

# Yazılım Laboratuvarı Raporu

## KARGO DAĞITIM SİSTEMİ

Berke  
YILDIZ  
Kocaeli  
Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği  
[190202049@kocaeli.edu.tr](mailto:190202049@kocaeli.edu.tr)

Çağrı  
ŞENTÜRK  
Kocaeli  
Üniversitesi  
Mühendislik  
Fakültesi  
Bilgisayar Mühendisliği  
[190202055@kocaeli.edu.tr](mailto:190202055@kocaeli.edu.tr)

### Özet-

**Veri tabanına kullanıcı kaydı yapılması ve bukayıtlar sayesinde kullanıcı adı ve şifre kontrolü yapılarak sisteme giriş yapılarak kargo dağıtım sisteminin olduğu kısma geçilmektedir. Bu kısımda, kargolar konumları harita üzerinde tıklanarak veya adres girilerek ekleyebilme ve bu verilerin bulut veri tabanında tutulması, bulunan kargoların teslim durumlarının liste olarak**

**gösterildiği, kargo silme ve teslimat bilgilerinin girilme işlemleri gerçekleştirilir. Sonrasında girilen kargoların konumlarının ve kargocunun konumunun işaretlendiği, sadece haritanın bulunduğu arayüz ekrana gelir. Bu arayüzde, bulunan kargoların teslim edilebileceği en kısa yol rota olarak çizilir. Teslim edilen kargonun silinir ve teslim edildi bilgisi teslimat durumlarında güncellenir. Bu arayüz her kargo eklenme, kargo silinme ve kargo güncellenme durumlarında yenilenir ve yeni duruma göre en kısa yol çizilir.**

### Giriş

Program çalıştırıldığında ilk olarak Giriş sayfası açılmaktadır. Giriş sayfamızda 3 adet buton yer almaktadır bunlar: Giriş Yap, Kaydol, Şifre Değiştir Paneline Git. Kullanıcı girişi yapabilmek için kullanıcı adı ve şifrenin doğru girilmesi gerekmektedir. Veritabanında kontrol edilir. Yanlış girilir ise uyarı vermektedir. Kullanıcı sisteme kayıtlı değil ise Kaydol butonuna basar ve kayıt olma penceresi açılır. Buradaki bilgiler eksiksiz bir şekilde doldurulmak zorundadır.

Doldurulmadığı takdirde boş bırakmayınız hatası vermektedir. Kayıt olduktan sonra kullanıcıya ait bilgiler veri tabanına eklenir. Kullanıcı şifresini ister ise şifre değiştir butonuna tıklar. Şifre değiştirme paneli açılır. Burada kullanıcı adı, eski şifre ve yeni şifre kullanıcıdan istenir. Kullanıcı ve eski şifrenin doğruluğu veritabanından kontrol edilir, doğru ise şifre değiştirilir. Kullanıcı giriş yaptığında 2 form açılır. Bu formlardan birisi ana sayfa diğeri haritadır. Ana sayfa formumuzda kargo ekleme, silme, güncelleme işlemleri yapılabilmektedir. Bunlara ek olarak kullanıcı kargonun adresini isterse harita üzerinden seçerek isterse adres girerek konum belirleyebilir. Harita formumuzda ise kargocumuzun bulunduğu yeri ve teslim edilmemiş kargoları markerler ile görebiliriz. Buna ek olarak kargoları en kısa şekilde ulaştırabileceği rotaları çizdirdik. Kullandığımız algoritmanın detayları yöntemler kısmında mevcuttur. Kargo teslim et seçeneğine tıklandığında

kargocumuz rotamıza göre kargoları teslim eder ve veri tabanında teslim durumu güncellenir.

## Yöntemler

Bu başlık altında giriş kısmında söylenen işlemlerin hangi metotlarla yapıldığı açıklanacaktır.

### A. VERİTABANI

Programımızda 2 adet tablo bulunmaktadır. Bunlar kullanıcı ve kargolar tablolarıdır.

Kullanıcı tablomuzda kullanıcıya ait bilgileri tuttuk bunlar: kullanıcı\_id(int), kullanıcı\_adi(varchar(40)), kullanıcı\_email(varchar(40)), kullanıcı\_sifre(varchar(40)), kullanıcı\_id'yi PRIMARY KEY tayin edilerek eklenecek kayıtların birbirine karışmamasını aynı ID'ye sahip yeni bir kaydın eklenmemesi sağlandı.

kargolar tablomuzda kullanıcıya ait bilgileri tuttuk bunlar: kargo\_id(int), kargo\_konumx(double), kargo\_konumy(double), kargo\_teslim(varchar(10)), kargo\_sahibi(varchar(15)) kargo\_id'yi PRIMARY KEY tayin edilerek eklenecek kargoların birbirine karışmamasını aynı ID'ye sahip yeni bir kargonun

eklenmemesi sađlandı.

## B. ARAYÜZ PROGRAMLAMA

Projemizi çalıştırdığımızda ilk olarak Giriş olarak tanımladığımız bir form oluşturduk ve bunun üzerine Giriş Yap, Kayıt Ol ve Şifre Değıştir butonu ekledik. Kullanıcı adı ve şifre girildiğinde veri tabanımız ile kontrol ederek doğruysa AnaSayfa ve Harita adıyla oluşturduğumuz forma yönlendirilir. AnaSayfa'da veri tabanından çektiğimiz kargolar grupbox içersine DataGridView kullanarak yansıtılır. Kullanıcı burada isterse seçtiğı kargoları silebilir veya güncelleyebilir. İsterse checkBox'ı aktif ederek. Harita üzerinden tıklayarak konum seçebilir. İsterse textBox'a Adrees girerek kargo eklemesini yapabilir. Ekleme yapıldığında harita formumuzda en kısa yol eklenen kargo dahil edilerek tekrar hesaplanır ve haritaya çizilir. Harita formumuzda ise teslim edilmeyen kargolar veritabanından alınarak en kısa yol algoritması çalıştırılır. Harita üzerinde kargolara marker atılır ve en kısa yol rotası çizilir. Kargo teslim et butonuna tıklandığında. Kargocumuz hareket ederek kargo teslim edilir ve veritabanında kargo teslim bilgileri güncellenir.

## C. EN KISA YOL ALGORİTMASI

```
private List<string> YolHazirla()
{
    Random rand = new Random();
    List<string> yollar = new List<string>();
    List<int> randomList = new List<int>();
    string temp = "";
    int MyNumber = 0;
    int faktoriyel = 1;
    for (int i = 0; i < kargos.Count + 1; i++)
        faktoriyel = faktoriyel * (i + 1);
    while (yollar.Count != faktoriyel / (kargos.Count + 1))
    {
        randomList.Add(0);
        while (randomList.Count != kargos.Count + 1)
        {
            MyNumber = rand.Next(1, kargos.Count + 1);
            if (!randomList.Contains(MyNumber))
            {
                randomList.Add(MyNumber);
            }
        }
        for (int i = 0; i < kargos.Count + 1; i++)
        {
            temp = temp + Convert.ToString(randomList[i]);
        }
        randomList.Clear();
        if (!yollar.Contains(temp))
        {
            yollar.Add(temp);
            temp = "";
        }
    }
    return yollar;
}
```

Bu algoritmamızın çalışma mantığı şu şekildedir:

İlk olarak kaç kargomuz bulunuyorsa bu kargolar için kargo sayımızın faktöriyel kadar farklı yol olacağından kargo sayımızın faktöriyelini buluyoruz. Sonrasında bizim başlangıç noktamız belirli olduğundan kargo sayımızın faktöriyelini kargo sayımıza bölüyoruz. Bu sayede başlangıç noktamızın belirli olduğu farklı yol sayısına ulaşıyoruz. Bir while döngüsü kullanarak ve bu döngüde her seferinde kontrol ederek rastgele ve farklı sayıları randomList adında bir listeye ekleyerek bu verileri tutuyoruz. While döngümüzü bu listenin içindeki elemanların sayısı bizim kargo sayımıza eşit olana kadar devam ettiriyoruz.

Bu işlem sonucunda rastgele ve her biri farklı rakam olan kargo sayımızın uzunluğunda bir sayı dizisi elde ediyoruz. Bu elde ettiğimiz sayı dizisini de yollar adında bir liste oluşturarak bu listenin içerisinde bu sayı dizisi mevcut mu diye kontrol ediyoruz. Eğer bu sayı dizimiz mevcut değilse bu sayı dizimizi yollar listemize ekliyoruz. Bu işlemi de bir while döngüsü içinde yapıyoruz ve bu while döngümüzü yollar listemizin eleman sayısının kargo sayımızla eşit olduğu duruma kadar devam ettiriyoruz.

Yukarıda anlatığımız iç içe while döngülerimiz ve kontrollerimiz sayesinde her biri farklı olan ve içerisinde aynı rakamı bulundurmayan olasılıklarımızı elde ediyoruz. Sonrasında yollar adındaki listemizi return kullanarak yolluyoruz.

```
private int[] EnKisaYolBul()
{
    List<string> yolSecenekleri = YolHazirla();
    int[,] yollar = new int[yolSecenekleri.Count, kargos.Count + 1];
    int[] enKisaRota = new int[kargos.Count + 1];
    for (int i = 0; i < yolSecenekleri.Count; i++)
    {
        for (int j = 0; j < kargos.Count + 1; j++)
        {
            yollar[i, j] = Convert.ToInt32(yolSecenekleri[i].Substring(j, 1));
        }
    }
    double toplamYol = 0;
    double enKisaYol = 0;
    for (int i = 0; i < yolSecenekleri.Count; i++)
    {
        for (int j = 0; j < kargos.Count; j++)
        {
            if (j == 0)
            {
                toplamYol += MesafeBul(kargocu.point, kargos[yollar[i, j + 1] - 1].point);
            }
            else
            {
                toplamYol += MesafeBul(kargos[yollar[i, j] - 1].point, kargos[yollar[i, j + 1] - 1].point);
            }
        }
        if (i == 0)
        {
            enKisaYol = toplamYol;
        }
        if (toplamYol < enKisaYol)
        {
            for (int k = 0; k < kargos.Count + 1; k++)
            {
                enKisaRota[k] = yollar[i, k];
            }
            enKisaYol = toplamYol;
        }
    }
    toplamYol = 0;
    return enKisaRota;
}
```

### EnKisaYolBul()

Bu fonksiyonumuzun çalışma mantığı şu şekildedir:

**YolHazirla()** fonksiyonumuz sayesinde farklı bütün yolların durumunu bulmuştuk.

Bu durumları fonksiyonumuz içerisinde kullanabilmek için bir listeye çağırıyoruz. Sonrasında bu listedeki her bir sayı dizinine göre kargolarımız arasında sayı dizinimizin sırası ile aralarındaki mesafeyi buluyoruz ve toplamYol değişkenimize ekliyoruz. Ana döngümüzün her adımının sonunda toplamYol değişkenimizin tuttuğu değer enKisaYol değişkenimizin tuttuğu değerden küçük olup olmadığını kontrol ediyoruz. Eğer küçük ise enKisaYol değişkenimizi toplamYol değişkenimizle eşitliyoruz.

Ana döngümüz bittiğinde sonuç olarak enKisaYol değişkenimiz tüm olasılıkları deneyip bulduğumuz ve olabilecek en kısa yola eşit olur.

Böylelikle bu iki fonksiyonu kullanarak en kısa yolumuzu bulmuş oluyoruz.

## III. GELİŞTİRME ORTAMI VE

## KULLANILAN DİL

Projeyi C# programlama dilinde Windows işletim sistemi üzerinde gerçekleştirirken, geliştirme ortamı olarak Visual Studio 2019 idesini kullandık. Veritabanı işlemlerimizi Google Cloud üzerinden gerçekleştirdik. Harita apimizi de Google Map apisini kullandık.

## IV. PROGRAMIN GENEL YAPISI VE TASARIMI

Kullanıcı Adı

Şifre

GİRİŞ

Hesabınız yok mu? Yeni bir hesap oluşturun?

Şifrenizi mi değiştirmek istiyorsunuz?

Kocaeli Üniversitesi

Kabaoğlu

Sekbanlı

Sepetçi

Anzlı

75. Yıl ulusal Ormanı

Izmit

Izmit

Karadenizliler

x 40,79938844

y 29,947443008

Kargo Sahibi berke

adres: Ayazma, Üniversite Cd. No:20, 41060 İzmit/Kocaeli, Turkey

Adresi Getir

Kargo Sil

Kargo Ekle

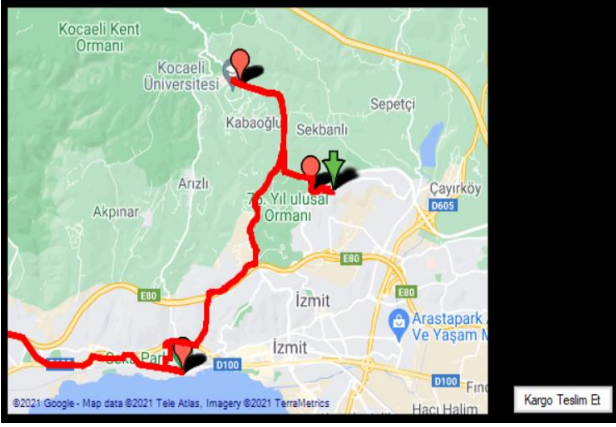
Kargo Güncelle

Kargoları Getir

Tıklayarak Konum Seçme

KARGOLAR

	kargo_id	kargo_konumx	kargo_konu
▶	16	40,353707	27,971405
	17	40,76221103250...	29,9408340...
	19	40,82377982402...	29,9241399...
	43	40,75978	29,905258
	44	40,76718414067...	29,9405336...



## V. DENEYSEL SONUÇLAR

C# Windows form uygulaması ile oluşturduğumuz arayüz ile veritabanımızı bağlayarak dinamik bir yapı kurduk. Bu yapı kurulurken birçok sorgulama hataları ile karşılaştık bunları araştırarak istediğimiz sonucu elde ettik ve bilgilerimiz pekiştirdik. En kısa yol algoritmasını araştırırken birçok yöntem bulduk bunları kendi kodumuza uygulamaya çalışırken birçok hata ile karşılaştık. Sonuç olarak her algoritmadan bizim ihtiyacımız olan kısımlarını uyguladık.

## VI. SONUÇLAR

Projemiz sayesinde bulut veri tabanında çalışmayı, hazır apileri projelerimize entegre etmeyi ve iki form arasında dinamik yapı kurmayı öğrendik. Ayrıca daha önceden bildiğimiz nesneye yönelik programlama ve arayüz oluşturma konusunda tecrübelenmemize olanak sağladı.

## VII. YALANCI KOD

- Başla
- Giriş ekranını göster
  - 1)Kayıt ekranı aç
  - 2)Şifre Değiştirme ekranını aç
  - 3)Giriş yap
- Kayıt ekranına bütün bilgileri gir.
- Kaydol butonuna bas ve bütün bilgileri veri tabanına gönder.
- Şifre değiştirmek için kullanıcı adı eski şifre ve yeni şifre gir.
- Veri tabanında kullanıcı adı eski şifre doğru mu kontrol et.
- Doğruysa şifreyi güncelle, aksi halde uyarı

ver.

- Giriş yapmak için kullanıcı adı ve şifre gir.
- Veri tabanında doğru mu değil mi diye kontrol et.
- Doğruysa uygulamayı aç, aksi halde uyarı ver.
- Ana Sayfa ve Harita formunu aç
- Eklemek istediğin kargonun haritadan veya adres bilgilerini girerek yerini seç
- Kargo sahibi adını gir
- Kargo ekle butonuna tıkla
- Bilgiler veri tabanına kaydedilsin
- En kısa yol algoritması çalışsın harita güncellensin
- Silmek istediğin kargoyu seç
- Kargoyu sil butonuna tıkla
- Veritabanından silinsin
- Güncellemek istediğin kargoyu seç
- Verileri gir kargoyu güncelleme tıkla
- Veritabanında güncellensin
- En kısa yol algoritması çalışsın harita güncellensin
- Harita formu açılsın
- Veri tabanından kargo bilgileri alınsın
- En kısa yol algoritması çalışsın
- En kısa yol bulunsun
- Rota çizilsin
- Kargo teslim et butonuna tıklandığında
- Veri tabanında kargo teslim bilgisi güncellensin
- Teslim edilen kargonun yolu ve markerı silinsin

## VII. KAYNAKÇA

1. C# Form Uygulamaları:  
<https://youtube.com/playlist?list=PLSuhOGv534vTFRq9dFB9Yn7su-4hIc0xg>
2. GMap:  
<https://youtube.com/playlist?list=PLL7nT-mUjVuqlhWVfaNhnpqCZmNcA9e>
3. En Kısa Yol Algoritması:  
<https://www.youtube.com/watch?v=cY4HiiFHO1o&t=885s>