Proje Raporu: Görme Engelliler İçin Sesli Navigasyon Uygulaması

# Giriş

Bu proje, görme engellilerin bağımsız olarak yönlerini bulmalarına yardımcı olmak amacıyla geliştirilmiştir. Uygulama, kullanıcının sesli komutlarını alarak belirttiği lokasyona Google Haritalar API'sini kullanarak yönlendirme sağlar. Bu süreçte, sesli yönlendirme ile kullanıcıya adım adım talimatlar verilir.

# Pseudo Kod

// Uygulama başlatılır

main(): // Ana giriş noktası. Uygulama başlatılır.

runApp(MyApp()) // MyApp sınıfı çalıştırılır.

class MyApp extends StatelessWidget: // Stateless widget tanımlanır.

build(context): // Kullanıcı arayüzü oluşturulur.

return MaterialApp(theme, debugShowCheckedModeBanner, home) // Tema, debug banner ve ana sayfa ayarları yapılır.

class HomePage extends StatefulWidget: // Stateful widget tanımlanır.

createState(): // State oluşturulur.

return \_HomePageState() // HomePage için state döndürülür.

class \_HomePageState extends State<HomePage>: // HomePage için state yönetimi.

initState(): // Widget başlatıldığında çağrılır.

// Mevcut konumu al

getCurrentLocation() // Kullanıcının mevcut konumunu alır.

getCurrentLocation(): // Kullanıcının mevcut konumunu alır.

position = Geolocator.getCurrentPosition() // Geçerli konum alınır.

set currentPosition to position // currentPosition değişkeni konum ile güncellenir.

update map with current position marker // Harita mevcut konum işaretçisi ile güncellenir.

onMapCreated(controller): // Harita oluşturulduğunda çağrılır.

set mapController to controller // mapController değişkeni kontrol cihazı ile güncellenir.

getPolyPoints(destination): // İki nokta arasındaki polilini alır.

if currentPosition is not null: // Mevcut konum null değilse.

polylinePoints = PolylinePoints.getRouteBetweenCoordinates(currentPosition, destination) // Polilineleri alır.

update map with polyline // Harita poliline ile güncellenir.

getDirections(start, end): // İki nokta arasındaki yönleri alır.

response = http.get(apiUrl) // API'dan yanıt alınır.

data = json.decode(response.body) // JSON yanıtı çözümlenir.

return parseDirections(data) // Yönler ayrıştırılır ve döndürülür.

parseDirections(data): // Yönleri ayrıştırır.

if routes exist: // Eğer rotalar varsa.

instructions = [] // Talimatlar listesi oluşturulur.

walkingDirections = getWalkingDirections(data) // Yürüyüş talimatları alınır.

instructions.addAll(walkingDirections) // Yürüyüş talimatları eklenir.

busInfo = getBusInfo(data) // Otobüs bilgisi alınır.

instructions.add(busInfo) // Otobüs bilgisi eklenir.

return instructions // Talimatlar döndürülür.

else:

return "" // Rota yoksa boş döndürülür.

getWalkingDirections(data): // Yürüyüş yönlerini alır.

steps = data['routes'][0]['legs'][0]['steps'] // Adımlar alınır.

for step in steps: // Her adım için.

if step['travel\_mode'] == 'WALKING': // Eğer yürüyüş adımıysa.

add formatted walking instruction to list // Formatlanmış yürüyüş talimatı listeye eklenir.

return list // Yürüyüş talimatları listesi döndürülür.

getBusInfo(data): // Otobüs bilgilerini alır.

steps = data['routes'][0]['legs'][0]['steps'] // Adımlar alınır.

for step in steps: // Her adım için.

if step['travel\_mode'] == 'TRANSIT': // Eğer toplu taşıma adımıysa.

return formatted bus info // Formatlanmış otobüs bilgisi döndürülür.

getTextToAdress(adress): // Adresi metne çevirir.

locations = locationFromAddress(adress) // Adresten konumlar alınır.

directions = getDirections(currentPosition, locations[0]) // Yönler alınır.

speak(directions) // Yönler sesli olarak okunur.

getVoice(): // Kullanıcı sesli komut verir.

if speech not listening: // Eğer konuşma dinlemiyorsa.

speech.initialize() // Konuşma tanıma başlatılır.

speech.listen(onResult) // Sonuçları dinler.

else:

speech.stop() // Konuşma durdurulur.

after delay:

getTextToAdress(spokenText) // Gecikmeden sonra metni adrese çevirir ve yönleri alır.

# Blok Diyagramı

Blok Diyagramı şu şekildedir:  
1. Kullanıcı Sesli Komut Verir  
2. Ses Tanıma Modülü - Kullanıcının konuşmasını metne dönüştürür.  
3. Adres Çözümleyici Modül - Adresi çözümler ve koordinatları alır.  
4. Google Maps API - Güzergah bilgilerini alır.  
5. Yönlendirme Modülü - Yönlendirmeleri kullanıcının sesli olarak duymasını sağlar.

# Durum ve Akış Diyagramları

- Durum Diyagramı:  
 1. Bekleme: Kullanıcıdan sesli komut beklenir.  
 2. Ses Tanıma: Kullanıcının konuşması tanınır.  
 3. Adres Çözümleme: Verilen adres çözülür ve koordinatları alınır.  
 4. Yönlendirme: Kullanıcıya güzergah bilgileri sesli olarak iletilir.  
  
- Akış Diyagramı:  
 1. Kullanıcı uygulamayı açar.  
 2. Kullanıcı sesli komut verir.  
 3. Komut ses tanıma modülüne gönderilir.  
 4. Adres çözümleme modülü çalışır.  
 5. Google Maps API'ye güzergah bilgisi talebi yapılır.  
 6. Yönlendirme bilgisi kullanıcıya sesli olarak iletilir.

# Tasarlanan Sayfalar

- Ana Sayfa: Kullanıcının sesli komut vermesi için mikrofon butonu bulunur.  
- Yönlendirme Sayfası: Kullanıcının gitmek istediği yerin bilgilerini ve yönlendirmeleri gösterir.

# Literatür Taraması

Görme engelliler için geliştirilen çeşitli navigasyon uygulamaları bulunmaktadır. Örneğin, BlindSquare ve Seeing Eye GPS gibi uygulamalar kullanıcıya sesli yönlendirme sunmaktadır. Bu projeler genellikle GPS ve haritalama servisleri kullanarak kullanıcıya yön bulmada yardımcı olmaktadır. Bizim uygulamamız da benzer bir amaç gütmekle birlikte, daha kolay erişilebilir bir arayüz ve sesli komut ile adres çözümleme özellikleriyle öne çıkmaktadır.

# Yazılımsal Mimariler, Yöntemler ve Teknikler

- Flutter: Uygulamanın kullanıcı arayüzünü geliştirmek için kullanıldı.  
- Google Maps API: Güzergah bilgilerini almak için kullanıldı.  
- Geolocator: Kullanıcının mevcut konumunu belirlemek için kullanıldı.  
- Speech to Text: Kullanıcının sesli komutlarını metne dönüştürmek için kullanıldı.  
- Flutter TTS: Yönlendirme bilgilerini sesli olarak iletmek için kullanıldı.

# Karşılaşılan Zorluklar ve Çözümleri

- Konum Doğruluğu: GPS konumlarının doğruluğunu artırmak için yüksek doğruluk modu kullanıldı.  
- Ses Tanıma Hassasiyeti: Ses tanıma doğruluğunu artırmak için kullanıcıya net ve anlaşılır konuşma yapması önerildi.  
- API Kısıtlamaları: Google Maps API kullanımı sırasında günlük limitler ve kotalar göz önünde bulunduruldu ve gerekirse premium hizmetler kullanıldı.

# Projenin Katkıları

- Takım Çalışması: Proje ekibi arasında etkili iletişim ve işbirliği sağlandı.  
- Teknik Bilgi: GPS, haritalama ve ses teknolojileri konusunda derinlemesine bilgi sahibi olundu.  
- Kullanıcı Deneyimi: Görme engelli kullanıcıların ihtiyaçlarını anlama ve onlara yönelik çözümler üretme becerisi geliştirildi.