

OtoVitrin Mobil Uygulması

Talha Kalyoncu
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
241307096
İstanbul/Üsküdar
talhakly02@gmail.com

Özet— Bu proje, ikinci el araçların alım satım işlemlerini kolaylaştırmayı ve pazarı dijitalleştirmemeyi amaçlayan “OtoVitrin” adlı mobil uygulamadır. Uygulama Android platformlarında çalışmaktadır ve Kotlin programlama dili kullanılarak geliştirilmiştir. Veri tabanı ve sunucu işlemleri için Google Firebase platformunun Firestore Database, Authentication ve Storage servisleri kullanılmıştır. Aynı zamanda uygulamanın ana sayfasında güncel dolar ve euro kuru ile benzin ve dizel fiyatlarının görüntülenmesi için harici API servislerinden yararlanılmıştır. Uygulama genel olarak kullanıcıların ilan verebildiği ve ilanları inceleyerek satıcıyla iletişime geçebildiği bir yapı sunar. Admin yetkisi olan kullanıcılar ise uygunsuz ilanları silebilir veya kullanıcı hesaplarını kapatabilir.

Anahtar Kelimeler— *Android, Kotlin, Firebase, Firestore, API, Mobil İlan Platformu*

1. GİRİŞ

Bu proje kapsamında geliştirilen “OtoVitrin” uygulaması, araç alım-satım süreçlerini modernize etmeyi ve kullanıcı deneyimini iyileştirmeyi hedeflemektedir. Uygulama, kullanıcıların detaylı filtreleme seçenekleriyle aradıkları araca en kısa sürede ulaşmalarını sağlarken, aynı zamanda satıcıların ilanlarını geniş bir kitleye ulaştırmamasına olanak tanır.

Projenin sadece bir ilan platformu olmasının ötesinde, kullanıcılarına ekonomik veriler sunması da amaçlanmıştır. Bu doğrultuda sisteme entegre edilen harici API servisleri sayesinde, kullanıcılar güncel döviz kurlarını ve akaryakıt fiyatlarını uygulama içerisinde anlık olarak takip edebilmektedir. Ayrıca, platformun güvenliği ve veri tutarlılığı için geliştirilen yönetici (Admin) paneli sayesinde, hatalı veya uygunsuz ilanların denetimi sağlanarak güvenli bir pazar yeri ortamı oluşturulmuştur.

2. YÖNTEM VE KULLANILAN TEKNOLOJİLER

A. Geliştirme Ortamı ve Dili

Uygulama, Google tarafından önerilen resmi Android geliştirme ortamı olan Android Studio üzerinde geliştirilmiştir. Programlama dili olarak, Java'ya kıyasla daha güvenli, kısa ve okunabilir kod yapısı sunan Kotlin tercih edilmiştir. Kotlin'in null-safety (boş değer güvenliği) özelliği sayesinde, uygulama çökümlerine neden olan hatalar minimuma indirilmiştir.

B. Veri Tabanı ve Bulut Hizmetleri

Bulut Hizmetleri için sunucu kurulumu gerektirmeyen Google Firebase platformu kullanılmıştır:

Firestore Database:

İlan verileri ve kullanıcı bilgileri, NoSQL tabanlı bu bulut veritabanında tutulmaktadır.

Firebase Storage:

Kullanıcıların yüklediği araç fotoğrafları burada güvenli bir şekilde depolanır.

Firebase Authentication:

Kullanıcıların güvenli bir şekilde kayıt olmasını ve giriş yapmasını sağlar.

C. Harici Servisler (API Entegrasyonu)

Uygulama, sadece bir ilan platformu değil, aynı zamanda ekonomik bir takip aracıdır. Bu amaçla iki harici servis sisteme entegre edilmiştir:

Frankfurter API:

Anahtarsız ve açık kaynaklı bir yapıya sahip olan bu servis üzerinden anlık Dolar ve Euro kurları çekilmektedir.

CollectAPI:

HTTP Header yetkilendirmesi (Auth Key) kullanılarak Türkiye geneli güncel Benzin ve Motorin fiyatları kullanıcıya sunulmaktadır.

D. Performans ve Kullanıcı Deneyimi Kod yapısında performansı artıran iki kritik kütüphane kullanılmıştır:

Glide:

İlan listelerindeki yüksek çözünürlüklü araç resimleri, bellek (RAM) tasarrufu sağlamak için ölçeklendirilerek yüklenir.

Karanlık Mod (Dark Mode):

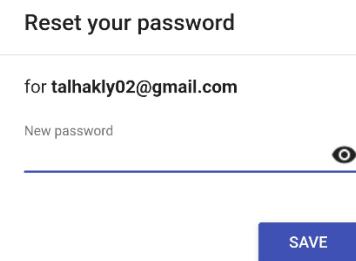
Uygulama, cihazın sistem temasını (Gece/Gündüz) algılayarak arayüz renklerini otomatik olarak kullanıcının tercihine göre ayarlar.

3. UYGULAMA DETAYLARI VE AKIŞ DİYAGRAMI

A. GİRİŞ KISMI

Uygulama açılışında kullanıcıyı karşılayan bu modül, Firebase Authentication altyapısını kullanır. Kullanıcı kayıtlı değilse; Ad-Soyad, Telefon, E-posta ve Şifre bilgileriyle sisteme kayıt olur. Giriş işlemi sırasında "Şifremi Unuttum" özelliği sayesinde, kullanıcının e-posta

adresine Firebase tarafından otomatik bir sıfırlama bağlantısı gönderilir



- Oturum Kontrolü (Session Management):** Cihaz üzerinde daha önce giriş yapılmışsa, uygulama açılışında FirebaseAuth.CurrentUser kontrolü sağlanır. Oturum aktifse, kullanıcı giriş ekranını görmeden doğrudan Ana Sayfaya yönlendirilir.
- Pasif Hesap Kontrolü:** Giriş yapan kullanıcının veritabanındaki hesapAktif durumu sorgulanır. Eğer hesap "Pasif/Donuk" ise, kullanıcıya bir AlertDialog gösterilir ve hesabı tekrar aktifleştirmek isteyip istemediği sorulur.

Hesabınız Dondurulmuş

Hesabınızı daha önce dondurmuşsunuz. Tekrar aktif edip giriş yapmak ister misiniz?

[HAYIR](#) [EVET, AÇ](#)

B. ANA SAYFA

Ana sayfa yüklenirken sistem, kullanıcının rolünü (Admin/User) sorgular ve arayüzü buna göre dinamik olarak şekillendirir. Bu sayede tek bir tasarım dosyası üzerinden hem Yönetici hem de Son Kullanıcı deneyimi yönetilerek kod tekrarı (DRY Prensibi) önlenmiştir.

KullaniciRolu = KULLANICI_VERISI.GetString("rol")

EĞER KullaniciRolu == "Admin" İSE:

//Kullanıcı Yöneticisi ise gizli butonları aç

YonetimButonu.Gorunurluk = GORUNUR(VISIBLE)

IstatistikButonu.Gorunurluk = GORUNUR (VISIBLE)

DEĞİLSE:

YonetimButonu.Gorunurluk = GIZLI(GONE)

IstatistikButonu.Gorunurluk = GIZLI (GONE)

Bu visible ve gone işlemini ilan görüntülemeye de kullanılmıştır bu sayede hem admine hem kullanıcıya farklı tasarımlar yapıp kod kalabalığından (kod tembelliğinden) kaçınılmıştır.

Ana sayfada birde popup menüsü bulunmaktadır. Popup menüsünden ayarlara gidilenir yada hesaptan çıkış yapıp giriş ekranına geri dönülebilir.

Vitrin Butonu:

Vitrin butonu bizi ilanların listelendiği bölüme götürür ve gerekli filtrelemeleri yapma seçeneği olur.

İlan Ekle Butonu:

Yeni bir ilan eklemek istendiğinde bizi gerekli ekrana götürür.

Yönetim Vitrini Butonu:

Vitrin butonu gibi ilanları listeler ve admin ilanları inceleyerek uygunsuzluk durumu olup olmadıklarını inceler.

İstatistik Raporu:

Admine kaç ilan olduğunu, üye sayısını, pasif üye sayısını ve ortalama ilan fiyatı olan istatistikleri bir cardview ile anasayfa ekranın üzerinde gösterir.

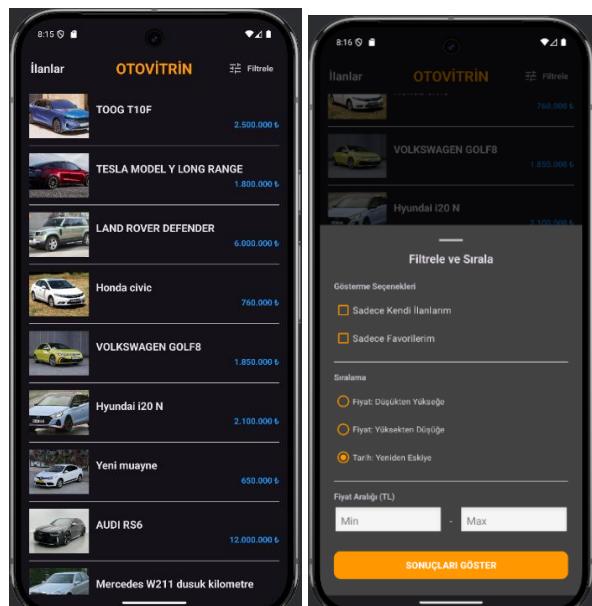
Ana sayfada internetten çektiğimiz 2 API yi de görebiliyoruz. Dolar ve euro kuru ile benzin ve motorin(İstanbul iline göre fiyatları) fiyatlarını görüntüler.

C. VİTRİN VE İLAN GÖRÜNTÜLEME

Vitrin: Tüm ilanların RecyclerView kullanılarak listelendiği alandır. Kullanıcılar burada detay sayfasına girmeden ilanın temel bilgilerini (Fiyat, Başlık, Resim) görebilir.

Filtreleme: Kullanıcılar; fiyat aralığı (Min-Max), tarih sıralaması (Yeniden eskiye) ve "Sadece Favorilerim" / "Sadece Kendi İlanlarım" gibi kriterlere göre dinamik sorgular oluşturabilir.

Yönetim Vitrini (Admin): Admin yetkisine sahip kullanıcılar, bu ekranla uygunsuz gördükleri ilanları anında yayından kaldırabilir.



İlan Görüntüleme:

Uygulamanın detay sayfaları, o an uygulamayı kullanan kişinin kimliğine göre şekillenir. Tek bir XML tasarımları (fragment_ilan_goruntuleme), arka plandaki Kotlin kodu ile üç farklı senaryoya uyum sağlar

İlan Sahibi:

Kendi ilanını görüntüleyip "Düzenle" ve "Sil" butonlarını görür.

Admin (Yönetici):

Sistem yönetici ise "Kullanıcıyı Banla" paneli aktif olur.

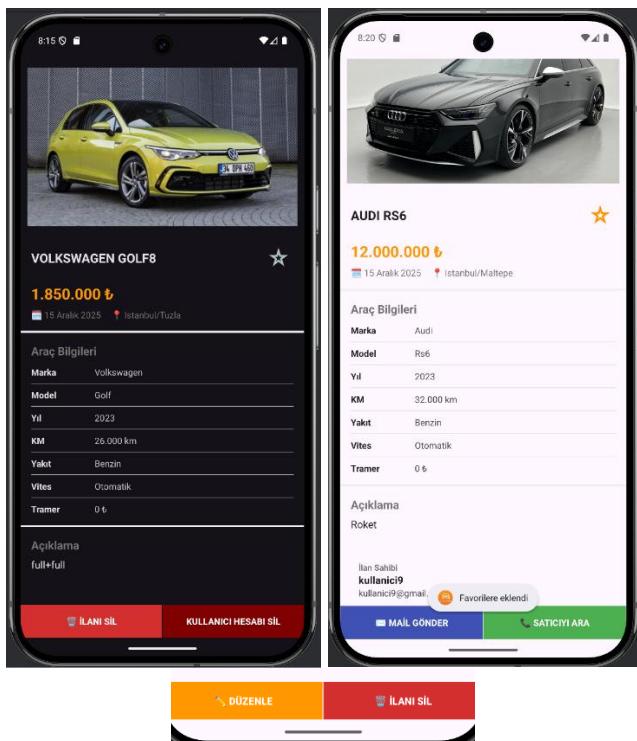
Ziyaretçi:

Sadece "Mail Gönder" ve "Ara" iletişim butonlarını görür.

Favoriye Alma:

Yıldız ikonundan kullanıcı beğendiği ilanı favoriye alabiliyor favoriye aldığından ikon sarı renge döner.

Okunurluk açısından fiyat, kilometre gibi değerlerin rahat okunabilmesi için binlik sistemden araya nokta konan bir fonksiyon ile okunurluğu çok fazla artırmıştır.

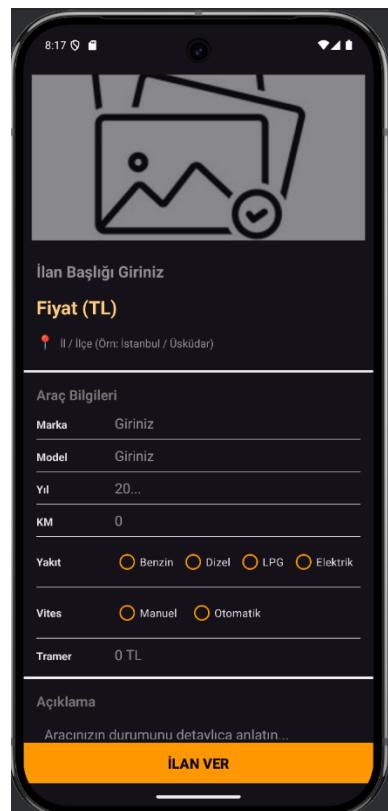


D. İLAN EKLEME / GÜNCELLEME

Bu modül, hem yeni ilan ekleme hem de mevcut ilanı düzenleme işlevini tek bir Fragment üzerinde gerçekleştirir.

- İzin Yönetimi:** Uygulamanın farklı Android sürümlerinde çalışılabilmesi için galeri erişiminde; Android 13+ için READ_MEDIA_IMAGES, eski sürümler için READ_EXTERNAL_STORAGE izinleri dinamik olarak yönetilir.
- Veri Kaydı:** İlan verileri (Marka, Model, Yıl, Yakıt, Vites vb.) Firestore Database üzerinde, ilan fotoğrafları ise Firebase Storage üzerinde tutulur. Resim yükleme işlemi tamamlanmadan veritabanı kaydı oluşturulmaz, bu sayede veri bütünlüğü sağlanır.

Fiyat, konum, ilan başlığı ilan açıklaması, aracın; resmi, kilometresi, trameri , markası , modeli ,yılı , yakıt ve vites seçenekleriyle 11 adet veri firestore database' inde toplanır. Resim öncelikle Storage a kaydedilir, resmin url i linki firestore a kaydedilir.



E. İSTATİSTİK RAPORU(SADECE ADMİN İÇİN)

Admine kaç ilan olduğunu, üye sayısını, pasif üye sayısını ve ortalama ilan fiyatı olan istatistikleri bir cardview ile anasayfa ekranının üzerinde gösterir.

Diaolog fragmentta buton olarak kapatma butonu bulunur. Ve toplanan 4 veriyi gösterir.

a) İlan Sayısı

Aktif görünümde olan toplam ilan sayısını gösterir.

b) Üye Sayısı

Adminler ve pasif kullanıcılar ile birlikte bulutta kaç kullanıcı olduğu görünür. Pasif üyelerin görülmeye sebebi tekrar hesabı aktifleştirip geri dönenilecek olabileceğinden görünülmesi gereklidir. Onlarda hala üye ve bulut veri tabanında yer kaplar.



c) Ortalama İlan Fiyatı

Bütün ilanların fiyatlarının toplamının ilan sayısına oranlandığı ortalama 1 ilan fiyatının ne kadar olduğunu gösterir.

Araba piyasasıyla alakalı ilan sayısı arttıkça doğruluğu da artan bir ortalama piyasanın ne olduğunu hangi fiyat bandında gezdiğini görüntüler.(İleride projenin gelişiminde markaya göre ortalama fiyatların görüntülenebileceği ve ayrıca sağlanan veri ile kullanıcı ilan koyarken aracının benzer özelliklerine sahip araçları ortalama fiyatını elde edip aracını satmaya karar veren kullanıcıya yardımcı olabilir.)

d) Pasif Üye Sayısı

Kullanıcının kendi arzusuyla hesabını pasifleştirdiği pasiflik durumunun kaç kullanıcı yaptığı admin görüntüler.

F. AYARLAR SEKMESİ

Karanlık mod:

Kullanıcının isteyebileceği karanlık ya da açık tema istegini sağlar ve uygulama kapandıktan sonra cihaz hafızasına atılır ve uygulama tekrar kullanılmak istenildiğinde hangi modda bırakılmışsa o modda tema gösterir.

Bilgi Güncelleme:

Ayarlar sekmesinde kullanıcı ad soyad, telefon numarası ve şifresini değiştirebilir. Kullanıcı bilgilerini düzenleyebilir.

Hesabı Dondurma(Pasifleştirme):

Pasiflik butonu ile hesabını pasifleştirebilir. Eğer hesabını pasifleştirirse firebase **batch** ile hepsini toplayıp yanında olan bütün ilanları silinir. Kullanıcı tekrar aktifleştirmek istediğiinde silinmiş olaan ilanlarına erişemez ve tekrar yüklenmez. Adminde ise pasiflik butonu bulunmaz. Adminin hesabını pasifleştirmesi mantıklı olmaz.

FONKSİYON HesabıDondurVeTemizle(userId):

// 1. Adım: Kullanıcının veritabanındaki tüm ilanlarını bul

ILANLAR = Firestore.Sorgula("ilanlar").Where("sahipId", userId)

// 2. Adım: Toplu işlem bohçası (Batch) oluştur

ISLEM_PAKETI = Firestore.Batch()

// 3. Adım: Her ilanı silme listesine ekle

DONGU (Her İlan İçin):

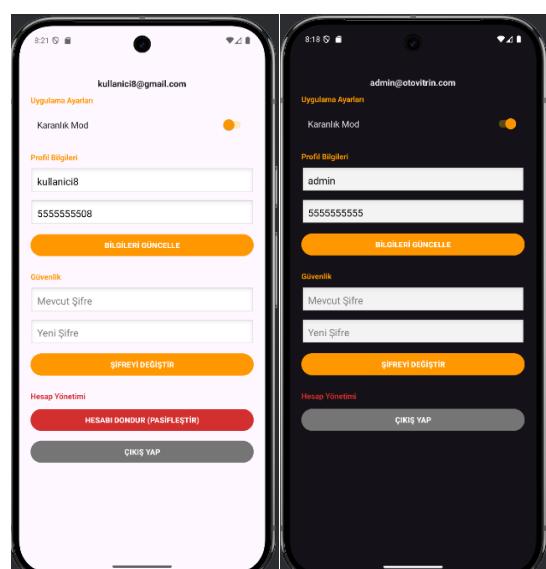
ISLEM_PAKETI.Sil(ilan.Referans)

// 4. Adım: Kullanıcıyı pasif duruma getir

ISLEM_PAKETI.Guncelle("Users/" + userId, "aktiflik", FALSE)

// 5. Adım: Paketi tek seferde uygula

ISLEM_PAKETI.Commit()



4.SONUÇ

Bu proje çalışmasında, ikinci el araç piyasası için güvenli, performanslı ve kullanıcı dostu bir mobil uygulama ("OtoVitrin") geliştirilmiştir. Kotlin dili ve Firebase teknolojileri kullanılarak oluşturulan sistem, sadece statik veri sunmakla kalmamış; Frankfurter ve CollectAPI servisleri entegre edilerek finansal bir takip aracına dönüştürülmüştür. Projenin teknik kazanımları arasında; veri bütünlüğünü sağlayan Batch Write algoritmaları, dinamik arayüz yönetimi ve performans optimizasyonları yer almaktadır.

5.KAYNAKÇA

- [1] developer.android.com.
- [2] firebase.google.com/docs
- [3] Frankfurter API, "Open Source Exchange Rates," frankfurter.app.
- [4] CollectAPI, "Turkey Fuel Price API Documentation," collectapi.com.
- [5] Bumptech, "Glide Image Loading Library for Android," github.com/bumptech/glide.

6.LİNKLER

Github: https://github.com/talha-kalyoncu/Oto_Vitrin

Drive:

https://drive.google.com/drive/folders/16qLfTeSZb-NIkGr9-Kojpf_HKYN9rAfZ?usp=drive_link