

**T.C.
KOCAELİ ÜNİVERSİTESİ
BİTİRME PROJELERİ ARA RAPORU**

**Makine Öğrenmesi Destekli Etkin Madde
Tabanlı İlaç Öneri Sistemi**

PROJE NO: 1919B012401901

**Proje Yürütücüsü : Mustafa Toprak
Araştırmacılar : Mustafa Toprak
Proje Türü : Teknik Bilimler > Bilgisayar Bilimleri > Yazılım
Başlangıç Tarihi : 08/11/2024
Birim/Bölüm : Bilişim Sistemleri Mühendisliği
Ara Rapor Dönemi : 21/03/2025 - 04/04/2024
Ara Rapor No : 3**

KOCAELİ

BİTİRME PROJELERİ ARA RAPORU

1. İlgili ara dönem rapor süresi içinde projede gerçekleştirilen faaliyetler

Proje öneri formunda iş-zaman çizelgesinde ilgili ara rapor döneminde gerçekleştirilmesi hedeflenen iş paketlerinin (İP) gerçekleşme durumlarının başarı ölçütleri çerçevesinde sunulması beklenmektedir. Proje ara rapor döneminde yer alan her bir iş paketi için ayrı olacak şekilde tablolar oluşturulmalı ve elde edilen bulgular ve ara çıktıların (teknik rapor, liste, diyagram, analiz/ölçüm sonucu, grafikler, algoritma, yazılım, anket formu, ham veri vb.) detaylı sunulması beklenmektedir.

İP No	1
İP Adı	Uygulama Tasarımı ve Mimarisi
İP Tamamlanma Durumu (Yüzde Belirtilmelidir)	%50
İP Kapsamında Yapılan Çalışmalar ve Elde Edilen Bulgular <i>Elde edilen bulgular ve ara çıktıların (teknik rapor, liste, diyagram, analiz/ölçüm sonucu, grafikler, algoritma, yazılım, anket formu, ham veri vb.) detaylı sunulması beklenmektedir.</i>	<p>Flutter frameworkü kullanılarak geliştirilen uygulama için sürdürülebilirlik ve kod kalitesi açısından modüler bir yapı benimsenmiştir. Dosya mimarisi, "Bloc" mimari deseni etrafında organize edilmiş olup, belirli işlevler için sorumlulukların ayrıştırılması sağlanmıştır.</p> <p>Mevcut Dosya Mimarisi</p> <p>Uygulamanın dosya mimarisi aşağıdaki yapıda organize edilmiştir:</p> <ol style="list-style-type: none">lib/: Ana kaynak kod dizini<ul style="list-style-type: none">bloc/: Uygulama mantığını ve durum yönetimini içeren bileşenler<ul style="list-style-type: none">active_ingredient_bloc.dart: Etken madde verilerini yöneten bloc sınıfımedicine_cubit.dart: İlaç verilerinin durum yönetimipatient_cubit.dart: Hasta verilerinin durum yönetimipatient_detail_cubit.dart: Hasta detay sayfası için veri yönetimisign_in_bloc.dart: Giriş işlemleri için durum yönetimicomponents/: Yeniden kullanılabilir UI bileşenleri<ul style="list-style-type: none">not_found.dart: Veri bulunamadığında gösterilecek bileşenshow_snackbar.dart: Bildirim gösterimi için yardımcı bileşenconstants/: Uygulama genelinde kullanılan sabit değerler<ul style="list-style-type: none">app_colors.dart: Renk şemaları için tanımlamalarconstants.dart: Genel sabitler için tanımlamalarmodels/: Veri modelleri<ul style="list-style-type: none">request_models/: API istekleri için model sınıflarıresponse_models/: API yanıtları için model sınıflarıapi_response.dart: API yanıtlarını standardize eden modelservice/: Veri erişim ve işleme katmanı<ul style="list-style-type: none">active_ingredient_service.dart: Etken madde verilerini getiren servisapi_response.dart: API yanıtlarını işleyen yardımcı sınıfapiInfo.dart: API yapılandırması için bilgilermedicine_service.dart: İlaç verilerini getiren servismy_http_overrides.dart: HTTP istek özelleştirmeleripages_service.dart: Sayfa verilerini getiren servispatient_service.dart: Hasta verilerini getiren servissin_in_service.dart: Kimlik doğrulama servisi

- **views/**: Kullanıcı arayüzü ekranları
 - **main.dart**: Uygulamanın giriş noktası

Tasarım ve Mimari Özellikleri

1. **Bloc Pattern Uygulaması:**

- Kullanıcı arayüzü ile veri mantığının ayrıştırılması
- Reactive programlama yaklaşımı ile state yönetimi
- Bloc/Cubit sınıflarında ilgili iş mantığının enkapsüle edilmesi

2. **Servis Katmanı:**

- Hasta, ilaç, etken madde gibi farklı veri türleri için özelleştirilmiş servisler
- API isteklerinin yönetimi ve önbelleğe alma stratejileri
- Hata yönetimi ve ağ bağlantı durumlarının kontrolü

3. **Veri Modelleri:**

- İlaç bilgileri, hasta detayları, etken maddeler için ayrı model sınıfları
- Request/Response pattern ile API iletişimde tip güvenliği
- Serileştirme/deserileştirme işlemleri için yardımcı metotlar

4. **Uygulama Özellikleri:**

- Hasta bilgilerinin görüntülenebilmesi ve yönetimi
- İlaç bilgilerinin detaylı gösterimi
- Etken madde bilgilerinin listelenmesi ve detaylı incelenmesi
- İlaçların içerdiği etken maddelerin gösterimi
- Etken maddelerin hangi ilaçlarda bulunduğu gösterimi
- Hasta geçmişi ve tedavi planlarının yönetimi

Gerçekleştirilen İşler

1. Temel uygulama mimarisinin tasarlanması ve implementasyonu
2. Bloc pattern kullanılarak durum yönetimi altyapısının oluşturulması
3. Servis katmanında API iletişiminin yapılandırılması
4. Model sınıflarının temel yapısının oluşturulması
5. Hasta bilgileri ve ilaç verileri arasındaki ilişkilerin tanımlanması
6. Yeniden kullanılabilir UI bileşenlerinin geliştirilmeye başlanması
7. Uygulama renk şeması ve temel tasarım sabitlerinin belirlenmesi

Sonraki Adımlar

- UI ekranlarının görsel tasarımlarının tamamlanması
- Eksik servislerin implementasyonu

Bu aşamada, mimari yapının %50'si tamamlanmış olup, temel yapı oluşturulmuş ancak bazı ekran tasarımları ve servis entegrasyonları henüz tamamlanmamıştır.

--