

Voice Map Guide

Bilişim Sistemleri Mühendisliği

Yunus Emre Gölbaş, Yunus Emre Gül,

Serhat Akçadağ

211307043@kocaeli.edu.tr

211307009@kocaeli.edu.tr

211307032@kocaeli.edu.tr

Abstract— Android Studio ve Kotlin kullanarak Google Maps API entegrasyonu ile geliştirilen bir Sesli Harita Rehberi uygulamasını içermektedir. Uygulamanın temel amacı, kullanıcılara interaktif ve erişilebilir bir navigasyon aracı sunarak, gerçek zamanlı sesli yönlendirme ve harita işlevsellikleri sağlamaktır.

Keywords— Sesli Harita Rehberi, Android Studio, Kotlin, Google Maps API, Görme Engellilik

GİRİŞ

Google Maps API'sini kullanarak doğru ve güncel harita verileri sunar. Başlıca özellikler arasında hedef belirlemek için sesli komutlar, adım adım navigasyon talimatları bulunmaktadır. Bu uygulama, özellikle görme engelli kullanıcılar veya eller serbest çalışmayı tercih edenler için navigasyon deneyimini geliştirmeyi hedeflemektedir.

UYGULAMANIN AMACI

Uygulamanın temel amacı, kullanıcıların sesli komutlarla hedeflerini belirlemelerine ve gerçek zamanlı sesli talimatlarla hedeflerine ulaşmalarına yardımcı olmaktır. Bu, görme engelli kullanıcıların yol güvenli bir şekilde yol bulmalarını sağlar.

A. Kullanılan Teknolojiler ve Araçlar

- Android Studio: Geliştirme ortamı
- Kotlin: Programlama dili
- Google Maps API: Harita ve yönlendirme hizmetleri
- OkHttp: HTTP isteklerini yönetmek için
- TextToSpeech (TTS): Sesli yönlendirme için
- SpeechRecognizer: Sesli komutları algılamak için

B. Temel Özellikler

- Sesli Komutlar: Kullanıcılar, sesli komutlar ile hedeflerini belirleyebilir.
- Gerçek Zamanlı Navigasyon: Harita üzerinde adım adım yönlendirme sağlar.
- Uyarlanabilir Rotalama: Gerçek zamanlı trafik koşullarına göre en uygun rotayı belirler.
- Erişilebilirlik: Görme engelli kullanıcılar için sesli yönlendirme sunar.
- Eller Serbest Kullanım: Kullanıcıların ellerini kullanmadan navigasyon yapmalarını sağlar.

C. Uygulama Akışı ve Kontrol Yapısı

C.1 İzin Kontrolü ve Ayarlar

Uygulama, sesli komutları algılayabilmek için kullanıcılardan RECORD_AUDIO izni talep eder:

```
if (ContextCompat.checkSelfPermission(context, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(
        activity, this,
        arrayOf(Manifest.permission.RECORD_AUDIO),
        requestCode: 9
    )
}
```

C.2 Sesli Komutların Algılanması Sesli komutlar SpeechRecognizer kullanılarak algılanır:

```
SpeechRecognizer = SpeechRecognizer.createSpeechRecognizer(context, this)
intent = Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, value: "tr-TR")

speechRecognizer.setRecognitionListener(object : RecognitionListener {
    override fun onResults(results: Bundle) {
        val data = results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION)
        speechDestination = data?.get(0)
        Toast.makeText(context, this@MainActivity, text: "Speech input: $speechDestination"
        startRouteCalculation()
    }

    override fun onPartialResults(partialResults: Bundle?) {
        println("onpartialresults")
    }

    override fun onEvent(eventType: Int, params: Bundle?) {
```

C.3 Text-to-Speech (TTS) Ayarları Uygulama, sesli talimatları sağlamak için TextToSpeech kullanır:

```
override fun onInit(status: Int) {
    if (status == TextToSpeech.SUCCESS) {
        tts.language = Locale.US
        tts.setSpeechRate(1.0f)
    }
}

fun speak(text: String) {
    if (!tts.isInitialized) {
        tts.language = Locale.forLanguageTag(languageTag: "tr")
        tts.setSpeechRate(1.0f)
        tts.speak(text, TextToSpeech.QUEUE_ADD, params: null, utteranceId: "hello_world_
    )
}
```

C.4 Harita ve Konum Servisleri Google Maps API ve konum servisleri kullanılarak harita ve konum bilgileri yönetilir:

```

val mapFragment = supportFragmentManager.findFragmentById(R.id.
mapFragment.getMapAsync { googleMap ->
    googleMap.mapType = GoogleMap.MAP_TYPE_NORMAL

    if (ActivityCompat.checkSelfPermission(
        context: this,
        android.Manifest.permission.ACCESS_FINE_LOCATION
    ) == PackageManager.PERMISSION_GRANTED
    ) {
        val locationManager = getSystemService(Context.LOCATION_SERVICE) as Loca
        val location = locationManager.getLastKnownLocation(LocationManager.GPS_

        if (location != null) {
            val latLng = LatLng(location.latitude, location.longitude)
            googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, z
        } else {
            ActivityCompat.requestPermissions(
                activity: this,
                arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION),
                requestCode: 1
            )
        }

        if (ActivityCompat.checkSelfPermission(
            context: this,
            android.Manifest.permission.ACCESS_FINE_LOCATION
        ) == PackageManager.PERMISSION_GRANTED
        ) {
            googleMap.isMyLocationEnabled = true

```

C.5 Rota Hesaplama ve Çizim Bu bölüm, uygulamanın rota hesaplama ve rota çizim işlemlerini detaylandırır.

C.5.1 Rota Hesaplama İşlemi Rota hesaplama işlemi, kullanıcının belirlediği hedefe ulaşmak için Google Directions API'yi kullanır. Bu işlem, kullanıcının mevcut konumunu ve hedefini alarak en uygun rotayı hesaplar.

C.5.1.1 Mevcut Konumun Alınması Uygulama, kullanıcının mevcut konumunu almak için FusedLocationProviderClient kullanır:

```

val fusedLocationClient = LocationServices.getFusedLocationProviderClient(activity: this@
if (ActivityCompat.checkSelfPermission(
    context: this@MainActivity,
    Manifest.permission.ACCESS_FINE_LOCATION
) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
    context: this@MainActivity,
    Manifest.permission.ACCESS_COARSE_LOCATION
) != PackageManager.PERMISSION_GRANTED
) {
    Log.e(tag: "Location", msg: "Permission not granted")
    return
}

```

C.5.1.2 Geocoding ile Hedef Konumun Alınması Kullanıcının belirlediği hedef adres, Google Geocoding API

kullanılarak enlem ve boylam bilgisine dönüştürülür:

```

val destinationText = replaceTurkishChars(speechDestination.toString()).rep
val geocodingUrl = "https://maps.googleapis.com/maps/api/geocode/json?addre
val geocodingClient = OkHttpClient()
val geocodingRequest = Request.Builder().url(geocodingUrl).build()

geocodingClient.newCall(geocodingRequest).execute().use { geocodingResponse
    if (!geocodingResponse.isSuccessful) {
        println("Tekeret")
        return@use
    }

    val geocodingJsonResponse = JSONObject(geocodingResponse.body!!.string(
    val results = geocodingJsonResponse.getJSONArray("results")

    if (results.length() > 0) {
        val firstResult = results.getJSONObject(index: 0)
        val geometry = firstResult.getJSONObject("geometry")
        val location = geometry.getJSONObject("location")
        val destinationLatitude = location.getDouble("lat")
        val destinationLongitude = location.getDouble("lng")
        val destination = "$destinationLatitude,$destinationLongitude"

        if (destination == currentLocationString) {
            Toast.makeText(context: this@MainActivity, text: "Zaten hedef kon
            return@use
        }
    }
}

```

C.5.1.3 Rota Talebi ve Yönlendirme Geocoding işleminden sonra, mevcut konum ve hedef konum kullanılarak Google Directions API'den rota bilgisi alınır:

```

if (destination != ",") {
    val url = "https://maps.googleapis.com/maps/api/directions/json?" +
        "origin=$currentLocationString" +
        "&destination=$destination" +
        "&mode=$mode" +
        "&key=$apiKey"

    val client = OkHttpClient()
    val request = Request.Builder()
        .url(url)
        .build()
    client.newCall(request).execute().use { response ->
        if (!response.isSuccessful) throw IOException("Failed to get directions: $respo
        val jsonResponse = JSONObject(response.body!!.string())
        val routes = jsonResponse.getJSONArray("routes")
        if (routes.length() > 0) {
            val firstRoute = routes.getJSONObject(index: 0)
            val legs = firstRoute.getJSONArray("legs")
            val firstLeg = legs.getJSONObject(index: 0)
            val steps = firstLeg.getJSONArray("steps")
            if (legs.length() == 1 && steps.length() == 1) {
                val lastStep = steps.getJSONObject(index: 0)
                val lastDistance = lastStep.getJSONObject("distance").getInt("name
                if (lastDistance <= 5) {
                    speak(text: "Hedefe vardınız")
                    stopRouteCalculation()
                    return@use
                }
            }
        }
    }
}

```

C.5.2 Rota Çizim İşlemi Rota çizim işlemi, Google Maps üzerinde kullanıcıya görsel bir yönlendirme sağlamak için yapılır. Bu işlem, Google Directions API'den alınan polilin (polyline) verisini harita üzerine çizer.

C.5.2.1 Polilin Dekodlama ve Çizim Polilin verisi, PolylineEncoding kullanılarak decode edilir ve harita üzerine çizilir:

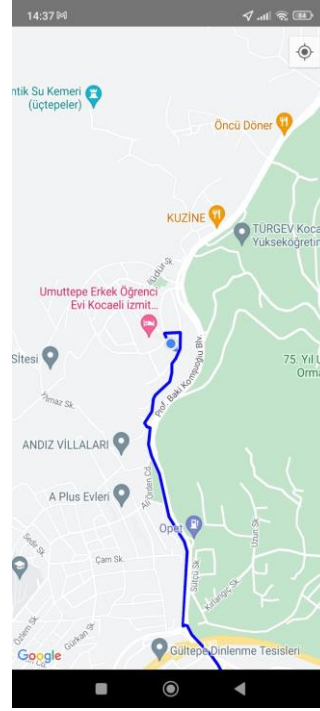
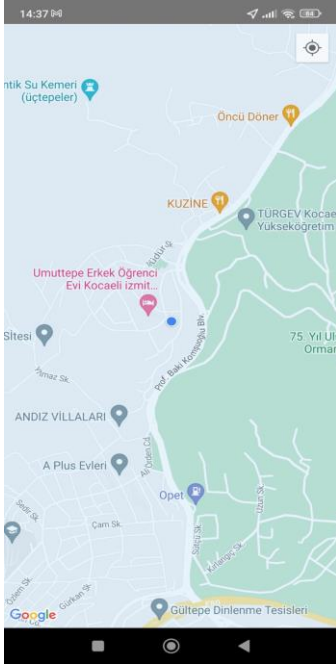
```

yunusemregul0
fun drawRoute(polyline: String, destinationLat: Double, destinationLng: Double) {
    val mapFragment = supportFragmentManager.findFragmentById(R.id.map) as SupportMapFragment
    mapFragment?.getMapAsync { googleMap ->
        googleMap.clear()
        val decodedPath = PolylineEncoding.decode(polyline)
        val options = PolylineOptions()
        for (point in decodedPath) {
            options.add(LatLng(point.lat, point.lng))
        }
        options.width(10f)
        options.color(Color.Blue.hashCode())
        googleMap.addMarker(MarkerOptions().position(LatLng(destinationLat, destinationLng)))
        googleMap.addPolyline(options)
    }
}

```

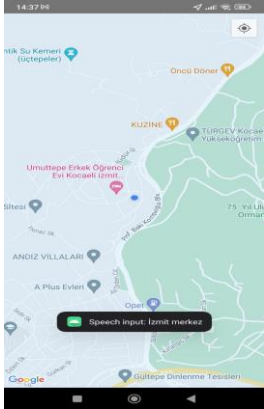
D) Uygulamanın Kullanımı ve Ekran Görüntüleri

Uygulama ilk açıldığında kullanıcının konumu harita üzerinde gösteriliyor:



E. Sonuç Bu proje, sesli komutlarla kullanıcıların hedeflerini belirleyip, Google Maps API'yi kullanarak gerçek zamanlı navigasyon sağlama amacını başarıyla yerine getiren bir mobil uygulamadır. Özellikle erişilebilirlik açısından önemli bir ihtiyacı karşılayan uygulama, görme engelli kullanıcılar için önemli bir yardım aracı olarak kullanılabilir.

Ekranı bir kez dokunduğumuzda mikrofon açılıyor ve ses girişi bekleniyor, örneğin “izmit merkez”:



Sonrasında rota oluşturuluyor ve her 10 saniyede bir güncel konuma göre ses ile yol tarif ediliyor.

F) KAYNAKÇA

- [1] Android Studio: <https://developer.android.com/studio>
- [2] TextToSpeech: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
- [3] SpeechRecognizer: <https://developer.android.com/reference/android/speech/SpeechRecognizer>

[4] Kotlin: <https://kotlinlang.org/>

[5] Google Maps Developer Api Documentation:
<https://developers.google.com/maps/documentation>

[6] OkHTTP: <https://square.github.io/okhttp/>