

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü
Yazılım Laboratuvarı I
KARGO DAĞITIM SİSTEMİ

Taha Uz/Mahmut Alper Yılmaz

190202013@kocaeli.edu.tr[Taha]/180202004@kocaeli.edu.tr[Alper]

ÖZET

Yazılım Laboratuvarı I Projesi olarak bizden “Kargo Dağıtım Sistemi” adlı uygulama geliştirmemiz beklenmektedir.

Biz proje için Java, JavaScript,C# ve SQL dillerini kullanmaya çalıştık fakat JavaScript de başarısız olduğumuz için Java, C# ve SQL devam ettik. Netbeans, Visual Studio Code ve Visual Studio 2019 geliştirme ortamında ve Google Cloud sisteminin içerisindeki MySQL veri tabanı sistemini kullanmayı tercih ettik.

Proje dökümanında bizden , akıllı kargo dağıtım sistemi yapan bir masaüstü uygulaması geliştirmemiz beklenmektedir. Proje 2 GUI den oluşup ilk GUI kullanıcının kargo bilgileri alınıp Cloud’daki veri tabanına kayıt ediyoruz ve kurye admin girişi yaparak kargo durumunu ve hareket tarihini değiştirebilmektedir. İkinci GUI de ise kurye kargoları görebileceği ve kendisine en kısa yoldan yol tarifi yapılan arayüze sahip olması sağlanıyor. Bu arayüzde Cloud’daki veri tabanından müşterilerin kargo bilgilerini alıp bu

bilgileri işliyoruz. Daha sonra en kısa yol algoritmasına sokup bu algorithmadan rota oluşturuyoruz. Oluşturduğumuz bu rotayı kurye gösteriyoruz böylelikle az maaliyette kargolar teslim edilmiş oluyor. Eğer kurye teslimatta iken yeni kargo gelirse gelen kargo ile birlikte en kısa rota tekrardan oluşturuluyor.

1.GİRİŞ

Proje için Java, JavaScript,C# ve SQL dillerini kullanmaya çalıştık fakat JavaScript de başarısız olduğumuz için Java, C# ve SQL devam ettik. Netbeans, Visual Studio Code ve Visual Studio 2019 geliştirme ortamında ve Google Cloud sisteminin içerisindeki MySQL veri tabanı sistemini kullanmayı tercih ettik.

Java programlama dili ; Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış açık kodlu , nesneye yönelik , zeminden bağımsız , yüksek verimli ,çok

işlevli , yüksek seviye , adım adım işletilen (yorumlanan-interpreted) bir dildir.

C# ; Microsoft tarafından .NET Teknolojisi için geliştirilen modern bir programlama dilidir. Sözdizimi C-like bir deneyim sunar. Microsoft tarafından geliştirilmiş olsa da ECMA ve ISO standartları altına alınmıştır.

SQL , verileri yönetmek ve tasarlamak için kullanılan bir dildir. SQL , kendisi bir programlama dili olmamasına rağmen birçok kişi tarafından programlama dili olarak bilinir . SQL herhangi bir veri tabanı ortamında kullanılan bir alt dildir .

Netbeans platformu; Oracle tarafından geliştirilen bir java geliştirme ortamıdır (IDE) ve ücretsiz olarak dağıtılmaktadır. Özellikle kullanıcı arayüzü tasarımında sağladığı kolaylıklardan dolayı tercih edilmektedir .

Visual Studio Code platformu : Microsoft tarafından Windows, Linux ve MacOS için geliştirilen bir kaynak kodu düzenleyicisidir. Hata ayıklama, gömülü Git kontrolü, sözdizimi vurgulama, akıllı kod tamamlama, snippet'ler ve kod yeniden yapılandırma desteği içerir.

MySQL , altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı , çok kullanıcı , hızlı ve sağlam bir veri tabanı yönetim sistemidir. UNIX , OS/2 ve Windows platformları için ücretsiz dağıtılmakla birlikte ticari lisans kullanmak isteyenler için de ücretli bir lisans seçeneği de mevcuttur.

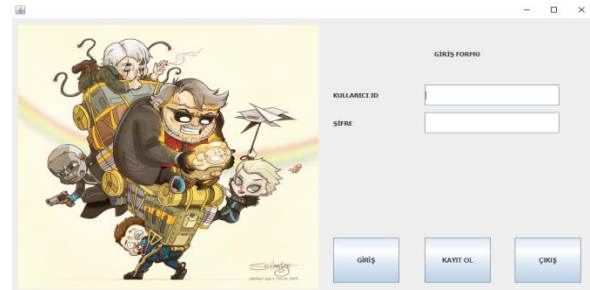
Google Cloud, Google firmasının Google arama motoru ve Youtube gibi sitelerin de kullandığı sunucu altyapı hizmetlerini son kullanıcıya sunduğu bir bulut bilgi platformudur. Basit web sitelerinden kompleks uygulamalara kadar çeşitli tipte yazılım Google Cloud üzerinden geliştirebilir.

2.TEMEL BİLGİLER

Projemizi anlatırken alt başlıklara ayırarak anlatacaz. Bu başlıklar ; Müşteri GUI, Kurye GUI, Kısa Yol Algoritması ve Cloud SQL Bağlantısı olmak üzere 4 alt başlık altında anlatacağız.

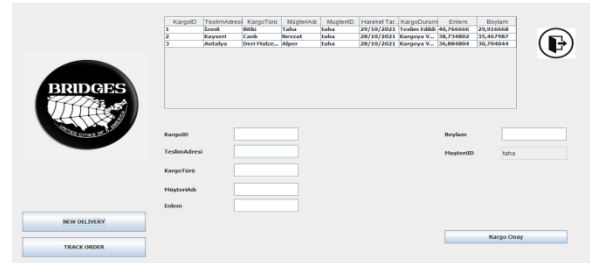
2.1.Müşteri GUI

Kullanıcı ilk olarak Login ekranı ile karşılaşır. Burada “id” ve “şifre” girilerek kullanıcı eğer ilk defa sisteme giriş yapıyorsa “Kayıt Ol” butonuna basarak kullanıcı bilgileri veri tabanına alınıyor ve kullanıcıya özel kargo tablosu veri tabanında oluşturuluyor. Kullanıcı kayıt olduktan sonra giriş yap butonuna basarak veri tabanında girdiği bilgilerin doğruluğu kontrol edilerek sisteme girişini yapıyor.

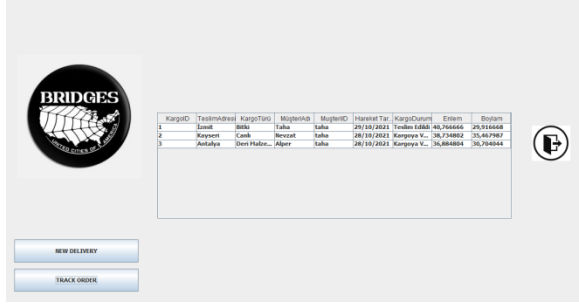


Resim 1 : Giriş Arayüzü

Sistemde eğer müşteri olarak giriş yapmış ise “New Delivery ” ve “Track Order” olmak üzere iki buton ile karşılaşılıyorlar. “New Delivery” butonu kullanıcının kargo bilgilerini sisteme kayıt etmesine yardımcı olur. Girilen bu bilgileri temiz bir arayüzde güncel durumunu takip etmek için “Track Order” butonu ile tablo içerisinde takip edebilir.

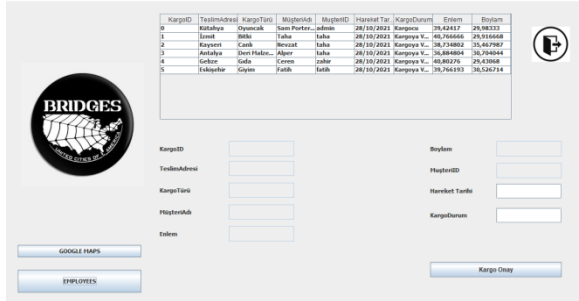


Resim 2 : Kullanıcı Arayüzü “New Delivery” Sistemi

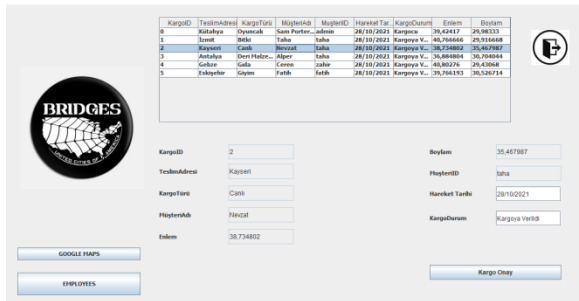


Resim 3 : Kullanıcı Arayüzü “Track Order” Sistemi

Kullanıcı admin olarak sisteme giriş yaparsa sadece bir adet buton ile karşılaşır. “Employees” butonu ile admin yetkisine sahip firma çalışanı sadece kargo hareket tarihi ve kargo durum bilgilerinde değişiklik yapılabilmesi sağlanmıştır. “Google Maps” butonu eğer projeyi daha sonra geliştirmek isteyen geliştiriciler için “html” uzantılı dosyaları Google Maps Api ‘si ile bağlantısı yapılabilir amaç ile kaldırmadık.



Resim 4 : Kullanıcı Arayüzü Admin Girişi



Resim 5 : Kullanıcı Arayüzü Admin Girişi 2

Kullanıcıdan bu bilgileri aldıktan sonra ikinci GUI ekranı olan “Kurye GUI” ekranına geçiş yapılıyor.

2.2.Kurye GUI

Burada Google Map ile çalışarak kargocuya müşterilerin harita üzerinde adreslerini işaretleyerek müşterilere gidilecek en kısa rotayı oluşturuyoruz.



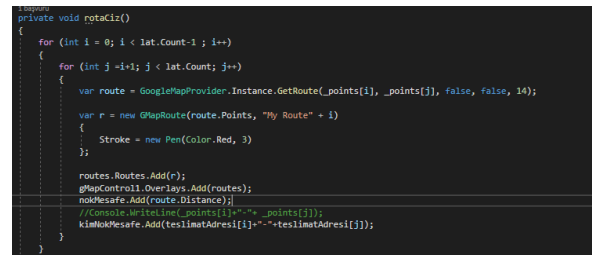
Resim 6 : Kurye Arayüzü

İlk olarak Cloud’daki veri tabanından müşteri bilgilerini çekiyoruz. Bu bilgilerle haritada müşterilerin adreslerini göstererek müşterinin “Adres”, “ID” ve “Name” bilgileri ile kargocuyu bilgilendiriyoruz.



Resim 7 : Haritaya Noktaları Ekleyen Kod Parçası

Bilgilendirmeyi yaptıktan sonra haritaya noktalar arasında yolları arayüze çizdirerek noktalar arası uzaklıkları API yardımıyla hesaplıyoruz.



Resim 8 : Haritaya Yolları Çizdiren Kod Parçası

Daha sonra bu bilgileri “rotaUretici” fonksiyonuna yollayarak en kısa rota

[illegible]

Eğer kargocu yola çıktığında veri tabanına yeni bir kargo girilirse “haritayı güncelle” butonuna tıklayıp rotayı tekrardan oluşturabilir veya önceki noktada teslimat gerçekleştirilirse “teslim edildi” butonuna bastığında noktalar güncellenirken yeni nokta varsa onun kontrolü yapılarak harita güncellenir.

Yaptığımız kısa yol algoritması Traveling Salesman yönteminden yola çıkılarak girilen noktaları permütasyon hesabına sokarak daha sonrasında ise bu çıktıların maliyetlerini hesaplayarak rota belirleme algoritması yazdık. Detaylı anlatmak gerekirse ; ilk olarak "rotaciz() fonkiyonunda API yardımıyla hesaplanan noktalar arasındaki mesafe o noktaların isimleri ve noktalar arasındaki uzaklığı 2 farklı "ArrayList" tipindeki "kimNokMesafe" ile "nokMesafe" adlı değişkenlerde saklanır. "kimNokMesafe" iki noktanın adres bilgisini tutar örneğin "İzmit-Gebze" gibi. "nokMesafe" ise bu noktaların arasındaki uzaklığı saklar.

```
nokMesafe.Add(route.Distance);  
//Console.WriteLine(_points[i]+"-"+_points[j]);  
kimNokMesafe.Add(teslimatAdresi[i]+"-"+teslimatAdresi[j]);
```

4

```
//string yaz;
//Arraylist rota;

double tempRotaCost;

Console.WriteLine("[1]");
foreach (var list in lists)
{
    if (list[0].Equals(testimatAdresi[0]))
    {
        //yaz = null;
        rota = new ArrayList();
        foreach (var l in list)
        {
            //yaz = yaz + " " + l;
            rota.Add(l);
        }
        //Console.WriteLine(yaz);
        //String aaz = null;
        tempRotaCost = 0.0;
        for (int i = 0; i<rota.Count-1; i++)
        {
            for (int j = 0; j < kinMokMesafe.Count; j++)
            {
                if(kinMokMesafe[j].Equals(rota[i])+"-"+rota[i+1]) || kinMokMesafe[j].Equals(rota[i+1] + "-" + rota[i]))
                {
                    tempRotaCost += Convert.ToDouble(nokMesafe[j]);
                }
            }
            //aaz = aaz + " " + rota[i+1].ToString();
        }
        //Console.WriteLine(aaz);
        //Console.WriteLine(tempRotaCost);

        if (tempRotaCost < rotaCost)
        {
            rotaCost = tempRotaCost;
            for (int k = 0; k < rota.Count; k++)
            {
                asilRota[k] = rota[k].ToString();
            }
        }
    }
}
```

Bu rota asıl rota yapılarak “enKisaRotaOlustur” adlı fonksiyonu devreye girerek kuryenin arayüzündeki “nextStation” kısmında kurye bilgilendirilir ve “Teslim Edildi” butonundaki “enKisaRota” adlı listeye rota aktarılarak kuryenin rotasını belirleme işlemi tamamlanır.

```
nextStation.Text = $"STASIUN TESJIMAT | \n Adress : {teslimatAdresi[Convert.ToInt32(enKisaNota[1])]}, \n Customer ID : {musteriId[Convert.ToInt32(enKisaNota[1])]}, " +
    $" \n Customer Name : {musteriAd[Convert.ToInt32(enKisaNota[1])]}";
```

İlk önce Google Cloud 'a üye olduk. Daha sonra SQL veri tabanı oluşturup bu veri tabanını kullanabilmek için API iznini aldık. İzni aldıktan sonra Java'da yapılan kullanıcı arayüzünü Cloud'daki MySQL veri tabanı ile haberleştirdik.

```
String driver = "com.mysql.cj.jdbc.Driver";
String url = "jdbc:mysql://34.135.94.17:3306/BridgesCargo?useSSL=false";
String userName = "root";
String password = "";
Class.forName(driver);

Connection myConn = DriverManager.getConnection(url, userName, password);
```

Resim 13 : Java Cloud SQL Bağlantı Kod Parçası

Böylece kullanıcıya özel tablo oluştururken, kullanıcıdan yeni kargo veya yeni kullanıcı girişi alırken ya da kargocu kargo hareketini güncellerken bu bağlantı ile veri tabanı ile haberleşerek bu işlemleri gerçekleştiriyor.

```
try {
    String driver = "com.mysql.cj.jdbc.Driver";
    String url = "jdbc:mysql://34.135.94.17:3306/BridgeCargo?useSSL=false";
    String userName = "root";
    String password = "";
    Class.forName(driver);

    Connection myConn = DriverManager.getConnection(url, userName, password);

    String sorgu = "INSERT INTO kullanicilar values('"+id+"','"+sifre+"')";
    Statement st = myConn.createStatement();
    int x = st.executeUpdate(sorgu);
    if(x == 0){
        JOptionPane.showMessageDialog(null,"Kullanici zaten var.");
    }else{
        JOptionPane.showMessageDialog(null,"Mesabiniz basariyla olusturuldu.");
        onay=1;
    }
} catch (Exception e) {
    if(e.getMessage().indexOf("Duplicate") != -1){
        JOptionPane.showMessageDialog(null,"Bu kullanicilar aze ve sifresinde bir kullanicilar zaten var.");
    }else{
        JOptionPane.showMessageDialog(null,"Data Base Hatali : "+e.getMessage());
    }
}
}
```

Resim 14 : Kullanıcı Giriş Sorgusu

```

    }

    String driver = "com.mysql.cj.jdbc.Driver";
    String url = "jdbc:mysql://193.100.94.27:3306/SQLInjection?useSSL=false";
    String username = "root";
    String password = "root";
    Class.forName(driver);

    Connection myConn = DriverManager.getConnection(url, username, password);
    String sql = "SELECT * FROM users WHERE username = 'ADMIN' OR NOT NULL,username VARCHAR(40) NOT NULL,password VARCHAR(40) NOT NULL,email VARCHAR(40) NOT NULL";
    Statement stmt = myConn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);

    Search(searching);

    JspPage.showHomePageUsing(url,"Index Base Data : "+rs.getDatabase());
}

```

Resim 15 : Kullanıcıya Özel Tablo Oluşturma

[illegible]

Resim 16 : Yeni Kargo Ekleme Sorgusu

[illegible]

```
1 bayvuru
private void MySqlRdy()
{
    string cs = @"server=34.135.94.17;userid=root;password=;database=BridgesCargo";

    var con = new MySqlConnection(cs);
    con.Open();

    string sql = "SELECT * FROM delivery";
    var cmd = new MySqlCommand(sql, con);


    MySqlDataReader rdr = cmd.ExecuteReader();


    while (rdr.Read())
    {
        if (!rdr.GetString(6).Equals("Teslim Edildi"))
        {
            kargoId.Add(rdr.GetString(0));
            teslimatAdresi.Add(rdr.GetString(1));
            musterId.Add(rdr.GetString(3));
            musterIId.Add(rdr.GetString(4));
            lat.Add(rdr.GetString(7));
            lng.Add(rdr.GetString(8));
        }
    }

    noktalar = new String[teslimatAdresi.Count];
}
}
```

Resim 18 : C# MySQL Bağlantısı

3.DENEYSEL SONUÇLAR





	KargoID	TeslimAdres	KargoTari	MailetAd	MailetID	Hareket Tar.	KargoDurum	Enlem	Boylam
1	Zamel	Sakla	Taha	Sakla		27/10/2021	Teslim Edildi	38,746666	23,625668
2	Kayseri	Lamb	Revard	Sakla		28/10/2021	Karguya Y.	38,746883	25,642987
3	Ardahan	Devri Habin	Alper	Sakla		28/10/2021	Karguya Y.	38,884884	38,704664

KargoID

TeslimAdresi

KargoTari

HayranAd

Enlem

Boylam

HayranID

Isht

[NEW DELIVERY](#)

[TRACK ORDER](#)

[Kargo Onay](#)

KargoID	TestisAdresi	KargoTari	MajelisAd	MajelisID	Hariwal Tar	KargoDurum	Enam	Bosman
1	Kutubay	Diyarbakir	Sams Pertanto	adinda	26/10/2021	Kargonya V...	26.04.517	26.04.517
2	Suandi	edike	Taha	taha	26/10/2021	Kargonya V...	26.05.666	26.05.666
3	Kargaya	Cakir	Bismail	taha	26/10/2021	Kargonya V...	26.07.882	26.07.882
4	Arindah	Dereh Indone	Alfery	taha	26/10/2021	Kargonya V...	26.08.084	26.08.084
5	Kelcaro	Soda	Cernie	jukur	26/10/2021	Kargonya V...	26.08.276	26.08.276
6	Tekayalar	Seymen	Fatih	Fatih	26/10/2021	Kargonya V...	26.08.513	26.08.513

KargoID

2

TestisAdresi

Kajipari

KargoTari

Cakir

HiperKas

Nicral

Enam

36.74852

Bosman

35.467567

HiperID

taha

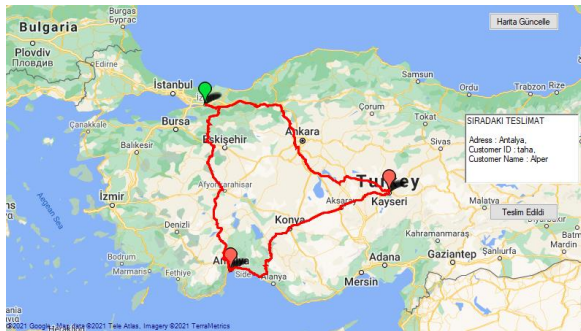
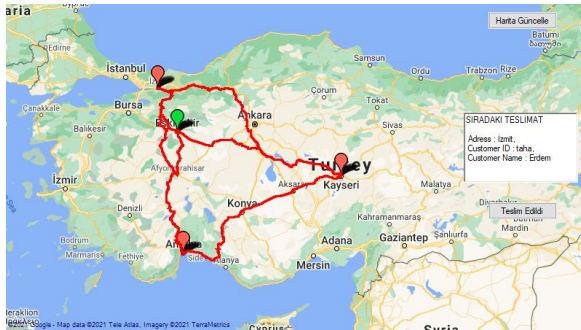
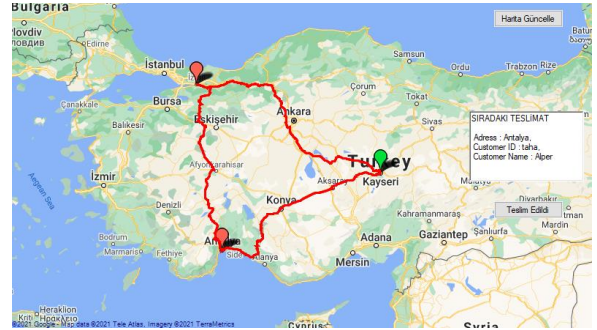
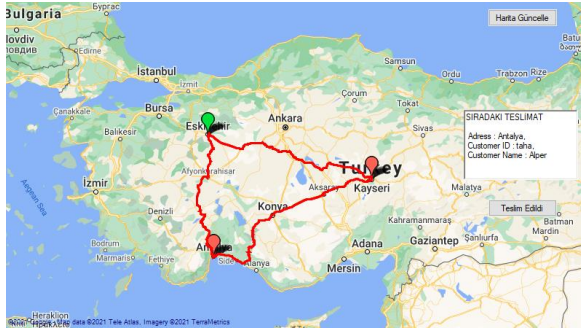
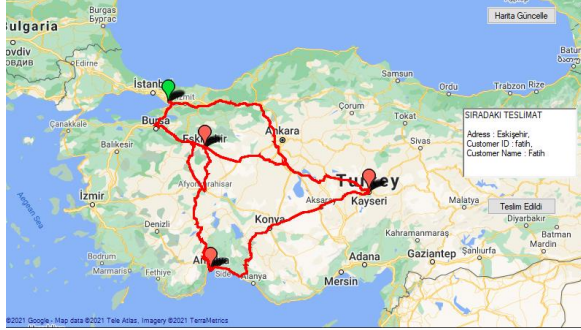
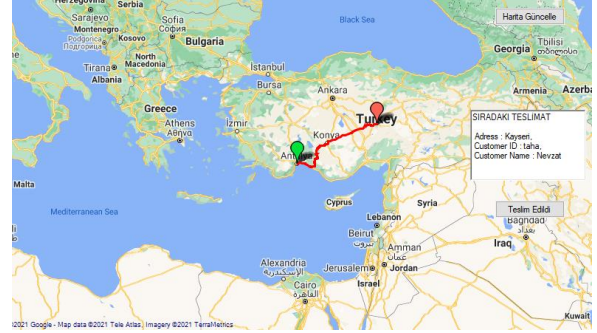
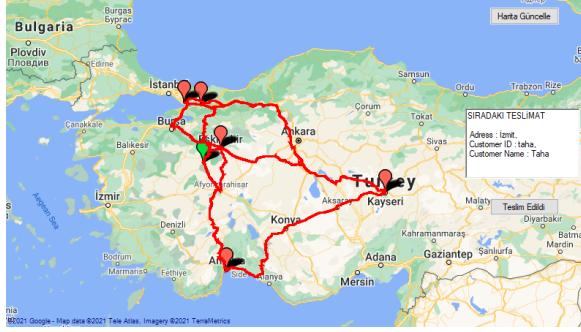
Hariwal Tari

26/10/2021

KargoDurum

Kargonya VERSI

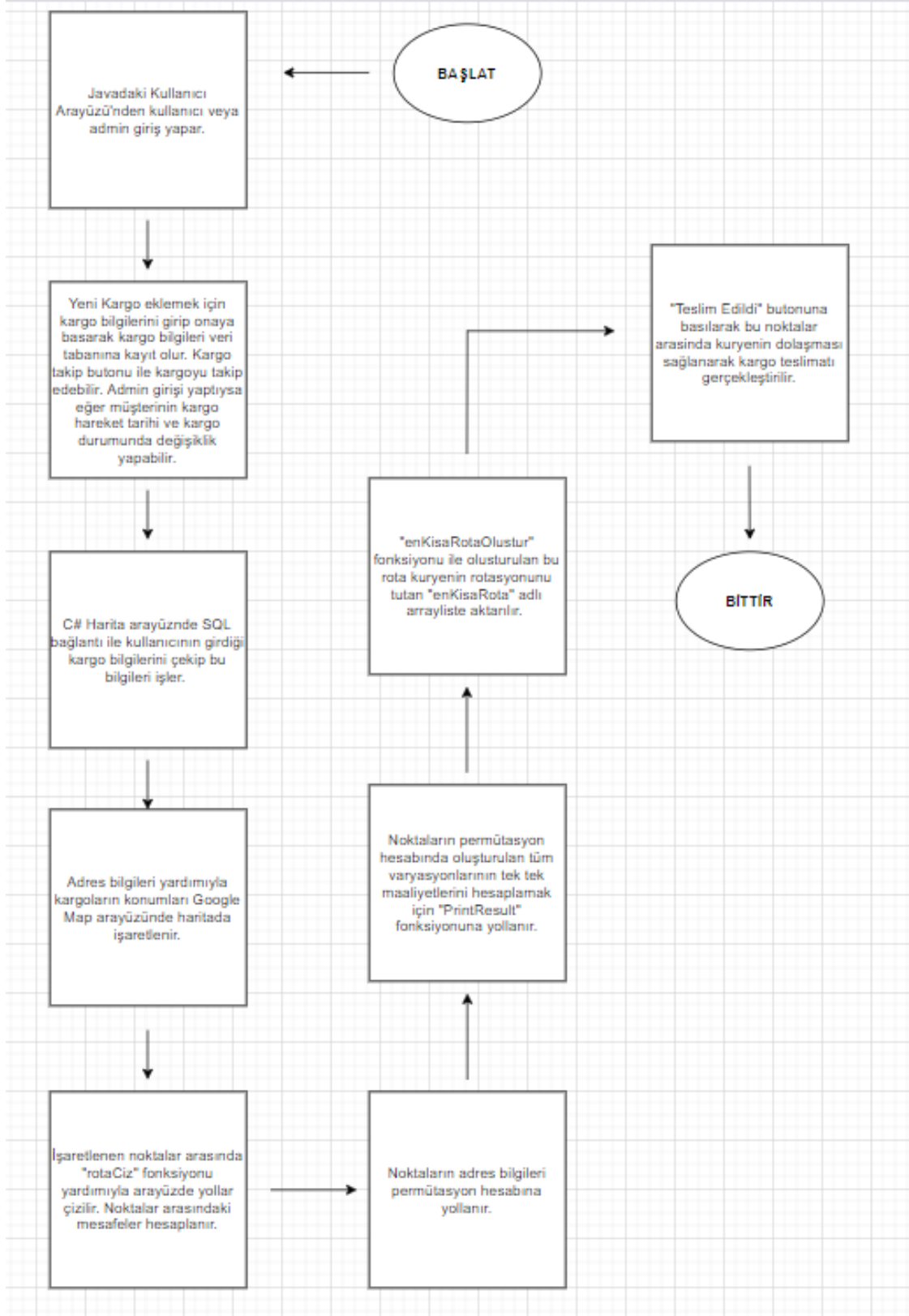
Kargo Diny



4.SONUC

Proje farklı algoritmaların nasıl birbiri ile haberleşebileceğini, en kısa yol algoritmasının nasıl yazılacağını ve Google Maps API ' nin nasıl kullanılacağını öğretti.

5.AKIŞ DİAGRAMI



6.KAYNAKÇA

1.Konumların Koordinatlarını Aldık

<https://latitudelongitude.org/>

2.Traveling Salesperson Problem

<https://developers.google.com/optimization/ro/uting/tsp>

3.Google Map API C# GMap Tutorial Video Serisi

<https://www.youtube.com/playlist?list=PLID7nT-mUjVuqlhWVfaNhnpqCZmNcA9e>

4. Kullanıcı Arayüzü Oluşturulurken
Esinlendiğimiz Arayüz Videosu

https://www.youtube.com/watch?v=JQ_awALyvMM&t=425s