

# ParkVision: Akıllı Otopark Asistanı

## Gereksinim Analiz Raporu

BMU401 Bilgisayar Mühendisliği Tasarım Dersi

Öğrenci No	Ad Soyad
215260008	Hasan Yılmaz Gürsoy
225260016	İsmail Şahin
225260058	Fuat Yavaş

# 1. GİRİŞ

## 1.1 Amaç

Bu doküman, ParkVision AKILLI Otopark Asistanı sisteminin gereksinimlerini detaylı olarak tanımlamaktadır. Sistem, kentsel alanlarda park yeri bulma sürecini kolaylaştırmayı, sürücülerin zaman kaybını azaltmayı ve trafik yoğunluğunu hafifletmeyi amaçlamaktadır.

## 1.2 Kapsam

ParkVision sistemi, görüntü işleme teknolojisi, bulut altyapısı ve mobil uygulama entegrasyonu ile çalışan hibrit bir park yönetim çözümüdür. Sistem şunları kapsar:

- Gerçek zamanlı park yeri doluluk tespiti
- Kullanıcıya özel mobil uygulama
- Park yeri rezervasyon sistemi
- Navigasyon entegrasyonu
- Kitle kaynaklı veri toplama

## 1.3 Tanımlar ve Kısaltmalar

Terim	Açıklama
YOLO	You Only Look Once - Nesne algılama algoritması
CNN	Convolutional Neural Network - Evrişimli Sinir Ağısı
API	Application Programming Interface - Uygulama Programlama Arayüzü
ROI	Region of Interest - İlgi Alanı
IoU	Intersection over Union - Kesişimin Birleşime Oranı

# 2. SİSTEM GENEL BAKIŞ

## 2.1 Sistem Mimarisi

ParkVision üç katmanlı bir mimari yapıya sahiptir:

- Veri Toplama Katmanı:** Kameralar, sensörler ve kullanıcı bildirimleri
- İşleme Katmanı:** Bulut tabanlı görüntü işleme ve veri analizi
- Sunum Katmanı:** Mobil uygulama ve web arayüzü

## 2.2 Kullanıcı Rollerleri

Kullanıcı Rolü	Sorumluluklar
Sürücü	<ul style="list-style-type: none"><li>Park yeri arama ve bulma</li><li>Rezervasyon yapma</li><li>Boş yer bildirimi</li></ul>
Otopark Yöneticisi	<ul style="list-style-type: none"><li>Otopark bilgilerini güncelleme</li><li>Kamera sistemlerini yönetme</li><li>Raporlama ve analiz</li></ul>

Kullanıcı Rolü	Sorumluluklar
Sistem Yöneticisi	<ul style="list-style-type: none"><li>Sistem bakım ve güncelleme</li><li>Güvenlik yönetimi</li><li>Performans izleme</li></ul>

## 3. FONKSİYONEL GEREKSİNİMLER

### 3.1 Görüntü İşleme Modülü

- FR1: Sistem, otopark kamera görüntülerini gerçek zamanlı olarak analiz edebilmelidir.
- FR2: YOLO algoritması kullanarak park alanlarındaki araçları tespit edebilmelidir.
- FR3: Her park yerinin dolu/boş durumunu yüksek doğrulukla belirleyebilmelidir.
- FR4: ROI (Region of Interest) belirleme özelliği ile sadece park alanlarını analiz edebilmelidir.

### 3.2 Mobil Uygulama

- FR5: Kullanıcılar sisteme kayıt olabilmeli ve giriş yapabilmelidir.
- FR6: Harita üzerinde yakındaki otoparkları ve doluluk oranlarını görüntüleyebilmelidir.
- FR7: Seçilen boş park yerine GPS navigasyon başlatabilmelidir.
- FR8: Park yeri rezervasyonu yapabilmeli ve iptal edebilmelidir.
- FR9: 'Aracımı Bul' özelliği ile park edilen konumu kaydedebilmelidir.
- FR10: Kullanıcılar boş park yeri bildirimi yapabilmelidir (Crowdsourcing).
- FR11: Push bildirimleri ile rezervasyon hatırlatmaları alabilmelidir.

### 3.3 Backend Sistemi

- FR12: Kamera görüntülerini bulut depolamaya yükleyebilmelidir.
- FR13: Serverless fonksiyonlar ile görüntü işleme tetikleyebilmelidir.
- FR14: RESTful API'ler aracılığıyla mobil uygulamaya veri sağlayabilmelidir.
- FR15: Veritabanında park yeri durumlarını güncelleystirmelidir.
- FR16: Rezervasyon işlemlerini yönetebilmelidir.
- FR17: Kullanıcı verilerini güvenli şekilde saklayabilmelidir.

### 3.4 Yönetim Paneli

- FR18: Otopark yöneticileri yeni otopark alanları tanımlayabilmelidir.
- FR19: Park yeri sayısı ve konumları güncellenebilmesidir.
- FR20: Ücretlendirme bilgileri tanımlanabilmesidir.
- FR21: Doluluk istatistikleri ve raporlar görüntülenebilmesidir.
- FR22: Kamera ayarları ve ROI tanımlamaları yapılabilmelidir.

## 4. FONKSİYONEL OLMAYAN GEREKSİNİMLER

### 4.1 Performans Gereksinimleri

- NFR1: Görüntü işleme modülü, park yerlerinin doluluk durumunu yüksek bir doğrulukla tespit etmelidir. Mobil uygulama API yanıt süresi minimum seviyede tutulmalıdır.
- NFR2: Mobil uygulamanın, API'den otopark doluluk verilerini çekme ve haritada gösterme süresi, iyi bir internet bağlantısında 3 saniyeyi aşmamalıdır. Park yeri durumu güncellemleri maksimum 5 saniye gecikmeyle yansıtılmalıdır.
- NFR3: Bir park yerinin durumundaki değişiklik (örn: bir aracın park etmesi), kamera tarafından algılandıktan sonra en fazla 60 saniye içinde mobil uygulamada güncellenmiş olmalıdır.

### 4.2 Güvenilirlik

- NFR4: Sistem yüksek kullanılabilirlik (uptime) sağlamalıdır.
- NFR5: Kamera arızası durumunda alternatif veri kaynakları kullanılabilmelidir.
- NFR6: Veri kaybını önlemek için otomatik yedekleme yapılmalıdır.
- NFR7: Sistem hataları loglanmalı ve izlenmelidir.

### 4.3 Güvenlik

- NFR8: Kullanıcı şifreleri veritabanında geri döndürülemez şekilde (örn: hash'lenerek) saklanmalıdır.
- NFR9: Mobil uygulama ve arka uç API arasındaki tüm veri iletişimini SSL/TLS (HTTPS) üzerinden şifrelenmelidir.

### 4.4 Kullanılabilirlik

- NFR10: Mobil uygulama Android platformunu desteklemelidir.
- NFR11: Arayüz Türkçe dilini desteklemelidir.
- NFR12: Kullanıcı arayüzü sezgisel ve kullanımı kolay olmalıdır.

### 4.5 Ölçeklenebilirlik

- NFR13: Arka uç mimarisi, artan kullanıcı sayısını ve sisteme eklenecek yeni otoparkları kaldırabilecek şekilde (örn: sunucusuz fonksiyonlar kullanılarak) ölçeklenebilir olmalıdır.
- NFR14: Mobil uygulama, Android işletim sistemlerinde sorunsuz çalışmalıdır.

## 5. KULLANICI HİKAYELERİ VE SENARYOLAR

### 5.1 Kullanıcı Hikayeleri

ID	Kullanıcı Hikayesi	Kabul Kriterleri
US-01	Bir sürücü olarak, yakınımdaki boş park yerlerini harita üzerinde görmek istiyorum.	<ul style="list-style-type: none"><li>Harita 5 km yarıçaptaki otoparkları göstermeli</li><li>Doluluk oranları renk kodlarıyla belirtilmeli</li></ul>
US-02	Bir sürücü olarak, park yerini önceden rezerve edebilmek istiyorum.	<ul style="list-style-type: none"><li>Rezervasyon maksimum 30 dakika geçerli olmalı</li><li>Rezervasyon onay bildirimini gönderilmeli</li></ul>
US-03	Bir otopark yöneticisi olarak, otoparkımın doluluk istatistiklerini görmek istiyorum.	<ul style="list-style-type: none"><li>Günlük, haftalık, aylık raporlar oluşturulmalı</li><li>Yoğun saatler analizi yapılmalı</li></ul>

### 5.2 Kullanım Senaryoları

#### Senaryo 1: Park Yeri Bulma ve Navigasyon

- Kullanıcı mobil uygulamayı açar.
- Konum izni verir ve harita ekranı açılır.
- Yakındaki otoparklar doluluk oranlarıyla birlikte görüntülenir.
- Kullanıcı uygun bir otopark seçer.
- Navigasyon başlatılır ve kullanıcı otoparka yönlendirilir.
- Otoparka varıldığında boş yerler güncellenir.

#### Senaryo 2: Park Yeri Rezervasyonu

- Kullanıcı hedefteki boş yeri seçer.
- 'Rezerve Et' butonuna tıklar.
- Sistem 30 dakikalık rezervasyon oluşturur.
- Kullanıcıya rezervasyon kodu ve süre bilgisi gönderilir.
- Kullanıcı belirtilen sürede otoparka ulaşır.
- QR kod veya rezervasyon numarası ile giriş yapar.

## 6. SİSTEM MİMARİSİ

### 6.1 Teknoloji Yığını

Katman	Teknolojiler
Görüntü İşleme	Python, OpenCV, YOLOv8/v11, TensorFlow/PyTorch
Backend	Node.js/Python FastAPI, PostgreSQL, Redis, Docker
Bulut Altyapısı	Digital Ocean / Huggingface Spaces
Mobil Uygulama	React Native/Flutter, Google Maps API/ Yandex Maps
Yönetim Paneli	React.js, Material-UI, Chart.js

### 6.2 Veri Akışı

- Veri Toplama:** Kameralar görüntülerini yakalar
- Ön İşleme:** Görüntüler normalize edilir ve ROI uygulanır
- Analiz:** YOLO modeli araçları tespit eder
- Post-Processing:** IoU hesaplanır ve park yeri durumu belirlenir
- Veri Güncelleme:** Veritabanı güncellenir
- Bildirim:** Kullanıcılara anlık durum bilgisi iletilir

## 7. KISITLAMALAR VE RİSKLER

### 7.1 Teknik Kısıtlamalar

- Kamera kalitesi ve açısı görüntü işleme doğruluğunu etkileyebilir
- Gece görüşü için ek aydınlatma gerekebilir
- İnternet bağlantı kalitesi sistem performansını etkileyebilir

### 7.2 Riskler ve Azaltma Stratejileri

Risk	Olasılık	Azaltma Stratejisi
Model doğruluğunun düşük olması	Orta	Geniş veri seti ile eğitim, sürekli model güncelleme
Kullanıcı adaptasyon zorluğu	Düşük	Kullanıcı dostu arayüz, eğitim videoları
Veri güvenliği ihlali	Orta	Şifreleme, güvenlik testleri

# ParkVision Akıllı Otopark Asistanı

# CRC (Class-Responsibility-Collaborator) Kartları

BMU401 Bilgisayar Mühendisliği Tasarım Dersi

## CRC Kartları Genel Bakış

CRC kartları, nesne yönelimli tasarımda sınıfları, sorumluluklarını ve işbirlikçilerini tanımlamak için kullanılan basit ama etkili bir tekniktir. ParkVision projesi için hazırlanan CRC kartları, sistemin ana bileşenlerini ve bunların birbirleriyle olan ilişkilerini göstermektedir.

## 1. Park Yeri (ParkingSpot) Sınıfı

### ParkingSpot

Sorumluluklar	İşbirlikçiler
<ul style="list-style-type: none"><li>□ Park yeri ID'sini tutma</li><li>□ Konum bilgilerini saklama (x, y koordinatları)</li><li>□ Dolu/boş durumunu yönetme</li><li>□ Rezervasyon durumunu kontrol etme</li></ul>	<ul style="list-style-type: none"><li>□ ParkingLot</li><li>□ Reservation</li><li>□ Vehicle</li><li>□ ImageProcessor</li></ul>

## 2. Otopark (ParkingLot) Sınıfı

### ParkingLot

Sorumluluklar	İşbirlikçiler
<ul style="list-style-type: none"><li>□ Otopark bilgilerini yönetme</li><li>□ Park yerlerini listeleme</li><li>□ Doluluk oranını hesaplama</li><li>□ Konum ve adres bilgilerini tutma</li><li>□ Ücretlendirme bilgilerini yönetme</li><li>□ Kamera sistemlerini yönetme</li></ul>	<ul style="list-style-type: none"><li>□ ParkingSpot</li><li>□ Camera</li><li>□ ParkingManager</li><li>□ MapService</li></ul>

## 3. Kullanıcı (User) Sınıfı

### User

Sorumluluklar	İşbirlikçiler
<ul style="list-style-type: none"><li>□ Kullanıcı bilgilerini tutma</li><li>□ Giriş/çıkış işlemlerini yönetme</li><li>□ Araç bilgilerini kaydetme</li><li>□ Rezervasyon geçmişini tutma</li></ul>	<ul style="list-style-type: none"><li>□ Vehicle</li><li>□ Reservation</li><li>□ NotificationService</li><li>□ AuthenticationService</li></ul>

<input type="checkbox"/> Bildirim tercihlerini yönetme	
<input type="checkbox"/> Favori otoparkları listeleme	

## 4. Araç (Vehicle) Sınıfı

### Vehicle

Sorumluluklar	İşbirlikçiler
<input type="checkbox"/> Plaka bilgisini tutma <input type="checkbox"/> Marka ve model bilgilerini saklama <input type="checkbox"/> Araç tipini belirleme <input type="checkbox"/> Park edildiği konumu kaydetme <input type="checkbox"/> Park süresi hesaplama	<input type="checkbox"/> User <input type="checkbox"/> ParkingSpot <input type="checkbox"/> ImageProcessor <input type="checkbox"/> ParkingHistory

## 5. Görüntü İşleyici (ImageProcessor) Sınıfı

### ImageProcessor

Sorumluluklar	İşbirlikçiler
<input type="checkbox"/> Kamera görüntülerini analiz etme <input type="checkbox"/> YOLO modelini çalıştırma <input type="checkbox"/> Araç tespiti yapma <input type="checkbox"/> ROI (Region of Interest) belirleme <input type="checkbox"/> IoU hesaplama <input type="checkbox"/> Park yeri durumunu güncelleme	<input type="checkbox"/> Camera <input type="checkbox"/> ParkingSpot <input type="checkbox"/> Vehicle <input type="checkbox"/> CloudStorage <input type="checkbox"/> MLModel

## 6. Rezervasyon (Reservation) Sınıfı

### Reservation

Sorumluluklar	İşbirlikçiler
<input type="checkbox"/> Rezervasyon oluşturma <input type="checkbox"/> Rezervasyon kodunu üretme <input type="checkbox"/> Süre kontrolü yapma <input type="checkbox"/> Rezervasyonu iptal etme <input type="checkbox"/> Rezervasyon durumunu güncelleme	<input type="checkbox"/> User <input type="checkbox"/> ParkingSpot <input type="checkbox"/> NotificationService <input type="checkbox"/> Timer

## 7. Mobil Uygulama (MobileApp) Sınıfı

### MobileApp

Sorumluluklar	İşbirlikçiler
<ul style="list-style-type: none"><li><input type="checkbox"/> Kullanıcı arayüzüni yönetme</li><li><input type="checkbox"/> Harita görüntüleme</li><li><input type="checkbox"/> GPS konumu alma</li><li><input type="checkbox"/> Push bildirimler gönderme</li><li><input type="checkbox"/> Navigasyon başlatma</li><li><input type="checkbox"/> Kullanıcı girdilerini işleme</li></ul>	<ul style="list-style-type: none"><li><input type="checkbox"/> User</li><li><input type="checkbox"/> APIService</li><li><input type="checkbox"/> MapService</li><li><input type="checkbox"/> LocationService</li><li><input type="checkbox"/> NotificationService</li></ul>

## 8. Bulut Servisi (CloudService) Sınıfı

### CloudService

Sorumluluklar	İşbirlikçiler
<ul style="list-style-type: none"><li><input type="checkbox"/> Görüntüleri depolama</li><li><input type="checkbox"/> Serverless fonksiyonları tetikleme</li><li><input type="checkbox"/> Otomatik ölçeklendirme</li><li><input type="checkbox"/> Veri yedekleme</li><li><input type="checkbox"/> API Gateway yönetimi</li><li><input type="checkbox"/> CDN hizmeti sağlama</li></ul>	<ul style="list-style-type: none"><li><input type="checkbox"/> ImageProcessor</li><li><input type="checkbox"/> Database</li><li><input type="checkbox"/> APIService</li><li><input type="checkbox"/> BackupService</li><li><input type="checkbox"/> MonitoringService</li></ul>

## CRC Kartları İlişki Özeti

### Ana İlişkiler

#### 1. Veri Akışı İlişkileri:

- Camera → ImageProcessor → ParkingSpot → Database
- ImageProcessor, YOLO modelini kullanarak görüntüleri analiz eder ve park yeri durumlarını günceller

#### 2. Kullanıcı Etkileşim İlişkileri:

- User → MobileApp → APIService → Backend Services
- Kullanıcı mobil uygulama üzerinden tüm işlemlerini gerçekleştirir

#### 3. Rezervasyon İlişkileri:

- User → Reservation → ParkingSpot → NotificationService
- Rezervasyon sistemi kullanıcı, park yeri ve bildirim servisleri arasında koordinasyon sağlar

#### 4. Otopark Yönetim İlişkileri:

- ParkingLot → ParkingSpot → ParkingManager
- Her otopark birden fazla park yerini yönetir ve merkezi yönetim sistemine bağlıdır

