

FLUTTER

Bildirim ve Paylaşım

31. hafta • Cihaz Özellikleri ve Kalite

36 hafta, 72 bağımsız çalışan Flutter projesi, gerçek uygulama ekranları, uygulamalı dersler ve öğrenci geri bildiriminden sonra incelenebilen çözüm kaynakları.

Bu kitap güncel ders kaynaklarından otomatik üretilmiştir; özet ya da jenerik şablon değildir.

Öğrenme Haritası

Hafta	Ders	Projeler
31	Bildirim ve Paylaşım	Hatırlatma • Acil Durum Bilgi Kartı

31. Hafta — Bildirim ve Paylaşım

Bu hafta ne yapacağız?

Flutter uygulamasını işletim sisteminin iki özelliğine bağlayacağız. İlk uygulama, kullanıcı iznini yöneten gerçek bir yerel bildirim servisiyle çalışma hatırlatıcısı oluşturacak. İkinci uygulama hassas bilgileri yalnız kullanıcının açık komutuyla sistem paylaşım menüsüne gönderecek.

Kazanımlar

- flutter_local_notifications paketini başlatma
- Android bildirim kanalı, önem ve öncelik değerlerini tanımlama
- Android ve iOS bildirim ayrıntılarını ayırma
- Bildirim iznini doğru zamanda isteme
- share_plus ile sistem paylaşım sayfasını açma
- Paylaşılacak metni model verilerinden güvenli biçimde üretme
- Eklenti bağımlılıklarını arayüz sınıflarıyla ayırma
- Sahte servislerle bildirim ve paylaşım widget testleri yazma

Çözümlü uygulama — Ders Hatırlatıcısı

Bildirim servisi sözleşmesi

```
abstract class BildirimServisi {  
  Future<void> hazirla();  
  Future<void> goster({  
    required String baslik,  
    required String govde,  
  });  
}
```

Arayüzün görevi bildirim paketiyle konuşmak değil, kullanıcı girdisini doğrulayıp bu sözleşmeyi çağırmaktır. Testte gerçek bildirim eklentisi yerine çağrılarını kaydeden sahte servis kullanılır.

Eklenti başlatma ve izin isteme

```
const ayarlar = InitializationSettings(  
  android: AndroidInitializationSettings('@mipmap/ic_launcher'),  
  iOS: DarwinInitializationSettings(),  
);  
  
await eklenti.initialize(ayarlar);  
  
await eklenti  
  .resolvePlatformSpecificImplementation<  
    AndroidFlutterLocalNotificationsPlugin>()  
  ?.requestNotificationsPermission();
```

Bildirim izni uygulama açılır açılmaz sebepsiz istenmez. Öğrenci “Bildirimi dene” işlemini seçtiğinde hazırlanır.

Bildirim kanalı ve içeriği

```
const ayrnti = NotificationDetails(  
  android: AndroidNotificationDetails(  
    'ders_hatirlatmalari',  
    'Ders Hatırlatmaları',  
    channelDescription: 'Öğrencinin oluşturduğu çağırma bildirimleri',  
    importance: Importance.high,  
    priority: Priority.high,  
  ),  
  iOS: DarwinNotificationDetails(),  
);  
  
await eklenti.show(31, baslik, govde, ayrnti);
```

Android'de kanal kimliği sonradan rastgele değiştirilmemelidir. Kullanıcı kanal ayarlarını işletim sistemi üzerinden yönetebilir.

Form doğrulaması ve kullanıcı geri bildirimi

```
if (konu.text.trim().isEmpty) {  
  setState(() => durum = 'Hatırlatma konusu boş bırakılamaz');  
  return;  
}  
  
await widget.servis.hazirla();  
await widget.servis.goster(  
  baslik: 'Çalışma zamanı',  
  govde: '${konu.text.trim()} • $dakika dakika',  
);
```

Bildirim içeriği sabit bir görsel değil; öğrencinin konu ve süre seçiminden oluşturulur.

Ders Hatırlatıcısı



Kendi çalışma bildiriminizi
oluşturun.

Çalışma konusu



Flutter tekrarını tamamla

Çalışma süresi: 15 dakika



Bildirimi dene



Bildirim cihaza gönderildi
Bildirim izni yalnız gerektiğinde istenir.

Ders Hatırlatıcısı gerçek Flutter ekranı

Widget testi düğmeye bastı; servisin hazırlandığını, başlığın “Çalışma zamanı” olduğunu ve gövdenin öğrenci girdisinden üretildiğini doğruladı.

Özgün uygulama — Acil Durum Bilgi Kartı

Bu uygulama ad, yakın telefonu ve sağlık notundan bir bilgi kartı üretir. Sağlık ve iletişim bilgileri hassas olduğundan otomatik paylaşım yapılmaz.

Paylaşım servisi

```

abstract class PaylasimServisi {
    Future<void> paylas(String metin);
}

class CihazPaylasimServisi implements PaylasimServisi {
    @override
    Future<void> paylas(String metin) async {
        await SharePlus.instance.share(
            ShareParams(
                text: metin,
                subject: 'Acil Durum Bilgi Kartı',
            ),
        );
    }
}

```

share_plus belirli bir mesaj uygulamasına zorla göndermez. İşletim sisteminin paylaşım sayfasını açar ve son seçimi kullanıcıya bırakır.

Paylaşım metnini üretme

```

String get paylasimMetni => [
    'ACİL DURUM BİLGİ KARTI',
    'Ad: ${ad.text.trim()}',
    'Yakınları: ${telefon.text.trim()}',
    'Sallık notu: ${not.text.trim()}',
].join('\n');

```

Eksik alanları engelleme

```

if ([ad, telefon, not].any((alan) => alan.text.trim().isEmpty)) {
    setState(() => durum = 'Karttaki bütün alanlar doldurun');
    return;
}

await widget.servis.paylas(paylasimMetni);

```

Acil Durum Bilgi Kartı

**Ece Yılmaz**
05... .. 42
Alerji: Fıstık

Ad soyad

Ece Yılmaz

Yakın telefonu

05... .. 42

Sağlık notu

Alerji: Fıstık

Kartı paylaş

 Sistem paylaşım menüsü açıldı
Sağlık ve iletişim bilgileri hassas veridir.

Acil Durum Bilgi Kartı gerçek Flutter ekranı

Test, uygulama açıldığında paylaşım yapılmadığını doğruladı. Yalnız “Kartı paylaş” düğmesine basıldıktan sonra üretilen metnin ad ve sağlık notunu içerdiğini ve başarı geri bildiriminin ekrana geldiğini denetledi.

Bildirim ile paylaşımın farkı

Özellik	Bildirim	Paylaşım
Amaç	Kullanıcıya uygulama dışından bilgi vermek	İçeriği başka uygulamaya aktarmak
İzin	Platform ve sürüme göre bildirim izni gerekir	Genellikle sistem menüsünü açmak için ayrı izin gerekmez
Son işlem	Uygulama bildirimi gösterir	Hedefi kullanıcı seçer
Hassas veri	Bildirim kilit ekranında görünebilir	İçerik seçilen hedefe gönderilebilir
Test	Sahte bildirim servisi	Sahte paylaşım servisi

Güvenlik ve kullanıcı deneyimi

- Bildirim başlığında hassas öğrenci bilgisi göstermeyin.
- Kullanıcının istemediği sık bildirimleri göndermeyin.
- Bildirim kanalının amacını açıkça yazın.
- Paylaşım öncesinde gönderilecek içeriği kullanıcıya gösterin.
- Sağlık, konum ve iletişim bilgilerini otomatik paylaşmayın.
- Hedef uygulamanın davranışının sizin kontrolünüzde olmadığını unutmayın.
- İzin reddedildiğinde uygulamanın temel işlevleri çalışmaya devam etmelidir.

Sık yapılan hatalar

- Eklenti başlatmadan bildirim göstermeye çalışmak
- Kanal kimliğini her bildirimde değiştirmek
- İzin reddini yakalamamak
- Bildirim kimliğini yönetmeden aynı kaydı istenmeden ezmek
- Paylaşımı sayfa açılır açılmaz başlatmak
- Boş veya doğrulanmamış içeriği paylaşmak
- Gerçek eklenti widget testinde çağırıp testi platforma bağımlı yapmak

Üç uygulama görevi

Kolay — Su içme hatırlatıcısı

Kullanıcıdan bardak hedefi ve hatırlatma metni alın. Bildirim izni reddedilirse ekranda çalışan sayaç özelliğini koruyun.

Orta — Ödev paylaşım özeti

Ders, ödev başlığı ve teslim tarihinden biçimlendirilmiş metin üretin. Paylaşım öncesinde önizleme ve eksik alan denetimi sunun.

İleri — Bildirim merkezi

Birden fazla ders için farklı bildirim kanalları oluşturun. Bildirimi açma, kapatma ve yenileme seçenekleri ekleyin; geçmiş bildirimleri yerel veritabanında tutun. Hassas içeriği kilit ekranına yazmayın.

Test ve kontrol

- Bildirim paketi platform ayarlarıyla başlatılıyor mu?
- Android kanal kimliği ve açıklaması anlamlı mı?
- İzin yalnız kullanıcı bildirim istediğinde soruluyor mu?
- Başlık ve gövde gerçek form verisinden oluşuyor mu?
- Boş konu bildirimi engelliyor mu?

- Paylaşım metni bütün gerekli alanları içeriyor mu?
- Paylaşım yalnız düğme işlemi sonrasında başlıyor mu?
- Hata durumunda kullanıcı anlaşılır geri bildirim alıyor mu?
- Testler gerçek platform kanalı yerine sahte servis kullanıyor mu?

Geri bildirim ve çözüm erişimi

Öğrenci uygulamasını gönderdikten sonra İpucunu göster, eklenti başlatma–izin–işlem sırasını açıklar. Örnek çözümü göster, gerçek bildirim kanalı, SharePlus, form doğrulama ve servis soyutlama kodlarını sunar. İndirme bağlantıları iki bağımsız Flutter projesini içerir.

Kaynak projeler

- Konu uygulaması: 08 - Cihaz Özellikleri ve Kalite/Uygulamalar/Uygulama-31-Hatırlatma
- Özgün uygulama: 08 - Cihaz Özellikleri ve Kalite/Uygulamalar/Ozgun-31-Acil Durum Bilgi Kartı