



Kocaeli Üniversitesi

Teknoloji Fakültesi

Bilişim Sistemleri Mühendisliği

Yazılım Geliştirme Laboratuvarı

Mustafa Toprak
211307054
Teknoloji Fakültesi
Kocaeli, Türkiye

Ayşenur Elibüyük
211307087
Teknoloji Fakültesi
Kocaeli, Türkiye

Muhammet İkbâl Çakır
211307072
Teknoloji Fakültesi
Kocaeli, Türkiye

Özet—Bu proje, kullanıcıların belirli bir konuma gitmelerini kolaylaştırmak amacıyla geliştirilmiştir. Kullanıcı, gitmek istediği yeri telefona sesli olarak söyleyerek giriş yapar. Bu sesli girdi, metne çevrilir ve ardından kullanıcının harita üzerindeki hedef konumu belirlenir. Belirlenen konum ile kullanıcının mevcut konumu arasındaki en yakın durak veya özel işaretlenmiş bir nokta arasında bir rota oluşturulur. Kullanıcı, oluşturulan bu rotada sesli yönlendirmeler ile ilerler. Yönlendirmeler, "10 metre düz git, sağa dön, 5 metre sonra sola dön" gibi basit ve anlaşılır komutlardan oluşur. Yönlendirme sırasında kullanıcıya sesli komutların duyulamaması durumunda cihaz titreşimi ile bildirim yapılabilir ve komutlar birkaç kez tekrarlanabilir. Ayrıca, kullanıcı yanlış bir yöne yöneldiğinde "Rotadan çıktıldı" gibi uyarılarla bilgilendirilir. Kullanıcı belirtilen hedefe ulaştığında sesli yönlendirme sona erer ve bir sonraki hedef için yeni bir sesli giriş beklenir. Bu proje, kullanıcıların doğru bir şekilde hedeflerine ulaşmalarını sağlamayı amaçlar ve engellerden sakınma gibi ileri seviye özellikler eklenmesi halinde daha da geliştirilebilir.

Keywords— Sesli Yönlendirme, Metin Çevirisi, Harita Navigasyonu, Konum Tespiti, Rota Oluşturma

I. GİRİŞ

Bu projenin temel odak noktası, görme engelli kullanıcıların günlük hayatta karşılaştıkları zorlukları azaltmak ve onların belirli bir konuma ulaşmalarını

kolaylaştırmaktır. Geliştirilen uygulama, kullanıcıların sesli komutlarla hedeflerini belirtmelerine olanak tanır ve bu sesli komutları metne çevirerek işler. Bu özellik, görme engelli bireylerin kolayca iletişim kurmasını ve hedeflerini belirtmesini sağlar. Google Maps API'leri, harita üzerindeki konum belirleme, rota oluşturma ve yönlendirme gibi işlevleri sunar. Bu özellikler, görme engelli kullanıcıların belirli bir konuma gitmelerini kolaylaştırmak için hayati öneme sahiptir. Sesli yönlendirmelerle birlikte, kullanıcılar adım adım doğru yola yönlendirilir ve hedeflerine güvenli bir şekilde ulaşabilirler. Ayrıca, titreşim bildirimleri gibi ek özellikler, kullanıcıların dikkatini çekmek ve gerektiğinde yönlendirmeleri duyurmak için kullanılır. Yanlış yöne yönelme durumunda yapılan uyarılar da görme engelli kullanıcıların doğru rotaya dönmesine yardımcı olur. Bu proje, görme engelli bireylerin günlük yaşamlarını kolaylaştırmak ve onlara bağımsızlıklarını artırmak amacıyla geliştirilmiştir. Yolculuklarında daha güvenli ve rahat hissetmelerini sağlayarak, yaşam kalitelerini artırmayı hedefler.

A. Projenin Mobil Yapısı

BLoC (Business Logic Component), Flutter uygulamalarında kullanıcı arayüzü ve iş mantığını ayrı tutarak uygulamanın daha modüler ve yönetilebilir olmasını sağlayan bir tasarım desendir. Bu mimari, uygulamaların geliştirilmesini, test edilmesini ve bakımını kolaylaştırırken,

kodun daha okunabilir ve sürdürülebilir olmasını sağlar. Projemizin mobil kısmında, kullanıcı uygulamayı başlattığında karşısına bir hoş geldin ekranı çıkar ve burada "Nereye gitmek istersiniz?" gibi bir soruyla karşılaşır. Bu soru, kullanıcının hedefini belirtmesi için bir giriş noktasıdır. Kullanıcı, sesli olarak veya metin girişiyle bir hedef belirtir; örneğin, "Beni en yakın otobüs durağına götür" gibi bir talepte bulunabilir. Bu talep alındığında, kullanıcının konumu belirlenir. Flutter uygulamalarında bu, cihazın konum servislerinden alınan verilere dayanarak gerçekleştirilir. Kullanıcının anlık konumu (latitude ve longitude) tespit edilir ve bu bilgi Google Haritalar API'leri kullanılarak işlenir. Google Haritalar API'leri, en yakın otobüs durağını belirlemek için kullanılır. Bu API'ler aracılığıyla, kullanıcının bulunduğu konuma en yakın otobüs durağının koordinatları tespit edilir. Daha sonra, bu iki nokta arasında bir rota oluşturulur ve kullanıcıya sunulur. Bu işlemler BLoC mimarisi kullanılarak yönetilir. Kullanıcı arayüzünden gelen talepler, ilgili BLoC sınıfına iletilir. BLoC sınıfı, iş mantığını yürütür, gerekli API çağrılarını gerçekleştirir ve sonuçları kullanıcı arayüzüne iletir. BLoC mimarisi, iş mantığını ve kullanıcı arayüzünü birbirinden bağımsız hale getirir. Bu, uygulamanın daha kolay test edilmesini, bakımını ve geliştirilmesini sağlar. Ayrıca, kod tekrarının azaltılmasına ve uygulamanın ölçeklenebilirliğinin artırılmasına yardımcı olur.

1) Anasayfa

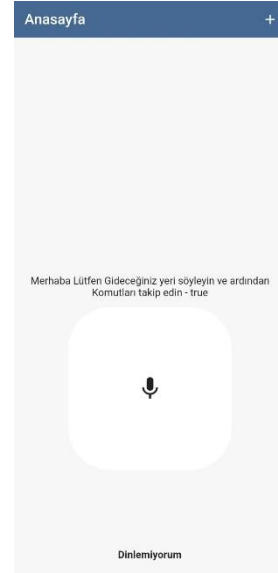
Anasayfa, uygulama açıldığında kullanıcıyı karşılayan ilk ekrandır. Bu ekran, kullanıcıdan gitmek istediği konumu belirtmesini ister ve bu işlemi şu şekilde yönetir:

Kullanıcının Konum Belirleme Süreci:

- 1. Sesli Giriş Talebi:** Uygulama açıldığında kullanıcıya "Nereye gitmek istersiniz?" şeklinde bir sesli talep gönderilir. Bu, görme engelli kullanıcıların kolayca hedeflerini belirtmelerine olanak tanır. [Resim 1]
- 2. Sesli Girdinin Alınması:** Kullanıcı, hedefini sesli olarak belirtir. Bu aşamada, cihazın mikrofonu kullanılarak sesli giriş alınır. Flutter'da sesli giriş almak için `speech_to_text` gibi paketler kullanılır.
- 3. Sesli Girdinin Metne Çevrilmesi:** Alınan sesli girdi, metne çevrilir. Bu işlem için Google Cloud Speech-to-Text API kullanılır. Bu API, sesli girdiyi alır ve metne dönüştürerek, uygulamanın işleyebileceği bir formatta geri döner.
- 4. Onay İsteği:** Metne dönüştürülen girdi, kullanıcıya geri okunur ve "Gitmek istediğiniz konum bu mu?" şeklinde bir onay talebi yapılır. Bu, kullanıcının belirttiği konumun doğru bir şekilde alınıp alınmadığını doğrulamak için önemlidir.
- 5. Onay İşlemi:** Kullanıcı, sesi duyduktan sonra "Evet" veya "Hayır" şeklinde yanıt verir. Eğer kullanıcı "Evet" derse, uygulama konum belirleme ve rota oluşturma işlemlerine geçer.

BLoC Kullanımı:

BLoC mimarisi, yukarıdaki işlemlerin yönetilmesinde kritik bir rol oynar. Kullanıcı arayüzünden gelen sesli giriş, BLoC sınıfına iletilir. BLoC sınıfı, sesli girişi metne çevirme ve metni işleme mantığını yürütür. Ayrıca, kullanıcından gelen onay yanıtını işler ve rota oluşturma işlemini başlatır.



[Resim 1] - Anasayfa

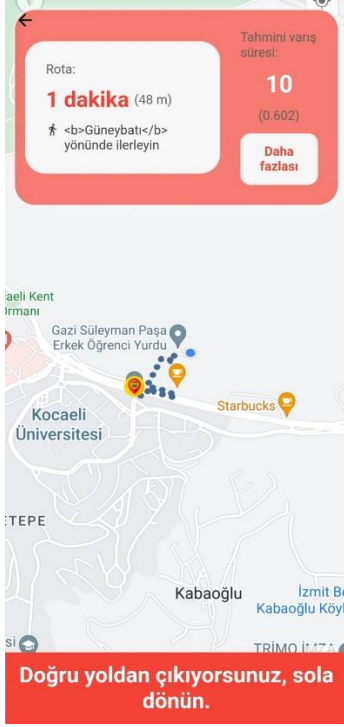
2) Harita Sayfası

Harita sayfası, kullanıcının belirttiği konuma olan rotayı görsel ve sesli olarak sunan ekrandır. Bu sayfa, kullanıcının anlık konumunu ve hedef konumunu belirler, ardından bu iki nokta arasında bir rota oluşturur ve kullanıcıyı adım adım yönlendirir. [Resim 2]

Rota Oluşturma Süreci:

- 1. Kullanıcının Anlık Konumunu Belirleme:** Harita sayfası açıldığında, uygulama kullanıcının anlık konumunu belirler. Bu işlem, cihazın GPS ve konum servisleri kullanılarak gerçekleştirilir. Flutter'da `geolocator` veya `location` gibi paketler kullanılarak konum bilgisi alınır.
- 2. En Yakın Otobüs Durağını Bulma:** Kullanıcının anlık konumu belirlendikten sonra, Google Places API kullanılarak en yakın otobüs durağı tespit edilir. Bu API, kullanıcının belirttiği konum etrafındaki ilgi noktalarını (POI) belirler ve en yakın otobüs durağının koordinatlarını döner.
- 3. Rota Oluşturma:** Google Directions API kullanılarak, kullanıcının anlık konumu ile en yakın otobüs durağı arasındaki rota oluşturulur. Bu API, iki nokta arasındaki en uygun rotayı hesaplar ve bu rotayı harita üzerinde gösterir.
- 4. Sesli Yönlendirme:** Kullanıcı, harita üzerinde oluşturulan rota boyunca sesli yönlendirmelerle bilgilendirilir. Örneğin, "10 metre düz git, sağa dön, 5 metre sonra sola dön" gibi komutlar verilerek kullanıcıya yol tarifi yapılır. Bu sesli yönlendirmeler, Google Maps Navigation API tarafından sağlanan veriler kullanılarak oluşturulur.
- 5. Titreşim Bildirimi:** Yönlendirme sırasında, kullanıcı sesli komutları duyamayabilir. Bu durumu önlemek için cihaz titreşim bildirimi yapar. Bu, her yönlendirme komutunda cihazın titreşmesini sağlayarak kullanıcıya ek bir bildirim sunar.
- 6. Yanlış Yön Uyarısı:** Kullanıcı yanlış yöne döndüğünde veya rotadan çıktığında, uygulama

"Rotadan çıkıldı" şeklinde bir uyarı verir ve kullanıcının doğru yola dönmesini sağlar. Bu, Google Maps API'nin konum takip ve rota yeniden hesaplama özellikleri kullanılarak gerçekleştirilir.



[Resim 2] Harita Sayfası

BLoC Kullanımı:

BLoC mimarisi, harita sayfasında da önemli bir rol oynar. Kullanıcının anlık konum bilgisi, en yakın otobüs durağını belirlenmesi, rota oluşturma ve sesli yönlendirme işlemleri BLoC katmanında yönetilir. Bu, kullanıcı arayüzünden bağımsız olarak iş mantığının yürütülmesini sağlar ve kodun daha temiz ve yönetilebilir olmasını sağlar.

B. Projenin Api Yapısı

Projenin API yapısı, Google Maps Platform'un sunduğu çeşitli API'lerle entegrasyonu içerir ve bu API'lerin kullanımını ve yönetimini içerir. Google Maps API'leri, harita tabanlı hizmetler sağlayan ve konum tabanlı işlevleri destekleyen güçlü bir araç setidir. Bu projede, API yapısı aşağıdaki unsurları içerir:

1) Dio Kütüphanesi Kullanımı

- Dio kütüphanesi, Flutter uygulamalarında HTTP istekleri yapmak için tercih ettiğimiz bir araçtır. Bu kütüphane, kolay kullanımı ve güvenilir performansı ile öne çıkar.

- Uygulamamızda, Google Maps API'leri ile iletişim kurmak için Dio kütüphanesini kullandık. Dio, RESTful API'lerle iletişim kurmak için uygundur ve bu özelliği sayesinde Google Maps API'leriyle etkili bir şekilde entegre olmamızı sağladı.

2) Google Maps Api

- Google Maps API'leri, harita tabanlı hizmetler sunan ve çeşitli konum tabanlı işlevleri destekleyen bir hizmet paketidir.
- Projemizde, Google Maps Platform'un Directions API, Places API ve Speech-to-Text API'lerini kullandık. Directions API, kullanıcının konumunu belirledikten sonra en uygun rota algoritmasını kullanarak rotalar oluştururken, Places API, kullanıcının bulunduğu konuma en yakın otobüs durağını tespit etmek için kullanıldı. Ayrıca, Speech-to-Text API, kullanıcıların sesli girişlerini metne dönüştürmek için kullanıldı.

3) Ücretlendirme Modeli

- Google Maps Platform'un API'leri, belirli bir kullanım limiti aşıldığında ücretlendirme yapar. Bu ücretlendirme modeli, API kullanımının türüne ve miktarına göre değişir.
- Projemizde, özellikle kullanıcıların sık sık otobüs durağı araması gibi yoğun API kullanımı gerekebileceğini göz önünde bulundurarak ücretlendirme modelini dikkate aldık. Bu doğrultuda, gereksiz API çağrılarını minimize ederek maliyeti düşürmeye çalıştık.

4) Bloc ve MVVM Kullanımı

- Projemizde, BLoC (Business Logic Component) mimarisi ile MVVM (Model-View-ViewModel) tasarımı desenini bir araya getirdik
- BLoC, iş mantığını kullanıcı arayüzünden izole ederek, uygulamanın daha modüler ve yönetilebilir olmasını sağladı. Ayrıca, MVVM, kullanıcı arayüzünü ve iş mantığını birbirinden ayırarak, kodun daha okunabilir ve sürdürülebilir olmasını sağladı.

Bu yapı, uygulamanın güvenilir, ölçeklenebilir ve bakımı kolay olmasını sağlarken, Google Maps API'leri gibi güçlü araçlarla entegrasyon yaparak kullanıcılara zengin ve etkili bir deneyim sunmayı hedefledik.

KAYNAKÇA

- [1] <https://docs.flutter.dev>
- [2] <https://bloclibrary.dev>
- [3] <https://developers.google.com/docs/api/reference/rest?hl=tr>