

KOCAELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

Yazılım Laboratuvarı 1

I. Proje Kargo Yol Takip Sistemi

Muhammed Enbiya DEMİR - 190202018
enbiyademir39@gmail.com

I - GIRIS

Kullanıcın, üye olup giriş yaptıktan sonra belirlediği kargo hedeflerine postacının en kısa yolda ulaşacağı bir rota çizdiren program tasarladık.

II – YONTEM

Program Java ve Javascript programlama dillerinin yanında Html ve Css işaretleme dilleri kullanılarak yazılmıştır. Kullanıcı arayüzü için Grafik kütüphanesi olarak Swing kullanılmıştır. Bulut veritabanı olarak Microsoft azure bulut sistemlerinde çalışan Mysql veritabanı kullanılmıştır. Veritabanında oluşturduğumuz tablo ile kullanıcı üye olabiliyor ve üye olduktan sonra email ve şifre sorgusu Azure üzerindeki Mysql veritabanlarından yapılabilir. Tablo da oluşturduğumuz sütunlar aşağıda yer verilmiştir. Usertable tablosunda (user_id, user_name, user_email,user_password).

Cloud Sql ile yazdığımız programın arasında ki bağlantıyı nasıl sağlayacağımız konusunda gerekli araştırmaları yaptık. Bu bağlantıyı sağlamak için “mysql-connector.jar” dosyayı projemize eklememiz gerekti ve yazdığımız kodlarda da Login Class’ı içinde sql bağlantısını sağlamak ve

Çağkan Çağman CERAN - 190202028
190202028@kocaeli.edu.tr

CRUD operasyonlarını gerçekleştirmek için için bazı kodları kullandık.

Giriş yaptıktan sonra bir tarafta kargo listeleri diğer tarafta harita olmak üzere iki farklı ekran açtık. Kargo listeleri kısmında Kargo ekle ve kargo iptal butonları bulunmakta, bunlar listeye ekleme ve silme yapıyor. Bu kodlar ‘Kargo’ isimli Class’ta tutuluyor.

Harita ekranında ise haritaya tıklayarak belirlediği kargo hedeflerine postacının en kısa yolda ulaşacağı bir rota çizdiren bir alanımız var. Bu alanı ‘Map’ sınıfında html dosyası açmamıza yarayan JxBrowser ile yaptık. Pom.xml dosyası içinde özel dependency’leri ekledik.Tasarımı için css dosyası kullandık. Ayrıca haritamızı barındıran html dosyasına javascript dosyası ekledik. Bu javascript dosyasında çağırdığımız Google Maps API hizmetlerini kullanarak Google haritasını açabiliyor, haritada istediğimiz noktaya işaretçi yerleştirebiliyor ve işaretçiler arası en kısa yolu ve tüm işaretçiler arasındaki yolun uzunluğunu hesaplayabiliyoruz.

Tüm noktaların birbirleri arasındaki yolun farkını hesaplayıp sonuçlarını bir matrise aktardıktan sonra ‘Traveling Salesman’ metodunu kullanarak postacının birden çok hedefi olduğunda hangi rota da en az yol kat ederek tüm noktalara kargoları ulaştırabileceğini hesapladık. Elimizdeki bilgiyi kullanarak Google Maps API yardımıyla işaretçiler arasındaki yolu mavi bir çizgi ile çizdirdik.

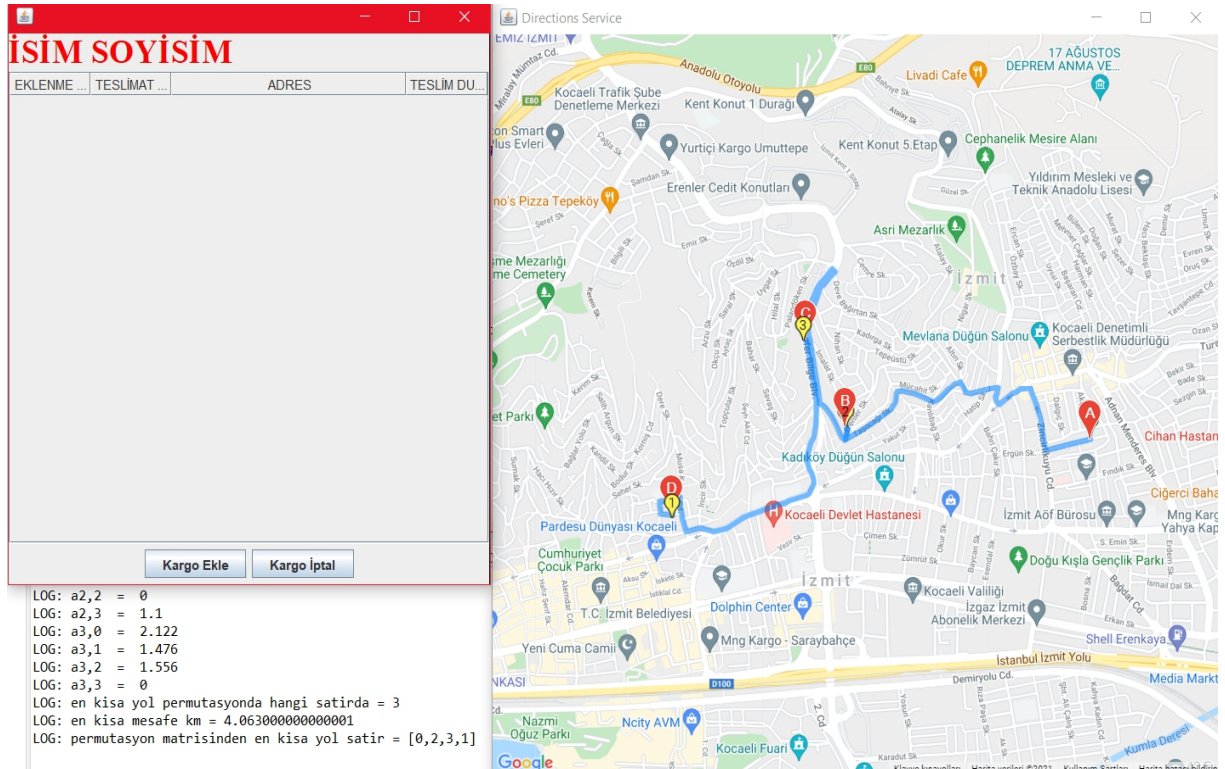
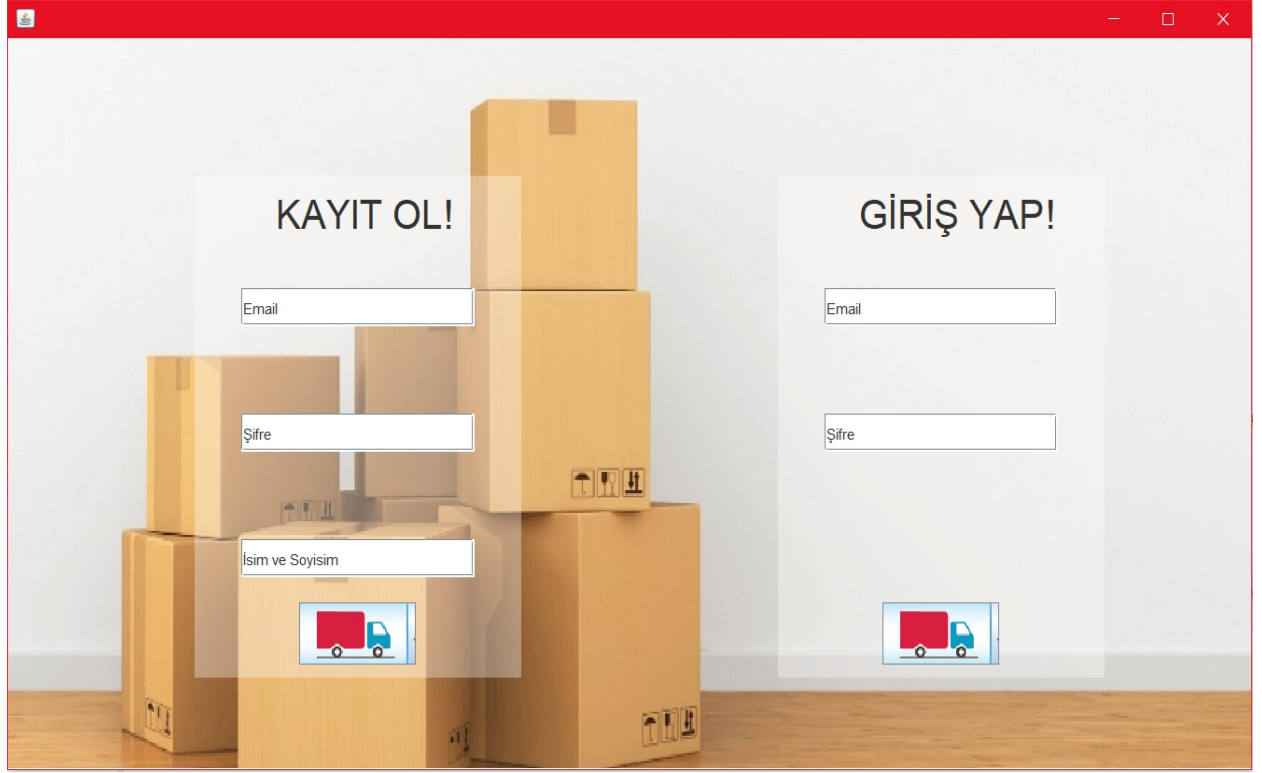
III – SONUC

Giriş ve kayıt arayüzü oluşturduk. Sql sorgularını bulut tabanlı bir sistemle entegre kullanmayı öğrendik. Genel API kullanımını, Google Maps kullanımı ve araçlarını nasıl etikili şekilde kullanabileceğimizi anladık.

IV – KAYNAKÇA

- [1]
“A Java MySQL CRUD Operations”.
[Online]. Available:
<https://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples> [Accessed: 15Oct.-2021].
- [2]
“JSON Document Parse”.
[Online]. Available:
https://www.w3schools.com/js/js_json_parse.asp [Accessed: 20-Oct.-2021].
- [3]
“Google Maps API Documentation”.
[Online]. Available:
<https://developers.google.com/maps/documentation/javascript/overview> [Accessed: 15 Oct.-2021].
- [4]
“Travelling Salesman Problem”.
[Online]. Available:
<https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/> [Accessed: 24-Oct.-2021].
- [5]
“JxBrowser Dev Team”.
[Online]. Available:
<https://www.teamdev.com/jxbrowser> [Accessed: 24-Oct.-2021].

EKRAN GÖRÜNTÜLERİ



```
-- Product name: JxBrowser
-- Licensed version: 6.x
-- Licensed to:
-- License type: Evaluation
-- Generation date: 12 Nis 2017
-- Expiration date: 12 Haz 5000
-- License info: JxBrowser Demo License
-- Current date: 1 Kas 2021
JxBrowser license valid.
Baglanti olustu.
LOG: "40.77133376549607, 29.955944065857445" kordinati eklendi.
LOG: "40.771415017269376, 29.951137547302757" kordinati eklendi.
LOG: uzaklik matrisini yazdir
LOG: a0,0 = 0
LOG: a0,1 = 0.586
LOG: a1,0 = 0.649
LOG: a1,1 = 0
LOG: en kısa yol permutasyonda hangi satirda = 0
LOG: en kısa mesafe km = 0.586
LOG: permutasyon matrisinden en kısa yol satir = [0,1]
LOG: "40.77385515055993, 29.95316095557941" kordinati eklendi.
LOG: uzaklik matrisini yazdir
LOG: a0,0 = 0
LOG: a0,1 = 0.586
LOG: a0,2 = 0.398
LOG: a1,0 = 0.649
LOG: a1,1 = 0
LOG: a1,2 = 0.511
LOG: a2,0 = 0.4
LOG: a2,1 = 0.838
LOG: a2,2 = 0
LOG: en kısa yol permutasyonda hangi satirda = 3
LOG: en kısa mesafe km = 0.911
LOG: permutasyon matrisinden en kısa yol satir = [1,2,0]
```

