

YAZILIM TASARIM DOKÜMANI

İÇİNDEKİLER

İÇİNDEKİLER.....	1
1 KAPSAM.....	2
1.1 Tanım.....	2
1.2 Sisteme Genel Bakış	2
1.3 Dokümana Genel Bakış.....	3
2 İLGİLİ DOKÜMANLAR	4
3 YKE ÇAPINDA TASARIM KARARLARI	5
3.1 Girdi ve Çıktı ile İlgili Tasarım Kararları	5
3.2 Girdilere ve Koşullara Göre YKE Davranışı ile İlgili Tasarım Kararları.....	5
3.3 Veri Tabanı / Veri Dosyası ile İlgili Tasarım Kararları.....	5
3.4 Emniyet, Güvenlik ve Gizlilik Gereksinimlerini Karşılama Yaklaşımı.....	5
3.5 Esneklik, Elde Edilebilirlik ve İdame Ettirilebilirlik Sağlama Yaklaşımı.....	6
4 YKE’NİN YAPISAL TASARIMI.....	6
4.1 YKE Bileşenleri	6
4.2 Genel Çalıştırma (execution) Kavramı	8
4.3 Arayüz Tasarımı.....	12
5 YKE DETAYLI PLANI	13
5.1 YKE-CAM (Kamera Yönetim Modülü).....	13
5.2 YKE-OBJ (Nesne Tespit Modülü)	14
5.3 YKE-DEP (Derinlik Tahmini Modülü)	15
5.4 YKE-TTS (Sesli Geri Bildirim Modülü).....	16
5.5 YKE-CTRL (Kontrol ve Koordinasyon Modülü).....	17
6 GEREKSİNİMLERİN İZLENEBİLİRLİĞİ.....	17
6.1 Yazılım Biriminden YKE Gereksinimlerine İzlenebilirlik.....	18
6.2 Yazılım Konfigürasyon Elemanından Yazılım Birimine İzlenebilirlik	18
7 NOTLAR	18
7.1 Kısaltmalar	19
7.2 Terimler ve Tanımlar.....	19
7.3 Geçmiş Bilgi.....	20
8 EKLER.....	20

1 KAPSAM

1.1 Tanım

Bu doküman, BM-314 Yazılım Mühendisliği dersi kapsamında geliştirilen "Görme Engelliler için Nesne Tespiti" mobil uygulamasının yazılım tasarım dokümanıdır (Software Design Description - SDD).

Sistem; görme engelli bireylerin çevresel farkındalığını artırmak amacıyla, mobil cihaz kamerası kullanarak çevredeki nesneleri tespit eden, mesafelerini tahmin eden ve bu bilgileri sesli geri bildirimle kullanıcıya aktaran bir yapay zekâ destekli mobil uygulamayı kapsamaktadır.

Projede iOS işletim sistemi hedeflenmiş olup geliştirme sürecinde Swift dili, CoreML, AVFoundation ve Text-to-Speech API'leri kullanılmıştır.

- **Proje Adı:** Görme Engelliler için Nesne Tespiti Uygulaması
- **Kısaltma:** GENET
- **Versiyon Numarası:** 1.0
- **Yayın Numarası:** SDD-001
- **Hedef Platform:** iOS 15.0 ve üzeri
- **Geliştiriciler:** Furkan Egecan Nizam, Dilay Ece Maral
- **Yayın Tarihi:** 14.05.2025

Bu doküman, sistemin genel yapısını, alt sistem tasarımlarını, veri akışlarını ve bileşenler arasındaki etkileşimleri detaylı şekilde açıklamayı amaçlamaktadır.

1.2 Sisteme Genel Bakış

Bu doküman, BM-314 Yazılım Mühendisliği dersi kapsamında geliştirilen "Görme Engelliler için Nesne Tespiti" mobil uygulamasına ait yazılım tasarım dokümanıdır.

Sistemin temel amacı, görme engelli bireylerin çevrelerindeki nesneleri tespit etmelerini, bu nesnelerin konum ve mesafe bilgilerini edinmelerini ve bu bilgilerin sesli geri bildirim yoluyla kendilerine iletilmesini sağlamaktır. Sistem, kullanıcıya çevresel farkındalık kazandırarak günlük yaşam aktivitelerinde bağımsız hareket etmelerine yardımcı olmayı hedeflemektedir.

Sistem, iOS tabanlı mobil cihazlarda çalışacak şekilde geliştirilmiştir ve tamamen offline çalışacak biçimde tasarlanmıştır. Çevresel görüntüler mobil cihaz kamerası aracılığıyla alınmakta; nesne tespiti ve derinlik tahmini işlemleri CoreML tabanlı yapay zekâ modelleri ile gerçekleştirilmektedir. Elde edilen bilgiler iOS Text-to-Speech API kullanılarak kullanıcıya sesli olarak aktarılmaktadır.

Sistem geliştirme süreci Scrum modeli çerçevesinde yönetilmiştir. Operasyon ve bakım süreçleri için uygulamanın App Store dağıtımı sonrası yazılım güncellemeleri ve sürüm iyileştirmeleri planlanmaktadır.

Şu anda sistem yalnızca bireysel kullanım için geliştirilmiş olup, ilerleyen aşamalarda çoklu dil desteği, harici veri entegrasyonu ve yeni model versiyonlarının desteklenmesi gibi genişletilebilirlik özellikleri planlanmaktadır.

İlgili taraflar:

- **Müşteri:** BM-314 Yazılım Mühendisliği Dersi Öğretmeni Prof.Dr.Hacer Karacan
- **Kullanıcı:** Görme engelli bireyler
- **Geliştirici:** Furkan Egecan Nizam, Dilay Ece Maral
- **Destek Birimi:** Proje geliştirici ekibi (bakım ve güncellemeler için)

Şu anki ve planlı işlevsel bileşenler:

- Nesne Tespiti Modülü
- Derinlik Tahmini Modülü
- Sesli Geri Bildirim Modülü
- Kamera Arayüzü
- Kullanıcı Arayüzü (başlat/durdur komutları)

İlişkili diğer dokümanlar:

- Software Project Management Plan (SPMP)
- Software Requirements Specification (SRS)
- Software Test Documentation (STD)

1.3 Dokümana Genel Bakış

Bu doküman, "Görme Engelliler için Nesne Tespiti" mobil uygulamasının yazılım tasarımını (Software Design Description - SDD) detaylı şekilde açıklamak amacıyla hazırlanmıştır. Doküman, sistemin genel mimarisini, alt bileşenlerin işlevlerini, bileşenler arasındaki veri akışlarını, kullanılan teknolojileri ve tasarım kararlarını kapsamlı bir biçimde sunmaktadır.

Dokümanın amacı, yazılım geliştirme sürecinde tüm ekip üyeleri, bakım personeli ve ilgili paydaşlar için ortak bir referans kaynağı oluşturmak; sistemin doğru, tutarlı ve güvenli bir şekilde geliştirilmesini sağlamaktır. Ayrıca, ileride yapılacak güncellemeler, iyileştirmeler veya hata düzeltmeleri için sistemin yapısının anlaşılmasını kolaylaştırmak hedeflenmiştir.

Bu dokümanda tanımlanan tasarım, görme engelli bireylerin güvenliğini sağlamak üzere özel güvenlik önlemleri göz önünde bulundurularak hazırlanmıştır. Özellikle:

- Tüm sistem işlemleri offline (çevrimdışı) olarak gerçekleştirilecek şekilde tasarlanmış, veri gizliliği korunmuştur.
- Yanlış pozitif veya yanlış negatif tespitlerin önlenmesi için minimum güven skorları belirlenmiş ve hata yönetimi mekanizmaları entegre edilmiştir.
- Kritik hata durumlarında kullanıcıyı uyararak güvenli mod mekanizmaları dokümana dahil edilmiştir.

Bu dokümanda ayrıca, sistem bileşenlerinin tanımı, işlevsel ayrımları, sistem içi veri akışı, dış arayüzler ve dahili arayüzler gibi konular ayrıntılı bir şekilde ele alınmış, sistemin güvenli ve erişilebilir şekilde çalışmasını desteklemek için tasarım kararları açık bir biçimde belirtilmiştir.

2 İLGİLİ DOKÜMANLAR

No	Doküman Başlığı	Yayın Numarası	Değişiklik Numarası	Tarih	Kaynak
1	Software Project Management Plan (SPMP)	SPMP-001	0	19.03.2025	Proje Geliştirici Ekibi
2	Software Requirements Specification (SRS)	SRS-001	0	16.04.2025	Proje Geliştirici Ekibi
3	Software Design Description (SDD)	SDD-001	0	30.04.2025	Proje Geliştirici Ekibi
4	Software Test Description (STD)	STD-001	0	Planlanıyor	Proje Geliştirici Ekibi

Açıklamalar:

- Dokümanların kaynakları proje geliştirici ekibi olan Furkan Egecan Nizam ve Dilay Ece Maral tarafından sağlanmaktadır.

3 YKE ÇAPINDA TASARIM KARARLARI

Bu bölümde, "Görme Engelliler için Nesne Tespiti" mobil uygulamasının kullanıcı bakış açısından sistem davranışı ile yazılım birimlerinin seçim ve tasarımını etkileyen temel kararlar açıklanmaktadır. Tasarım kararları, sistemin güvenli, hızlı, erişilebilir ve sürdürülebilir çalışmasını sağlamak amacıyla belirlenmiştir.

3.1 Girdi ve Çıktı ile İlgili Tasarım Kararları

- Sistem girdisi olarak mobil cihazın kamerasından elde edilen görüntüler kullanılmaktadır.
- Görüntüler AVFoundation API'si kullanılarak belirli aralıklarla (her 2 saniyede bir) alınmakta ve CoreML uyumlu formata dönüştürülmektedir.
- Çıktılar ise kullanıcıya sesli geri bildirim şeklinde sunulmaktadır. Text-to-Speech API aracılığıyla, nesnenin konumu, nesne adı ve uzaklık bilgisi birleştirilerek sesli bir mesaj oluşturulmaktadır.
- İlk versiyonda görsel çıktı kullanılmamakta, kullanıcıya sadece sesli bilgilendirme yapılmaktadır.

3.2 Girdilere ve Koşullara Göre YKE Davranışı ile İlgili Tasarım Kararları

- Kamera erişimi mevcut değilse sistem kullanıcıyı sesli olarak bilgilendirir ve güvenli moda geçer.
- Nesne tespit edilemediği durumlarda kullanıcıya "Nesne tespit edilemedi" mesajı verilir.
- Derinlik tahmini başarısız olursa, sistem "Mesafe tahmin edilemedi" şeklinde sesli uyarı verir.
- Yanlış pozitif ve yanlış negatif olasılıklarını azaltmak için %60'ın üzerinde güven skoruna sahip nesneler dikkate alınır.
- Kritik hata durumlarında (örneğin: model donması), sistem işlem yapmayı durdurur ve kullanıcıyı bilgilendirir.

3.3 Veri Tabanı / Veri Dosyası ile İlgili Tasarım Kararları

- Sistem kalıcı veri saklamamakta, yalnızca geçici RAM belleğinde işlem yapmaktadır.
- Yapay zekâ model dosyaları (.mlmodel formatında) uygulamaya gömülü şekilde yerleştirilmiştir.
- Kullanıcıdan alınan görüntü verisi cihazda hiçbir şekilde kaydedilmemekte, işlem tamamlandığında bellekten silinmektedir.

3.4 Emniyet, Güvenlik ve Gizlilik Gereksinimlerini Karşılama Yaklaşımı

- Tüm işlemler offline (çevrimdışı) gerçekleştirilmektedir; sistem dış dünya ile veri paylaşımı yapmaz.

- Kamera ve mikrofon erişimi sadece kullanıcı izin verdiğinde aktif olur.
- GDPR ve KVKK gibi veri koruma yönetmeliklerine uygunluk sağlanmıştır; kullanıcı verisi toplanmaz veya saklanmaz.
- Kritik hata anlarında kullanıcıya sesli uyarı verilerek sistem güvenli moda alınır.

3.5 Esneklik, Elde Edilebilirlik ve İdame Ettirilebilirlik Sağlama Yaklaşımı

- Yapay zekâ modelleri modüler şekilde entegre edilmiştir; ileride farklı modeller veya iyileştirilmiş versiyonlar kolaylıkla eklenebilir.
- Uygulama Swift dilinde, iOS platformuna özel geliştirilmiş olup tüm güncellemeler App Store üzerinden dağıtılabilecek şekilde yapılandırılmıştır.
- Kullanıcı arayüzü minimal ve sesli etkileşim odaklı tasarlandığından, erişilebilirlik ve kullanım kolaylığı önceliklendirilmiştir.
- Sistem tasarımı, düşük işlem gücü olan cihazlarda da çalışabilecek şekilde optimize edilmiştir (örneğin TensorFlow Lite dönüşümleri).

4 YKE’NİN YAPISAL TASARIMI

4.1 YKE Bileşenleri

a) Yazılım Birimleri ve Tanımlayıcıları

Uygulama aşağıdaki temel yazılım birimlerinden oluşmaktadır:

Yazılım Birimi	Tanımlayıcı Kodu	Açıklama
Kamera Yönetim Modülü	YKE-CAM	Mobil cihazın kamerasından görüntü alınmasını sağlar.
Nesne Tespit Modülü	YKE-OBJ	Görüntülerden nesne tespiti yapar. (CoreML modeli kullanır)
Derinlik Tahmini Modülü	YKE-DEP	Tespit edilen nesnenin mesafesini tahmin eder. (CoreML modeli kullanır)
Sesli Geri Bildirim Modülü	YKE-TTS	Kullanıcıya sesli bilgi iletimi yapar. (Text-to-Speech API)
Kontrol ve Koordinasyon Modülü	YKE-CTRL	Modüller arası veri akışını ve sistemin genel işleyişini koordine eder.

b) Yazılım Birimlerinin Amacı ve İlgili YKE Gereksinimleri

- **YKE-CAM (Kamera Yönetim Modülü):**

Bu modül, mobil cihazın kamerasına erişimi sağlar ve belirlenen aralıklarla (örneğin her 2 saniyede bir) görüntü yakalar. Alınan görüntüleri işlenmek üzere Nesne Tespit Modülü'ne aktarır. Kamera izni, görüntü çözünürlüğü ve çekim sıklığı bu modül tarafından yönetilir.

İlgili Gereksinimler: Kamera Arayüzü (1.3.3), Dahili Kamera → Nesne Tespiti Veri Akışı (1.4.1).

- **YKE-OBJ (Nesne Tespit Modülü):**

Bu modül, kamera tarafından alınan görüntüler üzerinde nesne tespiti işlemlerini gerçekleştirir. Önceden eğitilmiş bir CoreML modeli kullanılarak, görüntüde nesne olup olmadığı, nesnenin sınıfı (örneğin sandalye, masa gibi) ve konumu (sağ, sol, karşı gibi) belirlenir.

Sadece %60 güven skoru üzerindeki nesneler dikkate alınır.

İlgili Gereksinimler: Nesne Tespiti Fonksiyonelliği (1.2.1), Model Arayüzü (1.3.5).

- **YKE-DEP (Derinlik Tahmini Modülü):**

Bu modül, tespit edilen nesnelerin cihaz ile olan mesafesini tahmin eder. Depth Estimation modeli kullanılarak, nesnenin yaklaşık uzaklığı (örneğin 1.5 metre) hesaplanır. Hesaplanan mesafe, sesli bildirim hazırlanmak üzere diğer modüllere aktarılır.

İlgili Gereksinimler: Derinlik Tahmini Fonksiyonelliği (1.2.2), Model Arayüzü (1.3.5).

- **YKE-TTS (Sesli Geri Bildirim Modülü):**

Bu modül, tespit edilen nesne ve mesafe bilgilerini birleştirerek kullanıcıya sesli olarak sunar. Belirlenen mesaj formatı ("Sağ tarafta 2 metre uzaklıkta sandalye var." gibi) iOS Text-to-Speech API ile seslendirilmektedir.

İlgili Gereksinimler: Sesli Geri Bildirim Fonksiyonelliği (1.2.3), Sesli Çıktı Arayüzü (1.3.4).

- **YKE-CTRL (Kontrol ve Koordinasyon Modülü):**

Sistem içindeki tüm modüllerin senkronizasyonunu sağlar. Kamera, nesne tespiti, derinlik tahmini ve sesli bildirim adımlarının doğru sırada çalışmasını yönetir. Ayrıca hata durumlarını algılayarak güvenli mod mekanizmalarını devreye alır ve kullanıcıyı bilgilendirir.

İlgili Gereksinimler: Dahili Arayüzlerin Yönetimi (1.4), Emniyet Gereksinimleri (1.7).

c) Geliştirme Durumu / Türü

Yazılım Birimi	Geliştirme Durumu	Açıklamalar
YKE-CAM	Yeni Tasarım	Swift ve AVFoundation API ile geliştirilmiştir.
YKE-OBJ	Mevcut Modelin Uygulama İçi Kullanımı	Önceden eğitilmiş CoreML object detection modeli (v1.0) kullanılmaktadır.
YKE-DEP	Mevcut Modelin Uygulama İçi Kullanımı	Önceden eğitilmiş CoreML depth estimation modeli (v1.0) kullanılmaktadır.
YKE-TTS	Platform API Kullanımı	iOS Text-to-Speech API kullanılmaktadır.

YKE-CTRL	Yeni Tasarım	Swift dili kullanılarak tüm modülleri koordine eden yapı geliştirilmiştir.
----------	--------------	--

Mevcut Kullanılan Modeller:

- Object Detection Modeli: Versiyon 1.0, CoreML formatı, offline kullanım, dökümantasyonu proje klasöründe mevcuttur.
- Depth Estimation Modeli: Versiyon 1.0, CoreML formatı, offline kullanım.

d) Bilgisayar Donanım Kaynakları Kullanımı

Kaynak Türü	Planlanan Kullanım
İşlemci (CPU):	Object detection ve depth estimation işlemleri sırasında işlemci kullanımının %50–60 arası olması beklenmektedir.
Bellek (RAM):	Ortalama 300–400 MB bellek kullanımı öngörülmektedir.
Girdi/Çıktı:	Kamera ile her 2 saniyede bir görüntü alınacaktır; sesli çıktılar ise gerçek zamanlı sağlanacaktır.
Depolama:	Uygulama ve model dosyaları için yaklaşık 200 MB kalıcı depolama alanı gereklidir.
Haberleşme:	Uygulama offline çalışacak şekilde tasarlandığından ağ bağlantısına ihtiyaç duyulmamaktadır.

4.2 Genel Çalıştırma (execution) Kavramı

Bu bölüm, "Görme Engelliler için Nesne Tespiti" uygulamasındaki yazılım birimleri arasında çalıştırma sırasını ve birimler arası dinamik ilişkileri açıklamaktadır.

Sistem, gerçek zamanlı olarak veri akışı ve kontrol akışı üzerinden modüller arasında senkronize şekilde işlemektedir. Aşağıda çalışma kavramı detaylı şekilde açıklanmıştır:

Çalıştırma Akışı (Execution Flow)

1. Sistem Başlatılması:

- Kullanıcı uygulamayı başlattığında Kontrol Modülü (YKE-CTRL) aktif hale gelir.
- Sistem başlangıç kontrollerini (kamera izni, model yüklemesi gibi) yapar.

2. Kamera Verisi Toplama:

- Kamera Yönetim Modülü (YKE-CAM) çalıştırılır.
- Her 2 saniyede bir cihazın kamerasından yeni bir görüntü alınır.

3. Nesne Tespiti:

- Alınan görüntü, Nesne Tespit Modülüne (YKE-OBJ) iletilir.
- Model görüntü üzerinde nesne tespiti yapar.
- Eğer güven skoru %60'ın üzerinde bir veya birden fazla nesne tespit edilirse, konum ve nesne sınıf bilgileri çıkarılır.

4. Derinlik Tahmini:

- Nesne tespit edilmişse, aynı görüntü Derinlik Tahmini Modülüne (YKE-DEP) aktarılır.
- Derinlik modeli nesnenin cihaza olan mesafesini tahmin eder.

5. Sesli Geri Bildirim:

- Nesnenin adı, konumu ve mesafesi bilgileri Sesli Geri Bildirim Modülüne (YKE-TTS) iletilir.
- Bu bilgiler Text-to-Speech API kullanılarak cümle haline getirilir ve kullanıcıya sesli olarak aktarılır.

6. Süreklilik:

- Tüm bu adımlar 2 saniyede bir tekrarlanır ve sistem sürekli canlı çalışır.

Dinamik İlişkiler ve Durumlar

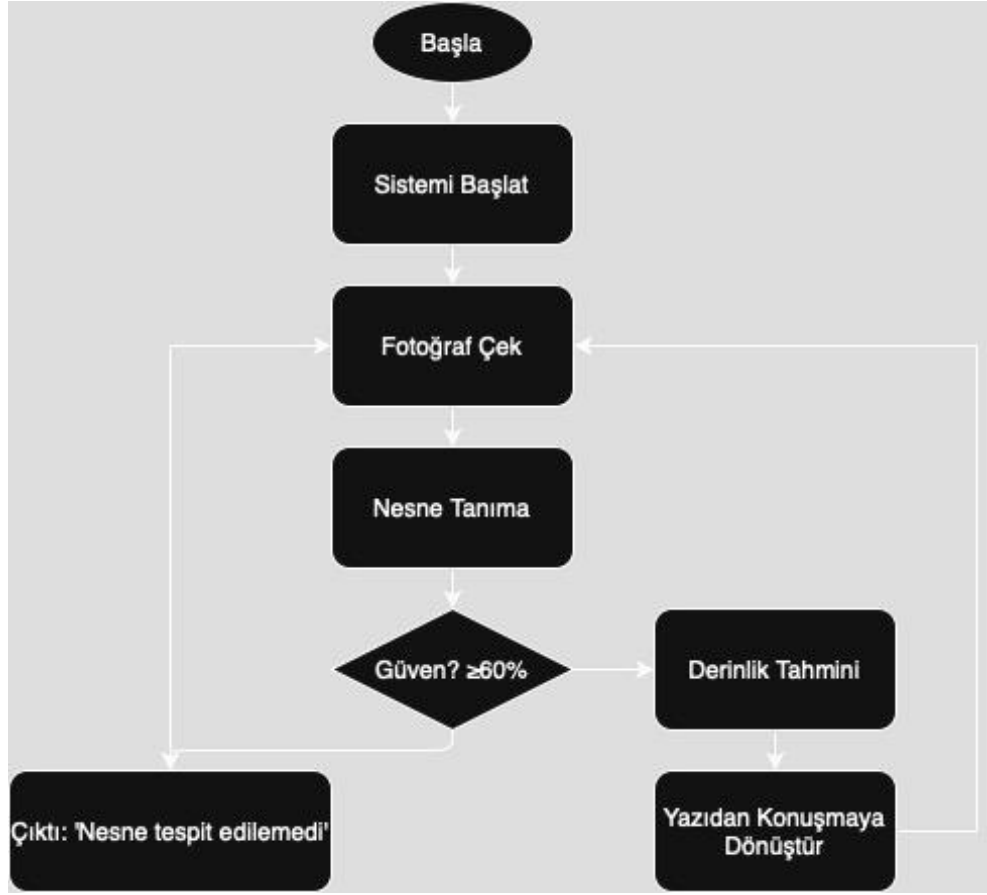
• Kontrol Akışı:

- YKE-CTRL modülü, tüm modülleri sırayla tetikler: önce kamera, sonra nesne tespiti, ardından derinlik tahmini ve en sonunda sesli bildirim.

- **Veri Akışı:**
 - Kamera → Nesne Tespiti → Derinlik Tahmini → Sesli Bildirim modülleri arasında veriler ardışık şekilde aktarılır.
- **Durum Geçişleri:**
 - Sistem 4 temel mod arasında çalışır:
 - **Hazır Mod:** Başlangıç kontrolleri yapılıyor.
 - **Aktif Mod:** Nesne tespiti ve sesli bildirim döngüsü aktif.
 - **Hata Modu:** Kamera erişimi yoksa veya model çalışmazsa kullanıcıya hata mesajı verilir.
 - **Kapanış Modu:** Uygulama kapatıldığında tüm modüller kapanır.
- **İstisnaların İdaresi:**
 - Kamera görüntüsü alınamazsa veya model tahmininde hata oluşursa, sistem güvenli moda geçer ve kullanıcı bilgilendirilir ("Kamera erişilemiyor", "Nesne tespit edilemedi" gibi).
- **Zamanlama İlişkileri:**
 - Her görüntü çekiminden sonra en fazla 4 saniye içinde (kamera → tespit → mesafe → sesli çıktı) tüm işlemler tamamlanmalıdır.
- **Eş Zamanlılık:**
 - Ana çalışma döngüsü sırayla işler; çoklu iş parçacığı (multi-threading) doğrudan kullanılmaz.
 - iOS'un sistem seviyesindeki asenkron işlemleri (örneğin kamera veri akışı ve TTS) yönetilmektedir.

Çalışma Diyagramı (Basitleştirilmiş Akış Diyagramı)

Şekil 1’de Görme Engelliler için Nesne Tespiti Uygulaması’nın çalışma diyagramı verilmiştir.



Şekil 1 - Görme Engelliler için Nesne Tespiti Uygulaması’nın Çalışma Diyagramı

Özel Hususlar

- **Dinamik Bellek Kullanımı:**
Görüntüler geçici olarak RAM üzerinde tutulur ve işlendikten sonra silinir.
- **Dinamik Nesne Yönetimi:**
Tespit edilen her nesne ayrı ayrı işlenir ve en fazla 5 nesneye kadar bilgi yönetilir.
- **Zamanlama ve Öncelik:**
 - Öncelik sıralaması: Kamera görüntü alımı > Nesne tespiti > Mesafe tahmini > Sesli geri bildirim.
 - Sesli bildirim süresi en fazla 3 saniye ile sınırlıdır.

- **Hata Yönetimi:**

Kritik hatalarda sistem işlemleri durdurur ve kullanıcıya güvenli hata mesajı ile sesli bildirim verir.

4.3 Arayüz Tasarımı

Bu bölümde, "Görme Engelliler için Nesne Tespiti" mobil uygulamasında yer alan arayüzlerin tasarımı tanımlanmıştır.

Her arayüz için projeye özel bir tanımlayıcı atanmış, arayüzün hangi sistem, modül veya kullanıcı ile etkileşim sağladığı belirtilmiştir. Ayrıca arayüzlerin mevcut API'leri kullanıp kullanmadığı veya proje kapsamında geliştirilip geliştirilmediği bilgisi verilmiştir.

Arayüz Listesi ve Tanımlamaları

Arayüz Tanımlayıcı	Arayüz İsmi	Elemanlar (İlgili Modüller)	Durum (Mevcut / Geliştirilen)	Versiyon	İlgili Doküman
INT-001	Kamera Arayüzü	YKE-CAM ↔ Kamera Donanımı	Mevcut (AVFoundation API kullanımı)	iOS 15+	SRS 1.3.3
INT-002	Kamera Arayüzü	YKE-CAM → YKE-OBJ	Geliştirildi	v1.0	SRS 1.4.1
INT-003	Derinlik Tahmini Arayüzü	YKE-OBJ → YKE-DEP	Geliştirildi	v1.0	SRS 1.4.2
INT-004	Sesli Geri Bildirim Arayüzü	YKE-DEP → YKE-TTS	Geliştirildi	v1.0	SRS 1.4.3
INT-005	Kullanıcı Arayüzü (Sesli Komutlar)	YKE-TTS ↔ Kullanıcı	Mevcut (iOS Text-to-Speech API ile desteklenmiş)	iOS 15+	SRS 1.3.2

Arayüzlerin Açıklamaları

- **INT-001 Kamera Arayüzü:**

Mobil cihazın kamerasından görüntü alımı yapılır. AVFoundation API kullanılarak gerçekleştirilmiştir. Hazır (mevcut) bir iOS API'sidir, geliştirilmemiştir.

- **INT-002 Nesne Tespiti Arayüzü:**

Kamera Modülü tarafından alınan görüntü, nesne tespiti modülüne aktarılır. Bu arayüz sistem

içinde geliştirilen bir veri aktarım ve formatlama işlemidir. (Yeni geliştirilmiştir.)

- **INT-003 Derinlik Tahmini Arayüzü:**
Nesne Tespit Modülü tarafından işaretlenen nesne bilgileri, Derinlik Tahmini Modülüne aktarılır. Bu iç arayüz, proje kapsamında özel olarak geliştirilmiştir.
- **INT-004 Sesli Geri Bildirim Arayüzü:**
Derinlik tahmin sonuçları, kullanıcıya sesli aktarım için Sesli Geri Bildirim Modülüne iletilir. Bu arayüz, uygulama içinde veri formatlama ve sesli çıktı hazırlama amaçlı geliştirilmiştir.
- **INT-005 Kullanıcı Arayüzü (Sesli Komutlar):**
Kullanıcıya sesli bilgi veren arayüzdür. iOS Text-to-Speech API kullanılarak yapılmıştır. Mevcut bir sistem API'si kullanılmıştır, geliştirilmeye gerek duyulmamıştır.

Özel Notlar:

- **Mevcut Arayüzler:** AVFoundation ve Text-to-Speech gibi iOS'un sunduğu sistem API'leridir. (Değiştirilmemiştir.)
- **Geliştirilen Arayüzler:** Modüller arası veri alışverişi (INT-002, INT-003, INT-004) proje kapsamında sıfırdan tasarlanmış ve uygulama içine entegre edilmiştir.
- **Versiyon Bilgisi:** Proje içinde geliştirilen arayüzler için ilk sürüm (v1.0) kabul edilmiştir.

5 YKE DETAYLI PLANI

Bu bölümde, "Görme Engelliler için Nesne Tespiti" mobil uygulamasında tanımlı her bir yazılım birimi için detaylı planlama yapılmıştır.

Her yazılım birimi için tasarım kararları, kullanılan teknolojiler, veri akışları, işlem mantığı ve özel durumlar açıklanmaktadır.

5.1 YKE-CAM (Kamera Yönetim Modülü)

a) Birim Tasarım Kararları

- AVFoundation kütüphanesi kullanılarak cihaz kamerasından 2 saniyede bir görüntü alınmasına karar verilmiştir.
- Görüntü formatı CVPixelBuffer olacak şekilde belirlenmiştir (CoreML uyumu için).

b) Kısıtlamalar / Sınırlandırmalar

- Kamera erişimi için kullanıcı izni zorunludur.
- Yetersiz ışık koşullarında görüntü kalitesi düşebilir.

c) Programlama Dili

- Swift programlama dili kullanılmıştır. AVFoundation kütüphanesi Swift için optimize edilmiştir.

d) İşlevsel Komutlar

- Kamera başlatma, görüntü alma ve kapanış fonksiyonları AVFoundation fonksiyonları ile sağlanır:
 - AVCaptureSession, AVCaptureDeviceInput, AVCaptureVideoDataOutput.

e) Veri Akışı

- Girdi: Kamera görüntüsü (RGB formatında).
- Çıktı: CVPixelBuffer veri yapısında nesne tespiti modülüne iletilir.

f) Mantık ve Dinamik Kontroller

- Kamera açıldığında sürekli döngüde çalışır.
- Belirlenen çekim aralığında yeni görüntü alınır ve işlenir.
- Kamera bağlantısında hata oluşursa kullanıcı sesli uyarılır.

5.2 YKE-OBJ (Nesne Tespit Modülü)

a) Birim Tasarım Kararları

- Object detection için önceden eğitilmiş CoreML modeli (mobil optimizasyonlu) kullanılmaktadır.
- %60 güven skoru eşiğinin altında kalan nesneler dikkate alınmaz.

b) Kısıtlamalar / Sınırlandırmalar

- Model giriş boyutu 640x480 çözünürlüğü desteklemektedir.
- Maksimum 5 nesne aynı anda işlenebilir.

c) Programlama Dili

- Swift dili kullanılmış, CoreML framework'ü entegre edilmiştir.

d) İşlevsel Komutlar

- Model yüklemesi ve inference işlemleri:
 - MLModel, VNCoreMLModel, VNCoreMLRequest kullanılarak yapılır.

e) Veri Akışı

- Girdi: CVPixelBuffer görüntü.
- Çıktı: Nesne sınıfı, konumu, güven skoru bilgileri.

f) Mantık ve Dinamik Kontroller

- Her yeni görüntüde model çalıştırılır.
- %60 güven skorunun üzerindeki sonuçlar alınır, diğerleri elenir.
- Tespit başarısız olursa hata mesajı oluşturulur.

5.3 YKE-DEP (Derinlik Tahmini Modülü)

a) Birim Tasarım Kararları

- Depth estimation işlemi için ayrı bir CoreML modeli kullanılmıştır.

b) Kısıtlamalar / Sınırlandırmalar

- Model yalnızca tespit edilen nesnelerin olduğu görüntü kesitlerinde çalışır.

c) Programlama Dili

- Swift dili + CoreML kullanımı.

d) İşlevsel Komutlar

- Depth prediction fonksiyonları:
 - MLModel yüklenmesi ve VNCoreMLRequest ile derinlik tahmini yapılması.

e) Veri Akışı

- Girdi: Tespit edilen nesnenin kırpılmış görüntüsü.
- Çıktı: Mesafe tahmini (metre cinsinden).

f) Mantık ve Dinamik Kontroller

- En yakın nesneye öncelik verilir.
- Derinlik tahmini başarısız olursa sistem hata mesajı üretir.

5.4 YKE-TTS (Sesli Geri Bildirim Modülü)

a) Birim Tasarım Kararları

- Text-to-Speech API kullanılarak doğal sesli çıktılar üretilir.

b) Kısıtlamalar / Sınırlandırmalar

- Maksimum mesaj uzunluğu 3 saniyeyi geçmemelidir.
- Sesli yanıtlar kesintisiz olmalıdır.

c) Programlama Dili

- Swift dili + AVSpeechSynthesizer API.

d) İşlevsel Komutlar

- Sesli mesaj oluşturma ve oynatma komutları:
 - AVSpeechUtterance, AVSpeechSynthesizer.

e) Veri Akışı

- Girdi: Nesne adı, konumu ve mesafe bilgisi (string).
- Çıktı: Sesli mesaj.

f) Mantık ve Dinamik Kontroller

- Yeni bir sesli mesaj başlamadan önce mevcut mesaj tamamlanır.

- Hata durumunda varsayılan sesli hata mesajı oynatılır.

5.5 YKE-CTRL (Kontrol ve Koordinasyon Modülü)

a) Birim Tasarım Kararları

- Tüm modüller arasındaki veri akışı ve sıra kontrolü bu modül tarafından sağlanır.

b) Kısıtlamalar / Sınırlandırmalar

- Modüller arası gecikmeler en fazla 1 saniye olmalıdır.
- Herhangi bir modülde hata olması durumunda sistem güvenli moda geçer.

c) Programlama Dili

- Swift dili.

d) İşlevsel Komutlar

- Modül tetiklemeleri ve hata yönetimi:
 - DispatchQueue, Error Handling yapıları.

e) Veri Akışı

- Kamera → Nesne Tespiti → Derinlik → Sesli Bildirim sırası korunur.

f) Mantık ve Dinamik Kontroller

- Sistemin her döngüsü tamamlandığında yeni döngü otomatik başlatılır.
- Hata algılandığında sistemin güvenli kapanması sağlanır.

6 GEREKSİNİMLERİN İZLENEBİLİRLİĞİ

Bu bölümde, "Görme Engelliler için Nesne Tespiti" uygulamasında Yazılım Tasarım Dokümanı (YTD) kapsamında tanımlanan her yazılım biriminin, ona tahsis edilen YKE gereksinimleriyle olan izlenebilirliği sağlanmıştır.

Ayrıca, Yazılım Konfigürasyon Elemanlarından (YKE bileşenleri) ilgili yazılım birimlerine izlenebilirlik de tablolar halinde sunulmuştur.

6.1 Yazılım Biriminden YKE Gereksinimlerine İzlenebilirlik

Yazılım Birimi	Tahsis Edilen YKE Gereksinimleri
YKE-CAM (Kamera Yönetim Modülü)	SRS 1.3.3 Kamera Arayüzü, 1.4.1 Dahili Kamera → Nesne Tespiti Arayüzü
YKE-OBJ (Nesne Tespit Modülü)	SRS 1.2.1 Nesne Tespiti Fonksiyonelliği, 1.3.5 Model Arayüzü
YKE-DEP (Derinlik Tahmini Modülü)	SRS 1.2.2 Derinlik Tahmini Fonksiyonelliği, 1.3.5 Model Arayüzü
YKE-TTS (Sesli Geri Bildirim Modülü)	SRS 1.2.3 Sesli Geri Bildirim Fonksiyonelliği, 1.3.4 Sesli Çıktı Arayüzü
YKE-CTRL (Kontrol ve Koordinasyon Modülü)	SRS 1.4 Dahili Arayüzler, 1.7 Emniyet Gereksinimleri

6.2 Yazılım Konfigürasyon Elemanından Yazılım Birimine İzlenebilirlik

Yazılım Konfigürasyon Elemanı	Tahsis Edildiği Yazılım Birimi
Kamera Arayüzü (AVFoundation API)	YKE-CAM
Object Detection CoreML Modeli	YKE-OBJ
Depth Estimation CoreML Modeli	YKE-DEP
Text-to-Speech API (iOS)	YKE-TTS
Modüller Arası Kontrol Akışı	YKE-CTRL

7 NOTLAR

Bu bölüm, "Görme Engelliler için Nesne Tespiti" mobil uygulamasına ait Yazılım Tasarım Dokümanının (YTD) daha iyi anlaşılmasını sağlamak amacıyla çeşitli genel bilgiler, kullanılan terimler, kısaltmalar ve açıklamaları içermektedir.

7.1 Kısaltmalar

Kısaltma	Açıklama
YKE	Yazılım Konfigürasyon Elemanı
YTD	Yazılım Tasarım Dokümanı
SRS	Software Requirements Specification (Yazılım Gereksinimleri Dokümanı)
SDD	Software Design Description (Yazılım Tasarım Dokümanı)
SPMP	Software Project Management Plan (Yazılım Proje Yönetim Planı)
STD	Software Test Documentation (Yazılım Test Dokümanı)
API	Application Programming Interface (Uygulama Programlama Arayüzü)
TTS	Text-to-Speech (Metinden Konuşmaya Dönüşüm)
AVFoundation	iOS Kamera ve Ses Çerçevesi (Apple Framework'ü)
CoreML	Apple Makine Öğrenimi Modeli Çalıştırma Framework'ü

7.2 Terimler ve Tanımlar

Terim	Tanım
Görme Engelliler için Nesne Tespiti	Mobil cihaz kamerası kullanılarak çevredeki nesnelerin tespit edilmesi, mesafelerinin tahmini ve kullanıcıya sesli geri bildirim verilmesi süreci.
Yazılım Konfigürasyon Elemanı (YKE)	Sistemin yapısal bir parçası olan, bağımsız geliştirilip yönetilebilen yazılım birimi.
CoreML Modeli	Mobil cihazlarda makine öğrenimi çıkarımı (inference) yapabilen optimize edilmiş yapay zeka modeli dosyası (.mlmodel uzantılı).
Text-to-Speech (TTS)	Yazılı metni doğal insan sesi formatında sesli çıktıya çeviren teknoloji.

AVFoundation	Apple tarafından sağlanan, görüntü ve ses işleme işlemleri için kullanılan yerleşik iOS API kütüphanesi.
Güven Skoru	Bir nesne tespiti sonucunda, modelin o nesnenin doğru olduğunu düşündüğü olasılık değeri. (%60 gibi).
Güvenli Mod	Sistem çalışmasında hata tespit edildiğinde kullanıcıya güvenli bildirim verip sistemi riskli işlem yapmadan durduran özel çalışma modu.
Inference	Makine öğrenimi modelinin yeni bir veri girdisi üzerinde tahmin yapma süreci.

7.3 Geçmiş Bilgi

- Bu proje, **BM-314 Yazılım Mühendisliği** dersi kapsamında gerçekleştirilmiştir.
- Uygulama sadece **iOS 15.0 ve üzeri** sürümlerde çalışacak şekilde tasarlanmıştır.
- Geliştirme sürecinde **Scrum** yöntemi benimsenmiş, iteratif geliştirme ve sürekli test prensipleri uygulanmıştır.
- Tüm sistem tasarımı, **offline çalışma, gizlilik, erişilebilirlik ve emniyet** odaklı geliştirilmiştir.

8 EKLER

Bu doküman için ek içerik bulunmamaktadır.