya hatalı bir yazılım ürünü çıkmasına sebep olabilir.



Bir yazılım projesinde kullanıcı ihtiyaçları kısa sürede değişiklik gösterebilir, projenin geliştirme sürecinin herhangi bir aşamasında hata oluşabilir veya yazılım projesi yayımlandıktan sonra projeye bakım ihtiyacı olabilir. Bu durumlarda ilgili adıma dönüp o aşamadan itibaren bütün aşamaların tekrar gerçekleştirilmesi gerekir. Örneğin bir yazılım projesinde test aşamasına gelindiğinde müşteri istekleri değişebilir. Böyle bir durumda planlama aşamasından itibaren sürecin tekrar uygulanması gerekir.



Görsel 1.1: Yazılım projesi uygulama süreci adımları



ARAŞTIRMA

Yazılım yaşam döngüsü kavramını araştırarak bulduğunuz sonuçları arkadaşlarınızla paylaşınız.



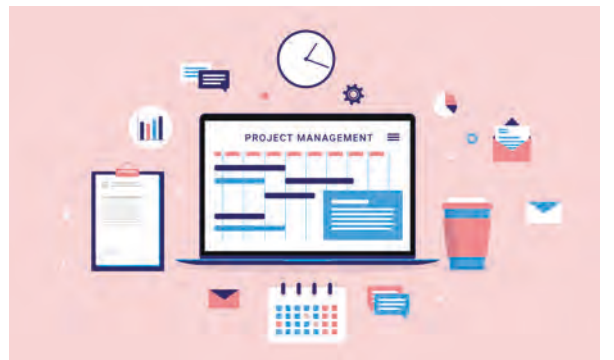
NOT

Yazılım projeleri ile ilgili en çok yapılan hatalardan biri, projenin yalnızca kodlama aşamasından oluştuğunun düşünülmesidir. Geliştiriciler, basit projeleri doğrudan kodlama yaparak geliştirebilir fakat orta ve büyük ölçekli projelerde, proje uygulama sürecinin adımları uygulanmalıdır. Bu durum, projenin bir plana bağlı geliştirilmesini sağlayarak ortaya çıkabilecek karmaşıklıkları önler ve proje yönetimini kolaylaştırır.

Yazılım projesi uygulama sürecinin temel amacı, müşteri ihtiyaçlarını karşılayan kaliteli bir yazılım ürününü mümkün olan en kısa sürede ortaya çıkarmaktır. Bu da proje geliştirme süreci adımlarının doğru bir şekilde uygulanmasıyla mümkündür. Yazılım projesi uygulama süreci, projenin büyüklüğüne ve yazılım geliştirme modeline göre farklılıklar gösterse de genel olarak planlama, tasarım, kodlama, test ve bakım adımlarından oluşur.

1.2.1. Planlama

Yazılım projesi geliştirme sürecinin ilk aşaması olan planlama aşamasında talep alma (müşteri ihtiyaçları), uygulanabilirlik, maliyet hesabı, kaynak ve zaman planlaması, proje takvimi oluşturulması gibi işlemler yapılır (Görsel 1.2). Planlamanın yapılabilmesi için proje ile ilgili temel ihtiyaçların önceden belirlenmesi gerekir. Projenin neleri içereceği (kapsamı), hangi probleme çözüm üreteceği veya nasıl bir kolaylık sağlayacağı genel hatlarıyla belirlenmeli ve buna göre planlama yapılmalıdır.



Görsel 1.2: Planlama



Planlama aşamasında yapılabilecek işlemler şunlardır:

- Müşterinin temel ihtiyaçlarının belirlenmesi
- Projenin sınırlarının belirlenmesi
- Proje sürecini takip edecek ve koordinasyonu sağlayacak proje lideri ile takım liderinin belirlenmesi
- Proje ekiplerinin oluşturulması ve iş bölümünün yapılması
- Zaman ve kaynak planlamasının yapılması
- Proje maliyet hesabının yapılması
- Risk yönetimi ve analizinin yapılması
- Proje takvimi veya proje planının oluşturulması



ÖRNEK

Telefon Rehberi projesi için planlama aşaması aşağıdaki gibi gerçekleştirilebilir.

- **Müşterinin Temel İhtiyaçları**
 - * Kişilerimin telefon numarası bilgilerini saklayacağım bir telefon rehberine ihtiyacım var.
 - * Yalnızca bir bilgisayarda kullanılacak masaüstü uygulaması olmalıdır.
 - * Bu rehberde kişiler, gruplara (arkadaşlar, aile, iş vb.) göre sınıflandırılmalıdır.
- **Projenin Sınırlarının Belirlenmesi**
 - * Proje; kişiler ve kişi grupları ile ilgili ekleme, silme, güncelleme, listeleme ve arama gibi işlemleri içerecektir.
- **Proje Uygulama Takviminin veya Proje Planının Oluşturulması**

Müşterinin temel ihtiyaçları öğrenilip projenin sınırları belirlendikten sonra Tablo 1.1'deki gibi bir proje uygulama takvimi veya proje planı hazırlanır.

Tablo 1.1: Telefon Rehberi Projesi Uygulama Takvimi

No	Proje Bölümü	Yapacak Personel	Süre	Başlangıç	Bitiş
1	Risk yönetimi ve analizinin yapılması	Öğrenci1, Öğrenci2	2 gün	1. gün	2. gün
2	Gereksinim analizinin yapılması	Öğrenci1, Öğrenci2	3 gün	3. gün	5. gün
3	Sistem modelleme	Öğrenci1	1 gün	6. gün	6. gün
4	Varlık ilişki şemalarının oluşturulması	Öğrenci1	1 gün	7. gün	7. gün
5	Bileşen ve arayüzlerin tasarımı	Öğrenci2	2 gün	6. gün	7. gün
6	Veri akış diyagramlarının oluşturulması	Öğrenci1	1 gün	8. gün	8. gün
7	Tasarımın gözden geçirilmesi	Öğrenci2	1 gün	8. gün	8. gün
8	Veri tabanının oluşturulması	Öğrenci2	1 gün	9. gün	9. gün
9	Projenin kodlanması	Öğrenci1	2 gün	9. gün	10. gün
10	Projenin test edilmesi	Öğrenci2	1 gün	10. gün	10. gün

Proje Süresi: 10 iş günü



SIRA SİZDE

Müşterinin temel ihtiyaçları verilen Personel Takip Programı projesinin planlama aşamasını gerçekleştiriniz (Proje sürecini ikişer veya üçer kişilik gruplar hâlinde tamamlayınız.).

Bu proje için müşterinin temel ihtiyaçları şunlardır: Firmamızda çalışan personellerin takibinin yapılacağı bir yazılım ürününe ihtiyacımız var. Ürün; personelin öz geçmiş, departman, maaş, izin gibi kişisel bilgilerini saklamalı ve bunlarla ilgili gerekli işlemleri (ekleme, silme, güncelleme, listeleme, arama vb.) yapabilmelidir.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Kullanıcı ihtiyaçlarını göz önünde bulundurarak projenin sınırlarını belirledi.		
2. Projede yapılacak işlemlerin süresini dikkate alarak proje uygulama takvimini oluşturdu.		

Yazılım Projesinde Risk Yönetimi

Yazılım projesinin geliştirilmesini ve doğru çalışmasını olumsuz etkileme ihtimali olan olay ve durumlara **risk** denir. **Risk yönetimi**; yazılım projesinde ortaya çıkabilecek riskleri belirlemek, belirlenen riskleri analiz etmek ve bu risklerin olumsuz etkilerini en aza indirmek için yapılan çalışmalar olarak adlandırılabilir. Risk yönetiminin temel amacı, riskleri ortaya çıkmadan önce tespit ederek gerekli önlemleri almaktır (Görsel 1.3).



Görsel 1.3: Risk yönetimi

Yazılım projelerinin hepsinde yanlış maliyet ve zaman tahmini, gereksinimlerin proje sürecinde değişiklik göstermesi, personel özellikleri ve eksikliği, hatalı tasarım ve kodlama gibi muhtemel riskler vardır. Bunların dışında her proje kendine özgü riskler barındırır. Bu riskler göz önünde bulundurularak risk yönetimi yapılmalıdır.



Yazılım projelerinde risk yönetimi genel olarak şu adımlardan oluşur:

- **Risk Tanımlama (Risk Belirleme):** Projenin uygulama sürecini olumsuz etkileyen risklerin belirlendiği aşamadır.
- **Risk Analizi ve Sıralama:** Belirlenen risklerin olma olasılığı ve proje üzerinde oluşturacağı olumsuz etkilerin büyüklüğü değerlendirilir. Risklerin proje üzerindeki etkilerinin hangi boyutlarda olacağı doğru analiz edilmeli, bu risklerin kabul edilebilir olup olmadığına karar verilmelidir. Bu aşamada riskler önceliklendirilir ve sınıflandırılır.
- **Risk Planlaması:** Bu aşamada önceliklerine göre risklerin olumsuz etkilerini ortadan kaldıracak veya en aza indirecek bir eylem planı hazırlanır.
- **Risk İzleme ve Kontrol:** Proje uygulama süreci boyunca riskler ve risklere yönelik alınan tedbirler izlenir. Uygulanan eylem planları sonucunda risklerin durumu kontrol edilir. Projeye olumsuz etkisi istenen düzeye indirilememiş riskler için eylem planı revize edilir veya yeni eylem planı oluşturulur.

Yazılım projelerinde risk yönetimi, proje hazırlama süreci boyunca devam eder. Belirli periyotlarda yeni risklerin olup olmadığı kontrol edilir ve risk yönetimi adımları tekrar uygulanır. Ayrıca risk yönetim sürecinin belgelenmesi de risklerin takibinin yapılması açısından oldukça önemlidir.



ÖRNEK

Telefon Rehberi projesi için risk yönetimi Tablo 1.2'ye göre gerçekleştirilir.

Tablo 1.2: Telefon Rehberi Projesi Risk Yönetimi

Risk Tablosu				
No	Risk	Gerçekleşme Olasılığı	Projeye Etkisi	Alınacak Önlemler
1	Aynı numaranın birden fazla kişi için kaydedilmesi	Orta	Yüksek	Veri tabanında numara alanı Unique Key (Tekil Anahtar) olarak belirlenir.
2	Numara alanına rakam dışında değerler girilebilmesi	Yüksek	Orta	Numaralar kaydedilmeden önce telefon numarası olup olmadığı kontrol edilir.
3	Rehberde aynı ad ve soyada sahip birden fazla kişi olması ve bu kişilerin birbiriyle karıştırılabilmesi	Orta	Orta	Aynı ad ve soyada sahip kişilerin karışmaması için unvan veya takma ad gibi başka bir alan kullanılır.
4	Aynı grubun birden fazla kaydedilmesi	Orta	Yüksek	Veri tabanında grup adı alanı Unique Key olarak belirlenir.



SIRA SİZDE

Personel Takip Programı projesinin gerçekleştirilmesi sırasında ve sonrasında olabilecek problemleri ve riskleri analiz ederek risk yönetim tablosu oluşturunuz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje risklerini belirledi.		
2. Risklerin gerçekleşme olasılığını hesapladı.		
3. Risklerin projeye etkisini hesapladı.		
4. Risklere karşı alınacak önlemleri belirledi.		

1.2.2. Tasarım

Tasarım adımına başlanmadan önce müşteri taleplerinin net olarak anlaşılması için gereksinim analizi yapılmalıdır. Gereksinim analizi, proje için hayati önem taşır. Bu analiz sonucunda ürünün bütün özellikleri ve işlevleri belirlenir. Projenin tasarımı da buna göre yapılır. Gereksinim analizi eksik veya yanlış yapılırsa projenin tasarımı ve kodlaması da eksik veya yanlış olacaktır. Bundan dolayı gereksinim analizi yapılırken, müşteri ve kullanıcılara ürünle ilgili sorular sorularak projenin tüm detaylarının ortaya çıkarılması gerekir (Görsel 1.4).



Görsel 1.4: Gereksinim analizi

Gereksinim analizinin aşamaları şunlardır:

- Müşteri ve kullanıcılardan ihtiyaçların toplanması
- İhtiyaçların uygunluğunun ve uygulanabilirliğinin analiz edilmesi
- Ürünün işlevlerinin belirlenmesi (Kullanıcı senaryoları veya kullanım durumu diyagramları oluşturulabilir.)
- Sistemin donanımsal ve yazılımsal gereksinimlerinin belirlenmesi (Ürünün kullanılabilmesi için gerekli olan donanım ve yazılımlar)



- Performans ve güvenlik gereksinimlerinin belirlenmesi
- Belirlenen ihtiyaç ve işlevlerin belgelenmesi (Gereksinimler açık, net ve anlaşılır olmalıdır.)
- İhtiyaçların ve işlevlerin doğruluğunun kontrol edilmesi



NOT

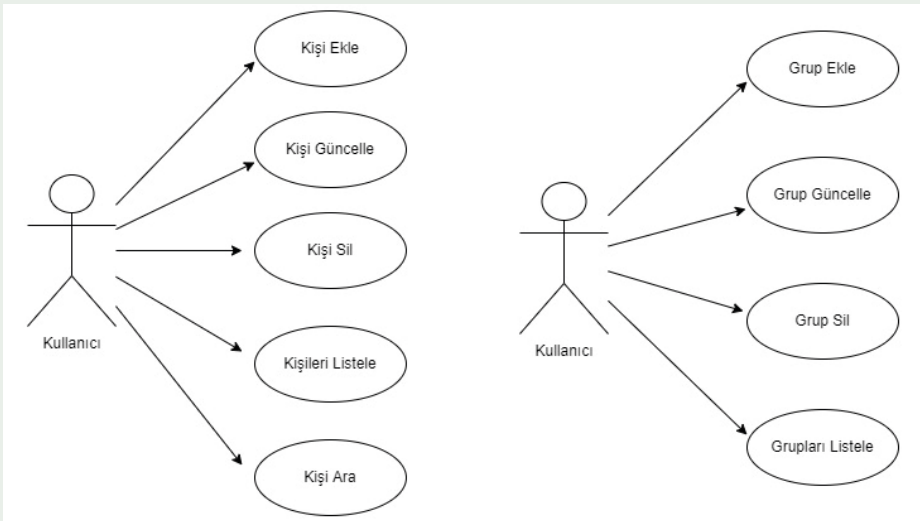
Müşteri gereksinimlerinin belgelenmesi, proje uygulama sürecinin ileriki aşamalarında kolaylık sağlar. Ayrıca proje bitiminde, geliştirilen yazılım ürünü ile ilgili müşteriyle yaşanabilecek olumsuzlukların da önüne geçilir.



ÖRNEK

Telefon Rehberi projesi için gereksinim analizi şu şekilde gerçekleştirilir:

- Müşteri gereksinimleri, bir başka deyişle müşterinin üründen beklentileri tespit edilir.
 - * Ürün, internet bağlantısı gerektirmeden çalışan bir masaüstü uygulaması (form uygulaması) olmalıdır.
 - * Programda yalnızca bir kullanıcı olacaktır (Kullanıcı rolleri ve kullanıcı giriş bölümüne ihtiyaç yoktur.).
 - * Kişinin ad, soyad, telefon, mail, kişi grubu (aile, arkadaşlar, iş vb.) bilgileri sistemde kayıtlı olacaktır. Ad, soyad, telefon ve grup bilgilerinin girilmesi zorunlu olacaktır. Zorunlu alanlardaki veriler girilmediğinde kullanıcıya uyarı mesajı verilmelidir.
 - * Eklenen her kişi için iki adet telefon girişi yapılabilecektir. İkinci numaranın girilmesi zorunlu değildir.
 - * Kişi ve kişi gruplarını ekleme, silme, güncelleme ve listeleme işlemleri yapılabilir. Listeleme işlemlerinde veriler, ad ve soyada göre A'dan Z'ye doğru sıralanmalıdır.
 - * Kişiler, gruplara göre listelenebilmeli ve kişiler içinde ad, soyad ve telefon numarasına göre arama işlemi yapılabilir.
- Kullanım durumu (Use case) diyagramları Görsel 1.5'teki gibi oluşturulur.



Görsel 1.5: Telefon rehberi projesine ait kullanım durumu (use case) diyagramları



- Sistem gereksinimleri şu şekilde belirlenir:
 - * Windows 7 ve üzeri işletim sistemi yüklü olmalıdır.
 - * .NET Framework 4.6 veya üzeri yüklü olmalıdır.
 - * MySQL veri tabanı yönetim sistemi yüklü olmalıdır.
 - * MySQL Connector/.NET eklentisi yüklü olmalıdır.



SIRA SİZDE

Personel Takip Programı projesi için gereksinim analizini gerçekleştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Müşteri gereksinimlerini belirledi.		
2. Use case diyagramlarını oluşturdu.		
3. Sistem gereksinimlerini belirledi.		

Tasarım adımı, projenin fiziksel ve mantıksal tasarımlarının yapıldığı bölümdür. **Fiziksel tasarım**, bileşenlerin ve arayüzlerin oluşturulması gibi son kullanıcıya yönelik tasarımlardır. **Mantıksal tasarım** ise kullanıcının görmediği ve sistemin mimarisinin, bir başka deyişle altyapısının oluşturulduğu tasarımlardır. Bu iki tasarımın birbirine bağımlı olmaması önemlidir. Bu durum, projenin değişime açık olmasını sağlar. Bir başka deyişle projeye esneklik kazandırır.

Tasarım adımında ürünün özelliklerinin ve yeteneklerinin belirlenmesi, veriler ve aralarındaki ilişkilerin organize edilmesi, kullanıcı arayüzlerinin tasarlanması gibi geliştirme süreci ile ilgili kritik işlemler yapılır (Görsel 1.6). Bütün bu yapılar, kullanıcı ihtiyaçlarına göre tasarlanmalıdır. Bu aşamada Birleşik Modelleme Dili (Unified Modeling Language-UML) diyagramları oluşturularak sistem görselleştirilebilir ve sistemin daha kolay anlaşılması sağlanır.



Görsel 1.6: Tasarım



Yazılım ürününün nasıl oluşturulacağı tasarım aşamasında netleşir. Uygulama geliştiriciler, bu tasarıma bağlı olarak kodlama işlemini gerçekleştirir. Bu aşama tamamlanmadan önce hatalar varsa düzeltilmelidir. Kodlama aşamasından sonra bu bölüme geri dönülerek hataların giderilmesi daha maliyetli olacaktır.

Tasarım aşamasında yapılabilecek işlemler şunlardır:

- Gereksinim analizinin yapılması
- Sistem modellemesinin yapılması (UML diyagramları oluşturularak yapılır.)
- Varlık ilişki şemalarının oluşturulması
- Veri akış diyagramlarının hazırlanması
- Sınıf diyagramlarının oluşturulması
- Kullanılacak teknik ve yöntemlerin belirlenmesi (VTYS, geliştirme ortamı, tasarım deseni, yazılım dili, veri erişim teknolojisi vb.)
- Veri tabanı tasarımının yapılması
- Kullanıcının isteklerine göre arayüzlerin tasarlanması
- Tasarım sürecinin gözden geçirilmesi ve varsa hataların düzeltilmesi

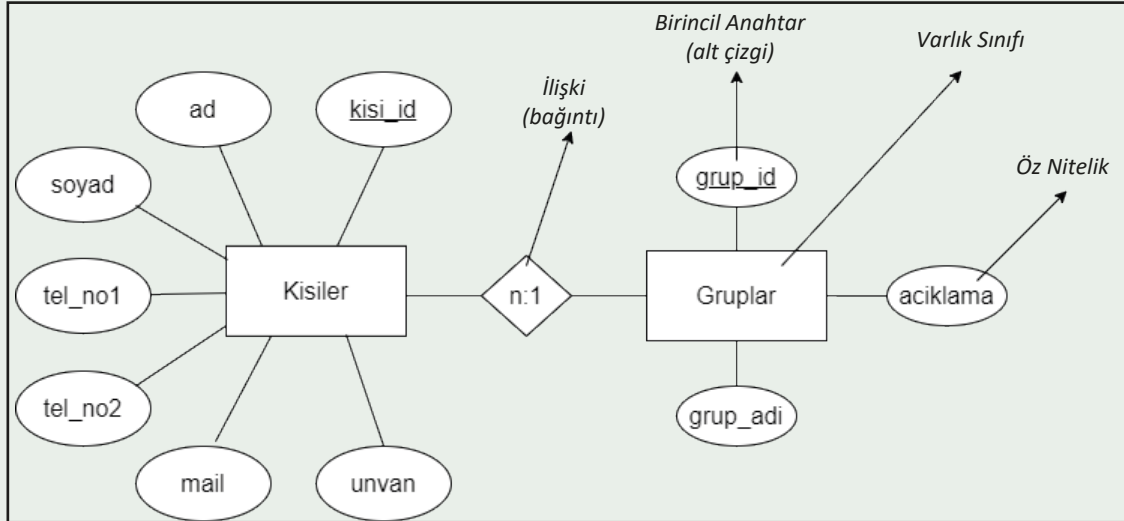


ÖRNEK

Telefon Rehberi projesi için tasarım aşaması şu şekilde gerçekleştirilir:

Tasarım aşamasında sistem modellemesi yapmak, projenin paydaşlar tarafından daha iyi anlaşılmasını sağlar. Grafikler ve diyagramlar kullanılarak sistemin genel yapısı gösterilir. Projenin büyüklüğüne ve karmaşıklığına göre UML diyagramlarından biri veya birkaçı kullanılarak sistem modellemesi yapılabilir. Proje ne kadar büyük ve karmaşık olursa sistemi modellemek için o kadar fazla UML diyagramına ihtiyaç duyulur.

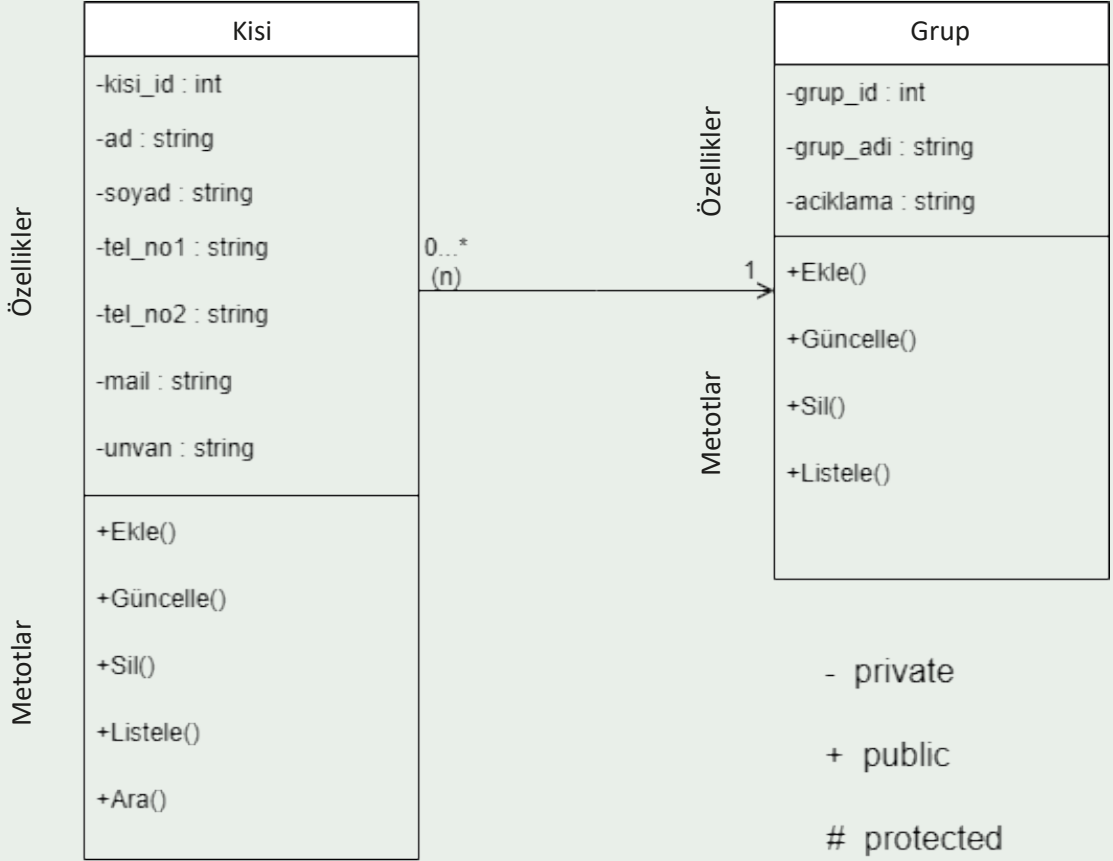
- Varlıkların özellikleri ve varlıklar arasındaki ilişkilerin grafiksel gösterimi için “varlık ilişki şemaları” kullanılır (Görsel 1.7).



Görsel 1.7: Telefon Rehberi projesine ait varlık ilişki şeması

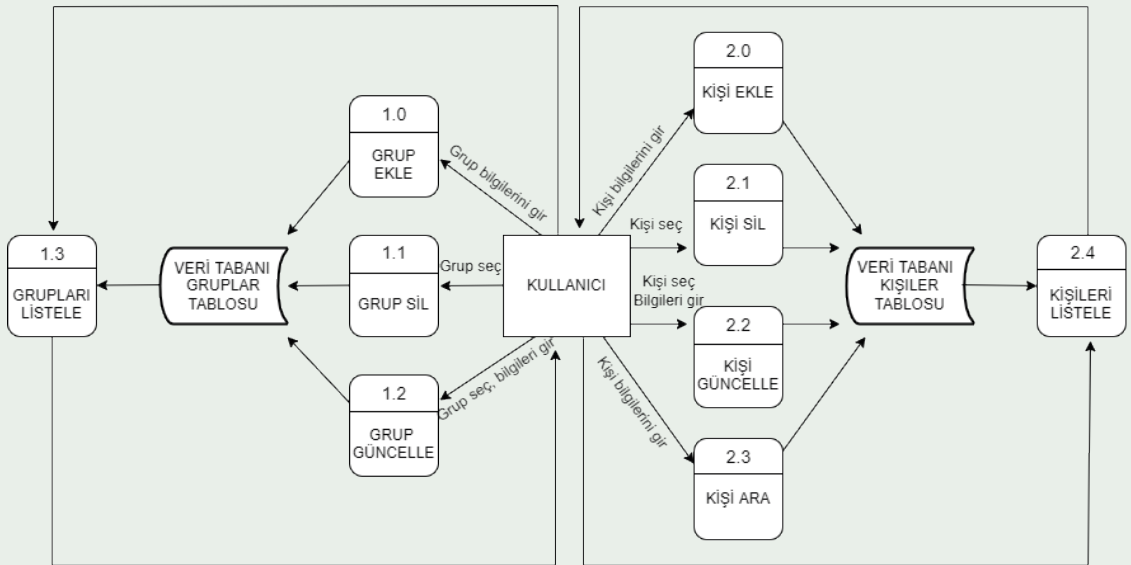


- Sınıfları, sınıflara ait özellikleri ve metotları ve bu sınıfların birbirleriyle bağlantılarını grafiklerle göstermek için “sınıf diyagramları” kullanılır (Görsel 1.8).



Görsel 1.8: Telefon rehberi projesine ait sınıf diyagramı

- Sistemdeki veri akışını grafiklerle göstermek için “veri akış diyagramları” kullanılır (Görsel 1.9).



Görsel 1.9: Telefon rehberi projesine ait veri akış diyagramı



- Kodlama aşamasında kolaylık sağlaması için kullanıcı arayüzleri (form ekranları), tasarım aşamasında taslak olarak oluşturulur (Görsel 1.10).

Görsel 1.10: Telefon rehberi projesine ait kullanıcı arayüzlerinin tasarımı



SIRA SİZDE

Personel Takip Programı projesi için tasarım işlemlerini gerçekleştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Varlık ilişki şemalarını oluşturdu.		
2. Sınıf diyagramlarını oluşturdu.		
3. Veri akış diyagramlarını oluşturdu.		
4. Kullanıcı arayüzlerinin tasarımını yaptı.		



1.2.3. Kodlama

Projede tasarım adımı tamamlandıktan sonra kodlama aşamasına geçilir. **Kodlama**, projenin ürün hâline getirildiği aşamadır (Görsel 1.11). Projeye uygun geliştirme ortamı seçilir ve tasarıma bağlı kalınarak belirlenen yazılım dili veya dilleri ile kodlama yapılır.



Görsel 1.11: Kodlama

Bu aşamada belirli bir standarda göre kodlama yapılması gerekir. Yazılan kodlar, diğer uygulama geliştiriciler tarafından kolaylıkla anlaşılabilmesi ve gerektiğinde projenin diğer bölümleri etkilenmeden değiştirilebilmelidir.

Kodlama aşaması temel olarak şu iki bölümden oluşur:

- Projenin arka yüzünün (sunucu etkileşimi, veri tabanı işlemleri ve ön uç veri alışverişinin yapıldığı kısım) kodlanması
- Projenin ön yüzünün (ziyaretçi tarafından görülen arayüz kısmı) kodlanması

Yazılım projesinde kodlama yapılırken yazılımın ön ve arka yüzünün birbirinden bağımsız olarak kodlanması, projenin değişimlere cevap verebilmesi açısından büyük avantaj sağlar. Bunlar, farklı geliştiriciler tarafından kodlanabileceği gibi özellikle küçük ölçekli projelerde tek kişi tarafından da geliştirilebilir. Proje tek kişi tarafından geliştirilse de etkin kod yazım stili için kullanılan yöntemler uygulanmalıdır. Bu durum, kodun okunabilirliğini artırır ve ileride kodda değişiklik yapılması gerektiğinde uygulama geliştiriciye kolaylık sağlar.

Etkin kod yazım stili için kullanılan yöntemler şunlardır:

- Sınıf, metot, değişken, veri tabanı tabloları gibi nesneler tanımlanırken anlamlı isimlendirmeler kullanılmalıdır. Örneğin içinde ürüne ait fiyat bilgisini saklayacak bir değişkene “fiyat” veya “urunFiyati” gibi bir isim verilmesi doğru olacaktır. Bir sınıfı adlandırmada “Urun” gibi isim türünde kelime kullanılırken metot adlandırmalarında ise “Kaydet” veya “Sil” gibi fiil türünde kelimeler kullanılmalıdır.
- Gerektiğinde yazılan kodlara açıklama satırı eklenerek bilgi verici açıklamalar yapılmalıdır.
- Yazılan kodlar düzenli olmalıdır.
- Yapılacak her işlem için en uygun algoritma seçilmelidir.
- Tekrarlanan işlemler için metot yapıları kullanılmalıdır ve her metodun bir tek görevi olmalıdır.
- Projede gereksiz işlemlere ve kodlara yer verilmemelidir.
- Özellikle orta ve büyük ölçekli projelerde sürdürülebilirlik ve verimlilik için Nesne Yönelimli Programlama (Object Oriented Programming-OOP) yapısı kullanılmalıdır.
- Projenin özelliğine ve büyüklüğüne göre gerekli görüldüğünde en uygun tasarım deseni kullanılmalıdır.
- Programın istenen dışında hata vermemesi için hata yönetimi yapılmalıdır.



ARAŞTIRMA

Temiz kod (Clean code) kavramını araştırınız.

Kodlama aşamasında yapılabilecek işlemler şunlardır:

- Veri tabanı ve tablolarının oluşturulması
- Sınıfların oluşturulması
- Veri tabanı işlemleri (ekleme, listeleme, güncelleme, silme, filtreleme) için gerekli kodlamanın yazılması
- Kullanıcı arayüzlerinin oluşturulması
- Arayüz sayfalarının arka plan kodlamalarının yapılması
- Olağan dışı durum çözümleme işlemlerinin yapılması
- Kodlanan her bir bölümün test edilmesi



NOT

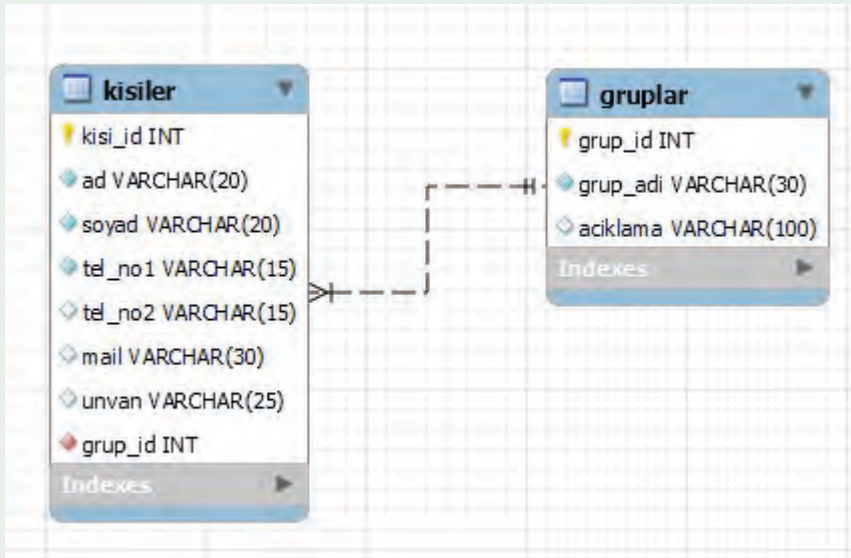
Özellikle birden fazla kişi tarafından geliştirilen orta ve büyük ölçekli projelerde versiyon kontrol sistemi kullanılması avantaj sağlayacaktır.



ÖRNEK

Telefon Rehberi projesi için kodlama aşaması şu şekilde gerçekleştirilir:

- “telefonrehberidb” isminde bir veri tabanı, veri tabanı tabloları ve bu tablolar arasındaki ilişki Görsel 1.12’deki gibi oluşturulur.



Görsel 1.12: Telefon rehberi projesine ait veri tabanı ve tabloların oluşturulması



- Veri tabanı bağlantı işlemini tek bir sınıf üzerinden yürütmek için “VeritabaniBaglantisi” sınıfı oluşturulur. Ayrıca bağlantı cümlesi “App.config” dosyasında saklanarak ihtiyaç duyulduğunda buradan çağrılır (Görsel 1.13). Program setup hâline getirilerek başka bilgisayarlara kurulduğunda bağlantı cümlesinin değişmesi gerekirse sadece bu dosyada değişiklik yapmak yeterlidir.

```

App.config" x
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>...</startup>
  <connectionStrings>
    <add name="TelefonRehberiBaglantiCumlesi" connectionString="Server=localhost;Database=telefonrehberidb;Uid=root;Pwd=19675561;"/>
  </connectionStrings>
</configuration>

namespace TelefonRehberi.Siniflar
{
    internal class VeritabaniBaglantisi
    {
        // App.config dosyasının içindeki veri tabanı bağlantı cümlesi bir değişkene aktarılır.
        string baglantiCumlesi = ConfigurationManager.ConnectionStrings["TelefonRehberiBaglantiCumlesi"].ConnectionString;

        public MySqlConnection baglan()
        {
            // Veri tabanı bağlantısı oluşturulur.
            MySqlConnection baglanti = new MySqlConnection(baglantiCumlesi);
            MySqlConnection.ClearPool(baglanti); // Önceki bağlantılar temizlenir.
            return baglanti; // Oluşturulan bağlantı fonksiyonun çağırıldığı yere gönderilir.
        }
    }
}

```

Görsel 1.13: Veri tabanı bağlantı sınıfının kodlanması

Varlıkları temsil etmek için oluşturulan “Kisi” ve “Grup” sınıfları, özelliklerin yanı sıra veri tabanı işlemlerinin yapılacağı metotları da içerir (Görsel 1.14). Bu sınıflar, veri tabanı işlemlerini yapabilmek için “VeritabaniBaglantisi” sınıfını kullanır.

```

namespace TelefonRehberi.Siniflar
{
    class Kisi
    {
        public int KisiId { get; set; }
        public string KisiAdi { get; set; }
        public string KisiSoyadi { get; set; }
        public string TelNo1 { get; set; }
        public string TelNo2 { get; set; }
        public string Mail { get; set; }
        public string Unvan { get; set; }
        public int GrupId { get; set; }
        VeritabaniBaglantisi veritabaniBaglantisi;
        MySqlConnection baglanti;
        MySqlCommand komut;
        public Kisi()
        {
            veritabaniBaglantisi = new VeritabaniBaglantisi();
            baglanti = veritabaniBaglantisi.baglan();
            komut = new MySqlCommand();
            komut.Connection = baglanti;
        }
        public void KisiEkle()...
        public void KisiyiGuncelle()...
        public void KisiSil()...
        public DataTable Kisileriliste()...
        public DataTable KisiAra() ...
    }
}

namespace TelefonRehberi.Siniflar
{
    class Grup
    {
        public int GrupId { get; set; }
        public string GrupAdi { get; set; }
        public string Aciklama { get; set; }

        VeritabaniBaglantisi veritabaniBaglantisi;
        MySqlConnection baglanti;
        MySqlCommand komut;
        public Grup()
        {
            veritabaniBaglantisi = new VeritabaniBaglantisi();
            baglanti = veritabaniBaglantisi.baglan();
            komut = new MySqlCommand();
            komut.Connection = baglanti;
        }

        public void GrupEkle()...
        public void GrubuGuncelle()...
        public void GrupSil()...
        public DataTable Gruplariliste()...
    }
}

```

Görsel 1.14: Kişi ve grup sınıflarının oluşturulması

- Projede veri tabanı ile ilgili ekleme, silme, güncelleme ve listeleme gibi işlemleri yapmak için gerekli metotlar sınıfların içine yazılır. Bu metotlar gerekli yerlerde çağrılarak veri tabanı işlemleri gerçekleştirilir. Veri tabanı işlemleri için farklı yöntemler vardır. Bu projede ADO.NET veriyeye erişim teknolojisi kullanılarak veri tabanı işlemleri yapılmıştır.



Uygulamada kişi grupları eklemek ve grupları listelemek için Görsel 1.15'teki metotlar oluşturulur.

```
public void GrupEkle()
{
    try
    {
        if (baglanti.State != ConnectionState.Open)
        {
            baglanti.Open();
        }
        komut.CommandText = "INSERT INTO gruplar(grup_adi,aciklama) " +
            "VALUES(@grupAdi,@aciklama)";
        komut.Parameters.AddWithValue("@grupAdi", GrupAdi);
        komut.Parameters.AddWithValue("@aciklama", Aciklama);
        komut.ExecuteNonQuery();
        MessageBox.Show("Kayıt işlemi başarılı");
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Kayıt işleminde hata oluştu",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    baglanti.Close();
}

public DataTable Gruplarilistele()
{
    try
    {
        komut.CommandText = "SELECT * FROM gruplar ORDER BY grup_adi ASC";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(komut);
        DataTable grupListesi = new DataTable();
        dataAdapter.Fill(grupListesi);
        return grupListesi;
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Listeleme işleminde hata oluştu",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        return null;
    }
}
```

Görsel 1.15: Grup sınıfı için ekleme ve listeleme metotları

Veri tabanı işlemleri için gerekli kodlamalar yapılırken olağan dışı durum çözümleme işlemleri de yapılmalı, program istenenin dışında hata vererek kapanmamalıdır.



Uygulamada kişi gruplarını güncellemek ve silmek için Görsel 1.16'daki metotlar oluşturulur.

```
public void GrubuGuncelle()
{
    try
    {
        if (baglanti.State != ConnectionState.Open)
        {
            baglanti.Open();
        }
        komut.CommandText = "UPDATE gruplar SET grup_adi=@grupAdi,aciklama=@aciklama " +
            "where grup_id=@grupId";
        komut.Parameters.AddWithValue("@grupAdi", GrupAdi);
        komut.Parameters.AddWithValue("@aciklama", Aciklama);
        komut.Parameters.AddWithValue("@grupId", GrupId);

        komut.ExecuteNonQuery();
        MessageBox.Show("Güncelleme işlemi başarılı");
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Güncelleme işleminde hata oluştu",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    baglanti.Close();
}

public void GrupSil()
{
    try
    {
        if (baglanti.State!=ConnectionState.Open)
        {
            baglanti.Open();
        }
        komut.CommandText = "DELETE FROM gruplar WHERE grup_id=@grupId";
        komut.Parameters.AddWithValue("@grupId", GrupId);
        komut.ExecuteNonQuery();
        MessageBox.Show("Silme işlemi başarılı");
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Silme işleminde hata oluştu",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    baglanti.Close();
}
```

Görsel 1.16: Grup sınıfı için güncelleme ve silme metotları



Uygulamada rehber eklemek ve kişileri listelemek için Görsel 1.17'deki metotlar oluşturulur.

```
public void KisiEkle()
{
    try
    {
        if (baglanti.State != ConnectionState.Open)
        {
            baglanti.Open();
        }
        komut.CommandText = "INSERT INTO kisiler(ad,soyad,tel_no1,tel_no2,mail,unvan,grup_id) " +
            "VALUES(@ad,@soyad,@telNo1,@telNo2,@mail,@unvan,@grupId)";
        komut.Parameters.AddWithValue("@ad", KisiAdi);
        komut.Parameters.AddWithValue("@soyad", KisiSoyadi);
        komut.Parameters.AddWithValue("@telNo1", TelNo1);
        komut.Parameters.AddWithValue("@telNo2", TelNo2);
        komut.Parameters.AddWithValue("@mail", Mail);
        komut.Parameters.AddWithValue("@unvan", Unvan);
        komut.Parameters.AddWithValue("@grupId", GrupId);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kayıt işlemi başarılı");
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Kayıt işleminde hata oluştu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

public DataTable Kisilerilistele()
{
    try
    {
        komut.CommandText = "SELECT kisi_id,ad,soyad,tel_no1,tel_no2,mail,unvan,grup_adi FROM kisiler, gruplar "+
            "WHERE kisiler.grup_id=gruplar.grup_id ORDER BY ad ASC,soyad ASC";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(komut);
        DataTable kisilistesi = new DataTable();
        dataAdapter.Fill(kisilistesi);
        return kisilistesi;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message,"Listeleme işleminde hata oluştu", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return null;
    }
}
```

Görsel 1.17: Kişi sınıfı için ekleme ve listeleme metotları

Uygulamada kişileri güncellemek ve silmek için Görsel 1.18'deki metotlar oluşturulur.

```
public void KisiyiGuncelle()
{
    try
    {
        if (baglanti.State != ConnectionState.Open)
        {
            baglanti.Open();
        }
        komut.CommandText = "UPDATE kisiler SET ad=@ad,soyad=@soyad,tel_no1=@telNo1,tel_no2=@telNo2," +
            "mail=@mail,unvan=@unvan,grup_id=@grupId where kisi_id=@kisiId";
        komut.Parameters.AddWithValue("@ad", KisiAdi);
        komut.Parameters.AddWithValue("@soyad", KisiSoyadi);
        komut.Parameters.AddWithValue("@telNo1", TelNo1);
        komut.Parameters.AddWithValue("@telNo2", TelNo2);
        komut.Parameters.AddWithValue("@mail", Mail);
        komut.Parameters.AddWithValue("@unvan", Unvan);
        komut.Parameters.AddWithValue("@grupId", GrupId);
        komut.Parameters.AddWithValue("@kisiId", KisiId);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Güncelleme işlemi başarılı");
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message,"Güncelleme işleminde hata oluştu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```



```

1 reference
public void KisiSil()
{
    try
    {
        if (baglanti.State != ConnectionState.Open)
            baglanti.Open();
        komut.CommandText = "DELETE FROM kisiler WHERE kisi_id=@kisiId";
        komut.Parameters.AddWithValue("kisiId", KisiId);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Silme işlemi başarılı");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme işleminde hata oluştu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Görsel 1.18: Kişi sınıfı için güncelleme ve silme metotları

Rehberdeki kişiler içinde arama yapmak için Görsel 1.19'daki "KisiAra" metodu oluşturulur.

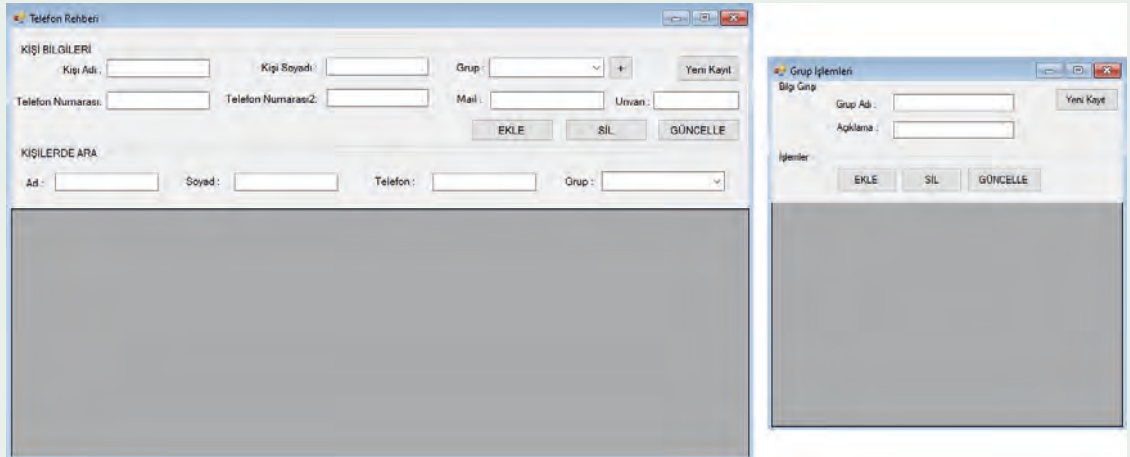
```

public DataTable KisiAra()
{
    try
    {
        // GrupId'nin 0 olması arama bölümünde "Tümü"nin seçildiği anlamı taşır. Bu durumda grup ile ilgili filtreleme yapılmaz.
        if (GrupId == 0) {
            komut.CommandText = "SELECT kisi_id,ad,soyad,tel_no1,tel_no2,mail,unvan,grup_adi FROM kisiler, gruplar " +
                "WHERE kisiler.grup_id=gruplar.grup_id and ad LIKE @ad and soyad LIKE @soyad and " +
                "(tel_no1 LIKE @telNo or tel_no2 LIKE @telNo) ORDER BY ad ASC,soyad ASC";
        }
        else { // Eğer değeri 0 değilse GrupId de filtreleme işlemine dâhil edilir.
            komut.CommandText = "SELECT kisi_id,ad,soyad,tel_no1,tel_no2,mail,unvan,grup_adi FROM kisiler, gruplar " +
                "WHERE kisiler.grup_id=gruplar.grup_id and ad LIKE @ad and soyad LIKE @soyad and " +
                "(tel_no1 LIKE @telNo or tel_no2 LIKE @telNo) and gruplar.grup_id=@grupId ORDER BY ad ASC,soyad ASC";
        }
        komut.Parameters.AddWithValue("@ad", KisiAdi + "%");
        komut.Parameters.AddWithValue("@soyad", KisiSoyadi + "%");
        komut.Parameters.AddWithValue("@telNo", TelNo1 + "%");
        komut.Parameters.AddWithValue("@grupId", GrupId + "%");
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(komut);
        DataTable arananKisilistesi = new DataTable();
        dataAdapter.Fill(arananKisilistesi);
        return arananKisilistesi;
    }
    catch (Exception ex) {
        MessageBox.Show(ex.Message, "Arama işleminde hata oluştu", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return null;
    }
}

```

Görsel 1.19: Kişi sınıfında arama metodunun oluşturulması

- Tasarım aşamasında oluşturulan taslağa göre kullanıcı arayüzleri oluşturulur (Görsel 1.20).



Görsel 1.20: Kişi ve grup işlemleri için kullanıcı arayüzlerinin oluşturulması



- Sınıfların içinde oluşturulan metotlar, form nesnelerinin ilgili olaylarında (Load, Click vb.) çağrılarak kullanılır. Tekrar eden işlemler için yeni metotlar oluşturulabilir. Ayrıca kodlanan her bölümün doğruluğu test edilmelidir.
- Grup işlemleri formunun arka planında Görsel 1.21'deki olay ve metotlar oluşturulur.

```
namespace TelefonRehberiUygulaması
{
    public partial class frmGrupIslemleri : Form
    {
        public frmGrupIslemleri()
        {
            InitializeComponent();
        }
        //Silme veya güncelleme işlemi için bir grubun seçilip seçilmediğini kontrol etmek için değişken oluşturulur.
        int seciliGrupId = -1; // -1 değeri DataGridView'de seçili grup olmadığını gösterir.

        private void frmGrupIslemleri_Load(object sender, EventArgs e) {...}

        private void btnEkle_Click(object sender, EventArgs e) {...}

        // Güncelleme ve silme işlemi için DataGridView üzerinde seçilen kişilerin bilgileri form elemanlarına aktarılır.
        private void grdGruplar_CellClick(object sender, DataGridViewCellEventArgs e) {...}

        private void btnSil_Click(object sender, EventArgs e) {...}

        private void btnGuncelle_Click(object sender, EventArgs e) {...}

        private void btnYeni_Click(object sender, EventArgs e) {...}

        private void FormuTemizle() {...}
    }
}
```

Görsel 1.21: Grup işlemleri formunun olay ve metotları

Grup işlemleri formunda daha önceden oluşturulan formun yüklenmesi ve “EKLE” butonunun tıklanması olayları Görsel 1.22'deki gibi kodlanır.

```
private void frmGrupIslemleri_Load(object sender, EventArgs e)
{ // Form yüklendiğinde gruplar DataGridView nesnesinde listelenir.
    Grup grup = new Grup();
    grdGruplar.DataSource = grup.GruplariListele();
    grdGruplar.Columns["grup_id"].HeaderText = "ID";
    grdGruplar.Columns["grup_id"].Width = 30;
    grdGruplar.Columns["grup_adi"].HeaderText = "Grup Adı";
    grdGruplar.Columns["grup_adi"].Width = 100;
    grdGruplar.Columns["aciklama"].HeaderText = "Açıklama";
}

1 reference
private void btnEkle_Click(object sender, EventArgs e)
{ // Ekle butonu tıklandığında bilgileri girilen grup kaydedilir.
    Grup grup = new Grup();
    if (txtGrupAdi.TextLength < 2)
        MessageBox.Show("Grup adını giriniz.");
    else{
        grup.GrupAdi = txtGrupAdi.Text;
        grup.Aciklama = txtAciklama.Text;
        grup.GrupEkle();
        grdGruplar.DataSource = grup.GruplariListele();
        FormuTemizle();
    }
}
```



```

4 references
private void FormuTemizle()
{ // Form temizleme işlemi birden fazla kullanılacağı için metod hâlinde yazılır.
    txtGrupAdi.Clear();
    txtAciklama.Clear();
    grdGruplar.ClearSelection();
    seciliGrupId = -1;
}
1 reference
private void btnYeni_Click(object sender, EventArgs e){
    FormuTemizle();
}

```

Görsel 1.22: Formun yüklenmesi ve ekle butonun tıklanması

Grupların listelendiği DataGridView nesnesinde seçilen bir kaydın silinebilmesi veya güncellenebilmesi için gerekli kodlar Görsel 1.23'teki gibi yazılır.

```

// Güncelleme ve silme işlemi için DataGridView üzerinde seçilen kişilerin bilgileri form elemanlarına aktarılır.
1 reference
private void grdGruplar_CellClick(object sender, DataGridViewCellEventArgs e) {
    try{
        seciliGrupId = Convert.ToInt32(grdGruplar.CurrentRow.Cells["grup_id"].Value.ToString());
        txtGrupAdi.Text = grdGruplar.CurrentRow.Cells["grup_adi"].Value.ToString();
        txtAciklama.Text = grdGruplar.CurrentRow.Cells["aciklama"].Value.ToString();
    }
    catch (Exception ex){
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
1 reference
private void btnSil_Click(object sender, EventArgs e){
    if (seciliGrupId != -1) { // DataGridView nesnesinde bir grup seçilmişse silme işlemi yapılır.
        Grup grup = new Grup();
        grup.GroupId = seciliGrupId;
        grup.GroupSil();
        grdGruplar.DataSource = grup.GruplariListele();
        FormuTemizle();
    }
}
1 reference
private void btnGuncelle_Click(object sender, EventArgs e) {
    if (seciliGrupId != -1) { // DataGridView nesnesinde bir grup seçilmişse güncelleme işlemi yapılır.
        if (txtGrupAdi.TextLength < 2){
            MessageBox.Show("Grup adını giriniz.");
        }
        else{
            Grup grup = new Grup { // Sınıf özelliklerine değer atamanın daha kısa bir yöntemidir.
                GroupId = seciliGrupId,
                GroupAdi = txtGrupAdi.Text,
                Aciklama = txtAciklama.Text };
            grup.GrubuGuncelle();
            grdGruplar.DataSource = grup.GruplariListele();
            FormuTemizle();
        }
    }
}
}

```

Görsel 1.23: Grup listesinde seçilen kaydın silinmesi ve güncellenmesi



- Telefon rehberi formunda kişilerle ilgili işlemler (ekleme, silme, güncelleme, listeleme ve arama) yapılır. İlk olarak formun kod bölümünde olaylar ve metotlar oluşturulur (Görsel 1.24).

```
public partial class frmTelefonRehberi : Form
{
    public frmTelefonRehberi()...
    int seciliKisiId = 0; // Seçili kişi olup olmadığını kontrol edecek olan bir değişken tanımlanır.
    private void frmTelefonRehberi_Load(object sender, EventArgs e)...
```

Görsel 1.24: Telefon rehberi formunun olay ve metotları

Form yüklenirken yapılacak işlem, kişilerin ve kişiler için seçilecek grupların gerekli form elemanlarında listelenmesidir (Görsel 1.25).

```
private void frmTelefonRehberi_Load(object sender, EventArgs e)
{
    BilgileriYukle();
}
2 references
private void BilgileriYukle()
{
    Grup grup = new Grup();
    cmbGrup.DataSource = grup.Gruplariliste();
    cmbGrup.ValueMember = "grup_id";
    cmbGrup.DisplayMember = "grup_adi";
    DataTable araGrupDt = grup.Gruplariliste();
    DataRow dr = araGrupDt.NewRow();
    dr["grup_adi"] = "Tümü"; // Arama bölümündeki grup listesine "Tümü" isminde bir eleman eklenir.
    dr["grup_id"] = 0; // Bu id değeri ile kişi arama metodunda bütün gruplara ait kişiler listelenir.
    araGrupDt.Rows.Add(dr);
    cmbAraGrup.DataSource = araGrupDt;
    cmbAraGrup.ValueMember = "grup_id";
    cmbAraGrup.DisplayMember = "grup_adi";
    cmbAraGrup.SelectedValue = 0;
    Kisi kisi = new Kisi();
    grdKisiler.DataSource = kisi.Kisileriliste();
    // DataGridView nesnesindeki sütunların başlıklarını ve genişliklerini ayarlayacak kodlar yazılır.
    grdKisiler.Columns["kisi_id"].HeaderText = "";
    grdKisiler.Columns["kisi_id"].Width = 0;
    grdKisiler.Columns["ad"].HeaderText = "Kişi Adı";
    grdKisiler.Columns["ad"].Width = 120;
    grdKisiler.Columns["soyad"].HeaderText = "Kişi Soyadı";
    grdKisiler.Columns["soyad"].Width = 90;
    grdKisiler.Columns["tel_no1"].HeaderText = "Telefon No 1";
    grdKisiler.Columns["tel_no1"].Width = 110;
    grdKisiler.Columns["tel_no2"].HeaderText = "Telefon No 2";
    grdKisiler.Columns["tel_no2"].Width = 110;
    grdKisiler.Columns["mail"].HeaderText = "Mail";
    grdKisiler.Columns["mail"].Width = 170;
    grdKisiler.Columns["unvan"].HeaderText = "Unvan";
    grdKisiler.Columns["unvan"].Width = 100;
    grdKisiler.Columns["grup_adi"].HeaderText = "Kişi Grubu";
}
```

Görsel 1.25: Telefon rehberi formunun Load [Loud (Yükleme)] olayının kodlanması



Telefon rehberi formunun bilgi girişi bölümünde bulunan “+” butonu tıklandığında grup işlemlerinin yapılacağı sayfanın açılması sağlanır. Ayrıca grup işlemleri tamamlandıktan sonra yapılan değişikliklerin uygulanması için gerekli olan metod, formun kapanması durumunda çağrılır (Görsel 1.26).

```
private void btnGrupIslemleri_Click(object sender, EventArgs e)
{
    frmGrupIslemleri frmGrup = new frmGrupIslemleri();
    DialogResult result = frmGrup.ShowDialog();
    if (result==DialogResult.Cancel)
    {
        BilgileriYukle();
    }
}
```

Görsel 1.26: Grup işlemleri formunun gösterilmesi

Telefon rehberine kişi ekleme ve eklenen kişileri güncelleme işlemlerinden önce bilgilerin doğru girilip girilmediği kontrol edilmelidir. Bunun için “KayitKontrol” isimli bir metod oluşturulur (Görsel 1.27).

```
private bool KayitKontrol()
{
    bool kontrol = false;
    if (txtAd.TextLength < 2)
    {
        MessageBox.Show("Kişinin adını giriniz.");
    }
    else if (txtSoyad.TextLength < 2)
    {
        MessageBox.Show("Kişinin soyadını giriniz.");
    }
    else if (txtNumara1.TextLength < 10 || txtNumara1.TextLength > 14)
    {
        MessageBox.Show("Kişinin numarasını doğru giriniz (10-14 karakter olmalıdır).");
    }
    else if (cmbGrup.SelectedIndex < 0)
    {
        MessageBox.Show("Kişinin grubunu seçiniz.");
    }
    else
    {
        kontrol = true;
    }
    return kontrol;
}
```

Görsel 1.27: Telefon rehberindeki kişi bilgilerinin kontrolü



Telefon rehberi form ekranındaki telefon numarası alanlarına rakam dışında bilgi girişini engellemek için gerekli kodlar Görsel 1.28’deki gibi yazılır.

```
// Numara girişi yapılacak olan alanlara yalnızca rakam girişi yapılması sağlanır.  
1 reference  
private void txtNumara1_KeyPress(object sender, KeyPressEventArgs e)  
{  
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);  
}  
1 reference  
private void txtNumara2_KeyPress(object sender, KeyPressEventArgs e)  
{  
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);  
}  
1 reference  
private void txtAraNumara_KeyPress(object sender, KeyPressEventArgs e)  
{  
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);  
}
```

Görsel 1.28: Telefon numarası alanlarına yalnızca rakam girişi yapılmasını sağlayan kodlar

Telefon rehberi form ekranında “Yeni Kayıt” butonuna tıklanarak form elemanları temizlenir. Bilgi girişi yapıldıktan sonra bilgilerin kaydedilebilmesi için “EKLE” butonunun tıklanma olayına Görsel 1.29’daki kodlar yazılır. Kişi, telefon rehberine kaydedilmeden önce bilgilerin doğruluğu kontrol edilir.

```
private void btnYeni_Click(object sender, EventArgs e)  
{  
    FormuTemizle();  
}  
1 reference  
private void btnEkle_Click(object sender, EventArgs e)  
{  
    if (KayitKontrol())  
    {  
        Kisi kisi = new Kisi();  
        kisi.KisiAdi = txtAd.Text;  
        kisi.KisiSoyadi = txtSoyad.Text;  
        kisi.TelNo1 = txtNumara1.Text;  
        kisi.TelNo2 = txtNumara2.Text;  
        kisi.Mail = txtMail.Text;  
        kisi.Unvan = txtUnvan.Text;  
        kisi.GrupId = Int32.Parse(cmbGrup.SelectedValue.ToString());  
        kisi.KisiEkle();  
        grdKisiler.DataSource = kisi.KisileriListele();  
        FormuTemizle();  
    }  
}
```

Görsel 1.29: Telefon rehberine kişi ekleme kodları



İşlemler yapıldıktan sonra telefon rehberi formundaki bilgilerin ve seçimlerin temizlenmesi için “FormuTemizle” metodu yazılır ve bu metod, gerekli bölümlerde çağrılarak kullanılır (Görsel 1.30).

```
private void FormuTemizle()
{
    txtAd.Clear();
    txtSoyad.Clear();
    txtNumara1.Clear();
    txtNumara2.Clear();
    txtMail.Clear();
    txtUnvan.Clear();

    txtAraAd.Clear();
    txtAraSoyad.Clear();
    txtAraNumara.Clear();

    grdKisiler.ClearSelection();
    seciliKisiId = -1;
}
```

Görsel 1.30: Telefon rehberi form elemanlarının içeriklerinin ve seçimlerin temizlenmesi

Silme veya güncelleme işlemi yapılmadan önce “DataGridView” nesnesi üzerinde işlem yapılacak kişinin seçilmesi ve seçilen kişiye ait bilgilerin form elemanlarında gösterilmesi gerekir. Bunun için gerekli kodlar Görsel 1.31’deki gibi yazılır.

```
// Güncelleme ve silme işlemi için seçilen kişilerin bilgileri form elemanlarına aktarılır.
1 reference
private void grdKisiler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdKisiler.SelectedRows != null)
        {
            seciliKisiId = Convert.ToInt32(grdKisiler.CurrentRow.Cells["kisi_id"].Value.ToString());
            txtAd.Text = grdKisiler.CurrentRow.Cells["ad"].Value.ToString();
            txtSoyad.Text = grdKisiler.CurrentRow.Cells["soyad"].Value.ToString();
            txtNumara1.Text = grdKisiler.CurrentRow.Cells["tel_no1"].Value.ToString();
            txtNumara2.Text = grdKisiler.CurrentRow.Cells["tel_no2"].Value.ToString();
            txtMail.Text = grdKisiler.CurrentRow.Cells["mail"].Value.ToString();
            txtUnvan.Text = grdKisiler.CurrentRow.Cells["unvan"].Value.ToString();
            cmbGrup.Text = grdKisiler.CurrentRow.Cells["grup_adi"].Value.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Olustu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 1.31: Seçilen kişinin bilgilerinin form elemanlarına aktarılması



DataGridView nesnesi üzerinde seçilen bir kaydın silinmesi için “SİL” butonunun tıklanması olayına Görsel 1.32’deki kodlar yazılır.

```
private void btnSil_Click(object sender, EventArgs e)
{ // Seçili bir kişi varsa silme işlemi gerçekleştirilir.
    if (seciliKisiId != -1)
    {
        Kisi kisi = new Kisi();
        kisi.KisiId = seciliKisiId;
        kisi.KisiSil();
        grdKisiler.DataSource = kisi.KisileriListele();
        FormuTemizle();
    }
}
```

Görsel 1.32: Telefon rehberinden kişi silme kodları

DataGridView nesnesi üzerinde seçilen kaydın bilgileri form elemanlarına aktarılır. Yeni bilgiler girildikten sonra “GÜNCELLE” butonu tıklanarak güncelleme işlemi yapılır. Güncelleme işlemi için butonun tıklanması olayına Görsel 1.33’teki kodlar yazılır.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    // Seçili bir kişi varsa güncelleme işlemi gerçekleştirilir.
    if (seciliKisiId != -1)
    {
        if (KayitKontrol())
        {
            Kisi kisi = new Kisi();
            kisi.KisiId = seciliKisiId;
            kisi.KisiAdi = txtAd.Text;
            kisi.KisiSoyadi = txtSoyad.Text;
            kisi.TelNo1 = txtNumara1.Text;
            kisi.TelNo2 = txtNumara2.Text;
            kisi.Mail = txtMail.Text;
            kisi.Unvan = txtUnvan.Text;
            kisi.GrupId = Int32.Parse(cmbGrup.SelectedValue.ToString());
            kisi.KisiyiGuncelle();
            grdKisiler.DataSource = kisi.KisileriListele();
            FormuTemizle();
        }
    }
}
```

Görsel 1.33: Telefon rehberinde kişi güncelleme kodları

Arama bölümündeki form elemanlarından herhangi birine bilgi girişi yapıldığı anda arama işlemi gerçekleştirilir. Bunun için arama işlemi yapılacak bütün form elemanlarının içeriğinin değiştirilmesi olayına arama kodları yazılır (Görsel 1.34).



```
private void KisiAra()
{
    Kisi kisi = new Kisi();
    kisi.KisiAdi = txtAraAd.Text;
    kisi.KisiSoyadi = txtAraSoyad.Text;
    kisi.TelNo1 = txtAraNumara.Text;
    kisi.GrupId = Int32.Parse(cmbAraGrup.SelectedValue.ToString());
    grdKisiler.DataSource = kisi.KisiAra();
}
1 reference
private void txtAraAd_TextChanged(object sender, EventArgs e)
{
    KisiAra();
}
1 reference
private void txtAraSoyad_TextChanged(object sender, EventArgs e)
{
    KisiAra();
}
1 reference
private void txtAraNumara_TextChanged(object sender, EventArgs e)
{
    KisiAra();
}
1 reference
private void cmbAraGrup_SelectionChangeCommitted(object sender, EventArgs e)
{
    KisiAra();
}
```

Görsel 1.34: Telefon rehberinde kişi arama kodları

**NOT**

Proje yayımlanmadan önce tablolardaki veriler silinerek id değerlerinin 1'den başlaması istenirse "TRUNCATE TABLE tablo_adı" komutunun çalıştırılması gerekir.



SIRA SİZDE

Planlaması yapılan bir "Personel Takip Programı" için veri tabanı oluşturma ve programlama şeklinde kodlama aşamalarını gerçekleştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Veri tabanını oluşturdu.		
2. Veri tabanı tablolarını oluşturdu.		
3. Sınıflar ve özelliklerini oluşturdu.		
4. Veri tabanı işlemleri için gerekli metotları oluşturdu.		
5. Kullanıcı arayüzlerini oluşturdu.		
6. Arayüzlerin kod bölümünde daha önce oluşturulan metotları kullanarak veri tabanı işlemlerini yaptı.		

1.2.4. Test

Projede kodlama işlemi tamamlandıktan sonra test aşamasına geçilir. Bu aşamada uygulamanın amaç ve hedeflerine uygun olarak çalışıp çalışmadığı test edilir. Projedeki eksikler ve hatalar bu aşamada ortaya çıkarılır ve belgelenir (Görsel 1.35). Sistemin bütün işlevlerinin çalışıp çalışmadığının ve sistemin güvenliğinin test edilmesi gerekir. Belirlenen girdiler karşılığında istenen çıktıların alınıp alınmadığı kontrol edilir. Test adımı ortaya çıkarılan hatalar hangi aşamada yapılmışsa o adıma geri dönülerek hatalar düzeltilir. Test aşaması tamamlanmadan proje yayımlanmamalıdır.



Görsel 1.35: Test



Yazılım projesinin yanlış çalışmasına neden olan hatalar, kodlama aşamasında meydana gelebileceği gibi genellikle ihtiyaçların analiz edilmesinde ve tasarım aşamasında yapılan yanlışlıklardan kaynaklanır. İyi analiz edilmeyen ihtiyaçlar ve yapılan kötü tasarım, projenin yanlış inşa edilmesine sebep olur. Bu durum, ürünün doğru çalışmasını engeller.



ARAŞTIRMA

Test yazılımlarını araştırınız. Araştırma sonuçlarınızı arkadaşlarınızla karşılaştırınız.

Proje hazırlandıktan ve test edildikten sonra kullanıcı kabul testi yapılır. Müşteri, yazılımı test eder ve varsa hataları, düzeltilmesini istediği bölümleri bildirir. Bu bölümler ve hatalar düzeltildikten sonra proje yayımlanabilir.



ÖRNEK

Telefon Rehberi projesi için test aşaması şu şekilde gerçekleştirilir:

- Projenin grup sayfasında ekleme, silme, güncelleme ve listeleme fonksiyonlarının doğru çalışıp çalışmadığı test edilir (Görsel 1.36).
- Projenin telefon rehberi sayfasında ekleme, silme, güncelleme, listeleme ve arama fonksiyonlarının çalışıp çalışmadığı test edilir (Görsel 1.37, Görsel 1.38).
- Bilgi girişi kontrollerinin yapıp yapılmadığı test edilir.
- Projede olmayan dış durum çözümü işlemlerinin yapıp yapılmadığı test edilir.

The image shows three screenshots of the 'Grup İşlemleri' (Group Operations) form, illustrating different test scenarios:

- Left Screenshot:** The 'Yeni Grup' (New Group) button is highlighted. A dialog box titled 'Kayıt işlemi başarılı' (Record operation successful) is shown, indicating a successful addition of a new group.
- Middle Screenshot:** The 'GÜNCELLE' (Update) button is highlighted. A dialog box titled 'Güncelleme işlemi başarılı' (Update operation successful) is shown, indicating a successful update of an existing group.
- Right Screenshot:** The 'SİL' (Delete) button is highlighted. A dialog box titled 'Silme işlemi başarılı' (Delete operation successful) is shown, indicating a successful deletion of a group.

Each screenshot shows the form with fields for 'Grup Adı' (Group Name) and 'Açıklama' (Description), and a table of existing groups with columns for ID, Group Name, and Description.

Görsel 1.36: Grup işlemleri formunun test edilmesi



Telefon Rehberi

KİŞİ BİLGİLERİ

Kişi Adı: Esat Kişi Soyadı: E. Grup: Öğrenciler + Yeni Kayıt

Telefon Numarası: 05695241236 Telefon Numarası2: Mail: esate@ornek.com Unvan: Elektrik Mühendisi

EKLE SİL GÜNCELLE

KİŞİLERDE ARA

Ad: Soyad: Telefon: Grup: Tümü

Kişi Adı	Kişi Soyadı	Telefon No 1	Telefon No 2	Mail	Unvan	Kişi Grubu
Ahmet	T.	0455615561		ahmett@deneme.com		
Ali	Ö.	07556246161		alio@ornek.com	Mu	
Ali	Ö.	6985200003			Öğ	
Ali	Y.	9853652033				
Arzu	D.	08965214578	3458741203			
Emre	K.	0444343444		emrek@ornek.com		
Eren	T.	1652163456	6589652103		Öğretmen	İş Arkadaşları
Esat	E.	3608778987		esate@ornek.com	Elektrik Müh.	Öğrenciler
Fatma	B.	784521362003				Aile
fds	dfsd	4534534534534				İş Arkadaşları
Fuat	Ş.	36565020341		sfuat@deneme.com	Öğretmen	Arkadaşlar
Hacer	A.	741256325896	9632541254			İş Arkadaşları

Güncelleme işlemi başarılı

Tamam

Görsel 1.37: Telefon rehberi formunda güncelleme işleminin test edilmesi

Telefon Rehberi

KİŞİ BİLGİLERİ

Kişi Adı: Kişi Soyadı: Grup: Öğrenciler + Yeni Kayıt

Telefon Numarası: Telefon Numarası2: Mail: Unvan:

EKLE SİL GÜNCELLE

KİŞİLERDE ARA

Ad: yası Soyad: Telefon: Grup: Okul

Kişi Adı	Kişi Soyadı	Telefon No 1	Telefon No 2	Mail	Unvan	Kişi Grubu
Yasemin	Z.	060032185624		ysmnm@deneme.com		Okul
Yasin	Y.	6542318547	5112334323		Müdür	Okul

Görsel 1.38: Telefon rehberi formunda arama işleminin test edilmesi

Projenin test aşaması tamamlandıktan sonra varsa hataları düzeltilir ve proje tekrar test edilir. Bir hata yoksa proje setup (kurulum dosyası) hâline getirilir ve kurulumu yapılır. Bir başka deyişle proje yayımlanır.



SIRA SİZDE

Personel Takip Programı projesinin test aşamasını gerçekleştiriniz ve projeyi setup hâline getiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Projenin müşteri ihtiyaçlarını karşılayıp karşılamadığını test etti.		
2. Uygulamanın istenen dışında hata vererek kapanıp kapanmadığını kontrol etti.		
3. Tespit edilen eksiklikleri giderdi.		
4. Tespit edilen hataları düzeltti.		
5. Ürünü, kurulum dosyası hâline getirdi.		

1.2.5. Bakım

Bakım aşaması, proje yayıma alındıktan sonraki süreci kapsar. Proje uygulama sürecinin son adımı olan bakım aşamasında proje yayımlandıktan sonra ortaya çıkan hataların giderilmesi, müşteri ihtiyaçları doğrultusunda yazılımda iyileştirmeler yapılması ve ürüne bazı işlevlerin eklenmesi gibi işlemler yapılır (Görsel 1.39). İyi tasarlanmış ve doğru kodlanmış bir ürünün bakımını yapmak kolaydır. Proje, değişikliğe cevap verebilecek şekilde tasarlanmamışsa veya etkin kod yazım stilleri kullanılarak kodlanmamışsa bu ürünün bakımını yapmak zor olacaktır.



Görsel 1.39: Bakım



NOT

Yazılım projesi uygulama sürecinde projenin türüne, büyüklüğüne, personel özelliklerine ve teknik altyapıya uygun olan bir yazılım geliştirme süreç modeli kullanılabilir. Yazılım geliştirme süreç modellerinden bazıları şunlardır:

- Kodla ve Düzelt (Code and Fix) Modeli
- Şelale (Waterfall) Modeli
- V Modeli
- Sarmal (Spiral) Model
- Prototipleme (Prototyping)
- Yinelemeli (Iterative) Model
- Artımlı (Incremental) Model
- Çevik (Agile) Modeller
 - * Scrum
 - * Kanban
 - * XP (Extreme Programming)



ARAŞTIRMA

Yazılım geliştirme süreç modellerinin birbirlerine göre avantaj ve dezavantajlarını araştırınız. Edindiğiniz bilgileri arkadaşlarınızla değerlendiriniz.

1.3. PROJE SONUÇ RAPORU

Yazılım projesi tamamlandıktan sonra projenin özetini, ürünün ve sistemin gereksinimlerini, proje planını, uygulanan adımları ve proje sürecinin sonunda elde edilen sonuçlar gibi projenin bütün aşamalarını içeren detaylı bir proje raporu hazırlanmalıdır. Proje raporundaki bilgiler doğru ve tutarlı olmalıdır. Bu rapor, proje ile ilgili bütün bilgileri içermeli fakat raporda gereksiz bilgiler yer almamalıdır. Ayrıca rapor içeriği basit ve anlaşılır olmalı; grafik ve görsellerle desteklenmelidir. Hazırlanan rapor, teknik bilgiye sahip olmayan kişiler tarafından bile kolaylıkla anlaşılmalıdır.

Rapor hazırlamak, bütün paydaşların projenin süreçlerine hâkim olmasına olanak sağlar. Yazılım projesini raporlamak, raporu inceleyen kişiye proje hakkında fikir vereceği gibi bakım ihtiyacı oluştuğunda değişikliklerin daha kolay yapılmasını sağlar. Örneğin yıllar önce yapılmış büyük ölçekli bir projede bakım ihtiyacı oluştuğu ve projede görev almış personellerin şirketten ayrıldığı varsayılırsa böyle bir durumda proje hakkında hiçbir bilgisi olmayan personel veya personellerin bakım aşamasını gerçekleştirmesi gerekecektir. Bu projeye ait iyi hazırlanmış bir proje raporu, personellerin projede değişiklik yapmasını kolaylaştıracaktır. Aksi hâlde personelin öncelikle proje içindeki sistemi çözmek için zaman harcaması gerekir.

Projenin türüne ve büyüklüğüne göre farklılıklar gösterse de proje sonuç raporu genel olarak şu aşamalardan oluşur:

1. GİRİŞ

Giriş bölümü, projenin tanıtıldığı aşamadır. Bu aşamada projenin amacı, kapsamı hangi probleme çözüm üreteceği, müşteri ve kullanıcıya sağlayacağı faydalar basit bir dille açıklanmalıdır.

2. ARAÇLAR VE YÖNTEMLER

Bu bölüm, proje ile ilgili teknik detayları içerir. Proje oluşturulurken hangi araçların kullanıldığı ve projenin



nasıl oluşturulduğu bu bölümde açıklanır. Ayrıca kullanılan metot, teknik ve yazılım dilleri de tercih edilme nedenleriyle birlikte bu aşamada belirtilebilir.

Araçlar ve yöntemler bölümünde müşteri gereksinimleri, sistemin donanımsal ve yazılımsal gereksinimleri, ürünün işlevleri, kullanım durumları gibi projeyi hazırlama aşamasındaki ihtiyaçlar da raporlanabilir. Ayrıca kullanıcı senaryoları ve kullanım durumu diyagramları oluşturularak ürünün fonksiyonlarının daha iyi gösterilmesi sağlanır.

Tasarım aşamasında oluşturulan diyagramlar da araçlar ve yöntemler bölümüne eklenebilir. Projenin görselleri de yine bu bölümde gösterilebilir ve projenin çalışma mantığı anlatılır.

3. SONUÇ

Bu bölümde ürünün hedefleri, özellikleri ve beklenen sonuçlar açıklandıktan sonra bu hedeflere ne kadar ulaşıldığı belirlenir. Müşteri gereksinimleri ve ürünün işlevlerine göre yazılım ürünü test edilerek sonuçlar raporlanır. Ayrıca ürün ile ilgili hatalar, eksiklikler ve bunlar için çözüm önerileri de sonuç bölümünde yer alır. Sonuç raporuna göre üründe iyileştirmeler yapılabilir. Projenin sürdürülebilir olup olmadığı da sonuç aşamasında ortaya çıkarılır.

4. SÖZLÜK

Herkes tarafından anlaşılamayacak teknik terimlerin veya kısaltmaların açıklandığı bölümdür. Rapor içeriğinde bu durumdaki bütün terim ve kısaltmalar açıklanmalı, raporun anlaşılır olması sağlanmalıdır.

5. KAYNAKÇA

Bu bölümde proje ve rapor oluşturulurken kullanılan kaynaklar belirtilmelidir. Bilgilerin toplandığı kitap, makale, tez, web sitesi gibi kaynaklar bu aşamada yazılır.



NOT

Orta ve büyük ölçekli projelerde, proje sonuç raporundaki sayfa sayısı fazla olacağı için raporda "İÇİNDEKİLER" bölümü oluşturulabilir.

Telefon rehberi uygulamasına ait örnek proje raporu çerçeve içinde verilmiştir.

PROJE ADI: TELEFON REHBERİ

1. GİRİŞ

Projede kullanıcının telefon numaralarını kaydedebileceği bir masaüstü uygulamasının geliştirilmesi amaçlanmıştır. Kullanıcı bu uygulamayla kişiler ve kişi grupları ile ilgili ekleme, silme, güncelleme ve listeleme işlemleri yapabilecektir. Ayrıca uygulamada ad, soyad, telefon numarası ve kişi grubuna göre arama işlemi gerçekleştirilecektir. Uygulama, kullanıcının tercihi nedeniyle internet bağlantısı gerektirmeyen bir form uygulaması olarak geliştirilmiştir.

2. ARAÇLAR VE YÖNTEMLER

Proje, kişi işlemlerini içeren "Telefon Rehberi" ve kişi grupları işlemlerini içeren "Grup İşlemleri" olmak üzere iki sayfadan oluşmuştur.

Proje; ücretsiz bir IDE olan "Visual Studio Community" programı üzerinde, .NET Framework ortamında, C# yazılım dili ile geliştirilmiştir. Veri tabanı yönetim sistemi olarak yine ücretsiz olan MySQL Server kullanılmıştır.

MySQL'de oluşturulan veri tabanı tablolarında ekleme, silme, güncelleme ve listeleme işlemleri yapmak için ADO.NET veriye erişim teknolojisi kullanılmıştır. Bu teknoloji ile belirli sınıflar ve metotlar kullanılarak veri tabanı üzerinde SQL sorguları çalıştırılabilir.



Uygulamada veri tabanı bağlantısı için ayrı bir sınıf oluşturulduğu için bağlantı cümlesi değiştiğinde bunun yalnızca veri tabanı bağlantısı sınıfında güncellenmesi yeterli olur. Bu da geliştirici için kolaylık sağlar. Ayrıca veri tabanı işlemlerinin arayüzden bağımsız olarak ayrı sınıflar içinde metotların içine yazılıp gerekli yerlerde çağırılması, projenin değişimlere cevap vermesi açısından oldukça önemlidir. Projenin arayüzü değiştirilmek istendiğinde veri tabanı ile ilgili kodların değiştirilmesine gerek kalmaz. Veri tabanı işlemlerinin yapıldığı metotlar yeni arayüz içinden çağrılarak da kullanılabilir.

Uygulama çalıştırıldıktan sonra kullanıcı, bilgi girişi bölümünden eklemek istediği kişinin bilgilerini girerek “EKLE” butonu ile rehberine yeni kişi ekleyebilir. Ayrıca alt bölümde listelenen kişilerden birini seçerek “SİL” butonu ile o kaydı silebilir veya seçtiği kişi için yeni bilgileri girerek “GÜNCELLE” butonu ile o kişinin bilgilerini güncelleyebilir. Daha önce bir kişi seçilmişse yeni bir kişi eklemek için “YENİ KAYIT” butonu tıklanmalıdır. Arama bölümündeki form nesnelerine değer girilerek arama işlemi gerçekleştirilebilir.

Kişi grupları ile ilgili ekleme, silme ve güncelleme işlemleri yapılmak istenirse “+” butonuna tıklanır ve kullanıcı, “Grup İşlemleri” sayfasına yönlendirilir (Görsel 1.40).

Görsel 1.40: Kişi grupları ile ilgili ekleme, silme ve güncelleme işlemleri



NOT

Araçlar ve Yöntemler bölümünde, gereksinimler ve tasarım adımıyla oluşturulan diyagramlar da yer alabilir. Tekrarı önlemek için gereksinimler ve diyagramlar bu bölüme eklenmemiştir.

3. SONUÇ

Geliştirilen uygulama test edildiğinde şu işlemleri yerine getirdiği görülür:

- Kişi grupları ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması
- Kişiler ile ilgili ekleme, silme ve güncelleme işlemlerinin yapılması
- Bilgi girişi yapılırken zorunlu alanlar girilmediğinde uyarı mesajı gösterilmesi
- Kişilerin A'dan Z'ye göre sıralanarak listelenmesi
- Kişilerin içinde ad, soyad, telefon ve grup bilgisine göre arama işlemi yapılması

Geliştirilen yazılım ürününün işlevleri test edildiğinde müşteri ihtiyaçlarını karşıladığı görülür. Ürün setup hâline getirilerek dağıtılabilir.



Ürünün eksiklikleri şunlardır:

- Ürünün arayüz tasarımları geliştirilmelidir. Uygulamada arayüzler tasarlanırken form tasarımlarına görsel açıdan zenginlik katmak için geliştirilen eklentiler kullanılabilir.
- Veri tabanı işlemleri için gösterilen hata mesajları kullanıcılar için daha anlamlı hâle getirilebilir.

4. SÖZLÜK

IDE (Integrated Development Environment): Tümleşik geliştirme ortamı olarak Türkçeye çevrilen IDE, geliştiricilerin ihtiyaç duyduğu araçları ve eklentileri içinde barındırarak uygulama geliştirmeyi kolaylaştıran yazılımlardır. Ayrıca diğer geliştirme ortamlarında kullanıcıların manuel yapması gereken birçok işlemi otomatik şekilde yaparak yazılımın geliştirilme süresini de kısaltır.

Visual Studio Community: Konsol uygulamaları, masaüstü uygulamaları, web siteleri, web uygulamaları ve mobil uygulamalar geliştirmek için gerekli olan yazılım geliştirme araçlarını içinde barındıran ücretsiz bir tümleşik geliştirme ortamıdır.

MySQL: Açık kaynak kodlu, ücretsiz bir veri tabanı yönetim sistemi yazılımıdır. En çok kullanılan VTYS'lerden biridir.

.NET Framework: Microsoft tarafından geliştirilen bir uygulama geliştirme platformudur. Geliştirilen uygulamaların çalıştırılacağı ortamlarda da bu yazılımın yüklü olması gerekir.

ADO.NET (ActiveX Data Objects.NET): Microsoft tarafından .NET ortamında veri tabanı işlemlerinin gerçekleştirilmesi için geliştirilen veriye erişim teknolojisidir.

SQL (Structured Query Language): Yapılandırılmış sorgu dili anlamına gelen SQL, veri tabanı işlemleri yapmak için geliştirilmiş bir sorgulama dilidir.

Proje Sunumu Aşamaları

Proje raporu tamamlandıktan sonra projenin paydaşlara anlaşılır bir şekilde sunulması gerekir. Proje sunumunda dikkat edilecek hususlar şunlardır:

- Sunumla ilgili en önemli iş içeriğinin hazırlanmasıdır. Önemli ve ilgi çekici bilgilerin, dinleyici kitlesini gözeterek seçilip düzenlenmesi sunumun başarısını belirler.
- Sunum, dinleyici odaklı olmalıdır.
- Sunumun bir ana mesajı olmalı, tüm içerikler ana mesajı desteklemelidir.
- Eldeki tüm bilgi sunuma aktarılmalıdır.
- Sunum alanı, alanın ışığı, ses düzeni ve düzeyi önceden kontrol edilmelidir.
- Sunum akışı iyi planlanmalıdır.
- Mesajın iletilmesinde slaytların görselliği oldukça etkilidir ancak görsel öğeler yalnızca gerektiğinde kullanılmalıdır.
- Görseller iletilmek istenen mesajla ilgili olmalıdır.
- Mesajı en iyi aktaracak grafikler seçilmelidir.
- Sunum çokça prova edilmelidir.
- Konuşma, dinleyicinin ilgisini canlı tutacak şekilde kurgulanmalıdır.
- Dinleyicilerle sürekli göz teması kurulmalıdır.
- Vücut dilinin ve ses tonunun önemi bilinmelidir.
- Soru ve cevap bölümüne önceden hazırlık yapılmalıdır.



SIRA SİZDE

Personel Takip Programı projesinin sonuç raporunu hazırlayınız ve projenin sunumunu yapınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Projenin giriş bölümünü hazırladı.		
2. Projenin araçlar ve yöntemler bölümünü hazırladı.		
3. Projenin sonuç bölümünü hazırladı.		
4. Projenin sözlük bölümünü hazırladı.		
5. Projenin kaynakça bölümünü hazırladı.		
6. Projenin sunumunu yaptı.		



ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruların doğru seçeneğini işaretleyiniz.

1. Bir yazılım projesi hazırlanırken ilk olarak aşağıdaki işlemlerden hangisi yapılmalıdır?

- A) Fizibilite çalışması
- B) Risk analizi
- C) Varlık ilişki şemaları
- D) Veri akış diyagramı
- E) Veri tabanı tasarımı

2. Aşağıdakilerden hangisi yazılım projesi hazırlama süreci adımlarından biri değildir?

- A) Kodlama
- B) Planlama
- C) Sunum
- D) Tasarım
- E) Test

3. Aşağıdakilerden hangisi etkin kod yazım stillerinden biri değildir?

- A) Gerektiğinde yazılan kodlara açıklama satırı eklenmelidir.
- B) Bütün kodlar aynı sınıf içine yazılmalıdır.
- C) Tekrarlanan işlemler için metot yapıları kullanılmalıdır.
- D) Yapılacak her işlem için en uygun algoritma seçilmelidir.
- E) Yazılan kodlar düzenli olmalıdır.

4. Yazılım projesi hazırlanırken varlık ilişki şemaları aşağıdaki hangi aşamada oluşturulur?

- A) Bakım
- B) Kodlama
- C) Planlama
- D) Tasarım
- E) Test

5. Yazılım projesi aşağıdaki hangi aşamadan sonra yayımlanmalıdır?

- A) Analiz
- B) Kodlama
- C) Planlama
- D) Tasarım
- E) Test

6. Aşağıdakilerden hangisi yazılım geliştirme süreç modellerinden biri değildir?

- A) Kodla ve Düzelt (Code and Fix) Modeli
- B) Ortak Model
- C) Şelale (Waterfall) Modeli
- D) V Modeli
- E) Yinelemeli (Iterative) Model

7. Aşağıdakilerden hangisi projenin sunumunda dikkat edilecek hususlardan değildir?

- A) Dinleyicilerle sürekli göz teması kurulmalıdır.
- B) Proje ile ilgili bilgiler ekrandan veya eldeki notlardan okunmalıdır.
- C) Soru ve cevap bölümüne önceden hazırlık yapılmalıdır.
- D) Sunum akışı iyi planlanmalıdır.
- E) Sunum esnasında vücut dili kullanılmalıdır.



2. ÖĞRENME BİRİMİ

NESNE TABANLI PROGRAMLAMA PROJESİ HAZIRLAMA



KONULAR

- 2.1. STOK TAKİP PROGRAMININ PLANLAMA AŞAMASI
- 2.2. STOK TAKİP PROGRAMININ TASARIM AŞAMASI
- 2.3. STOK TAKİP PROGRAMININ KODLAMA AŞAMASI
- 2.4. STOK TAKİP PROGRAMININ TEST AŞAMASI
- 2.5. STOK TAKİP PROGRAMININ BAKIM AŞAMASI
- 2.6. STOK TAKİP PROGRAMININ RAPOR AŞAMASI

NELER ÖĞRENECEKSİNİZ?

- Nesne tabanlı programlama projesini planlama
- Projenin gereksinim analizini gerçekleştirme
- Projenin risk analizini gerçekleştirme
- Projenin tasarımını yapma
- Projenin kodlamasını yapma
- Projenin test ve bakımını yapma
- Proje raporunu hazırlama

TEMEL KAVRAMLAR

Bakım, database first, entity framework, gereksinim analizi, kodlama, nesne tabanlı programlama projesi, ORM, planlama, rapor, risk yönetimi, tasarım, test, veri tabanı

HAZIRLIK ÇALIŞMALARI

1. Masaüstü program yazılımlarının kullanım alanlarına örnekler veriniz.
2. Yüksek seviyeli programlama dilleri dendiğinde aklınıza ne gelmektedir? Belirtiniz.
3. Günümüzde kullanılan veri tabanı sistemlerini araştırınız. Ulaşılan sonuçları arkadaşlarınızla tartışınız.



2.1. STOK TAKİP PROGRAMININ PLANLAMA AŞAMASI

Oluşturulacak stok takip programı planlama aşaması; müşterinin temel ihtiyaçlarının tespit edilmesi, projenin sınırlarının belirlenmesi, proje uygulama takviminin oluşturulması, risk yönetimi ve analizinin yapılmasını kapsar.

2.1.1. Müşterinin Temel İhtiyaçları

Bilişim teknolojileri malzemeleri satışı yapan bir firma için ürünlerin satışlarının ve stok takibinin yapılacağı bir uygulamaya ihtiyaç vardır. Ürünlerin hangi firmadan tedarik edildiği, hangi müşteriye satıldığı ve işlemin hangi personel tarafından ne zaman yapıldığı gibi bilgiler saklanmalıdır. Personelin programı kullanabilmesi için kullanıcı adı ve şifresiyle giriş yapması gerekir.

2.1.2. Projenin Sınırlarının Belirlenmesi

- Proje; personeller ile ilgili ekleme, silme, güncelleme ve listeleme gibi işlemleri içerecektir. İşlem yapacak personel, kullanıcı adı ve şifresi ile sisteme giriş yapmalıdır.
- Sistemde ürünlerle ilgili ekleme, silme, güncelleme, listeleme ve arama işlemleri yapılacaktır.
- Sistemde tedarikçiler ve müşteriler kayıtlı olacaktır. Bunlarla ilgili ekleme, silme, güncelleme ve listeleme işlemleri yapılacaktır.
- Sistemde ürün satışı ve satışı iptal etme işlemleri olacak, ürünlerin stok takibi yapılacaktır.

2.1.3. Proje Uygulama Takviminin Oluşturulması

Proje uygulama takvimi Tablo 2.1’de verilmiştir.

Tablo 2.1: Stok Takip Uygulaması Proje Uygulama Takvimi

Zaman	Proje Bölümü	Yapacak Personel	Süre
1. Hafta	Risk yönetimi ve analizinin yapılması	Öğrenci1, Öğrenci2	1 hafta
	Gereksinim analizinin yapılması		
2, 3 ve 4. Hafta	Projenin tasarımının yapılması	Öğrenci1, Öğrenci2 Öğrenci3	3 hafta
5, 6 ve 7. Hafta	Projenin kodlanması	Öğrenci3, Öğrenci4	3 hafta
8. Hafta	Projenin test edilmesi ve yayımlanması	Öğrenci1, Öğrenci4	1 hafta

2.1.4. Risk Yönetimi ve Analizinin Yapılması

Risk yönetimi ve analizinin yapılması Tablo 2.2’de verilmiştir.

Tablo 2.2: Stok Takip Uygulaması Risk Yönetimi

No	Risk	Gerçekleşme Olasılığı	Projeye Etkisi	Alınacak Önlemler
1	Müşterinin ve son kullanıcıların deneyimsiz olması, ihtiyaçlarını net olarak ifade edememesi	Orta	Yüksek	Ürünle ilgili yapılacak proje toplantılarına alanında uzman kişilerin katılımı sağlanarak görüşlerinin alınması, müşteri ve kullanıcıya projenin detaylarını ortaya çıkaracak sorular sorulması
2	Müşteri gereksinimlerinin değişimi	Orta	Orta	Gereksinimlerin değişiminin projeye etkisinin belirlenmesi, gerekirse zaman ve maliyetin güncellenmesi
3	Veri tabanı ve programdaki kullanıcı şifrelerinin güvenliğinin sağlanamaması	Orta	Yüksek	Veri tabanı ve kullanıcı parolalarının güçlü olmasının sağlanması, kullanıcı şifreleri veri tabanına kaydedilirken şifreleme algoritmasının kullanılması



2.2. STOK TAKİP PROGRAMININ TASARIM AŞAMASI

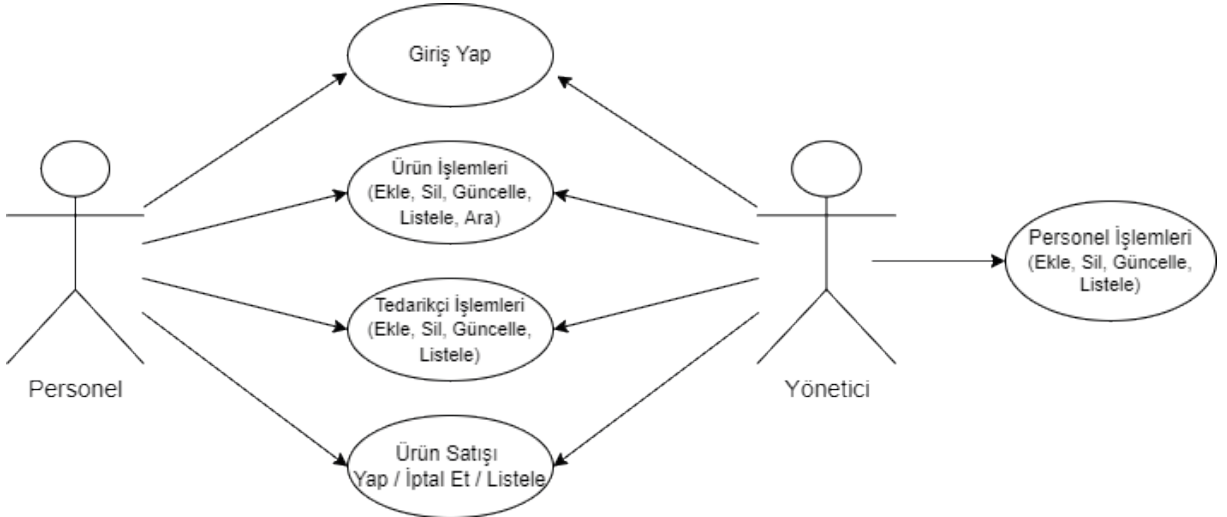
Stok takip programı tasarım aşaması; gereksinim analizi, varlık ilişki şemalarının oluşturulması, veri akış diyagramlarının hazırlanması, kullanıcı arayüzleri tasarımı, veri tabanı tasarımı, kullanılacak teknik ve yöntemlerin belirlenmesini kapsayan bir aşamadır.

2.2.1. Gereksinim Analizi

Müşteri gereksinimleri şunlardır:

- Ürünün internet bağlantısı gerektirmeden çalışan bir masaüstü uygulaması (form uygulaması) olmalıdır.
- Programda iki tür kullanıcı olacaktır. Personel kullanıcısı; tedarikçiler, müşteriler, ürünler ve ürün satışı ile ilgili bütün işlemleri yapabilirken, yönetici kullanıcısı buna ek olarak personellerle ilgili kayıt işlemleri, kullanıcı adı ve şifre işlemlerini de yapabilmelidir.
- Sistemde personel, ürün, tedarikçiler, satışlar, müşterilerle ilgili bilgiler tutulmalı ve bunlarla ilgili ekleme, silme, güncelleme, listeleme işlemleri yapılabilmelidir.
- Ürünlerin stok takibi yapılmalıdır. Ürün stok bilgisi güncellenebilmeli, satış yapıldığında ürünün stok adedinden satılan miktar kadar düşülmelidir.
- Ürünlerin benzersiz bir ürün kodu olmalı ve ürünün adı, kategorisi, stok adedi, satış fiyatı gibi ürünün detaylarını içeren bilgileri saklanmalıdır.
- Ürünlerin satış bilgileri, işlemi yapan personel ve işlem tarihi ile birlikte sistemde saklanmalı ve gerektiğinde bu bilgilere ulaşılabilmelidir.
- Satışlar liste fiyatı üzerinden yapılmalı, satış esnasında fiyat değişikliği yapılmamalıdır.
- Gerektiğinde yapılan bir satış iptal edilebilmelidir. İptal edilen satışlar sistemden silinmelidir.

Kullanım Durumu (Use Case) Diyagramları: Stok takip projesinde yer alan aktörlerin yapacakları işlemlerin belirlenmesi için kullanım durumu diyagramları oluşturulur (Görsel 2.1).



Görsel 2.1: Stok Takip Uygulaması projesine ait kullanım durumu diyagramları

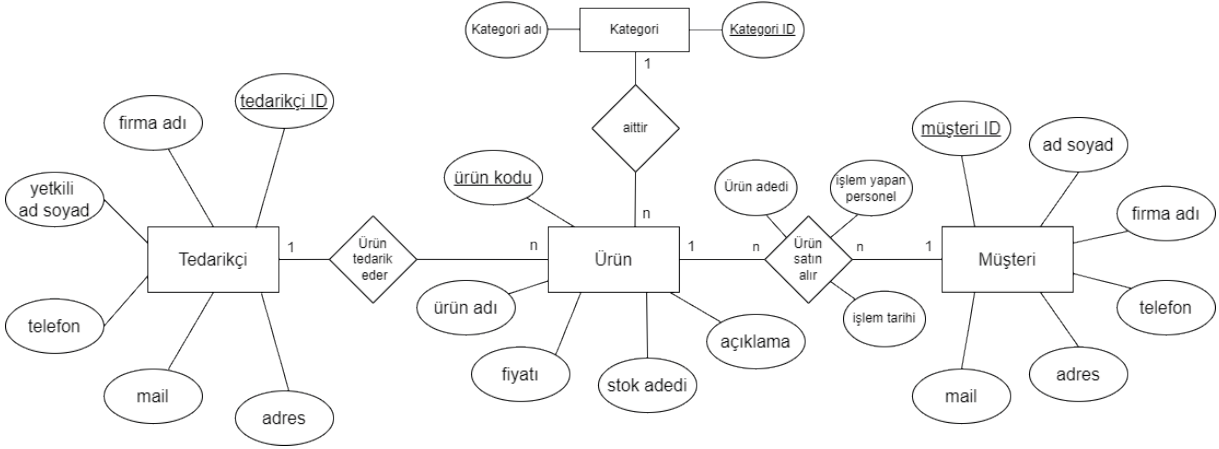
Sistem gereksinimleri şunlardır:

- Windows 7 ve üzeri işletim sistemi bilgisayarda yüklü olmalıdır.
- .NET Framework 4.7 veya üzeri yüklü olmalıdır.
- MySQL veri tabanı yönetim sistemi yüklü olmalıdır.
- MySQL Connector/.NET eklentisi yüklü olmalıdır.



2.2.2. Varlık İlişki Şemalarının Oluşturulması

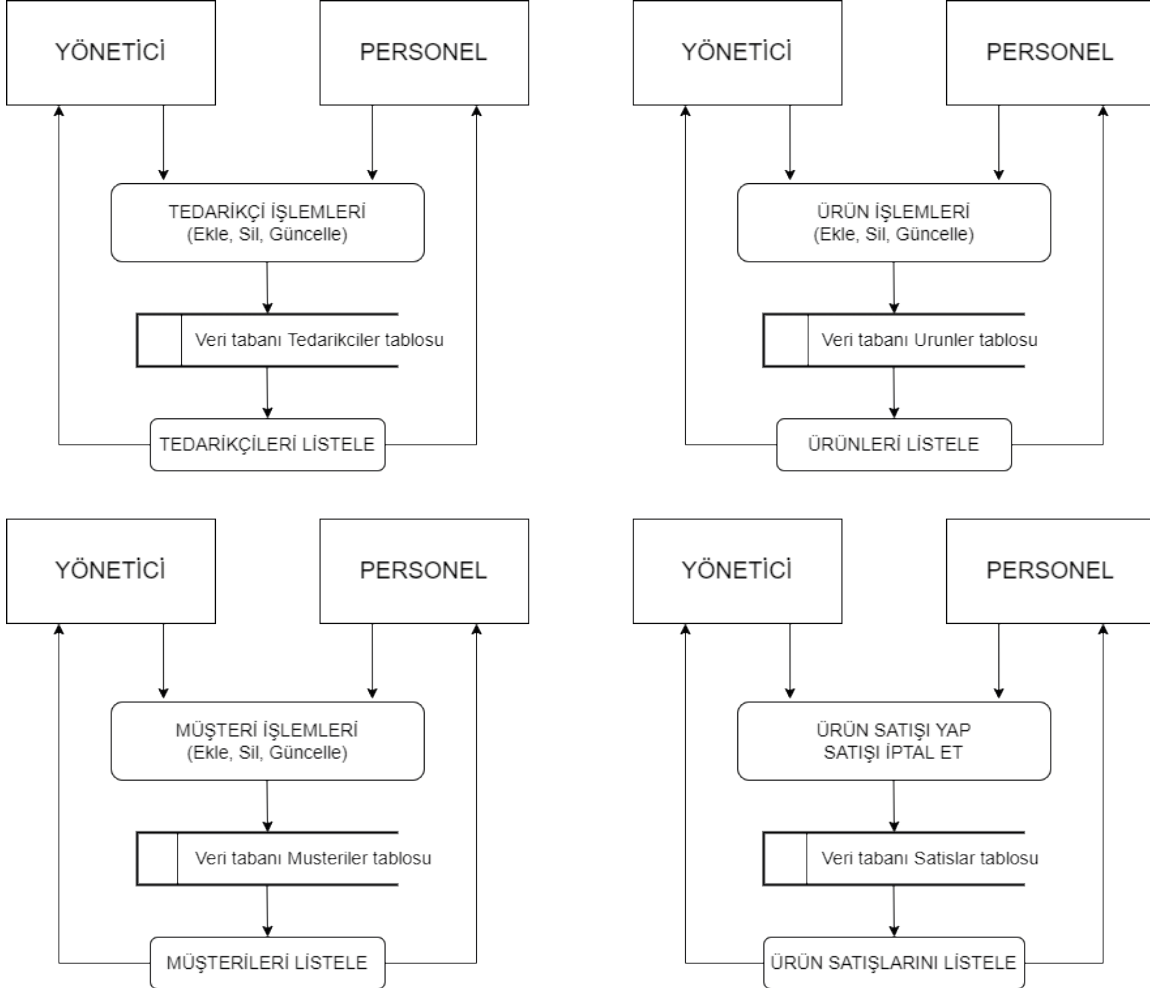
Stok Takip Uygulaması projesinde bulunan varlıkların özellikleri ve varlıklar arasındaki ilişkileri göstermek için varlık ilişki şeması oluşturulur (Görsel 2.2).



Görsel 2.2: Stok Takip Uygulaması projesine ait kullanım durumu diyagramları

2.2.3. Veri Akış Diyagramlarının Hazırlanması

Stok Takip Uygulaması projesindeki veri akışını göstermek için veri akış diyagramları oluşturulur (Görsel 2.3).



Görsel 2.3: Stok Takip Uygulaması projesine ait veri akış diyagramı



2.2.4. Kullanıcı Arayüzleri Tasarımı

Proje anasayfasının üst bölümünde diğer sayfalara yönlendirme işlemi yapacak menüler vardır. Sol bölümde ürün listesi, sağ bölümde ise seçilen ürünlerin eklenebileceği satılacak ürünler listesi ve ürünün satılacağı müşterinin seçileceği bölüm bulunur (Görsel 2.4). Ürün satış işlemi bu sayfada gerçekleştirilecektir.

MENÜLER			
ÜRÜN LİSTESİ			Müşteri Seçimi
Sütun 1	Sütun 2	Sütun 3	
Hücre 1	Hücre 2	Hücre 3	
Hücre 4	Hücre 5	Hücre 6	EKLE
Hücre 7	Hücre 8	Hücre 9	SİL
Hücre 10	Hücre 11	Hücre 12	
			SATIŞI YAPILACAK ÜRÜNLER
Sütun 1	Sütun 2	Sütun 3	
Hücre 1	Hücre 2	Hücre 3	
Hücre 4	Hücre 5	Hücre 6	
Hücre 7	Hücre 8	Hücre 9	
Hücre 10	Hücre 11	Hücre 12	
			SATIŞ YAP
			GENEL TOPLAM

Görsel 2.4: Anasayfa form ekranı tasarımı

Satış Detayları sayfasında sol bölümde satışlar listelenir. Listelenen satışların içinden bir kayıt seçildiğinde o satışa ait ürünler sağ tarafta görüntülenir. Ayrıca bu sayfada seçilen satışı iptal etmek için “SATIŞI İPTAL ET” butonu bulunur (Görsel 2.5).

MENÜLER			
SATIŞ LİSTESİ			
Sütun 1	Sütun 2	Sütun 3	
Hücre 1	Hücre 2	Hücre 3	
Hücre 4	Hücre 5	Hücre 6	
Hücre 7	Hücre 8	Hücre 9	
Hücre 10	Hücre 11	Hücre 12	
			SATILAN ÜRÜNLER
Sütun 1	Sütun 2	Sütun 3	
Hücre 1	Hücre 2	Hücre 3	
Hücre 4	Hücre 5	Hücre 6	
Hücre 7	Hücre 8	Hücre 9	
Hücre 10	Hücre 11	Hücre 12	
			SATIŞI İPTAL ET
			SATIŞ TOPLAM FİYAT

Görsel 2.5: Satış Detayları form ekranının tasarımı



Kategori İşlemleri, Tedarikçi İşlemleri, Müşteri İşlemleri, Personel İşlemleri ve Ürün İşlemleri sayfaları benzer yapıda oluşturulur. Bu sayfaların üst bölümünde gerekli bilgilerin girileceği bilgi girişi alanı ve butonlar, alt bölümünde ise verilerin listeleneceği bir “DataGridView” nesnesi bulunur (Görsel 2.6).

BİLGİ GİRİŞİ

Bilgi girişi için gerekli form elemanları (TextBox, ComboBox, NumericUpDown, MaskedTextBox ..vb) bu bölümde yer alır

YENİ KAYITEKLESİLGÜNCELLE

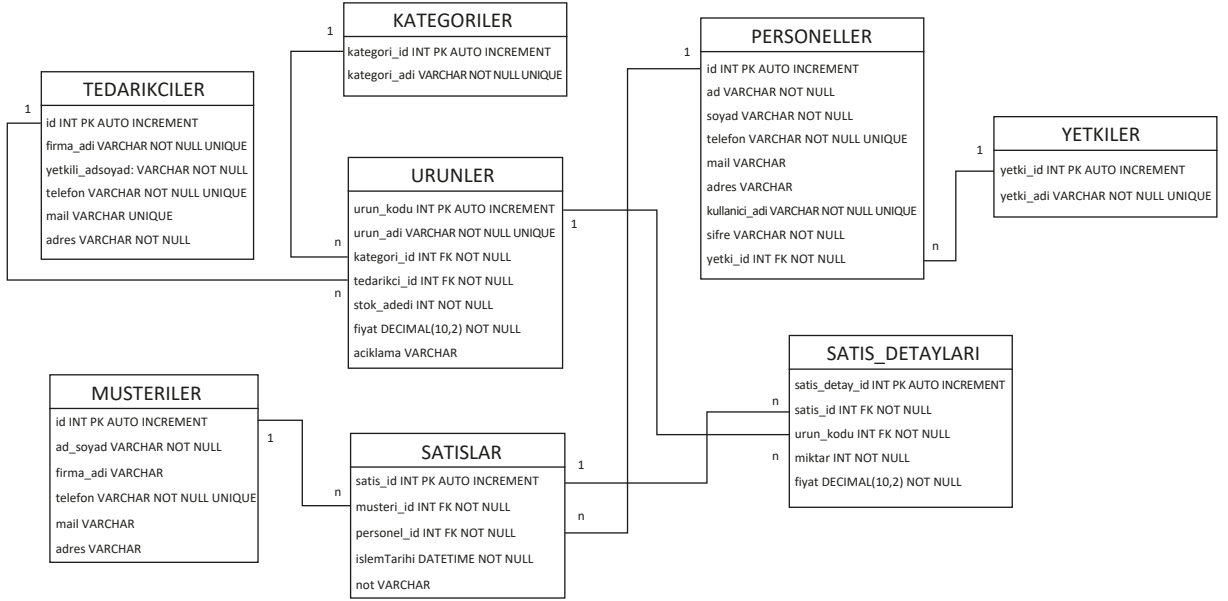
Sütun 1	Sütun 2	Sütun 3
Hücre 1	Hücre 2	Hücre 3
Hücre 4	Hücre 5	Hücre 6
Hücre 7	Hücre 8	Hücre 9
Hücre 10	Hücre 11	Hücre 12

Görsel 2.6: Diğer sayfaların kullanıcı arayüzü tasarımları

2.2.5. Veri Tabanı Tasarımı

Uygulamada kaydedilmesi gereken bilgilere ait özellikler analiz edilerek bunlara uygun veri türleri belirlenir. Ayrıca benzersiz olan değerler, boş bırakılamayan alanlar, yalnızca bir kayda ait olabilecek bilgiler belirlenerek bir veri tabanı tasarımı oluşturulur (Görsel 2.7). Kodlama aşamasında bu tasarıma göre veri tabanı ve tablolar hazırlanır.

- Uygulamada aynı isimle birden fazla ürün kaydı yapılması engellenmelidir.
- Uygulamada aynı isimle birden fazla kategori kaydı yapılması engellenmelidir.
- Tedarikçilerin firma adı, telefon ve mail bilgileri benzersiz olmalıdır.
- Bir telefon numarası yalnızca bir müşteriye ait olabilir.
- Personellerin kullanıcı adı ve telefon numarası değerleri benzersiz olmalıdır.



Görsel 2.7: Stok Takip Uygulaması projesi veri tabanı tasarımı

2.2.6. Kullanılacak Teknik ve Yöntemlerin Belirlenmesi

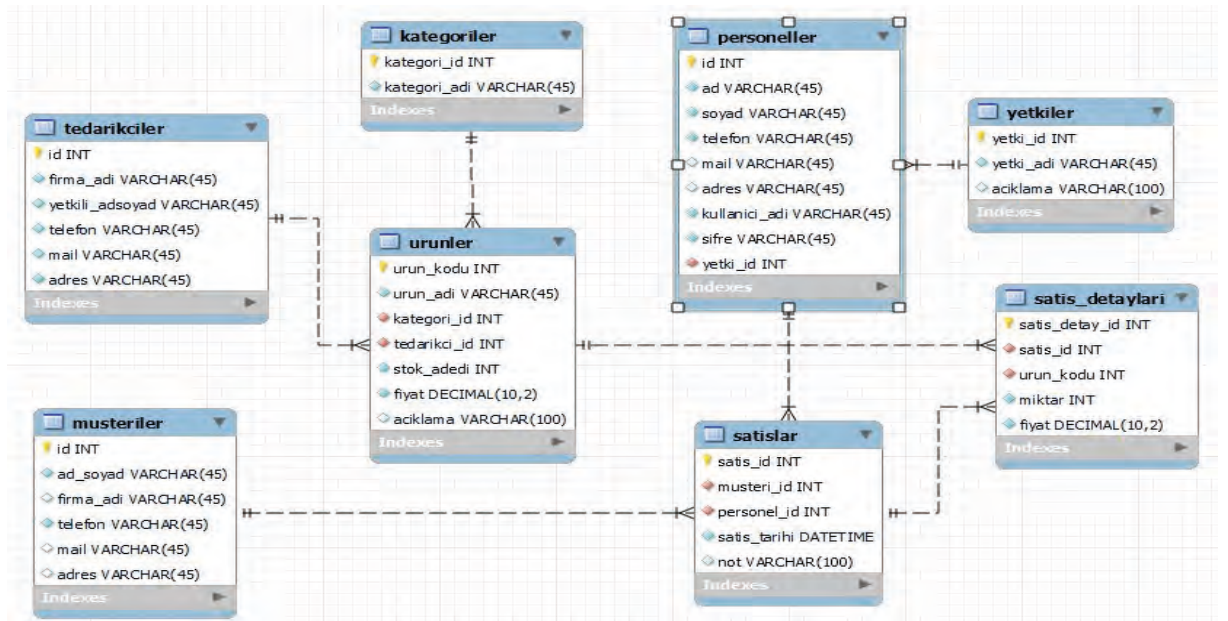
Proje, Visual Studio Community programında C# programlama dili ile geliştirilir. VTYS olarak MySQL kullanılır. Ayrıca veri tabanı işlemleri için Entity Framework Database First ORM aracı kullanılır.

2.3. STOK TAKİP PROGRAMI KODLAMA AŞAMASI

Kodlama aşamasında isimlendirme kurallarına uymak kaydıyla proje, sayfa ve nesne isimleri değiştirilebilir. Nesnelerin form içindeki yerleşimi için "Dock" özellikleri kullanılabilir.

2.3.1. Veri Tabanının Oluşturulması

Kodlama aşamasında ilk olarak "stok_takip_db" isminde bir veri tabanı, veri tabanı tabloları ve tablolar arasındaki ilişkiler oluşturulur (Görsel 2.8).



Görsel 2.8: Stok Takip Uygulaması projesine ait veri tabanı tasarımı



NOT

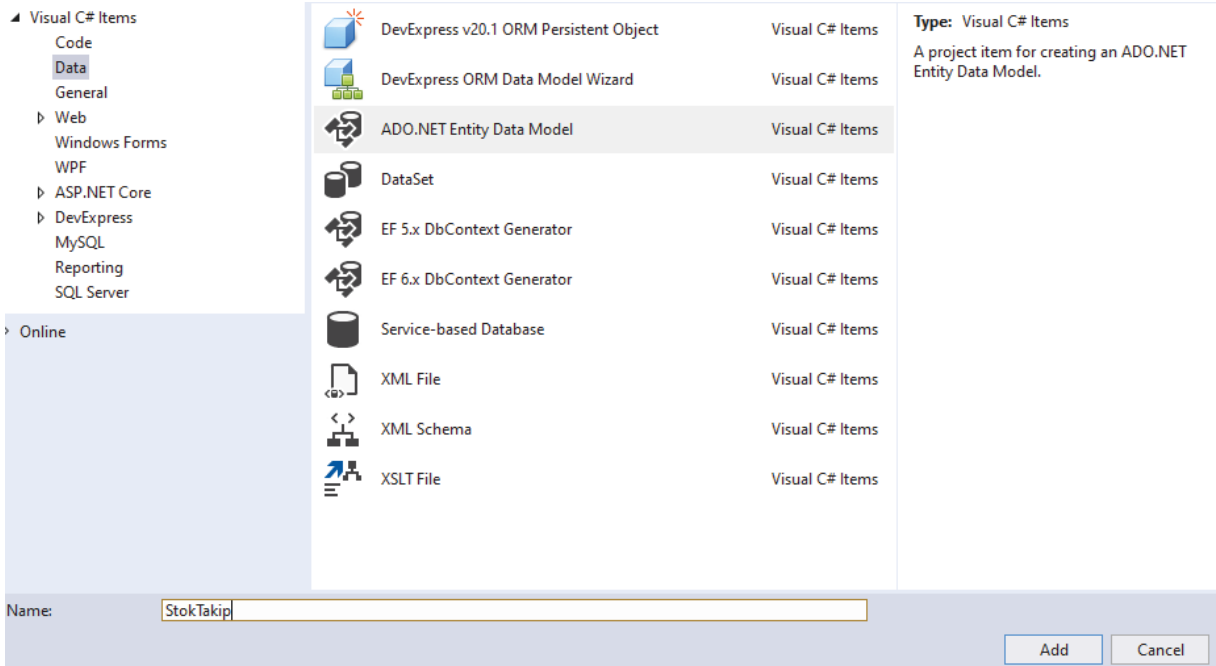
Projedeki formlar oluşturulurken veri tabanı tablolarında bulunan alanların özelliklerine göre sayfalardaki form elemanlarına bilgi girişi kısıtlanmalı, programın hata vermesi engellenmelidir. Örneğin DECIMAL veri tipinde olan fiyat alanına metinsel ifade girişi engellenmelidir veya maksimum karakter değeri “45” olan bir alan için oluşturulan metin kutusunun maksimum karakter değeri 45 ile sınırlandırılmalıdır.

2.3.2. Form Uygulamasının Oluşturulması ve Veri Tabanı Bağlantısının Kurulması

Kod editörü üzerinde yeni bir form uygulaması (.NET Framework) oluşturulur ve “StokTakipUygulaması” olarak isimlendirilir.

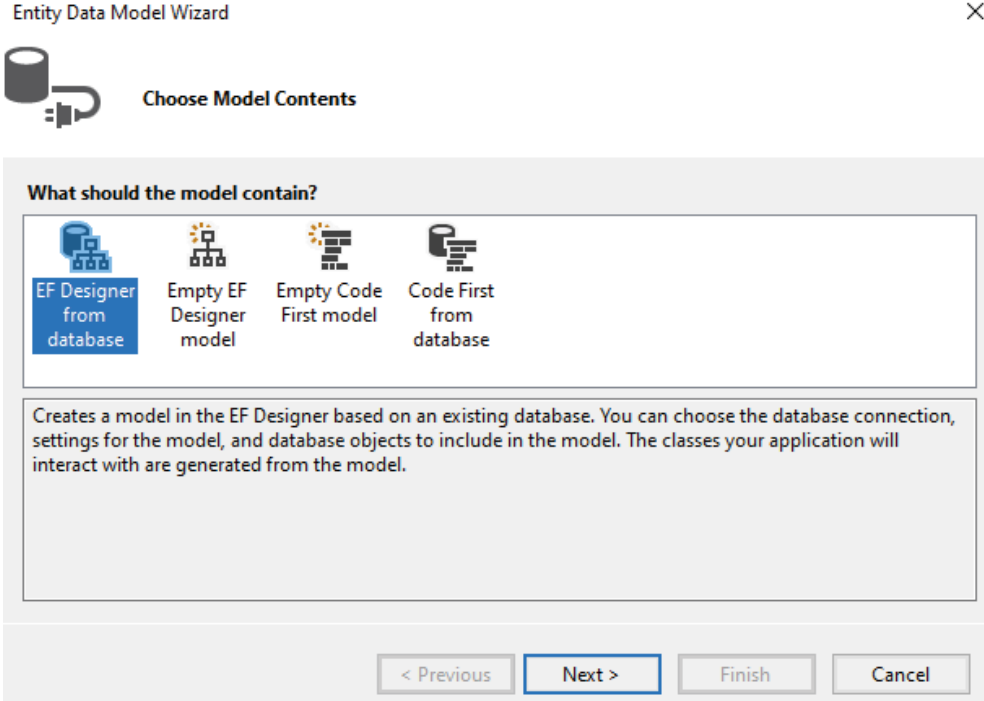
Entity Framework ile veri tabanı bağlantısı yapabilmek için Solution Explorer penceresinde proje üzerine sağ tıklanarak “Manage NuGet Packages” seçeneği seçilir. Burada bulunan paketler içinden **MySql.Data.EntityFramework** yüklenir.

Daha sonra proje üzerine sağ tıklanarak Add > New Item seçeneği seçilir. Gelen pencerede **ADO.NET Entity Data Model** seçilir. “StokTakip” adı verilerek Add butonuna tıklanır (Görsel 2.9).



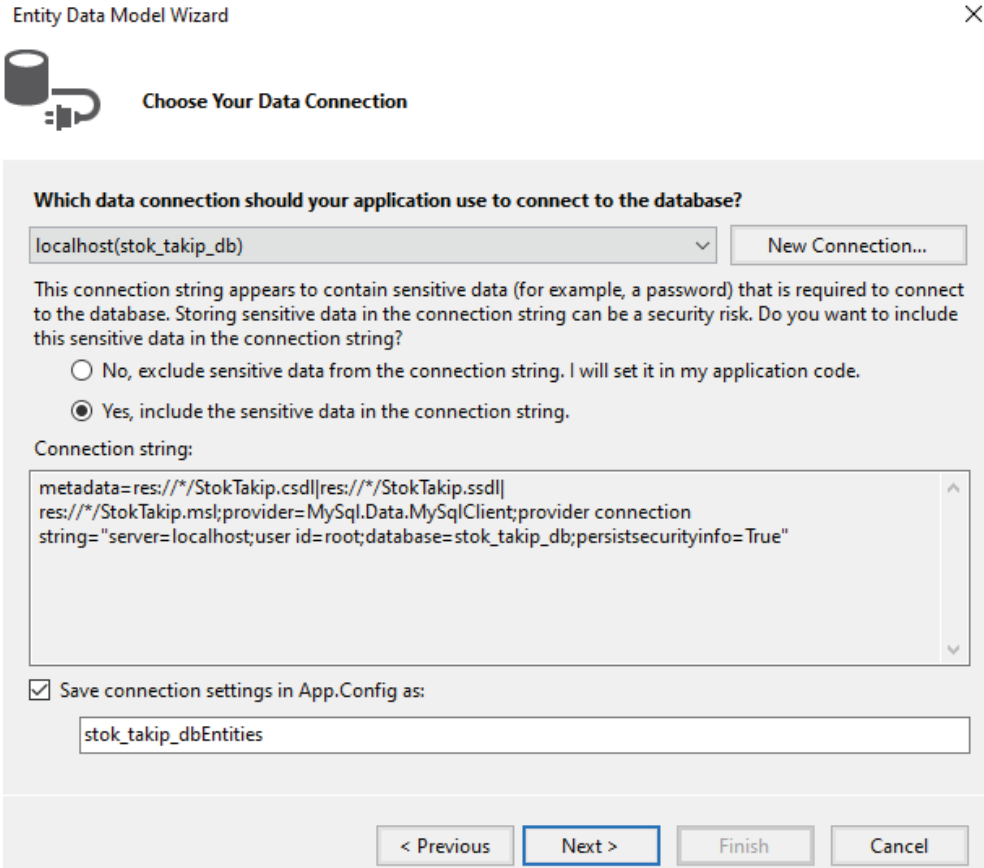
Görsel 2.9: Projeye ADO.NET Entity Data Model eklenmesi

Gelen pencereden “EF Designer from database” seçilerek Next tuşu ile işleme devam edilir (Görsel 2.10). “EF Designer from database” seçeneğinin seçilmesi ile Database First yaklaşımının kullanılacağı belirlenir.



Görsel 2.10: Kullanılacak Entity Framework yaklaşımının seçilmesi

Gelen pencereden bağlantı ve kullanılacak veri tabanı seçilerek oluşturulan bağlantının, belirtilen isimle “App.config” dosyasında saklanacağı Görsel 2.11’de görülür.



Görsel 2.11: Bağlantı seçim ekranı



Kullanılacak tablolar ve model ismi belirlenerek “ADO.NET Entity Data Model” ekleme işlemi bitirilir (Görsel 2.12).

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables
☐ Views
☐ Stored Procedures and Functions

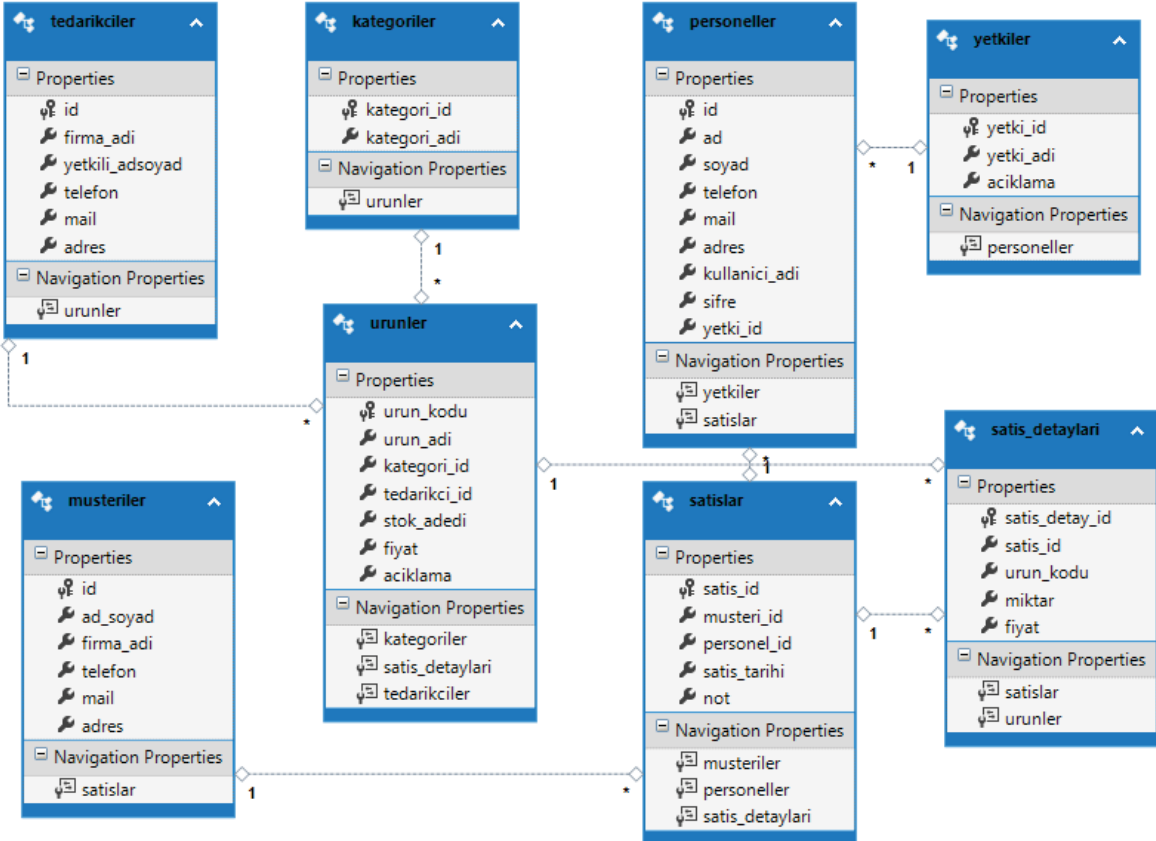
☐ Pluralize or singularize generated object names
☒ Include foreign key columns in the model
☐ Import selected stored procedures and functions into the entity model

Model Namespace:
stok_takip_dbModel

< Previous Next > Finish Cancel

Görsel 2.12: Veri tabanı nesneleri seçim ekranı

İşlem bittikten sonra model ve sınıfların oluşturulması için bir süre beklenir. Oluşan model, “StokTakipModel.edmx” dosyasında diyagram şeklinde görüntülenir (Görsel 2.13).



Görsel 2.13: Oluşturulan modelin görüntülenmesi



ADO.NET Entity Data Model oluşturulduktan sonra "Solution Explorer" penceresinde veri tabanı tablolarına karşılık gelen sınıflar ve bu sınıfların veri tabanı işlemlerinde kullanılmasını sağlayacak "stok_takip_dbEntities" sınıfının otomatik olarak oluşturulduğu görülür (Görsel 2.14).

DbContext sınıfından türetilen stok_takip_dbEntities sınıfı, veri tabanı işlemlerinin kolaylıkla yapılmasını sağlar.

Proje formlarından ilk olarak ürün ve satışlar için kullanılacak yardımcı sayfalar oluşturulur. Bir ürünü eklemek için o ürünün ait olduğu kategori ve ürünün temin edildiği tedarikçi eklenmiş olmalıdır veya ürünün satışı yapılabilmesi için satışı yapacak personel ve ürünün satılacağı müşterinin kayıtlı olması gerekir.

```
public partial class stok_takip_dbEntities : DbContext
{
    // references
    public stok_takip_dbEntities()
        : base("name=stok_takip_dbEntities")
    {
    }
    // references
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }
    // references
    public virtual DbSet<kategori> kategoriler { get; set; }
    // references
    public virtual DbSet<musteriler> musteriler { get; set; }
    // references
    public virtual DbSet<personeller> personeller { get; set; }
    // references
    public virtual DbSet<satıs_detaylari> satıs_detaylari { get; set; }
    // references
    public virtual DbSet<satıslar> satıslar { get; set; }
    // references
    public virtual DbSet<tedarikciler> tedarikciler { get; set; }
    // references
    public virtual DbSet<urunler> urunler { get; set; }
    // references
    public virtual DbSet<yetkiler> yetkiler { get; set; }
}
```

Görsel 2.14: stok_takip_dbEntities sınıfı



NOT

Form sayfalarına eklenen "DataGridView" nesnelerinde seçim esnasında satırın tamamının seçilmesi sağlanmalı (SelectionMode: FullRowSelect) ve çoklu seçim işlemine izin verilmemelidir (MultiSelect: False). Ayrıca AutoSizeColumnsMode özelliği "Fill" şeklinde belirlenerek sütunların DataGridView nesnesini doldurması sağlanır.

2.3.3. Kategori İşlemleri

Projede ürünün ait olduğu kategoriler ile ilgili işlemleri yapmak için "frmKategoriİşlemleri" isimli yeni bir form oluşturulur (Görsel 2.15). Formdaki nesneler; **txtKategoriAdi**, **btnYeni**, **btnEkle**, **btnSil**, **btnGuncelle**, **grdKategoriler** şeklinde isimlendirilir. Daha sonra form elemanlarının özellikleri uygun şekilde belirlenir. DataGridView nesnesi üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.

Görsel 2.15: Kategori işlemleri formunun tasarımı



frmKategoriIslemleri sayfasının arka planına ekleme, silme, güncelleme ve listeleme işlemleri için gerekli kodlar yazılır. Sayfanın yüklenmesi olayında kategoriler listelenir. Silme ve güncelleme işlemleri için “DataGridView” nesnesinde listelenen kategorilerden seçim yapılması sağlanır (Görsel 2.16).

```
stok_takip_dbEntities stok_Takip;
int seciliKategoriId = 0;
1 reference
private void frmKategoriIslemleri_Load(object sender, EventArgs e)
{
    BilgileriListele();
}
4 references
private void BilgileriListele()
{
    stok_Takip = new stok_takip_dbEntities();
    grdKategoriler.DataSource = stok_Takip.kategoriler.ToList();
    grdKategoriler.Columns.Remove("urunler");
    grdKategoriler.Columns["kategori_id"].Width = 65;
    grdKategoriler.ClearSelection();
}
1 reference
private void btnYeniKayit_Click(object sender, EventArgs e)
{
    seciliKategoriId = 0;
    txtKategoriAdi.Clear();
}
1 reference
private void grdKategoriler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdKategoriler.CurrentRow != null)
        {
            seciliKategoriId = Convert.ToInt32(grdKategoriler.CurrentRow.Cells["kategori_id"].Value.ToString());
            txtKategoriAdi.Text = grdKategoriler.CurrentRow.Cells["kategori_adi"].Value.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluşturdu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.16: Kategorilerin listelenmesi ve listelenen kayıtlardan seçim yapılması

Adı girilen kategorinin kaydedilmesi için “EKLE” butonunun tıklanma olayına Görsel 2.17’deki kodlar yazılır.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliKategoriId == 0)
        {
            if (!string.IsNullOrEmpty(txtKategoriAdi.Text))
            {
                stok_Takip = new stok_takip_dbEntities();
                kategoriler kategori = new kategoriler();
                kategori.kategori_adi = txtKategoriAdi.Text;
                stok_Takip.kategoriler.Add(kategori);
                stok_Takip.SaveChanges();
                txtKategoriAdi.Clear();
                BilgileriListele();
            }
        }
    }
}
```



```

        else
        {
            MessageBox.Show("Kategori adını giriniz.");
        }
    }
    else
    {
        MessageBox.Show("Yeni kategori eklemek için 'Yeni Kayıt' butonuna tıklayınız");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Kayıt Hatası");
}
}

```

Görsel 2.17: Kategori ekleme kodları

DataGridView üzerinde seçilen kategorinin silinmesi için “SİL” butonunun tıklanma olayına Görsel 2.18’deki kodlar yazılır.

```

private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliKategoriId != 0)
        {
            var kategoriKontrol = stok_Takip.urunler.
                Where(u => u.kategori_id.ToString().Contains(seciliKategoriId.ToString())).ToList();
            if (kategoriKontrol.Count() > 0)
            {
                MessageBox.Show("Bu kayıt silinemez. Silmek istediğiniz kategoriye ait ürünler mevcut.");
            }
            else
            {
                stok_Takip = new stok_takip_dbEntities();
                kategoriler silinecekKategori = stok_Takip.kategoriler.
                    Where(k => k.kategori_id == seciliKategoriId).First();
                stok_Takip.kategoriler.Remove(silinecekKategori);
                stok_Takip.SaveChanges();
                BilgileriListele();
                txtKategoriAdi.Clear();
            }
        }
        else
        {
            MessageBox.Show("Silinecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme İşleminde Hata");
    }
}

```

Görsel 2.18: Kategori silme kodları



DataGridView üzerinde seçilen kategorinin güncellenmesi için “GÜNCELLE” butonunun tıklanma olayına Görsel 2.19’deki kodlar yazılır.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliKategoriId != 0)
        {
            if (!string.IsNullOrEmpty(txtKategoriAdi.Text))
            {
                stok_Takip = new stok_takip_dbEntities();
                kategoriler guncellenecekKategori = stok_Takip.kategoriler.
                    Where(k => k.kategori_id == seciliKategoriId).First();
                guncellenecekKategori.kategori_adi = txtKategoriAdi.Text;
                stok_Takip.SaveChanges();
                BilgileriListele();
                txtKategoriAdi.Clear();
            }
            else
            {
                MessageBox.Show("Kategori adını giriniz.");
            }
        }
        else
        {
            MessageBox.Show("Güncellenecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Güncelleme Hatası");
    }
}
```

Görsel 2.19: Kategori güncelleme kodları

2.3.4. Tedarikçi İşlemleri

Projede “frmTedarikciIslemleri” isimli yeni bir form oluşturulur (Görsel 2.20). Formdaki nesneler; **txtFirmaAdi**, **txtYetkiliAdi**, **txtAdres**, **mskTelefon**, **txtMail**, **btnYeni**, **btnEkle**, **btnSil**, **btnGuncelle**, **grdTedarikciler** şeklinde isimlendirilir. Form elemanlarının özellikleri uygun şekilde belirlenir. DataGridView nesnesi üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.



Görsel 2.20: Tedarikçi işlemleri formunun tasarımı

frmTedarikciIslemleri sayfasının arka planına ekleme, silme, güncelleme ve listeleme işlemleri için gerekli kodlar yazılır. Sayfanın yüklenmesi olayında tedarikçiler listelenir. Ayrıca silme ve güncelleme işlemleri için DataGridView üzerinde seçim yapılması sağlanır (Görsel 2.21).

```

stok_takip_dbEntities stok_Takip;
int seciliTedarikciID = 0;
1 reference
private void frmTedarikciIslemleri_Load(object sender, EventArgs e)
{
    TedarikcileriListele();
}
4 references
private void TedarikcileriListele()
{
    stok_Takip = new stok_takip_dbEntities();
    grdTedarikciler.DataSource = stok_Takip.tedarikciler.OrderBy(t=>t.firma_adi).ToList();
    grdTedarikciler.Columns.Remove("urunler");
    grdTedarikciler.Columns["id"].Width = 35;
    grdTedarikciler.ClearSelection();
}
1 reference
private void grdTedarikciler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdTedarikciler.CurrentRow != null)
        {
            seciliTedarikciID = Convert.ToInt32(grdTedarikciler.CurrentRow.Cells["id"].Value.ToString());
            txtFirmaAdi.Text = grdTedarikciler.CurrentRow.Cells["firma_adi"].Value.ToString();
            txtYetkiliAdi.Text = grdTedarikciler.CurrentRow.Cells["yetkili_adsoyad"].Value.ToString();
            txtMail.Text = grdTedarikciler.CurrentRow.Cells["mail"].Value.ToString();
            txtAdres.Text = grdTedarikciler.CurrentRow.Cells["adres"].Value.ToString();
            mskTelefon.Text = grdTedarikciler.CurrentRow.Cells["telefon"].Value.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Görsel 2.21: Tedarikçilerin listelenmesi ve listelenen kayıtlardan seçim yapılması



Ekleme ve güncelleme işlemleri için “BilgiGirisKontrol” isimli bir metod oluşturularak kullanıcıdan alınan bilgilerin doğruluğu ve geçerliliği kontrol edilir (Görsel 2.22).

```
private bool BilgiGirisKontrol()
{
    bool kontrol = false;
    if (txtFirmaAdi.TextLength < 2)
    {
        MessageBox.Show("Firma adını giriniz.");
    }
    else if (txtYetkiliAdi.TextLength < 2)
    {
        MessageBox.Show("Yetkili adını giriniz.");
    }
    else if (!mskTelefon.MaskCompleted)
    {
        MessageBox.Show("Telefon numarasını 10 haneli olarak giriniz.");
    }
    else if (txtMail.TextLength < 5)
    {
        MessageBox.Show("Mail adresini giriniz.");
    }
    else if (txtAdres.TextLength <= 2)
    {
        MessageBox.Show("Firma adresini giriniz.");
    }
    else
    {
        kontrol = true;
    }
    return kontrol;
}
```

Görsel 2.22: Tedarikçi bilgilerinin doğruluk kontrolü

Gerekli kontroller yapıldıktan sonra bilgileri girilen tedarikçinin kaydedilmesi için “EKLE” butonunun tıklanma olayına Görsel 2.23’teki kodlar yazılır.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliTedarikciID==0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                tedarikciler tedarikci = new tedarikciler();
                tedarikci.firma_adi = txtFirmaAdi.Text;
                tedarikci.yetkili_adsoyad = txtYetkiliAdi.Text;
                tedarikci.telefon = mskTelefon.Text;
                tedarikci.mail = txtMail.Text;
                tedarikci.adres = txtAdres.Text;
                stok_Takip.tedarikciler.Add(tedarikci);
                stok_Takip.SaveChanges();
                TedarikcileriListele();
                FormuTemizle();
            }
        }
    }
}
```



```

        else
        {
            MessageBox.Show("Yeni tedarikçi eklemek için 'Yeni Kayıt' butonuna tıklayınız");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Kayıt Hatası");
    }
}

```

Görsel 2.23: Tedarikçi ekleme kodları

Yapılan işlemlerden sonra form elemanlarının içeriklerini temizlemek için Görsel 2.24'teki "FormuTemizle" metodu oluşturulur.

```

private void FormuTemizle()
{
    txtFirmaAdi.Clear();
    txtYetkiliAdi.Clear();
    txtMail.Clear();
    txtAdres.Clear();
    mskTelefon.Clear();
    grdTedarikciler.ClearSelection();
    seciliTedarikciID = 0;
}

```

Görsel 2.24: Form elemanlarının içeriklerini temizleme kodları

DataGridView üzerinde seçilen tedarikçinin silinmesi için "SİL" butonunun tıklanma olayına Görsel 2.25'teki kodlar yazılır.

```

private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliTedarikciID != 0)
        {
            var tedarikciKontrol = stok_Takip.urunler.
                Where(u => u.tedarikci_id.ToString().Contains(seciliTedarikciID.ToString())).ToList();
            if (tedarikciKontrol.Count() > 0)
            {
                MessageBox.Show("Bu kayıt silinemez. Silmek istediğiniz tedarikçiye ait ürünler mevcut.");
            }
            else
            {
                stok_Takip = new stok_takip_dbEntities();
                tedarikciler silinecekTedarikci = stok_Takip.tedarikciler.
                    Where(t => t.id == seciliTedarikciID).First();
                stok_Takip.tedarikciler.Remove(silinecekTedarikci);
                stok_Takip.SaveChanges();
                TedarikcileriListele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Silinecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme İşleminde Hata");
    }
}

```

Görsel 2.25: Tedarikçi silme kodları



DataGridView üzerinde seçilen tedarikçinin güncellenmesi için “GÜNCELLE” butonunun tıklanma olayına Görsel 2.26’daki kodlar yazılır.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliTedarikciID != 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                tedarikciler guncellenecekTedarikci = stok_Takip.tedarikciler.
                    Where(t => t.id == seciliTedarikciID).First();
                guncellenecekTedarikci.firma_adi = txtFirmaAdi.Text;
                guncellenecekTedarikci.yetkili_adsoyad = txtYetkiliAdi.Text;
                guncellenecekTedarikci.telefon = mskTelefon.Text;
                guncellenecekTedarikci.mail = txtMail.Text;
                guncellenecekTedarikci.adres = txtAdres.Text;
                stok_Takip.SaveChanges();
                Tedarikcilerilistele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Güncellenecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Güncelleme Hatası");
    }
}

1 reference
private void btnYeniKayit_Click(object sender, EventArgs e)
{
    FormuTemizle();
}
```

Görsel 2.26: Tedarikçi güncelleme kodları

Yeni kayıt butonunun tıklanma olayında “FormuTemizle” metodu çağrılarak Tedarikçi İşlemleri sayfası tamamlanır.

2.3.5. Ürün İşlemleri

Projede “frmUrunIslemleri” isimli yeni bir form oluşturulur (Görsel 2.27). Formdaki nesneler; **txtUrunAdi**, **txtAciklama**, **nmrStok**, **txtFiyat**, **cmbKategori**, **btnKategori**, **cmbTedarikci**, **btnYeni**, **btnEkle**, **btnSil**, **btnGuncelle**, **grdUrunler** şeklinde isimlendirilir. Form elemanlarının özellikleri uygun şekilde belirlenir. DataGridView nesnesi üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.

Görsel 2.27: Ürün işlemleri formunun tasarımı



“frmUrunIslemleri” sayfasının arka planına ekleme, silme, güncelleme ve listeleme işlemleri için gerekli kodlar yazılır. Sayfanın yüklenmesi olayında bilgilerin gerekli form elemanlarında listelenmesi için Görsel 2.28’deki kodlar yazılır.

```

stok_takip_dbEntities stok_Takip;
int seciliUrunID = 0;
1 reference
private void frmUrunIslemleri_Load(object sender, EventArgs e)
{
    BilgileriListele();
}
5 references
private void BilgileriListele()
{
    // Bilgilerin anlamlı olması için Ürünler kategori adı ve tedarikçi adı ile listelenir
    stok_Takip = new stok_takip_dbEntities();
    var sonuc = (from u in stok_Takip.urunler
        join k in stok_Takip.kategoriler on u.kategori_id equals k.kategori_id
        join t in stok_Takip.tedarikciler on u.tedarikci_id equals t.id
        select new
        {
            u.urun_kodu,u.urun_adi,u.stok_adedi,u.fiyat,k.kategori_adi,t.firma_adi,u.aciklama
        }).ToList();
    grdUrunler.DataSource = sonuc;
    grdUrunler.Columns["urun_kodu"].Width = 60;
    grdUrunler.ClearSelection();
    // Seçilecek olan kategori ve tedarikçi bilgileri ComboBox nesnelerinde listelenir.
    cmbKategori.DataSource = stok_Takip.kategoriler.OrderBy(k => k.kategori_adi).ToList();
    cmbKategori.DisplayMember = "kategori_adi";
    cmbKategori.ValueMember = "kategori_id";
    cmbTedarikci.DataSource = stok_Takip.tedarikciler.OrderBy(t => t.firma_adi).ToList();
    cmbTedarikci.DisplayMember = "firma_adi";
    cmbTedarikci.ValueMember = "id";
}

```

Görsel 2.28: Bilgilerin form elemanlarında listelenmesi

Ürün ekleme ve güncelleme işlemleri için “BilgiGirisKontrol” isimli bir metot oluşturularak kullanıcıdan alınan bilgilerin doğruluğu ve geçerliliği kontrol edilir (Görsel 2.29).

```

private bool BilgiGirisKontrol()
{
    decimal kont;
    bool kontrol = false;
    if (string.IsNullOrEmpty(txtUrunAdi.Text))
    {
        MessageBox.Show("Ürün adını giriniz");
    }
    else if (!Decimal.TryParse(txtFiyat.Text, out kont))
    {
        // Girilen değer Decimal'e dönüştürülemediyse mesaj gösterilir
        MessageBox.Show("Ürün fiyatını giriniz");
    }
    else
    {
        kontrol = true;
    }
    return kontrol;
}

```

Görsel 2.29: Ürün bilgilerinin kontrolü için gerekli kodlar



Gerekli kontroller yapıldıktan sonra bilgileri girilen ürünün kaydedilmesi için “EKLE” butonunun tıklanma olayına Görsel 2.30’daki kodlar yazılır.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliUrunID == 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                urunler urun = new urunler();
                urun.urun_adi = txtUrunAdi.Text;
                urun.kategori_id = Convert.ToInt32(cmbKategori.SelectedValue);
                urun.tedarikci_id = Convert.ToInt32(cmbTedarikci.SelectedValue);
                urun.stok_adedi = Convert.ToInt32(nmrStok.Text);
                urun.fiyat = Convert.ToDecimal(txtFiyat.Text);
                urun.aciklama = txtAciklama.Text;
                stok_Takip.urunler.Add(urun);
                stok_Takip.SaveChanges();
                Bilgilerilistele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Yeni ürün eklemek için 'Yeni Kayıt' butonuna tıklayınız");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Kayıt Hatası");
    }
}
```

Görsel 2.30: Ürün ekleme kodları

Yapılan işlemlerden sonra form elemanlarının içeriklerini temizlemek için Görsel 2.31’deki “FormuTemizle” metodu oluşturulur ve gerekli yerlerde kullanılır.

```
private void FormuTemizle()
{
    txtUrunAdi.Clear();
    txtAciklama.Clear();
    txtFiyat.Clear();
    nmrStok.Value = 0;
    grdUrunler.ClearSelection();
    seciliUrunID = 0;
}
```

Görsel 2.31: Form elemanlarının içeriklerini temizleme kodları



Silme ve güncelleme işlemleri için DataGridView nesnesi üzerinden ürün seçilmesini sağlayan kodlar Görsel 2.32'deki gibi yazılır. Ürün seçildikten sonra bu ürünü silme veya güncelleme işlemi yapılabilir.

```
private void grdUrunler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdUrunler.CurrentRow != null)
        {
            seciliUrunID = Convert.ToInt32(grdUrunler.CurrentRow.Cells["urun_kodu"].Value.ToString());
            txtUrunAdi.Text = grdUrunler.CurrentRow.Cells["urun_adi"].Value.ToString();
            txtAciklama.Text = grdUrunler.CurrentRow.Cells["aciklama"].Value.ToString();
            txtFiyat.Text = grdUrunler.CurrentRow.Cells["fiyat"].Value.ToString();
            nmrStok.Text = grdUrunler.CurrentRow.Cells["stok_adedi"].Value.ToString();
            cmbKategori.Text = grdUrunler.CurrentRow.Cells["kategori_adi"].Value.ToString();
            cmbTedarikci.Text = grdUrunler.CurrentRow.Cells["firma_adi"].Value.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.32: DataGridView üzerinde işlem yapılacak kaydın seçilmesini sağlayan kodlar

DataGridView üzerinde seçilen ürünün silinmesi için “SİL” butonunun tıklanma olayına Görsel 2.33'teki kodlar yazılır.

```
private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliUrunID != 0)
        {
            stok_Takip = new stok_takip_dbEntities();
            urunler silinecekUrun = stok_Takip.urunler.
                Where(u => u.urun_kodu == seciliUrunID).First();
            stok_Takip.urunler.Remove(silinecekUrun);
            stok_Takip.SaveChanges();
            Bilgilerilistele();
            FormuTemizle();
        }
        else
        {
            MessageBox.Show("Silinecek ürünü seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme İşleminde Hata");
    }
}
```

Görsel 2.33: Seçilen ürünü silme kodları



DataGridView üzerinde seçilen ürünün güncellenmesi için “GÜNCELLE” butonunun tıklanma olayına Görsel 2.34’teki kodlar yazılır.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliUrunID != 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                urunler guncellenecekUrun = stok_Takip.urunler.Where(u => u.urun_kodu == seciliUrunID).First();
                guncellenecekUrun.urun_adi = txtUrunAdi.Text;
                guncellenecekUrun.kategori_id = Convert.ToInt32(cmbKategori.SelectedValue);
                guncellenecekUrun.tedarikci_id = Convert.ToInt32(cmbTedarikci.SelectedValue);
                guncellenecekUrun.stok_adi = (int)nmrStok.Value;
                guncellenecekUrun.fiyat = Convert.ToDecimal(txtFiyat.Text);
                guncellenecekUrun.aciklama = txtAciklama.Text;
                stok_Takip.SaveChanges();
                Bilgilerilistele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Güncellenecek ürünü seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Güncelleme Hatası");
    }
}
```

Görsel 2.34: Seçilen ürünü güncelleme kodları

Kategorileri içeren ComboBox nesnesinin yanındaki butona tıklandığında “frmKategorislemeleri” sayfasının açılması için gerekli kodlar Görsel 2.35’te verilmiştir. Form kapatıldığında yapılan değişikliklerin “frmUrunislemeleri” sayfasında görüntülenebilmesi için bilgiler tekrar listelenir.

```
private void btnKategori_Click(object sender, EventArgs e)
{
    frmKategoriIslemeleri frmKategori = new frmKategoriIslemeleri();
    DialogResult result = frmKategori.ShowDialog();
    if (result == DialogResult.Cancel)
    {
        Bilgilerilistele();
    }
}
```

Görsel 2.35: Kategori işlemleri sayfasına yönlendirme için gerekli kodlar

btnYeniKayit butonunun tıklanması olayında “FormuTemizle” metodu çağrılarak “frmUrunislemeleri” sayfası tamamlanır.

2.3.6. Müşteri İşlemleri

Projede “frmMusterislemeleri” isimli yeni bir form oluşturulur (Görsel 2.36). Formdaki nesneler; **txtAdSoyad**, **txtFirmaAdi**, **txtAdres**, **mskTelefon**, **txtMail**, **btnYeni**, **btnEkle**, **btnSil**, **btnGuncelle**, **grdMusteriler** şeklinde isimlendirilir. Form elemanlarının özellikleri uygun şekilde belirlenir. DataGridView nesnesi üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.



Görsel 2.36: Müşteri işlemleri formunun tasarımı

“frmMusteriIslemleri” sayfasının arka planına ekleme, silme, güncelleme ve listeleme işlemleri için gerekli kodlar yazılır. Sayfanın yüklenmesi olayında müşteriler listelenir. Silme ve güncelleme işlemleri için DataGridView üzerinde seçim yapılması sağlanır (Görsel 2.37).

```

stok_takip_dbEntities stok_Takip;
int seciliMusteriID = 0;
1 reference
private void frmMusteriIslemleri_Load(object sender, EventArgs e)
{
    Musterileriliste();
}
4 references
private void Musterileriliste()
{
    stok_Takip = new stok_takip_dbEntities();
    grdMusteriler.DataSource = stok_Takip.musteriler.OrderBy(m=>m.ad_boyad).ToList();
    grdMusteriler.Columns.Remove("satislar");
    grdMusteriler.Columns["id"].Width = 35;
    grdMusteriler.ClearSelection();
}
1 reference
private void grdMusteriler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdMusteriler.CurrentRow != null)
        {
            seciliMusteriID = Convert.ToInt32(grdMusteriler.CurrentRow.Cells["id"].Value.ToString());
            txtFirmaAdi.Text = grdMusteriler.CurrentRow.Cells["firma_adi"].Value.ToString();
            txtAdSoyad.Text = grdMusteriler.CurrentRow.Cells["ad_boyad"].Value.ToString();
            txtMail.Text = grdMusteriler.CurrentRow.Cells["mail"].Value.ToString();
            txtAdres.Text = grdMusteriler.CurrentRow.Cells["adres"].Value.ToString();
            mskTelefon.Text = grdMusteriler.CurrentRow.Cells["telefon"].Value.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Görsel 2.37: Müşterilerin listelenmesi ve listelenen kayıtlardan seçim yapılması



Ekleme ve güncelleme işlemleri için “BilgiGirisKontrol” isimli bir metod oluşturularak kullanıcıdan alınan bilgilerin doğruluğu ve geçerliliği kontrol edilir (Görsel 2.38).

```
private bool BilgiGirisKontrol()
{
    bool kontrol = false;
    if (txtAdSoyad.TextLength <= 2)
    {
        MessageBox.Show("Müşteri adını soyadını giriniz.");
    }
    else if (!mskTelefon.MaskCompleted)
    {
        MessageBox.Show("Telefon numarasını 10 haneli olarak giriniz.");
    }
    else
    {
        kontrol = true;
    }
    return kontrol;
}
```

Görsel 2.38: Müşteri bilgilerinin doğruluk kontrolü

Gerekli kontroller yapıldıktan sonra bilgileri girilen müşterinin kaydedilmesi için “EKLE” butonunun tıklanma olayına Görsel 2.39’daki kodlar yazılır.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliMusteriID == 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                musteriler musteriler = new musteriler();
                musteriler.firma_adi = txtFirmaAdi.Text;
                musteriler.ad_soyad = txtAdSoyad.Text;
                musteriler.telefon = mskTelefon.Text;
                musteriler.mail = txtMail.Text;
                musteriler.adres = txtAdres.Text;
                stok_Takip.musteriler.Add(musteriler);
                stok_Takip.SaveChanges();
                Musterilerilistele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Yeni müşteri eklemek için 'Yeni Kayıt' butonuna tıklayınız");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Kayıt Hatası");
    }
}
```

Görsel 2.39: Müşteri ekleme kodları



Yapılan işlemlerden sonra form elemanlarının içeriklerini temizlemek için Görsel 2.40'taki "FormuTemizle" metodu oluşturulur.

```
private void FormuTemizle()
{
    txtFirmaAdi.Clear();
    txtAdSoyad.Clear();
    txtMail.Clear();
    txtAdres.Clear();
    mskTelefon.Clear();
    grdMusteriler.ClearSelection();
    seciliMusteriID = 0;
}
```

Görsel 2.40: Form elemanlarının içeriklerini temizleme kodları

DataGridView üzerinde seçilen müşterinin silinmesi için "SİL" butonunun tıklanma olayına Görsel 2.41'deki kodlar yazılır.

```
private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliMusteriID != 0)
        {
            stok_Takip = new stok_takip_dbEntities();
            musteriler silinecekMusteri = stok_Takip.musteriler.
                Where(t => t.id == seciliMusteriID).First();
            stok_Takip.musteriler.Remove(silinecekMusteri);
            stok_Takip.SaveChanges();
            MusterileriListele();
            FormuTemizle();
        }
        else
        {
            MessageBox.Show("Silinecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme İşleminde Hata");
    }
}
```

Görsel 2.41: Müşteri silme kodları



DataGridView üzerinde seçilen müşterinin güncellenmesi için “GÜNCELLE” butonunun tıklanma olayına Görsel 2.42’deki kodlar yazılır.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliMusteriID != 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                musteriler guncellenecekMusteri = stok_Takip.musteriler.
                    Where(t => t.id == seciliMusteriID).First();
                guncellenecekMusteri.firma_adi = txtFirmaAdi.Text;
                guncellenecekMusteri.ad_soyad = txtAdSoyad.Text;
                guncellenecekMusteri.telefon = mskTelefon.Text;
                guncellenecekMusteri.mail = txtMail.Text;
                guncellenecekMusteri.adres = txtAdres.Text;
                stok_Takip.SaveChanges();
                MusterileriListele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Güncellenecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Güncelleme Hatası");
    }
}
```

Görsel 2.42: Müşteri güncelleme kodları

Yeni kayıt butonunun tıklanma olayında “FormuTemizle” metodu çağrılarak frmMusteriIslemleri sayfası tamamlanır.

2.3.7. Personel İşlemleri

Projede “frmPersonelIslemleri” isimli yeni bir form oluşturulur (Görsel 2.43). Formdaki nesneler; **txtAd**, **txtSoyad**, **txtAdres**, **mskTelefon**, **txtMail**, **txtKullanici**, **txtSifre**, **btnYeni**, **btnEkle**, **btnSil**, **btnGuncelle**, **grdPersoneller** şeklinde isimlendirilir. Form elemanlarının özellikleri uygun şekilde belirlenir. DataGridView nesnesi üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.



Görsel 2.43: Personel işlemleri formunun tasarımı

frmPersonelIslemleri sayfasının arka planına ekleme, silme, güncelleme ve listeleme işlemleri için gerekli kodlar yazılır. Sayfanın yüklenmesi olayında personeller ve yetkiler listelenir (Görsel 2.44). Uygulamada kullanılacak iki yetkili (Personel, Yönetici) olduğu için bunlar, veri tabanındaki yetkiler tablosuna VTYS üzerinden kaydedilir.

```
public int oturumAcanPersonelId;
stok_takip_dbEntities stok_Takip;
int seciliPersonelId = 0;
1 reference
private void frmPersonelIslemleri_Load(object sender, EventArgs e)
{
    Bilgilerilistele();
}
4 references
private void Bilgilerilistele()
{
    stok_Takip = new stok_takip_dbEntities();
    var sonuc = (from p in stok_Takip.personeller
                join y in stok_Takip.yetkiler on p.yetki_id equals y.yetki_id
                select new
                {
                    p.id,p.ad,p.soyad,p.telefon,p.mail,p.adres,p.kullanici_adi,p.sifre,y.yetki_adi
                }).ToList();
    grdPersoneller.DataSource = sonuc;
    grdPersoneller.Columns["id"].Width = 35;
    grdPersoneller.Columns["ad"].Width = 80;
    grdPersoneller.Columns["soyad"].Width = 80;
    grdPersoneller.Columns["yetki_adi"].Width = 80;
    grdPersoneller.Columns["kullanici_adi"].Width = 80;
    grdPersoneller.ClearSelection();
    // Seçilecek olan yetki bilgileri ComboBox nesnesinde listelenir.
    cmbYetkiler.DataSource = stok_Takip.yetkiler.ToList();
    cmbYetkiler.DisplayMember = "yetki_adi";
    cmbYetkiler.ValueMember = "yetki_id";
}
```

Görsel 2.44: DataGridView nesnesinde personellerin listelenmesi



Ekleme ve güncelleme işlemleri için “BilgiGirisKontrol” isimli bir metod oluşturularak kullanıcıdan alınan bilgilerin doğruluğu ve geçerliliği kontrol edilir (Görsel 2.45).

```
private bool BilgiGirisKontrol()
{
    bool kontrol = false;
    if (txtAd.TextLength <= 2)
    { //Bilgi girişi kontrolü farklı yöntemlerle yapılabilir
        MessageBox.Show("Personelin adını giriniz.");
    }
    else if (txtSoyad.TextLength <= 2)
    {
        MessageBox.Show("Personelin soyadını giriniz.");
    }
    else if (!mskTelefon.MaskCompleted)
    {
        MessageBox.Show("Telefon numarasını 10 haneli olarak giriniz.");
    }
    else if (txtKullanici.TextLength < 5)
    {
        MessageBox.Show("Personelin kullanıcı adını giriniz.");
    }
    else
    {
        kontrol = true;
    }
    return kontrol;
}
```

Görsel 2.45: Girilen bilgilerin doğruluk kontrolü

Güvenlik için personellere ait parolaların şifrlenerek veri tabanına kaydedilmesi gerekir. Bu yüzden kayıt işleminden önce bu değerler şifrelenir. Şifreleme için farklı yöntemler kullanılabilir. Girilen değeri şifrelemek için Görsel 2.46’daki metod oluşturulur.

```
private static string Sifrele(string metin)
{ // Basit bir şifreleme yöntemi - MD5
    byte[] byteDegeri = Encoding.UTF8.GetBytes(metin);
    MD5CryptoServiceProvider csp = new MD5CryptoServiceProvider();
    byte[] sifreliByteDegeri = csp.ComputeHash(byteDegeri);
    return Convert.ToBase64String(sifreliByteDegeri);
}
```

Görsel 2.46: Girilen metni şifreleme kodları

Gerekli kontroller yapıldıktan sonra bilgileri girilen personelin kaydedilmesi için “EKLE” butonunun tıklanma olayına Görsel 2.47’deki kodlar yazılır.



```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliPersonelId == 0)
        {
            if (BilgiGirisKontrol())
            {
                if (txtSifre.TextLength >= 8)
                {
                    stok_Takip = new stok_takip_dbEntities();
                    personeller personel = new personeller();
                    personel.ad = txtAd.Text;
                    personel.soyad = txtSoyad.Text;
                    personel.telefon = mskTelefon.Text;
                    personel.mail = txtMail.Text;
                    personel.adres = txtAdres.Text;
                    personel.kullanici_adi = txtKullanici.Text;
                    personel.sifre = Sifrele(txtSifre.Text);
                    personel.yetki_id = Convert.ToInt32(cmbYetkiler.SelectedValue);
                    stok_Takip.personeller.Add(personel);
                    stok_Takip.SaveChanges();
                    BilgileriListele();
                    FormuTemizle();
                }
                else
                {
                    MessageBox.Show("En az 8 haneli bir şifre giriniz.");
                }
            }
            else
            {
                MessageBox.Show("Yeni personel eklemek için 'Yeni Kayıt' butonuna tıklayınız");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Kayıt Hatası");
        }
    }
}
```

Görsel 2.47: Personel ekleme kodları

Yapılan işlemlerden sonra form elemanlarının içeriklerini temizlemek için Görsel 2.48'deki "FormuTemizle" metodu oluşturulur.

```
private void FormuTemizle()
{
    txtAd.Clear();
    txtSoyad.Clear();
    txtAdres.Clear();
    txtMail.Clear();
    txtKullanici.Clear();
    txtSifre.Clear();
    mskTelefon.Clear();
    grdPersoneller.ClearSelection();
    seciliPersonelId = 0;
}
```

Görsel 2.48: Form elemanlarının içeriklerini temizleme kodları



DataGridView nesnesinde listelenen kayıtlardan işlem yapılacak personelin seçilmesi için gereken kodlar Görsel 2.49'daki gibi yazılır. Böylece silme ve güncelleme işleminin hangi personel için yapılacağı belirlenir. Seçilen kayda ait bilgiler, ilgili form elemanlarında listelenir.

```
private void grdPersoneller_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdPersoneller.CurrentRow != null)
        {
            seciliPersonelId = Convert.ToInt32(grdPersoneller.CurrentRow.Cells["id"].Value.ToString());
            txtAd.Text = grdPersoneller.CurrentRow.Cells["ad"].Value.ToString();
            txtSoyad.Text = grdPersoneller.CurrentRow.Cells["soyad"].Value.ToString();
            txtMail.Text = grdPersoneller.CurrentRow.Cells["mail"].Value.ToString();
            txtAdres.Text = grdPersoneller.CurrentRow.Cells["adres"].Value.ToString();
            mskTelefon.Text = grdPersoneller.CurrentRow.Cells["telefon"].Value.ToString();
            txtKullanici.Text = grdPersoneller.CurrentRow.Cells["kullanici_adi"].Value.ToString();
            cmbYetkiler.Text = grdPersoneller.CurrentRow.Cells["yetki_adi"].Value.ToString();
            txtSifre.Clear();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.49: İşlem yapılacak personelin seçilmesi için gereken kodlar

DataGridView üzerinde seçilen personelin silinmesi için “SİL” butonunun tıklanma olayına Görsel 2.50'deki kodlar yazılır.

```
private void btnSil_Click(object sender, EventArgs e)
{
    try
    {
        // Gerekli kontrol yapılarak oturum açan personelin silinmesi engellenir.
        if (seciliPersonelId != oturumAcanPersonelId)
        {
            if (seciliPersonelId != 0)
            {
                stok_Takip = new stok_takip_dbEntities();
                personeller silinecekPersonel = stok_Takip.personeller.
                    Where(p => p.id == seciliPersonelId).First();
                stok_Takip.personeller.Remove(silinecekPersonel);
                stok_Takip.SaveChanges();
                BilgileriListele();
                FormuTemizle();
            }
            else
            {
                MessageBox.Show("Silinecek kaydı seçiniz.");
            }
        }
        else
        {
            MessageBox.Show("Oturum sahibi olan personel silinemez!");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Silme İşleminde Hata");
    }
}
```

Görsel 2.50: Personel silme kodları



DataGridView üzerinde seçilen personelin güncellenmesi için “GÜNCELLE” butonunun tıklanma olayına Görsel 2.51’deki kodlar yazılır.

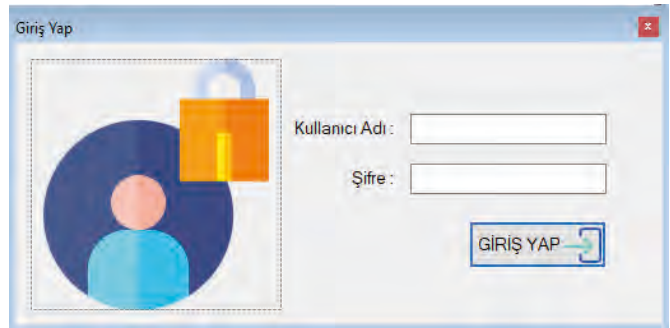
```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliPersonelId != 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                personeller guncellenecekPersonel = stok_Takip.personeller.
                    Where(t => t.id == seciliPersonelId).First();
                guncellenecekPersonel.ad = txtAd.Text;
                guncellenecekPersonel.soyad = txtSoyad.Text;
                guncellenecekPersonel.telefon = mskTelefon.Text;
                guncellenecekPersonel.mail = txtMail.Text;
                guncellenecekPersonel.adres = txtAdres.Text;
                guncellenecekPersonel.kullanici_adi = txtKullanici.Text;
                guncellenecekPersonel.yetki_id = Convert.ToInt32(cmbYetkiler.SelectedValue);
                // Şifre alanı boş gelecek, sadece geçerli bir şifre girilirse güncellenecektir.
                if (txtSifre.TextLength >= 8)
                {
                    guncellenecekPersonel.sifre = Sifrele(txtSifre.Text);
                }
                stok_Takip.SaveChanges();
                BilgileriListele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Güncellenecek kaydı seçiniz.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Güncelleme Hatası");
    }
}
```

Görsel 2.51: Personel güncelleme kodları

Yeni kayıt butonunun tıklanma olayında “FormuTemizle” metodu çağrılarak “frmPersonelIslemleri” sayfası tamamlanır. Giriş bölümü tamamlandıktan sonra sisteme giriş yapılabilmesi için en az bir adet “Yönetici” yetkisine sahip personel tanımlanmalıdır.

2.3.8. Giriş Sayfası

Personel işlemleri tamamlandıktan sonra projede “frmGiris” isimli bir sayfa oluşturulur (Görsel 2.52). Bu sayfa, kullanıcıların, kullanıcı adı ve şifresi ile sisteme giriş yapmasını sağlar. Formun içine **txtKullaniciAdi**, **txtSifre** ve **btnGiris** nesneleri eklenir. Girilen şifrenin karakterlerinin görünmemesi için “txtSifre” nesnesinin **PasswordChar** özelliğine “*” sembolü yazılır.



Görsel 2.52: Giriş sayfasının tasarımı



Şifre kontrolünün yapılabilmesi için kullanıcının girdiği parolanın şifrelenerek veri tabanındaki değerle eşleştirilmesi gerekir. Şifreleme için Görsel 2.53'teki metod oluşturularak gerekli yerde kullanılır.

```
private static string Sifrele(string metin)
{
    byte[] byteDegeri = Encoding.UTF8.GetBytes(metin);
    MD5CryptoServiceProvider csp = new MD5CryptoServiceProvider();
    byte[] sifreliByteDegeri = csp.ComputeHash(byteDegeri);
    return Convert.ToBase64String(sifreliByteDegeri);
}
```

Görsel 2.53: Şifreleme metodunun kodları

Kullanıcı adı ve şifre değeri girildikten sonra "GİRİŞ YAP" butonu tıklanarak kullanıcının sisteme giriş yapabilmesi için butonun tıklanma olayına Görsel 2.54'teki kodlar yazılır.

```
public static int personelId;
public static string personelAdSoyad;
public static string personelYetki;
stok_takip_dbEntities stok_Takip;
private void btnGiris_Click(object sender, EventArgs e)
{
    try
    {
        stok_Takip = new stok_takip_dbEntities();
        string kAdi = txtKullaniciAdi.Text;
        string sifre = Sifrele(txtSifre.Text);
        var sonuc = (from p in stok_Takip.personeller
                    join y in stok_Takip.yetkiler
                    on p.yetki_id equals y.yetki_id
                    where p.kullanici_adi == kAdi && p.sifre == sifre
                    select new {
                        p.id, p.ad, p.soyad, p.kullanici_adi, p.sifre, y.yetki_adi
                    }).FirstOrDefault();
        if (sonuc != null)
        {
            this.Hide();
            frmAnaSayfa frmAnaSayfa = new frmAnaSayfa();
            frmAnaSayfa.PersonelId = sonuc.id;
            frmAnaSayfa.personelAdSoyad = sonuc.ad + " " + sonuc.soyad;
            frmAnaSayfa.personelYetki = sonuc.yetki_adi;
            frmAnaSayfa.ShowDialog();
        }
        else
        {
            MessageBox.Show("Kullanıcı adı veya şifre hatalı");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Giriş hatası");
    }
}
```

Görsel 2.54: Kullanıcı girişi için gerekli kodlar



Uygulama çalıştırıldığında ilk olarak giriş sayfasının görüntülenmesi için “Program.cs” dosyasında başlangıç formunun “frmGiris” olarak değiştirilmesi gerekir (Görsel 2.55).

```
static class Program
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    0 references
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new frmGiris());
    }
}
```

Görsel 2.55: Başlangıç formunun belirlenmesi

Giriş sayfası kapatıldığında uygulamanın sonlandırılması için formun kapanması olayına Görsel 2.56’daki kodlar yazılır.

```
private void frmGiris_FormClosed(object sender, FormClosedEventArgs e)
{
    Application.Exit();
}
```

Görsel 2.56: Uygulamanın kapatılması kodları

2.3.9. Proje Anasayfası

Projede “frmAnaSayfa” isimli bir form oluşturulur (Görsel 2.57). Bu form hem diğer formlara yönlendirme işleminin yapılacağı hem de satış işleminin gerçekleştirileceği bölümdür. Bu sayfada listelenen ürünlerin içinden seçilen ürünler satış listesine eklenir. Satış yapılacak müşteri, telefon numarası girilerek seçilir. “Satış Yap” butonu tıklandığında satış listesindeki ürünlerin seçilen müşteriye satışı gerçekleştirilir.

Görsel 2.57: Anasayfa formunun tasarımı

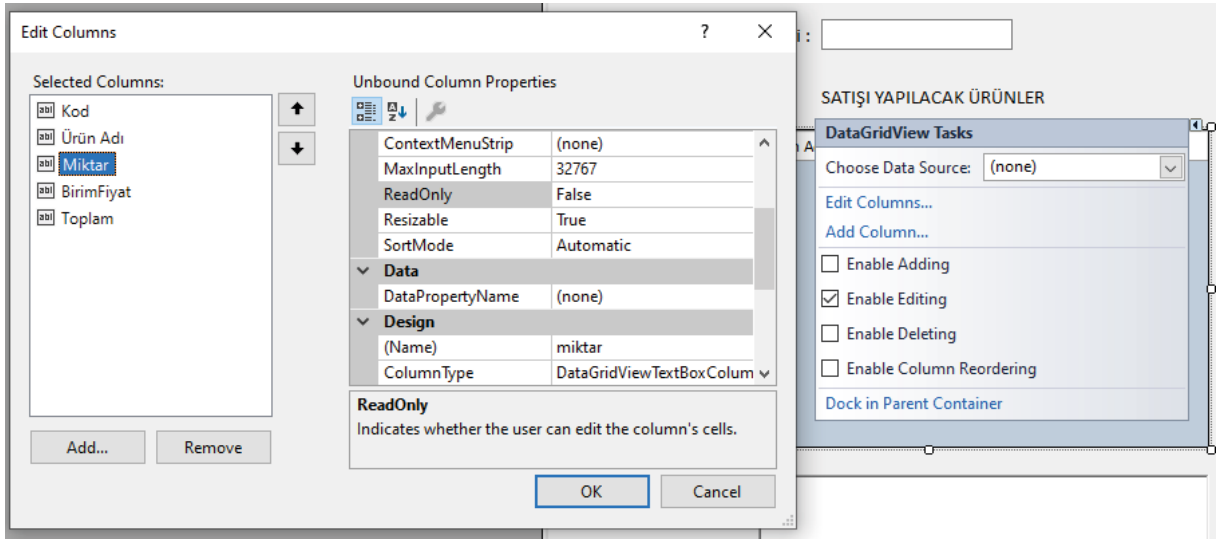


“frmAnaSayfa” formundaki menü elemanları sırasıyla **menuTedarikci**, **menuUrun**, **menuSatis**, **menuMusteri**, **menuPersonel** olarak isimlendirilir. Menünün sağ tarafında bulunan label nesnelerinin isimleri de **lblPersonelAdSoyad**, **lblYetki** olarak belirlenir.

Diğer nesneler; **mskTxtMusteriTelefon**, **btnMusteriSec**, **txtMusteriAdSoyad (Enabled:False)**, **btnEkle**, **btnSil**, **txtNot**, **btnSatisYap**, **txtGenelToplam(Enabled:False)** olarak isimlendirilir.

Ürünlerin listeleneceği, sol bölümde bulunan DataGridView nesnesi **grdUrunListesi** olarak isimlendirilir. DataGridView üzerinde ekleme, silme ve güncelleme işlemlerinin yapılması engellenir.

Satılacak ürünlerin listeleneceği, sağ bölümde bulunan DataGridView nesnesine **grdSatilacakUrunler** ismi verilerek içine görseldeki sütunlar eklenir. Bu nesnede satış değeri girilebileceği için üzerinde düzenleme yapılmasına izin verilir fakat ekleme ve silme işlemlerinin yapılması engellenir. Sadece “Miktar” sütununa veri girişi yapılacağı için bu sütunun ReadOnly özelliği False, diğer sütunların ReadOnly özelliği ise True olarak ayarlanır (Görsel 2.58).



Görsel 2.58: grdSatilacakUrunler nesnesinin sütunlarını düzenleme

Anasayfanın üst bölümündeki menülere tıklandığında gerekli sayfaların açılması için menü elemanlarının tıklanma olaylarına Görsel 2.59’daki kodların yazılması gerekir. Açılan sayfada yapılan değişikliklerin anasayfaya uygulanması için sayfa kapandığında bilgiler yeniden listelenir.

```
private void menuTedarikci_Click(object sender, EventArgs e)
{
    frmTedarikciIslemleri frmTedarikci = new frmTedarikciIslemleri();
    DialogResult result = frmTedarikci.ShowDialog();
    if (result == DialogResult.Cancel)
    {
        BilgileriListele();
    }
}

private void menuUrun_Click(object sender, EventArgs e)
{
    frmUrunIslemleri frmUrun = new frmUrunIslemleri();
    DialogResult result = frmUrun.ShowDialog();
    if (result == DialogResult.Cancel)
    {
        BilgileriListele();
        grdSatilacakUrunler.Rows.Clear();
    }
}
```



```
private void menuSatis_Click(object sender, EventArgs e)
{
    frmSatisDetaylari frmSatisDetaylari = new frmSatisDetaylari();
    DialogResult result = frmSatisDetaylari.ShowDialog();
    if (result == DialogResult.Cancel)
    { BilgileriListele(); }
}
// reference
private void menuMusteri_Click(object sender, EventArgs e)
{
    frmMusteriIslemleri frmMusteri = new frmMusteriIslemleri();
    DialogResult result = frmMusteri.ShowDialog();
    if (result == DialogResult.Cancel)
    { FormuTemizle(); }
}
// reference
private void menuPersonel_Click(object sender, EventArgs e)
{ // PersonelId değeri personel işlemleri sayfasına gönderilir
    frmPersonelIslemleri frmPersonel = new frmPersonelIslemleri();
    frmPersonel.oturumAcanPersonelId = PersonelId;
    frmPersonel.ShowDialog();
}
// reference
private void frmAnaSayfa_FormClosed(object sender, FormClosedEventArgs e)
{ // Anasayfa kapatıldığında giriş sayfasına geri dönülür
    frmGiris frm = new frmGiris();
    frm.Show();
}
```

Görsel 2.59: Diğer sayfalara yönlendirme işlemi için gerekli kodlar

Form elemanlarının içeriklerinin temizlenmesi için Görsel 2.60'taki "FormuTemizle" metodu oluşturulur ve bu metod gerekli yerlerde kullanılır.

```
private void FormuTemizle()
{
    seciliMusteriId = 0;
    txtMusteriAdSoyad.Clear();
    mskTxtMusteriTelefon.Clear();
    txtNot.Clear();
    grdUrunListesi.ClearSelection();
}
```

Görsel 2.60: Form elemanlarının içeriklerini temizleme



NOT

Veri tabanı işlemleri için oluşturulan "stok_takip_dbEntities" sınıfını kullanmak için using yöntemi tercih edilmelidir. Bu yöntemde veri tabanı bağlantısı yalnızca using ifadesinin kapsamında açık kalır. Bir başka deyişle gereksiz yere veri tabanı bağlantısının açık kalması engellenir.



Satılacak ürünlerin satış listesine eklenebilmesi için formun yüklenmesi olayında ürünler, “grdUrunListesi” isimli DataGridView nesnesinde listelenir (Görsel 2.61).

```
int seciliMusteriId = 0;
public int PersonelId = 0;
public string personelAdSoyad = "";
public string personelYetki = "";
1 reference
private void frmAnaSayfa_Load(object sender, EventArgs e)//
{
    lblPersonelAdSoyad.Text = personelAdSoyad;
    lblYetki.Text = personelYetki;
    // Giriş yapan personel "yönetici" değilse Personel İşlemleri menüsü pasif duruma getirilir
    if (personelYetki == "Yönetici")
        menuPersonel.Enabled = true;
    else
        menuPersonel.Enabled = false;
    BilgileriListele();
}
6 references
private void BilgileriListele()//
{
    // Ürünler listelenir ve satışı yapılacak ürünler buradan seçilir.
    using (var stok_Takip = new stok_takip_dbEntities())
    {
        var urunSonuc = (from u in stok_Takip.urunler
                        join k in stok_Takip.kategoriler on u.kategori_id equals k.kategori_id
                        join t in stok_Takip.tedarikciler on u.tedarikci_id equals t.id
                        where u.stok_adedi >= 1
                        select new
                        {
                            u.urun_kodu, u.urun_adi, u.stok_adedi, u.fiyat, k.kategori_adi, t.firma_adi, u.aciklama
                        }).ToList();
        grdUrunListesi.DataSource = urunSonuc;
    }

    grdUrunListesi.Columns["urun_kodu"].Width = 60;
    grdUrunListesi.Columns["urun_adi"].Width = 230;
    grdUrunListesi.Columns["stok_adedi"].Width = 60;
    grdUrunListesi.Columns["fiyat"].Width = 50;
    grdUrunListesi.Columns["kategori_adi"].Width = 130;
    grdUrunListesi.Columns["firma_adi"].Width = 110;
}
```

Görsel 2.61: Ürünleri DataGridView nesnesinde listeleme kodları

Satış yapılacak müşterinin seçilmesi için müşterinin telefon numarası girilip, “SEÇ” butonuna tıklandığında müşterinin ad ve soyad bilgisi alt bölümdeki TextBox nesnesinde görüntülenir. Bu işlem için “SEÇ” butonunun tıklanma olayına Görsel 2.62’deki kodlar yazılır.

```
private void btnMusteriSec_Click(object sender, EventArgs e)
{
    try
    {
        musteriler musteriler = new musteriler();
        if (mskTxtMusteriTelefon.MaskCompleted)
        {
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                musteriler = stok_Takip.musteriler.Where(m => m.telefon == mskTxtMusteriTelefon.Text).First();
                txtMusteriAdSoyad.Text = musteriler.ad_soyad;
                seciliMusteriId = musteriler.id;
            }
        }
        else
        {
            MessageBox.Show("Geçerli bir telefon numarası giriniz");
        }
    }
}
```



```

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata oluştu");
}
}

```

Görsel 2.62: Satış yapılacak müşterinin seçilmesi için gereken kodlar

Ürün listesinden seçilen ürünün satılacak ürünlere eklenmesi, eklenen ürünün listeden silinmesi için gerekli kodlar “EKLE” ve “SİL” butonlarının tıklanma olaylarına Görsel 2.63’teki gibi yazılır. Ayrıca ekleme ve silme işlemlerinden sonra “Hesapla” metodu çağrılarak ürünlerin toplam fiyatının ve genel toplamın hesaplanması sağlanır.

```

private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (grdUrunListesi.CurrentRow != null)
        {
            int urunKodu = Convert.ToInt32(grdUrunListesi.CurrentRow.Cells["urun_kodu"].Value.ToString());
            bool kontrol = true;
            foreach (DataGridViewRow row in grdSatilacakUrunler.Rows)
            { // Ürünün daha önce satış listesine eklenip eklenmediği kontrol edilir
                if (urunKodu == Convert.ToInt32(row.Cells[0].Value))
                    kontrol = false;
            }
            if (kontrol)
            {
                string urunAdi = grdUrunListesi.CurrentRow.Cells["urun_adi"].Value.ToString();
                int miktar = 1;
                decimal birimFiyat = Convert.ToDecimal(grdUrunListesi.CurrentRow.Cells["fiyat"].Value);
                decimal toplam = miktar * birimFiyat;
                grdSatilacakUrunler.Rows.Add(urunKodu, urunAdi, miktar, birimFiyat, toplam);
                Hesapla();
            }
            else
                MessageBox.Show("Seçtiğiniz ürün satılacak ürünler listesinde mevcut");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

// reference
private void btnSil_Click(object sender, EventArgs e)
{
    if (grdSatilacakUrunler.CurrentRow != null)
    {
        grdSatilacakUrunler.Rows.Remove(grdSatilacakUrunler.CurrentRow);
        Hesapla();
    }
}

```

Görsel 2.63: Satış listesine ürün ekleme ve silme



Satış listesine ürün eklenip silindiğinde ve DataGridView üzerinde satış miktarı güncellendiğinde her ürüne ait toplam fiyatın ve satışın genel toplamının hesaplanması gerekir. Bunun için öncelikle miktar değerinin doğruluğu kontrol edilir. Miktar değerinin kontrol edilmesi ve hesaplamaların yapılması için Görsel 2.64'teki kodlar yazılır.

```
private void Hesapla()
{
    if (grdSatilacakUrunler.CurrentRow != null)
    {
        int miktar = Convert.ToInt32(grdSatilacakUrunler.CurrentRow.Cells[2].Value);
        decimal birimFiyat = Convert.ToDecimal(grdSatilacakUrunler.CurrentRow.Cells[3].Value);
        decimal toplam = miktar * birimFiyat;
        grdSatilacakUrunler.CurrentRow.Cells[4].Value = toplam;
        decimal genelToplam = grdSatilacakUrunler.Rows.Cast<DataGridViewRow>()
            .Sum(t => Convert.ToDecimal(t.Cells[4].Value));
        txtGenelToplam.Text = genelToplam.ToString();
    }
    else
        txtGenelToplam.Text = "0";
}

// Referans:
private void grdSatilacakUrunler_CellEndEdit(object sender, DataGridViewCellEventArgs e)
{
    // Miktar bilgisi değiştirildiğinde toplam fiyat değerleri otomatik olarak değiştirilmelidir.
    try
    {
        string stokDegeri;
        int yeniMiktar;
        if (grdSatilacakUrunler.CurrentRow.Cells[e.ColumnIndex].Value == null)
            stokDegeri = "1";
        else
            stokDegeri = grdSatilacakUrunler.CurrentRow.Cells[e.ColumnIndex].Value.ToString();
        if (Int32.TryParse(stokDegeri, out yeniMiktar))
        {
            // Girilen stok miktarı ürünün stok adedinden fazlaysa miktar değeri ürünün stok adedine eşitlenir.
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                int urunKodu = Convert.ToInt32(grdSatilacakUrunler.CurrentRow.Cells[0].Value);
                urunler stokKontrolUrun = stok_Takip.urunler.Where(u => u.urun_kodu == urunKodu).FirstOrDefault();
                if (yeniMiktar > stokKontrolUrun.stok_adedi)
                    yeniMiktar = stokKontrolUrun.stok_adedi;
                grdSatilacakUrunler.CurrentRow.Cells[e.ColumnIndex].Value = yeniMiktar;
            }
        }
        else
        {
            // int tipinde değeri girilmezse stok değeri 1'e eşitlenir
            grdSatilacakUrunler.CurrentRow.Cells[e.ColumnIndex].Value = 1;
        }
        Hesapla();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata oluştu");
    }
}
}
```

Görsel 2.64: Miktar değerinin kontrolü ve fiyat hesaplama kodları

Satış yapılacak müşteri ve satılacak ürünler seçildikten sonra varsa satış notu eklenir ve "SATIŞ YAP" butonu tıklanarak ürünlerin satışı yapılır. Ürün satışının gerçekleşmesi için butonun tıklanma olayına Görsel 2.65'teki kodlar yazılır.



```
private void btnSatisYap_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliMusteriId == 0)
            MessageBox.Show("Satış yapılacak müşteriyi seçiniz");
        else if (grdSatilacakUrunler.RowCount < 1)
            MessageBox.Show("Satışı yapılacak ürünleri ekleyiniz");
        else
        {
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                satislar satis = new satislar();
                satis.musteri_id = seciliMusteriId;
                satis.personel_id = PersonelId;
                satis.satis_tarihi = DateTime.Now;
                satis.not = txtNot.Text;
                stok_Takip.satislar.Add(satis);
                urunler urun;
                satis_detaylari satisDetay;
                foreach (DataGridViewRow row in grdSatilacakUrunler.Rows)
                {
                    satisDetay = new satis_detaylari();
                    satisDetay.satis_id = satis.satis_id;
                    satisDetay.urun_kodu = Convert.ToInt32(row.Cells[0].Value);
                    satisDetay.miktar = Convert.ToInt32(row.Cells[2].Value);
                    satisDetay.fiyat = Convert.ToDecimal(row.Cells[3].Value);
                    stok_Takip.satis_detaylari.Add(satisDetay);
                    urun = stok_Takip.urunler.Where(u => u.urun_kodu == satisDetay.urun_kodu).FirstOrDefault();
                    urun.stok_adedi -= satisDetay.miktar; //Her ürünün stoğu satış miktarı kadar azaltılır
                }
                stok_Takip.SaveChanges();
            }
            MessageBox.Show("Satış işlemi başarılı");
            BilgileriListele();
            grdSatilacakUrunler.Rows.Clear();
            FormuTemizle();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.65: Ürün satış işleminin kodları

2.3.10. Satış Detayları

Satışlar ile ilgili detaylı bilgilerin tutulduğu “frmSatisDetaylari” formu oluşturulur (Görsel 2.66). Yapılan satışlarla ilgili detaylar bu sayfada görüntülenir.

“frmSatisDetaylari” formuna satışların listelenmesi için **grdSatislar** ve seçilen satışa ait ürünlerin listeleneceği **grdSatilanUrunler** isimli iki adet DataGridView nesnesi eklenir. Ayrıca satışa ait toplam fiyatın görüneceği **txtSatisToplam(Enabled:False)** adında bir TextBox ve **btnSatisIptal** adında bir buton eklenir.

Görsel 2.66: Ürün satış detayları form tasarımı



Satış detayları sayfasında, satışların grdSatislar nesnesinde listelenmesi için formun yüklenmesi olayına Görsel 2.67'deki kodlar yazılır.

```
int seciliSatisId = 0;
1 reference
private void frmSatisDetaylari_Load(object sender, EventArgs e)
{
    Bilgileriliste();
}
2 references
private void Bilgileriliste()
{
    using (var stok_Takip = new stok_takip_dbEntities())
    {
        var satisSonuc = (from s in stok_Takip.satislar
                          join m in stok_Takip.musteriler on s.musteri_id equals m.id
                          join p in stok_Takip.personeller on s.personel_id equals p.id
                          orderby s.satis_tarihi descending
                          select new
                          {
                              s.satis_id,
                              Musteri = m.ad_soyad,
                              PersonelAd = p.ad,
                              PersonelSoyad = p.soyad,
                              SatisTarihi = s.satis_tarihi,
                              SatisNotu = s.not
                          }).ToList();
        grdSatislar.DataSource = satisSonuc;
    }
    grdSatislar.Columns["satis_id"].Width = 60;
    grdSatislar.Columns["Musteri"].Width = 120;
    grdSatislar.Columns["PersonelAd"].Width = 120;
    grdSatislar.Columns["PersonelSoyad"].Width = 120;
    grdSatislar.Columns["SatisTarihi"].Width = 120;
    grdSatislar.ClearSelection();
}
```

Görsel 2.67: Satışların listelenmesi kodları

Satılan ürünlerin toplam fiyatını hesaplamak için Görsel 2.68'de görüldüğü gibi "Hesapla" adında bir metod oluşturulur ve bu gerekli yerlerde kullanılır.

```
private void Hesapla()
{
    try
    {
        if (grdSatilanUrunler.RowCount > 0)
        {
            decimal genelToplam = grdSatilanUrunler.Rows.Cast<DataGridViewRow>()
                .Sum(t => Convert.ToDecimal(t.Cells["Toplam"].Value));
            txtSatisToplam.Text = genelToplam.ToString();
        }
        else
        {
            txtSatisToplam.Text = "0";
        }
    }
}
```



```
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

Görsel 2.68: Toplam satış fiyatının hesaplanması

Satışlar listelendikten sonra seçilen satışa ait ürünlerin grdSatilanUrunler nesnesinde gösterilmesi için Görsel 2.69'daki kodlar yazılır.

```
private void grdSatislar_CellClick(object sender, DataGridViewCellEventArgs e)
{ // Satış seçildiğinde o satışa ait ürünlerin yan tarafta listelenmesi sağlanır.
    try
    {
        if (grdSatislar.CurrentRow!=null)
        {
            seciliSatisId = Convert.ToInt32(grdSatislar.CurrentRow.Cells[0].Value);
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                var satisSonuc = (from u in stok_Takip.urunler
                                join s in stok_Takip.satis_detaylari on u.urun_kodu equals s.urun_kodu
                                where s.satis_id == seciliSatisId
                                select new
                                {
                                    SatisId = s.satis_id,
                                    UrunKodu = s.urun_kodu,
                                    UrunAdi = u.urun_adi,
                                    Miktar = s.miktar,
                                    Fiyat = s.fiyat,
                                    Toplam = s.miktar * s.fiyat
                                }).ToList();
                grdSatilanUrunler.DataSource = satisSonuc;
            }
            grdSatilanUrunler.Columns["SatisId"].Width = 50;
            grdSatilanUrunler.Columns["UrunKodu"].Width = 55;
            grdSatilanUrunler.Columns["Miktar"].Width = 45;
            grdSatilanUrunler.Columns["Fiyat"].Width = 50;
            grdSatilanUrunler.Columns["Toplam"].Width = 55;
            Hesapla();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.69: Satışa ait ürünlerin listelenmesi kodları

“grdSatislar” nesnesinde seçilen satışın iptal edilmesi için “SATIŞ İPTAL” butonunun tıklanma olayına Görsel 2.70’teki kodlar yazılır.

```
private void btnSatisIptal_Click(object sender, EventArgs e)
{
    try
    {
        if (grdSatislar.CurrentRow!=null)
        {
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                urunler urun;
                // İlk olarak satis_detay tablosunda o satışa ait ürünler silinir.
                List<satis_detaylari> iptalSatisListesi = stok_Takip.satis_detaylari.
                    Where(d => d.satis_id == seciliSatisId).ToList();
            }
        }
    }
}
```



```

stok_Takip.satis_detaylari.RemoveRange iptalSatisListesi);
// Daha sonra seçilen satış, satışlar tablosundan silinir.
satislar iptalEdilecekSatis = stok_Takip.satislar.
    Where(s => s.satis_id == seciliSatisId).FirstOrDefault();
stok_Takip.satislar.Remove(iptalEdilecekSatis);

foreach (DataGridViewRow row in grdSatilanUrunler.Rows)
{
    // Satışı iptal edilen ürünlerin stok bilgileri satılan miktar kadar artırılır.
    int urunKodu = Convert.ToInt32(row.Cells["UrunKodu"].Value);
    int miktar = Convert.ToInt32(row.Cells["Miktar"].Value);
    urun = stok_Takip.urunler.Where(u => u.urun_kodu == urunKodu).FirstOrDefault();
    urun.stok_adedi += miktar;
}
stok_Takip.SaveChanges();
MessageBox.Show("Satış iptal edildi");
Bilgilerilistele();
grdSatilanUrunler.DataSource = null;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata Oluştı", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

Görsel 2.70: Satışın iptal edilmesi için gerekli kodlar

2.4. STOK TAKİP PROGRAMININ TEST AŞAMASI

Proje tamamlandıktan sonra uygulamanın kullanıcı ihtiyaçlarını karşılayıp karşılamadığı kontrol edilir. Bu ihtiyaçlara göre formlar ve formların işlevleri test edilir (Görsel 2.71). Form elemanlarına girilen değerlere göre alınan sonuçlar kontrol edilir. Ayrıca hatalı işlemler yapıldığında programın kapanıp kapanmadığı denetlenir.

The screenshot displays several windows of the 'Stok Takip Programı'. The 'BİLGİ GİRİŞİ' (Information Entry) window is active, showing a list of categories with columns 'kategori_id' and 'kategori_adı'. A dialog box prompts for 'Kategori adını giriniz.' (Enter category name). Other windows include 'PERSONEL İŞLEMLERİ' (Personnel Operations) with fields for name, phone, and address, and a 'Yeni Kayıt' (New Record) button. A 'Hata Oluştı' (Error Occurred) dialog box is also visible, indicating a problem with the data entry process.

Görsel 2.71: Proje sayfaalarının test edilmesi

Ürün işlemleri sayfası test edildiğinde müşterinin isteklerinden "Ürün Arama" işlevinin eksik olduğu görülür. Bunun için formun içine bir adet TextBox nesnesi eklenerek, adı **txtAraUrunAdi** olarak ayarlanır. Sonrasında bu nesnenin TextChanged olayına Görsel 2.72'deki kodlar yazılır.



```
private void txtAraUrunAdi_TextChanged(object sender, EventArgs e)
{
    var sonuc = (from u in stok_Takip.urunler
        join k in stok_Takip.kategoriler on u.kategori_id equals k.kategori_id
        join t in stok_Takip.tedarikciler on u.tedarikci_id equals t.id
        // Arama text kutusundaki değeri içeren kayıtlar listelenir
        where u.urun_adi.Contains(txtAraUrunAdi.Text)
        // Girilen değer ile başlayan kayıtların listelenmesi istenirse koşul bölümü şu şekilde oluşturulur.
        //where u.urun_adi.StartsWith(txtAraUrunAdi.Text)
        select new
        {
            u.urun_kodu,
            u.urun_adi,
            u.stok_adedi,
            u.fiyat,
            k.kategori_adi,
            t.firma_adi,
            u.aciklama
        }).ToList();

    grdUrunler.DataSource = sonuc;
}
```

Görsel 2.72: Ürün arama kodları

Ürün İşlemleri sayfasında arama fonksiyonu test edilir (Görsel 2.73).

	urun_kodu	urun_adi	stok_adedi	fiyat	kategori_adi	firma_adi	aciklama
▶	15	SMTAL VDSL Modem 300Mbps	38	700,00	Ağ ve İnternet Ürünleri	Atakum MTAL	
	16	CCPAL Kablosuz Modem	25	1250,00	Ağ ve İnternet Ürünleri	Çayırkent ÇPAL	1200Mbps Dual Band 4x...
	61	AyMTAL Kablosuz Mouse	65	55,00	Çevre Birimleri	Ayyacak MTAL	
	62	HTMAL 24 inc Monitör	40	2500,00	Çevre Birimleri	Atakum MTAL	1920x1080 FHD IPS
	63	STL Kablolu Mouse	80	45,00	Çevre Birimleri	Deneme Bilisim	
	64	27 inc Monitör	15	3299,00	Çevre Birimleri	Atakum MTAL	1920x1080 FHD

Görsel 2.73: Ürün arama işlevinin test edilmesi

Ürünün temel işlevi olan ürün satış işlemi test edilir. Satış işleminde hata olup olmadığı kontrol edilir (Görsel 2.74).

Kod	Ürün Adı	Miktar	Birim Fiyatı	Toplam	
▶	8	500 GB 3.5 Flash Bellek	4	99,00	396,00
	9	CCP Harici SSD 512GB	1	1400,00	1400,00
	15	HMTAL IP Kamera	5	550,00	2750,00
	19	Caİ8 Ethernet Kablosu 505 ...	2	700,00	1400,00
	20	Sata Kablo	20	25,00	500,00
	4	AyMTAL Harddisk	2	650,00	1300,00

Görsel 2.74: Ürün satış işleminin test edilmesi



Satışa ait detayların görüntülediği ve seçilen satışın iptal edildiği Satış Detayları sayfası test edilir (Görsel 2.75).

SATIŞ LİSTESİ							Satış Detayları					
	satıs_id	Musteri	PersonelAd	PersonelSoyad	SatısTarihi	SatısNotu	SATILAN ÜRÜNLER					
▶	17	Arzu Ko...	Merve	Öz...	10.09.2022 20:09							
	16	Fatih S....	Fatih	Te....	10.09.2022 20:03							
	15	Esat Er...	Yücel	Al...	10.09.2022 02:37	Ürünler 1 hafta sonra teslim edil...						
	14	Ali Ö.	Fatih	Te....	09.09.2022 23:52							
	13	Arzu Ko...	Fatih	Te....	09.09.2022 20:40							
	12	Kemal S.	Fatih	Te....	09.09.2022 19:07							
	11	Ali Ö.	Fatih	Te....	09.09.2022 02:20							
	10	Fatma Ay...	Fatih	Te....	08.09.2022 20:37							
	9	Fatih S....	Fatih	Te....	08.09.2022 14:22							
	8	Yasin Dem...	Yücel	Al...	08.09.2022 13:46							
	7	Ahmet Ba...	Yücel	Al...	08.09.2022 12:36							
	6	Ali Ö.	Fatih	Te....	08.09.2022 12:33							
	5	Kemal S.	Merve	Öz...	08.09.2022 00:11							
	4	Cevat Ça...	Fatih	Te....	08.09.2022 00:07							
	3	Esat Er...	Fatih	Te....	07.09.2022 23:54							

SatısId	UrunKodu	UrunAdi	Miktar	Fiyat	Toplam
▶ 17	4	AyMTAL Harddisk	1	650,00	650,00
17	19	Cat6 Ethernet Ka...	2	700,00	1400,00
17	13	HMTAL IP Kamera	6	550,00	3300,00
17	8	SB 32 GB Flash Be...	3	99,00	297,00
17	6	ata Anakart	1	2900,00	2900,00
17	15	SMTAL VDSL Mod...	1	700,00	700,00

SATIŞI İPTAL ET Satış Toplam Fiyat : 9247,00 ₺

Görsel 2.75: Satış detayları sayfasının test edilmesi

Satış Detayları sayfası test edildiğinde satış iptal edilirken kullanıcıya herhangi bir uyarı verilmediği görülür. Kullanıcıların yanlışlıkla butona tıklama ihtimaline karşı kullanıcıdan onay alınması gerekir. Bunun için “frmSatisDetaylari” formundaki “SATIŞI İPTAL ET” butonunun tıklanma olayına Görsel 2.76’da işaretli olarak gösterilen kodlar eklenir.

```
private void btnSatisIptal_Click(object sender, EventArgs e)
{
    var cevap = MessageBox.Show("Seçilen satışa iptal etmek istediğinizden emin misiniz?", "SATIŞ İPTAL İŞLEMİ", MessageBoxButtons.YesNo);
    if (cevap == DialogResult.No)
        return; // Kullanıcı "HAYIR" butonuna tıklarsa işlem yapmadan metottan çıkılır.

    try
    {
        if (grdSatislar.CurrentRow != null)
        {
            using (var stok_Takip = new stok_takip_dbEntities())
            {
                urunler urun;
                // İlk olarak satis_detay tablosunda o satışa ait ürünler silinir.
                List<satis_detaylari> iptalSatisListesi = stok_Takip.satis_detaylari.
                    Where(d => d.satis_id == seciliSatisId).ToList();
                stok_Takip.satis_detaylari.RemoveRange(iptalSatisListesi);
                // Daha sonra seçilen satış, satislar tablosundan silinir.
                satislar iptalEdilecekSatis = stok_Takip.satislar.
                    Where(s => s.satis_id == seciliSatisId).FirstOrDefault();
                stok_Takip.satislar.Remove(iptalEdilecekSatis);

                foreach (DataGridViewRow row in grdSatilanurunler.Rows)
                {
                    // Satışı iptal edilen ürünlerin stok bilgileri satılan miktar kadar artırılır.
                    int urunkodu = Convert.ToInt32(row.Cells["urunkodu"].Value);
                    int miktar = Convert.ToInt32(row.Cells["Miktar"].Value);
                    urun = stok_Takip.urunler.Where(u => u.urun_kodu == urunkodu).FirstOrDefault();
                    urun.stok_adedi += miktar;
                }
                stok_Takip.SaveChanges();
                MessageBox.Show("Satış iptal edildi");
                BilgileriListele();
                grdSatilanUrunler.DataSource = null;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hata Olustu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Görsel 2.76: Satış iptal işleminde kullanıcı onayı alınmasının kodlaması



SIRA SİZDE

Test aşamasında fark edildiği üzere projenin diğer sayfalarında (Tedarikçi, Ürün, Müşteri) silme işlemlerinde uyarı mesajı çıkmaz. Bu durum, hatalara sebep olabilir. Bu nedenle silme işlemlerinde uyarı mesajı gönderip, test ve bakım adımı olarak kullanıcı onayı almayı sağlayınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Tedarikçi işlemleri sayfasının kod bölümünü açtı.		
2. Tedarikçi silme işlemi için uyarı mesajı kodlarını yazdı.		
3. Uyarı mesajında “Hayır” butonu tıklanmışsa işlem yapılmadan silme meto- dundan çıkılmasını sağladı.		
4. Ürün işlemleri sayfasının kod bölümünü açtı.		
5. Ürün silme işlemi için uyarı mesajı kodlarını yazdı.		
6. Uyarı mesajında “Hayır” butonu tıklanmışsa işlem yapılmadan silme meto- dundan çıkılmasını sağladı.		
7. Müşteri işlemleri sayfasının kod bölümünü açtı.		
8. Müşteri silme işlemi için uyarı mesajı kodlarını yazdı.		
9. Uyarı mesajında “Hayır” butonu tıklanmışsa işlem yapılmadan silme meto- dundan çıkılmasını sağladı.		

Kullanıcı kabul testinde, uygulamada kayıt ve güncelleme işlemleri yapılırken hata oluştuğunda müşteri, gösterilen hata mesajının kullanıcılar için anlamsız olduğunu bildirir ve bunun düzeltilmesini talep eder. Örneğin başka bir müşteriye ait olan telefon numarası ile yeni müşteri kaydı yapılmak istenirse Görsel 2.77’deki gibi bir hata mesajı ile karşılaşılır.

BİLGİ GİRİŞİ

Ad Soyad : Murat Yi... Firma Adı : Adres : Sürmene / TRABZON

Telefon : (777) 562-5896 Mail : murat@ornek.com

Kayıt Hatası

An error occurred while updating the entries. See the inner exception for details.

mail **adres**

ahmetb@deneme.com	ilkadım/SAMSUN
arzu@mail.com	Kadıköy / İSTANBUL
deneme mail	deneme adres
fatih@xyz.com	Korgan / ORDU
	Silvan/DIYARBAKIR
	Gölcük / KOCAELİ
	Pendik/İSTANBUL
mahmut@deneme	

Görsel 2.77: Müşteri işlemleri sayfası hata mesajı ekranı



Bu sorunun sebebi, sistemde yakalanan hatanın doğrudan kullanıcıya gösterilmesidir. Bu hata, geliştiriciler için anlamlı olsa da kullanıcılar için bir anlam ifade etmez. Sorunu çözmek için Müşteri İşlemleri sayfasındaki ekleme işleminde gösterilen hata mesajı anlamlı bir hâle getirilir (Görsel 2.78).

```
private void btnEkle_Click(object sender, EventArgs e)
{
    try
    {
        if (seciliMusteriID == 0)
        {
            if (BilgiGirisKontrol())
            {
                stok_Takip = new stok_takip_dbEntities();
                musteriler musteriler = new musteriler();
                musteriler.firma_adi = txtFirmaAdi.Text;
                musteriler.ad_soyad = txtAdSoyad.Text;
                musteriler.telefon = mskTelefon.Text;
                musteriler.mail = txtMail.Text;
                musteriler.adres = txtAdres.Text;
                stok_Takip.musteriler.Add(musteriler);
                stok_Takip.SaveChanges();
                Musterilerilistele();
                FormuTemizle();
            }
        }
        else
        {
            MessageBox.Show("Yeni müşteri eklemek için 'Yeni Kayıt' butonuna tıklayınız");
        }
    }
    catch
    {
        MessageBox.Show("Kayıt işleminde hata oluştu. Lütfen girilen bilgileri kontrol ediniz.");
    }
}
```

Görsel 2.78: Müşteri İşlemleri sayfası hata mesajı düzeltme kodları



SIRA SİZDE

Uygulamanın diğer sayfalarındaki (Kategori, Müşteri, Ürün ve Tedarikçi) ekleme ve güncelleme işlemlerindeki hata mesajlarını kullanıcılar için anlamlı hâle getiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Kategori ekleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
2. Kategori güncelleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
3. Müşteri ekleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
4. Müşteri güncelleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		



Değerlendirme Ölçütleri	Evet	Hayır
5. Ürün ekleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
6. Ürün güncelleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
7. Tedarikçi ekleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		
8. Tedarikçi güncelleme işlemindeki hata mesajını kullanıcılar için anlamlı hâle getirdi.		

Kullanıcı kabul testi yapıldıktan sonra projenin yayımlanma aşamasına geçilir. “Microsoft Visual Studio Installer Projects” eklentisi indirildikten sonra Add>New Project>Setup Project adımları kullanılarak yeni bir kurulum projesi eklenir (Görsel 2.79).

Configure your new project

Setup Project

Project name

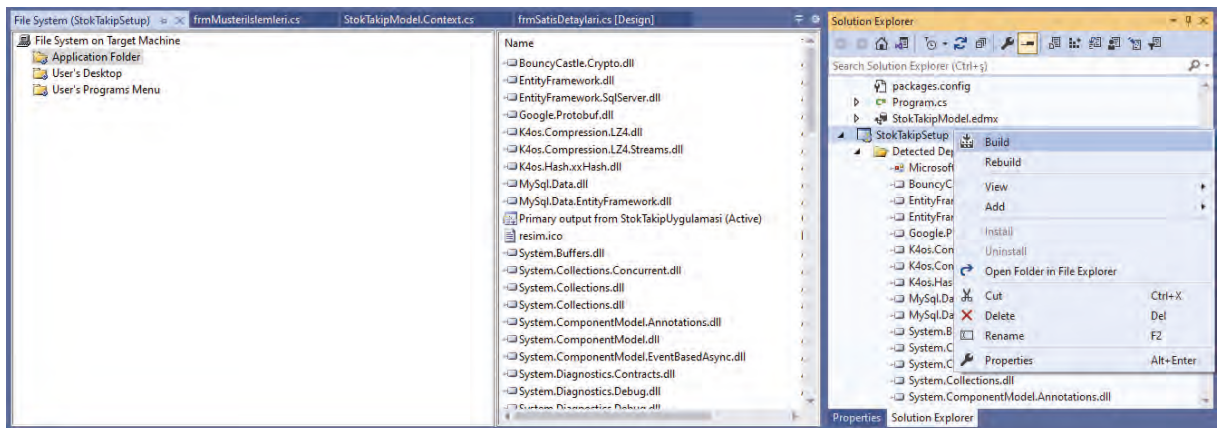
Location

Back

Create

Görsel 2.79: Uygulamaya setup projesi ekleme

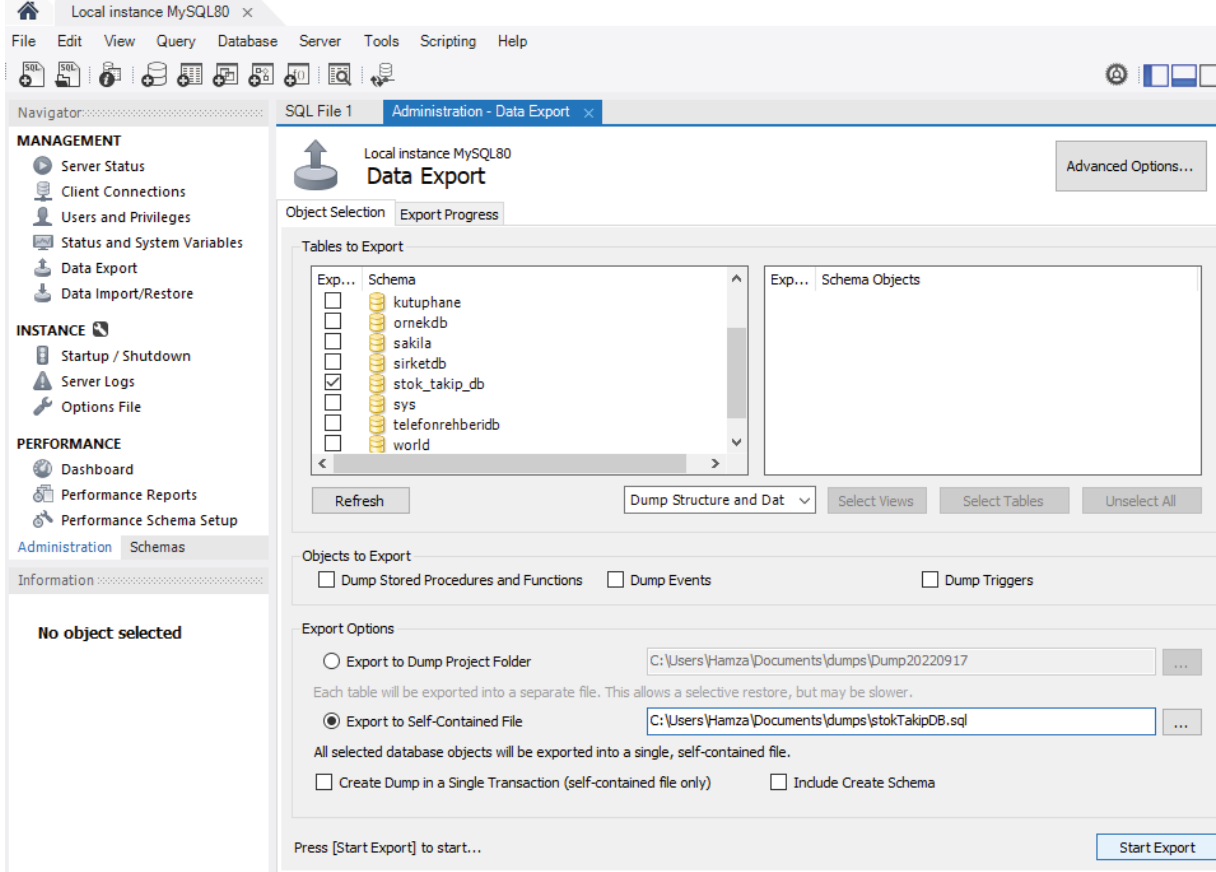
Kurulum projesi ile ilgili gerekli ayarlamalar yapıldıktan sonra projeye sağ tıklanarak “Build” seçeneği seçilir (Görsel 2.80). Derleme işleminden sonra belirtilen klasörde Debug klasörünün içinde kurulum dosyalarının olduğu görülür.



Görsel 2.80: Setup projesinin derlenmesi



Kurulumun başka bilgisayara yapılabilmesi için veri tabanının o bilgisayarın sunucusuna taşınması gerekir. “Data Export” işlemi ile veri tabanının “script” dosyası oluşturularak veri tabanının yedeği alınır (Görsel 2.81). Bu dosya kurulum yapılacak bilgisayardaki sunucuya “Data Import/Restore” seçeneği ile dâhil edilir.



Görsel 2.81: Data Export işlemiyle veri tabanını yedekleme

Veri tabanı dosyası, kurulum yapılacak bilgisayarın sunucusuna eklendikten sonra “Stok Takip Uygulaması” kurulumu yapılır. Kurulum klasöründe yer alan “StokTakipUygulaması.exe.config” dosyasındaki bağlantı ayarları, kurulum yapılan bilgisayara göre güncellenir ve uygulama çalıştırılır (Görsel 2.82).

```
<connectionStrings><add name="stok_takip_dbEntities"
  connectionString="metadata=res://*/StokTakipModel.csdl|res://*/StokTakipModel.ssdl|res://*/StokTakipModel.msl;
  provider=MySql.Data.MySqlClient; provider
  connection string="server=localhost;user id=root;password=Meb*BT.1453;database=stok_takip_db;
  persistsecurityinfo=True";" providerName="System.Data.EntityClient"/>
</connectionStrings>
```

Görsel 2.82: Güncellenecek veri tabanı bağlantı cümlesi

2.5. STOK TAKİP PROGRAMI BAKIM AŞAMASI

Proje bir süre kullanıldıktan sonra ürünler arttıkça ürün stoklarının takibinin zorlaştığı görülür. Projede bununla ilgili bir geliştirme yapılması gerekir. Stok miktarı 20'nin altında olan ürünlerin listelendiği “Stoku Azalan Ürünler” sayfası oluşturularak bu sorun çözülebilir. Bunun için ilk olarak “frmStokuAzalanUrunler” adında bir form oluşturulur (Görsel 2.83).



Görsel 2.83: Stoku azalan ürünler form tasarımı

Bu formdaki nesneler; **txtAraUrunAdi**, **nmrStokAdedi**, **btnStokGuncelle** ve **grdStokuAzalanUrunler** olarak isimlendirilir. DataGridView nesnesi ile ilgili bütün satırı seçme ve tek satır seçilmesine izin verme gibi gerekli ayarlamalar yapılır. Ayrıca bu nesne üzerinde ekleme, silme ve düzenleme işlemlerine izin verilmez.

Sayfada ilk olarak stok miktarı 20'nin altında olan ürünler listelenir. Ayrıca arama kutusundaki her değişiklikte ürün arama işleminin gerçekleştirilmesi için TextBox nesnesinin içeriğinin değişmesi olayında listeleme metodu çağrılır (Görsel 2.84).

```
private void frmStokuAzalanUrunler_Load(object sender, EventArgs e)
{
    Bilgileriliste();
}
3 references
private void Bilgileriliste()
{
    using (stok_takip_dbEntities stok_Takip = new stok_takip_dbEntities())
    {
        var sonuc = (from u in stok_Takip.urunler
                     where (u.stok_adedi < 20 && u.urun_adi.Contains(txtAraUrunAdi.Text))
                     select new
                     {
                         u.urun_kodu,
                         u.urun_adi,
                         u.fiyat,
                         u.kategoriler.kategori_adi,
                         u.tedarikciler.firma_adi,
                         u.aciklama,
                         u.stok_adedi
                     }).ToList();

        grdStokuAzalanUrunler.DataSource = sonuc;
        grdStokuAzalanUrunler.Columns["urun_kodu"].Width = 60;
        grdStokuAzalanUrunler.Columns["stok_adedi"].Width = 70;
        grdStokuAzalanUrunler.Columns["urun_adi"].Width = 200;
        grdStokuAzalanUrunler.ClearSelection();
    }
}
1 reference
private void txtAraUrunAdi_TextChanged(object sender, EventArgs e)
{
    Bilgileriliste();
}
```

Görsel 2.84: Stoku azalan ürünleri listeleme ve arama işlemi kodları



DataGridView nesnesi üzerinde seçilen ürünün stok miktarı NumericUpDown nesnesine aktarılır. Yeni stok değeri girilerek GÜNCELLE butonuna tıklanır ve stok değeri güncellenir. Bu işlemin gerçekleştirilmesi için gereken kodlar Görsel 2.85'teki gibi yazılır. Ürünler tekrar listelenir ve stok değeri 20 ve 20'den fazla girilen ürünün, stoku azalan ürünler listesinden kaldırıldığı görülür.

```
private void grdStokuAzalanUrunler_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (grdStokuAzalanUrunler.CurrentRow != null)
        {
            nmrStokAdedi.Value = Convert.ToInt32(grdStokuAzalanUrunler.CurrentRow.Cells["stok_adedi"].Value);
        }
    }
    catch
    {
        MessageBox.Show("Hata oluştu");
    }
}

1 reference
private void btnStokGuncelle_Click(object sender, EventArgs e)
{
    try
    {
        using (stok_takip_dbEntities stok_Takip = new stok_takip_dbEntities())
        {
            if (grdStokuAzalanUrunler.SelectedRows.Count > 0)
            {
                int urunKodu = Convert.ToInt32(grdStokuAzalanUrunler.CurrentRow.Cells["urun_kodu"].Value);
                urunler urun = stok_Takip.urunler.Where(u => u.urun_kodu == urunKodu).FirstOrDefault();
                urun.stok_adedi = Convert.ToInt32(nmrStokAdedi.Value);
                stok_Takip.SaveChanges();
                BilgileriListele();
                nmrStokAdedi.ResetText();
            }
            else
            {
                MessageBox.Show("Stok miktarını güncelleyeceğiniz ürünü seçiniz");
            }
        }
    }
    catch
    {
        MessageBox.Show("Stok güncelleme işleminde hata oluştu");
    }
}
```

Görsel 2.85: Stok miktarını güncelleme



SIRA SİZDE

Uygulama kullanılırken satışlar arttıkça kullanıcının bu satışlar içinden istediği kaydı bulması giderek zorlaşacaktır. Bu nedenle "frmSatisDetaylari" formunda listelenen satışlarda müşteri adı, personel adı ve seçilen tarihe göre arama işleminin yapılmasını sağlayınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. "frmSatisDetaylari" formunu açtı.		
2. Forma GroupBox nesnesi ile arama bölümü ekledi.		



Değerlendirme Ölçütleri	Evet	Hayır
3. Aranacak müşteri adı için TextBox nesnesi ekledi.		
4. Aranacak personel adı için TextBox nesnesi ekledi.		
5. Aranacak satış tarihi için DateTimePicker nesnesi ekledi.		
6. “frmSatisDetaylari” formunun kod bölümünde müşteri adı, personel adı ve satış tarihine göre arama işlemi yapacak bir metot oluşturdu.		
7. Nesnelerin her birinin değişiklik yapılması olayında arama metodunu çağırdı.		
8. Arama işlemini test etti.		

Projede bakım işlemi yapıldıktan sonra test işlemi yapılır ve hatalar varsa düzeltilir. Daha sonra proje tekrar setup hâline getirilir ve dağıtılır.

2.6. STOK TAKİP PROGRAMI RAPOR AŞAMASI

Stok takip programı rapor aşaması; proje adı, giriş, araçlar ve yöntemler, sonuç, ürün eksiklikleri, sözlük, kaynakça içeren ayrıntılı bir raporun hazırlanmasını kapsar.

PROJE ADI: STOK TAKİP UYGULAMASI

1. GİRİŞ

Projede bilişim teknolojileri ürünlerinin satışını yapan firmanın, ürün satışı ve stok takibinin yapılacağı bir masaüstü uygulamasını geliştirmek amaçlanmıştır. Kullanıcılar bu uygulama ile ürün satışı ve yapılan satışı iptal etme işlemlerini gerçekleştirebilir. Satışı yapılan ürünlerin stok miktarı, satış adedi kadar azaltılır. Aynı şekilde iptal edilen satışıdaki ürünlerin stok miktarı da iptal edilen satışıdaki ürün miktarı kadar artırılır.

Ayrıca programda ürünler, tedarikçiler, müşteriler ve personellere ait bilgiler de saklanır. Bunlarla ilgili ekleme, silme, güncelleme ve listeleme işlemleri de yapılabilir.

Uygulamada “Yönetici” ve “Personel” olmak üzere iki adet yetki bulunur. Yönetici yetkisine sahip kullanıcılar, uygulamadaki bütün işlemleri yapabilirken personel yetkisine sahip kullanıcı, personel işlemleri dışındaki bütün işlemleri yapabilir.

Uygulama, kullanıcının tercihi nedeniyle internet bağlantısı gerektirmeyen bir form uygulaması olarak geliştirilmiştir.

2. ARAÇLAR VE YÖNTEMLER

Proje; Giriş, Anasayfa, Tedarikçi İşlemleri, Ürün İşlemleri, Stoku Azalan Ürünler, Satış Detayları, Müşteri İşlemleri ve Personel İşlemleri olmak üzere 8 sayfadan oluşur.

Proje; ücretsiz bir IDE olan “Visual Studio Community” programı üzerinde, .NET Framework ortamında, C# yazılım dili ile geliştirilmiştir. Veri tabanı yönetim sistemi olarak yine ücretsiz olan MySQL Server kullanılmıştır.



NOT

Araçlar ve Yöntemler bölümünde, gereksinimler ve tasarım adımı oluşturulan diyagramlar da yer alabilir. Tekrarı önlemek için gereksinimler ve diyagramlar bu bölüme eklenmemiştir.



Projede veri tabanı işlemleri için Entity Framework ORM (Object–Relational Mapping) aracı kullanılmıştır. Database First yöntemi ile VTYS programında oluşturulan veri tabanı tablolarına karşılık gelen sınıflar otomatik olarak oluşturulmuştur. Daha sonra DbContext sınıfının özelliklerini kalıtım yoluyla alan “stok_takip_dbEntities” sınıfı kullanılarak veri tabanı işlemleri gerçekleştirilmiştir.

Kullanıcılara ait parolalar MD5 şifreleme algoritması ile şifrelenerek parolaların veri tabanına kaydedilmesi sağlanmıştır. Kullanıcı girişi yapılırken de girilen parola yine MD5 ile şifrelenerek veri tabanında kayıtlı olan şifrelenmiş parola değerleri ile eşleştirilir ve buna göre kullanıcının sisteme girişi yapılması sağlanır.

Program çalıştırılıp kullanıcı girişi yapıldıktan sonra programın anasayfasında listelenen ürünler seçilerek satış listesine eklenebilir veya silinebilir (Görsel 2.86). Sistemde kayıtlı olan müşteri, telefon numarası bilgisi ile seçildikten sonra “SATIŞ YAP” butonu ile satış işlemini gerçekleştirebilir.

The screenshot shows the 'STOK TAKIP PROGRAMI' application. On the left, there is a list of products with columns: ürün_kod, ürün_adi, stok_adet, fiyat, kategori_adi, firma_adi, and aciklama. The list includes various items like SMTAL VDSL Modem, CCPAL Kabloşuz Modem, and different types of RAM and monitors. On the right, there is a 'SATIŞ İŞLEMİ' form. It includes fields for 'Müşteri Telefon : (777) 562-5896', 'Müşteri : Esat Er...', and a table for 'SATIŞ YAPILACAK ÜRÜNLER'. The table has columns: Kod, Ürün Adı, Miktar, BirimFiyat, and Toplam. It lists items like 'ata D0R4 8GB RAM', 'HMTAL 24 inc Monitör', and 'SB 32 GB Flash Bellek'. At the bottom, there is a 'SATIŞ YAP' button and a 'Genel Toplam : 8846,00' label.

Görsel 2.86: Programın anasayfasında listelenen ürünlerin seçilerek satış listesine eklenmesi veya silinmesi

Satış detayları menüsüne tıklanarak açılan “SATIŞ DETAYLARI” sayfasında listelenen satışlar içinden seçilen satışa ait detaylar yan tarafta görüntülenir. “SATIŞI İPTAL ET” butonu tıklanarak seçilen satış işlemi iptal edilebilir (Görsel 2.87).

The screenshot shows the 'İmSatışDetayları' application. On the left, there is a list of sales with columns: satis_id, Müsteri, PersonelAd, PersonelSoyad, SatisTarihi, and SatisNotu. The list includes sales made by Arzu Ko..., Cevat Ç..., and Fatih S... On the right, there is a 'Satış Detayları' form. It includes a table for 'SATILAN ÜRÜNLER' with columns: Satisid, ÜrünKodu, ÜrünAdı, Miktar, Fiyat, and Toplam. It lists items like 'AyMTAL Anakart', 'HMTAL Ekran Kartı', and 'HMTAL 24 inc Monitör'. At the bottom, there is a 'SATIŞI İPTAL ET' button and a 'Satış Toplam Fiyat : 199703,00' label.

Görsel 2.87: Seçilen satışa ait detayların görüntülenmesi ve satış işleminin iptal edilmesi



Uygulamada tedarikçi firma, ürün, müşteri ve satış yapacak olan personel bilgileri de saklanır. Bunlarla ilgili silme, güncelleme ve listeleme işlemleri yapılabilir. Bu işlemleri yapmak için anasayfada bulunan menü elemanlarına tıklanarak ilgili sayfaya geçiş yapılmalıdır. Örneğin ürünler ile ilgili ekleme, silme, güncelleme veya arama işlemi için “ÜRÜN İŞLEMLERİ” menüsüne tıklanarak işlem yapılacak sayfanın açılması sağlanır (Görsel 2.88). Daha sonra bu sayfada ürünlerle ilgili istenen işlemler yapılabilir.

ÜRÜN İŞLEMLERİ

BİLGİ GİRİŞİ

Ürün Adı : Kategori : Tedarikçi :

Açıklama : Stok Adedi : Fiyat :

Ürün Arama

Ürün Adı :

urun_kodu	urun_adi	stok_adedi	fiyat	kategori_adi	firma_adi	aciklama
15	SMTAL VDSL Modem 30...	38	700,00	Ağ ve İnternet Ürünleri	Atakum MTAL	
16	CCPAL Kablosuz Modem	21	1250,00	Ağ ve İnternet Ürünleri	Çaykent ÇPAL	1200Mbps Dual Band 4xG...
19	Cat6 Ethernet Kablosu 30...	15	700,00	Ağ ve İnternet Ürünleri	Şahinbey MTAL	1000 Mbps
20	Sata Kablo	1017	25,00	Bilgisayar Aksesuarları	Deneme Bilisim	Data
3	AyMTAL Anakart	32	1475,50	Bilgisayar Bileşenleri	Ayvack MTAL	DDR4 3200mhz USB 3.1
4	AyMTAL Harddisk	59	650,00	Bilgisayar Bileşenleri	Ayvack MTAL	500GB 7200Rpm SATA
6	ata Anakart	20	2900,00	Bilgisayar Bileşenleri	Atakum MTAL	DDR4 4400 mhz USB 3.2 ...
7	ata DDR4 8GB RAM	21	950,00	Bilgisayar Bileşenleri	Atakum MTAL	3600 mhz
10	HMTAL İşlemci	98	2400,00	Bilgisayar Bileşenleri	Haydarpaşa MTAL	3.8 ghz 8 Core
11	HMTAL Ekran Kartı	18	10800,00	Bilgisayar Bileşenleri	Haydarpaşa MTAL	8gb DDR5 256 bit
12	SB DDR4 16GB RAM	25	1550,00	Bilgisayar Bileşenleri	Şahinbey MTAL	3200mhz
14	AyMTAL 550W Power Su...	42	1100,00	Bilgisayar Bileşenleri	Ayvack MTAL	80 Plus Bronze
17	ata İşlemci	25	2000,00	Bilgisayar Bileşenleri	Atakum MTAL	3.4 ghz 6 Core 12 Thread
18	ata 1TB Nvme M.2 SSD	35	1700,00	Bilgisayar Bileşenleri	Atakum MTAL	3200MB/S-3000MB/S
61	AyMTAL Kablosuz Mouse	63	55,00	Çevre Birimleri	Ayvack MTAL	

Görsel 2.88: Ürün işlemleri

3. SONUÇ

Geliştirilen uygulama test edildiğinde uygulamanın şu işlemleri yerine getirdiği görülür:

- Kullanıcı girişi yapılması
- Ürünler ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması
- Ürün kategorileri ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması
- Tedarikçiler ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması
- Müşteriler ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması
- Personeller ile ilgili ekleme, silme, güncelleme ve listeleme işlemlerinin yapılması (Bu işlem yalnızca “Yönetici” yetkisine sahip kullanıcılar tarafından yapılabilir.)
- Ürün satışı yapılması
- Yapılan satışın iptal edilmesi
- Ürünlerin stok takibinin yapılması

Geliştirilen yazılım ürününün işlevleri test edildiğinde müşteri ihtiyaçlarını karşıladığı tespit edilmiştir. Ürün, kurulum dosyası hâline getirilerek dağıtılabilir.

Ürünün Eksiklikleri

- Ürünün arayüz tasarımları geliştirilmelidir. Uygulamada arayüzler tasarlanırken form tasarımlarına görsel açıdan zenginlik katmak için geliştirilen eklentiler kullanılabilir.
- Veri tabanı işlemlerinde Entity Framework ORM aracının Code First yöntemi tercih edilebilirdi.
- Ürün satışı yapıldıktan sonra satışın güncellenebilmesi gerekirdi.
- Ürünler ve satışlar ile ilgili daha detaylı filtreleme işlemleri yapılabilirdi.



SÖZLÜK

IDE (Integrated Development Environment): Tümleşik geliştirme ortamı olarak Türkçeye çevrilen IDE, geliştiricilerin ihtiyaç duyduğu araçları ve eklentileri içinde barındırarak uygulama geliştirmeyi kolaylaştıran yazılımlardır. Ayrıca diğer geliştirme ortamlarında kullanıcıların manuel yapması gereken birçok işlemi otomatik şekilde yaparak yazılımın geliştirilme süresini de kısaltır.

Visual Studio Community: Konsol uygulamaları, masaüstü uygulamaları, web siteleri, web uygulamaları ve mobil uygulamalar geliştirmek için gerekli olan yazılım geliştirme araçlarını içinde barındıran ücretsiz bir tümleşik geliştirme ortamıdır.

MySQL: Açık kaynak kodlu, ücretsiz bir veri tabanı yönetim sistemi yazılımıdır. En çok kullanılan VTYS'lerden biridir.

.NET Framework: Microsoft tarafından geliştirilen bir uygulama geliştirme platformudur. Geliştirilen uygulamaların çalıştırılacağı ortamlarda da bu yazılımın yüklü olması gerekir.

Entity Framework: Veri tabanında yer alan tablo ve alanların nesne olarak kullanılmasını sağlayan bir ORM aracıdır. Bu araç sayesinde veri tabanı ve SQL kullanılmadan veri tabanı işlemleri yapılabilir.



ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1. Veri tabanındaki veriler olarak adlandırılan nesnelerin içinde saklanır.
2. Veri tabanı tablolarında yalnızca bir kaydı temsil eden benzersiz alanlar olarak belirlenir.
3. Buton tıklandığında çalışması istenen kodlar, nesnenin olayına yazılır.
4. Form uygulamalarında veri tabanı tablolarındaki verilerin tablo şeklinde gösterimi için form elemanı kullanılır.
5. Kullanıcılara mesaj göstermek için sınıfı kullanılır.

B) Aşağıdaki soruların doğru seçeneğini işaretleyiniz.

6. Sayfa yüklendiğinde çalışması istenen kodlar, form nesnesinin aşağıdaki hangi olayına yazılmalıdır?

- A) Click
- B) Form_Closed
- C) KeyUp
- D) Load
- E) Move

7. Bir form elemanını pasif duruma getirmek için aşağıdaki hangi özelliğin "False" olarak ayarlanması gerekir?

- A) Enabled
- B) Name
- C) Size
- D) Text
- E) Visible

8. DataGridView nesnesinde seçim yapılırken satırın tamamının seçilmesini sağlamak için "SelectionMode" özelliği aşağıdakilerden hangisi olarak belirlenmelidir?

- A) CellSelect
- B) ColumnHeaderSelect
- C) FullColumnSelect
- D) FullRowSelect
- E) RowHeaderSelect

9. Aşağıdakilerden hangisi belirlenen formata göre kullanıcının bilgi girişi yapmasını sağlayan form elemanıdır?

- A) Button
- B) MaskedTextBox
- C) NumericUpDown
- D) RichTextBox
- E) TextBox

10. TextBox nesnesinin içeriğini temizlemek için aşağıdaki metotlardan hangisi kullanılır?

- A) Clear
- B) Copy
- C) Count
- D) Delete
- E) Remove



3. ÖĞRENME BİRİMİ

MOBİL UYGULAMA PROJESİ HAZIRLAMA



KONULAR

- 3.1. MOBİL UYGULAMANIN PLANLAMASI
- 3.2. MOBİL UYGULAMANIN TASARIMI
- 3.3. MOBİL UYGULAMANIN KODLAMASI
- 3.4. MOBİL UYGULAMANIN TEST VE BAKIMI
- 3.5. MOBİL UYGULAMANIN RAPORU

NELER ÖĞRENECEKSİNİZ?

- Mobil uygulama için tasarım ekranları
- Mobil uygulamalar için platformlar
- Mobil uygulama işletim sistemleri
- Mobil uygulama tasarlanırken planlama
- Mobil uygulamalar için kullanılan veri tabanları
- Mobil uygulama geliştirirken risk analizi oluşturma

TEMEL KAVRAMLAR

Bulut, cloud server, mobil, uygulama geliştirme

HAZIRLIK ÇALIŞMALARI

1. Kullandığınız mobil uygulamalar nelerdir? Bu uygulamaları geliştirmek sizin elinizde olsaydı neleri değiştirmek isterdiniz?
2. Mobil işletim sistemleri türlerinden bildiklerinizi arkadaşlarınızla paylaşınız.

3.1. MOBİL UYGULAMANIN PLANLAMASI

Mobil uygulama geliştirme projeleri, diğer proje geliştirme ortamlarına nazaran biraz daha farklıdır. Bunun en büyük nedeni, sürekli kendini güncelleyen ve yenileyen işletim sisteminin olmasıdır. Yeni gelişen ve güncellenen her işletim sistemine göre mobil uygulama geliştirme teknolojilerinin altyapısı da değişir. Örneğin Android 10 sürümü kullanılırken geliştirilen mobil uygulama teknolojileri ile 11 veya 12 sürümünde kullanılan teknolojiler birbirinden farklılık gösterebilir. Bunun dışında farklı mobil platformlar, farklı türde işletim sistemi kullanılabilir (IOS, Android vs.) ve üretilen uygulamalar “Cross” platformda yazılmamışsa birbirlerinde çalışmaz. Mobil uygulamalarda bir diğer sorun da mobil ekranın (TV, tablet, saat, telefon vb.) çok çeşitli ve farklı boyutlarda olmasıdır. Yazılan uygulamalar bu nedenle en baştan çok detaylı planlanmalı ve planlamalar doğrultusunda oluşturulmalıdır.

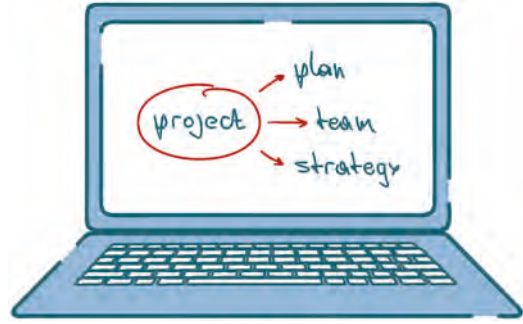


NOT

Mobil uygulama projesi geliştirme bölümünde uygulama olarak verilen proje üzerinden planlama, tasarım, kodlama, test ve bakım, rapor adımları tek tek incelenecektir.

Örneğin; her ülke için gezilmeye değer yerleri, uygulama içine kaydettikçe anasayfasında bir liste şeklinde ortaya çıkaran ve bu kayıtları Firebase altyapısı olarak bulut serverda tutan bir mobil uygulama hazırlanacaktır.

Uygulamayı sadece kodlama açısından ele almamak için uygulama öncesinde yapılacak iş ve işlemler vardır (Görsel 3.1). Bu adımların sırasıyla ele alınması proje geliştirme sürecinde kolaylık, etkileşim, zaman ve maliyet tasarrufu sağlar ve sonrasında yaşanabilecek sorunların da önüne geçer.



Görsel 3.1: Planlama işlemleri

• Yapılmak İstenen Çalışma

- » Her ülke için gezilecek yerleri kaydettikçe listeleyen ve bu kayıtları Firebase altyapısında bulut serverda saklayan bir mobil uygulama tasarımı

• Yapılacak Uygulama İçin Müşterinin İstekleri

- » Müşteri, Android işletim sisteminde API-23 seviyesi ve üzerinde sorunsuz çalışabilecek bir mobil uygulama istemektedir.
- » Mobil uygulamanın çoklu aygıt desteği sunmasını istemektedir (tablet, telefon vb.).
- » Uygulamanın çoklu dil desteği sunmasını beklemektedir (İngilizce, Türkçe).
- » Bulut veri tabanının kullanılmasını istemektedir. Anlık veri kaydedip veri çekebilme özelliğini beklemektedir.
- » Açılan anasayfada önce bir logo, sonrasında da ana menünün gelmesini istemektedir. Müşterinin isteği üzerine liste içinde kayıtlı olan yerler sıralanmalı, gezilecek yere tıklandığında da yeni bir sayfada, gezilecek yere ait görsel, ülke bilgisi, bölge bilgisi, yerin ismi, tarihçesi ve hakkında ifadeleri yer almalıdır. Menü yardımı ile yeniden anasayfaya dönmelidir. Ayrıca dil değişimi burada bulunan menüde de sunulmalıdır.
- » Ana menünün sağ üst köşesinde bir menü yer almalı ve menüde “Giriş Yap, Yer Ekle, Yer Sil, Dil Seçimi” menüleri bulunmalıdır. Giriş yapıldıktan sonra da “Giriş Yap” menüsü yerine giriş yapanın kullanıcı adı görünmelidir. Aynı zamanda menü altına da bir “Çıkış” seçeneği eklenmelidir.
- » Ayarlar menüsü adı altında daha sonradan bir seçenek eklenip “içeriği artırılabilir” olarak anlaşma sağlanması beklenebilir.
- » Bakım ve güncelleme işlemleri için en az 5 yıllık anlaşma sağlanması istenmektedir.



NOT

Müşterinin istekleri içinde sondan ikinci maddede görüldüğü üzere müşteriler bazı isteklerini net olarak belirtmeyip sonrasında da güncelleyebilecek düzeyde bırakabilir. Böyle durumlara hazırlıklı olarak tasarım ve kodlama aşamalarını bu düzeyde esnek hazırlamak, yazılımcıya fayda sağlar.

Bu noktadan sonra müşterinin istekleri ile yazılımının kafasındaki planın birbiriyle bağdaşması gerekir. Uçuk ve gerçekleştirilemeyecek müşteri istekleri kabul edilirse iş sonunda istenen ürün ortaya çıkmayabilir. Dolayısıyla hazırlanan ürün hem elde kalabilir hem de yazılı yapılan anlaşmalar neticesinde tazminat ödenebilir. Bu nedenle müşteri isteklerini yerine getirecek gereksinim analizleri, takım ile birlikte oluşturulmalıdır (Görsel 3.2).



Görsel 3.2: Gereksinim analizi

• Müşteri İstekleri Ölçüsünde İhtiyaç Duyulan Gereksinimler

- » Android Studio son sürüm yazılım programı
- » Emülatör (Dönüştürücü Yazılım Programı), bilgisayar üzerinde kullanılacaksa yüksek bir işlemci ve RAM kapasitesine sahip bilgisayar
- » Google Firebase kullanabilmek için bir Google hesabı
- » Bulut veri tabanını kullanmak, etkileşime girebilmek ve Google dokümantasyonlarını inceleyebilmek için internet bağlantısı
- » Projede görsellerin geliştirilip düzenlenebileceği bir fotoğraf düzenleme programı (Photoshop vb.)
- » Görselleri mobil uygulamaya yönelik farklı boy ve oranlarda paketleyebilmek için bir yazılım programı (Adobe XD vb.)

• Proje Takım Üyelerini Belirlemek ve Takım Üyelerine İş Paylaşımının Yapılması

- » Proje tek kişi tarafından yürütüleceği için takım bölümlerine ayrılmaz. Tek takım üyesinin hangi işleri, hangi zaman diliminde yapacağı ise “Proje Zaman Takvimi”nde belirtilir.

• Fizibilite Çalışmaları Listesi

- » Android işletim sisteminde çalışması düşünülmektedir.
- » Firebase veri tabanı kullanılması planlanmaktadır. Firebase Firestore kısmından tablolar ayarlanır.
- » Java dilinde geliştirilmesi amaçlanmaktadır.
- » Müşteri isteği olarak açılan anasayfada liste yer alacaktır. Bu listede kayıtlı ülkeler seçildikten sonra o ülkeye ait gezilecek yerlerin listesi anasayfa yenilenerek gösterilmelidir. İlgili yere tıklandığında yeni sayfa açılmalı ve o sayfada ilgili yere ait görseller, ülke bilgisi, bölge bilgisi, yerin ismi, tarihçesi ve hakkında ifadeleri yer almalıdır. Bir buton ile yeniden anasayfaya geçilmeli veya Geri butonu ile seçilen bölgelerin listesine yeniden dönülmelidir. Her sayfa bölümünün sağ üst köşesinde bir menü yer almalıdır ve menüde “Giriş Yap, Yer Ekle, Yer Sil” menüleri bulunmalıdır.
- » Çoklu dil desteği eklenmelidir. Eklenen her yer için İngilizce ve Türkçe girişler zorunlu tutulmalıdır.
- » Logo, yeniden tasarlanmadan, müşterinin logosu alınarak kullanılacak tasarım programında Android Studio’ya göre paketleme yapılmalıdır.



• Proje Maliyet Hesabı

Projelerde gereksinim analizi hazırlandıktan sonra gerekli ekipman, uygulama ve ürünler ortaya çıkar (Görsel 3.3). Ortaya çıkan bu ürünlerin maliyetlerinin hesabı, müşteriye gerekli tutarın doğru hesaplanabilmesi ve yazılımcının kâr ve zarar dengesinin kurulabilmesi için projeye başlanmadan önce yapılmalıdır. Doğru biçimde oluşturulmayan maliyet hesapları daha sonrasında yazılımcı için külfet oluşturabilir ve satıcının emeğinin karşılığı olan maddi tutarı alamamasına yol açabilir.



Görsel 3.3: Maliyet hesabı

- » Firebase kullanımı, küçük projelerde ücretsiz olmasına karşın veri çekme ve kullanıcı sınırı bulunur. Bu nedenle müşterinin potansiyel veri çekme bandı henüz tahmin edilemez. Ücretsiz seçeneği ile başlanır ve müşteri için sonrasında “kullan öde” mantığı devreye girer. Ödemeleri Google servisine müşterinin kendisi yapar.
- » Android Studio için herhangi bir maliyet olmaz.
- » Geliştirici markete konulması için “25\$” maliyeti vardır.
- » Telifli alınacak görseller için ek tablo sonrasında oluşturulur.

• Proje Takvimi

- » Geliştirilecek uygulama için takvim oluşturulması hem müşterinin hem de takımın yer alacak diğer üyelerin bu takvime bağlı kalmasını sağlar (Görsel 3.4). Proje takvimi, aynı zamanda işin ne zaman biteceğini de belirttiği için müşteriye bu noktada güvence verir. Hazırlanacak mobil uygulama projesinin takvim planlaması da Tablo 3.1’de verilmiştir.



Görsel 3.4: Proje takvimi

Tablo 3.1: Proje Takvim Tablosu

Proje Bölümü	Mart – 2023 (10-15)	Mart – 2023 (15-30)	Nisan – 2023 (01-30)	Nisan – 2023 (01-15)
Uygulama sayfalarının tasarımlarının ön görünümünün planlanması	x			
Veri tablolarının hazırlanması ve ilişkilendirilmesi		x		
Tasarımların oluşturulması ve kodlama yapısının hazırlanması		x	x	
Kodlama işlemleri		x	x	
Grafiklerin çizimi veya temin edilmesi			x	x
Deneme ve analiz çalışmaları			x	x



- **Risk Analizi**

Geliştirilecek mobil uygulama için ortaya çıkarılan risk analizi Tablo 3.2’de verilmiştir. Bu tabloda yer alan risklere yönelik önlem planları da tek tek açıklanır.

Tablo 3.2: Risk Planı

RİSK	OLASILIK	ETKİ		
		DÜŞÜK	ORTA	YÜKSEK
Müşteri taleplerinin değişmesi	Orta			
Ekip üyelerinin sistematik ve iş birlikçi çalışmaması	Orta			
İnternet arızaları veya sunucu problemleri	Düşük			
Mobil uygulamanın tasarlanacağı ortamın çıkaracağı sorunlar	Yüksek			
Ödeme alma problemleri	Düşük			
Müşteriler tarafından beğenilmeme ve geri çevrilme	Düşük			
Müşteri isteklerinin, yazılacak uygulamaya aktarılmada eksik kalması	Düşük			

- **Risk Yönetim Kararları**

- » **Müşteri Taleplerinin Değişmesi**

Müşteri taleplerinin değişmesi, her projede karşılaşılabilecek ortak bir sorundur. Sorun ortaya çıkmadan önce bu sorunu aşmak mümkündür. Bu nedenle planlama aşamasında müşteriden isteklerini belirten bir liste alınmalı ve bu liste yazılı olarak da müşteriye onaylatılmalıdır. Bu sayede süreç içinde değişebilecek her istek için ekstra durum ortaya çıkarılıp ayrı bir biçimde (maddi ve manevi) değerlendirmeye alınabilir.

- » **Ekip Üyelerinin Sistematik ve İş Birlikçi Çalışmaması**

Bu proje tek başına değil de ekiple hazırlanacak olsa ekip üyelerinin en az %40-%50’sini daha önce birlikte çalışmış ve birbirlerine uyum sağlamış bir kadrodan seçmek uygundur. Bu sayede iş birliği ile çalışan bir grup oluşturulurken diğer ekip üyelerinin onlara ayak uydurması beklenir ve süreç içinde bu durum gözlemlenebilir. Gözlem esnasında da az sayıda kişiye odaklanılması sebebiyle süreç yönetimi daha rahat geçer.

- » **İnternet Arızaları veya Sunucu Problemleri**

Proje içinde bulut servis olarak kullanılacak Firebase sistemi, günümüz teknolojileri arasında oldukça güvenli ve revaçta olan bir sunucu servisi türüdür. Sunucu servisinin altyapısının Google tarafından hazırlanması ekstra bir prestij sağlarken aynı zamanda ona bir güvenlik yamasını da ekler. Günümüzde internet bağlantı problemleri de oldukça azalmıştır ve problem yaşanması ihtimalinde kısa sürede sorunlar çözüme kavuşturulur.

- » **Mobil Uygulamanın Tasarlanacağı Ortamın Çıkaracağı Sorunlar**

Oluşturulacak proje Android Studio üzerinden yazılacaktır. Firebase Data Server, Android Studio ile tam donanımlı çalışacak biçimde düzenlenmiştir. Dolayısı ile aralarındaki bağlantıda bir problem yaşanması düşünülmez fakat Android Studio’nun yüksek performanslı bilgisayarlarda çalışması bir sorun yaratabilir. Bunun nedeni, Android Studio emülatörünün fazla işlemci ve RAM kapasitesi kullanmasıdır. Bu nedenle emülatörle ortaya çıkabilecek hataları en aza indirmek amacıyla her ihtimale karşı bir Android telefon ve Android tablet ile bağlantı sağlanır. Geliştirilen projenin gerçek zamanlı deneme işlemleri telefon ve tablet ile gerçekleştirilir.



» Ödeme Alma Problemleri

Ödeme noktasında problem yaşamamak için müşteri ile işi alan kişi veya kurum arasında ya-pılan bir yazılı anlaşma olmalıdır. Ayrıca işe başlanmadan önce mutlaka belirli bir tutar ödeme alınıp bu durum yazılı hâle getirilmeli, müşteride de işi alan kişi veya kurumda da bu evrakın yedeği tutulmalıdır. İş alan kişi veya kurum tarafından mağdur edilen bir müşteri bu belge sayesinde hakkını arayarak tazminat talep edebilmeli, aynı şekilde müşteri tarafından mağdur edilebilecek kişi veya kurum da hakkını arayabilmelidir. Müşterinin, işten haksız yere vazgeçmesi ihtimaline önlem olarak ekip üyelerinin veya kendinin mağdur olmaması amacı ile ön ödeme olarak bir kısım tutar alınmalı ayrıca haksız fesihte bu tutarın yanacağı belirtilmelidir.

» Müşteriler Tarafından Beğenilmeme ve Geri Çevrilme

Yapılan proje, belirli aralıklarla müşteriye sunulmalı ve gidişat hakkında müşterinin fikri alınmalıdır. Beğenilmeyen duruma karşı anında çözüm üretilmesi, tüm projenin bitiminden sonra sorunu çözmeye çalışmaktan daha kolaydır.

» Müşteri İsteklerinin Yazılacak Uygulamaya Aktarılmasında Eksik Kalması

Android Studio ve altyapısının kullanıldığı Java dili, zamanla geliştirilip yılların birikimi ile oldukça popüler hâle gelen bir dildir. Böylece esnek yapısı ile müşterinin tüm istekleri uygulamalara yansıtılabilir.



SIRA SİZDE

Bir müşterinin "Resim Galerisi Uygulaması" doğrultusundaki talebini karşılayacak projenin planlama, maliyet hesabı, proje takvimi ve risk analizi basamaklarını oluşturunuz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Resim Galerisi Uygulaması için yapılmak istenen çalışmayı belirledi.		
2. Resim Galerisi Uygulaması için müşterinin isteklerini belirledi.		
3. Resim Galerisi Uygulaması için müşterinin istekleri ölçüsünde ihtiyaç duyulan gereksinimleri belirledi.		
4. Resim Galerisi Uygulaması için takım üyelerini belirleyerek sorumluluk ve iş paylaşımı yaptı.		
5. Resim Galerisi Uygulaması için maliyet hesaplaması hazırladı.		
6. Resim Galerisi Uygulaması için proje takvimi hazırladı.		
7. Resim Galerisi Uygulaması için risk analizi hazırladı.		



3.2. MOBİL UYGULAMANIN TASARIMI

Projelerde tasarım denildiğinde akla sadece kullanıcının karşısına çıkacak görseller, şekiller ve modüller gelmemelidir. Bu durum, tasarımın basamaklarından yalnızca biridir. Tasarım genellikle hem fiziksel hem de mantıksal tasarım olarak ortaya çıkar. Fiziksel tasarım kullanıcıya hitap eden bölüm olurken mantıksal tasarım da yazılımcının bu projeyi nasıl geliştireceğinin altyapısını belirler. Bunun içeriğinde sınıf yapıları, veri tabanı tasarımı vb. yer alır. Bu nedenle tasarım, proje kodlanmadan önce genel hatları ile ortaya dökülmelidir (Görsel 3.5).



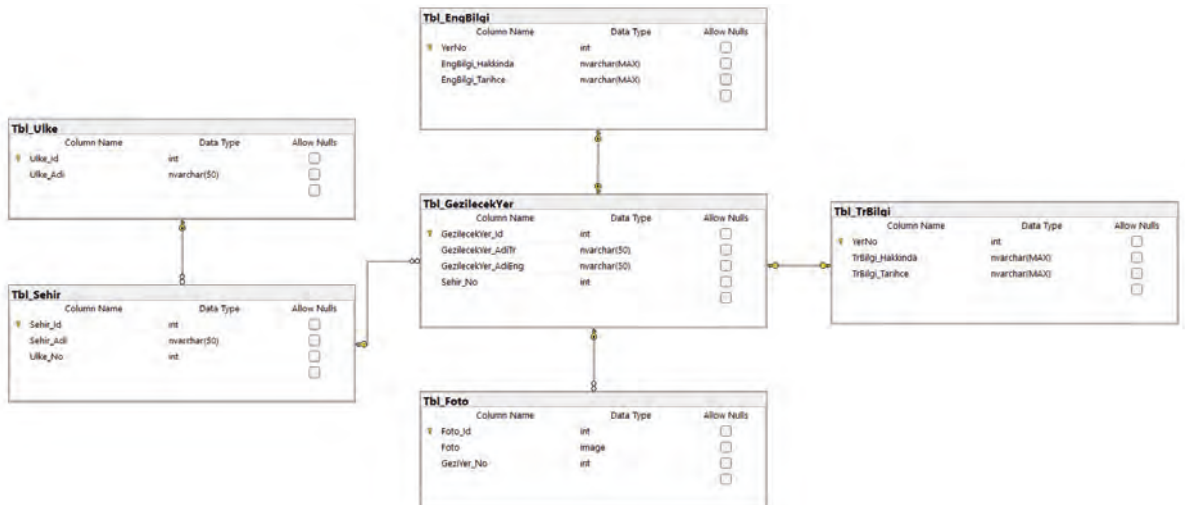
Görsel 3.5: Tasarım

3.2.1. Mantıksal Tasarım

Uygulama, Firebase üzerinde bulut serverda (sörvır) tutulacaktır. Firebase veri tabanı yönetim sistemi diğer veri tabanı dosyalarından biraz daha farklıdır. Tablolardan oluşmaz. Veriler JSON objeleri şeklinde tutulur. Bu her ne kadar veri ekleme ve silmeyi kolaylaştırır da çok büyük veri tabanı işlemlerinde kullanışsız sonuçlar da doğurabilir. Bu nedenle hem uygulama “Sql veri tabanında” hazırlandığında tablolarının ve ilişkilerinin nasıl olacağı hem de Firebase cloud serverda hazırlandığında yapının nasıl olacağı verilmiştir.

3.2.1.1. Sql Olarak İlişkisel Tablo Diyagramı

Sql olarak düzenlenen veri tabanı dosyaları tablo mantığında oluşturulur ve ilişkisel veri tabanı denilen bir düzen hâlinde tasarlanır. Mobil uygulamaya ait veri tabanı Sql üzerinden yapılırsa Görsel 3.6’daki gibi bir tasarım oluşturmak mantıklıdır.



Görsel 3.6: Sql diyagram yapısı



3.2.1.2. Firebase Olarak Hazırlanan Yapı

Veriler; Firebase olarak “Authentication”, “Storage” ve “Firestore database” şeklinde üç parçada tutulacaktır. Authentication şeklinde tutulacak veriler kullanıcıların kayıtlarını ele alır. Bu bölümde yer kaydetmeye yetkili kişilerin kayıtları tutulur ve kayıtlı kişiler, uygulamaya giriş yapmaları hâlinde yeni bir yetkili kullanıcı da oluşturabilirler. Storage kısmında, gezilecek yerlere ait fotoğraflar tutulur ve bu fotoğraflara ait id’ler Firebase tarafından benzersiz şekilde otomatik olarak eklenir. Firestore database bölümünde ise text türünde oluşturulan kayıtlar tutulur ve fotoğraftan alınan id de burada text olarak işlenir. Her ne kadar buradaki yapı Android Studio üzerinde kodla oluşturulacak olsa da Firestore database bölümünde yer alacak yapı Görsel 3.7’de gösterilmiştir. Bu yapı el ile oluşturulmayıp Android Studio üzerinden kodlandığı zaman otomatik olarak gelir. Kodlama bölümünde de bu yapı tekrar ele alınır.

Field	Type	Value
GeziYeri_Id	number	
GeziYeri_Ulke	string	
GeziYeri_Sehir	string	
GeziYeri_TrAdi	string	
GeziYeri_EngAdi	string	
GeziYeri_TrHakki	string	
GeziYeri_EngHak	string	
GeziYeri_TrTarihc	string	
GeziYeriEngTarih	string	
Foto	string	

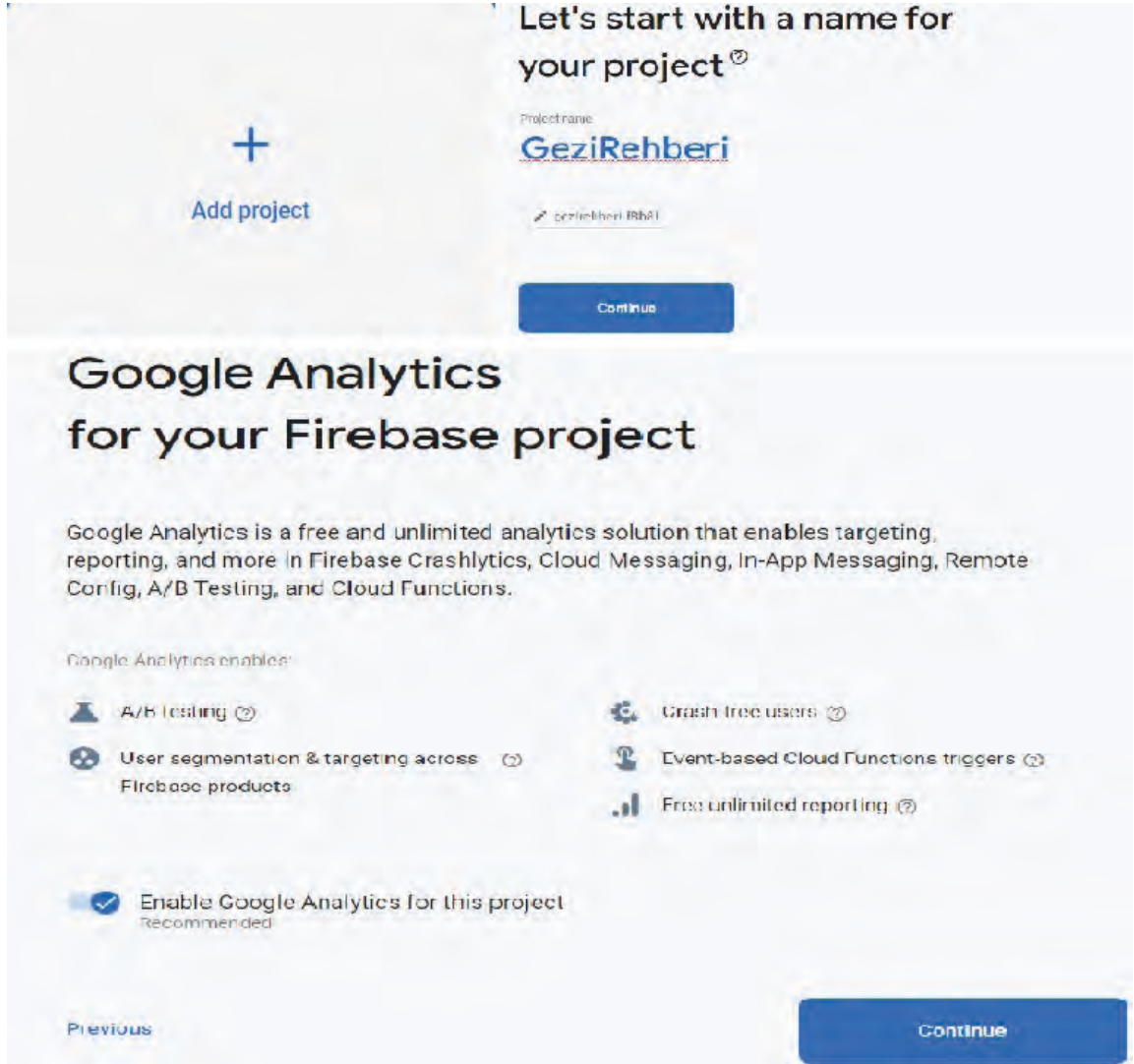
+ Add field

Görsel 3.7: Firestore database yapısı



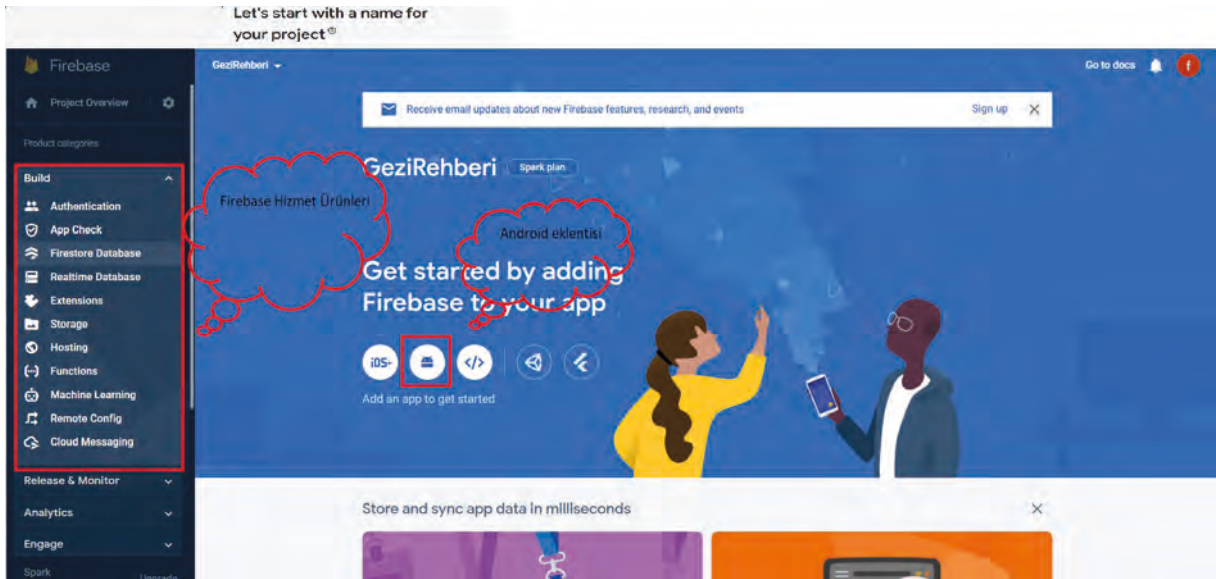
Uygulamanın Firebase'e bağlanması için şu adımlar izlenir:

- Bu yapıyı kullanabilmek için öncelikle Android Studio üzerinden yeni bir proje açılır.
- Açılan proje Empty Activity olarak başlatılıp programın yazılım dili de Java seçilir.
- En düşük API seviyesi belirlenerek (23 ve üzeri olması şartı ile) uygulama oluşturma tamamlanır.
- <https://console.firebase.google.com> linkine gidilerek istenen bir Google hesabı üzerinden Firebase dosyası oluşturulur. Bu işlem için Görsel 3.8'de olduğu gibi "Add Project" seçeneğine tıklanır. Gelen ekranda projenin ismi sorulur. Projeye belirlenen isim girilir. Bu proje için GeziRehberi ismi girilmiştir.
- "Continue" seçeneğine tıklandıktan sonra Firebase tarafından Google analiz servislerinin kullanılıp kullanılmayacağı sorulur. Tercihe uygun şekilde "Enable Google Analytics for this project" seçeneği işaretlenerek "Continue" seçeneğine tıklanır.
- Sistem tarafından Firebase dosyası hazırlanır. Hazırlama işlemi bittiğinde "Continue" seçeneğine tıklanır ve Firebase yönetim paneline ulaşılır.



Görsel 3.8: Firebase açılışı

- Firebase anasayfası Görsel 3.9'da olduğu gibi görünür. Sol bölümde bulunan "Build" sekmesi altında Google Firebase'e ait hizmetler yer alır. Bu bölümdeki "Authentication", "Firestore Database" ve "Storage" hizmetleri projede kullanılır. Yetkililerin kayıtları Authentication kısmında, gezilecek yerlerin bilgileri Firestore Database kısmında ve yerlerin fotoğrafları da Storage bölümünde tutulur. Öncelikle bu Firebase dosyasının projeye entegrasyonu gerekir. Bunun için Görsel 3.9'da işaretli olan Android bağlantı butonuna tıklanmalıdır.



Görsel 3.9: Firebase anasayfa

- Tıklama işleminin ardından Görsel 3.10'da olduğu gibi bir görüntü ile karşılaşılır. Burada bulunan “Android package name” bölümüne Android projesinin paket adı yazılır. Açılan projede paket adının “com.adinizSoyadiniz.uygulamaAdi” şeklinde bir formülle olması kullanım açısından doğrudur. Oluşturulan Android Studio dosyasından “MainActivity.java” dosyası içinde en üst kısımda paketin adı görülebilir. Paket adı bire bir kopyalanarak Android package name bölümüne yapıştırılır. “App nickname” yazan yere opsiyonel olarak bir isim girilir. App nickname seçeneğinin projeye aynı isimde olması zorunlu değildir. “Register app” butonuna tıklanır.

[illegible]

Görsel 3.10: Firebase senkronizasyon

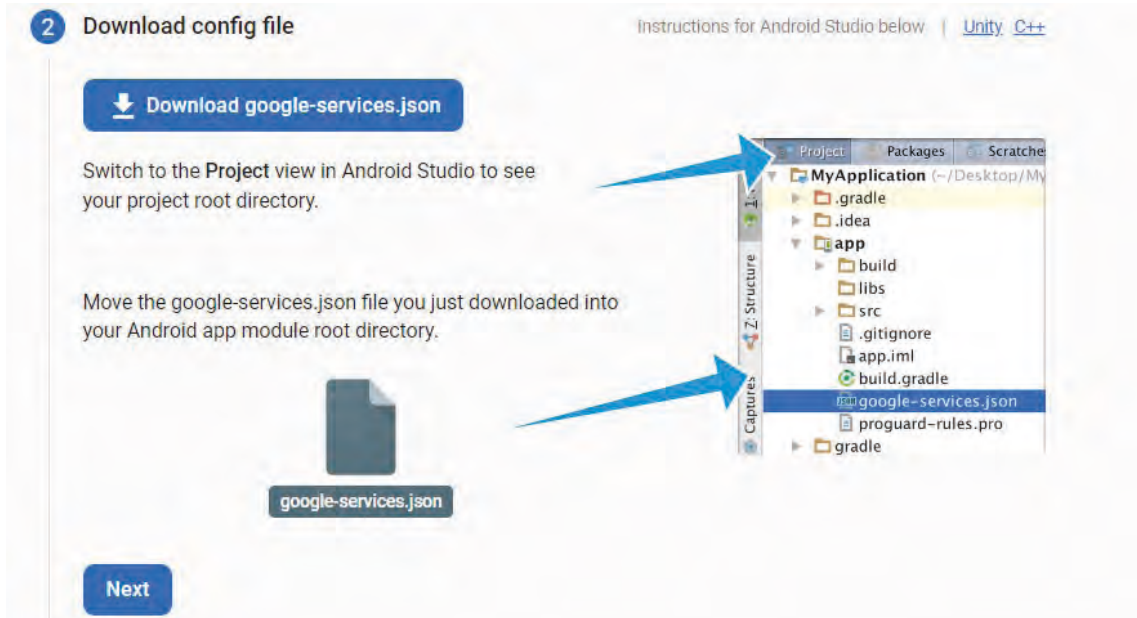


- Firebase dosya paketinin oluşturulması beklenir. Oluşturma işlemi tamamlandıktan sonra Görşel 3.11’de görüldüğü gibi **Download config file** bölümüne geçilir. **Download google-service.json** butonuna tıklanır ve dosya indirilir. Bu dosya, Görşel 3.11’de belirtildiği gibi Android projesi Project şeklinde gösterilerek, ilgili **app** klasörünün altına hiçbir değişiklik yapılmadan atılır. Ardından Next butonuna tıklanır.



NOT

Dosya içinde çıkan bilgiler üçüncü kişilerle paylaşılmalıdır. Bu bilgiler, üçüncü kişilerde olduğu takdirde veri tabanına sızmak çok kolaydır. Bu nedenle dosya, Android proje içinde sadece paket hâlinde kalmalıdır.



Görşel 3.11: Dowload config file

- Görşel 3.12’de ekrana gelen bölümde Android projesinin Gradle bölümüne eklenmesi gereken eklentiler gösterilir. **Project-level build.gradle** denilen bölüm, Android Studio projesindeki Project düzeyindeki gradle dosyasını; **App-level build.gradle** ise Android Studio projesindeki Module düzeyindeki gradle dosyasını temsil eder.



Görşel 3.12: Gradle dosyası eklentileri



- Bu bölümde yer alan kodlar ve eklentiler projedeki gradle ile karşılaştırılmalı ve olmayanlar tek tek eklenmelidir. Gradle dosyaları Project ve Module olarak ikiye ayrıldığı için ayrı ayrı dosyalar karşılaştırılmalıdır. Tüm ekleme işlemleri bittikten sonra Görsel 3.13'te görülen uyarıya dikkat edilmelidir.

```
// TODO: Add the dependencies for Firebase products you want to use
// When using the BoM, don't specify versions in Firebase dependencies
// https://firebase.google.com/docs/android/setup#available-libraries
```

Görsel 3.13: Diğer Firebase hizmetlerinin kullanımı uyarısı



NOT

Bu uyarıda, kullanılacak Firebase hizmetlerinin ilgili linke gidilerek projeye eklenti şeklinde dâhil edilmesi istenir. Tüm Firebase dosyaları durup dururken projeye dâhil edilirse çok büyük yer kaplayacağı için bu yol seçilir.

- İlgili linke gidilerek kütüphane içindeki Authentication, Cloud Storage ve Cloud Firestore karşısında yer alan ekler Module düzeyindeki Gradle dosyasına eklenir (Görsel 3.14).

```
dependencies {

    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:2.0.4'
    implementation 'com.google.firebase:firebase-analytics:21.1.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

    implementation platform('com.google.firebase:firebase-bom:30.3.2')
    implementation 'com.google.firebase:firebase-auth'
    implementation 'com.google.firebase:firebase-storage'
    implementation 'com.google.firebase:firebase-firestore'
}
```

Görsel 3.14: Module düzeyi gradle



NOT

Ekleme işlemlerinde ekleme sonrası **Sync Now** tıklanarak senkronizasyon yapılmalıdır. Senkronizasyonda hata alınırsa Android Studio üzerinde yer alan Tools menüsü altındaki Firebase asistanıyla da işlemler yapılabilir.

- Eklemler başarıyla bitirdikten sonra projenin Firebase bağlantısı tamamlanır ve Firebase tasarımı ana hâli ile oluşturulur. Android Studio üzerinden Firebase içine yapılacak ekleme ve silme işlemlerinde Firebase yönetim panelinde de kontrol sağlanır.



NOT

Gradle içinde **allproject** ekleme bölümünde hata alınması durumunda **settings.gradle** dosyası içine girilir ve **dependencyResolutionManagement** içeriği Görsel 3.15'teki gibi güncellenir. Ayrıca Firebase ürünlerinin eklentilerini eklerken hata oluşması durumunda Tools menüsü altındaki Firebase seçeneğinden otomatik olarak bu paketlerin güncel versiyonları eklenir.

```
dependencyResolutionManagement { DependencyResolutionManagement it ->
    repositoriesMode.set(RepositoriesMode.PREFER_SETTINGS)
    repositories { RepositoryHandler it ->
        google()
        mavenCentral()
        jcenter()
        maven { url 'https://jitpack.io' }
    }
}
```

Görsel 3.15: settings.gradle içeriği

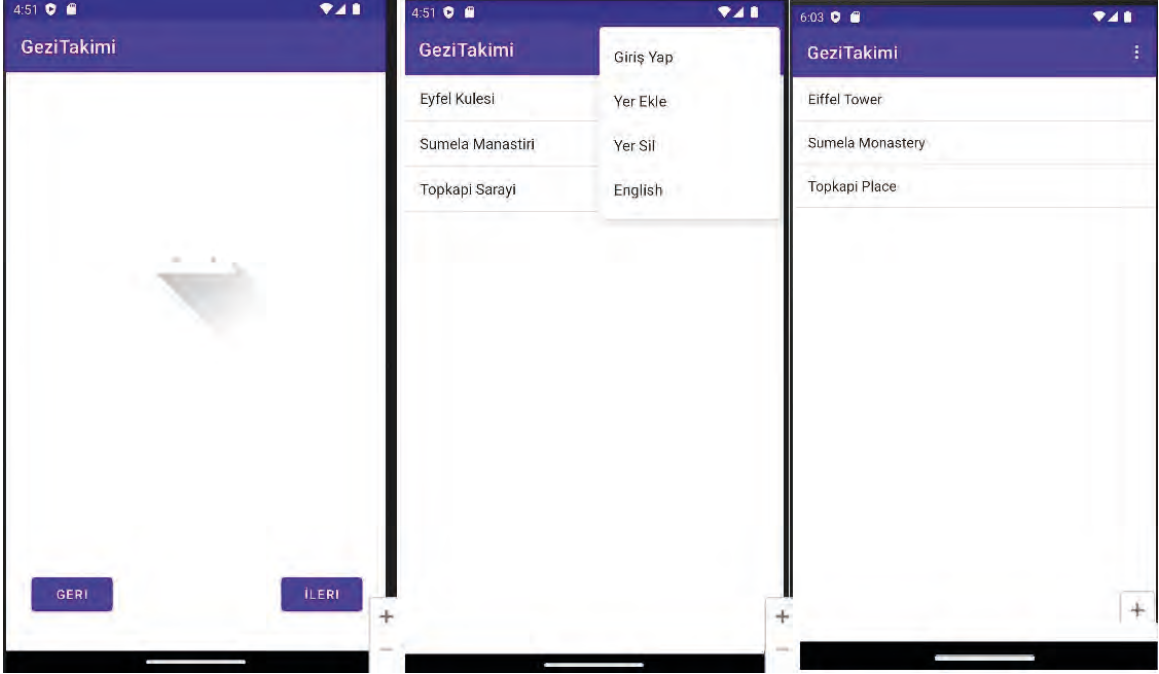
- Fiziksel tasarımda yapılması istenen adımlar; aslında kâğıt üzerinde eskiz şeklinde çalışması yapılan, tam olarak bitmemiş, üzerinde de değişiklik öngörülerek yapılan çalışmalar olmalıdır. Her insanın da eskizi farklı olabileceği için hazırlanan projede programın son hâli, gerçek fiziksel tasarımlar üzerinden görüntülenir. Uygulama başladığında ekrana gelen görüntü, Görsel 3.16'daki gibi logonun görüldüğü ve 5 sn. beklediği bir görüntüden oluşturulur.



Görsel 3.16: Uygulama başlangıç logosu

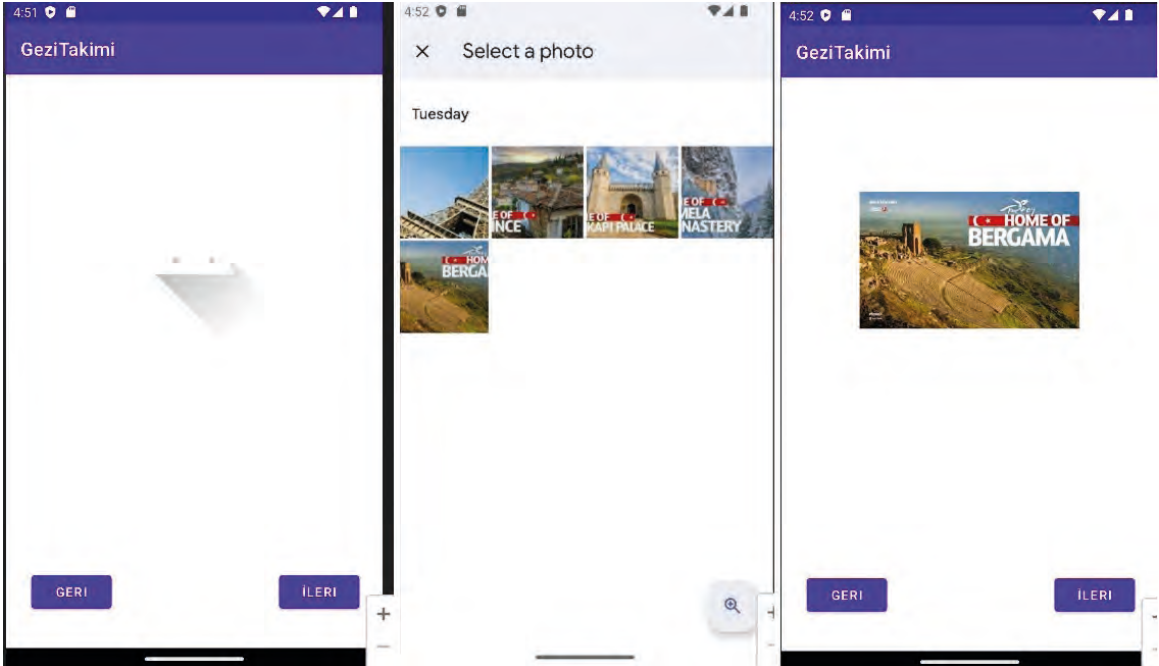


- 5 sn. bekleme sonucunda ekrana anasayfa gelir (Görsel 3.17). Anasayfada da Firebase bulut depolamada tutulan verilerdeki yer isimleri yer alır. Default olarak Türkçe yazılan isimler istenirse menüde yer alan English seçeneğine tıklanarak İngilizceye dönüştürülebilir. “Yer Ekle” seçeneğine tıklandığında da yeniden bir kayıt girilmesi için bir activity açılır.



Görsel 3.17: Uygulama ana ekranı

- Yer ekle butonuna tıklandığında Görsel 3.18’de bulunan ilk görselde olduğu gibi bir activity ekrana gelir. Burada imageView’e tıklanınca galeriye ilk defa erişim sağlanacaksa kullanıcıdan erişim izni istenir. Erişim izni verildiği takdirde galeriye Görsel 3.18’deki gibi erişim sağlanır. Galeriden bir fotoğraf seçilir ve imageView’e gelir. İleri butonuna tıklandığında da aynı activity içinde başka bir fragment açılıp bu kez Türkçe bilgilerin girilmesi istenir.



Görsel 3.18: Yeni kayıt fotoğraf ekleme



- İleri butonuna tıklandığında bilgiler EditText ve Multi EditTextler ile sorulur ve bilgilerin Türkçe olarak girilmesi istenir. Görsel 3.19'da Türkçe giriş bilgileri ekranı verilmiştir.

Görsel 3.19: Türkçe bilgi girişleri

- Tekrar İleri butonuna tıklandığında bu kez de bilgilerin İngilizce olarak girilmesi istenir. Bu sayede farklı dil seçeneği kullanımının önü açılır. Görsel 3.20'de olduğu gibi bilgiler girildikten sonra Kaydet seçeneğine tıklanırsa bir "AlertDialog" ile "Kaydetmek istiyor musunuz?" diye sorulur. Bu soruya "Evet" yanıtı verilirse bilgiler Storage ve Firebase Firestore'a kaydedilir. Bilgilerin kaydedilip kaydedilmediği Toast mesajı ile haber verilip ana menüye dönülür ve yeni kayıt da listelenir.

Görsel 3.20: İngilizce bilgi girişi



SIRA SİZDE

Planlama, maliyet analizi, risk planı ve proje takvimi hazırlanan “Resim Galerisi Uygulaması” için mantıksal olarak Sql ilişkisel tablo diyagramı hazırlayınız ve uygulamanın fiziksel tasarımı için örnek ekran görüntülerini şema şeklinde defterinizde oluşturunuz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

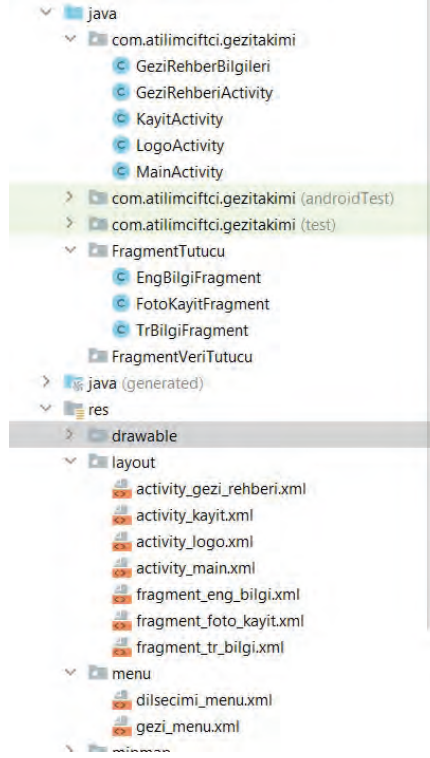
Değerlendirme Ölçütleri	Evet	Hayır
1. Resim Galerisi Uygulaması için Sql tabloları diyagramı oluşturdu.		
2. Sql tablolarını ilişkilendirdi.		
3. Resim Galerisi Uygulaması için fiziksel tasarım şemaları oluşturdu.		

3.3. MOBİL UYGULAMANIN KODLAMASI

Gezi Takımı projesinde yer alan kodlama için öncelikle bu kodların en etkin bir biçimde nasıl yazılacağına karar verilmelidir. Yazılan ve çalışan her kod, doğru kod anlamına gelmez. Bir kodun doğru olabilmesi için mümkün olduğunca tekrarsız, hızlı, pratik ve anlaşılabilir yazılması gerekir.

Kodlama bölümü hiçbir zaman sadece yazılardan oluşmaz. Kodlama bölümünde sürekli olarak fiziksel tasarımla ilgili iş ve işlemlerin geliştirilmesi yapılır çünkü tasarım ve kodlama birbirinden ayrılmaz bir bütündür.

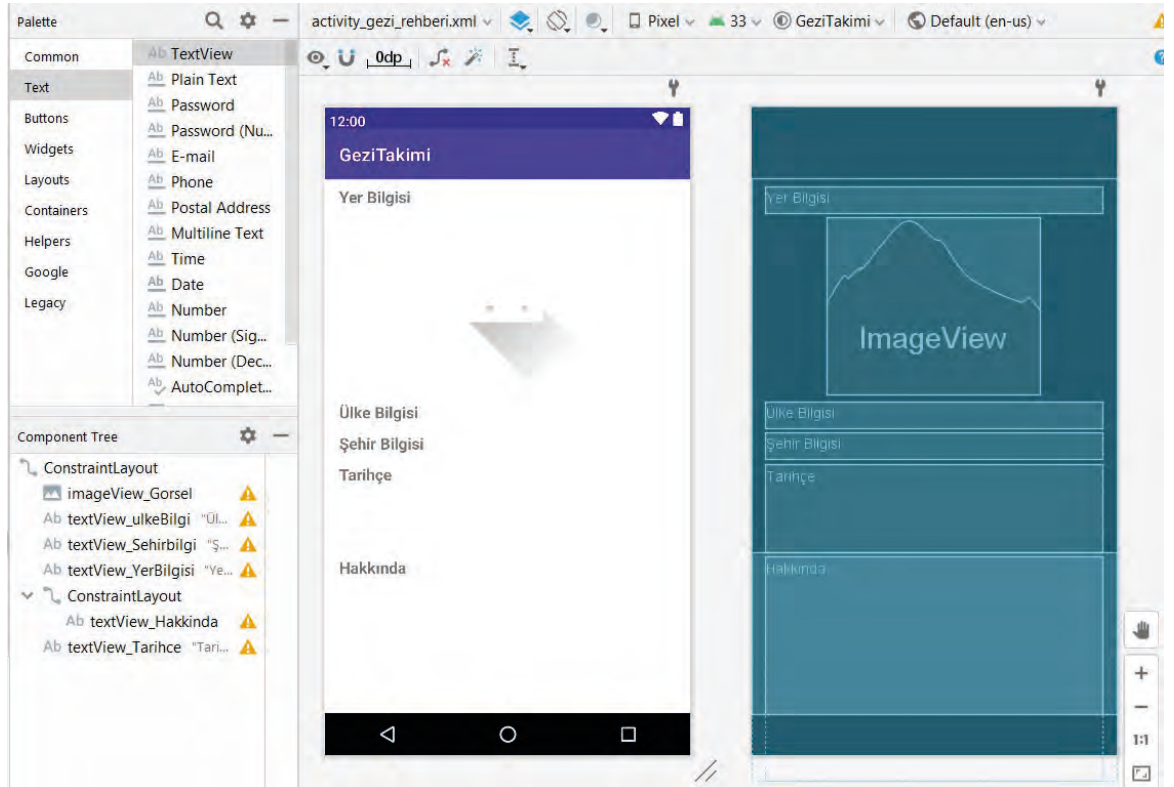
- Anasayfanın yer aldığı **MainActivity** isminde bir activity vardır. Bu activity içinde ListView oluşturulup eklenen gezi yerleri listelenir.
- Logonun açıldığı bir activity ve bu activitynin bir classı olmalıdır. Bunun için de **LogoActivity** isimli bir activity oluşturulur.
- “Yer Ekle” seçeneğine tıklandığı zaman açılacak ve kayıtların fragmentler ile Firebase’e gönderileceği bir activity oluşturulmalıdır. Bunun için **KayıtActivity** isimli bir activity oluşturulur.
- Anasayfada yer alan bir yerin üzerine tıklandığında o yere ait bilgilerin ImageView ve TextViewlerde gösterileceği bir activity gerekir. Bunun için de **GeziRehberiActivity** isminde bir activity oluşturulur.
- Anasayfada hem Firebase’den bilgiler alınıp ArrayList’e eklenirken hem de bilgilerin detaylı gösterileceği GeziRehberiActivity’e gönderilirken toplu tutulması amacıyla **GeziRehberBilgileri** isimli bir class oluşturulur.
- Ayrıca KayıtActivity içinde yer alacak üç adet fragment gerekir. Bu fragmentlerin ilkinde fotoğraf seçimi yapılır, ikincisinde Türkçe bilgileri girilir, üçüncüsünde ise İngilizce bilgilerinin girilmesi sağlanır. Bunun için üç adet Fragment oluşturularak, (Blank Fragment) class dosyasının içeriği silinip sıfırdan hazırlanır.
- Anasayfada, sağ üst köşede üç noktalı menü görünümü için menü dosyası oluşturulur. İsmine **gezi_menu** girilir. Ayrıca bilgilerin gösterileceği “GeziRehberiActivity” içinde de bir dil seçimi menüsü uygundur. Bunun için **dilsecimi_menu** isminde bir menü dosyası oluşturulur.
- Oluşturulan dosyaların son durumu Görsel 3.21’de gösterilmiştir.



Görsel 3.21: İngilizce bilgi girişi

3.3.1. activity_gezi_rehberi.xml

Layout içeriği Görsel 3.22’de gösterilmiştir. Bilgiler TextView ve Multiline Textler ile hazırlanır, görsel de imageViewden oluşur. İçerikteki komponentlerin id’si Component Tree’den görülebilir.

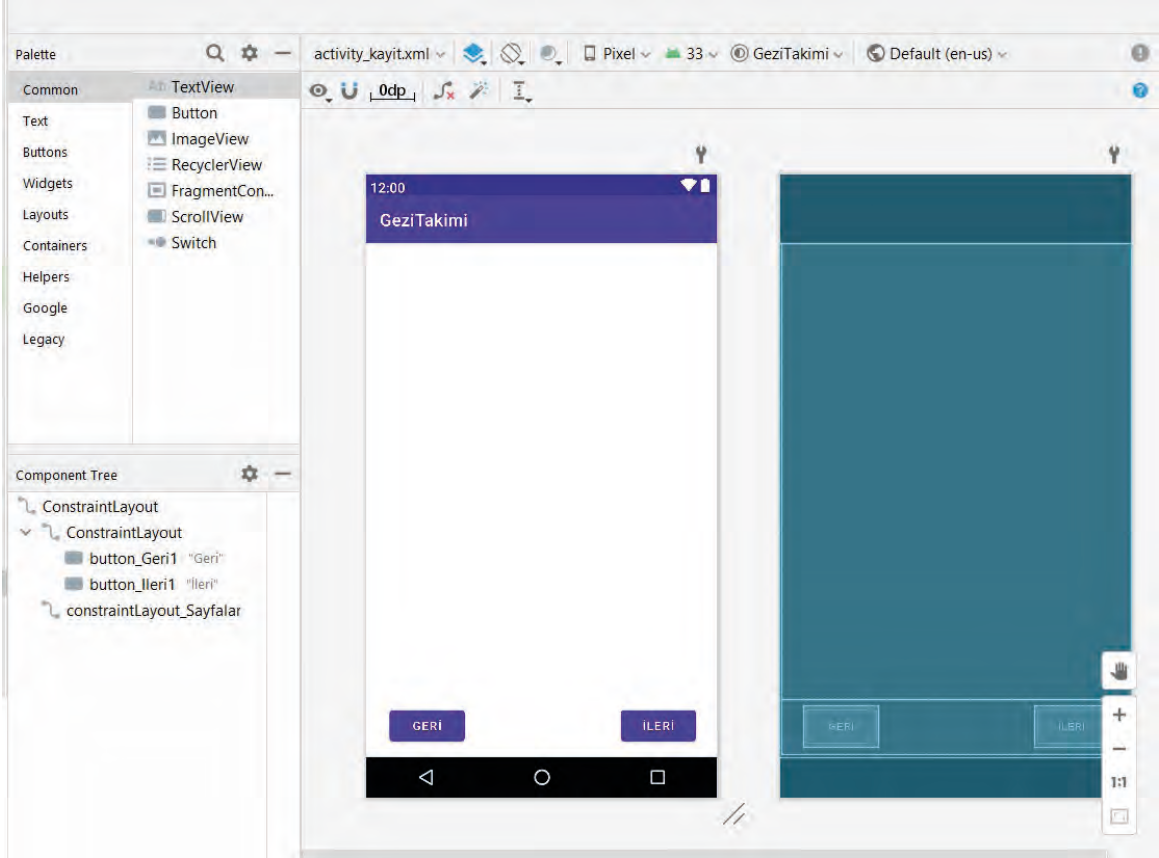


Görsel 3.22: activity_gezi_rehberi.xml



3.3.2. activity_kayit.xml

Layout içeriği Görsel 3.23'te gösterilmiştir. Bu sayfa alt bölümde yer alan ekstra bir ConstraintLayout içinde iki adet butondan oluşur. Üst bölümde yer alan diğer ConstraintLayout içine ise fragmentlerin layoutları bağlanır. İçeriğin komponent id'leri Component Tree'den görülebilir.



Görsel 3.23: activity_kayit.xml

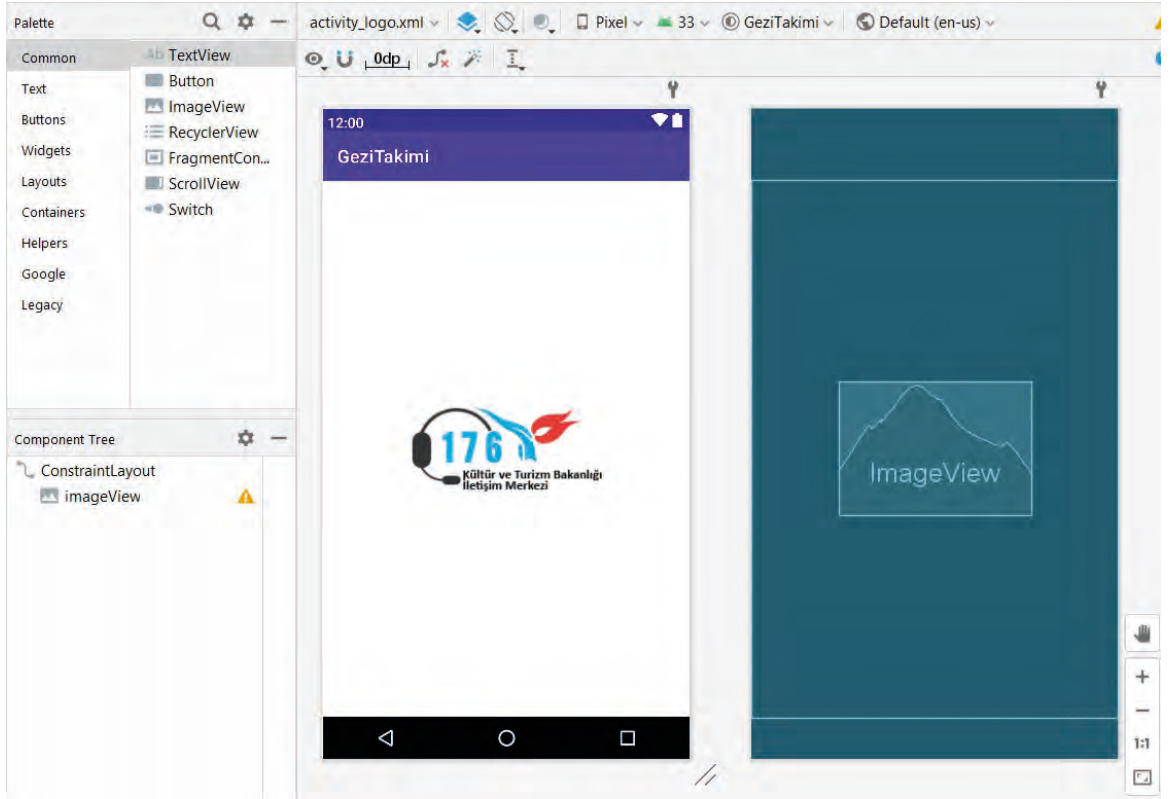
3.3.3. activity_logo.xml

Layout içeriği Görsel 3.24'te gösterilmiştir. Bu içerikten önce “res” klasörü altında yer alan drawable klasörüne bir adet görsel eklenir. Logo, imageView'e bağlanarak sayfanın ortasına yerleştirilir. İçeriğin komponent id'leri Component Tree'den görülebilir.



NOT

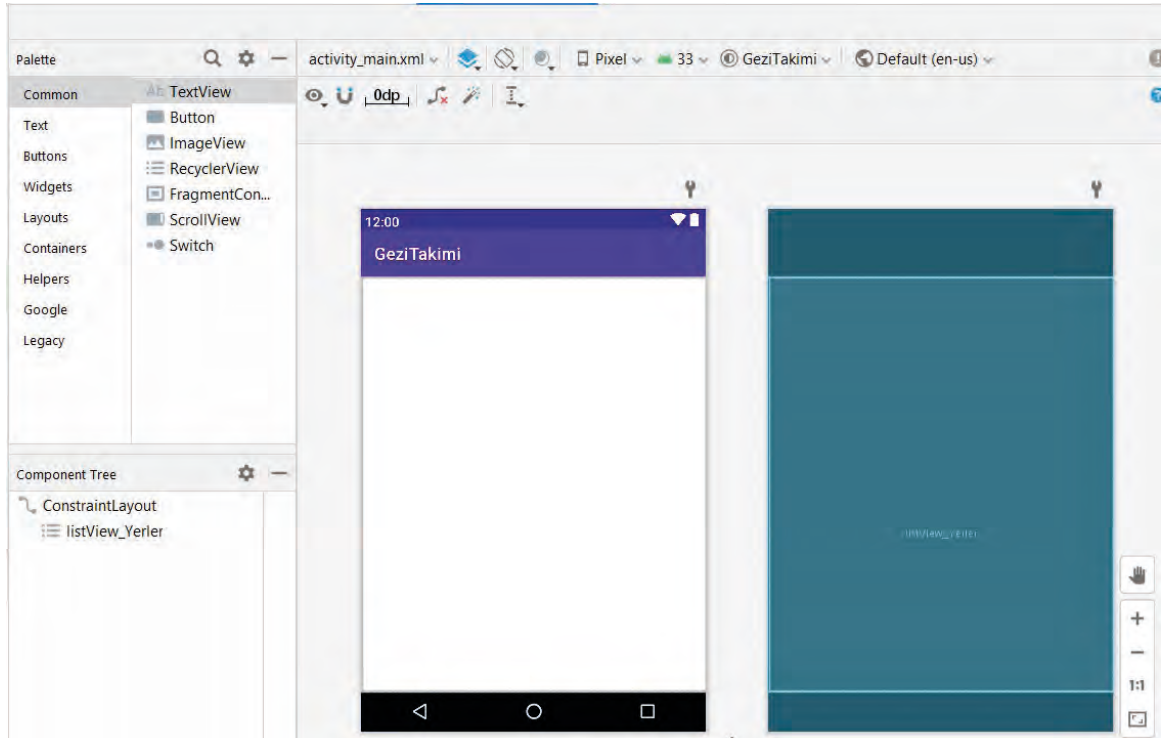
İlgili görsel içinde yer alan logo, Turizm Bakanlığının internet sitesinden alınmıştır. Bu uygulama sadece pratik yapmak amacıyla kullanıldığı sürece istenen görsel seçilebilir. Hazırlanan uygulama ticari bir amaçla markete konulacaksa kullanılan içerikteki görsellerin telif hakları mutlaka alınmalıdır.



Görsel 3.24: activity_logo.xml

3.3.4. activity_main.xml

Layout içeriği Görsel 3.25'te gösterilmiştir. İlgili içeriğe sadece bir adet ListView eklenir. İçeriğin komponent id'leri Component Tree'den görülebilir.

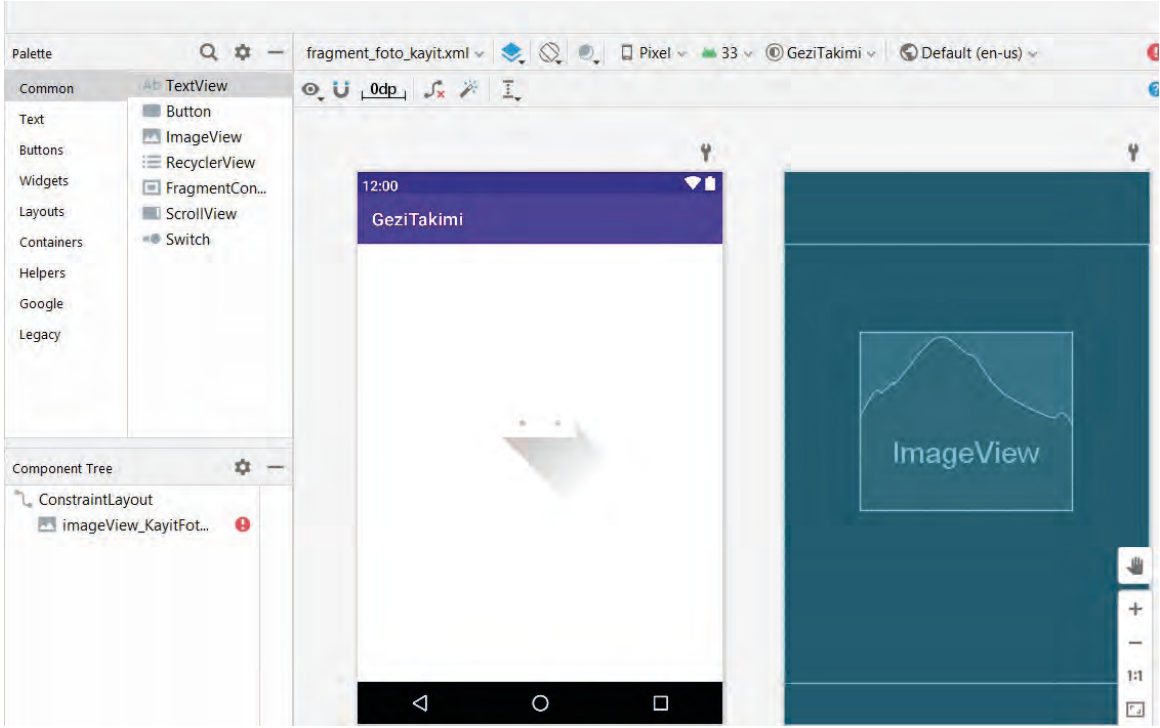


Görsel 3.25: activity_main.xml



3.3.5. fragment_foto_kayit.xml

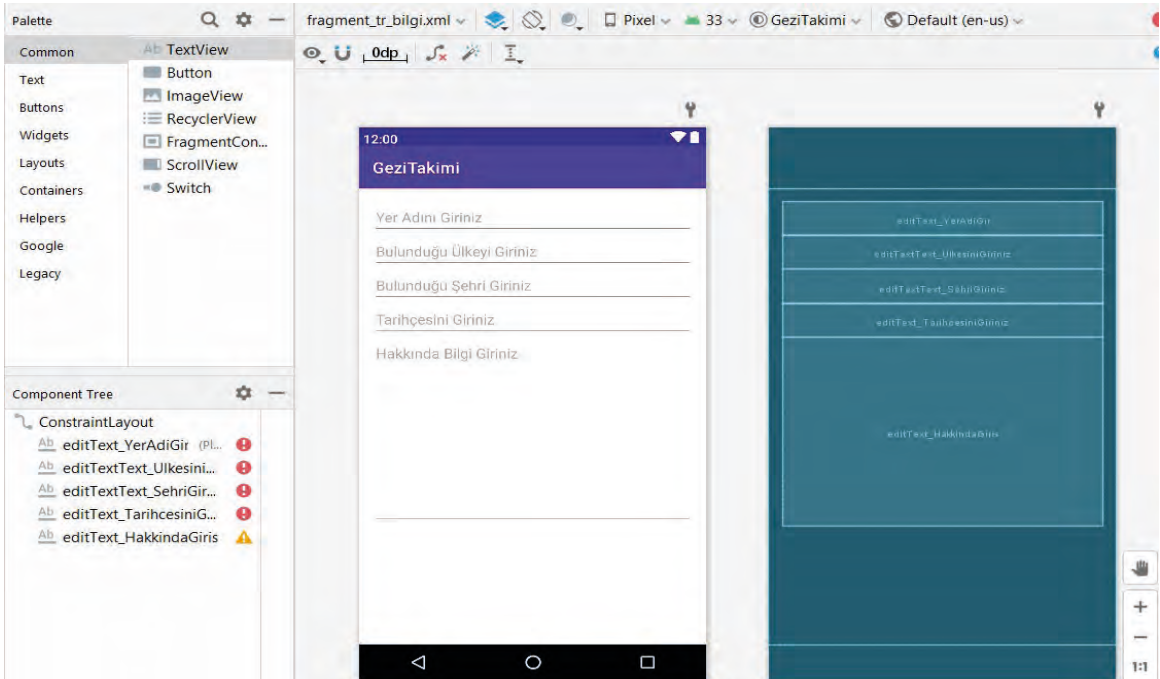
Fragmente ait layout içeriği Görsel 3.26'da gösterilmiştir. Tek bir imageViewden meydana gelen içeriğin imageView id'si Component Tree'den görülebilir.



Görsel 3.26: fragment_foto_kayit.xml

3.3.6. fragment_tr_bilgi.xml

Fragmente ait layout içeriği Görsel 3.27'de gösterilmiştir. İçerikteki komponentlerin türü ve id'si Component Tree'den görülebilir.

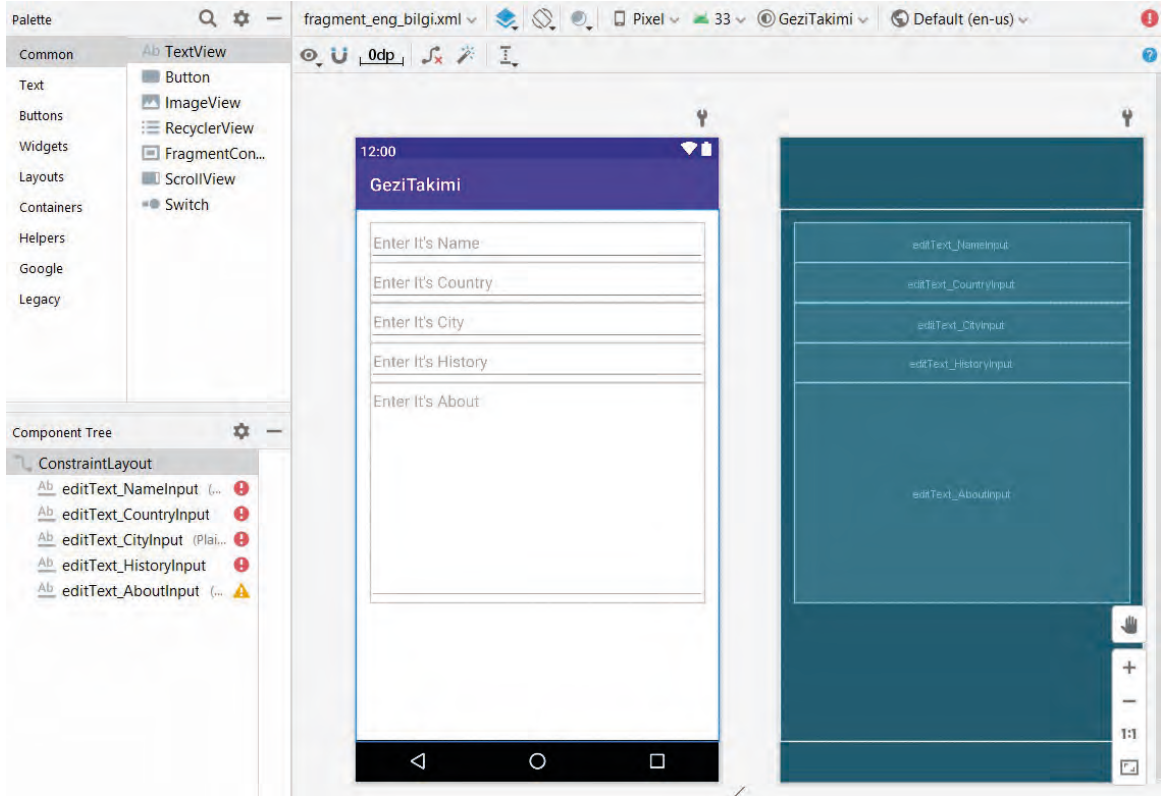


Görsel 3.27: fragment_tr_bilgi.xml



3.3.7. fragment_eng_bilgi.xml

Fragmente ait layout içeriği Görsel 3.28’de gösterilmiştir. İçerikteki komponentlerin türü ve id’si Component Tree’den görülebilir.



Görsel 3.28: fragment_eng_bilgi.xml

3.3.8. gezi_menu.xml

Kod olarak ayarlanan menü dosyasında dört adet item bulunur. Kod içinde bu itemlere ait id ve başlıkları yer alır.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/girisYap" android:title="Giriş Yap"></item>
  <item android:id="@+id/geziEkle" android:title="Yer Ekle"></item>
  <item android:id="@+id/geziSil" android:title="Yer Sil"></item>
  <item android:id="@+id/dilSecimi" android:title="English"></item>
</menu>
```

3.3.9. dilsecimi_menu.xml

Kod olarak ayarlanan menü dosyasında iki adet item bulunur. Kod içinde bu itemlere ait id ve başlıkları yer alır.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/secim" android:title="English"></item>
  <item android:id="@+id/geri" android:title="Geri"></item>
</menu>
```



3.3.10. build.gradle (Project:GeziTakimi)

Proje düzeyindeki gradle içeriği şu şekildedir:

```
buildscript {
    repositories {
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
    dependencies {
        classpath 'com.google.gms:google-services:4.3.13'
    }
}
plugins {
    id 'com.android.application' version '7.2.2' apply false
    id 'com.android.library' version '7.2.2' apply false
}
allprojects {
    repositories {
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
}
task clean(type: Delete) {
    delete rootProject.buildDir }
```

3.3.11. build.gradle (Module:GeziTakimi.app)

Modül düzeyindeki gradle içeriği şu şekildedir:

```
plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}
android {
    compileSdk 32
    defaultConfig {
        applicationId "com.atilimciftci.gezitakimi"
        minSdk 29
        targetSdk 32
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures{
        viewBinding true
    }
}
```



```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.5.0'
    implementation 'com.google.android.material:material:1.6.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'androidx.navigation:navigation-fragment:2.5.1'
    implementation 'androidx.navigation:navigation-ui:2.5.1'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

    //Firebase eklentilerinin yer aldığı kütüphaneler
    implementation platform('com.google.firebase:firebase-bom:30.3.2')
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'com.google.firebase:firebase-auth'
    implementation 'com.google.firebase:firebase-storage:20.0.2'
    implementation 'com.google.firebase:firebase-firestore'

    //Veri tabanından görselin download adresi geldiğinde görseli indirip gösterecek olan kütüphane
    implementation('com.squareup.picasso:picasso:2.71828')
}
```

3.3.12. AndroidManifest.xml

AndroidManifest içinde hem ilk açılacak activitynin hem de fotoğrafın galeriden seçilebilmesi için **READ_EXTERNAL_STORAGE** izninin tanımı yapılmalıdır. LogoActivity, <intent-filter> tagleri arasında yer alan action ile ilk açılacak activity olarak belirlenir.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.atilimciftci.gezitakimi">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.GeziTakimi"
        tools:targetApi="31">
        <activity
            android:name=".KayitActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="false" />
        <activity
            android:name=".GeziRehberiActivity"
            android:exported="false" />
        <activity
            android:name=".LogoActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



3.3.13. LogoActivity.java

Bu activity içinde, uygulama ilk çalıştırıldığında uygulamaya ait logonun 5 saniye gösterilmesi ve sonrasında MainActivity'ye gidilmesi, bir başka deyişle intent edilmesi gerekir. Bu nedenle 5 saniyeyi sayabilmek için LogoActivity içine ayrı bir class yazılıp Timer tanımlamaları ve Timer ayarlamaları belirtilen class içinde yapılır.

```
package com.atilimciftci.gezitakimi;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.MenuInflater;
import android.view.View;
import com.atilimciftci.gezitakimi.databinding.ActivityLogoBinding;
import java.util.Timer;

public class LogoActivity extends AppCompatActivity {
    private ActivityLogoBinding activityLogoBinding;
    ZamanSayaci timer;
    int i;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        activityLogoBinding = ActivityLogoBinding.inflate(getLayoutInflater());
        View view = activityLogoBinding.getRoot();
        setContentView(view);

        timer = new ZamanSayaci(5000,1000);
        i=0;
        timer.start();
    }
    class ZamanSayaci extends CountDownTimer{
        public ZamanSayaci(long millisInFuture, long countDownInterval) {
            super(millisInFuture, countDownInterval);
        }
        @Override
        public void onTick(long l) {
            i++;
        }
        @Override
        public void onFinish() {
            Intent intent = new Intent(LogoActivity.this,MainActivity.class);
            startActivity(intent);
        }
    }
}
```

3.3.14. GeziRehberBilgileri.java

Firebase yapısına gönderilecek bilgilerin toplu olarak tutulması ve veriler çekilmek istendiğinde verilerin paket hâline getirilmesi için oluşturulan class içeriği de yazılır.

```
package com.atilimciftci.gezitakimi;
import java.io.Serializable;
public class GeziRehberBilgileri implements Serializable {
    String imageID;
    String yerAdi, ulkeAdi, sehirAdi, tarihce, hakkında;
    String placeName, countryName, cityName, history, about;
```



```
package com.atilimciftci.gezitakimi;
import java.io.Serializable;
public class GeziRehberBilgileri implements Serializable {
    String imageID;
    String yerAdi, ulkeAdi, sehirAdi, tarihce, hakkinda;
    String placeName, countryName, cityName, history, about;
    public GeziRehberBilgileri(String imageID, String yerAdi, String ulkeAdi, String sehirAdi, String tarihce,
        String hakkında, String placeName, String countryName, String cityName, String history, String
        about) {
        this.imageID = imageID;
        this.yerAdi = yerAdi;
        this.ulkeAdi = ulkeAdi;
        this.sehirAdi = sehirAdi;
        this.tarihce = tarihce;
        this.hakkinda = hakkında;
        this.placeName = placeName;
        this.countryName = countryName;
        this.cityName = cityName;
        this.history = history;
        this.about = about;
    } }

```

3.3.15. FotoKayitFragment.java

Bu fragment, KayitActivity içinde sırasıyla gösterilecek fragment içeriği olacaktır. İçerikte galeriye gitmek için izin verilmesi gerektiğinden bu iznin kullanıcıdan istenmesi sağlanır. Bu izin verilmezse ve yine seçim yapılmaya çalışılırsa da Snackbar kullanılarak gerekli hatırlatmanın bu **Snackbar** ile yapılması ve iznin verilmesi sağlanır.

```
package FragmentTutucu;
import static android.app.Activity.RESULT_OK;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;

import androidx.activity.result.ActivityResult;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;

import android.provider.MediaStore;
import android.view.LayoutInflater;

import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.Toast;

import com.atilimciftci.gezitakimi.R;
import com.google.android.material.snackbar.Snackbar;

```



```
public class FotoKayitFragment extends Fragment {

    public static Uri imageData; // Galeriden gelen görüntünün Uri adresinin tutulacağı değişken
    ImageView imageView; // ImageView tanımı
    ActivityResultLauncher<Intent> activityResultLauncher; //Galeriye gitmek için gereken Intent
    ActivityResultLauncher<String> permissionResultLauncher; //İzin istemek için gereken tanımlama
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        activityResultLauncherBaslatmalari();
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_foto_kayit, container, false);
    }
    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        imageView = view.findViewById(R.id.imageView_KayitFotoSecim);
        if(imageData!=Uri.EMPTY){
            imageView.setImageURI(imageData);
        }
        //ImageViewe tıklanırsa ne olacağının yazıldığı listener
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(ContextCompat.checkSelfPermission(getContext(),
                    Manifest.permission.READ_EXTERNAL_STORAGE)!= PackageManager.PERMISSION_GRANTED){
                    if(ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),Manifest.permission.READ_EXTERNAL_STORAGE)){
                        Snackbar.make(view,"Permission needed gallery",Snackbar.LENGTH_INDEFINITE).setAction("Give Permission", new View.OnClickListener() {

                            @Override
                            public void onClick(View view) {
                                //Gerekli izin istenir.
                                permissionResultLauncher.launch(Manifest.permission.READ_EXTERNAL_STORAGE);
                            }
                        }).show();
                    }else{
                        // Gerekli izin istenir.
                        permissionResultLauncher.launch(Manifest.permission.READ_EXTERNAL_STORAGE);
                    }
                }else{
                    //Galeriye gidilir.
                    Intent intentToGalery = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                    //Launcher yazılarak getirilir.
                    activityResultLauncher.launch(intentToGalery);
                }
            }
        });
    }
}
```



```

private void activityResultLauncherBaslatmalari(){
    //Galeriden görseli ve görselin datasını döndürmek için yazılması gerekenler
    activityResultLauncher = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if(result.getResultCode()== RESULT_OK){
                Intent intentFromResult = result.getData();
                if(intentFromResult!=null){
                    imageData = intentFromResult.getData();
                    imageView.setImageURI(imageData);
                    System.out.println(imageData);
                }
            }
        }
    });

    //İzin istekleri için yazılması gerekenler
    permissionResultLauncher = registerForActivityResult(new ActivityResultContracts.RequestPermission(), new ActivityResultCallback<Boolean>() {
        @Override
        public void onActivityResult(Boolean result) {
            if(result){
                Intent intentToGalery = new Intent(Intent.ACTION_PICK,MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                activityResultLauncher.launch(intentToGalery);
            }else{
                Toast.makeText(getContext(),"Permission needed!", Toast.LENGTH_LONG).show();
            }
        }
    });
}
}

```

3.3.16. TrBilgiFragment.java

Fotoğraf seçimi yapıp İleri tuşuna basıldıktan sonra KayitActivity'nin içinde FotoKayitFragment yerine gelecek fragment türüdür. Bu fragmentte yerle ilgili Türkçe bilgi girişlerinin yapılması gerekir.

```

package FragmentTutucu;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;

import com.atilimciftci.gezitakimi.KayitActivity;
import com.atilimciftci.gezitakimi.R;
import com.atilimciftci.gezitakimi.databinding.FragmentTrBilgiBinding;

public class TrBilgiFragment extends Fragment {
    public static EditText yerAdi,ulkeAdi,sehirAdi,tarihce,hakkindaGiris;
}

```



```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_tr_bilgi, container, false);
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    yerAdi = view.findViewById(R.id.editText_YerAdiGir);
    ulkeAdi = view.findViewById(R.id.editTextText_UlkesiniGiriniz);
    sehirAdi = view.findViewById(R.id.editTextText_SehirGiriniz);
    tarihce = view.findViewById(R.id.editText_TarihcesiniGiriniz);
    hakkindaGiris = view.findViewById(R.id.editText_HakkindaGiris);

    veriEkranaGetir();
}

public void veriEkranaGetir(){
    if(!KayitActivity.yerAdiBilgi.equals("")){
        yerAdi.setText(KayitActivity.yerAdiBilgi);
    }
    if(!KayitActivity.ulkeAdiBilgi.equals("")){
        ulkeAdi.setText(KayitActivity.ulkeAdiBilgi);
    }
    if(!KayitActivity.sehirAdiBilgi.equals("")){
        sehirAdi.setText(KayitActivity.sehirAdiBilgi);
    }
    if(!KayitActivity.tarihceBilgi.equals("")){
        tarihce.setText(KayitActivity.tarihceBilgi);
    }
    if(!KayitActivity.hakkindaBilgi.equals("")){
        hakkindaGiris.setText(KayitActivity.hakkindaBilgi);
    }
}
}
```

3.3.17. EngBilgiFragment.java

Türkçe bilgilerin girişi yapıldıktan sonra İleri tuşuna basılıp KayitActivity'nin içinde TrBilgiFragment yerine gelecek fragment türüdür. Bu fragmentte yerle ilgili İngilizce bilgi girişlerinin yapılması gerekir.

```
package FragmentTutucu;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;

import com.atilimciftci.gezidakimi.R;
```



```

package FragmentTutucu;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;

import com.atilimciftci.gezitakimi.R;

public class EngBilgiFragment extends Fragment {
    public static EditText placeName, countryName, cityName, history, aboutInfo;

    public EngBilgiFragment() {
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

        return inflater.inflate(R.layout.fragment_eng_bilgi, container, false);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        placeName = view.findViewById(R.id.editText_NameInput);
        countryName = view.findViewById(R.id.editText_CountryInput);
        cityName = view.findViewById(R.id.editText_CityInput);
        history = view.findViewById(R.id.editText_HistoryInput);
        aboutInfo = view.findViewById(R.id.editText_AboutInput);
    }
}

```

3.3.18. KayitActivity.java

Tüm fragmentleri içinde tutan ve anasayfadaki menü üzerinden “Yer Ekle” seçeneği ile ulaşılan activitydir. Fragmentlere girilen tüm bilgiler KayitActivity içine fragmentlerden alınarak çekilir ve Firebase yapısına bu java dosyası üzerinden toplu hâlde gönderilir. Bu toplu gönderim esnasında bir **HashMap** kullanılır.

```

package com.atilimciftci.gezitakimi;
package com.atilimciftci.gezitakimi;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

```



```
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.sax.EndElementListener;
import android.view.View;
import android.widget.Toast;

import com.atilimciftci.gezitakimi.databinding.ActivityKayitBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FieldValue;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.util.HashMap;
import java.util.UUID;

import FragmentTutucu.EngBilgiFragment;
import FragmentTutucu.FotoKayitFragment;
import FragmentTutucu.TrBilgiFragment;

public class KayitActivity extends AppCompatActivity {
    FirebaseStorage firebaseStorage;
    FirebaseFirestore firebaseFirestore;
    StorageReference storageReference;
    private ActivityKayitBinding kayitBinding;
    private int pageNumber;
    Uri fotoBilgi; // Fotoğrafın Uri adresi tutulur.
    public static String yerAdiBilgi,ulkeAdiBilgi,sehirAdiBilgi,tarihceBilgi,hakkindaBilgi; //Tr bilgileri tutulur.
    public static String nameInfo, countryInfo, cityInfo, historyInfo, aboutInfo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Değişkenlerde veri içerik temizlemeleri yapılır.
        fotoBilgi=Uri.EMPTY;

        yerAdiBilgi="";
        ulkeAdiBilgi="";
        sehirAdiBilgi="";
        tarihceBilgi="";
        hakkindaBilgi="";

        nameInfo="";
        countryInfo="";
        cityInfo="";
        historyInfo="";
        aboutInfo="";

        firebaseStorage = FirebaseStorage.getInstance();
        firebaseFirestore = FirebaseFirestore.getInstance();
        storageReference = firebaseStorage.getReference();
    }
}
```



```

kayitBinding = ActivityKayitBinding.inflate(getLayoutInflater());
View view = kayitBinding.getRoot();
setContentView(view);
pageNumber = 0;

//Fotoğraf seçimi için ilk fragment çağrılır.
FragmentManager fragmentManager = getSupportFragmentManager();
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
FotoKayitFragment fotoKayitFragment = new FotoKayitFragment();
fragmentTransaction.replace(R.id.constraintLayout_Sayfalar,fotoKayitFragment).commit();

//İleri butonunun yapacağı iş listener ile dinlenir.
kayitBinding.butonIleri1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {

        if(pageNumber==0){
            fotoBilgi = FotoKayitFragment.imageData;
            FragmentManager fragmentManager = getSupportFragmentManager();
            FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
            TrBilgiFragment trBilgiFragment = new TrBilgiFragment();
            fragmentTransaction.replace(R.id.constraintLayout_Sayfalar,trBilgiFragment).commit();
            kayitBinding.butonIleri1.setText("İleri");
        }

        else if(pageNumber==1){
            yerAdiBilgi =

            TrBilgiFragment.yerAdi.getText().toString();
            ulkeAdiBilgi = TrBilgiFragment.ulkeAdi.getText().toString();
            sehirAdiBilgi = TrBilgiFragment.sehirAdi.getText().toString();
            tarihceBilgi = TrBilgiFragment.tarihce.getText().toString();
            hakkindaBilgi = TrBilgiFragment.hakkindaGiris.getText().toString();

            FragmentManager fragmentManager = getSupportFragmentManager();
            FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
            EngBilgiFragment engBilgiFragment = new EngBilgiFragment();
            fragmentTransaction.replace(R.id.constraintLayout_Sayfalar,engBilgiFragment).commit();

            kayitBinding.butonIleri1.setText("Kaydet");
        }

        else if(pageNumber==2){
            nameInfo = EngBilgiFragment.placeName.getText().toString();
            countryInfo = EngBilgiFragment.countryName.getText().toString();
            cityInfo = EngBilgiFragment.cityName.getText().toString();
            historyInfo = EngBilgiFragment.history.getText().toString();
            aboutInfo = EngBilgiFragment.aboutInfo.getText().toString();

            //Veriler veri tabanına kaydedilir.
            AlertDialog.Builder alertDialog = new AlertDialog.Builder(KayitActivity.this);
            alertDialog.setTitle("KAYIT");
            alertDialog.setMessage("Kayıt Etmek İstiyor Musunuz");
            alertDialog.setNegativeButton("HAYIR", new DialogInterface.OnClickListener() {

```



```

@Override
public void onClick(DialogInterface dialogInterface, int i) {
    //Herhangi bir şey yapılmaz.
}
});

alertDialog.setPositiveButon("EVET", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        //Veriler veri tabanına gönderilir.
        System.out.println(fotoBilgi);
        if(fotoBilgi != Uri.EMPTY){

            UUID uuid = UUID.randomUUID(); //Rastgele bir kimlik numarası oluşturulup fotoğrafa
            bu isim verilir.
            String imageName = "images/"+uuid+".jpg";

            storageReference.child(imageName).putFile(fotoBilgi).addOnSuccessListener(new On-
            SuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    //Fotoğraf kaydı başarılı olursa veriler veri tabanına alınır.
                    System.out.println("Foto Kayıt Başarılı");
                    StorageReference yeniReferans = FirebaseStorage.getReference(imageName);
                    yeniReferans.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<U-
ri>() {

                        @Override
                        public void onSuccess(Uri uri) {
                            String downloadUrl = uri.toString();

                            HashMap<String,Object> yerData = new HashMap<>();
                            yerData.put("adi",yerAdiBilgi);
                            yerData.put("ulkesi",ulkeAdiBilgi);
                            yerData.put("sehir",sehirAdiBilgi);
                            yerData.put("tarihce",tarihceBilgi);
                            yerData.put("hakkinda",hakkindaBilgi);
                            yerData.put("gorselURL",downloadUrl);

                            yerData.put("placeName",nameInfo);
                            yerData.put("countryName",countryInfo);
                            yerData.put("cityName",cityInfo);
                            yerData.put("historyInfo",historyInfo);
                            yerData.put("aboutInfo",aboutInfo);
                            yerData.put("kayitTarihi", FieldValue.serverTimestamp());

                            System.out.println(yerAdiBilgi+ " " + nameInfo+ " " + historyInfo);;
                            FirebaseFirestore.collection("YerKayit").add(yerData).addOnSuccessListener(-
new OnSuccessListener<DocumentReference>() {
                                @Override
                                public void onSuccess(DocumentReference documentReference) {
                                    //Veri kaydı başarılı olursa bildirim gönderilir.
                                    System.out.println("Veri kaydı başarılı");
                                }
                            });
                        //Veriler veri tabanına kaydedilir.
                        AlertDialog.Builder alertDialog = new AlertDialog.Builder(KayitActivity.this);
                        alertDialog.setTitle("KAYIT");

```



```

alertDialog.setMessage("Kayıt Etmek İstiyor Musunuz");
alertDialog.setNegativeButton("HAYIR", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        //Herhangi bir şey yapılmaz.
    }
});

alertDialog.setPositiveButon("EVET", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        //Veriler veri tabanına gönderilir.
        System.out.println(fotoBilgi);
        if(fotoBilgi != Uri.EMPTY){

            UUID uuid = UUID.randomUUID(); //Rastgele bir kimlik numarası oluşturulup fotoğrafa
            bu isim verilir.
            String imageName = "images/"+uuid+".jpg";
            Toast.makeText(getApplicationContext(),"Kayıt Başarılı",Toast.LENGTH_LONG).show();
            finish();
            Intent intent = new Intent(getApplicationContext(),MainActivity.class);
            startActivity(intent);
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            System.out.println(e.getLocalizedMessage());
            Toast.makeText(getApplicationContext(),e.getLocalizedMessage(),Toast.
LENGTH_LONG).show();
        }
    });
}
});
}
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        //Fotoğraf kaydı başarısız olursa hata mesajı e.getLocalizedMessage ile kullanıcıya
        gönderilir.
        Toast.makeText(getApplicationContext(),e.getLocalizedMessage(),Toast.LENGTH_
LONG).show();

        System.out.println("Başarısız");
        System.out.println(e.getLocalizedMessage());
    }
});
}
}
});
alertDialog.show();
}
if(pageNumber<2){
    pageNumber++;
}
}
System.out.println(pageNumber);
}
});

```



```
//Geri butonunun yapacağı iş listener ile dinlenir.
kayitBinding.butonGeri1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (pageNumber>0){
            pageNumber--;
        }
        else
        {
            //Çıkış gerektiği zamanlarda AlertDialog ile kullanıcının çıkışı onaylaması teyit edilir.
            AlertDialog.Builder alertDialog = new AlertDialog.Builder(KayitActivity.this);
            alertDialog.setTitle("ÇIKIŞ");
            alertDialog.setMessage("Çıkmak istiyor Musunuz? ");
            alertDialog.setNegativeButton("HAYIR", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                }
            });
            alertDialog.setPositiveButton("EVET", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    finish();
                    Intent intent = new Intent(KayitActivity.this,MainActivity.class);
                    startActivity(intent);
                }
            });
            alertDialog.show();
        }
    }

    System.out.println(pageNumber);
    if(pageNumber==0){
        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        FotoKayitFragment fotoKayitFragment = new FotoKayitFragment();
        fragmentTransaction.replace(R.id.constraintLayout_Sayfalar,fotoKayitFragment).commit();
        kayitBinding.butonIleri1.setText("İleri");
    }

    else if(pageNumber==1){
        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        TrBilgiFragment trBilgiFragment = new TrBilgiFragment();
        fragmentTransaction.replace(R.id.constraintLayout_Sayfalar,trBilgiFragment).commit();
        kayitBinding.butonIleri1.setText("İleri");
    }
}
});
}
```



3.3.19. MainActivity.java

Bu java dosyası anasayfa görevindedir. Listeler burada yer alır ve menü burada tutulur. “English” seçeneği seçildiğinde yer bilgilerinin İngilizce isimleri, “Türkçe” seçeneği seçilirse yer bilgilerinin Türkçe isimleri getirilir. Ayrıca üzerine tıklanan her yerin bilgilerinin detaylı gösterimi için **GeziRehberiActivity** içine “**GeziRehberiBilgileri**” classındaki **Serializable** özelliği kullanılarak toplu hâlde gönderim yapılır.

```
package com.atilimciftci.gezitakimi;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Toast;

import com.atilimciftci.gezitakimi.databinding.ActivityMainBinding;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.firebase.storage.FirebaseStorage;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.stream.Collectors;
import java.util.zip.Inflater;

public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding activityMainBinding;
    Map<String,Object> mapData; //Çekilecek veriler bu Map'te tutulur.
    FirebaseFirestore firebaseFirestore;
    FirebaseStorage firebaseStorage;
    ArrayList<GeziRehberiBilgileri> yerler;

    int dilSecimi=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        activityMainBinding = ActivityMainBinding.inflate(getLayoutInflater());
        View view = activityMainBinding.getRoot();
        setContentView(view);

        yerler = new ArrayList<>();
        FirebaseFirestore = FirebaseFirestore.getInstance();
        FirebaseStorage = FirebaseStorage.getInstance();
    }
}
```



```
Intent intent = getIntent();
dilSecimi=intent.getIntExtra("dilSecimi",0);
verileriAlTr(); //Verilerin veri tabanından getirileceği metottur.
}

@Override
public boolean onCreateOptionsMenu(@NonNull Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.gezi_menu,menu);
    if(dilSecimi==0){
        menu.getItem(3).setTitle("English");
    }else{
        menu.getItem(3).setTitle("Türkçe");
    }
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if(item.getItemId()==R.id.girisYap){
    }
    else if(item.getItemId()==R.id.geziEkle){
        Intent intent = new Intent(MainActivity.this,KayitActivity.class);
        startActivity(intent);
    }
    else if(item.getItemId()==R.id.geziSil){
    }
    else if(item.getItemId()==R.id.dilSecimi){
        if(dilSecimi==0){
            dilSecimi=1;
            item.setTitle("Türkçe");
            verileriAlTr();
        }else{
            dilSecimi=0;
            item.setTitle("English");
            verileriAlTr();
        }
    }
    return super.onOptionsItemSelected(item);
}

public void verileriAlTr(){
    yerler.clear();
    FirestoreFirestore.collection("YerKayit").addSnapshotListener(new EventListener<QuerySnapshot>()
{
    @Override
    public void onEvent(@Nullable QuerySnapshot value, @Nullable FirebaseFirestoreException error) {
        if(error != null){
            Toast.makeText(getApplicationContext(),error.getLocalizedMessage(),Toast.LENGTH_LONG).
show();
        }
        if(value!=null){
            for(DocumentSnapshot snapshot:value.getDocuments()){
                mapData = snapshot.getData();
            }
        }
    }
});
}
```




3.3.20. GeziRehberiActivity.java

Bu activity, MainActivity içinde tıklanan yer seçeneğindeki detayları ekrana getirir. Serializable olarak gelen verileri GeziRehberBilgileri türüne cast ederek (dönüştürerek) activity içinde gösterir. MainActivity içinde seçilen dil olarak göstereceği veriler aynı şekilde buradaki menü üzerinden de dil değişimi yapılarak değiştirilebilir.

```
package com.atilimciftci.gezitakimi;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;

import com.atilimciftci.gezitakimi.databinding.ActivityGeziRehberiBinding;
import com.squareup.picasso.Picasso;

import java.io.Serializable;

public class GeziRehberiActivity extends AppCompatActivity {
    private ActivityGeziRehberiBinding activityGeziRehberiBinding;
    int dilSecimi;
    GeziRehberBilgileri geziRehberBilgileri;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        activityGeziRehberiBinding = ActivityGeziRehberiBinding.inflate(getLayoutInflater());
        View view = activityGeziRehberiBinding.getRoot();
        setContentView(view);
        Intent intent = getIntent();
        geziRehberBilgileri = (GeziRehberBilgileri) intent.getSerializableExtra("geziYerleriDetay");
        dilSecimi = intent.getIntExtra("dilSecimi",0);

        verileriAl();
    }

    @Override
    public boolean onCreateOptionsMenu(@NonNull Menu menu) {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.dilsecimi_menu,menu);
        if(dilSecimi==0){
            menu.getItem(0).setTitle("English");
        }else{
            menu.getItem(0).setTitle("Türkçe");
        }
        return super.onCreateOptionsMenu(menu);
    }
}
```



```

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if(item.getItemId()==R.id.geri){
        finish();
        Intent intent = new Intent(GeziRehberiActivity.this,MainActivity.class);
        intent.putExtra("dilSecimi",dilSecimi);
        startActivity(intent);
    }else if(item.getItemId()==R.id.secim){
        if(dilSecimi==0){
            dilSecimi=1;
            item.setTitle("Türkçe");
            verileriAl();
        }else{
            dilSecimi=0;
            item.setTitle("English");
            verileriAl();
        }
    }
    return super.onOptionsItemSelected(item);
}

public void verileriAl(){
    Picasso.get().load(geziRehberBilgileri.imageID).into(activityGeziRehberiBinding.imageViewGorsel);
    if(dilSecimi==0){
        activityGeziRehberiBinding.textViewYerBilgisi.setText(geziRehberBilgileri.yerAdi);
        activityGeziRehberiBinding.textViewUlkeBilgi.setText(geziRehberBilgileri.ulkeAdi);
        activityGeziRehberiBinding.textViewSehirbilgi.setText(geziRehberBilgileri.sehirAdi);
        activityGeziRehberiBinding.textViewTarihce.setText(geziRehberBilgileri.tarihce);
        activityGeziRehberiBinding.textViewHakkinda.setText(geziRehberBilgileri.hakkinda);
    }else{
        activityGeziRehberiBinding.textViewYerBilgisi.setText(geziRehberBilgileri.placeName);
        activityGeziRehberiBinding.textViewUlkeBilgi.setText(geziRehberBilgileri.countryName);
        activityGeziRehberiBinding.textViewSehirbilgi.setText(geziRehberBilgileri.cityName);
        activityGeziRehberiBinding.textViewTarihce.setText(geziRehberBilgileri.history);
        activityGeziRehberiBinding.textViewHakkinda.setText(geziRehberBilgileri.about);
    }
}
}
}

```

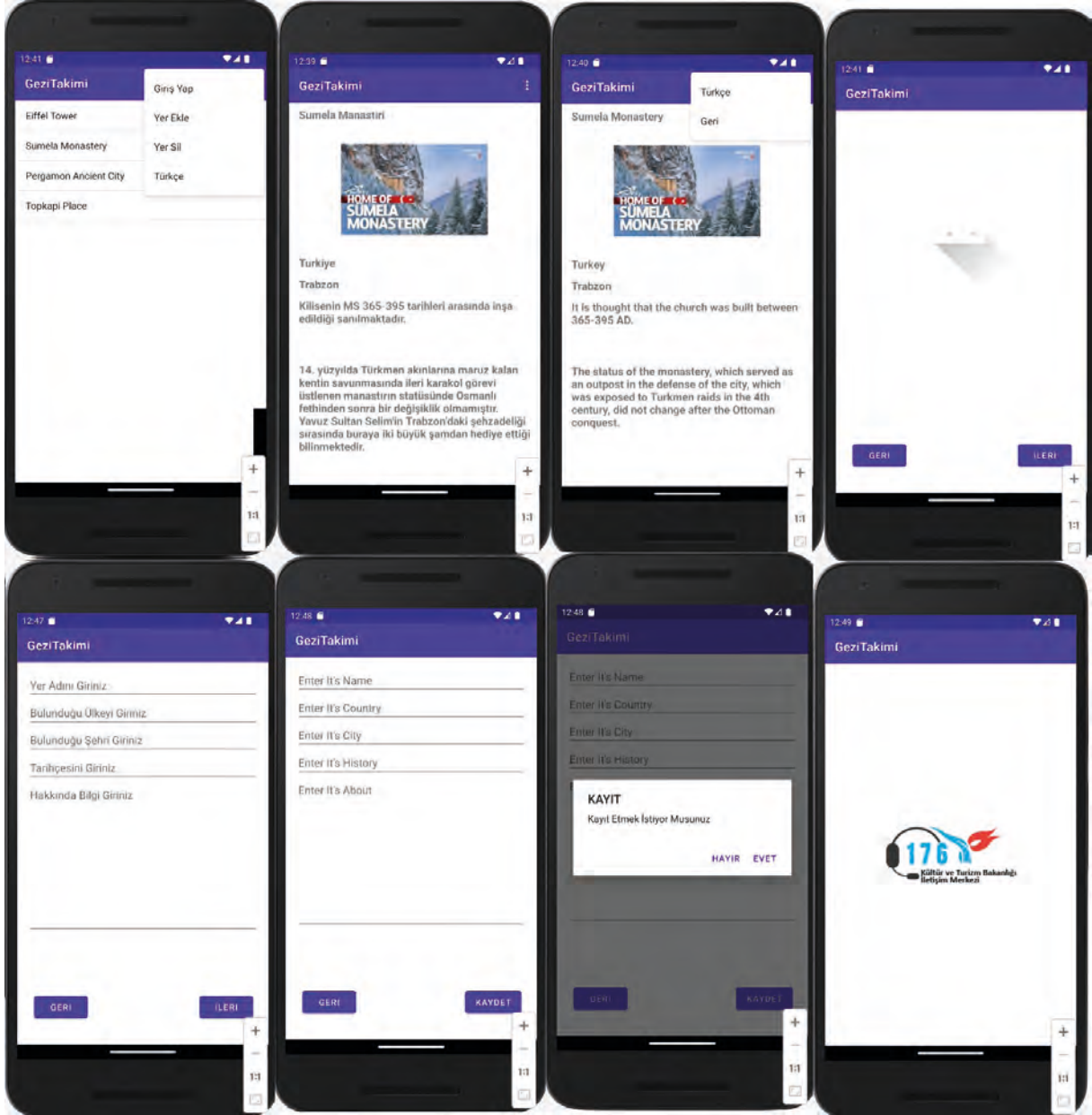


NOT

Projenin tüm kaynak kodları uygulamanın içinde sırasıyla verilmiştir. Bu kodların bir anda yazılma ihtimali asla mümkün değildir. Bunun için sırasıyla activityler arası geçişler takip edilmeli ve bu geçişlerden sonra KayitActivity içi doldurulmalıdır. Firebase içine veri girilmeden ListView yapısında verilerin görülmesi mümkün değildir. Firebase yönetim panelinde kayıtların yapıldığından emin olunduktan sonra MainActivity içinde bulunan ListView'e veriler çekilmeli ve verilerin görüntülenmesi sağlanmalıdır. Görüntülenmede bir problem yaşanmaması durumunda veriye tıklanarak, detayların yer aldığı GeziRehberiActivity içindeki kodlar yazılarak her bir komponente ilgili veriler getirilmelidir.

3.4. MOBİL UYGULAMANIN TEST VE BAKIMI

Kodlamanın bitmesi, uygulamanın tam olarak hazır olduğu anlamına gelmez. Yazılan uygulamanın türlü şekillerde testlere tabi tutulması gerekir. Hatta uygulamanın yazılımcısı, bu testleri yaparken sorun çıkarmaya çalışan bir kullanıcı rolü üstlenerek nerede, nasıl problem yaratabileceğini düşünmeli ve problem üretmeye çalışmalıdır. Bu sayede olası problemler, uygulama yayımlanmadan önce ortaya çıkar ve sorun henüz başındayken çözülebilir. Bu uygulama için test yapıldığı zaman Görsel 3.29'da görülen modül ortaya çıkar.



Görsel 3.29: Test ve bakım

Görsel üzerinde de görülebileceği gibi kaydedilen yerlerin hem İngilizce hem Türkçe isimleri seçime göre gösterilir. Yerlerden birine tıklandığında gösterilen dil seçeneğine göre detaylarının olduğu sayfa açılır. Ayrıca açılan detay sayfasından da dil değiştirilebilir ve geri olarak tıklandığında anasayfaya dönülüp son seçilen dile göre listeleme yapılır. Logo ekranında sorun bulunmaz. İstenen boyutta ve sürede ekranda kalır ve sonrasında da anasayfa ekrana gelir. "Yer ekle" bölümündeki kayıt girişlerinde bir problem olmamakla birlikte girilen bilgilerin **boş girilememesi** esas alınmalıdır. Boş giremeyecek kayıt için butonların aktif



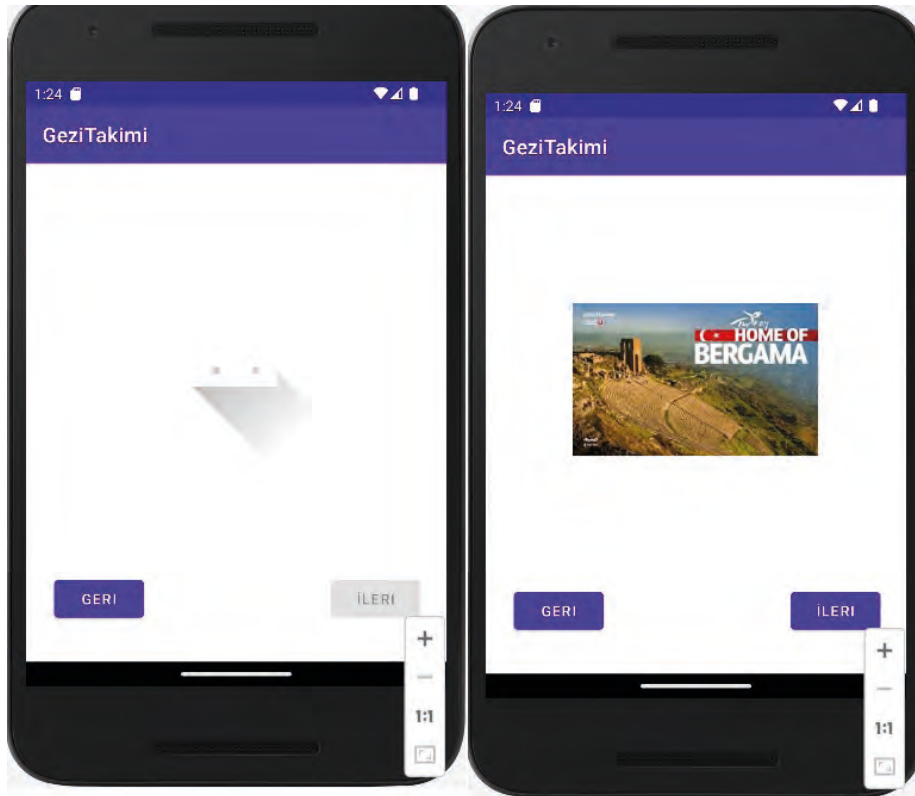
hâlde bırakılması proje açısından eksiklik olarak görülür. Öncelikle fotoğraf için “boş” seçili olduğunda İleri butonunun tıklanma özelliğini kapatmakla işe başlanabilir. Bunun için de **KayitActivity.java** dosyası içinde onCreate metoduna şu kodlar yazılır:

```
if(FotoKayitFragment.imageData==null && pageNumber==0){
    kayitBinding.butonIleri1.setEnabled(false);
}
else{
    kayitBinding.butonIleri1.setEnabled(true);
}
```

Bu sayede fragment ilk açıldığında veya bilgi dolumundan geri döndüğünde FotoKayitFragment içindeki imageData değişkeninin durumuna göre İleri butonu aktif veya pasif yapılır. Uygulama çalışırken fotoğraf seçimi yapıldığında da bu butonun durumunun değişmesi istendiği için KayitActivity’nin içine onResume metodu (yaşam döngüsü) eklenir. Yaşam döngüsüne yukarıdaki kod tekrar ettirilir.

```
@Override
protected void onResume() {
    super.onResume();
    if(FotoKayitFragment.imageData==null && pageNumber==0){
        kayitBinding.butonIleri1.setEnabled(false);
    }
    else{
        kayitBinding.butonIleri1.setEnabled(true);
    }
}
```

Uygulama tekrar test edildiğinde açılan anasayfadan “Yer Ekle” menüsüne tıklanır. Görsel 3.30’da görülen durum ortaya çıkar. Fotoğraf seçilmediğinde İleri butonu pasif, seçildiğinde ise aktif duruma geçer.



Görsel 3.30: Butonun aktif ve pasif durumu



SIRA SİZDE

Türkçe ve İngilizce veri girişlerinin olduğu bölümlerdeki butonların EditTextler içine bilgiler girilmeden aktif olmamasını sağlayınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. TrBilgiFragment içindeki İleri butonunun, Türkçe bilgi girişleri tamamlanmadan aktif olmamasını sağladı.		
2. EngBilgiFragment içindeki Kaydet butonunun, İngilizce bilgi girişleri tamamlanmadan aktif olmamasını sağladı.		
3. Geri butonuna tıklayarak önceki bilgi giriş ekranlarına döndüğünde bilgiler yerinde duruyorsa İleri butonlarının aktif kalmasını sağladı.		

3.5. MOBİL UYGULAMANIN RAPORU

Yazılan uygulamanın rapor evresine gelindikten sonra uygulamada eksik bir noktanın kalmamış olması önemlidir. Bu bölümde hem uygulamanın teslim edileceği müşteriye hem de uygulamayı kullanacak kullanıcı kitlesine uygulamanın kullanımı, avantajları, sınırlılıklarının tanıtılması ve onlarla uygulama ile ilgili tutulan raporların paylaşılması gerekir. Bu çerçevede uygulama kullanılırken alınan fotoğraf, video ve grafikler önemli yer tutar. Ayrıca müşteriye tanıtma aşamasında, uygulama hazırlama esnasında karşılaşılan problemlerin not alınarak çözüme kavuşturulduğunun belirtilmesi müşterinin o durumla ilgili endişelerini giderir.



ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. () Bir yazılım projesi için teklif alırken sadece sözlü anlaşma yapmak yeterlidir.
2. () Yazılım projesi geliştirme süreci sadece kodlamadan meydana gelen sistematik bir yapıdır.
3. () Belirli bir yazılım projesi için müşteri ile anlaşma sağlandığında bu anlaşmayı yazılı hâle dökmek ve belirli bir katora almak gerekir.
4. () Yazılım geliştirme ekibinin tamamını birbirinden farklı ve birbiriyle hiç çalışmamış kişilerden oluşturmak motivasyonu artırır.
5. () Firebase Cloud serverda veri tabloları ilişkisel bir biçimde tutulur.
6. () Firebase Cloud server IOS, Android ve Web projeleri gibi birçok farklı türde projeye destek verir.
7. () Yazılım geliştirme sürecinde tasarım, kodlamadan ayrı düşünülmediği için tasarım, kodlama ile birlikte sistematik olarak geliştirilmelidir.
8. () Bir müşteriden yazılım geliştirme projesi isteği alındığında proje takvimi oluşturmaya gerek duyulur.
9. () Tasarım aşaması, "Mantıksal Tasarım" ve "Fiziksel Tasarım" olarak iki gruba ayrılır.
10. () Proje risk analizinde tüm riskler, risk derecesi belirlenmeden listelenmeli ve hepsine eşit yaklaşılmalıdır.



4. ÖĞRENME BİRİMİ

WEB TABANLI UYGULAMA PROJESİ HAZIRLAMA



KONULAR

- 4.1. WEB TABANLI UYGULAMA PROJESİNİN PLANLANMASI
- 4.2. WEB TABANLI UYGULAMA PROJESİNİN TASARLANMASI
- 4.3. WEB TABANLI UYGULAMA PROJESİNİN KODLANMASI
- 4.4. WEB TABANLI UYGULAMA PROJESİNİN TEST VE BAKIMININ YAPILMASI
- 4.5. WEB TABANLI UYGULAMA PROJESİNİN RAPORUNUN OLUŞTURULMASI

NELER ÖĞRENECEKSİNİZ?

- Web tabanlı uygulama projesi planlama adımlarını gerçekleştirme
- Web tabanlı uygulama projesi tasarım adımlarını gerçekleştirme
- Web tabanlı uygulama projesini kodlama
- Web tabanlı uygulama projesinin test ve bakımını yapma
- Web tabanlı uygulama projesinin raporunu yazma

TEMEL KAVRAMLAR

Bakım, kodlama, planlama, tasarım, test ve rapor, web tabanlı uygulama projesi,

HAZIRLIK ÇALIŞMALARI

1. Web tabanlı yazılım projeleri hangi amaçlarla geliştirilir? Arkadaşlarınızla tartışınız.
2. Web kelimesinin ne anlama geldiğini sınıfta tartışınız.



4.1. WEB TABANLI UYGULAMA PROJESİNİN PLANLANMASI

Web tabanlı e-tercih uygulamasının planlanmasına geçilmeden önce bir yazılım projesinin planlama öncesi ihtiyaç duyulan bilgileri bir kez daha hatırlanmalıdır. Tüm projelerde olduğu gibi yazılım projelerinde de işin hedeflenen bütçe ve zaman dâhilinde bitirilmesi önem arz eder. Bunun için iyi bir proje planına ihtiyaç duyulur.

Proje planı, proje yöneticisi ve proje ekibine projenin bütçe dâhilinde ve zamanında bitirilmesini sağlayan bir yol haritasıdır. On temel ilkeye uyularak mükemmel bir proje planı hazırlamak mümkün olabilir. Bu ilkeler şöyle sıralanabilir:

1. Proje iş akışını tanımlamak
2. Proje planının kapsamına karar vermek
3. Süreç ve proje çıktıları netleştirmek
4. İş yükünü tahmin etmek
5. İş yükü tahminlerini sorgulamak
6. İlerlemeyi izlemek için kilometre taşları oluşturmak
7. Kaynakları tahmin etmek
8. Değişikliklere yer bırakmak
9. Hizmete geçişi planlamak
10. Proje sonrasını incelemek

4.1.1. Yazılım Geliştirme Planı

Yazılım geliştirme planı, bir gereksinim analiz dokümanının çalışan bir yazılıma nasıl dönüştürüleceğini ana hatlarıyla belirtir. Bu plan; planlama, fikir oluşturma, geliştirme, belgeleme, dağıtım, başlatma ve bakım aşamalarını kapsar. Bir yazılım geliştirme planı oluşturmaya başlamak için şunlara benzer bazı soruları yanıtlamak gerekir:

1. Hangi teknolojiler gerekecektir?
2. Projeyi kim yönetecektir?
3. Hangi ekipler ve kaynaklar dâhil edilecektir?
4. Kilit paydaşlar kimlerdir?
5. Dış bağımlılıklar nelerdir?
6. Başarı ölçütleri nelerdir?
7. Zaman çizelgeleri nelerdir ve bunlar nasıl tahmin edilmiştir?
8. Tahmini maliyetler nelerdir ve bunları hesaplamak için hangi varsayımlar kullanılmıştır?

4.1.2. Yazılım Geliştirme Planına Duyulan İhtiyaç

Bir müşteriye özel bir yazılım teklif edildiğinde bu teklifin bir parçası olarak ürünün ne zaman teslim edileceği ve fiyatının ne kadar olacağı belirtilmelidir. Bir yatırımcı, destek olacağı yazılım projesine yapacağı destek miktarını önceden bilmelidir. Birçok proje anlaşmasında ürünün teslimat tarihinin geciktirilmesi durumunda cezalar öngörülmüş olabilir. Bu cezalardan kaçınıp projeyi zamanında teslim edebilmek için de yazılım geliştirme planına ihtiyaç duyulur.

İyi bir yazılım geliştirme planı şu altı soruya yanıt vermelidir:

1. Yazılım projesinde kimler yer alacaktır?
2. Proje niçin tamamlanmalıdır?
3. Projede ne amaçlanmaktadır?
4. Projenin ne zaman teslim edilmesi gerekir?
5. Projedeki önemli kilometre taşları nelerdir?
6. Proje hangi yaklaşım, teknolojiler ve araçlar kullanılarak teslim edilecektir?



Bu sorulara yanıt olacak bilgiyi elde edebilmek için iyi bir proje başlangıç toplantısı yapılmalıdır. Proje başlangıç toplantısına katılan tüm paydaşlar, proje yaşam döngüsü boyunca sürece dâhil edilmelidir.

Bir yazılım geliştirme planı oluşturmak için girdiler ve istenen çıktılar belirlenmelidir. Girdiler, önerilen proje için gerekli olan malzeme ve bilgilerdir. Çıktılar ise elde edilmesi planlanan hususlardır. Bu parametreleri belirlemek için projenin paydaşlarıyla bir veya birkaç toplantı yapılmalıdır. Bu toplantılar, şu soruların cevaplandırılmasını sağlar:

- Bu projeye ilgili olarak hâlihazırda hangi materyaller hazırlanmıştır?
- Başka hangi keşif ve belge çalışmalarının yapılması gerekir?
- Bir yazılım geliştirme planında hangi ayrıntılar gereklidir?

4.1.3. Proje İş Akışının Tanımlanması

Üst düzey bir proje iş akış taslağı oluşturularak proje planı başlatılır. Baştan sona projenin ana aşamaları düşünülür. Örneğin başlatma, planlama, gerçekleştirme, teslimat ve kapanış şeklinde bir proje akış taslağı oluşturulabilir. Proje ayrıntılarına geçilmeden önce üst düzey proje iş akışı gerçekleştirilmelidir.

4.1.4. Proje Planının Kapsamına Karar Verilmesi

Proje planlarının kapsamı, projenin tamamı veya bir kısmı olabilir. Bu durum, projenin ne kadar büyük olduğuna bağlıdır. Altı ay içinde tamamlanacak herhangi bir proje baştan sona planlanmalıdır ancak altı aydan daha uzun sürmesi beklenen bir projenin aşamalı olarak planlanması önerilir. Bir proje ne kadar uzun sürerse başlangıç tahminlerinin yanlış olma olasılığı o kadar yüksektir. Proje planı ne kadar iyi düşünülmüş olursa olsun, projeyi uzun vadede kümülatif olarak etkileyebilecek bilinmeyenleri hesaba katmak zordur. Bu nedenle projeyi altışar aylık zaman dilimlerine ayırarak planlamak, proje planına uymayı kolaylaştırır.

4.1.5. Proje Sürecinin ve Çıktılarının Netleştirilmesi

Proje süreci fazla ilerlemeden proje çıktılarının ne olacağına karar verilmelidir. Başarılı bir son durumun ne olacağının açıkça tanımlanması gerekir. Tüm paydaşlar bu konuda hemfikir olmalıdır. Bunun yapılmaması ileride proje kapsamında önemli değişiklik taleplerine neden olabilir. Bu tür değişiklikler projede maliyetli aksamalara yol açar.

4.1.6. Projenin İş Yükünün Tahmin Edilmesi

Proje geliştirme başarısını elde etmek için gereken işi tahmin etmek önemlidir. En doğru iş tahmininde bulunabilmek, ekibin tüm üyelerinin kendi tahminlerini paylaşmasıyla mümkündür.

4.1.7. Projenin İş Yükü Tahmininin Sorgulanması

İyi bir tahmin projeyi başarıya götürebileceği için kötü bir tahmin de projeyi daha başlamadan mahvedebileceği için tahmin süreci boyunca doğru soruları sormak önemlidir. Açık uçlu sorularla tahmin kısıtlamaları ortaya çıkarılmalıdır. İş yükünü tahmin etmek için kullanılan yöntem ne olursa olsun, her zaman hata payı bırakılmalıdır. Öngörülemez karmaşıklık veya ek kapsam nedeniyle beklenmedik bir duruma izin vermek, projenin eksik teslim tarihlerinden korunmasına yardımcı olabilir. Tüm projeler için %10 ile %25 arasında beklenmedik durum oluşturulması önerilir.

4.1.8. Projenin Kilometre Taşlarının Oluşturulması

Müşteri, özellikle kurum sözleşmelerinde proje ilerlemesini takip etmek için kilometre taşları oluşturmayı faydalı bulabilir. Projenin kilometre taşları “Tasarım aşaması tamamlandı.”, “Prototip oluşturuldu.” veya “İlk ürün teslim edildi.” şeklinde olabilir. Bu kilometre taşlarına sahip olmak, müşteri ve paydaşların teslimat zaman çizelgelerini daha düzenli kontrol etmelerini kolaylaştırabilir. Projenin kilometre taşlarını önceden belirlemek, proje geliştirme ekibinin hazırladığı ürünü mümkün olduğunca erken bitirmeye odaklanmasına yardımcı olur.



4.1.9. Projenin Kaynaklarının Tahmin Edilmesi

Proje için gereken iş yükü ve kilometre taşları belirlendikten sonra projenin farklı aşamalarında hangi kaynakların gerekli olacağını tahmin etmeye başlamak mümkündür. Geliştirme ekibi oluşturulurken her ekip üyesinin dâhil edilmesi gereken proje noktalarını da belirlemek gerekir. Muhtemelen projenin tamamı boyunca her kaynağa ihtiyaç duyulmayacaktır. Örneğin ilk yazılım geliştirme planlama aşaması ve tasarım aşaması, tasarım ekibinden ve paydaşlardan daha çok yararlanmayı gerektirirken geliştirme ekibinin bu süreçte fazla bir işi olmayacaktır. Diğer yandan da tasarım ekibinin geliştirme aşamasında fazla bir iş yükü bulunmayacaktır.

4.1.10. Projede Değişikliklere Yer Bırakılması

Ne kadar iyi bir planlama yapılırsa yapılsın, bir şeylerin ters gitmesi her zaman muhtemeldir. Bu nedenle süreç içinde ters gitme ihtimali olan durumları tahmin etmeye çalışıp onlara karşı önceden tedbir almak, proje planının düzgün yürütmesine yardımcı olacaktır.

4.1.11. Projenin Hizmete Geçişinin Planlanması

Projenin teslimatına kadar kesintiye uğramadan işlemiş bir plana sahip olursa da yazılımın tesliminden sonra da yapılması gereken bazı işler vardır. Bir yazılım ürününün hizmete girmesinden sonra bir destek ekibinin planlanması da bu süreç için iyi bir başlangıç olabilir. Tamamlanan sistemi veya ürünü, bakımını yapacak ekibe teslim etmek için yeterli zaman ve kaynağa ihtiyaç vardır. Yazılım ürününün kullanılmaya başlanmasından sonra iyi bir destek ve bakım, müşteriyle olan ilişkiyi doğru bir şekilde sürdürmek açısından çok önemlidir.

4.1.12. Proje Sonrası İnceleme Yapılması

Proje ekipleri, ürün tesliminden sonra genellikle hızlı bir şekilde dağıtılır ve başka projelere tahsis edilir. Bu nedenle teslim sonrası bir proje genellikle ihmal edilir. Proje teslim sonrasının da gözlemlenmesi için bazı kaynakların tahsisi proje planına dâhil edilmelidir. Proje sonrası inceleme sırasında bilgi toplamak ve geri bildirimleri almak için şu açık uçlu soruların sorulması faydalıdır:

- Bu proje sırasında hangi dersler çıkarıldı?
- İyi giden ve gelecekteki projelerde tekrarlanması öngörülen şeyler nelerdi?
- Beklendiği gibi gitmeyen şeyler nelerdi? Gelecekteki projelerde bunlardan kaçınmak için ne yapılabilir?
- Proje hedeflerine ulaşıldı mı?
- Proje işletmeye ne gibi faydalar sağladı?

4.1.13. Risk Yönetimi

Risk yönetiminin temel amacı, risklerin oluşması durumunda negatif etkiyi en aza indirmektir. Bu amaç doğrultusunda bakıldığında risk yönetiminin faydaları, doğrudan sağlanan faydalar (birincil faydalar) ve dolaylı sağlanan faydalar (ikincil faydalar) olmak üzere iki grupta toplanabilir. Doğrudan sağlanan faydaların başında hedeflerin tam olarak karşılanması gelirken proje büyük risklerden de korunur. Risk yönetiminin iyi yapıldığı projelerde proje takımı ve tüm paydaşları oluşabilecek problemleri çözmeye hazırlıklıdır. Böylece kriz yönetimi pratikleri cesaretlendirilir. Doğrudan sağlanan bir diğer fayda da proje sonunda ortaya çıkan ürününün daha güvenilir olmasıdır. Ürün kalitesinin artmasıyla beraber düşük kaliteden kaynaklanan maliyet de düşer. Hedeflerin belirlenmesinden süreçlerin iyileştirilmesine kadar tüm süreç alanlarında sağlanan faydalar ise ikincil faydalar olarak sınıflandırılır. Proje yöneticisinin en önemli görevlerinden biri olan risk yönetimi, projenin zamanını veya geliştirilen yazılımın kalitesini etkileyecek risklerin belirlenmesi ve kontrol edilmesini kapsar.

Proje yönetim sürecinin alt adımlarından biri olan risk yönetimi şu dört aşamadan oluşur:

- Riskin tanımlanması
- Riskin planlanması
- Risk analizinin yapılması
- Riskin izlenmesi



4.1.13.1. Riskin Tanımlanması

Bu aşama, iç ve dış çevrelerde genel ve açık uçlu araştırmaya dayanır. Buradaki risk araştırmasının kapsamı, projenin kendi iç dinamiklerinin yanı sıra projeyi etkileyen tüm iş çevresinden kaynaklanan şartları tespit etmektir. Projenin hem iç hem de dış etkenlerden kaynaklanan riskleri ele alan birinci tip yaklaşımlar arasında beyin fırtınası, zihin haritalama ve benzerlik gibi sezgisel metotlar kullanılır. Ayrıca ilk on risk, risk kontrol listesi ve sınıflandırmaya dayalı anket gibi geçmişe dayalı metotlar da birinci tip yaklaşımlar olarak ele alınır. Riskleri belirlemek kolay bir süreç olmadığı için bunları proje yöneticisi tek başına değil, proje ekibi ile birlikte çalışmalıdır.

4.1.13.2. Risk Analizinin Yapılması

Risk değerlendirme olarak da adlandırılan risk analizi safhasında risklerin olma olasılığı ve proje üzerindeki negatif etkisi belirlenir. Riskler analiz edilip önceliklendirilirken bu iki değer çarpımı sonucu elde edilen risk değerine göre karar verilir. Bu işlemde "Risk Değeri = Olma Olasılığı x Etkisi" formülü kullanılır.

4.1.13.3. Riskin Planlanması

Risk planlama süreci, risk yönetimindeki dengenin en verimli şekilde ele alınmasıdır. Risk planlama safhasında belirlenen ve analiz edilen her riskin yönetimi için stratejiler geliştirilir. Bu stratejiler, riskin oluşması durumunda projeye olan etkisini azaltmayı hedefler. Risklere karşı alınan önlemler dört başlık altında toplanabilir:

- **Kaçınma:** Riski yaratan nedenlerin ortadan kaldırılmasıdır.
- **Aktarma:** Riskin üstesinden gelebilecek kişilere aktarılmasıdır.
- **Azaltma:** Riskin etkisinin kabul edilebilir bir seviyeye indirilmesidir.
- **Kabullenme:** Riskin etkisinin tüm proje paydaşlarının kabul edilmesidir.

4.1.13.4. Riskin İzlenmesi

Proje boyunca devam eden risk izleme sürecinde belirlenen risklerin ve risklere yönelik yapılan tahminlerin doğru olup olmadığı kontrol edilir. Sadece projenin başında belirlenen riskler değil, projenin diğer aşamalarında da risk kaynakları tahmin edilerek yeni oluşabilecek riskler izlenmelidir.



1. UYGULAMA

Web Tabanlı E-tercih projesi için planlama adımlarını gerçekleştiriniz.

1. Adım: Müşterinin temel ihtiyaçlarını belirleyiniz. Müşteri şunları talep etmiştir:

- Kullanıcıların kendilerine sunulan seçenek listesini tercih sırasına koyup kaydedebileceği bir web uygulamasına ihtiyacı var.
- İnternet üzerinden ulaşılacak bir web uygulaması olmalıdır.
- Tercihler önceden belirlenebilmeli, kullanıcı bu tercih listesini sıraya koyarak kaydedebilmektedir.

2. Adım: Projenin sınırlarını belirleyiniz.

- Proje; tercih listesi ekleme, silme, güncelleme, listeleme ve arama gibi işlemleri içerecektir. Ayrıca tercih listesini sıraya koyarak kaydetmeye de imkân vermelidir.

3. Adım: Proje uygulama takvimini veya proje planını oluşturunuz.



Müşterinin temel ihtiyaçları öğrenilip projenin sınırları belirlendikten sonra Tablo 4.1'deki gibi bir proje uygulama takvimi veya proje planı hazırlanır.

Tablo 4.1: Proje Uygulama Takvimi

No	Proje Bölümü	Yapacak Personel	Süre	Başlangıç	Bitiş
1	Risk yönetimi ve analizinin yapılması	Öğrenci1, Öğrenci2	2 gün	1. gün	2. gün
2	Gereksinim analizinin yapılması	Öğrenci1, Öğrenci2	3 gün	3. gün	5. gün
3	Sistem modelleme	Öğrenci1	1 gün	6. gün	6. gün
4	Varlık ilişki şemalarının oluşturulması	Öğrenci1	1 gün	7. gün	7. gün
5	Bileşen ve arayüzlerin tasarımı	Öğrenci2	2 gün	6. gün	7. gün
6	Veri akış diyagramlarının oluşturulması	Öğrenci1	1 gün	8. gün	8. gün
7	Tasarımın gözden geçirilmesi	Öğrenci2	1 gün	8. gün	8. gün
8	Veri tabanının oluşturulması	Öğrenci2	1 gün	9. gün	9. gün
9	Projenin kodlanması	Öğrenci1	2 gün	9. gün	10. gün
10	Projenin test edilmesi	Öğrenci2	1 gün	10. gün	10. gün



SIRA SİZDE

Web Tabanlı Okul Yönetim projesi için planlama adımlarını gerçekleştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Web Tabanlı Okul Yönetim projesi için müşterinin temel ihtiyaçlarını belirledi.		
2. Web Tabanlı Okul Yönetim projesinin sınırlarını belirledi.		
3. Web Tabanlı Okul Yönetim projesi uygulama takvimini oluşturdu.		
4. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
5. Zamanı verimli kullandı.		

4.2. WEB TABANLI UYGULAMA PROJESİNİN TASARLANMASI

Web tabanlı uygulama projesinin tasarımı, gereksinim analizi ve sistemin modellenmesi aşamalarını kapsar.

4.2.1. Gereksinim Analiz Aşamaları

Gereksinim analiz aşamaları şunlardır:

- Paydaşlardan ihtiyaçların toplanması
- Bu ihtiyaçları bir ürün oluşturma aşamasına dâhil ihtiyaçların uygunluğunu ve uygulanabilirliğinin anlaşılması için analiz edilmesi
- Toplanan ihtiyaçların belgelenmesi



4.2.2. Sistemin Modellenmesi

Visio veya çevrimiçi farklı tasarım araçları kullanılarak sistemin modellenmesi gerekir. Sistemin modellenmesi sırasında yazılım projesinin ekranları, veri akış şemaları, kullanım senaryosu şemaları ve iş akış diyagramlarına yer verilir.

4.3. WEB TABANLI UYGULAMA PROJESİNİN KODLANMASI

Web tabanlı e-tercih uygulamasının kodlama işlemleri yapılırken öncelikle sistemin ihtiyaç duyduğu veri tabanı tasarımı gerçekleştirilmelidir. Web tabanlı uygulama projeleri geliştirilirken aslında iki yaklaşımdan biri kullanılır. Yaklaşımlardan birincisi veri tabanı öncelikli yaklaşım, diğeri ise kod öncelikli yaklaşımdır. Veri tabanı öncelikli yaklaşımda önce veri tabanı tasarımı gerçekleştirilir, devamında Scaffolding adı verilen bir işlem yardımıyla tasarlanmış veri tabanının sınıf kodları otomatik olarak oluşturulur. Kod öncelikli yaklaşımda ise veri tabanındaki varlıklara ait sınıflar kodlanır, uygulama çalıştırıldığında veri tabanı otomatik olarak oluşturulur. Bu uygulamada birinci yaklaşım kullanılacaktır. Bir başka deyişle önce veri tabanı oluşturulacak, sihirbaz yardımıyla da veri tabanı varlıklarına ait sınıflar yaratılacaktır.

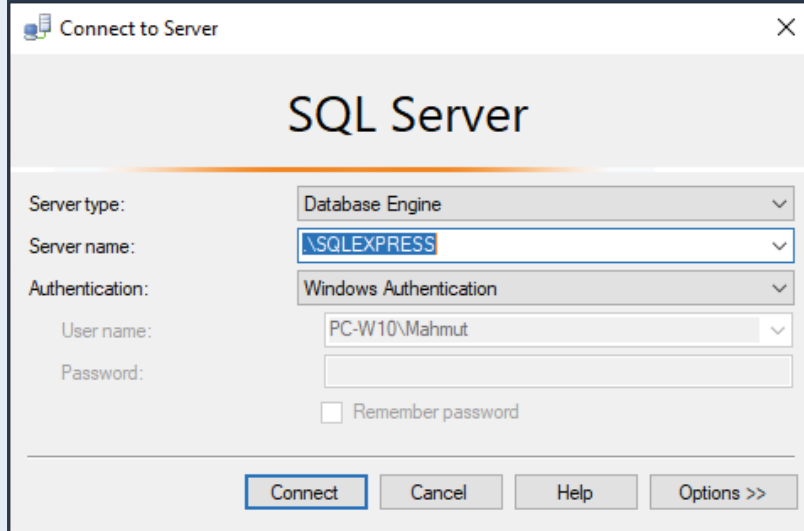


2. UYGULAMA

İşlem adımlarına göre web tabanlı e-tercih yazılım projesi veri tabanını oluşturunuz.

1. Adım: Veri tabanı oluşturma uygulamasını açınız.

Veri tabanını oluşturmak için Microsoft SQL Server Management Studio uygulaması kullanılacaktır. Uygulama bilgisayarınızda kurulu ise uygulamayı çalıştırınız. Uygulama bilgisayarınızda kurulu değilse internetten indirerek öncelikle uygulamayı kurunuz. Uygulamayı çalıştırdığınızda Görsel 4.1'deki ekranla karşılaşacaksınız.

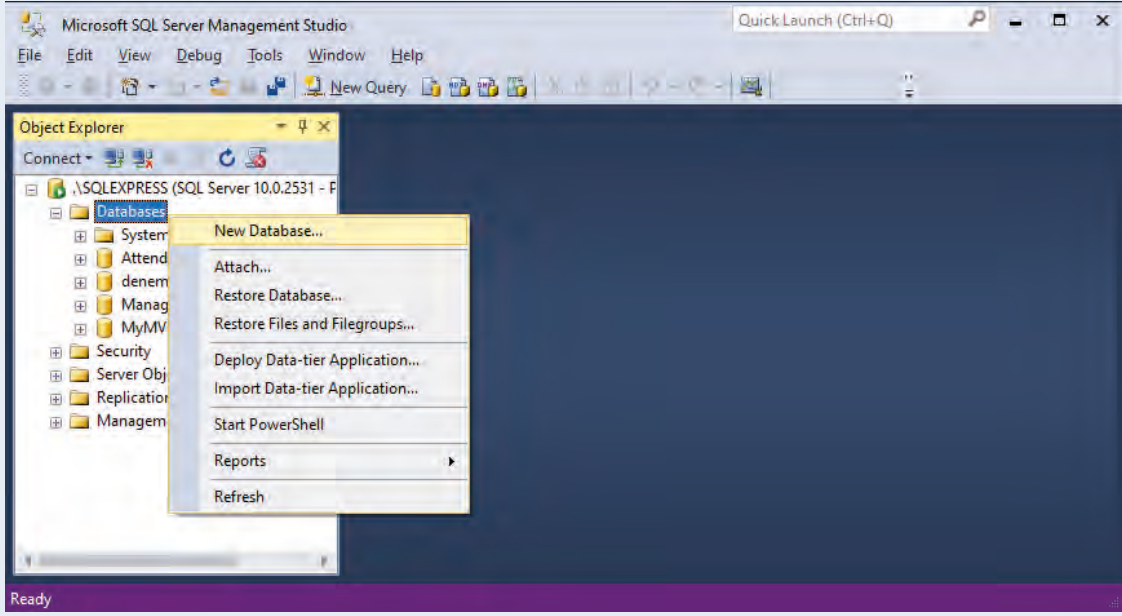


Görsel 4.1: Microsoft SQL Server Management Studio giriş ekranı

Giriş ekranında Görsel 4.1'deki girişleri yaptıktan sonra Connect düğmesine tıklayınız. Görsel 4.2'deki SQL Server Management Studio ana ekranını göreceksiniz.

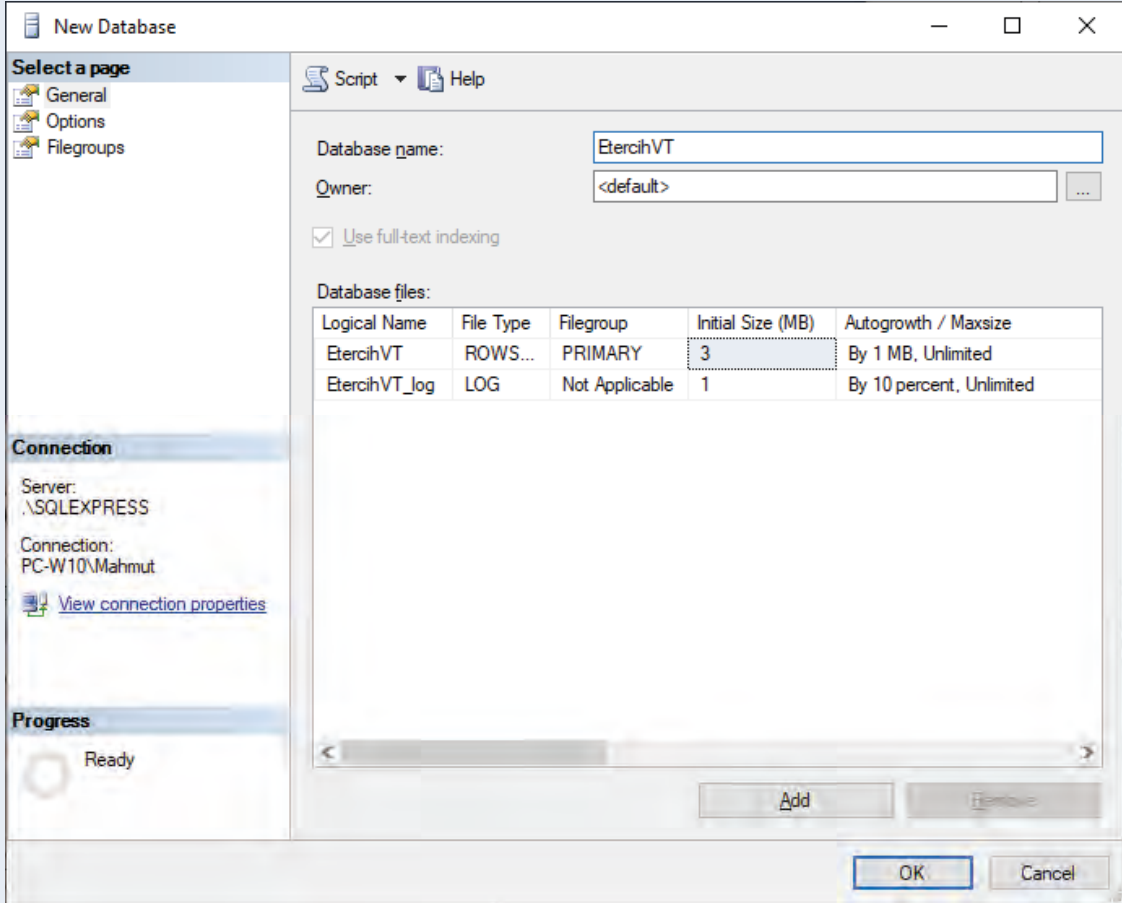
2. Adım: Veri tabanı oluşturunuz.

Görsel 4.2'deki gibi Databases seçeneğine sağ tıklayıp New Database... komutunu belirleyiniz.



Görsel 4.2: Veri tabanı oluşturma uygulaması ana ekranı

Görsel 4.3'teki veri tabanı oluşturma penceresi ekranda belirecektir.



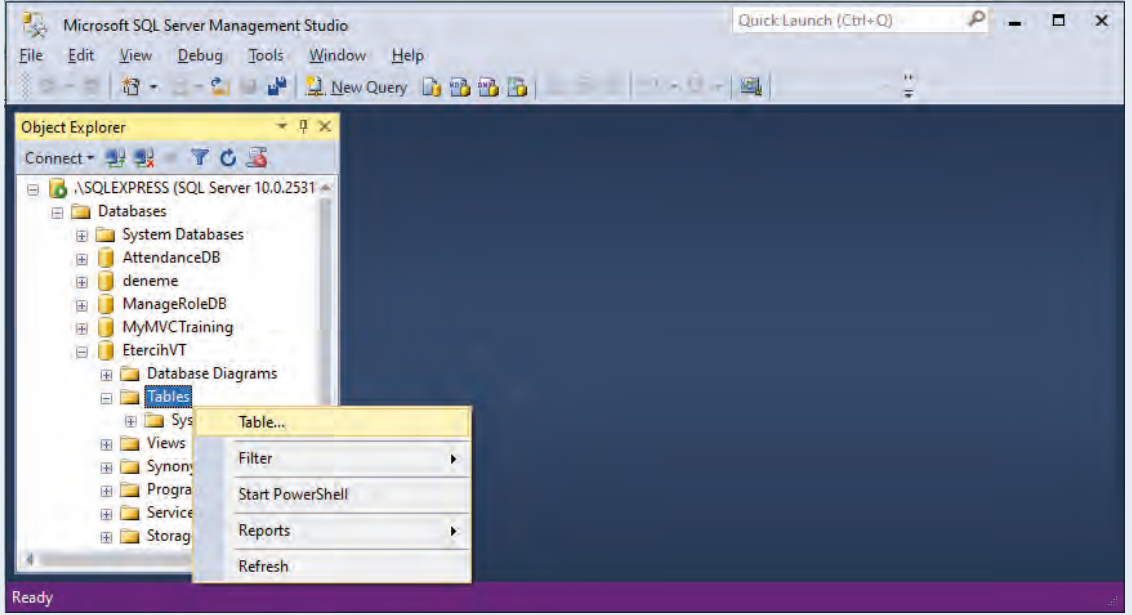
Görsel 4.3: Yeni veri tabanı oluşturma penceresi

Database name kutusuna EtercihVT yazılıp OK düğmesine tıklandıktan sonra veri tabanı dosyası oluşturulur.



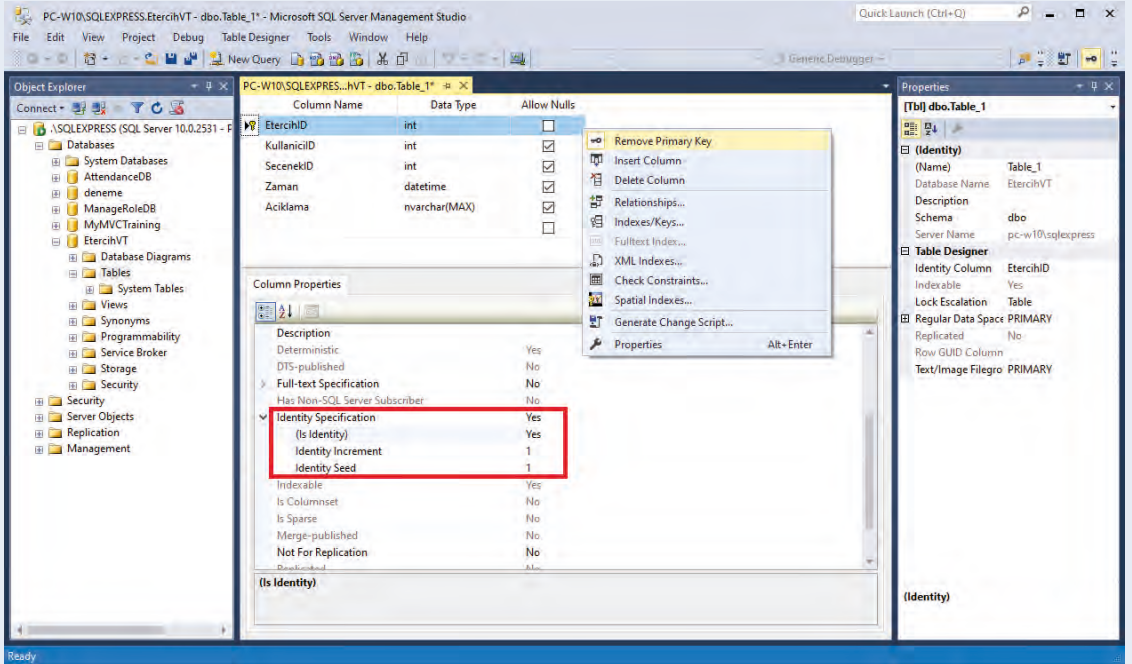
3. Adım: Tabloları oluşturunuz.

Görsel 4.4'te görüldüğü gibi Tables seçeneği üzerinde sağ tıklandığında açılan menüden Table... komutu belirlenir.



Görsel 4.4: Yeni tablo oluşturma seçeneğinin belirlenmesi

Bu durumda Görsel 4.5'teki yeni tablo oluşturma penceresi açılacaktır.



Görsel 4.5: Yeni tablo oluşturma penceresi

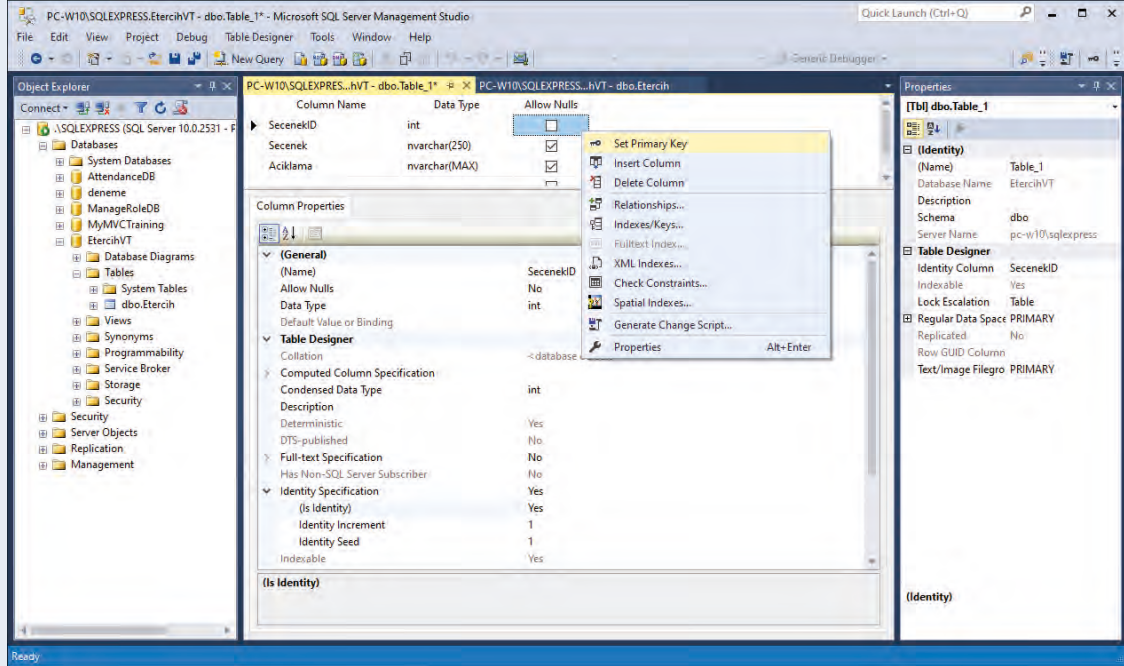
Yeni tabloyu Görsel 4.5'teki gibi oluşturunuz. İlk alanın (EtercihID) üzerinde sağ tıklayarak açılan menüden Set Primary Key seçeneğini belirlemeyi unutmayınız. Ayrıca kırmızı kutu içinde yer alan seçenekleri de Görsel 4.5'teki gibi ayarlayınız. Yapılan bu ayarlamalar, ilk alanın kayıt belirteci olarak kullanılmasını sağlayacaktır. Artık tabloyu Etercih adıyla kaydedebilirsiniz. Kaydetme işlemi araç



çubuğu üzerindeki disket resimli düğmeye tıklayarak veya klavyeden Ctrl + S tuşlarına basarak gerçekleştirebilirsiniz. Kaydetme işlemini uyguladığınızda size tablo adı sorulacaktır. İlgili kutuya Etercih yazarak onaylayabilirsiniz.

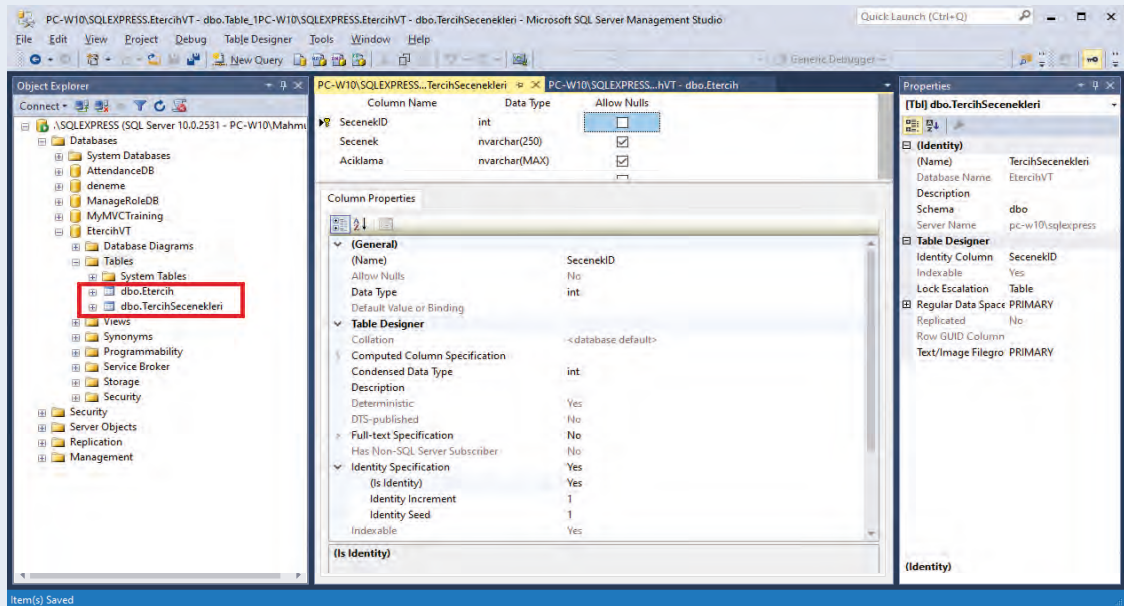
4. Adım: Tercih seçenekleri tablosunu oluşturunuz.

Görsel 4.6'da görüldüğü gibi tercih seçenekleri tablosunu ilgili ayarlamaları yaparak TercihSece- nekleri adıyla kaydediniz.



Görsel 4.6: Tercih seçenekleri tablosunun oluşturulması

Etercih web tabanlı uygulaması için gerekli olan veri tabanı yapısı oluşturuldu. Veri tabanı yapısını doğru bir şekilde oluşturduysanız Görsel 4.7'dekine benzer bir veri tabanı oluşturma uygulaması penceresi görmemiz gerekir.

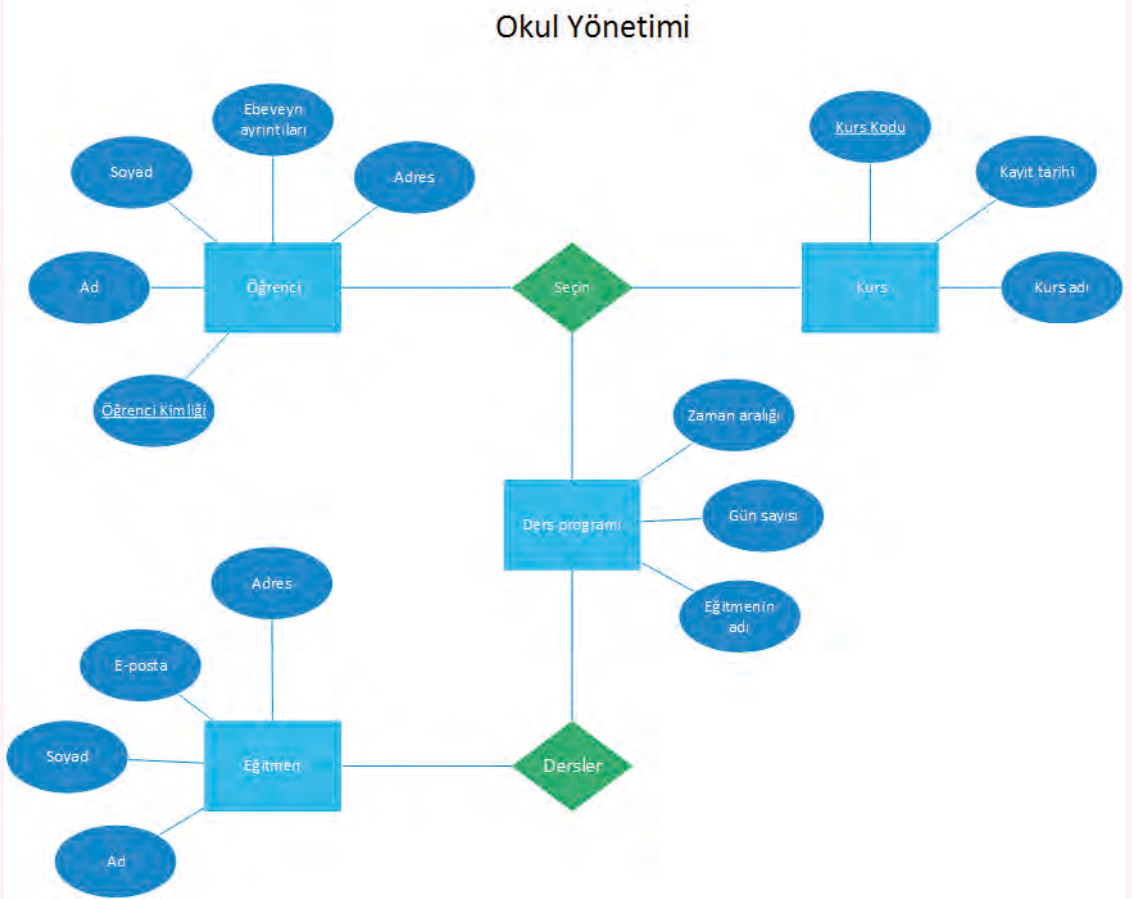


Görsel 4.7: Etercih web tabanlı uygulamasına ait veri tabanı yapısı



SIRA SİZDE

SQL Server Management Studio uygulamasını kullanarak Görsel 4.8’de varlık ilişki diyagramı görülen veri tabanı tablo yapısını oluşturunuz.



Görsel 4.8: Okul yönetimi veri tabanı varlık ilişki diyagramı

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

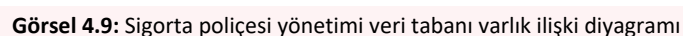
KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
2. Veri tabanı yönetim sistemi (VTYS) uygulamasına uygun bir şekilde bağlantı yaptı.		
3. Veri tabanı yönetim sisteminde Okul Yönetimi adında bir veri tabanı oluşturdu.		
4. Okul yönetimi veri tabanı içinde Öğretmen adında bir tablo oluşturdu.		
5. Okul yönetimi veri tabanı içinde Öğrenci adında bir tablo oluşturdu.		



SIRA SİZDE

Sigorta Poliçesi Yönetimi





DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

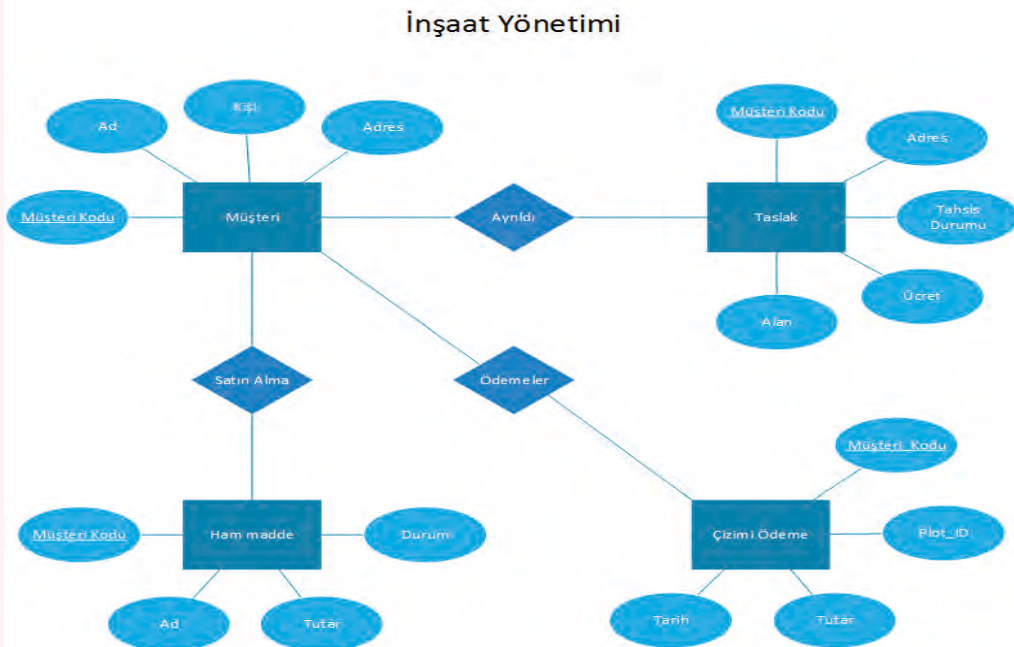
KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Veri tabanı yönetim sistemi uygulamasına uygun bir şekilde bağlantı yaptı.		
2. Veri tabanı yönetim sisteminde Sigorta Poliçesi Yönetimi adında bir veri tabanı oluşturdu.		
3. Sigorta poliçesi yönetimi veri tabanı içinde Aracı adında bir tablo oluşturdu.		
4. Sigorta poliçesi yönetimi veri tabanı içinde İlke adında bir tablo oluşturdu.		
5. Sigorta poliçesi yönetimi veri tabanı içinde Poliçe sahibi adında bir tablo oluşturdu.		
6. Sigorta poliçesi yönetimi veri tabanı içinde Lehdar adında bir tablo oluşturdu.		
7. Tablolarda uygun alanları birincil anahtar olarak tanımladı.		
8. Tabloları birbirleriyle uygun şekilde ilişkilendirdi.		
9. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
10. Zaman verimli kullandı.		



SIRA SİZDE

SQL Server Management Studio uygulamasını kullanarak Görsel 4.10'da varlık ilişki diyagramı görülen veri tabanı tablo yapısını oluşturunuz.



Görsel 4.10: İnşaat yönetimi veri tabanı varlık ilişki diyagramı



DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Veri tabanı yönetim sistemi uygulamasına uygun bir şekilde bağlantı yaptı.		
2. Veri tabanı yönetim sisteminde İnşaat Yönetimi adında bir veri tabanı oluşturdu.		
3. İnşaat yönetimi veri tabanı içinde Müşteri adında bir tablo oluşturdu.		
4. İnşaat yönetimi veri tabanı içinde Ham madde adında bir tablo oluşturdu.		
5. İnşaat yönetimi veri tabanı içinde Taslak adında bir tablo oluşturdu.		
6. İnşaat yönetimi veri tabanı içinde Çizimi Ödeme adında bir tablo oluşturdu.		
7. Tablolarda uygun alanları birincil anahtar olarak tanımladı.		
8. Tabloları birbirleriyle uygun şekilde ilişkilendirdi.		
9. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
10. Zamanı verimli kullandı.		



3. UYGULAMA

İşlem adımlarına göre web tabanlı e-tercih yazılım uygulamasının web projesini Visual Studio kullanarak oluşturunuz.

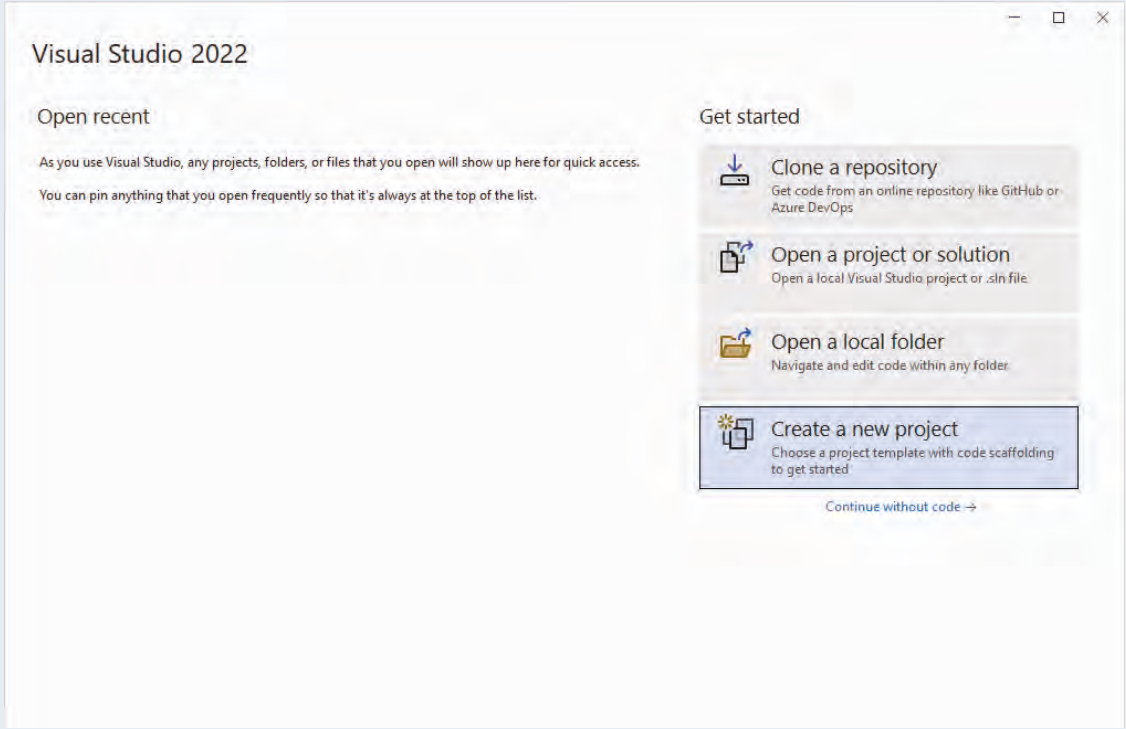


NOT

Birinci uygulamada veri tabanı oluşturulan projenin bu uygulamada web sayfa yapısı hazırlanacaktır. Web sayfa yapısı hazırlamanın farklı yaklaşımları vardır. Bu uygulamada MVC olarak kısaltılan Model, Görünüm ve Kontrolcü (**Model, View, Controller**) teknolojisi kullanılacaktır. Bu uygulama, Visual Studio uygulaması aracılığıyla gerçekleştirilecektir.

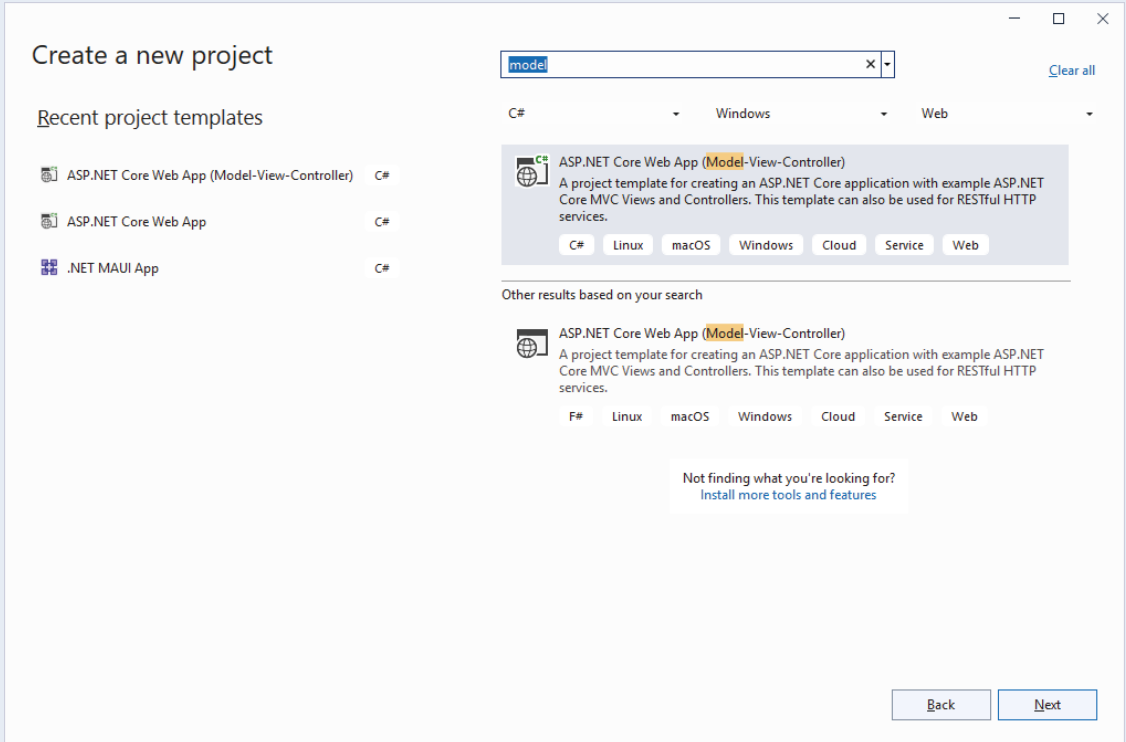
1. Adım: Visual Studio uygulamasını açınız.

Başlat düğmesine tıklayıp Visual Studio yazınız. Menüde beliren Visual Studio uygulamasını tıklayıp çalıştırınız. Uygulama çalıştırıldığında Görsel 4.11'dekine benzer bir pencere açılacaktır.



Görsel 4.11: Visual Studio 2022 açılış penceresi

2. Adım: Yeni bir proje oluşturmak için Görsel 4.11'deki pencerede Create a new project (Yeni bir proje oluşturun) düğmesine tıklayınız. Görsel 4.12'de görülen pencere açılacaktır.



Görsel 4.12: Proje şablonu arama ve seçim penceresi



3. Adım: Görsel 4.12’de görülen pencerenin arama kutusuna “model” yazınız. Bulunan sonuçlar içinden ASP.NET Core Web App (Model-View-Controller) seçeneğini belirleyip Next düğmesine tıklayınız. Görsel 4.13’te görülen pencere açılacaktır.

Görsel 4.13: Yeni proje yapılandırma penceresi

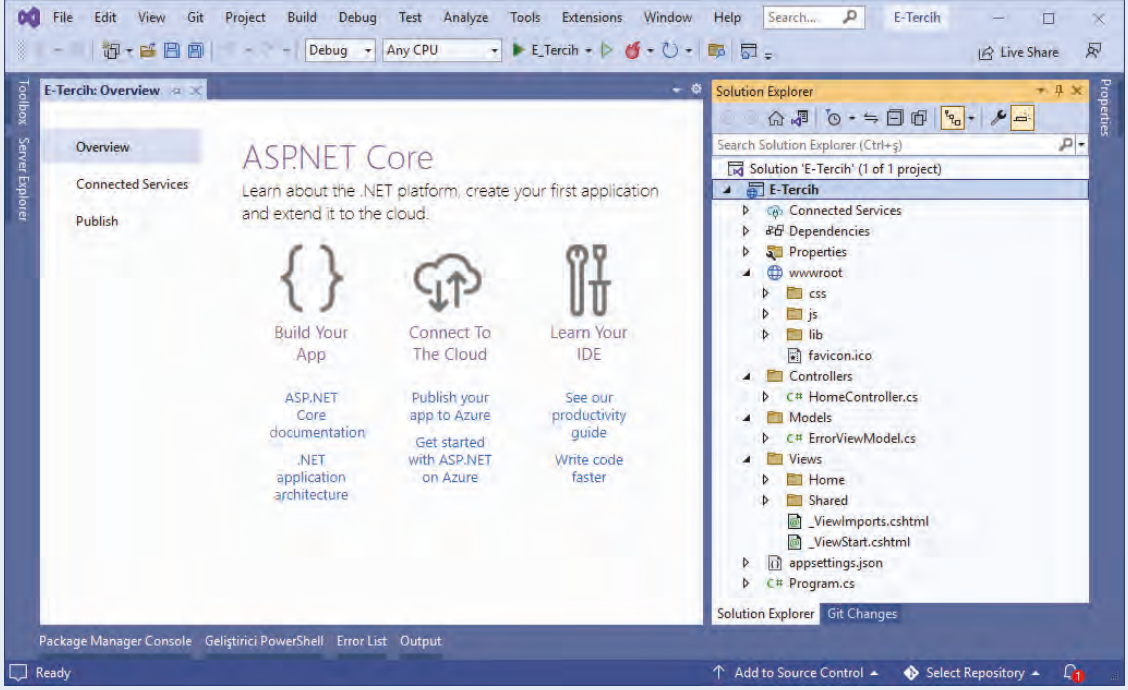
4. Adım: Görsel 4.13’teki pencerede projenin adı, projenin konumu ve çözüm adı bilgilerini girdikten sonra Next düğmesine tıklayınız. Görsel 4.14’te görülen pencere açılacaktır.

Görsel 4.14: Ek bilgi girme penceresi

5. Adım: Görsel 4.14’teki pencerede Framework, kimlik doğrulama tipi (Authentication type), HTTPS yapılandırması, Docker’in etkinleştirilmesi, Docker işletim sistemi gibi ek bilgileri belirledikten sonra



Create düğmesine tıklayınız. Projenin başlangıç yapılandırması ve dosyaları oluşturularak Görsel 4.15'te görülen pencereye benzer şekilde Visual Studio uygulaması açılacaktır.



Görsel 4.15: Visual Studio penceresinde oluşturulan projenin görünümü

Görsel 4.15'teki pencerede belirlenen seçeneklere göre oluşturulan projenin dosya yapısı görülür. Bu örnekte Model-View-Controller (MVC) şablonuna dayalı bir proje oluşturulduğu için dosya yapısında Controllers, Models, Views adında üç ayrı klasör vardır. Kontrolcülerle ilgili dosyalar Controllers klasörüne, modellerle ilgili dosyalar Models klasörüne, görünümle ilgili dosyalar ise Views klasörüne yerleştirilecektir. Genel hatlarıyla model ile veri tabanı tabloları, kontrolcü ile veriye erişimi kontrol eden sınıflar, görünümle de web sayfa yapısı ifade edilir.



4. UYGULAMA

İşlem adımlarına göre projeye Scaffolding işlemi için ihtiyaç duyulan paketleri Visual Studio içindeki NuGet paket yöneticisini kullanarak yükleyiniz.

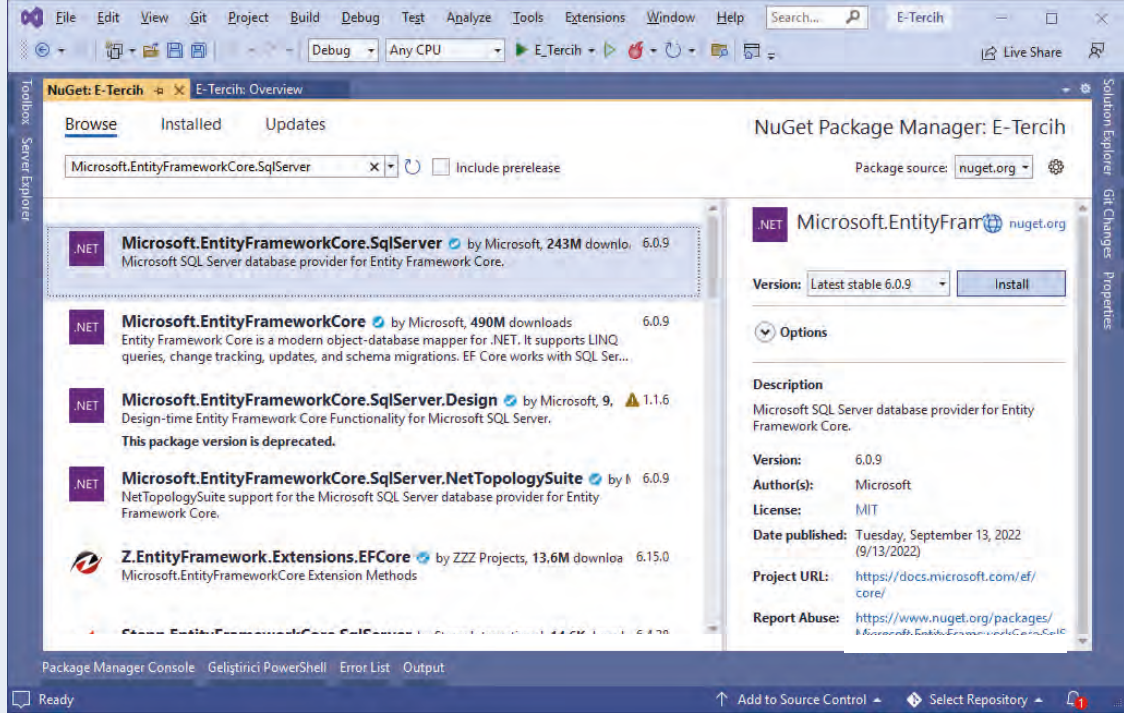


NOT

Profesyonel yazılımcılar proje için gereken tüm materyali kod yazarak oluşturabilirler. Başlangıç seviyesindeki yazılımcılar ise birtakım işlemler için otomatik kod oluşturma teknolojilerinden yararlanabilirler. Bu uygulamada bazı kod dosyalarının otomatik oluşturulması için Scaffolding adı verilen bir yöntem kullanılacaktır. Veri tabanı tablolarına karşılık gelen sınıf dosyalarını ve veri okuma, veri yazma, veri güncelleme, veri silme ile veriye erişim işlevlerini içeren sınıf dosyasını Scaffolding yöntemiyle oluşturabilmek için iki farklı pakete ihtiyaç duyulur.

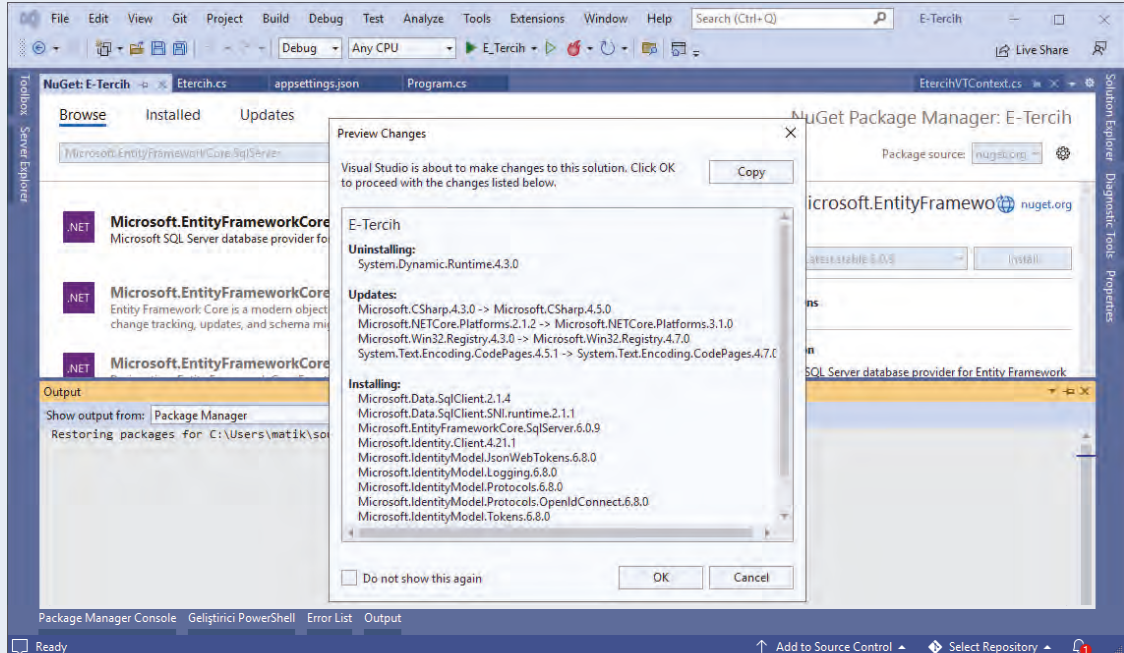


1. Adım: Görsel 4.15'teki proje dosya yapısının en üstünde yer alan E-tercih uygulama adına sağ tıklayınız. Açılan menüdeki Manage NuGet Packages... (Nuget Paketlerini Yönet...) seçeneğini belirleyiniz. Görsel 4.16'da görülen pencere açılacaktır.



Görsel 4.16: NuGet paket yöneticisi penceresi

2. Adım: Görsel 4.16'daki pencerede üstte görülen Browse yazılı bağlantıya tıklayarak arama kutusuna "Microsoft.EntityFrameworkCore.SqlServer" yazınız ve Enter tuşuna basınız. Arama sonuçları listesindeki ilk seçeneği belirledikten sonra sağ tarafta görülen Install düğmesine tıklayınız. Görsel 4.17'de görüldüğü gibi bir bilgilendirme penceresi açılacaktır.

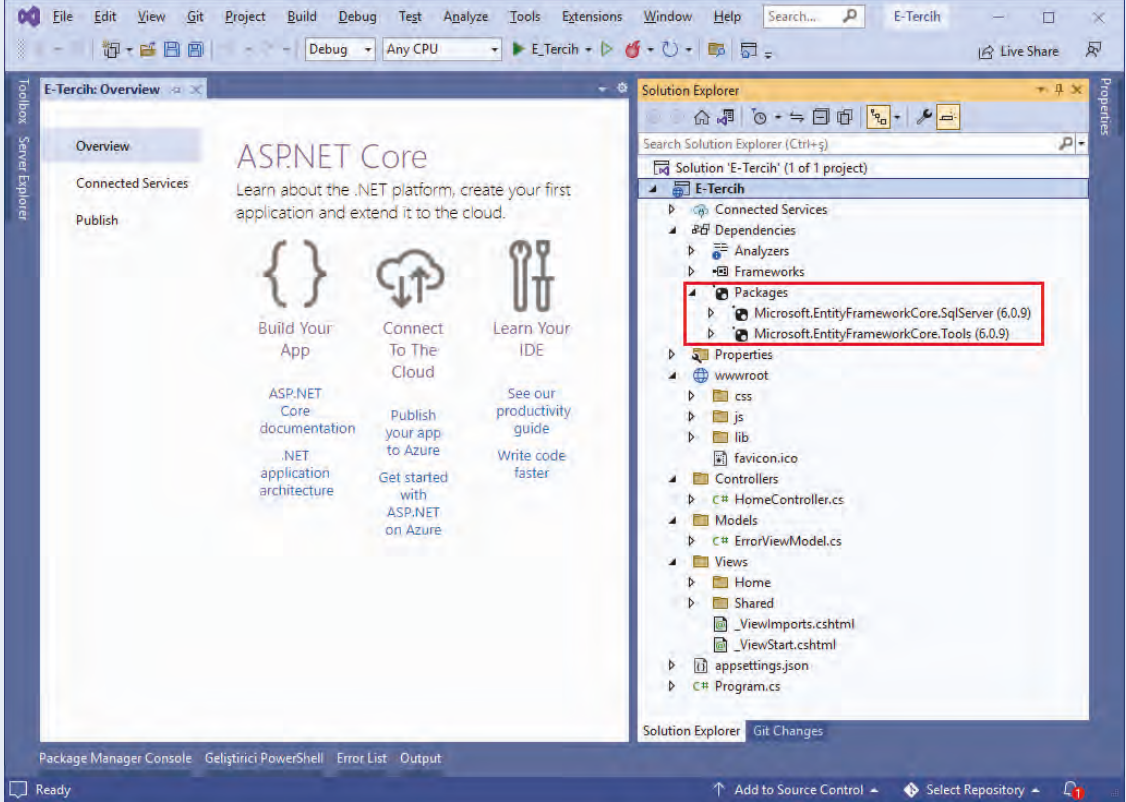


Görsel 4.17: Paketin yüklenmesi sonrasında projede yapılacak değişiklikler penceresi



3. Adım: Görsel 4.17'deki bilgilendirme penceresinde, projede yapılacak değişiklikleri inceledikten sonra "OK" düğmesine tıklayınız. Paketin yüklenmesiyle ilgili lisans sözleşmesi penceresi gösterilecektir. Bu lisans sözleşmesini kabul etmek de paketi yüklemenin bir ön koşuludur. I Accept (Kabul ediyorum.) düğmesine tıklayarak yüklemeye devam ediniz.

4. Adım: İkinci paketi yüklemek için bu uygulamanın 1, 2 ve 3. adımlarını tekrarlayınız. İkinci paketi yüklemek için Görsel 4.16'daki arama kutusuna "Microsoft.EntityFrameworkCore.Tools" yazınız. Her iki paketin de yüklenmesi tamamlandıktan sonra Görsel 4.18'de görülen proje dosya yapısına benzer bir yapı oluşacaktır.



Görsel 4.18: Proje dosya yapısının paketler yüklendikten sonraki görünümü



5. UYGULAMA

İşlem adımlarına göre veri tabanına bağlantı oluşturarak bağlantı dizesini elde ediniz.

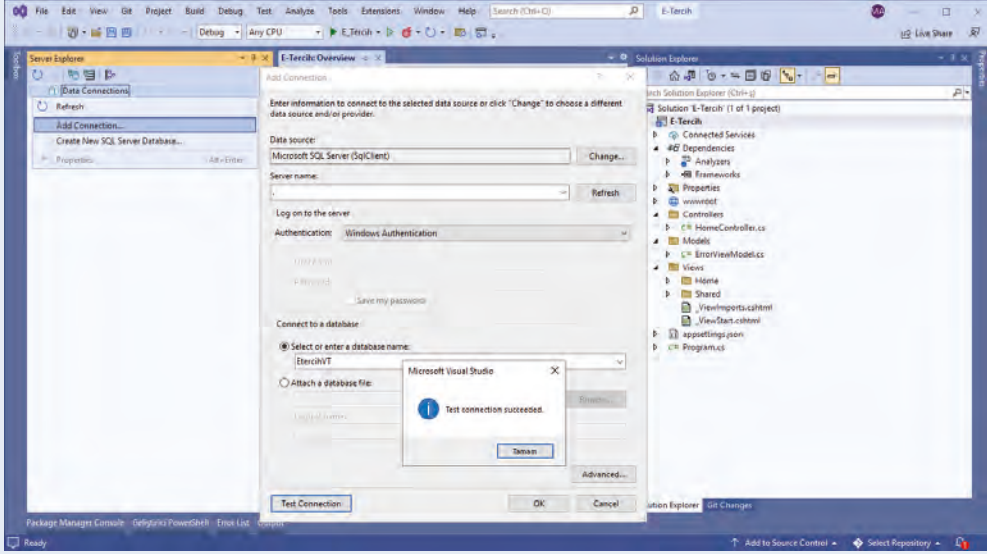


NOT

Scaffolding yöntemiyle model yapısının oluşturulabilmesi için Connection String (Bağlantı Dizesi) metnine ihtiyaç duyulur. Bu metni elde etmek için Server Explorer (Sunucu Gezgini) penceresi yardımıyla veri tabanına bağlantı oluşturmak gerekir.



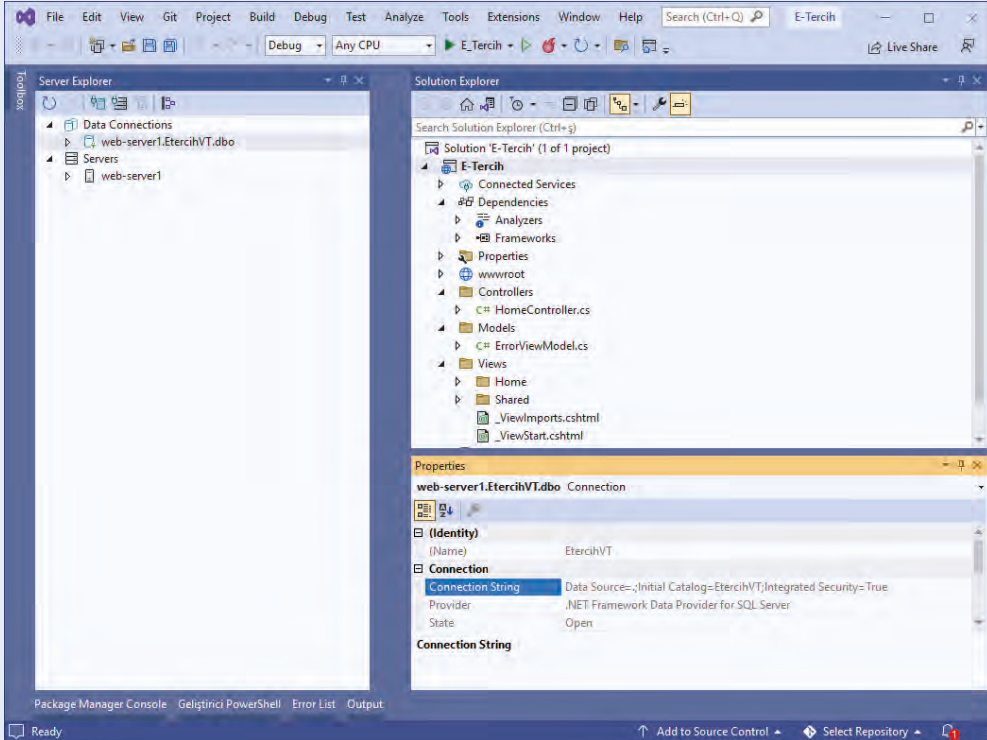
1. Adım: Visual Studio'nun View (Görünüm) ana menüsündeki Server Explorer seçeneğini belirleyiniz. Görsel 4.19'daki pencere açılacaktır.



Görsel 4.19: Veri tabanı sunucusuna bağlantı oluşturma penceresi

2. Adım: Görsel 4.19'da görülen Server Explorer penceresindeki Data Connections (Veri Bağlantıları) seçeneğini sağ tıklayarak açılan menüden Add Connection... (Bağlantı Ekle...) seçeneğine tıklayınız.

3. Adım: Görsel 4.19'da gösterilen Add Connection penceresinde ayarları yaptıktan sonra Test Connection (Bağlantıyı Test Et) düğmesine tıklayınız. "Test connection succeeded." bilgi mesajını aldıysanız bağlantı başarılı bir şekilde yapılmıştır. Tamam düğmesine, ardından OK düğmesine tıklayarak Add Connection penceresini kapatınız. Server Explorer penceresine Görsel 4.20'deki gibi yeni bir bağlantı eklendiğini göreceksiniz.



Görsel 4.20: Veri tabanı sunucusuna bağlantı oluşturma penceresi



4. Adım: Görsel 4.20’de gösterildiği gibi Server Explorer penceresindeki bağlantıyı (web-server1. EtercihVT.dbo) seçtikten sonra klavyeden F4 tuşuna basarak Properties (Özellikler) penceresini açınız.

5. Adım: Görsel 4.20’deki pencerede görüldüğü gibi Connection String özelliğinin değerini seçip kopyalayıp hafızaya alınız.



6. UYGULAMA

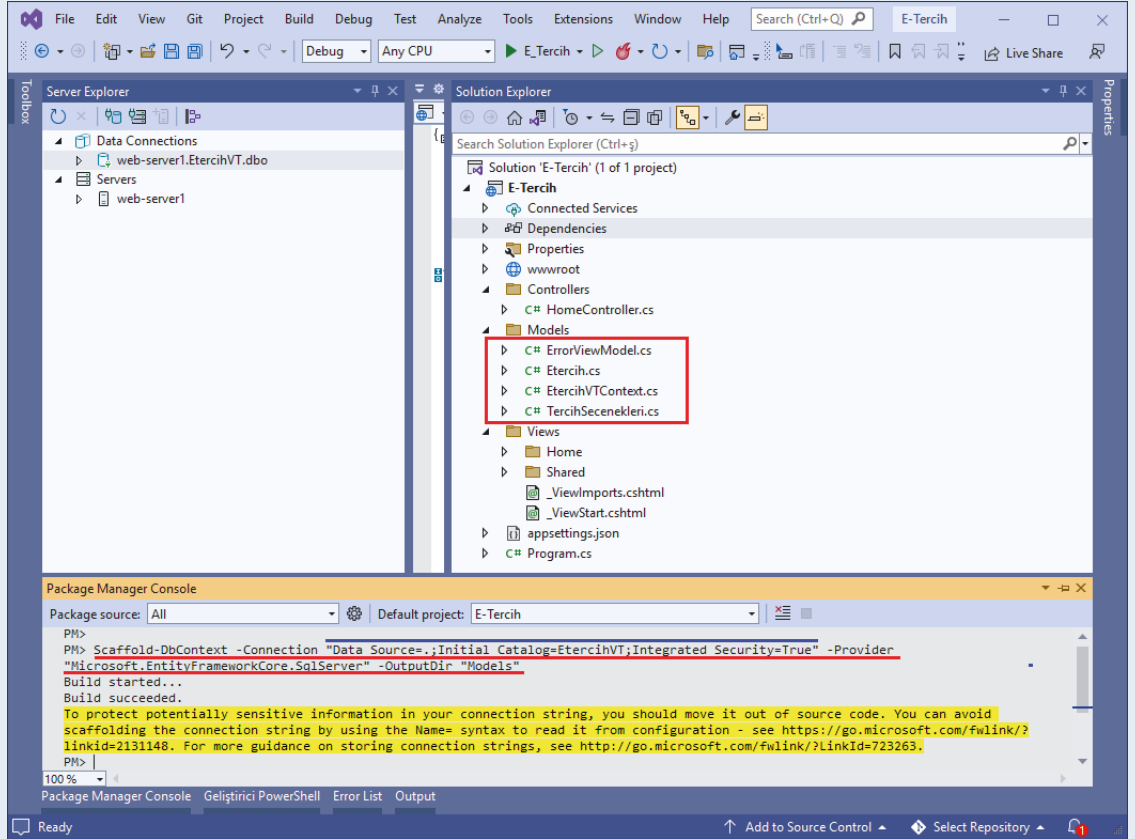
İşlem adımlarına göre Scaffolding komutunu yazarak model sınıflarını oluşturunuz.



NOT

Dördüncü uygulamada kopyalanarak hafızaya alınan Connection String değeri kullanılarak, Scaffolding komutu yazılıp model sınıfları oluşturulacaktır.

1. Adım: Visual Studio ana menüsündeki Tools (Araçlar) menüsü altında bulunan NuGet Package Manager (NuGet Paket Yöneticisi) altındaki Package Manager Console (Paket Yöneticisi Konsolu) seçeneğini tıklayınız. Görsel 4.21’de görülen pencere açılacaktır.



Görsel 4.21: Package Manager Console penceresi



2. Adım: Görsel 4.21’de altı kırmızı çizili olan komutu “PM>” komut satırına yazınız. Dördüncü uygulamada belleğe kopyaladığınız bağlantı dizesi metnini komut içindeki üstü mavi çizgili olan metnin bulunduğu yere yapıştırınız.

3. Adım: İkinci adımda yazdığınız komutu Enter tuşuna basarak çalıştırınız. “Build succeeded.” mesajını aldıysanız komut başarılı bir şekilde işini yapmıştır. Görsel 4.21’de kırmızı çerçeve ile gösterilen dosyaların oluştuğunu göreceksiniz.



7. UYGULAMA

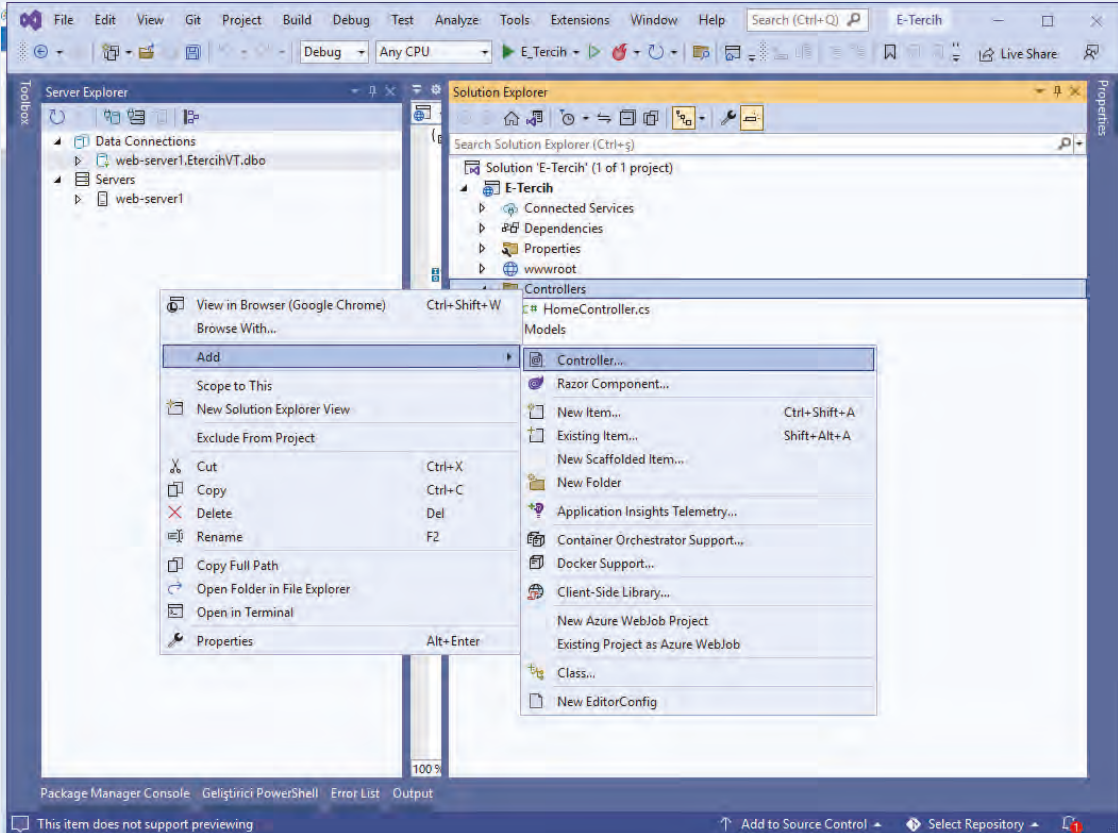
İşlem adımlarına göre Controllers ve Views klasörlerini ve bu klasörlerle ilgili dosyaları oluşturunuz.



NOT

Models klasöründeki dosyaların oluşturulmasından sonra Controllers ve Views klasörlerin ve bu klasörlerle ilgili dosyalarının da oluşturulması gerekir.

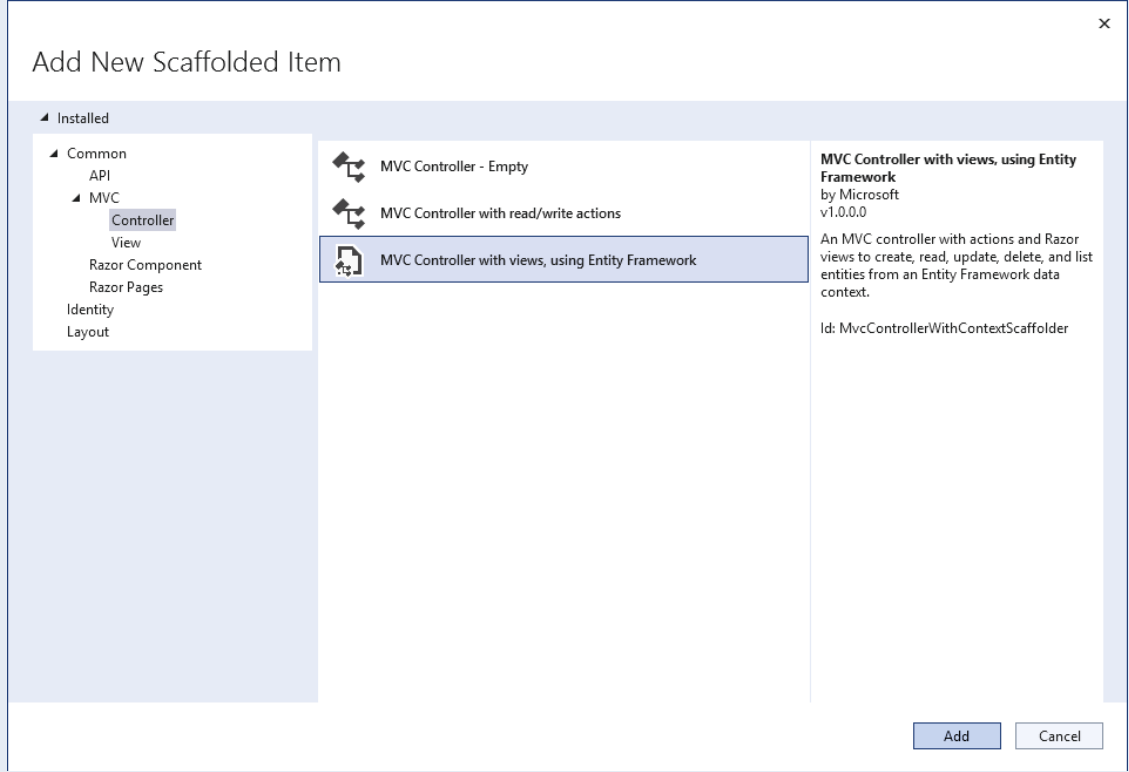
1. Adım: Görsel 4.22’de görüldüğü gibi Solution Explorer (Çözüm Gezgini) penceresinde görülen Controllers klasörüne sağ tıklayarak açılan menüden Add... (Ekle), ardından da Controllers (Kontrolcüler) seçeneğini tıklayınız.



Görsel 4.22: Solution Explorer penceresinde Controller (Kontrolcü) ekleme

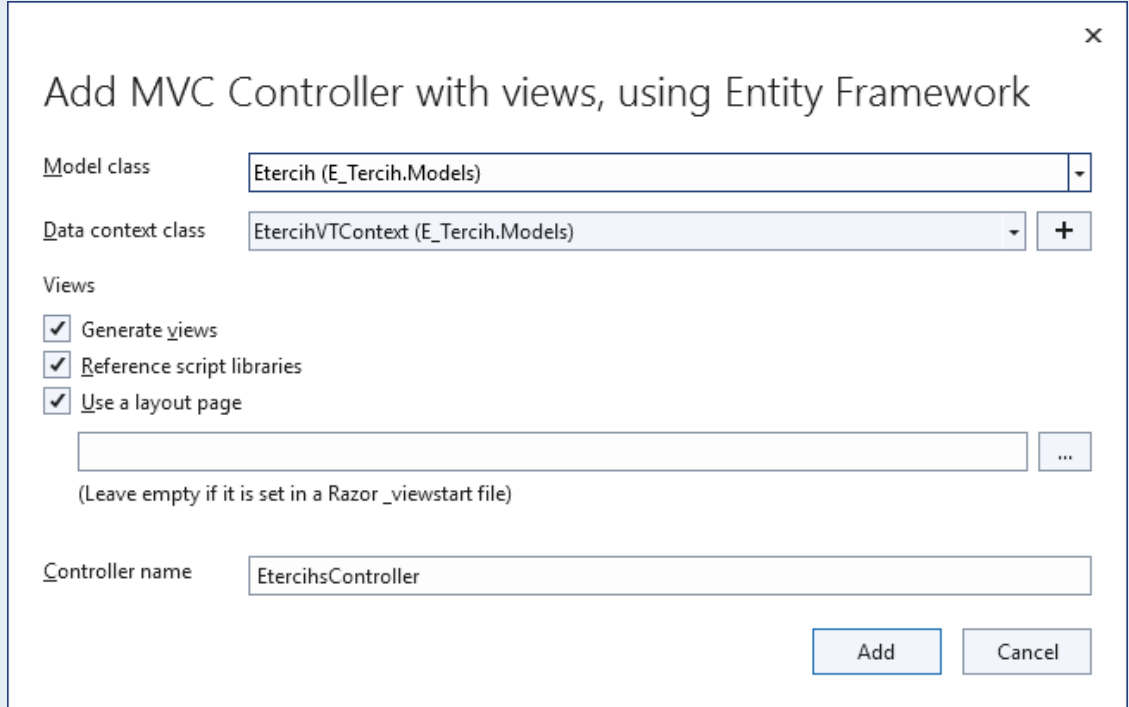


Görsel 4.23'te görülen pencere açılacaktır.



Görsel 4.23: Yeni bir Controller ekleme penceresi

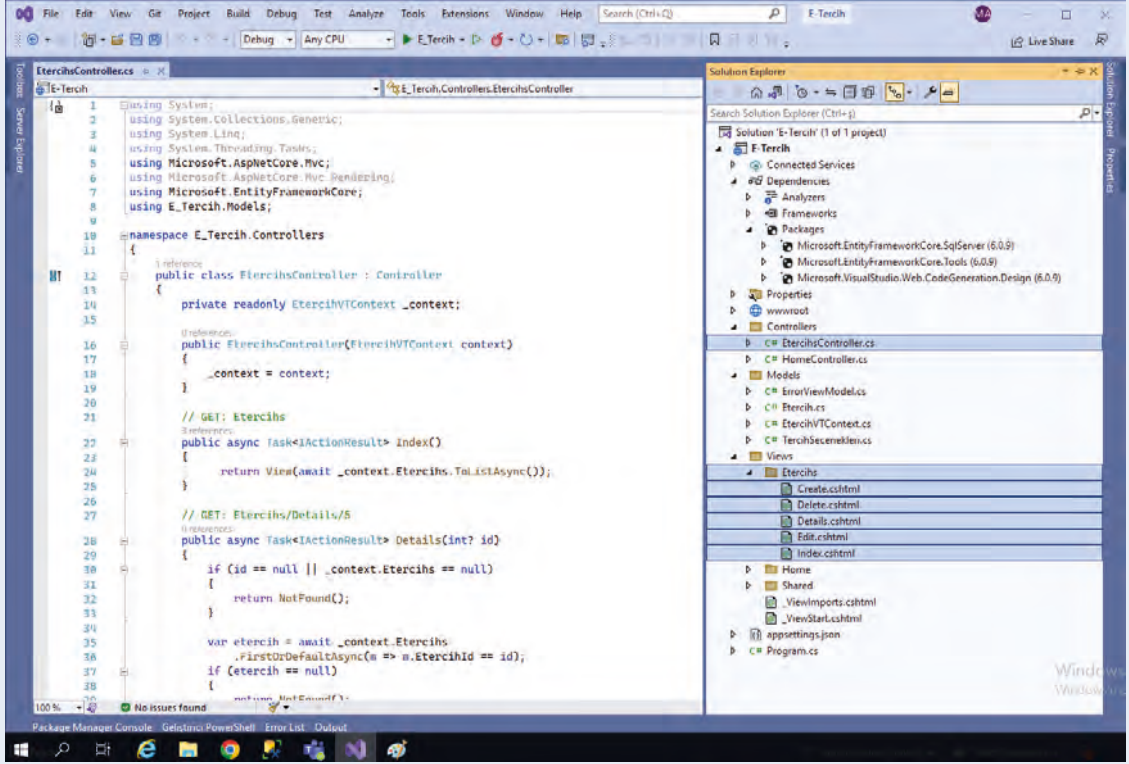
2. Adım: Görsel 4.23'te görüldüğü gibi “MVC Controller with views, using Entity Framework” seçeneğini belirleyerek Add düğmesine tıklayınız. Görsel 4.24'te görülen pencere açılacaktır.



Görsel 4.24: Controllers ile birlikte Views ekleme penceresi



3. Adım: Görsel 4.24'teki pencerede görülen ayarları belirledikten sonra Add düğmesine tıklayınız. Bu işlem Görsel 4.25'te görülen dosyaların projeye eklenmesini sağlayacaktır.



Görsel 4.25: Controllers ve Views dosyalarının eklenmesi

4. Adım: Görsel 4.25'te Solution Explorer penceresinde yer alan Etercihs klasörü içindeki dosyaları inceleyiniz.

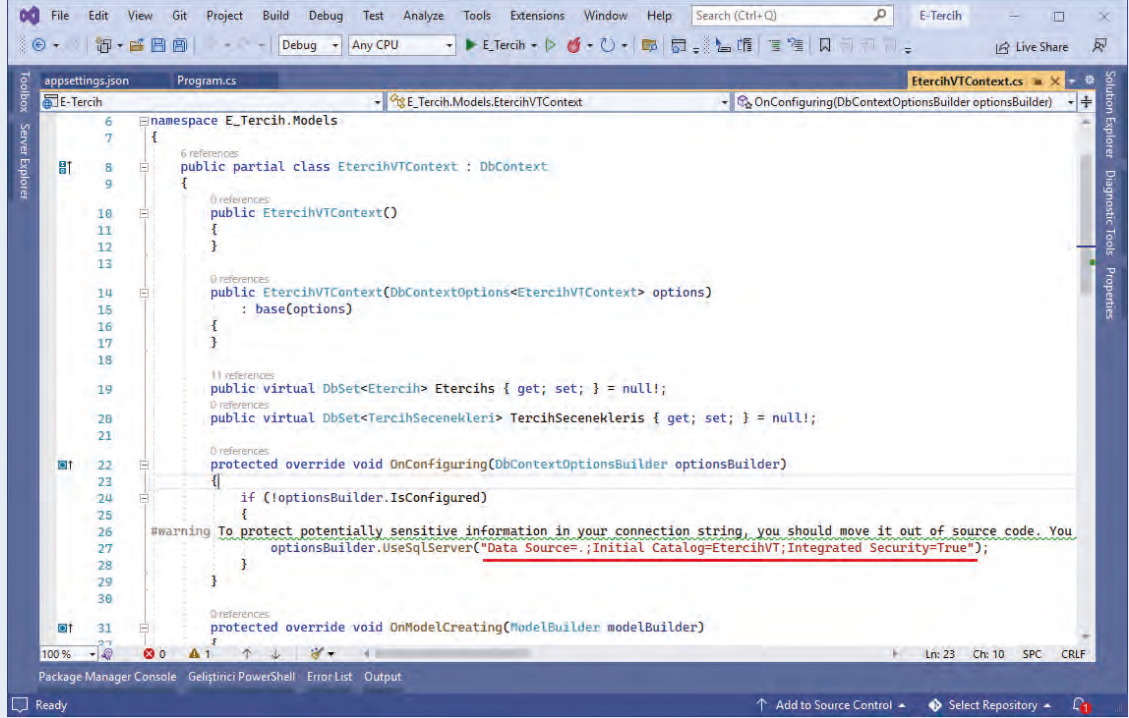
Bu klasör içinde Create.cshtml, Delete.cshtml, Details.cshtml, Edit.cshtml, Index.cshtml dosyaları yer alır. Bu dosyalar sırasıyla kayıt oluşturma, kayıt silme, kayıt ayrıntılarını görme, kayıt güncelleme, kayıtların listesini görme amaçlı kullanılan dosyalardır.



8. UYGULAMA

İşlem adımlarına göre appsettings.json ve Program.cs dosyası içine veri tabanı bağlantısının yapılabilmesi için gerekli kodları ekleyiniz.

1. Adım: Solution Explorer penceresinde yer alan Models klasöründeki EtercihVtContext.cs dosyasını Görsel 4.26'da görüldüğü gibi açınız.

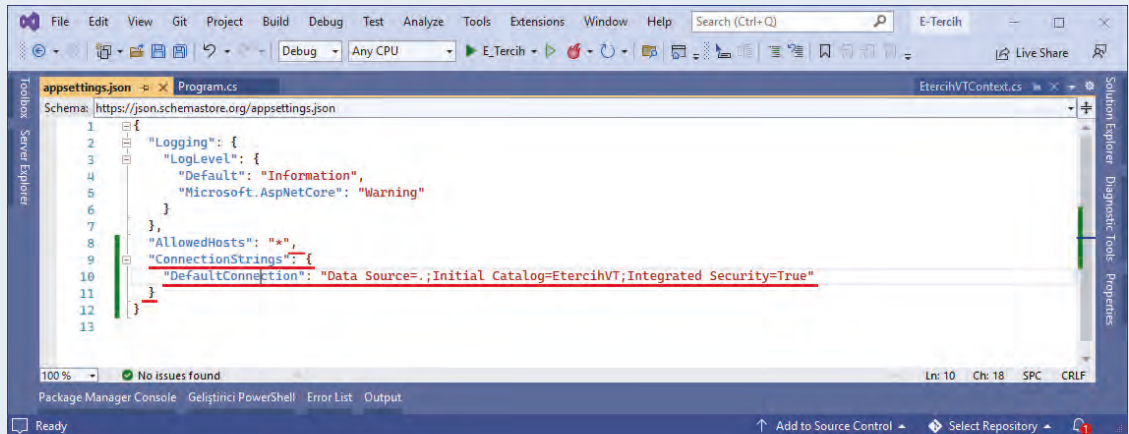


Görsel 4.26: EtercihVtContext.cs dosya içeriğini gösteren pencere

2. Adım: Görsel 4.26'daki pencerede, 27. satırda altı kırmızı çizili olan metni bulunuz ve hafızaya kopyalayınız.

3. Adım: Solution Explorer penceresinde appsettings.json dosyasını bularak açınız.

4. Adım: Görsel 4.27'de görüldüğü gibi appsettings.json dosyasının içeriğini, altı kırmızı çizili olan satırları ekleyerek düzenleyiniz ve dosyayı kaydediniz.



Görsel 4.27: appsettings.json dosyasının içeriğini gösteren pencere



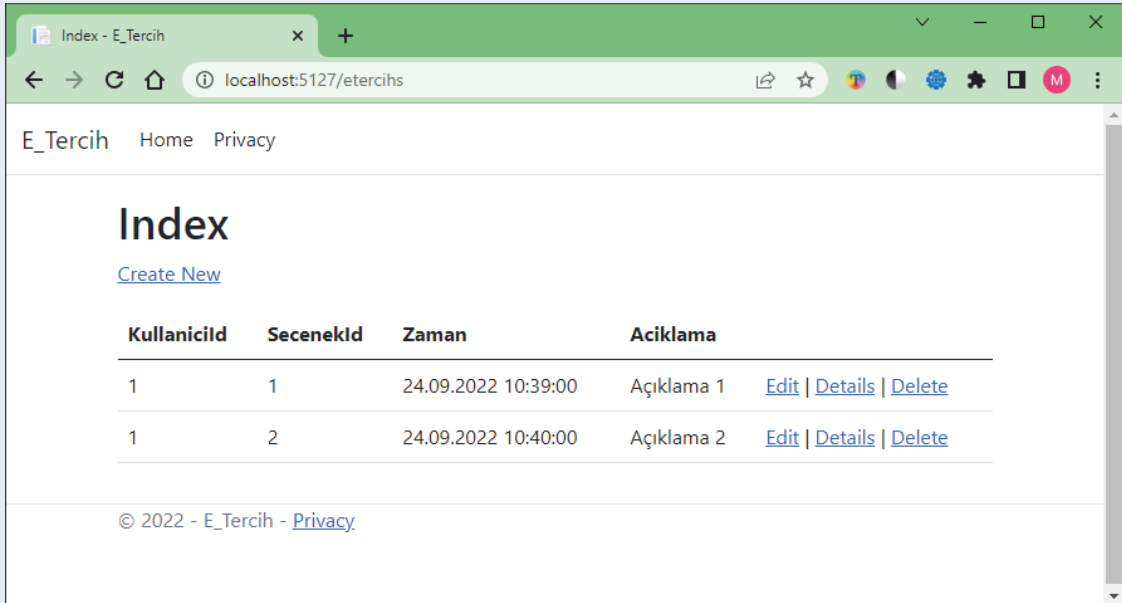
5. Adım: Solution Explorer penceresinden Program.cs dosyasını bularak açınız. Görsel 4.28’de görülen pencere açılacaktır.

```
1 using E_Tercih.Models;
2 using Microsoft.EntityFrameworkCore;
3
4 var builder = WebApplication.CreateBuilder(args);
5 string connection = builder.Configuration.GetConnectionString("DefaultConnection");
6
7 // Add services to the container.
8 builder.Services.AddControllersWithViews();
9 builder.Services.AddDbContext<EtercihVTCContext>(options =>
10 {
11     options.UseSqlServer(connection);
12 });
13 var app = builder.Build();
14
15 // Configure the HTTP request pipeline.
16 if (!app.Environment.IsDevelopment())
17 {
18     app.UseExceptionHandler("/Home/Error");
19 }
20 app.UseStaticFiles();
21 app.UseRouting();
22 app.UseAuthorization();
23
24 app.MapControllerRoute(
25     name: "default",
26     pattern: "{controller=Home}/{action=Index}/{id?}");
27
28 app.Run();
29
30
31
```

Görsel 4.28: Program.cs dosyasının düzenlenmiş durumunu gösteren pencere

6. Adım: Görsel 4.28’de görülen penceredeki gibi altı kırmızı çizili olan kod satırlarını Program.cs dosyasına ekleyerek dosyayı kaydediniz.

7. Adım: Klavyeden F5 tuşuna basarak uygulamayı test ediniz. Görsel 4.29’dakine benzer bir pencere ile karşılaşırsanız uygulamanız hatasız çalışır. Görsel 4.29’daki gibi adres satırında “/etercihs” ifadesini göremezseniz bu ifadeyi eklemeyi unutmayınız. Bu ifadeyi ekledikten sonra uygulamayı tekrar test etmeyi deneyiniz.



Görsel 4.29: E-tercih web tabanlı uygulamasının web tarayıcı penceresindeki görünümü



SIRA SİZDE

TercihSecenekleri.cs model dosyası için Controllers ve Views dosyalarını oluşturunuz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Controllers sağ menüsünden Add seçeneğinin altındaki Controller... komutuna tıkladı.		
2. "Add new scaffolded item" penceresinde "MVC Controllers with views, using entity framework" adlı üçüncü seçeneği seçti.		
3. Model Class ile Data Context Class'ı doğru bir şekilde belirledi.		
4. Controllers ve Views dosyalarının eklendiğini gösterdi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



9. UYGULAMA

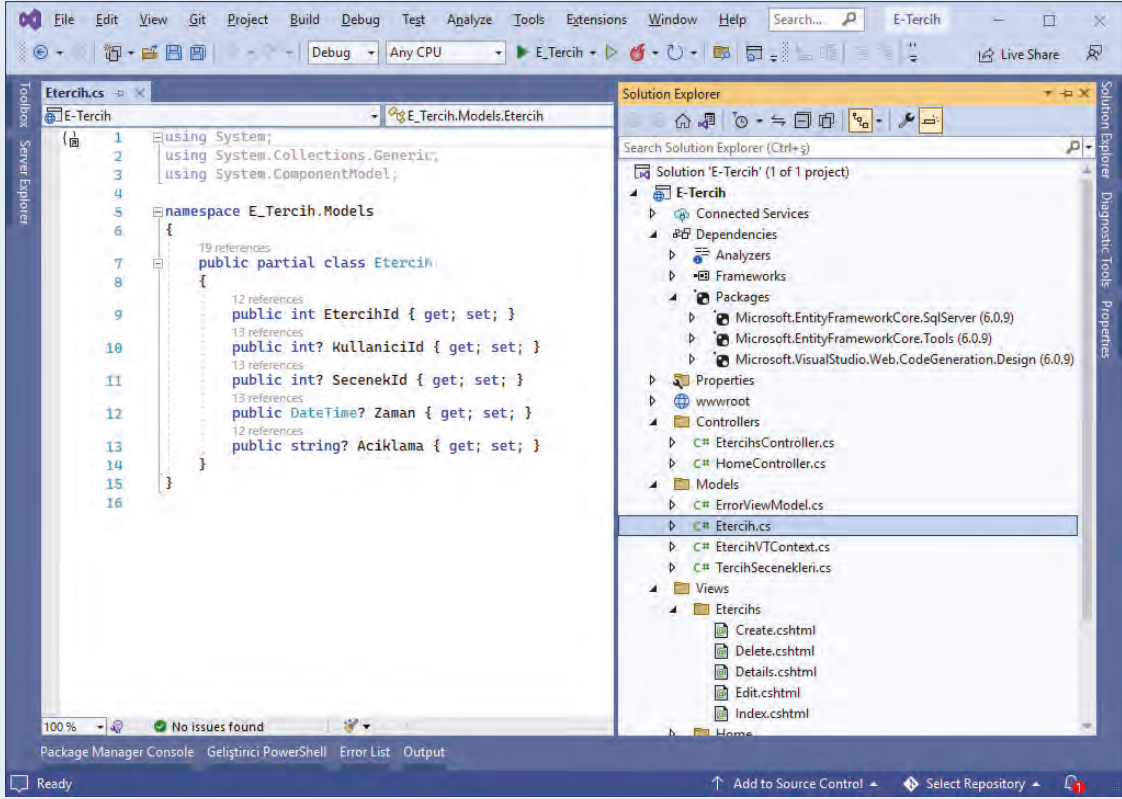
İşlem adımlarına göre Görsel 4.29'da görülen sayfadaki sütun başlıklarını uygun ifadelerle değiştiriniz.



NOT

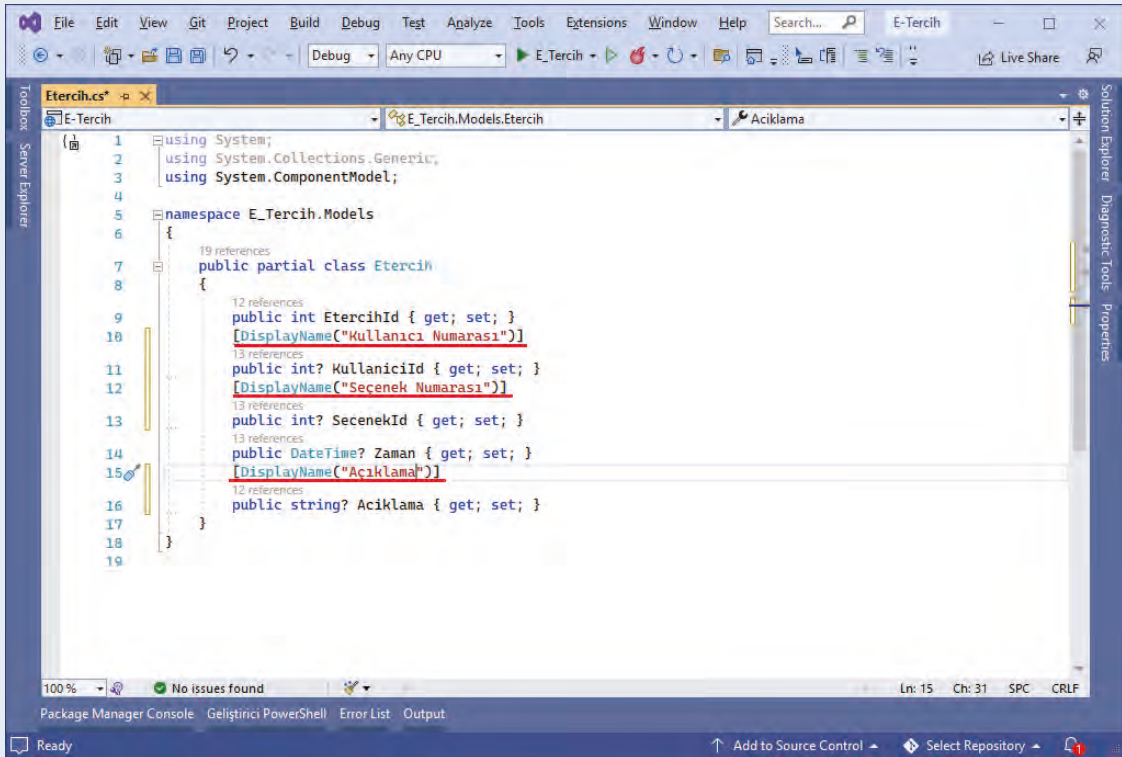
Görsel 4.29'da görülen sayfadaki sütun başlıkları düzeltilmelidir. "KullanıcıID", "Kullanıcı Numarası" şeklinde; "SecenekID", "Seçenek Numarası" şeklinde; "Acıklama" da "Açıklama" şeklinde değiştirilecektir. Son kullanıcının teknik nedenlerden dolayı Türkçe karakter kullanılmadan yazılan sütun başlıkları yerine Türkçe karakterler içeren ifadeleri görmesi gerekir. Bu değişiklikler Model üzerinde yapılacaktır.

1. Adım: Görsel 4.30'da görüldüğü gibi Etercih.cs dosyasını açınız.



Görsel 4.30: Etercih.cs dosyasının içeriği

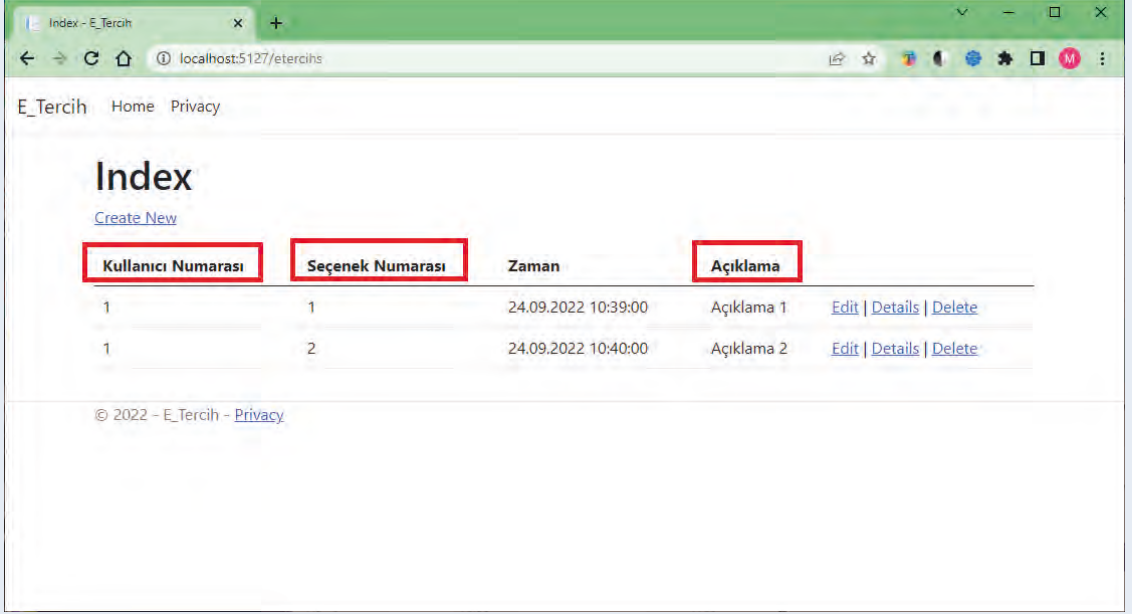
2. Adım: Görsel 4.31’de görüldüğü gibi Etercih.cs dosyası içine altı kırmızı çizili satırları ekleyiniz.



Görsel 4.31: Etercih.cs dosyasına sütun başlığı değiştiren kodların eklenmesi



3. Adım: Görsel 4.32’de görüldüğü gibi F5 tuşuna basarak sayfayı test ediniz. “Acıklama” yerine “Açıklama” ifadesinin yer aldığını ve diğer ifadelerin de düzeltildiğini göreceksiniz.



Görsel 4.32: Sütun başlıkları değiştirilmiş sayfanın web tarayıcısında görünümü



SIRA SİZDE

TercihSecenekleri.cs dosyasını açarak son kullanıcı ekranında görülmesi uygun olmayan ifadeleri değiştiriniz.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. TercihSecenekleri.cs dosyasını VS editöründe açtı.		
2. TercihSecenekleri.cs dosyası içindeki uygun satırlara “DisplayName” ifadesini ekledi.		
3. TercihSecenekleri.cs dosyasını web tarayıcısında açtı.		
4. Sütun başlıklarının değiştirildiğini gösterdi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



10. UYGULAMA

İşlem adımlarına göre Views/Etercihs/index.cshtml dosyası içindeki “Create New”, “Edit”, “Details” ve “Delete” ifadelerini sırasıyla “Yeni Oluştur”, “Değiştir”, “Ayrıntılar”, “Sil” şeklinde değiştiriniz.



NOT

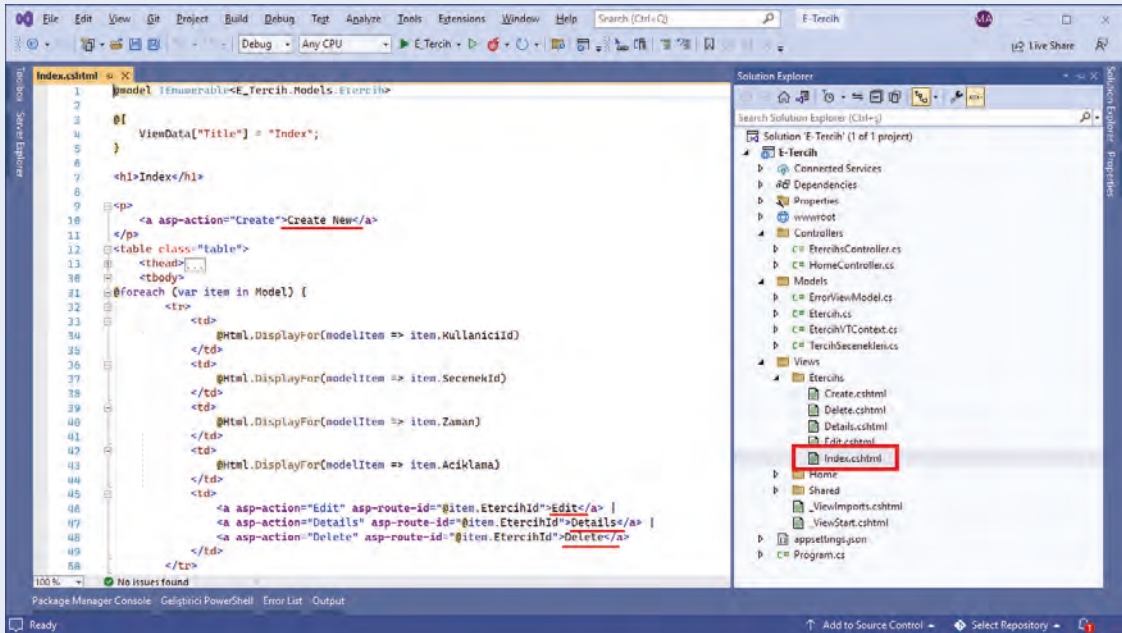
Sekizinci uygulamada Model üzerinde yapılan değişikliklerle sütun başlıkları daha anlaşılır ve Türkçe karakterler içeren ifadelerle dönüştürülmüştü. Sayfanın geriye kalanında yer alan “Create New”, “Edit | Details | Delete” gibi ifadeler değiştirilmelidir. Bu değişiklikler, View üzerinde yapılacaktır.

1. Adım: Görsel 4.33’te görüldüğü gibi Solution Explorer içindeki Views/Etercihs klasöründe yer alan “Index.cshtml” dosyasını açınız.



NOT

CSHTML dosyaları C Sharp (C#) komutları, ASP.NET etiketleri ve HTML etiketlerinin birlikte kullanıldığı dosyalardır.



Görsel 4.33: Index.cshtml dosyasının Visual Studio editöründe açılması



2. Adım: Görsel 4.33'te görülen Index.cshtml dosyasındaki altı çizili ifadeleri değiştiriniz. "Create New" ifadesini "Yeni Oluştur", "Edit" ifadesini "Değiştir", "Details" ifadesini "Ayrıntılar" ve "Delete" ifadesini de "Sil" şeklinde değiştiriniz. Sayfanızın yeni hâli Görsel 4.34'teki gibi görünmelidir.

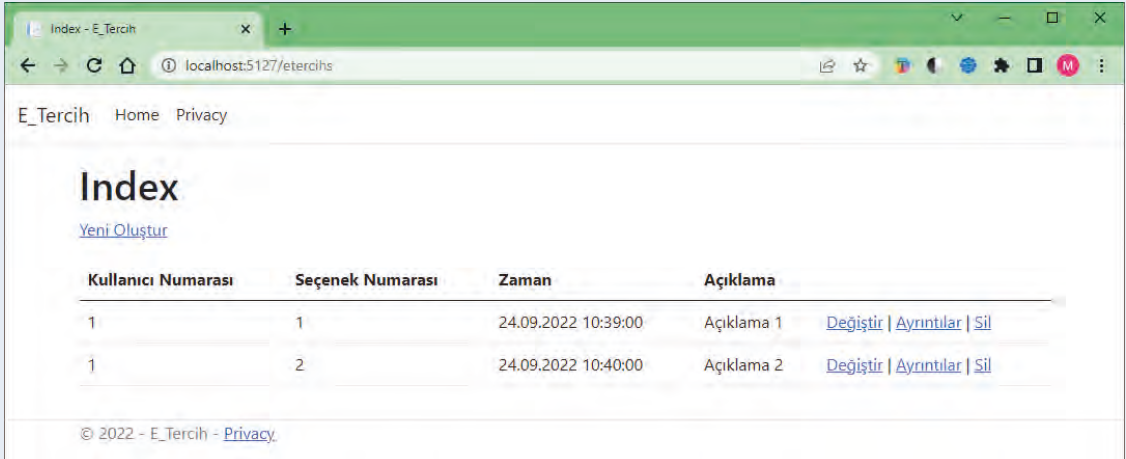
```

1 @model IEnumerable<E_Tercih.Models.Etercih>
2
3 @{
4     ViewData["Title"] = "Index";
5 }
6 <h1>Index</h1>
7
8 <p>
9     <a asp-action="Create">Yeni Oluştur</a>
10 </p>
11 <table class="table">
12     <thead>
13         <tr>
14             <th></th>
15         </tr>
16     </thead>
17     <tbody>
18         @foreach (var item in Model) {
19             <tr>
20                 <td>
21                     @Html.DisplayFor(modelItem => item.KullaniciId)
22                 </td>
23                 <td>
24                     @Html.DisplayFor(modelItem => item.SecenekId)
25                 </td>
26                 <td>
27                     @Html.DisplayFor(modelItem => item.Zaman)
28                 </td>
29                 <td>
30                     @Html.DisplayFor(modelItem => item.Aciklama)
31                 </td>
32                 <td>
33                     <a asp-action="Edit" asp-route-id="@item.EtercihId">Değiştir</a> |
34                     <a asp-action="Details" asp-route-id="@item.EtercihId">Ayrıntılar</a> |
35                     <a asp-action="Delete" asp-route-id="@item.EtercihId">Sil</a>
36                 </td>
37             </tr>
38         }
39     </tbody>
40 </table>

```

Görsel 4.34: Index.cshtml dosyasının değiştirilmiş hâli

3. Adım: Visual Studio editöründe iken F5 tuşuna basarak sayfayı Görsel 4.35'teki gibi web tarayıcısında görüntüleyiniz. Sayfanın tamamen Türkçe ifadelerle dönüştürüldüğü görülecektir.



Görsel 4.35: Index sayfasında ifadelerin düzenlenmiş hâli

4.4. WEB TABANLI UYGULAMA PROJESİNİN TEST VE BAKIMININ YAPILMASI

Bir yazılım projesinin test ve bakım gibi kritik aşamaları vardır. Test, projenin başlangıçta belirlenen görevleri ne ölçüde yerine getirdiğinin ve ihtiyaçlara ne kadar cevap verdiğinin denetlenmesidir. Ayrıca test aşamasında yazılım projesinin doğru çalışıp çalışmadığı ve gereken performansı gösterip göstermediği gibi ölçütlere göre de denetlenmesi gerekir. Bir yazılım projesi; alfa testi, beta testi, kabul testi gibi farklı aşamalarda gerçekleştirilen testlerden geçer.



Tüm testlerden başarıyla geçen web tabanlı yazılım projesi yayımlama aşamasına ulaşmıştır. Web tabanlı yazılım projeleri, ulaşılacak kullanıcı kitlesine göre farklı ortamlarda yayımlanabilir. Projenin yalnızca kurum içinden erişilip kullanılması gerekiyorsa intranet üzerinden yayımlanması yeterli olacaktır ancak web tabanlı yazılım projesine her yerden erişilecekse dünya çapında yayın yapan bir sunucunun kullanılması gerekir. Bu amaçla kurulmuş çok sayıda ticari şirket mevcuttur ve bunların uygun olanından hizmet satın alınabilir. Dünya çapında yayın yapılacaksa yazılım projesi için benzersiz bir ismin de kaydedtirilmesi gerekir. İsmi kaydedtirilmesi amacıyla hizmet veren çok sayıda kuruluş mevcuttur. Bu şirketlerin uygun olanından hizmet satın alınabilir. Web tabanlı yazılım projesini yayımlamak ve web tabanlı yazılım projesinin adını kaydetmek için hizmet alınan şirketler aynı veya farklı olabilir.

Tüm testleri geçip uygulamaya konulduktan sonra da geliştirilen yazılım projesinin düzenli aralıklarla bakımının yapılması esastır. Bakım yapılmasının amacı, yazılım projesinin başlangıçta sağladığı performansı, güvenilirliği ve güvenliğini sürdürmesidir. Bu nedenle proje sürekli izlenir. Projenin zamanla değişen ihtiyaçlara göre yeniden güncellenmesi de gerekebilir. Güncelleme aşamalarına farklı isimler veya numaralar verilir ve bu işlem, sürüm (versiyon) olarak adlandırılır.

4.5. WEB TABANLI UYGULAMA PROJESİNİN RAPORUNUN OLUŞTURULMASI

Gerçekleştirilen yazılım projesinin kullanımı ve özelliklerine dönük bir raporun hazırlanarak müşteriye sunulması esastır. Bu rapor, yazılım projesinin başlangıçta belirlenen ihtiyaçları ne ölçüde karşıladığından kullanım senaryolarına varıncaya kadar birçok görsel ve grafik içermelidir. Özellikle kullanıcının başlangıçta talep ettiği fonksiyonları gerçekleştirirken hangi yol ve yöntemleri izlemesi gerektiğini içermelidir. Bu rapor aynı zamanda son kullanıcının yazılım projesini kullanırken zaman zaman başvuracağı baş ucu kılavuzu niteliğini taşımaktadır.

Müşteri yüz yüze bir sunum talep ederse proje raporunun sunum şeklinde hazırlanan bir versiyonu müşteriye sunulmalıdır. Bu sunum sırasında müşterinin aklına takılan sorulara cevap bulması ve yazılım projesinin kullanımı konusunda bir fikir edinmesi önem taşır. Sunum sırasında yazılım projesinin bizzat canlı uygulamasının müşteriye sunulması da etkili olacaktır.



SIRA SİZDE

Geliştirdiğiniz web tabanlı e-kütüphane uygulamasına ait proje raporunu hazırlayınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje raporunda müşteri ihtiyaçları ile yazılım projesi özelliklerini karşılaştıran bir tablo oluşturdu.		
2. Proje raporunda yazılım projesine ait ekran görsellerini içeren kullanım senaryolarını oluşturdu.		
3. Proje raporuna müşterinin daha sonra başvurmak üzere kullanabileceği referans bilgiler ekledi.		
4. Proje raporuna içindekiler tablosu ekledi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



SIRA SİZDE

Geliştirdiğiniz web tabanlı e-sınav uygulamasına ait proje raporunu hazırlayınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje raporunda müşteri ihtiyaçları ile yazılım projesi özelliklerini karşılaştıran bir tablo oluşturdu.		
2. Proje raporunda yazılım projesine ait ekran görsellerini içeren kullanım senaryolarını oluşturdu.		
3. Proje raporuna müşterinin daha sonra başvurmak üzere kullanabileceği referans bilgiler ekledi.		
4. Proje raporuna içindekiler tablosu ekledi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



SIRA SİZDE

Geliştirdiğiniz web tabanlı e-kütüphane uygulamasına ait proje raporunun sunumunu yapınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje raporunun bir özetinden yararlanarak proje sunum belgesini oluşturdu.		
2. Proje sunumu sırasında sunuma katılanların yazılım projesiyle ilgili aklına takılan tüm sorularına cevap verdi.		
3. Proje sunumunda yazılım projesinin canlı uygulamasını gösterdi.		
4. Sunum sırasında jest ve mimikleri de kullanarak etkili bir sunum gerçekleştirdi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



SIRA SİZDE

Geliştirdiğiniz web tabanlı e-sınav uygulamasına ait proje raporunun sunumunu yapınız.

DEĞERLENDİRME

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
1. Proje raporunun bir özetinden yararlanarak proje sunum belgesini oluşturdu.		
2. Proje sunumu sırasında sunuma katılanların yazılım projesiyle ilgili aklına takılan tüm sorularına cevap verdi.		
3. Proje sunumunda yazılım projesinin canlı uygulamasını gösterdi.		
4. Sunum sırasında jest ve mimikleri de kullanarak etkili bir sunum gerçekleştirdi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat etti.		
6. Zamanı verimli kullandı.		



ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruların doğru seçeneğini işaretleyiniz.

1. Aşağıdakilerden hangisi gereksinim analiz aşamalarından biri değildir?

- A) İhtiyaçların toplanması
- B) İhtiyaçların uygulanabilirliğinin denetlenmesi
- C) Uygulamanın test edilmesi
- D) İhtiyaçların uygunluğunun analiz edilmesi
- E) İhtiyaçların belgelenmesi

2. Aşağıdakilerden hangisi risklere karşı alınan önlemlerden biri değildir?

- A) Aktarma
- B) Azaltma
- C) Giderme
- D) Kabullenme
- E) Kaçınma

3. Aşağıdakilerden hangisi risk yönetimi aşamalarından biri değildir?

- A) Riskin tanımlanması
- B) Riskin önlenmesi
- C) Risk analizinin yapılması
- D) Riskin planlanması
- E) Riskin izlenmesi

4. Varlık ilişki diyagramında aşağıdaki şekillerden hangisi yoktur?

- A) Elips
- B) Dikdörtgen
- C) Düz çizgi
- D) Paralelkenar
- E) Yamuk

5. Model View Controller mimarisinde View aşağıdakilerden hangisine karşılık gelir?

- A) Veri tabanı tablosu
- B) Veri tabanı sorgusu
- C) Yapılandırma dosyası
- D) Web sayfası
- E) Programlama dili sınıfı



CEVAP ANAHTARI

1. ÖĞRENME BİRİMİ

1)	A
2)	C
3)	B
4)	D
5)	E
6)	B
7)	B

2. ÖĞRENME BİRİMİ

1)	Tablo
2)	Primary Key
3)	Click
4)	DataGridView
5)	MessageBox
6)	D
7)	A
8)	D
9)	B
10)	A

3. ÖĞRENME BİRİMİ

1)	Y
2)	Y
3)	D
4)	Y
5)	Y
6)	D
7)	D
8)	D
9)	D
10)	Y

4. ÖĞRENME BİRİMİ

1)	C
2)	C
3)	B
4)	E
5)	D



KAYNAKÇA

- http://meslek.eba.gov.tr/Nesne_Tabanli_Programlama_10_3.pdf Erişim Tarihi: 07.09.2022 Saat: 10.00
- https://www.tubitak.gov.tr/sites/default/files/content_files/iletisim/sunum_el_kibabi.pdf Erişim Tarihi: 07.09.2022 Saat: 10.00
- MEB.(2020). Bilişim Teknolojileri Alanı Çerçeve Öğretim Programı, Ankara
- Türk Dil Kurumu, sözlükleri. (t.y.). <https://sozluk.gov.tr/>
- Türk Dil Kurumu (t.y.). <https://www.tdk.gov.tr/>

* Kaynakça, APA 6 referanslama sistemi kullanılarak oluşturulmuştur.

GÖRSEL KAYNAKÇASI



<http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=2427>