MOBİL SORGULAR

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ
YAZILIM LABORATUVARI PROJESİ
YASİN ÖMER KARA - FEYZA DEMİREL
180201077 – 190201110
y.omerkara1136@gmail.com feyzahae@gmail.com

ÖZET

Bizden bulut bilişim ve google map api kullanarak android platformunda bir uygulama geliştirmemiz istendi. Bu uygulama içerisinde TLC The New York City Taxi verilerinin Aralık 2020 verilerini kullanarak sorgular yapmamız istendi. Bu verileri sınırlandırarak öncelikle 15 gün ve +1000 veri olacak şekile getirdik ve Firebase'e yükledik.

Projede 3 farklı tipte sorgu vardı ve bu üç sorgunun da kendi içinde üç sorgusu vardı her sorgudan birini kullanmamız istendi sorgular şu şekildeydi:

Tip 1

• En fazla yolcu taşınan 5 günü ve toplam yolcu sayılarını listeleyiniz.

• Belirli mesafenin altında en çok seyahat yapılan günü ve seyahat uzunluğunu bulunuz (mesafe seçilebilmeli).

• En uzun mesafeli 5 yolculuktaki gün ve mesafeleri listeleyiniz.

Tip 2

• İki tarih arasında belirli bir lokasyondan hareket eden araç sayısı kaçtır? (tarihler ve lokasyon seçilebilmeli)

• Günlük seyahat başına düşen ortalama alınan ücretlere göre; en az ücret alınan iki tarih arasındaki günlük alınan ortalama ücretleri listeleyiniz.

• İki tarih arasında seyahat edilen en az mesafeli 5 yolculuk hangisidir? (tarihler seçilebilmeli)

Tip 3

• Belirli bir günde en uzun seyahatin harita üstünde yolunu çiziniz (Gün seçilebilmeli). Başlangıç ve varış konumları lokasyonun merkezi kabul edip mesafeye göre bir yol bulunmalıdır.

• Belirli bir günde aynı konumdan hareket eden araçların rasgele 5'inin yolunu çiziniz (Gün ve konum seçilebilmeli). Başlangıç ve varış konumları lokasyonun merkezi kabul edip mesafeye göre bir yol bulunmalıdır.

• En az 3 yolcunun bulunduğu seyahatlerden en kısa mesafeli ve en uzun mesafeli yolu çiziniz. Başlangıç ve varış konumları lokasyonun merkezi kabul edip mesafeye göre bir yol bulunmalıdır.

Tip 3 sorgusunda yolu haritada göstermek için Google Map Api kullanmamız gerekiyordu.

1.GİRİŞ

Projeyi Kotlin programlama dili kullanarak yazmaya karar verdik. Ve yaptığımız araştırmalar sonucunda bulut ortamı olarak da Firebase kullanmanın bizim için en uygun ortam olduğuna karar verdik.

Verilerimizi düzenleyerek Firebase'de istediğimiz veri ortamını oluşturduk.

Daha sonra android uygulamamızın tasarımını yaptık.

Bu aşamadan sonra hangi sorguları yapacağımıza karar verdik.

Tip 1 için;

En fazla yolcu taşınan 5 günü ve toplam yolcu sayılarını listeleyiniz.

Tip 2 için;

İki tarih arasında belirli bir lokasyondan hareket eden araç sayısı kaçtır? (tarihler ve lokasyon seçilebilmeli)

İki tarih arasında seyahat edilen en az mesafeli 5 yolculuk hangisidir? (tarihler seçilebilmeli)

Seçtiğimiz sorgular bu şekildeydi.

Araştırmalar yaparak Firebase'i nasıl projemize bağlayacağımızı öğrendik ve projemiz ile Firebase Database'imiz arasında bir bağlantı kurduk.

Daha sonra Firebase Database'inden nasıl veri çekeceğimize dair araştırmalar yaptık.

Verileri çekmeyi başardıktan sonra ise bizden istenilenlere göre kodumuzu uyarlamaya başladık.

2. YÖNTEM

İlk olarak sorgulara ulaşabileceğimiz bir anasayfa hazırladık. Burada onClick() fonskiyonları kullanarak butonlar yardımı ile kullanıcıyı anasayfadan istediği sorguya gitmesi için yönlendirdik.(Şekil A.1)

Şekil A.1:

Tip 1 sorgusu için veri tabanından verileri DataSnapshot ile aldık.

Tüm verileri dönerek verilerin içerisindeki "tpep_pickup_datetime" ve "passenger_count" altındaki verileri aldık.

Burada aldığımız datetime verisini String olarak kullanabiliyorduk ama yolcu sayısı verisini Integer'a çevirmemiz gerekiyordu.

IntorNull() fonksiyonunu kullanarak bu işlemi gerçekleştirdik.

Aldığımız tarih verilerini 2020-12-01'den 2020-12-15 tarihine kadar bir Hashmap'e atadık ve her bir gün için gelen yolcu sayısı bilgilerini saydırarak Hashmap'imize ikinci değer olarak ekledik. Haspmap'i List'e çevirerek sıraladık ve en büyük 5 veriyi uygulamamızda bastırdık. (Şekil A.2)

Şekil A.2:

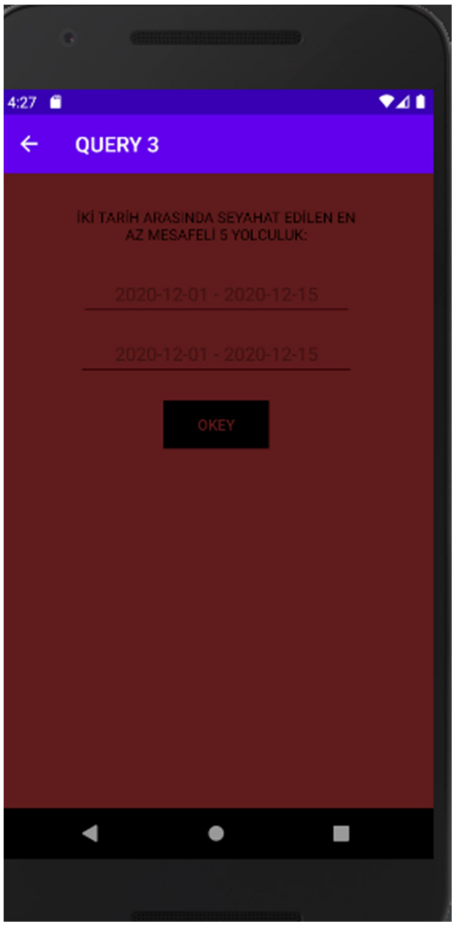
Tip 2 sorgusunun ilk maddesi için ise kullanıcıdan iki farklı tarih ve lokasyon bilgisi aldık. Daha sonra bu bilgileri kullanarak database içerisinde öncelikle tarihleri aratarak. İki tarih arasındaki verileri filtreledik. Bundan sonra ise bu iki tarih arasındaki veriler arasında kullanıcının girdiği lokasyon verisini arattık eğer veri yer alıyorsa araç_sayisi değişkenimizi arttırdık. Ve bu veriyi ekrana yazdırdık. (Şekil A.3)

Şekil A.3:

Okey butonuna tıklandığında kullanıcıya girdiği iki tarih arasında girdiği lokasyondan kalkan toplam araç sayısını gösterdik.

Tip 2 sorgusunun diğer maddesinde ise kullanıcının girdiği iki tarih verisini aldık. Daha sonra aldığımız bu veriyi database içerisinde sınırlandırarak bu iki tarih arasındaki "trip_distance" altındaki verileri alarak tarih verisiyle birlikte bir HashMap<String,Double> tipinde bir hashmap oluşturarak aktardık. HashMap'i List'e dönüştürerek verileri value'a göre sıraladık ve ilk 5 veriyi yazdırdık. (Şekil A.4)

Şekil A.4:



Kullanıcı Okey butonuna bastığında ona istediği tarih aralığındaki en az mesafeli 5 yolculuğu listeleyerek gösterdik.

3. DENEYSEL SONUÇLAR

Firebase Database'den veri çekerken en sorun yaşadığımız konu veriyi istediğimiz formata çevirme konusuydu. String'e çevirirken .toString() fonksiyonu kullanarak kolayca çevirebiliyorduk ama Integer'a çevireceğimiz zaman .toInt() yaptığımızda uygulamamız çalışmıyordu o yüzden Kotlin yapısını detaylı olarak inceleyerek Integer'a çevirme işlemini örnek olarak şu şekilde yaptık:

```
var int1: Int? = passenger.toIntOrNull()
if (int1 != null) {
    a1+=int1
}
```

Bu şekilde veriyi kontrollü bir şekilde almış olduk ve programımız da kolaylıkla çalıştı.

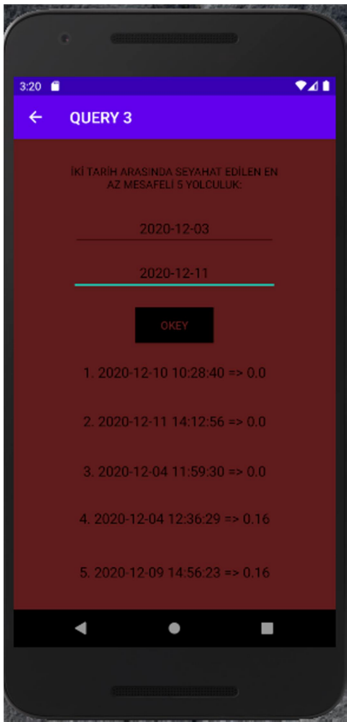
Firebase'den veri çekerken eğer o an Database üzerinde bir değişiklik olursa verinin hemen yansıması için onDataChange yapısını kullandık.

Örnek yapı şu şekilde:

```
var database =FirebaseDatabase.getInstance().reference
var getdata = object: ValueEventListener{
    override fun onDataChange(snapshot: DataSnapshot) {
        for (i in snapshot.children){
            var date = i.child( path: "tpep_pickup_datetime").getValue().toString()
            var passenger = i.child( path: "passenger_count").getValue().toString()
            if((date.contains( other: "2020-12-01", ignoreCase = true))==true){
                var int1: Int? = passenger.toIntOrNull()
```

Tip 2 sorgusu için kullanıcı verileri girdiğinde örnek bir çıktı örneği:

Tip 3 sorgusu için kullanıcı verileri girdiğinde örnek bir çıktı örneği:



4. KULLANDIĞIMIZ KÜTÜPHANELER

Kullandığımız kütüphanelerden bazıları şu şekilde:

- Database Error
- DataSnapshot
- FirebaseDatabase
- ValueEventListener
- View
- Bundle
- AppCombatActivity

5. KAYNAKÇA

[1].Web Site

https://firebase.google.com/docs/database/android/start

[2].Web Site

https://firebase.google.com/docs/database/android/read-and-write

[3].Web Site

https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/hash-map-of.html

[4].Web Site

https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.text/to-int-or-null.html

[5].Web Site

https://medium.com/firebase-tips-tricks/how-to-read-data-from-firebase-realtime-database-using-get-269ef3e179c5

6. AKIŞ DİYAGRAMI