

FLUTTER

Proje Analizi ve Tasarım

33. hafta • Bütünleşik Proje ve Yayınlama

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
33	Proje Analizi ve Tasarım	Proje Taslağı • Okul Kulübü Planı

33. Hafta — Proje Analizi ve Tasarım

Bu hafta ne yapacağız?

Kod yazmaya başlamadan önce çözülecek problemi, kullanıcıyı, gereksinimleri ve ilk sürüm kapsamını belirleyeceğiz. Gereksinimleri kullanıcı hikâyesi biçiminde yazıp önceliklendiren bir proje taslağı geliştireceğiz. Özgün uygulamada aynı yaklaşımı okul kulübü için puanlanmış sprint planına dönüştüreceğiz.

Kazanımlar

- Problem cümlesi ve hedef kullanıcı tanımlama
- İşlevsel ve işlevsel olmayan gereksinimleri ayırt etme
- Kullanıcı hikâyesi yazma
- Gereksinimleri önceliklendirme
- İlk çalışan sürümün kapsamını belirleme
- Büyük işi küçük ve ölçülebilir iş paketlerine bölme
- Tahmin puanı, sorumlu ve tamamlanma durumunu izleme
- Analiz verisini gerçek Flutter model ve arayüzüne dönüştürme

Projeye başlamadan önce

“Bir kütüphane uygulaması yapacağım” yeterli bir analiz değildir. Daha net bir problem cümlesi gerekir:

Öğrenciler okul kütüphanesindeki kitapları hızlı bulamıyor ve okumak istedikleri kitapları düzenli takip edemiyor.

Hedef kullanıcı, temel ihtiyaç ve başarı ölçütü açıkça yazılmalıdır.

Kullanıcı hikâyesi

Yaygın kalıp:

[Kullanıcı] olarak [işlemi] yapmak istiyorum; çünkü [elde edeceğim fayda].

Örnek:

Öğrenci olarak kitabı adına göre bulmak istiyorum; çünkü rafta arama süresini azaltmak istiyorum.

Hikâye arayüz düğmesini değil, kullanıcı değerini anlatır.

Çözümlü uygulama — Proje Taslağı

Gereksinim modeli

```
enum Oncelik {
    olmalı,
    olmalıysaİyi,
    olabilir,
}

class Gereksinim {
    Gereksinim({
        required this.baslik,
        required this.hikaye,
        required this.oncelik,
        this.tamam = false,
    });

    final String baslik;
    final String hikaye;
    final Oncelik oncelik;
    bool tamam;
}
```

Öncelikleri kullanıcıya gösterme

```
String oncelikAdi(Oncelik oncelik) => switch (oncelik) {
    Oncelik.olmalı => 'Olmazsa olmaz',
    Oncelik.olmalıysaİyi => 'Olursa iyi',
    Oncelik.olabilir => 'Gelecek sürüm',
};
```

İlk sürümde “Olmazsa olmaz” maddeler tamamlanmadan gelecek sürüm özelliklerine geçilmez.

Doğrulama ve ekleme

```
if (baslik.text.trim().isEmpty ||
    hikaye.text.trim().isEmpty) {
    setState(
        () => mesaj = 'Başlık ve kullanıcı hikâyesi zorunludur',
    );
    return;
}

gereksinimler.add(
    Gereksinim(
        baslik: baslik.text.trim(),
        hikaye: hikaye.text.trim(),
        oncelik: oncelik,
    ),
);
```

İlerleme oranı

```
final tamamlanan =
    gereksinimler.where((g) => g.tamam).length;

final oran = gereksinimler.isEmpty
    ? 0.0
    : tamamlanan / gereksinimler.length;
```

Proje Taslağı

**Mini Kütüphane**
1/3 gereksinim hazır

Gereksinim başlığı

Kullanıcı hikâyesi
... olarak ... yapmak istiyorum; çünkü ...

☒ Olmazsa olmaz ☐ Olursa iyi ☐ Gelecek sürüm

 Ürün listesine ekle

Gereksinim ürün listesine eklendi

Kitap arama
Olmazsa olmaz
Öğrenci olarak kitabı adına göre bulmak istiyorum.

☒

Okuma listesi
Olmazsa olmaz
Öğrenci olarak okuyacağım kitapları kaydetmek istiyorum.

☐

Barkodla arama
Olmazsa olmaz
Öğrenci olarak barkodla kitabı bulmak istiyorum.

☐

Proje Taslağı gerçek Flutter ekranı

Test gerçek form alanlarına “Barkodla arama” gereksinimini ve kullanıcı hikâyesini yazdı, ürüne ekledi ve sonuç kartını doğruladı.

Özgün uygulama — Okul Kulübü Sprint Planı

Sprint, kısa bir çalışma dönemidir. Her iş paketi bir sorumluya ve görelî tahmin puanına sahiptir:

```
class IsPaketi {  
  IsPaketi(  
    this.ad,  
    this.sorumlu,  
    this.puan, {  
      this.tamam = false,  
    });  
  
  final String ad;  
  final String sorumlu;  
  final int puan;  
  bool tamam;  
}
```

Tamamlanan puanı hesaplama

```
final toplam =  
    paketler.fold<int>(0, (t, p) => t + p.puan);  
  
final biten = paketler  
    .where((p) => p.tamam)  
    .fold<int>(0, (t, p) => t + p.puan);
```

Görev sayısı yerine puan kullanmak, farklı büyüklükteki işlerin ilerlemeye etkisini daha gerçekçi gösterir.

İş paketi ekleme

```
paketler.add(  
    IsPaketi(  
        gorev.text.trim(),  
        sorumlu.text.trim(),  
        puan,  
    ),  
);
```

Okul Kulübü Sprint Planı

BİLİM KULÜBÜ • SPRINT 1

3 / 13 puan tamamlandı

İş paketi sprint planına eklendi

İş paketi

Sorumlu

Tahmin: 3 puan

İş paketi ekle

3

Afiş tasarımı
Zeynep • 3 puan



5

Katılım formu
Mert • 5 puan



2

Duyuru metni
Elif • 2 puan



3

Sahne akışı
Deniz • 3 puan



Okul Kulübü Sprint Planı gerçek Flutter ekranı

Test “Sahne akışı” iş paketini Deniz sorumluluğunda ve üç puanla ekledi; kartı ve güncel sprint mesajını doğruladı.

Kabul ölçütü yazma

Bir hikâyenin tamamlandığını doğrulamak için ölçülebilir kabul ölçütleri yazılır:

- Öğrenci en az iki karakter yazarak kitap arayabilmeli.
- Sonuçlar başlığa göre alfabetik gösterilmeli.
- Sonuç yoksa boş ekran yerine açıklayıcı mesaj görünmeli.
- Arama büyük/küçük harfe duyarsız olmalı.
- Liste 100 kitapta belirgin takılma yaşamamalı.

“Güzel çalışmalı” kabul ölçütü değildir; ölçülemez.

İlk sürüm kapsamı

Mini Kütüphane için ilk sürüm:

- Kitapları listeleme
- Adına göre arama
- Okuma listesine ekleme/çıkarma
- Yerel saklama
- Boş, yükleniyor ve hata durumları

Sonraki sürüm:

- Barkod tarama
- Kullanıcı hesabı
- Uzak sunucu eşitleme
- Sosyal yorumlar

Kapsam sınırlamak eksik iş yapmak değildir; test edilebilir bir sürümü zamanında tamamlamaktır.

Sık yapılan hatalar

- Hedef kullanıcıyı tanımlamadan özellik listesi yazmak
- Kullanıcı hikâyesine teknik çözümü gömmek
- Her gereksinimi aynı öncelikte görmek
- Çok büyük işi tek görev olarak bırakmak
- Kabul ölçütü yazmamak
- Tahmin puanını saat gibi kesin süre sanmak
- İlk sürüme bütün fikirleri eklemeye çalışmak

Üç uygulama görevi

Kolay — Sınıf nöbet planı

Hedef kullanıcıları ve üç temel gereksinimi yazın. Her gereksinim için kullanıcı hikâyesi, öncelik ve kabul ölçütü ekleyin.

Orta — Okul etkinlik uygulaması

Etkinlik bulma, kayıt olma ve hatırlatma gereksinimlerini sprint iş paketlerine bölün. Sorumlu ve puan atayın.

İleri — Kantin sipariş sistemi

Öğrenci, kantin görevlisi ve yönetici için ayrı ihtiyaçlar çıkarın. Çelişen gereksinimleri belirleyin, ilk sürüm kapsamını ve güvenlik kabul ölçütlerini hazırlayın.

Kontrol listesi

- Problem tek cümlede anlaşılabilir mi?
- Hedef kullanıcı belli mi?
- Her özellik kullanıcı değerine bağlanmış mı?
- Kullanıcı hikâyeleri doğru kalıpta mı?
- Gereksinimlerin öncelikleri var mı?
- İlk sürüm sınırları açık mı?
- Büyük işler küçük paketlere bölünmüş mü?
- Her paketin sorumlusu ve tahmini var mı?
- Kabul ölçütleri test edilebilir mi?
- Uygulama formu eksik veriyi engelliyor mu?

Geri bildirim ve çözüm erişimi

Öğrenci proje analizini gönderdikten sonra İpucunu göster, kullanıcı hikâyesi kalıbını ve öncelik sorularını verir. Örnek çözümü göster, gerçek gereksinim modellerini, form doğrulamasını, ilerleme hesabını ve sprint kodunu açar. İndirme bağlantıları iki çalışan Flutter projesini sunar.

Kaynak projeler

- Konu uygulaması: 09 - Bütünleşik Proje ve Yayınlama/Uygulamalar/Uygulama-33-Proje Taslağı
- Özgün uygulama: 09 - Bütünleşik Proje ve Yayınlama/Uygulamalar/Ozgun-33-Okul Kulübü Planı