

**ANKARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM 4538 IOS İle Mobil Uygulama Geliştirme**  
**PROJE ÖDEV RAPORU**

**Hasan Tamer Tefon - 20290295**

GitHub: <https://github.com/tamertfn/sokak-hayvanlari-takip-uygulamasi>

Video ve APK linki: <https://drive.google.com/drive/folders/1wLLc8-TFfUpxc00L1q0-FwoX8B2mS40I>

**19.06.2025**

## Açıklamalar

Hocam size ilk verdiğimiz raporun bir örneğini githubda bulabilirsiniz.

GitHubdaki kaynak kod api keyleri sakladığımdan vs. expo start ile çalışmayacaktır, Apk'yı GitHub veya kapak sayfasındaki google drive linkinden edinebilirsiniz.

Bildirim sistemi yapmayı planlıyordum ancak yapamadım. Facebook ve Google ile girişi bilerek tamamlamadım.

GitHubdaki kaynak kodlar build'in iki sürüm öncesidir son apk ile birebir aynı değildir.

## İçindekiler

Açıklamalar .....	2
1. Proje Tanımı.....	4
2. Geliştirme Süreci .....	4
3. Uygulama Özellikleri .....	4
3.1 Kullanıcı Giriş ve Kayıt Sistemi.....	4
3.2 Patilerim Ekranı.....	4
3.3 Harita Ekranı .....	4
3.4 Profilim Ekranı .....	4
3.5 Pati Detayları Ekranı.....	5
4. Teknik Detaylar .....	5
4.1. Proje Yapısı .....	5
4.2. Veri Modelleri .....	5
4.3. Kullanılan API ve Servisler .....	6
5. Güvenlik ve Hata Yönetimi.....	6

## 1. Proje Tanımı

Bu proje, sokak hayvanlarının etkin bir şekilde takip edilmesi, durumlarının izlenmesi ve kullanıcıların etkileşimini artırmak amacıyla geliştirilmiş bir mobil uygulamadır.

Kullanıcılar uygulama üzerinden harita bazlı olarak sokak hayvanlarının bilgilerini paylaşabilir ve yönetebilir.

## 2. Geliştirme Süreci

1. Proje altyapısı oluşturuldu.
2. Firebase servisleri entegre edildi.
3. UI tasarımları geliştirildi.
4. Veri modelleri oluşturuldu ve CRUD işlemleri eklendi.
5. Gerçek zamanlı güncellemeler sağlandı.
6. Kullanıcı deneyimini artırıcı iyileştirmeler yapıldı

## 3. Uygulama Özellikleri

### 3.1 Kullanıcı Giriş ve Kayıt Sistemi

- Email ve şifre ile giriş (Firebase Auth)
- Yeni kullanıcı oluşturma
- Şifre yenileme seçeneği
- Güvenli kimlik doğrulama

### 3.2 Patilerim Ekranı

- Kullanıcının eklediği sokak hayvanlarının listesi
- Sağlık durumu ve yemek durumunu yönetme
- Not ve fotoğraf ekleme
- Favori olarak işaretleyebilme
- Yorum ekleme ve görüntüleme

### 3.3 Harita Ekranı

- Sokak hayvanlarının harita üzerinde gösterilmesi
- Sağlık durumuna göre filtreleme
- Yeni pati ekleme olanağı
- Hayvanların detaylarına hızlı erişim

### 3.4 Profilim Ekranı

- Kullanıcı bilgilerinin düzenlenmesi
- Favori hayvanların listelenmesi ve detaylarının incelenmesi

### 3.5 Pati Detayları Ekranı

- Hayvan detaylarının görüntülenmesi
- Diğer kullanıcıların eklediği hayvanların harita üzerinden incelenebilmesi
- Favorilere ekleme ve yorum yapma

## 4. Teknik Detaylar

- Firebase Firestore entegrasyonu
- Gerçek zamanlı veri senkronizasyonu
- Responsive ve kullanıcı dostu arayüz

### 4.1. Proje Yapısı

```
/app
├── /(tabs)
│   ├── index.tsx (Harita ekranı)
│   ├── patilerim.tsx (Patilerim ekranı)
│   ├── profilim.tsx (Profil ekranı)
│   └── yeni-pati.tsx (Yeni pati ekleme ekranı)
├── login.tsx (Giriş ekranı)
├── register.tsx (Kayıt ekranı)
└── user-patiler/[userId].tsx (Kullanıcı pati detay ekranı)
```

```
/src
├── config (Firebase, Cloudinary ayarları)
├── hooks (useAuth.ts - kimlik doğrulama)
├── services (animalService.ts - CRUD servisleri)
├── types (animal.ts, env.d.ts)
└── utils (imageUpload.tsx)
```

### 4.2. Veri Modelleri

```
type Pati = {
  id: string;
  name: string | null;
  healthStatus: string;
  imageUrl: string;
  createdAt: any;
  userId: string;
  hasFood: boolean;
  notes: string | null;
  location: {
    latitude: number;
    longitude: number;
  };
};
```

```
};
```

```
type Comment = {  
  id: string;  
  patId: string;  
  userId: string;  
  text: string;  
  createdAt: any;  
};
```

```
type FavoritePati = {  
  id: string;  
  name: string | null;  
  healthStatus: string;  
  imageUrl: string;  
  location: {  
    latitude: number;  
    longitude: number;  
  };  
  hasFood: boolean;  
  notes: string | null;  
  userId?: string;  
};
```

#### 4.3. Kullanılan API ve Servisler

- Firebase Authentication ve Firestore
- Cloudinary (Görsel yükleme)
- Google Maps API (Harita entegrasyonu)

#### 5. Güvenlik ve Hata Yönetimi

- Firebase güvenlik kuralları kullanıldı.
- Veri doğrulama ve hata yakalama mekanizmaları oluşturuldu.
- Güvenli kullanıcı giriş sistemi sağlandı.