

SideKick – Navigasyon Uygulaması

Kadir Aydoğdu¹, Oltan Gül², Saffet B. Akabalı³

¹Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
221307085@kocaeli.edu.tr

²Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
211307071@kocaeli.edu.tr

³Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
211307066@kocaeli.edu.tr

Özet

Bu doküman, görme engelli bireyler için sesli komutlarla yol tarifi veren, sesli yönlendirmeler yapan onların hayatına yardımcı olabilecek bir mobil tabanlı android uygulamanın geliştirilme aşamalarını detaylı bir şekilde ele alarak açıklamaktadır.

1. Giriş

Görme engelli veya ileri düzey görme bozuklukları olan bireyler için sesli bir şekilde yol tarifi yapan, mobil tabanlı android uygulama geliştirildi. Geliştirme ortamı olarak Android Studio kullanıldı. Programlama dili olarak Java kullanıldı. Gradle olarak Kotlin DSL kullanıldı. Uygulamamız Android 9.0 ve daha üst sürümlerde çalışmaktadır.

Projede tercihen gerçek zamanlı olarak harita üzerinden çalışabileceği Mapbox API kullanıldı. Tek sayfa olarak tasarlandı. Bu raporda öncelikle tasarım şablonundan, sonra gerekli yapılandırma ayarlarından, daha sonra ise API için gerekli izinlerden ve Mapbox SDK kurulumu ve implementasyonundan bahsedilecek; kodların ve metodların algoritması ile devam edecektir.

2. Tasarım Şablonu

Uygulamamızda ilk olarak RelativeLayout olarak ayarlıyoruz. Bu bize gerçek zamanlı haritanın daha doğru görünmesini sağlayacaktır. Harita için mapView sınıfından katmanı ekliyoruz. Ekranın tam ortasına bir vector mikrofon ekliyoruz. Bu kişinin telefonun ortalama ekranına dokunma konumunda olacaktır. Sağ en alta bulunduğu konumu ayarlayan bir buton, sol en alt tarafa ise rota çizdiren bir buton ekliyoruz. Bu şekilde olmasının temel sebebi kullanıcının daha rahat butonları kullanabilmesidir. Sağ üstte ise sesli yol tarifi için

sesini açıp kapatmak için bir buton ekliyoruz. Kullanıcın mekana göre sesini açıp kapayabilmesini sağlar.

2.1. Android Manifest Yapılandırması

Bu dosya projemizin gerekli izinleri, sayfa geçişlerini, iconlarının ve diğer özelliklerinin belirtildiği genel ayar sayfası olarak düşünülebilir. Biz burada API kullandığımız için gerekli izinleri manifest tagleri arasında belirteceğiz.

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

- Record Audio, ses girişimizi sağlayan mikrofonu kullanmamızı sağlayan izindir.
- Internet, internet bağlantısı kullanmamızı sağlayan izindir.
- Coarse, kullanıcının yaklaşık konumunu bulmaya izin verir.
- Fine, kullanıcının kesin konumunu bulmaya izin verir.
- Post, uygulama içerisinde kullanıcıya bildirim göndermesine izin verir.

2.2. Mapbox API Yapılandırması

Öncelikle mapbox sitesine girerek üyelik oluşturuyoruz. Sitede yeni token oluştur diyerek token oluşturma sayfasına giriyoruz. Token'e herhangi bir isim vererek "Downloads_read" iznini seçerek token oluşturuyoruz. Bu gizli anahtarı tek seferde kopyalamamız gereklidir çünkü sonradan bize de gösterilmez. Anahtarımızı aldıktan sonra

artık Mapbox SDK artık projeye dahil etmemiz gerekmektedir. Gradle script kısmından > gradle.properties dosyasına ulaşıyoruz. En alta dökümantasyonda belirtildiği gibi ekliyoruz. Res klasörü altında values klasörüne gidip “developer-config.xml” dosyası oluşturuyoruz. Bu dosyaya yine belirtildiği gibi anahtarımızı ekliyoruz. Gerekli kütüphaneleri bize sağlayacak olan anahtarımızı başarıyla ekledik. Şimdi kütüphaneleri “build.gradle.kts” dosyasına belirtildiği gibi implemente edelim. Kütüphaneleri ekledik fakat son bir aşama kaldı. Bu da bu kütüphaneleri edinecek olan repoları eklemektir. Şimdi “settings.gradle.kts” dosyasına erişerek maven reposunu belirtildiği gibi eklememiz gerekir. Bu repoyuda eklediğimize göre Mapbox API başarılı bir şekilde projemize entegre etmişiz demektir. Bu işin en ince noktalarından birini halletmiş olduk.

3. Proje Mimarisi

Proje çalışması şu şekilde gerçekleşmelidir. Öncelikle uygulama açıldığında konum izinleri kontrol edilecektir. Daha sonra sağda bulunan konum bul butonu ile mevcut konum bulunacaktır. Ortada bulunan mikrofona hedef konumu açık adres şeklinde belirterek solda bulunan rota çiz butonuna tıklanacaktır. Bu şekilde mevcut lokasyondan hedef konuma bir rota çizilecektir ve sesli yönlendirme işlemi başlayacaktır.

Harita uydü görünümü seçilmiştir. Daha canlı ve gerçekçi olması gerekmektedir. Bu yüzden uygun görülmüştür. Projenin bu tasarımını ve çalışmasını kullanıcıya bir sefer tanıtarak devamlı kullanımı sağlanabilir.

3.1. Mikrofondan Veri Alma

Projedeki “speak()” metodu mikروفon ekranının açılması ve sesi alması için gereklidir. Bunun için Google ses alma sistemi mevcuttur. onActivityResult() metodu, sesi veriye çektikten sonra bir dizide string değer olarak saklar ve bu stringi spokenText değişkeninde saklar. Bu değişken sonradan rota çizmede hedef lokasyon olarak kullanılacaktır.

3.2. Harita Kamerası

Harita kamerası belirli bir konuma ve açıya getirip göstermesi gereklidir. Bunun için “updateCamera” metodu kullanılmıştır. Burada harita kamerasını belirli bir nokta ve ön açıyla taşır.

3.3. Konum ve Sesli Yönlendirme

Konum bilgisini “onNew” ile başlayan metodlar işlemektedirler. Matcher Result metodu, konum bilgisi güncellendiğinde harita kamerasını bu konuma taşımakla görevlidir. OnMove metodları, kullanıcı harita üzerinde hareket etmek istediğinde çalışır. voiceInstructions() metodları, kullanıcının sesli talimatlarla yönlendirilmesini sağlar. onCreate() metodunda speechApi ve mapboxVoiceInstructionsPlayer değişkenleri lokal olarak kullandığımız dile ayarlanmalıdır. Aksi takdirde API bize default olarak ayarladığı dilde yani ingilizcede sesli yönlendirme yapar.

3.4. Rota Çizdirme

Kullanıcının mikrofondan alınan veriyi spokenText değişkeninde saklıyorduk. Şimdi bu değişken fetchRoute() metoduna gidiyor. Parametre olarak gönderilen bu argüman direkt işlenemeyeceği için API sağladığı geocoder nesneleri

ile enlem ve boylam biçimine dönüştürülür. Haritaların temel çalışma prensibi buna dayanır. Kullanıcının söylediği konum alındıktan sonra mevcut konumla orası arasında bir rota çizme işlemi gerçekleştirir. onRoutesChanged() metodu, kullanıcının yeni bir ses girişi ile hedef konum belirttiğinde çalışarak rota bilgisini günceller. Son olarak onDestroy metodu ile işlemler bittikten sonra aktiviteleri temizler.

4. Program Algoritması

1. BAŞLAT
2. onCreate()
3. UI bileşenlerini başlat.
4. Mapbox servislerini başlat.
5. Kullanıcı izinlerini kontrol et.
6. Haritayı yükle ve konumu takip et.
7. mVoiceBtn tıklama
8. speak()
9. onActivityResult()
10. setRoute tıklama
11. fetchRoute(spokenText)
12. Geocode ile adresi dönüştür
13. Mevcut konumu al
14. Rota isteği yap
15. Rota oluştur ve haritada çiz
16. locationObserver
17. onNewLocationMatcherResult()
18. Kullanıcı konumunu güncelle ve haritayı odakla
19. voiceInstructionsObserver
20. onNewVoiceInstructions()
21. Sesli talimatları üret ve oynat
22. onDestroy()
23. Mapbox servislerini kapat ve temizle.
24. BİTİR.

5. Kaynaklar

- [1] [Android Developers Documentation](#)
- [2] [Google AI, \(2024\) Gemini, Google](#)
- [3] [OpenAI Resmi Web Sitesi, \(2024\), GPT-3.5](#)
- [4] [W3Schools](#)
- [5] [Wikipedia](#)
- [6] [GitHub](#)
- [7] [MapBox Docs](#)
- [8] [Mapbox Geocoding API](#)
- [9] [StackOverflow](#)