

KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM LAB. II
2021-2022 BAHAR DÖNEMİ
PROJE II

180201018-Ayşenur Akpınar

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

aysenurakpinar98@gmail.com

190201100-Rukiye Canlı

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

rukiyecanli435@gmail.com

1.ÖZET

Bu projede bizden Kocaeli'nin ilçelerinden Kocaeli Üniversitesi'nce gelen servis araçları için yolcu ve rota planlaması yapmamız beklenmektedir. Yolculardan talep alınarak işletilen bu sistemde yolcular için dinamik olarak rota planlamaları yapılacaktır. Belirli araç ve durak sayısı bilgilerinin, duraklardaki yolcu sayısı bilgisi ile birleştirilerek en optimum rotaların oluşturulması hedeflenmektedir. Bunu yaparken araçların kiralama maliyeti ve yakıt tüketiminin de hesaplanarak yol-maliyet optimizasyonu da yapılması istenmektedir. Projenin mobil platform üzerinde gerçekleştirilmesi şarttır.

2.GİRİŞ

Proje Android Studio geliştirme ortamında Java programlama dili ile yazılmıştır. Uygulamada local olarak veritabanı ihtiyacını karşılama için SQLite kullanılmıştır. SQLite bir panele ya da sunucu tarafına ihtiyaç duymadan çalışan bir veritabanıdır. Harita için Google Maps API'den faydalanılmıştır.

Projenin isterleri detaylı olarak aşağıdaki gibidir: Mobil uygulamada yönetim ve kullanıcı paneli bulunmalıdır. Kullanıcı panelinde yolculuk yapacak kişinin bineceği durak seçebilmesi gerekir. Yönetim paneli aracılığıyla yeni durak bilgisi eklenebilmelidir. Yönetici panelinde rota planması yapılmalıdır. Rota planlaması yapıldıktan sonra kullanıcılara bineceği aracın güzergah bilgileri gösterilmelidir. Farklı araçların güzergah bilgilerine erişimi engellenmelidir. Tüm araçların rota bilgileri admin panelinde harita üstünde görüntülenmelidir. Güzergah yol çizdirme algoritması olarak planlanmalıdır. İki nokta arasındaki kusucu çizim kabul edilmeyecektir. Ayrıca hazır uygulamaların (google harita, yandex vs.) güzergâh çizimleri kabul değildir.

Yönetici panelinde kullanıcı paneli dışında gelen talepler için duraklardan binecek yolcu sayısı belirlenebilmelidir. Yönetici yolcuları ve bilgilerini sistemden girebilmelidir. Ayrıca yönetici her bir araç için oluşturulan rotaların maliyetlerini, toplam maliyetleri, her servis aracı için oluşturulan rotaları ve her araç içerisindeki

kullanıcıları görüntüleyebilmelidir. Ek olarak yönetici bütün olası senaryolar için özet tablo/ grafik oluşturabilmelidir. Kocaeli’deki ilçe merkezlerinin konumları durak olarak alınmalıdır. Uygulama içinde durak bilgisi harita üstünde gösterilmelidir.

Projede eklenen duraklara göre yönetici rota hesaplatılabilir. Bu rotayı kullanıcı da yönetici de görüntüleyebilir. Ancak belirli araç ve durak sayısı bilgilerinin, duraklardaki yolcu sayısı bilgisi ile birleştirilerek en optimum rotaların oluşturulması istenmesine rağmen problemin zorluğu sebebiyle gerçekleştirilememiştir. Projenin bu kısımları eksiktir.

3.YÖNTEM

Projemizi gerçekleştirmek için ilk olarak Android Studio ortamında GoogleMaps API'nin nasıl kullanılabileceğine dair araştırmalar yaptık. Araştırmalardan sonra "MapsActivity" sınıfını oluşturduk. Burada haritayı ekranda gösterme, marker oluşturma, iki lokasyon arası rotasyon çizdirme gibi hazır olan fonksiyonları - daha sonrasında bizim projemize uyarlamak üzere- yapıştırdık.

Harita işlemlerinden sonra projemizin "Sign up" sayfasını tasarladık. Bu sayfa aracılığıyla kullanıcı veya admin veri tabanına kaydedilir. Eğer kayıt olan kişi tipini "yönetici" olarak belirtirse veri tabanına yönetici olarak; "kullanıcı" olarak belirtirse veri tabanına "kullanıcı" olarak kaydedilir. Kişiler kayıt olduktan sonra "Sign in" sayfasına yönlendirilir. Bu sayfada kişi kullanıcı adı ve şifre bilgilerini girer ve tipine göre

"UserActivity" veya "AdminActivity" sayfalarına yönlendirilir.

UserActivity sayfasında kullanıcı durak listesinden bir durak seçer. Secilen bu durak veri tabanında kayıtlı değilse veri tabanına kaydedilir. Ayrıca veri tabanında durak kişi sayısına "1" değeri atanır. Eğer bu durak daha önceden kayıtlıysa veri tabanına kaydedilmez. Sadece kişi değeri "1" artılır. Kullanıcı durak seçiminden sonra "Çıkış yap" butonuna basarak giriş sayfasına yönlendirilir. Eğer yönetici rotayı hesaplatırsa artık kullanıcı, kullanıcı sayfasındaki "Rotayı görüntüle" butonuyla rotayı görüntüleyebilir.

AdminActivity sayfasında yönetici durak listesinden bir durak seçer. Ayrıca bu duraktaki kişi sayısını textField alanına elle girer. Seçilen durak veri tabanında yoksa kaydedilir. Durağın yanına girilen kişi sayısı da kaydedilir. Durak veri tabanında önceden kayıtlıysa tekrar kayıt edilmez sadece girilen kişi sayısı var olan kişi sayısına eklenir. Yönetici bu işlemlerden sonra “Rotayı hesapla” butonuna basarsa seçilen duraklara göre Nearest Neighbour Heuristic algoritması ile bir rota hesaplanır. Yönetici “Rotaları Görüntüle” butonuna basarsa MapsActivity sınıfı çalıştırılır. Bu sınıfta gelen rota bilgisine göre ekranda harita üzerinde markerlar oluşturulur ve rota çizdirilir.

MapsActivity sayfasına geldiğimizde ilk olarak matrixx fonksiyonunu kullanarak elemanmatris oluşturuyoruz. Matrixx fonksiyonunda, ilk olarak duration veritabanına kaydettiğimiz ilçeleri allselected dizisine atıyoruz.Daha sonra bu diziyi tsp sınıfının ceviri fonksiyonuna gönderip ilçeleri daha önceden tanımladığımız sayılara eşitliyoruz. Geri dönen bu sayı listesinide eleman isimli bir

integer dizisine atıyoruz.Sonra tüm ilçelerin arasındaki uzaklıkları içeren daha önceden tanımladığımız mesafematrixini ve eleman dizisini tsp sınıfının ilkeleman fonksiyonuna gönderiyoruz. Bu fonksiyon ilçelerin son durak olan Kocaeli Üniversitesi ile olan uzaklıklarını karşılaştırıp en uzak olan ilçeyi ilk eleman olarak atayıp diziyi geri gönderiyor. Geri gelen diziyi yenieleman isimli integer dizisine atıyoruz. Bundan sonra yenieleman dizisini ve mesafematrixi'ni yine tsp sınıfının yenimatrix fonksiyonuna gönderiyoruz.

Bu fonksiyon tüm mesafeleri içeren mesafematrixi içinden yenieleman dizisini kullanarak sadece seçili ilçelerin uzaklıklarını içeren yeni bir matriks oluşturuyor.Bu matriksi mesafematrixi2 isimli önceden oluşturduğumuz matrikse eşitliyoruz.Sonra mesafematrixi2'yi ve allselected dizisini tsp sınıfının tsp fonksiyonuna gönderip sezgisel bir en kısa yol algoritması olan nearest neighbour algoritması yardımı ile ilçelerin rotasını çizdiriyoruz ve çizdirdiğimiz rotayı yeniadres3 isimli string dizisine atıyoruz.

Son olarak yeniadres3 dizisini tsp sınıfının koordinatceviri fonksiyonuna atarak rotada olan ilçeleri isimlerinden koordinatlarına çevirip elemanmatrisine eşitliyoruz ve elemanmatrisini geri gönderiyoruz.Daha sonra tüm ilçeler için ilçelerin koordinatlarını kullanıp marker oluşturuyoruz.Sonra elemanmatrisin her iki elemanını for döngüsünün içinde biri origin biri destination olacak şekilde marker atayıp origin ve destination markerları arasını google maps ve execute fonksiyonu yardımı ile teker teker polyline çizdiriyoruz ve ekrana veriyoruz.

4.YALANCI KOD

```
LoginActivity.class{
```

```
Eğer giriş yapan kullanıcının tipi  
"kullanici" ise{
```

```
UserActivity sayfasını aç  
}
```

```
Eğer giriş yapan kullanıcının tipi  
"kullanici" ise{
```

```
AdminActivity sayfasını aç  
}
```

```
}
```

```
UserActivity.class{
```

```
Kullanıcı listeden durak seçer
```

```
secilen durak "selected" isimli String'e ata
```

```
Secilen durağın veri tabanında olup  
olmadığı CheckIsinDB() fonksiyonuyla  
kontrol et
```

```
Eğer bu durak veri tabanında kayıtlı  
değilse{
```

```
insertData() fonksiyonu ile veri tabanına  
ekle
```

```
getCount() fonksiyonuyla durak kişi  
değerine "1" ata
```

```
}
```

```
getRowsAsArrayList() fonksiyonu ile  
girilen tüm durak isimleri bir ArrayListe  
ata
```

```
Kullanıcı "Çıkış yap" butonuna basarsa{
```

```
LoginActivity sayfasını aç  
}
```

```
Kullanıcı "Rotayı görüntüle" butonuna
basarsa{

    MapsActivity.class çalıştır

}

AdminActivity.class{

    Yönetici listeden durak seçer

    secilen durak "selected" isimli String
    değişkene ata

    Yönetici bu durağa eklemek istediği kişi
    sayısını elle girer kişi sayısı "number"
    isimli integer değişkene ata

    Kaydet butonuna basarsa{

        Secilen durağın veri tabanında olup
        olmadığı CheckIsinDB() fonksiyonuyla
        kontrol et

        Eğer bu durak veri tabanında kayıtlı
        değilse{

            insertData() fonksiyonu ile veri tabanına
            ekle

            getCount() fonksiyonuyla durak kişi
            değerine "number" ata

        }

        Kayıtlıysa{

            getCount() fonksiyonuyla durak kişi
            değerine "number" ata

        }

        Yönetici "Rotayı hesapla" butonuna
        basarsa{

            tsp.java class'ından nesne örnekle

            Bu nesne ile tsp() fonksiyonunu çağır

        }

    }

}
```

```
Yönetici "Rotayı görüntüle" butonuna
basarsa{

    MapsActivity.class çalıştır

}

MapsActivity.class{

    onCreate(){

        matrix fonksiyonu kullanarak
        elemanmatris oluştur

        tüm ilçeler için markeroptions oluştur

        ilçelerin koordinatlarını ata

        i=0'dan elemanmatris.length-1'e kadar{

            origine elemanmatris[i]'yi eşitle

            destination'a elemanmatris[i+1]'i eşitle

            yeni downloadtask nesnesi oluştur

            origin ve destination'ı getdirectionsurl
            fonksiyonuna gönder

            sonucu url ye eşitle

            url'yi downloadtask'in execute
            fonksiyonuna gönder

        }

        origine elemanmatris.length-1'e eşitle

        destination'a umuttepe koordinatlarına
        eşitle

        yeni downloadtask nesnesi oluştur

        origin ve destination'ı getdirectionsurl
        fonksiyonuna gönder

        sonucu url ye eşitle

        url'yi downloadtask'in execute
        fonksiyonuna gönder

    }

}
```

```

}
onMapReady(){
yeni harita oluřtur
tüm ilçelerin markerlarını haritaya ekle
}
matrixx(){
yeni DBHelper2 objesi oluřtur
yeni allselected arraylist'i oluřtur
yeni allselected2 arraylist'i oluřtur
yeni tsp objesi oluřtur
mesafematrix'i tanımla
yeni eleman integer dizisi oluřtur
yeni yenieleman integer dizisi oluřtur
yeni mesafematrix2 integer matrixi oluřtur
yeni yeniadres3 string dizisi oluřtur
getrowsasarraylist("durations")'ı
allselected'a eşitle
allselected'ı allselected2'ye eşitle
yeni elemanmatris2 oluřtur
allselected2'yi tspobj'nin ceviri
fonksiyonuna gönder
sonucu eleman'a eşitle
mesafematrix ve elemanı tspobj'nin
ilkeleman fonksiyonuna gönder
sonucu yenieleman'a eşitle
mesafematrix ve yenielemanı tspobj'nin
yenimatrix fonksiyonuna gönder
sonucu mesafematrix2'ye eşitle

```

```

mesafematrixi2 ve allselected2'yi
tspobj'nin tsp fonksiyonuna gönder

sonucu yeniadres3'e eşitle

yeniadres3'ü tspobj'nin koordinatçeviri
fonksiyonuna gönder

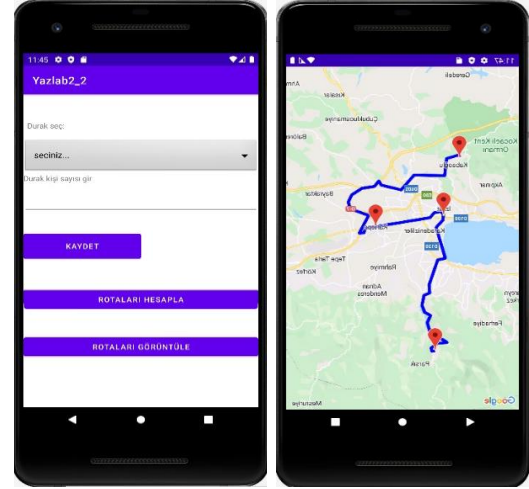
sonucu elemanmatris2'a eşitle

elemanmatris2'yi geri gönder
}
}

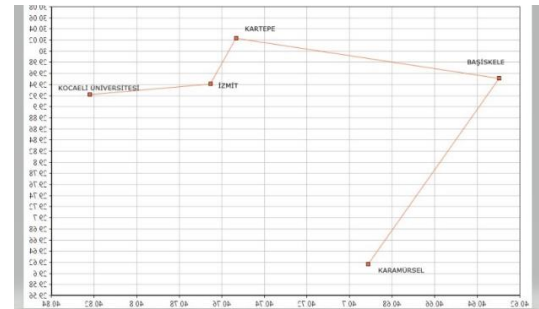
```

5.DENEYSEL SONUÇLAR

Ařağıda yönetici paneli ve yöneticinin butona basması ile ekrana gelen harita ara yüzü paylaşılmıřtır.



Ayrıca haritadaki lokasyon bilgileri grafiklenmiş ve durak-kıřı isimleri veri tabanı tablosu gösterilmiştir.



	duration	kisi ▼ 1
	Filtre	Filtre
1	Izmit	2
2	Gölcük	7
3	Körfez	10
4	Basiskele	15
5	Derince	22

6. SONUÇ

Oluşturduğumuz mobil uygulaması ile kullanıcılar veya yönetici hesap bilgileri aracılığıyla sisteme giriş yapabilir. Sisteme giriş yapabilen kullanıcılar arayüz aracılığıyla durak seçimi yapıp sayfadan çıkış yapabilir. Yöneticinin rota hesabı yaptırmasının ardından rota görüntüleyebilirler. Yönetici arayüz aracılığıyla durak seçimi yapıp bu durakta bekleyecek olan kişi sayısını elle girebilir. Tüm durak girişi işlemlerinden sonra rotayı hesaplayabilir ve rotayı harita üzerinde görüntüleyebilir. Projede bizden servis sayısı-kişi sayısı gibi parametrelere bağlı olarak optimizasyon gerçekleştirmemiz beklenmişti fakat algoritma bizi zorladığı için optimizasyon senaryolarını gerçekleştiremedik.

7. KAYNAKÇA

<https://www.youtube.com/watch?v=8obgNNlj3Eo>

<https://stackoverflow.com/questions/13377361/how-to-create-a-drop-down-list>

<https://stackoverflow.com/questions/4834750/how-to-get-the-selected-item-from-listview?noredirect=1&lq=1>

<https://stackoverflow.com/questions/27003486/printing-all-rows-of-a-sqlite-database-in-android>

<https://stackoverflow.com/questions/48443705/android-prevent-duplicate-insert-data-in-sqlite>

<https://stackoverflow.com/questions/33432487/increment-a-counter-in-database-sqlite>

<https://medium.com/@trientran/android-working-with-google-maps-and-directions-api-44765433f19>

<https://developers.google.com/maps/documentation/android-sdk/start#enable-api-sdk>

<https://www.youtube.com/watch?v=wRDLjUK8nyU>

<https://medium.com/@trientran/android-working-with-google-maps-and-directions-api-44765433f19>

<https://levelup.gitconnected.com/a-nearest-neighbor-solution-in-go-to-the-traveling-salesman-problem-d4d56125b571>

<https://medium.com/opex-analytics/heuristic-algorithms-for-the-traveling-salesman-problem-6a53d8143584>

<https://developers.google.com/maps/documentation/android-sdk/map-with-marker>

<https://developers.google.com/maps/documentation/android-sdk/polygon-tutorial>

<https://stackoverflow.com/questions/7387455/android-sqlite-how-to-retrieve-specific-data-from-particular-column>

