

**denz Uygulaması Dokümantasyonu**

İçindekiler

[**1. Giriş** 2](#_Toc118250154)

[**1.1 Genel Tanım** 2](#_Toc118250155)

[**1.2 Amaç** 2](#_Toc118250156)

[**1.3 Kullanılan Teknolojiler** 2](#_Toc118250157)

[**2. UYGULAMA İÇİ KULLANILAN YAPILAR** 2](#_Toc118250158)

[**2.1 UYGULAMAYI Çalıştırmak** 2](#_Toc118250159)

[**2.2 UYGULAMADA KULLANILAN PAKETLER** 3](#_Toc118250160)

[**2.3 UYGULAMANIN KLASÖR YAPISI** 4](#_Toc118250161)

[**Özel Not: .env dosyası** 4](#_Toc118250162)

[**3. UYGULAMA İÇİ EKRANLAR** 5](#_Toc118250163)

[**anasayfa** 5](#_Toc118250164)

[**İskele Kalkış Ekranı** 6](#_Toc118250165)

[**İskele Varış Ekranı** 7](#_Toc118250166)

[**Taşıt Seçim Ekranı** 8](#_Toc118250167)

[**Yolculuk Ekranı** 9](#_Toc118250168)

[**Üyelik Ekranı** 10](#_Toc118250169)

[**Giriş Ekranı** 11](#_Toc118250170)

# **1. Giriş**

## **1.1 Genel Tanım**

Bu uygulama ile kullanıcı istediği limana belirli deniz araçlarını çağırabilir veya rezervasyon yaptırabilir.

## **1.2 Amaç**

**Müşteri tarafından istenilen liman konumuna deniz aracı çağırmak.**

**Deniz taşımacılığını daha ulaşılabilir kılmak.**

**Deniz taşımacılığını kişiselleştirmek.**

## **1.3 Kullanılan Teknolojiler**

* Uygulamayı geliştirirken **React Native** framework kullanıldı.
* **JavaScript** dili ile yazıldı.
* **Expo CLI** kullanıldı.
* State yönetimi **Redux** ile yapıldı
* Backend ve Authentication bölümlerinde **Google Firebase** kullanıldı.
* Konum servisleri için:

**Google Maps Directions API**

**Google Maps Places API**

**Google Maps Distance Matrix API**

**Google Autocomplete** kullanıldı.

# **2. UYGULAMA İÇİ KULLANILAN YAPILAR**

## **2.1 UYGULAMAYI Çalıştırmak**

Uygulama dosyalarının çalışması için JavaScript çalıştıran bir IDE’ye ihtiyaç duyacaksınız (Visual Studio Code önerilir). Bunun ardından, NPM (Node Package Manager) dosyalarının indirilip çalıştırılması gerekir. Bunun için Visual Studio Code’dan projenin dosyalarını açtıktan sonra dosyaların ana dizininde Node JS terminali açmak ve ardından **npm install** yazmak yeterli olur.

C:\YourDirectory\Suber1>npm install

Buradan sonra uygulamayı çalıştırmak oldukça basit.

Uygulamanın çalışması için terminale **npm start** yazmanız yeterli olacak.

C:\YourDirectory\Suber1>npm start

Expo GO uygulamasını telefonunuza indirdikten sonra uygulamanızı gerçek zamanlı olarak fiziksel mobil cihazınızda kullanabilirsiniz. Burada EXPO CLI’ın sağladığı özelliklerden biri olan QR Kod okutma özelliği sayesinde uygulama telefonda kullanılmaya başlanabilir.

## **2.2 UYGULAMADA KULLANILAN PAKETLER**

    "@react-navigation/bottom-tabs": "^6.4.0",

    "@react-navigation/stack": "^6.3.1",

    "@reduxjs/toolkit": "^1.8.5",

    "expo": "~46.0.9",

    "expo-status-bar": "~1.4.0",

    "firebase": "^8.2.3",

    "react": "18.0.0",

    "react-native": "0.69.6",

    "react-native-dotenv": "^3.3.1",

    "react-native-google-places-autocomplete": "^2.4.1",

    "react-native-maps": "0.31.1",

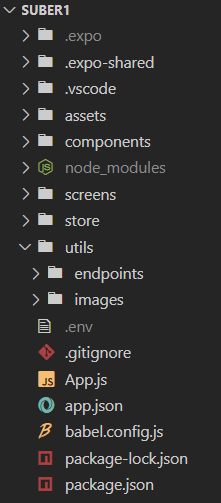
    "react-native-maps-directions": "^1.9.0",

    "react-native-safe-area-context": "4.3.1",

    "react-native-screens": "~3.15.0",

    "react-redux": "^8.0.4"

## **2.3 UYGULAMANIN KLASÖR YAPISI**

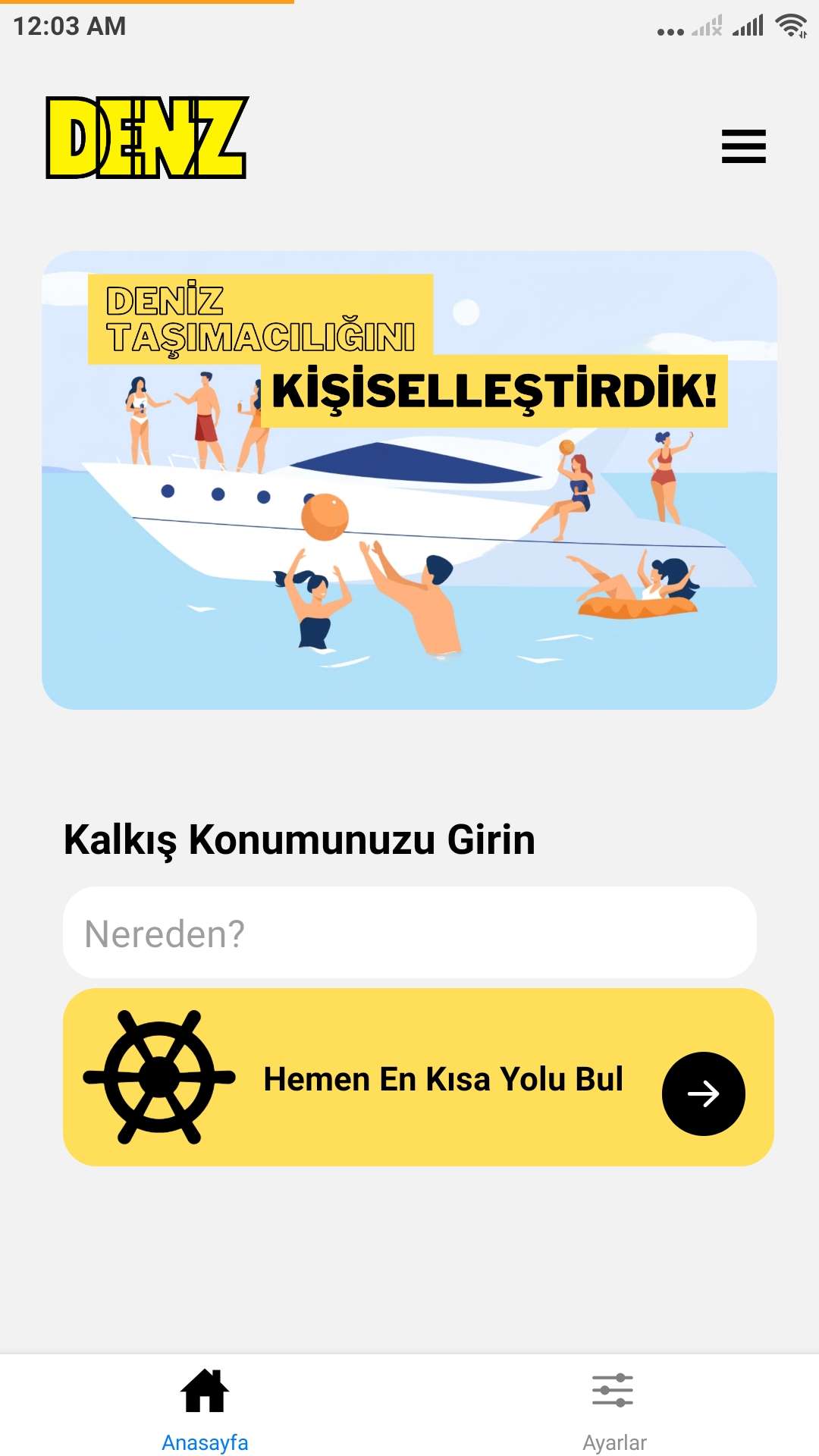


### **Özel Not: .env dosyası**

Bu dosyanın oluşturulma nedeni Google API anahtarını ayrı bir dosyada tutmak ve gerektiğinde başka dosyalar içinde kullanmaktır. Bu dosyayı kullanabilmek için babel.config.js isimli dosyada .env ayarlarını yaptık.

# **3. UYGULAMA İÇİ EKRANLAR**

## **anasayfa**



Bu sayfada kullanıcı bulunduğu konumu yazar ve “Hemen En Kısa Yolu Bul” butonuna tıklayarak deniz taşıtı servislerinin ilk işlemini başlatır.

## **İskele Kalkış Ekranı**

harita içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu sayfada kullanıcı bulunduğu konumdan seçtiği iskeleye gitmek için iskele butonlarının üstüne tıklar ve Google Maps API iki konum arasında karayolu çizer. İşleme devam etmek için “Varış Limanını Seçin” butonuna tıklar.

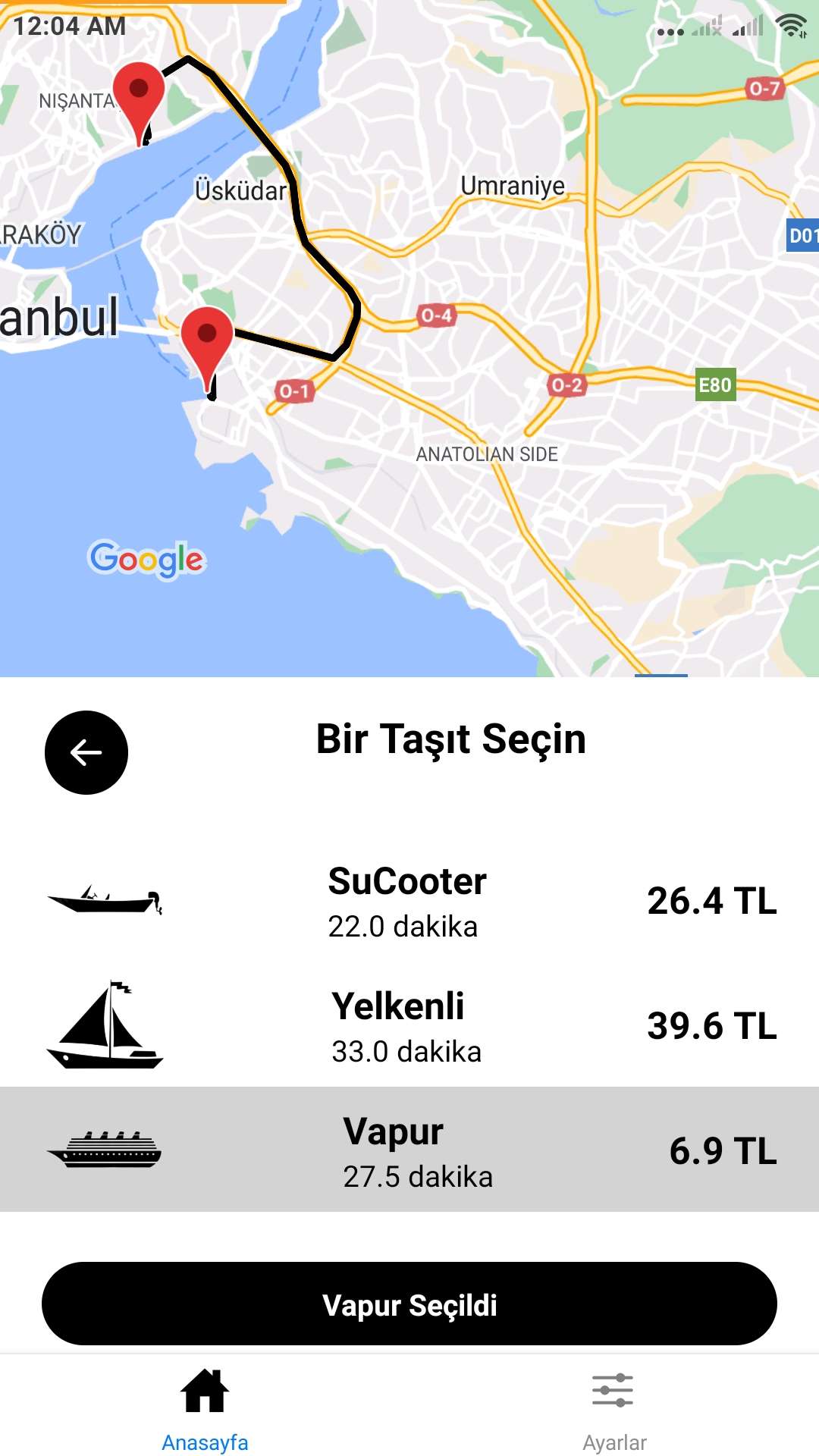
## **İskele Varış Ekranı**

harita içeren bir resim

Açıklama otomatik olarak oluşturuldu

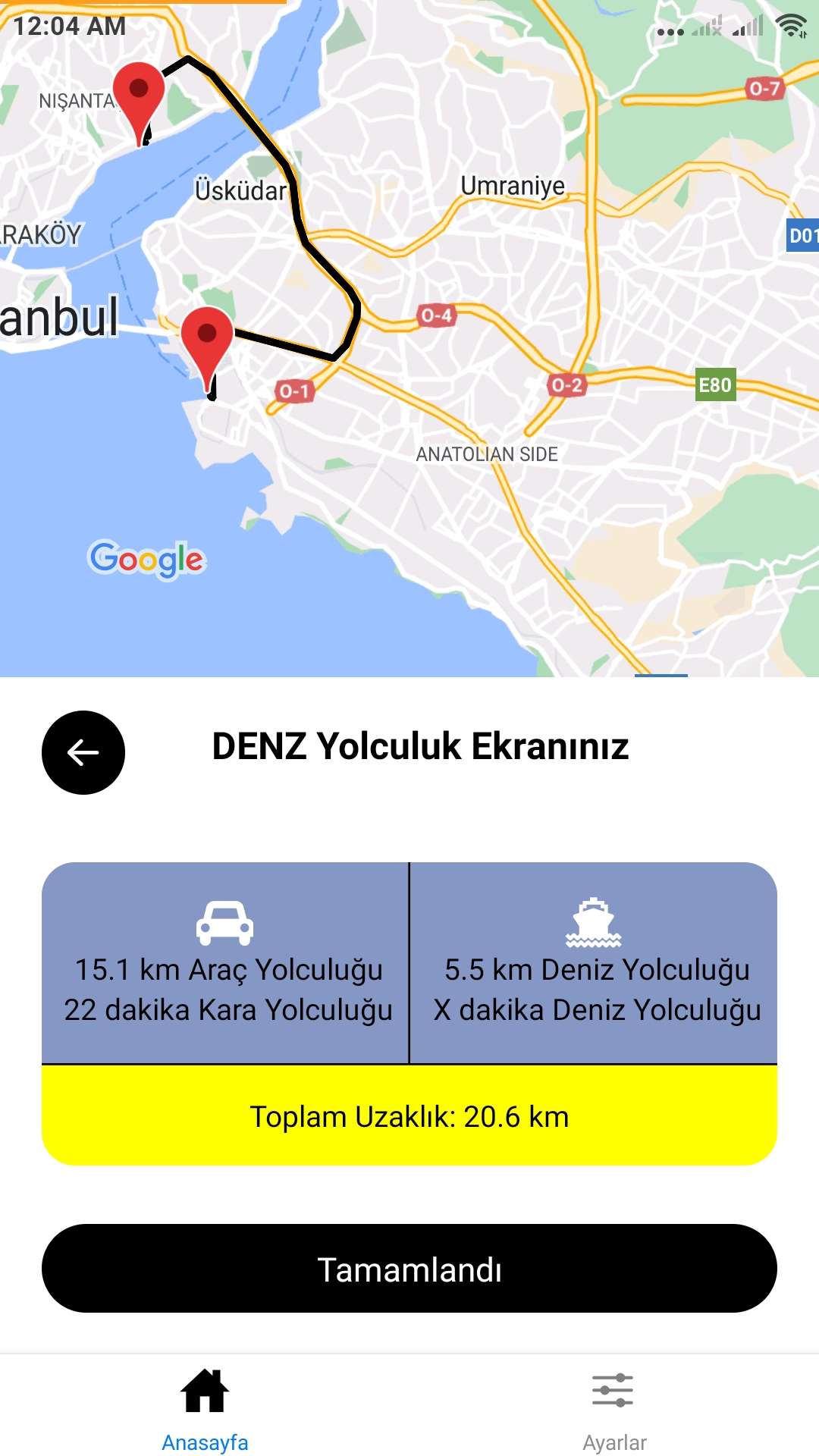
Bu sayfada kullanıcı bulunduğu iskeleden seçtiği iskeleye gitmek için iskele butonlarının üstüne tıklar ve Haversine Formülü ile iki konum arasındaki kuş uçuşu mesafe hesaplanır. Kullanıcı İşleme devam etmek için “Taşıt Seçeneklerini Görün” butonuna tıklar.

## **Taşıt Seçim Ekranı**



Bu sayfada kullanıcı bulunduğu iskeleden seçtiği iskeleye gitmek için hangi deniz taşıtını kulllanmak istiyorsa üzerine tıklar ve konumlar arası ücreti ve yolculuk süresini öğrenebilir. Kullanıcı İşleme devam etmek için “Vapur(Hangi taşıt seçildiyse) Seçildi” butonuna tıklar.

## **Yolculuk Ekranı**



Bu sayfada kullanıcı yolculuk ekranından kendi yolculuğunun süresini ve toplam yolculuk mesafesini görebilir. “Tamamlandı” butonuna basarak anasayfaya geri döner.

## **Üyelik Ekranı**



Bu ekranda kullanıcı Google Firebase sistemi ile e-posta ve şifresini belirleyip uygulamaya üye olabilir. Eğer önceden bir hesaba sahipse “Hesabınız var mı?” butonuna tıklayarak Giriş Ekranına ulaşır.

## **Giriş Ekranı**



Bu ekranda kullanıcı Google Firebase sistemi ile önceden belirlediği e-posta ve şifresini girerek uygulamaya giriş yapabilir. Eğer önceden bir hesabı yoksa “Hesabınız Yok mu?” butonuna tıklayarak Üyelik Ekranına ulaşır.