

IOS İLE MOBİL UYGULAMA GELİŞTİRME



DÖNEM PROJE DOKÜMANI HEALTHCARE

Yılmaz Talha Kılıç
22290194

Github: <https://github.com/ylmztalhaklc/ios-app>

VideoLink:

https://drive.google.com/file/d/1xeTmowJ_WElq1wQClCbECH28EOu9_Ocu/view?usp=sharing

1. Proje Tanımı ve Amaç

HealthCare projesi, hasta yakınları ile hasta bakıcıları arasında yürütülen günlük bakım sürecinin dijital ortamda planlanması, takip edilmesini ve kayıt altına alınmasını amaçlayan bir mobil uygulamadır. Projede hasta bakımı süreci; görev oluşturma, görevlerin bakıcılara atanması, görev durumlarının güncellenmesi, görevlerle ilgili iletişim saflanması ve bakım performansının izlenmesi şeklinde bütüncül bir yapı olarak ele alınmıştır.

Gerçek hayatta hasta bakımında görevler çoğunlukla sözlü iletişim yoluyla aktarılmakta, bu durum görevlerin unutulmasına, yanlış anlaşılma veya yapılan işlerin doğrulanamamasına yol açmaktadır. Ayrıca kritik görevlerin zamanında fark edilmemesi ciddi sorunlar doğurabilemektedir. Bu proje ile hasta yakınının bakım sürecini merkezi bir yapı üzerinden yönetebilmesi, bakıcının ise kendisine atanmış görevleri net bir şekilde görmesi ve yaptığı işlemleri kayıt altına alabilmesi hedeflenmiştir.

Uygulama; Flutter kullanılarak geliştirilen mobil kullanıcı arayüzü, FastAPI tabanlı backend servisleri ve SQLite veritabanı ile entegre şekilde çalışmaktadır. Bu yapı sayesinde platformdan bağımsız, sürdürülebilir ve geliştirilebilir bir sistem oluşturulmuştur.

2. Proje Mimarisi ve Teknik Yapı

Proje, istemci–sunucu (client–server) mimarisi esas alınarak geliştirilmiştir. Frontend ve backend katmanları birbirinden ayrıdır ve aralarındaki tüm veri alışverişi REST tabanlı API’ler üzerinden sağlanmaktadır. Bu mimari yaklaşım, sistemin modülerliğini artırmaktır ve ileride yapılabilecek geliştirmeler için esnek bir yapı sunmaktadır.

Backend Yapısı

Backend tarafında FastAPI framework’ü tercih edilmiştir. Backend kod yapısı, her dosyanın ve bileşenin tek bir sorumluluğa sahip olacağı şekilde tasarlanmıştır.

- **main.py**
Uygulamanın giriş noktasıdır. API router’ları bu dosya üzerinden uygulamaya dahil edilmekte ve servis başlatılmaktadır.
- **models.py**
Veritabanında yer alan User, Task, Notification, Message ve PerformanceMetric gibi veri modelleri bu dosyada tanımlanmıştır. Bu modeller; görev yönetimi, mesajlaşma, bildirim ve performans takibi gibi temel işlevleri temsil etmektedir.
- **database.py**
SQLite veritabanı bağlantısını kurar ve uygulama genelinde kullanılan veritabanı oturumlarını yönetir.
- **auth.py**
Kullanıcı kimlik doğrulama ve yetkilendirme işlemlerini içerir.

- **routers/**

- auth_router.py: Kullanıcı kayıt ve giriş işlemleri
- task_router.py: Görev oluşturma, düzenleme, silme, görev durumu ve kategori güncellemleri
- notification_router.py: Bildirimlerin oluşturulması ve okunma durumlarının yönetimi
- message_router.py: Kullanıcılar arası mesajlaşma işlemleri
- metrics_router.py: Bakıcı performans metriklerinin hesaplanması ve listelenmesi

Görevler sistem içerisinde kategori bilgisiyle birlikte modellenmiştir. Özellikle **təhlilikli olarak işaretlenen görevler** için otomatik bildirim mekanizması devreye girmektedir.

3. Kullanıcı Rollerleri ve Uygulama İşleyışı

Uygulama, rol bazlı erişim modeli ile geliştirilmiştir. Sistemde iki temel kullanıcı rolü bulunmaktadır: **Hasta Yakını** ve **Hasta Bakıcı**. Her rolün eriştiği ekranlar ve gerçekleştirdiği işlemler farklıdır.

3.1 Giriş ve Kayıt Süreci

Uygulama açıldığında kullanıcılar giriş yapma veya yeni hesap oluşturma seçenekleri ile karşılaşır. Kayıt sırasında kullanıcıdan ad-soyad, e-posta adresi, şifre ve kullanıcı rolü bilgileri alınmaktadır. Kayıt işlemi tamamlandıktan sonra kullanıcı otomatik olarak sisteme giriş yapar.

Giriş işlemi başarıyla tamamlandığında kullanıcı, seçtiği role uygun ana ekrana yönlendirilir. Bu sayede kullanıcılar yalnızca kendi yetkileri dahilindeki işlemleri gerçekleştirebilir.

Resim 3.1: Kullanıcıların sisteme giriş yapabildiği ve yeni hesap oluşturabildiği giriş ve kayıt ekranları. Bu ekranda kullanıcıdan e-posta, şifre ve rol bilgileri alınmaktadır.

The image displays two side-by-side screenshots of the HealthCare application's user interface. Both screenshots feature a light gray header bar with the 'HealthCare' logo. Below the header, there are two main sections: 'Giriş Yap' (Login) on the left and 'Kayıt Ol' (Register) on the right.

Left Screenshot (Giriş Yap): This section shows two input fields: 'Email' and 'Şifre' (Password). Below these fields is a teal-colored 'Giriş Yap' button.

Right Screenshot (Kayıt Ol): This section shows four input fields: 'Ad Soyad' (Name Surname), 'Email', 'Şifre' (Password), and 'Rol'. The 'Rol' field has a dropdown menu set to 'Hasta Yakını' (Patient Relative). Below these fields is a teal-colored 'Kayıt Ol' button.

3.2 Hasta Yakını Ana Ekranı ve Görev Yönetimi

Hasta yakını sisteme giriş yaptığında bakım sürecini yönettiği ana ekran ulaşır. Bu ekranın hasta yakını tarafından oluşturulan tüm görevler tarih, saat ve durum bilgileri ile birlikte kart yapısında listelenmektedir.

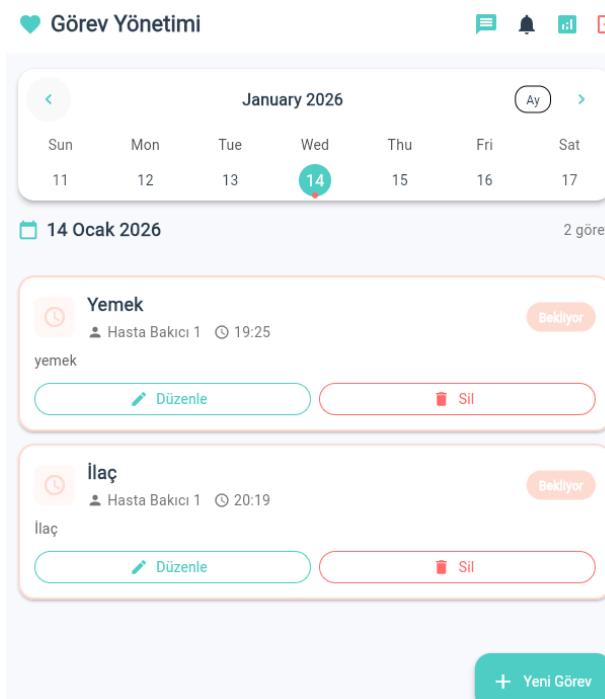
Hasta yakını bu ekran üzerinden:

- Yeni görev oluşturabilir
- Görev açıklaması, tarih ve saat bilgisi girebilir
- Görev kategorisini belirleyebilir (normal / tehlikeli)
- Görevi ilgili bakıcıya atayabilir

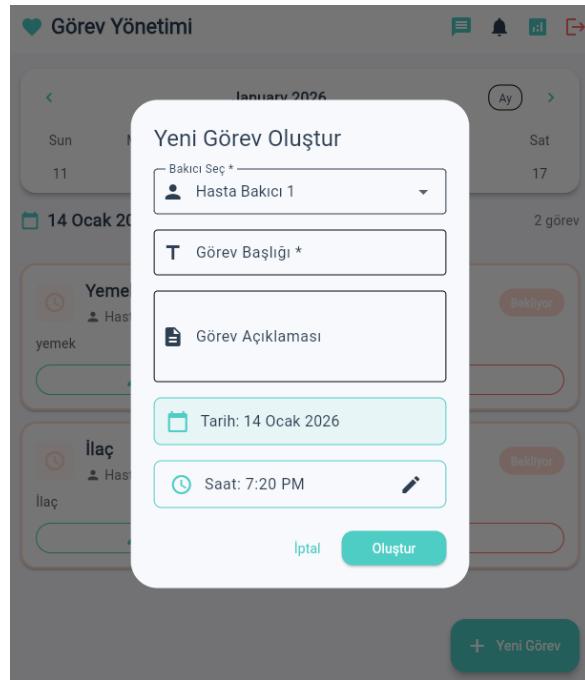
Görevler durumlarına göre renk kodları ile gösterilmektedir: beklemeye, tamamlandı ve sorun var. Hasta yakını görev kartları üzerinden görevleri düzenleyebilir veya silebilir. Yapılan her değişiklik sistem tarafından kayıt altına alınmakta ve bakıcıya otomatik bildirim gönderilirmektedir.

Tehlikeli olarak işaretlenen görevler oluşturulduğu anda bakıcıya **öncelikli otomatik bildirim** gönderilmektedir. Bu mekanizma sayesinde kritik görevlerin gözden kaçması engellenmektedir.

Resim 3.2: Hasta yakınının oluşturduğu görevlerin listelendiği ana ekran.



Resim 3.3: Yeni görev oluşturma ekranı. Görev açıklaması, tarih, saat, kategori ve bakıcı seçimi bu ekranada yapılmaktadır.



3.3 Hasta Bakıcı Ana Ekranı, Görev Güncelleme ve Chat

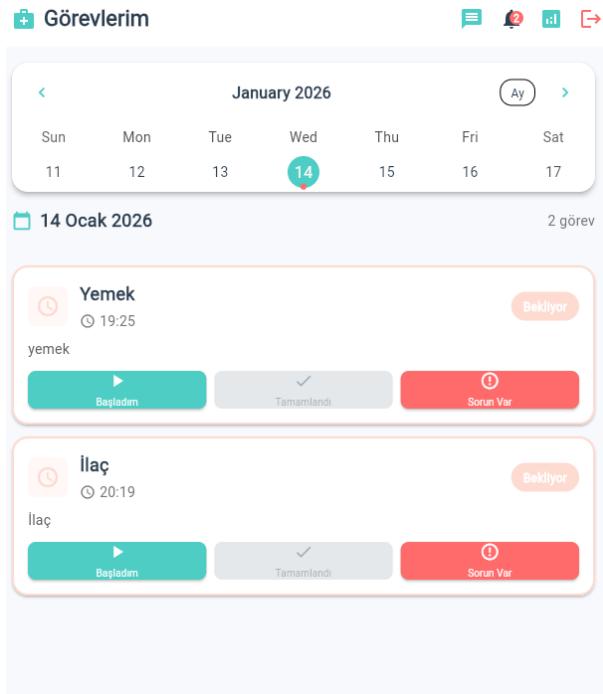
Hasta bakıcı sisteme giriş yaptığında kendisine atanmış görevleri görüntüler. Görevler, öncelik ve kategori bilgilerine göre listelenmektedir. Tehlikeli görevler görsel olarak ayırt edilmekte ve üst sıralarda gösterilmektedir.

Hasta bakıcı her görev için:

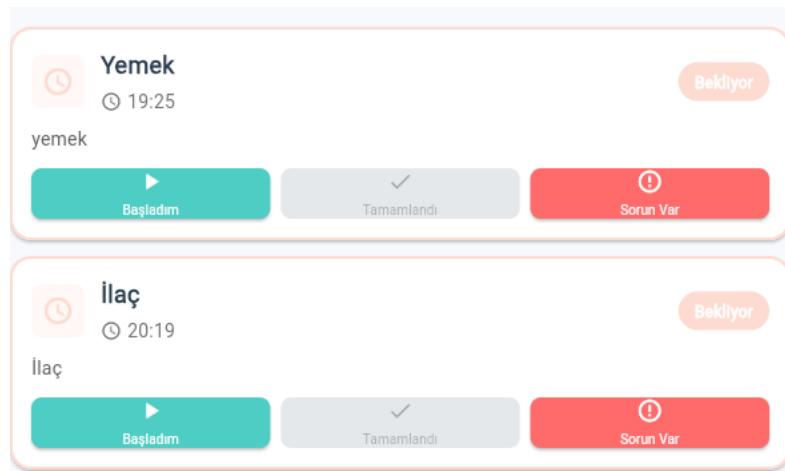
- Görevi tamamladığında “Yapıldı” olarak işaretleyebilir
- Görev sırasında yaşanan bir problemi “Sorun Var” olarak bildirebilir
- Sorun bildirimi sırasında açıklama ekleyebilir

Hasta yakını ile bakıcı arasında görevlerle ilgili iletişimini sağlanabilmesi için uygulama içerisinde **chat (mesajlaşma) ekranı** bulunmaktadır. Bu ekran üzerinden kullanıcılar görevlerle ilgili metin tabanlı mesajlaşma yapabilmektedir. Mesajlaşma yapısı, görev sürecinde yaşanan belirsizliklerin azaltılmasını amaçlamaktadır.

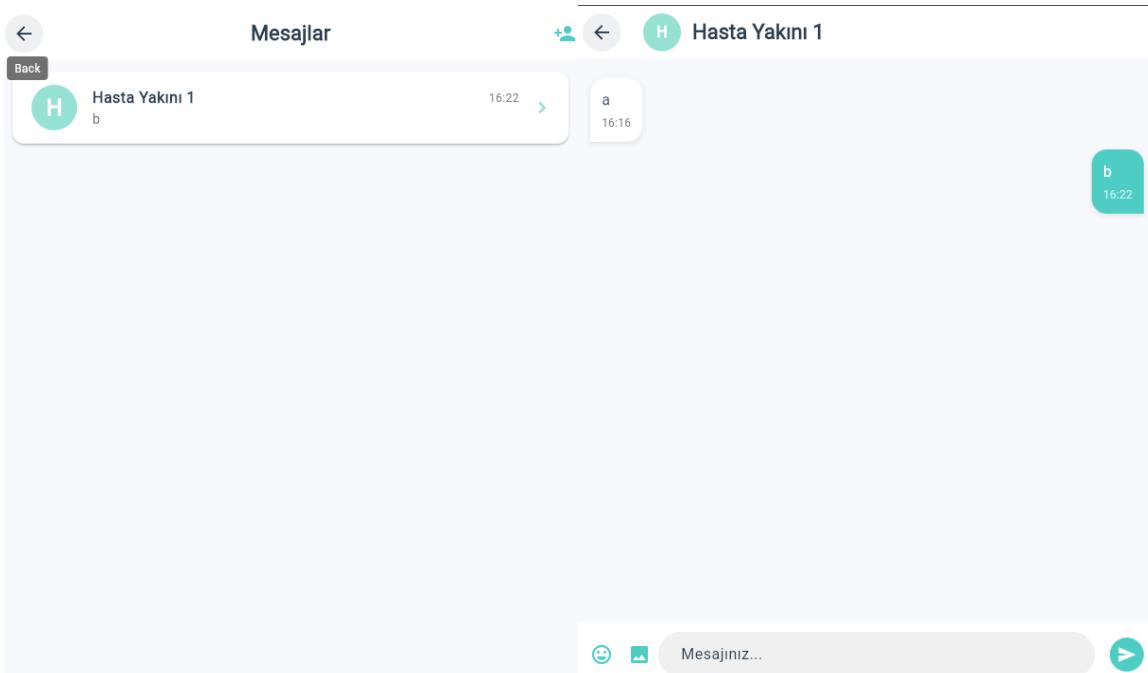
Resim 3.4: Hasta bakıcıya atanmış görevlerin listelendiği ekran.



Resim 3.5: Görev detay ve durum güncelleme ekranı.



Resim 3.6: Hasta yakını ve hasta bakıcı arasında görev bazlı mesajlaşmanın sağlandığı chat ekranı.



4. Bildirim Sistemi ve Performans Takibi

Uygulama içerisinde gerçekleştirilen tüm kritik işlemler bildirim sistemi ile desteklenmektedir. Bildirimler backend tarafından otomatik olarak oluşturulmakta ve ilgili kullanıcıya iletilmektedir.

Bildirim oluşturulan başlıca durumlar:

- Yeni görev oluşturulması
- Görev düzenlenmesi veya silinmesi
- Görevin tamamlanması
- Görev için sorun bildirilmesi
- Tehlikeli görev atanması

Buna ek olarak sistem, **bakıcı performans metriklerini** takip etmektedir. Bakıcıların tamamladığı görev sayısı, sorun bildirilen görevler ve tehlikeli görevlerdeki performans durumu hesaplanmakta ve hasta yakını tarafından görüntülenebilmektedir. Bu yapı, bakım sürecinin kalitesinin izlenmesini sağlamaktadır.

Resim 4.1: Uygulama içerisinde oluşturulan bildirimlerin listelendiği bildirim ekranı.

←

Bildirimler

Back

- 🔔 Yeni görev atandı. Tarih/Saat: 2026-01-14T19:25:00
14.01.2026 16:20**
- 🔔 Yeni görev atandı. Tarih/Saat: 2026-01-14T20:19:00
14.01.2026 16:19**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-22T22:00:00
22.11.2025 17:59**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-23T10:00:00
22.11.2025 17:57**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-23T10:00:00
22.11.2025 17:49**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-22T10:00:00
22.11.2025 17:49**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-22T09:00:00
22.11.2025 17:23**
- 🔔 Yeni görev atandı. Tarih/Saat: 2025-11-22T10:00:00
22.11.2025 16:48**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-22T09:00:00
22.11.2025 16:48**
- 🔔 Bir görevin zamanı güncellendi. Yeni tarih/saat: 2025-11-22T09:00:00
22.11.2025 16:48**

Resim 4.2: Bakıcı performans metriklerinin gösterildiği ekran.

← **Performans İstatistiklerim**

Genel Performans

Başarı Skoru
%0.0
Sorunsuz görev oranı



4	Toplam Görev
2	Tamamlanan
2	Bugün
0	Sorunlu

← **Performans İstatistiklerim**

%
%50.0
Tamamlanma

⚠
%0.0
Sorun Oranı

Son 7 Günün Özeti

Çar 14	0	0
Sal 13	0	0
Pzt 12	0	0
Paz 11	0	0
Cmt 10	0	0
Cum 9	0	0
Per 8	0	0