

PROJE 2: Mobil Sorgular

Erdem Nayın 180202050 – Melih Yeşilyurt 180202060

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

erdem_nayin@hotmail.com – Melih58yesilyurt@outlook.com

Proje, android uygulaması. Bu android uygulaması cloud ortamında bulunan database ile bağlantı kurarak daha önceden hazır olarak verilen sorguların yapılmasını, sorgu sonuçlarının uygulama ekranında gösterilmesi, sorgu sonucuna göre Google Maps API kullanılarak harita gösterimi, harita üzerinde iki lokasyon arasında rota çizimi gibi işlemleri gerçekleştirir.

1.Yazılım Dili ve Geliştirme Ortamı

Projenin kodları Android Studio kullanılarak Java dilinde yazılmış ve geliştirilmiştir.

2. Geliştirme Aşamaları

a. Uygulama Ara Yüzünün Tasarlanması

Bu projedeki aşamaları gerçekleştirmek adına ilk olarak android uygulamamızın kullanıcı ile etkileşime gireceği kısım olan kullanıcı ara yüzü tasarlandı. Ara yüz, projeyi 3 farklı soru grubunun içerisinde birer soru seçerek toplamda 3 soru gerçekleştirmemiz gerektiğinden 3 ayrı fragment olarak tasarlandı. Eğer sorgu için kullanıcı girişi gerekiyorsa kullanıcının istediği değerleri girebilmesi için textbox alanları ve butonlar tasarlandı.

b. Verilerin Cloud Ortamına Aktarılması

Bu aşamada sorgular yapılırken kullanılması için daha önceden bizlere verilen hazır veriler, Google Compute Engine üzerinde kurduğumuz Firebase CloudSQL'e aktarıldı. Hatalı veya eksik değeri olan kayıtlar, veri setlerinden silinerek doğru ve tutarlı sorgulama sonuçlarının alınması sağlandı.

c. Google Servisleriyle Bağlantı Kurulması

Bu aşamada ilgili API'ler ve API Key'leri kullanılarak, uygulamada kullanılan servislerin, Google servisleriyle bağlantısı sağlandı.

d. Sorguların Hazırlanması

Bu aşamada bizlere daha önceden verilen 3 tip sorgu grubu içerisinde birer soru seçildi. Daha sonra bu soruların ilgili database tabloları üzerinde doğru sonuçları bulması için gerekli Cloud SQL sorguları, MSSQL ile Cloud SQL'e bağlanılarak yazıldı ve kontrol edildi.

e. Uygulama ile Sorgu İşlemlerinin Entegrasyonu

Bu aşamada uygulama ile CloudSQL arasında bağlantı kurularak bir önceki adımda hazırlanan sorguların CloudSQL ortamında gerçekleştirilmesi ve sonuçları uygulamaya geri göndermesi gerçekleştirildi.

f. Sorgu Sonuçlarının Kullanıcı Ara Yüzünde Gösterilmesi

Bu aşamada ilk ve ikinci tip sorgular için, kullanıcıdan yapmak istediği sorguda kullanıcı girişi gerektiren değer varsa onunla birlikte, seçtiği sorgunun sonucunun ekranda gösterilmesi gerçekleştirildi.

g. Sorgu Sonucunun Harita Üzerinde Gösterilmesi

Bu aşamada üçüncü tip sorgu için ayrı bir ara yüz tasarlanarak Google Maps API ile oluşturulan harita ara yüzünde, sorgu sonucunda uygulamaya sorgu sonucu döndürülen iki lokasyon arasında rota çizimi gösterildi.

3. Uygulamanın Çalışması

Uygulama açıldığında uygulamanın üst kısmında üç sorguya ithafen yapılan üç fragmentin navigation barı görülür. Kullanıcı buradan sorgular arasında geçiş yapabilir.

Uygulama açıldığında kullanıcının karşısında ilk tip sorgunun yapıldığı birinci fragment kısmı gelmektedir. İlk tip sorgu olarak "En fazla yolcu

taşınan 5 günü ve toplam yolcu sayısını listelersiniz.” seçildi. İlgili butona tıklanıldığında sorgu CloudSQL’ e gönderilir, sorgu sonucu uygulamaya geri döndürülerek sorgu sonucunda elde edilen bilgiler ekranda gösterilir.

Kullanıcı eğer navigaton barı kullanarak ikinci tip sorgunun yapıldığı ikinci fragment kısmına girerse karşısına iki adet tarih girebileceği iki textbox gelir. İkinci tip sorgu olarak “İki tarih arasında seyahat edilen en az mesafeli 5 yolculuk hangisidir?” sorgusu seçildi. Kullanıcı istediği iki tarihi girdikten sonra ilgili butona tıkladığında sorgu CloudSQL’ e gönderilir, sorgu sonucu uygulamaya geri döndürülerek sorgu sonucunda elde edilen bilgiler ekranda gösterilir.

Kullanıcı eğer navigation barı kullanarak üçüncü tip sorgunun yapıldığı üçüncü fragment kısmına girerse karşısına tarih girebileceği bir textbox gelir. Üçüncü tip sorgu olarak “Belirli bir günde en uzun seyahatın harita üstünde yolunu çizersiniz. Başlangıç ve varış konumları lokasyonun merkezi kabul edip mesafeye göre bir yol bulunmalıdır” sorgusu seçildi. Kullanıcı istediği tarihi girdikten sonra ilgili butona tıkladığında sorgu CloudSQL’e gönderilir, sorgu sonucu uygulamaya geri döndürülür. Uygulama ara yüzünde harita açılır. Döndürülen veriler kullanarak harita üzerinde iki lokasyon arasında rota çizdirme işlemi yapılır ve harita yolculuk başlangıç lokasyonuna doğru zoomlanarak ilgili rota kullanıcıya gösterilir

4. Kullanılan Classlar

A. MainActivity

Bu class uygulamanın çalışmasını sağlar. Ayrıca database ile olan bağlantıyı bu class gerçekleştirmektedir. Bütün sorgular bu class da bulunur. Diğer classlar tarafından çağırılırlar.

B. Fragment1

Bu class tab menüsünün birinci sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde birinci sorguyu çağırır. En fazla yolcu taşınan 5 günü ve toplam yolcu sayılarını ekrana bastırır.

C. Fragment2

Bu class tab menüsünün ikinci sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde kullanıcıdan tarihleri alır ve ikinci sorguyu çağırır. İki tarih arasında seyahat edilen en az mesafeli 5 yolculuğu ekrana bastırır.

D. Fragment3

Bu class tab menüsünün üçüncü sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde kullanıcıdan tarih alır ve yeni bir activity açarak, SubActivity classını çalıştırır.

E. SubActivity

Bu class Fragment3 classı tarafından çalıştırılır. İçerisinde Google map apiye

bağlanmış harita bulundurmaktadır. Fragment3 classı tarafından gönderilen tarihi kullanarak 3. ve 4. Sorguları çalıştırır. Elde ettiği noktaların koordinatlarını haritada bularak aralarındaki en kısa yol güzergahını çizdirir.

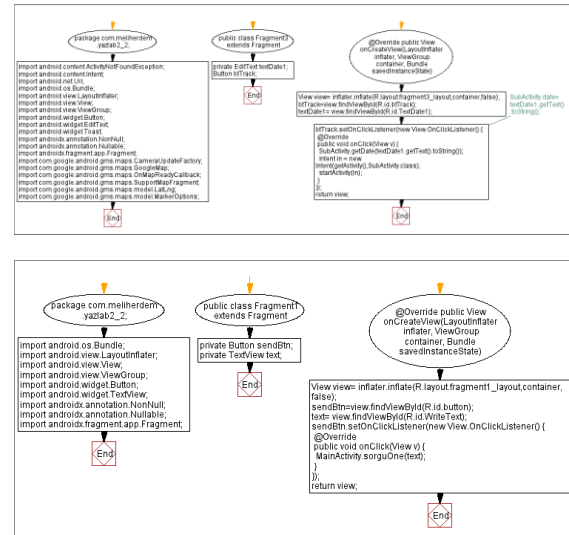
5. Karşılaşılan Problemler

1. Android ile uygulama geliştirme konusunda deneyimli olmadığımız için arayüz kısmını tasarlama aşamasında biraz zorlandık. Yaptığımız araştırmalar sonucunda fragment yapısını ve fragmentler arasında geçiş yapmayı sağlayan navigaton bar ikilisini kullanmaya karar verdik. Kullanıcı arayüzünü bu sayede 3 ayrı bölmeye ayırıp sorguların ve sorgu sonuçlarının birbirlerinden ayrılmasını sağladık.
2. Daha önceden bize verilen .csv uzantılı taksit kaydı datalarının cloud ortamına yüklenmesi konusunda zorluk yaşadık. Projede ilk önce Firebase Firestore kullanmaya karar verdik. CSV uzantılı dosyamızı Firestore database’ine direkt olarak yükleyemeyeceğimizi öğrendik. Bu dosyayı yüklemek için ilk önce JSON formatına getirip daha sonra yükleyebileceğimizi öğrendik. CSV dosyamızı JSON formatına dönüştürürken zorluklar yaşadık. CSV dosyasında yaklaşık 1.5 milyon kayıt bulunduğundan online converterlar dosyayı çeviremiyordu. Dosyadan kayıt silip çevirmeye çalıştığımızda bazı dataların hatalı şekilde çevrildiğini ve data kaybı yaşadığımızı gördük. Daha sonra çeşitli programlar aracılığıyla bu dosyayı çevirmeye karar verdik. Kullandığımız hiçbir program bu dosyayı çeviremedi. Daha sonra python kullanarak bu dosyayı çevirebileceğimizi öğrendik ve bulduğumuz python kodunu ve kütüphaneyi kullanarak datamızı çevirmeye çalıştık. Fakat data çok büyük olduğundan ve datada hatalı değerler bulunduğundan yine çeviremedik. Bu dosyadan daha küçük olan taxi zones dosyasını ise kolaylıkla çevirebildik. Daha sonra bu data çok büyük olduğundan datayı kırarak python koduyla birlikte çevirmeyi başardık ve Firestore’a yükleyebildik.
3. Firestore’a yüklediğimiz datalar üzerinde sorgularımızı yapmaya çalıştık fakat Firestore genelde güncel android uygulaması yazılımı koşullarını karşılıyordu. Bizim istediğimiz şekilde dataları sorgulamayı başaramadık. Firestore relational database olmadığı için işimize tam anlamıyla yaramadığını, sadece basit data al ver işlemlerini yapabildiğini gördük. SQL sorgularını Firestore sorgularına çeviren bir kütüphane bulduk ve onu kullanarak sorgulama yapmaya çalıştık fakat bu kütüphane de işimizi görmedi. Bu yüzden

yine Firebase'e ait olan RealtimeDatabase'e geçmeye karar verdik.

4. Firestore RealtimeDatabase'e csv dosyamızı kolaylıkla import edebildik çünkü RealtimeDatabase bu desteği kendi ara yüzünde sunmuştu. Yüklediğimiz datalar üzerinde Google Cloud hizmeti olan BigQuery hizmetini kullanarak bu datalar üzerinde sorgulama işlemlerimizin demosunu yapmış olduk ve sorgularımız hazırlanmıştı. Daha sonra RealtimeDatabase sorguları nasıl yapılır araştırmaya başladık. Sonuç olarak yine hüsrana uğradık ve RealtimeDatabase'in sorgularının Firestore sorgularından bile daha az kapsamlı olduğunu ve sql queryleri gibi yapılmadığını gördük. Bizim istediğimiz şekilde sorgular yapılamıyordu.
5. Daha sonra olarak yine Google'ın Cloud hizmeti olan ComputeEngine sanal bilgisayarı üzerinde tanımladığımız yine Google'ın Cloud hizmeti olan CloudSQL for MySQL kullanmaya karar verdik. Sorgularımızı MySQL sorgularına çevirdik. Yaşadığımız bazı hatalardan dolayı CloudSQL for MySQL kullanmayı bırakıp CloudSQL for MSSQL kullanmaya başladık. Microsoft SQL Server Management Studio kullanarak bağlandık ve sorgularımızı tekrardan MSSQL sorgularına uygun haline getirdik. CloudSQL ile uygulamamızın bağlantısını yaparak uygulama içinden sorguları CloudSQL'e göndererek gerçekleştirdik.
6. Üçüncü tip sorunun Google Maps API kullanılarak harita üzerinde gösterilmesinde sorunlar yaşadık. Araştırdığımız bazı çözümler, izlediğimiz bazı tutorialar eski olduğundan deprecated olmuş birçok fonksiyon ve kullanım yöntemleriyle karşılaştık. Bu sorunları Android Developers dokümantasyonlarını kullanarak düzeltmeye ve güncellemeye çalıştık fakat olmadı. Yaptığımız sayısız araştırma sonucu güzel bir biçimde nasıl çizdirebileceğimizi bulduk ve uygulamamızın harita üzerinde rota çizimi işlemini böylece gerçekleştirmiş olduk.
7. Üçüncü tip sorunun sonucunda elde edilen lokasyon bilgilerinin fragment ve activity arasında iletilmesinde sorun yaşadık. Kullandığımız taxi_zones.csv dosyasında latitude ve longitude veri tiplerini float olarak belirlemiştik ve bu şekilde importlamıştık fakat bunu uygulamaya çekerken data kaybı olduğunu gördük. Bu yüzden tabloyu silip baştan bütün değerleri nvarchar(max) veri tipinde yükledik ve uygulamanın içinde value casting işlemlerini gerçekleştirdik. Böylece rota çizdirme sorunlarımızın da üstesinden gelmiş olduk

6. Akış Diyagramı



7. Deneysel Sonuçlar

Yazlab2_2

TIP 1

TIP 2

TIP 3

En Fazla Yolcu Taşıyan 5 Gün ve Toplam Yolcu Sayıları

GÖNDER

Tarih	YolcuSayısı
2020-12-11	79963
2020-12-15	77965
2020-12-10	76960
2020-12-18	75984
2020-12-04	74543

Yazlab2_2

TIP 1

TIP 2

TIP 3

İki Tarih Arasında Seyahat Edilen En Az Mesafeli 5 Yolculuk

2020-12-08

1.Tarihi Yazınız

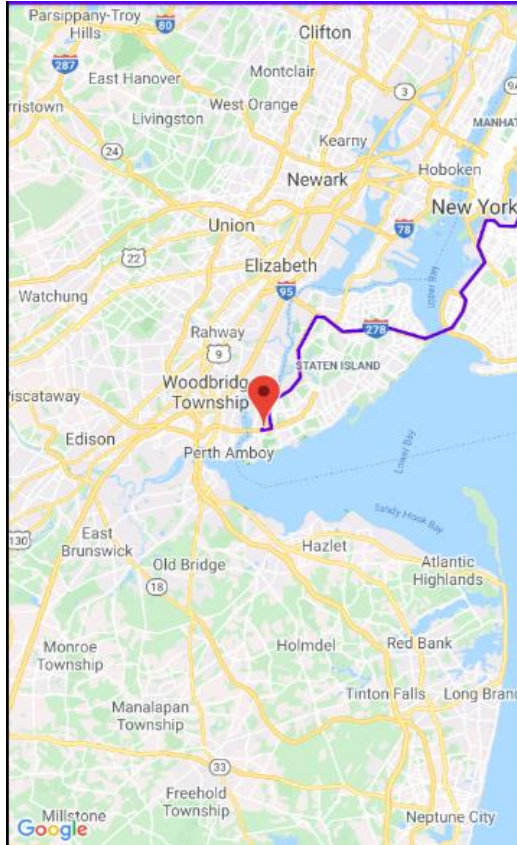
2020-12-14

2.Tarihi Yazınız

GÖNDER

Format: 2020-12-10

Pickup_date	Dropoff_date	Distance	PULocation	DOLocation	Amount
2020-12-08 07:43:54	2020-12-08 07:44:47	0.01	170	170	68.0
2020-12-08 00:33:55	2020-12-08 00:34:16	0.01	75	75	38.0
2020-12-08 05:34:46	2020-12-08 05:34:49	0.01	140	140	-38.0
2020-12-08 05:34:46	2020-12-08 05:34:49	0.01	140	140	38.0
2020-12-08 06:55:34	2020-12-08 06:55:41	0.01	151	238	396.0



8. Kaynakça

- <https://www.youtube.com/watch?v=TY-2Cx4IW9A&list=WL&index=40>
- <https://www.youtube.com/watch?v=fZIZ81fn7b4&list=WL&index=40>
- <https://www.youtube.com/watch?v=58AxNh2cWRU&list=WL&index=39>
- <https://www.youtube.com/watch?v=UnpbVqBSsf4&list=WL&index=37&t=451s>
- <https://introcs.cs.princeton.edu/java/44st/Location.java.html>
- <https://stackoverflow.com/questions/21646019/errors-in-sectionspageradapter-class-when-referring-to-specific-fragment>
- <https://stackoverflow.com/questions/10609268/what-is-the-difference-between-fragment-and-fragmentactivity>
- <https://medium.com/gokhanyavas/javada-dizi-kullanımı-5365ec363409>
- <https://www.youtube.com/watch?v=NwW2QTPArac>
- <https://www.youtube.com/watch?v=vmaVJazDcNc>
- <https://www.youtube.com/watch?v=TSFITR7D9Wo>
- <https://www.youtube.com/watch?v=VR8RKM9LTyA>
- <https://www.udemy.com/course/tam-kapsaml-android-programlama-egitimi/learn/lecture/11786118?start=15#overview>
- <https://www.youtube.com/watch?v=WXXhmWRPEtY>