

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**MOBİL PROGRAMLAMA DERSİ FİNAL PROJESİ**

**TO WATCH LIST UYGULAMASI**

22120205064 Emine ÇAKAL

22120205383 Çağla SARF

**Ders Sorumlusu**

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Ocak, 2025

İstanbul Medeniyet Üniversitesi, İstanbul

# İÇİNDEKİLER

**Sayfa No**

[**İÇİNDEKİLER 1**](#_heading=h.41mghml)

[**1.**](#_heading=h.2grqrue) **GİRİŞ 1**

[1.1. Proje’nin Amacı 2](#_heading=h.vx1227)

[1.2. Proje Kısıtları 2](#_heading=h.3fwokq0)

[**2.**](#_heading=h.1v1yuxt) **MATERYAL VE YÖNTEM 3**

[2.1 Dart: Projede Kullanılan Yazılım Dili 3](#_heading=h.4f1mdlm)

[2.2. Android Studio: IDE 4](#_heading=h.28h4qwu)

[2.3. SQLite Veritabanı Paketi: Uygulamanın Veri Tabanı İçin Kullanılan Araç 4](#_heading=h.nmf14n)

[2.4. Figma: UI Tasarımları İçin Kullanılan Tasarım Platformu 5](#_heading=h.37m2jsg)

[2.5. Google Fonts: Yazı Font’u Seçimi İçin Kullanılan Platform 5](#_heading=h.1mrcu09)

[2.6. Color Hunt 5](#_heading=h.46r0co2)

[**3.**](#_heading=h.2lwamvv) **UYGULAMA 6**

[3.1 Önyüz Tasarım 6](#_heading=h.111kx3o)

[3.2 Veritabanı 6](#_heading=h.3l18frh)

[3.3. Uygulama’nın Ekranları ve İşlevleri 6](#_heading=h.206ipza)

[1. Ana Sayfa: To Watch List 8](#_heading=h.4k668n3)

[2. Yeni Gösterim Ekleme Ekranı: Add New Show 9](#_heading=h.2zbgiuw)

[**4.**](#_heading=h.1egqt2p) **SONUÇ VE TARTIŞMA 11**

[**6. KAYNAKÇA 12**](#_heading=h.3ygebqi)

[**7. EKLER 12**](#_heading=h.2dlolyb)

# GİRİŞ

Bu bölümde projenin amacı, projede karşılaşılan kısıtlar açıklanmıştır.

## 1.1. Proje’nin Amacı

Projenin amacı; Dart yazılım dili ve Flutter framework'ü ile "To Watch List" temalı bir Android Uygulaması geliştirmektir. İnternet üzerinde mobil uygulamalar araştırılmış ve “Watch List”, “To-Do List” tarzında uygulamaların popüler olduğu görülmüştür. Bu proje ile kullanıcıların izlemek istedikleri veya izledikleri film ve dizileri kolayca yönetebilecekleri bir araç sunulması hedeflenmiştir.

## 1.2. Proje Kısıtları

**Kısıt 1:** Uygulama geliştirilirken kullanılan IDE olan Android Studio’da bazı yapılandırma sorunları yaşanmıştır. Flutter ile uyumlu bir geliştirme ortamı oluştururken Flutter SDK ve Dart sürümlerinin güncel olmaması nedeniyle derleme hataları meydana gelmiştir. Bu sorun, Flutter SDK ve Dart sürümlerinin manuel olarak güncellenmesiyle çözülmüştür.

**Kısıt 2:** Sqflite paketinin kullanılmasını öğrenilmesi bir süreç gerektirmiştir. Gerekli araştırmalar gerçekleştirilerek, projenin hayata geçirilmesi sağlanmıştır.

**Kısıt 3:** Android emülatörünün çalıştırılması sırasında bazı hatalarla karşılaşılmıştır. Bu sorun, kullanılan Android sürümünün güncellenmesiyle giderilmiş ve emülatör sorunsuz bir şekilde çalıştırılabilmiştir.

# MATERYAL VE YÖNTEM

**● Projede kullanılan yazılım dili:** Dart

**● Projede kullanılan geliştirme ortamı:** Android Studio

**● Uygulamanın veri tabanı için kullanılan araç:** SQLite Veritabanı Paketi

**● Proje’nin ekran tasarımları, renk kodları ve genel olarak UI’ı üzerine**

**analizlerinin yapıldığı platform:** Figma

● **Uygulama içerisindeki yazı fontları için kullanılan platform:** Google Fonts

● **Uygulamadaki renk kodlarının belirlendiği platform:** Color Hunt

## 2.1 Dart: Projede Kullanılan Yazılım Dili

Bu projede, 2011 yılında Google tarafından geliştirilen Dart yazılım dili kullanılmıştır. Dart, açık kaynaklı bir dil olarak geliştirilmiş ve özellikle Flutter framework’ü ile mobil, web ve masaüstü uygulama geliştirme süreçlerinde popülerlik kazanmıştır.

Peki, neden mobil uygulama geliştirmede uzun zamandır kullanılan Java veya Kotlin yerine Dart tercih edilmiştir? Bu soru birçok geliştirici tarafından sorulmuş ve Google; “Performansı artırmak, verimli ve kolay bir geliştirme deneyimi sunmak için Dart’ı geliştirdik.” şeklinde cevap vermiştir. Dart ile yazılan ve diğer dillerle yazılan kodlar karşılaştırıldığında, Dart’ın daha hızlı geliştirme süreci sunduğu ve Flutter ile kolayca uyum sağladığı görülmektedir. Yani Dart’ın çıkış amacı, platform bağımsız bir şekilde kullanıcı arayüzlerini kolayca geliştirmek ve yüksek performanslı uygulamalar oluşturmaktır.

Dart, Hot Reload gibi güçlü bir özelliği sayesinde geliştiricilerin kod değişikliklerini anında görmesini sağlayarak geliştirme sürecini hızlandırır. Ayrıca, Dart’ın widget tabanlı yaklaşımı, Flutter framework’ü ile birlikte kullanıldığında, kullanıcı dostu ve performanslı arayüzlerin kolayca oluşturulmasını mümkün kılar.

logo, yazı tipi, grafik, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.1.** Dart Logo.

## 2.2. Android Studio: IDE

Proje geliştirilirken kullanılan geliştirme platformu olarak Android Studio seçilmiştir. Çünkü Android Studio; Android geliştirme için gerekli güncelleştirmeleri ve emülatör seçeneklerini olabilecek en iyi şekilde barındırmaktadır.

**kırpıntı çizim, simge, sembol, daire, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Şekil 2.2.** Android Studio Logo.

## 2.3. SQLite Veritabanı Paketi: Uygulamanın Veri Tabanı İçin Kullanılan Araç

Bu projede, veritabanı işlemleri için **SQLite** tercih edilmiştir. SQLite, hafif ve bağımsız bir ilişkisel veritabanı motorudur. Mobil uygulamalarda sıkça tercih edilmesinin temel nedeni, dış bir sunucuya ihtiyaç duymadan cihaz üzerinde çalışabilmesidir. SQLite, verilerin kalıcı bir şekilde saklanması ve hızlı bir şekilde erişilmesini sağlar.

SQLite’ın önemli özelliklerinden biri, platform bağımsız olması ve doğrudan uygulama ile entegre bir şekilde çalışabilmesidir. Bu özellik, veritabanı yönetimi sırasında yüksek performans sağlar ve uygulamanın offline (çevrimdışı) modda da çalışmasına olanak tanır. Ayrıca, ACID (Atomicity, Consistency, Isolation, Durability) standartlarını desteklemesi sayesinde güvenli ve tutarlı bir veri saklama sunar.

Projede **Sqflite** adlı Flutter paketi kullanılarak SQLite ile bağlantı kurulmuş ve veritabanı işlemleri gerçekleştirilmiştir. Sqflite, SQLite ile kolay bir şekilde veri ekleme, okuma, güncelleme ve silme işlemlerini yapmamıza olanak tanımıştır.

logo, grafik, metin, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.3.** SQLite logosu.

## 2.4. Figma: UI Tasarımları İçin Kullanılan Tasarım Platformu

Figma’nın seçilme nedeni yazılım sektöründe mobil uygulama ve web sitelerinin analizlerinin yapılandırılması sırasında en çok kullanılan uygulamalardan biri olmasıdır. Ayrıca analiz yaparken birçok hazır bileşen ve template kolaylığı sağlamaktadır.

metin, yazı tipi, grafik, logo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.4.** Figma Logo.

## 2.5. Google Fonts: Yazı Font’u Seçimi İçin Kullanılan Platform

Google Fonts içinde birçok yazı tipi barındırmaktadır. Uygulama da kullanılmak istenen FontFamily seçildikten sonra ilgili bilgiler klasör olarak indirilebilmekte ve projenin kodlandığı IDE’de projeye entegre edilebilmektedir.

yazı tipi, metin, tipografi, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.5.** Google Fonts Logo

## 2.6. Color Hunt

Color Hunt birbiriyle uyumlu renk paletlerinin görülebileceği ve renk kodlarının elde edilebileceği bir platformdur. Bu projede kullanılan renk kodlarının araştırması ve seçimi bu platformdan yapılmıştır.

metin, logo, grafik, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2.6.** Color Hunt Logo.

# UYGULAMA

## 3.1 Önyüz Tasarım

Uygulamanın arayüz tasarımında Flutter framework'ü kullanılarak Dart dilinde programlama yapılmıştır. Flutter’ın sağladığı "widget tabanlı" yapı, kullanıcı dostu bir arayüz tasarlamak için kullanılmıştır. Özellikle "To Watch List" ve "Watched" listelerini barındıran ana ekran, kullanıcıların kolay bir şekilde film ve dizi listelerini yönetmelerine olanak tanıyacak şekilde tasarlanmıştır.

Her bir öğe için içerik bilgisi (başlık, IMDb puanı, açıklama) ve etkileşim butonları (tamamlama ve silme) düzenlenmiştir. Arayüz, estetik ve işlevsellik göz önünde bulundurularak basit ve kullanıcı odaklı bir yapıya sahiptir. Ek olarak, yeni içerik ekleme ekranında kullanıcının başlık, kategori, IMDb puanı ve açıklama bilgilerini rahatlıkla girebileceği bir form tasarlanmıştır.

## 3.2 Veritabanı

Uygulamanın veri saklama ihtiyaçları için SQLite veritabanı kullanılmış ve bu işlem Flutter'da kullanılan "sqflite" paketi ile gerçekleştirilmiştir. Veritabanında, kullanıcıların eklediği dizi ve film bilgileri (başlık, kategori, açıklama, IMDb puanı ve tamamlanma durumu) tablo yapısında saklanmıştır. Sqflite paketi, verilerin yerel olarak saklanmasını ve uygulamanın çevrimdışı modda da çalışmasını sağlamıştır.

Veritabanı işlemleri için CRUD (Create, Read, Update, Delete) yöntemleri uygulanmış ve SQLite'ın performanslı yapısı sayesinde kullanıcı deneyimi iyileştirilmiştir. Bu CRUD işlemleri, "DatabaseHelper" sınıfında aşağıdaki gibi gerçekleştirilmiştir:

**- Create (Oluşturma):** Kullanıcı tarafından yeni bir film veya dizi eklendiğinde, `addMovie()` yöntemi çağrılarak içerik veritabanına kaydedilir.

- **Read (Okuma):** Tüm filmleri ve dizileri listelemek için `getMovies()` yöntemi kullanılır. Bu yöntem, veritabanındaki tüm kayıtları alır ve uygulama arayüzüne yansıtır.

- **Update (Güncelleme):** Bir içeriğin durumu (örneğin, izlenip izlenmediği) değiştirildiğinde `updateMovie()` yöntemi ile veritabanında güncellenir.

**- Delete (Silme):** Bir içerik listeden kaldırıldığında, `deleteMovie()` yöntemi ile veritabanından silinir.

## 3.3. Uygulama’nın Ekranları ve İşlevleri

Bu bölümde uygulamaya ait ekranlar paylaşılacak ve ekranların nasıl çalıştığı hakkında bilgiler verilmiştir.

elektronik donanım, metin, ekran görüntüsü, İletişim Cihazı içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 3.1.** Anasayfa.

metin, İletişim Cihazı, mobil cihaz, küçük alet içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 3.2.** Yeni İçerik Ekleme Ekranı.

### 1. Ana Sayfa: To Watch List

* Bu ekran, kullanıcının izlemek istediği veya daha önce izlediği film ve dizileri yönetebileceği iki temel listeyi içerir:

**"To Watch List" (İzlemek İstediklerim):** Kullanıcının izlemek istediği içerikleri gösterir. Bu listede:

* + - Her bir öğe, içeriğin başlığını, IMDb puanını ve kısa açıklamasını içerir.
    - Kullanıcı, bir içeriğin durumunu tamamlandı olarak işaretlemek için **onay kutusunu (checkbox)** kullanabilir.
    - Kullanıcı, bir içeriği listeden kaldırmak için **silme butonuna (çöp kutusu simgesi)** tıklayabilir.

**"Watched" (İzlediklerim):** Kullanıcının tamamladığı içerikleri gösterir. Benzer şekilde:

* + - Her bir öğe, içerik bilgilerini içerir.
    - Kullanıcı, bir içeriğin durumunu tekrar "izlenecek" olarak değiştirebilir.
    - Silme işlemi bu listede de yapılabilir.
* **Alt kısımda bulunan "Add New Show" (Yeni Gösterim Ekle) butonu** ile kullanıcı, yeni bir içerik eklemek için ekleme ekranına yönlendirilir.

### 2. Yeni Gösterim Ekleme Ekranı: Add New Show

* Bu ekran, kullanıcının yeni bir içerik ekleyebilmesi için gerekli alanları içerir:
  1. **Title (Başlık):** İçeriğin adının girileceği alan.
  2. **Category (Kategori):** Kullanıcı, içeriği **dizi** veya **film** kategorisi altında sınıflandırabilir.
  3. **IMDb Rating:** İçeriğin IMDb puanı bu alanda belirtilir.
  4. **Description (Açıklama):** İçeriğe dair kısa bir açıklama eklenir.
* Save (Kaydet) butonu ile kullanıcı, girdiği bilgileri kaydedebilir.
* Eğer kullanıcı kategori seçmezse, ekranın alt kısmında **"No category selected"** uyarısı görünür ve içerik kaydedilmez.
* Movie sınıfı, bir film veya diziyi temsil etmek için bir modeldir. Sınıf, her bir film veya dizi hakkında bilgileri tutar. Aşağıda Movie sınıfının tanımlandığı kod yer almaktadır.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 3.3.** Movie Sınıfı Kod Bloğu.

* Factory constructor Map’ten bir Movie nesnesi oluşturur. Bu, veritabanından alınan veriyi Movie nesnesine dönüştürmek için kullanılır. Aşağıda Factory constructor’una ait kod bloğu yer almaktadır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 3.3.** Movie Sınıfı Factory Yapıcı Methodu.

# SONUÇ VE TARTIŞMA

Bu projede, SQLite veritabanını kullanarak film ve dizi listesini yönetme işlevselliği başarıyla gerçekleştirilmiştir. Uygulama, kullanıcıların izlemek istedikleri ve izledikleri film ve dizileri kolayca kategorize etmesine olanak tanırken, aynı zamanda verilerin kalıcı olarak saklanmasını sağlamaktadır. Ancak, projenin bazı sınırlamaları ve geliştirilebilecek yönleri bulunmaktadır:

**Düzenleme İşlevinin Eksikliği:**  
Listeye eklenen dizi ve filmler için düzenleme seçeneği bulunmamaktadır. Kullanıcılar, yanlış girilen bilgileri düzeltmek istediklerinde, mevcut girdiyi silmek ve yeniden eklemek zorunda kalmaktadır. Gelecekte, liste öğelerine bir düzenleme seçeneği eklemek, kullanıcı deneyimini önemli ölçüde iyileştirebilir.

**Silme İşleminde Onay Sorgusu:**  
Listeye eklenen bir dizi veya film silinmek istendiğinde, kullanıcıdan bir doğrulama veya "Emin misiniz?" sorgusu alınmamaktadır. Bu durum, yanlışlıkla yapılan silme işlemlerine neden olabilir. Silme işlemi öncesinde bir onay ekranı eklemek, veri kaybını önlemek için önemli bir geliştirme olacaktır.

**Arama ve Filtreleme Özelliklerinin Eksikliği:**  
Uygulama, liste üzerinde bir arama veya filtreleme özelliği sunmamaktadır. Kullanıcılar büyük listelerde belirli bir öğeyi bulmakta zorluk yaşayabilir. Bu tür özellikler, uygulamanın kullanılabilirliğini artırabilir.

**Veri Görselleştirme ve Ek Kategoriler:**  
İzlenen ve izlenmek istenen içeriklerin görsel olarak grafiklerle sunulması gibi geliştirmeler, uygulamanın kullanıcı etkileşimini artırabilir. Ayrıca, içeriklere tür, yıl veya platform gibi ek kategoriler eklemek, kullanıcıların daha detaylı bir sınıflandırma yapmasını sağlar.

**Kullanıcı Geri Bildirimi Alınması:**  
Projede eksik veya iyileştirilmesi gereken alanlar hakkında daha fazla bilgi edinmek için kullanıcı geri bildirimine açık bir sistemin uygulanması faydalı olabilir.

Bu eksikliklerin giderilmesi ve yeni özelliklerin eklenmesi, uygulamanın kapsamını genişletecek ve kullanıcı deneyimini daha da iyileştirecektir. Bu proje, temel bir yapı sunarken, daha kapsamlı bir film ve dizi yönetim uygulaması geliştirmek için sağlam bir temel oluşturmaktadır.

# 6. KAYNAKÇA

[1] “GitHub,” [Online]. Available: <https://github.com/cagla003/to_watch_list>. [Accessed 02 01 2025].

[2] Simge tasarımı için ,<https://iconmonstr.com/> sitesini kullandık

[3] https://www.figma.com/design/nWHKrBTdjTXcpMp1XJfnfc/Todo-Mobile-App-(Community)-(Community)?node-id=1-2369&t=DVWoGULd14uHP4pM-0

[4] Uygulamamızın renklerini seçmek içi,<https://colorhunt.co/> sitesini kullandık.

[5] https://youtube.com/playlist?list=PLJbRSPP1eDeWJO9qKA7gI6eExO6ta44pc&si=NUwfcjQlPpcxJe0k

**7. ÖZGEÇMİŞ**

**Adı Soyadı:** EMİNE ÇAKAL  
 **Doğum Yeri ve Tarihi:** SİNOP 28.05.2004  
 **E-posta:** eminecakal668@gmail.com  
 **Yabancı Dil:** İNGİLİZCE B1  
 **Eğitim Durumu:**

* Lisans: Bilgisayar Mühendisliği, İstanbul Medeniyet Üniversitesi,2026
* Lise: Boyabat Fen Lisesi-Açıköğretim

**Adı Soyadı:** ÇAĞLA SARF  
 **Doğum Yeri ve Tarihi:** İSTANBUL 04.07.2003  
 **E-posta:** caglassarf@gmail.com  
 **Yabancı Dil:** İNGİLİZCE B2  
 **Eğitim Durumu:**

* Lisans: Bilgisayar Mühendisliği, İstanbul Medeniyet Üniversitesi,2026
* Lise: Ahmed Hamdi Gökbayrak Lisesi