



**Proje Adı:** Pişir

## 1.Giriş

Günümüzde birçok kişi yoğun iş temposu nedeniyle zaman yönetiminde zorlanmaktadır. Vakti olmayan insanlar internette yemek ararken seçenekler arasında kaybolmakta ve kendilerine uygun tarifi bulamamaktadırlar. Bu proje yemek tariflerini ulaşılması kolay bir şekilde kullanıcıya sunmaktadır.

Bu amaçla çıktığımız yolda flutter üzerinden firebase tabanlı bir uygulama geliştirmek istedik. Bu uygulama hedef kitlenin günlük hayatını kolaylaştırmak üzere geliştirildi.

## 2.Uygulama Süreci

### 2.1.Mimari Tasarım

Pişir uygulaması, modern mobil uygulama geliştirme prensiplerine uygun olarak modüler, yeniden kullanılabilir ve test edilebilir bir yapı üzerine inşa edilmiştir. Flutter ekosistemi içinde basit ve etkili bir mimari yapı kullanılarak geliştirilmiştir. Uygulama, widget-tabanlı bir yaklaşım benimser ve karmaşık state management kütüphanelerine bağımlılığı minimuma indirger.

#### 2.1.1.Katmanlar:

##### 1. View (Görünüm) Katmanı

Bu katman, kullanıcıya gösterilen arayüzü ve kullanıcı etkileşimlerini yönetir. Flutter'ın widget yapısı kullanılarak geliştirilen bu katmanda her ekran belirli işlevlere ayrılmıştır:

- **MainScreen:** Alt menü ile sekme yönetimi
- **HomePage:** Tarif arama ve kategori filtreleme
- **RecipeSearchPage:** Tarif adıyla arama işlemleri

- **PantryPage:** Mutfak dolabı malzeme yönetimi
- **FavoritesPage:** Favori tarif listesi
- **SettingsPage:** Uygulama ayarları ve tema yönetimi
- **RecipeDetailPage:** Detaylı tarif görüntüleme

## 2. Durum Yönetimi Katmanı

Uygulama, Flutter'ın yerleşik **StatefulWidget** ve **setState()** pattern'ini kullanarak basit ve etkili durum yönetimi gerçekleştirir.

## 3. Model / Veri Katmanı

Hibrit veri kaynağı yaklaşımı benimsenmiştir:

- **Yerel Veri:** Tarifler assets klasöründeki txt dosyalarından yüklenir. Tariflerin ana sayfada gösterecekleri veriler burada saklanır.
- **Bulut Veri:** Kullanıcı mutfak dolabı verileri ve tarif detayları Firebase Firestore'da saklanır.
- **Yerel Ayarlar:** Tema tercihleri SharedPreferences kütüphanesi ile kalıcı olarak saklanır

### 2.1.2.Verİ Akışı ve Bağlantılar

**Tarif Yükleme:** Uygulama başlangıcında yerel assets dosyasından tarifler yüklenir

**Mutfak Dolabı:** Device ID bazlı kullanıcı tanımlama ile Firestore'dan veri çekilir

**Arama İşlemleri:** Yerel tarif listesi üzerinde filtreleme yapılır

**Mutfak Dolabı Karşılaştırma:** Tarif malzemeleri ile kullanıcı dolabı karşılaştırılarak eksik malzemeler belirlenir

**Tarif Detaylarının Görüntülenmesi:** Tarife tıklandığında görüntülenecek detaylı bilgiler Firestore'dan çekilir

### 2.1.3.Kullanılan Başlıca Teknolojiler

Teknoloji	Kullanım Alanı
Flutter	Mobil Arayüz Geliştirme
Firebase Firestore	Kullanıcı ve Detaylı Tarif Verileri
SharedPreferences	Yerel Ayar Depolama
Assets(txt)	Basitleştirilmiş Tarif Verileri
Material Design 3	UI/UX Tasarım Sistemi
Device Identification	Kullanıcı Tanımlama

### 2.1.4.Tasarım Kararları

#### Hibrit Veri Yaklaşımı:

- Basitleştirilmiş tarifler yerel dosyada → Hızlı erişim
- Detaylı tarifler Firestore'da → Uygulama boyutunu düşürmek
- Kullanıcı verileri Firestore'da → Senkronizasyon

#### Basit Durum Yönetimi:

- StatefulWidget + setState() → Minimal bağımlılık, kolay anlaşılabilirlik
- Karmaşık bir durum yönetimi aracı yok → Hızlı geliştirme ve bakım

#### Modüler Widget Yapısı:

- Her ekran bağımsız widget → Test edilebilirlik ve yeniden kullanılabilirlik
- Temiz kod organizasyonu → Kolay bakım ve geliştirme

#### Performans Optimizasyonları:

- Sayfalama sistemi (15-20'li yükleme) → Bellek verimliliği
- Lazy loading → Başlangıç süresini optimize etme
- Yerel asset ön bellekleme → Ağ bağımlılığını azaltma

### **Kullanıcı Deneyimi Odaklı:**

- Akıllı geri navigasyon sistemi → Doğal kullanım deneyimi
- Dolap-tarif karşılaştırması → Kişiselleştirilmiş öneriler
- Koyu/Açık tema desteği → Kullanıcı tercihi esnekliği

## **2.2. Tasarım Metodolojisi:**

### **Agile Yaklaşım:**

Piştir uygulamasının geliştirilmesinde iterative ve incremental bir yaklaşım benimsenmiştir. Projenin doğası gereği sürekli kullanıcı geri bildirimleri alınarak özellikler geliştirilmiş ve iyileştirilmiştir.

### **MVP (Minimum Viable Product) Stratejisi:**

Geliştirme süreci, temel işlevsellikle başlayıp kademeli olarak özellik eklenmesi prensibine dayalı olarak planlanmıştır

#### **Aşama 1(MVP):**

- Temel tarif listeleme
- Basit tarif detay görüntüleme
- Temel navigasyon sistemi

#### **Aşama 2(Temel Özellikler):**

- Tarif arama işlevselliği
- Mutfak dolabı yönetimi
- Favori ekleme sistemi entegrasyonu

#### **Aşama 3 (Gelişmiş UX):**

- Akıllı dolap-tarif karşılaştırması
- Gelişmiş navigasyon sistemi
- Tema yönetimi ve kişiselleştirme

## 2.3. Geliştirme Aşamaları:

### Aşama 1: Temel Altyapı

- Widget hiyerarşisi tasarımı
- Firebase entegrasyonu
- Temel veri modellerinin oluşturulması

### Aşama 2: Temel UI Geliştirilmesi

- Ana ekran(MainScreen) ve sekme navigasyonu
- Ana sayfa(HomePage) tarif listeleme
- Tarif detay sayfası(RecipeDetailPage) implementasyonu
- Dolap-tarif karşılaştırma mantığı

### Aşama 3: Gelişmiş Özellikler

- Dolap sayfasının(PantryPage) geliştirilmesi
- Favoriler sayfasının(FavoritesPage) entegrasyonu
- Performans iyileştirmeleri
- Arama işlevselliği

### Aşama 4: Kullanıcı Deneyimi(UX) Enhancements

- Akıllı geri navigasyon sistemi
- Yükleme durumları ve hata yönetimi
- Koyu/Açık tema sistemi
- Mikro-animasyonlar ve geçiş efektleri

### Flutter Seçimi:

- Birden fazla platform geliştirme avantajı
- Hızlı prototipleme imkanı
- Zengin widget kütüphanesi
- Material Design 3 desteği

### **Performans Stratejileri:**

- **Lazy Loading:** Tarifler sayfalama ile yüklenir (15-20'lik gruplar)
- **Ön bellekleme Stratejisi:** Yerel assets klasörü + Firebase ön belleği
- **Bellek Yönetimi:** Widget yaşam döngüsü(lifecycle) optimizasyonu

## **2.4.Kalite Kontrol Süreci:**

### **Manuel test etme:**

- **UI/UX Testi** → Her özellik için manuel test
- **Performans Testi** → Yükleme süresi testleri
- **Entegrasyon Testi** → Firebase bağlantı testleri

### **Kod inceleme süreci:**

- Özellik bazlı kod inceleme
- Performans değerlendirmesi
- UI tutarlılığı
- Uygulama standartlarına bağlılık

## **2.5. Deployment:**

### **Sürekli Entegrasyon:**

- GitHub repo yönetimi

### 3.Grup Yapısı:

**KTÜ BİLGİSAYAR MÜHENDİSLİĞİ ÖĞRENCİLERİ**

**415562 Furkan Koçak -> Leader/Backend**

**425422 Ziya Mandıralı -> Code-Reviewer/Database/Backend**

**425496 Emre Kabaoğlu -> Backend/Frontend**

**425482 Veysel Karani Sevim -> Frontend/UI Designer**

### 4.Kaynakça:

16000'den daha fazla tarifi <https://yemek.com/> 'dan lokale çekmek için bir github reposundan faydalandık.

<https://github.com/ilyasozkurt/turkish-food-recipes>