

KOCAELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM LABORATUVARI I

İbrahim Burak Tüysüz - Recep Kaya

190201033@kocaeli.edu.tr - 190201027@kocaeli.edu.tr

I.Giriş

Projede akıllı kargo dağıtım sistemi yapan bir masaüstü uygulaması geliştirmemiz beklenmektedir.

Masaüstü uygulamasında 2 adet GUI istenmektedir.1.GUI’de kullanıcının kargo bilgileri girilmesi ve bu bilgilerin veritabanına girilmesi istenmektedir.2.GUI’de veritabanından alınan bilgileri haritaya rota çizilmesi ve kuryenin çizilen rotadan kargoları teslim etmesi beklenmektedir.

2 GUI arası iletişim için thread mantığı kullanıldı.Giriş ekranı yapıldı. Giriş ekranında kullanıcı kayıt olabiliyor, giriş yapabiliyor ve şifresini değiştirebiliyor. Şifresini değiştirmek için kullanıcının eski şifresini de yazması gerekiyor.

II.YÖNTEM

Masaüstü uygulaması yapmak için java programlama dili kullanıldı.

Kullanıcı giriş ekranı, kullanıcı arayüzü ve haritada çizilen rota html’de yapıldı.

Veritabanı firebase firestore bulut veritabanında yapıldı ve uygulama ile bağlandı.

Harita üzerinde çizilecek rotanın en kısa yolu bulunması için heap algoritması kullanıldı.

Harita için gerekli apiler Google maps’ten alındı.

Veritabanında kullanıcı bilgileri ve kargo bilgileri tutuluyor

Arayüz ve harita haberleşmesi için thread mantığı kullanıldı.

a.Login.html

Bu html giriş ekranıdır. Kullanıcı buradan kayıt olabilir, giriş yapabilir ve şifresini değiştirebilir.

b.UI.html

Bu html kullanıcı arayüzüdür. Kullanıcı burada koordinatları girerek veya harita üzerinde tıklayarak veritabanına kargo bilgilerini gönderir.

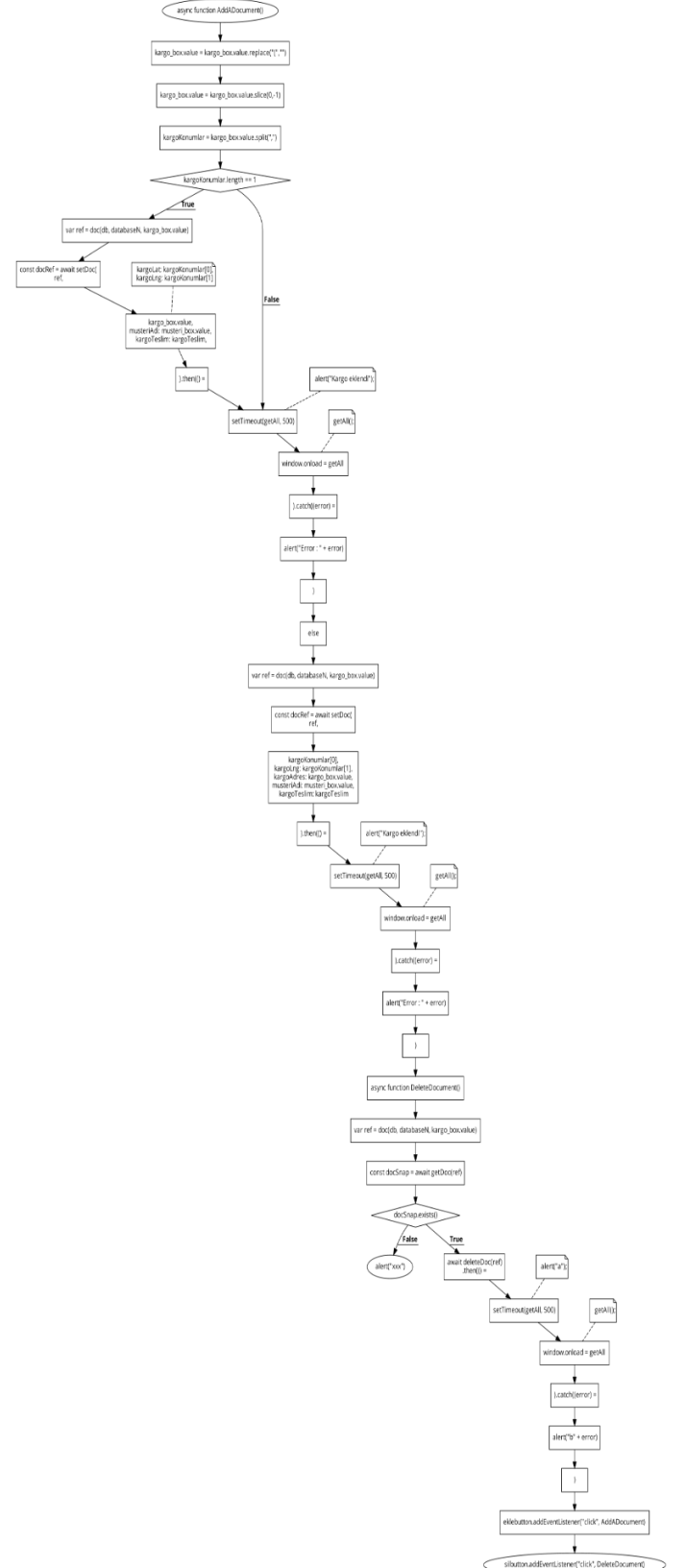
c.MapUI.html

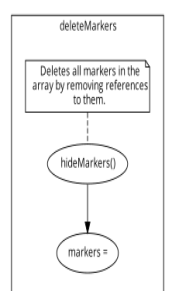
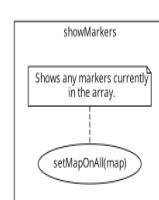
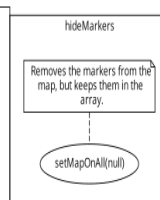
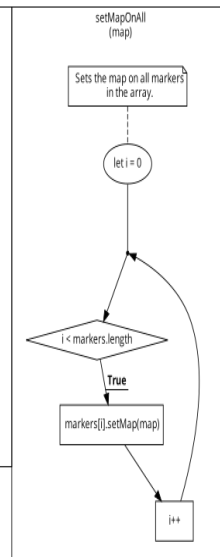
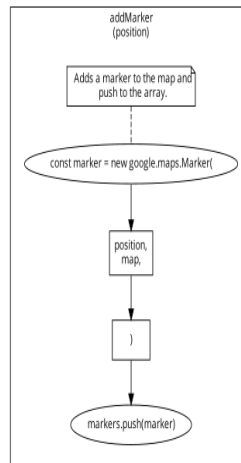
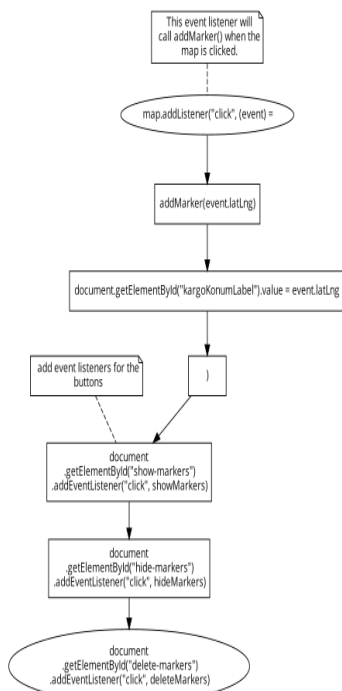
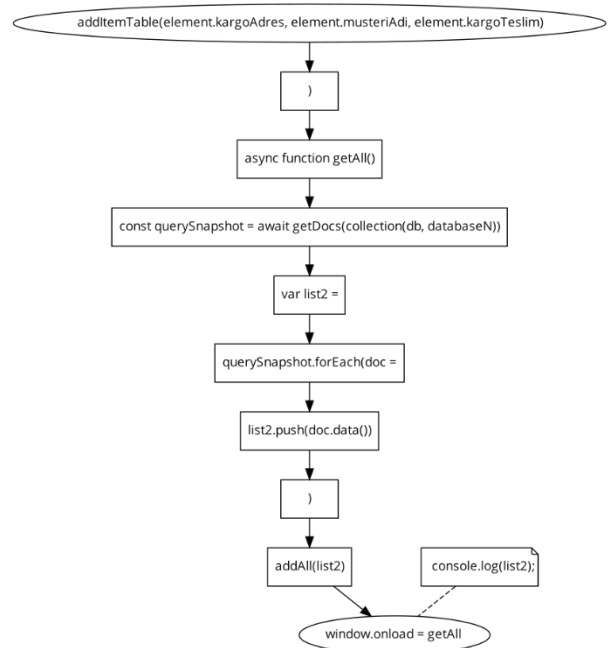
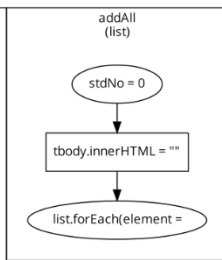
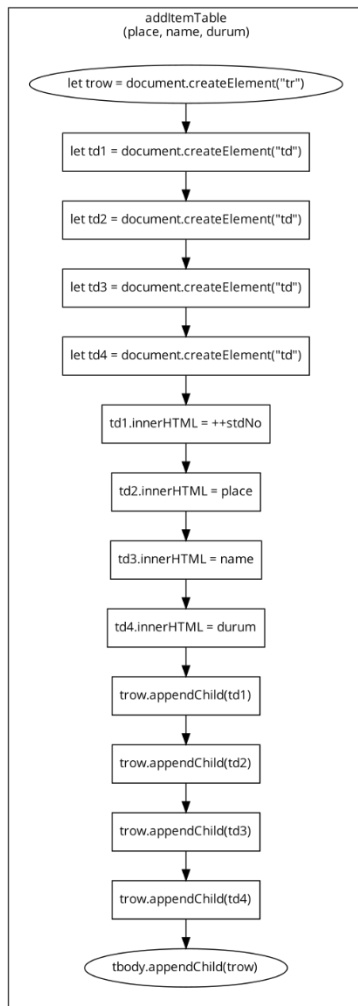
Bu html veritabanından aldığı kargo konumlarını en kısa yolu hesaplayarak harita üzerinde gösterir.

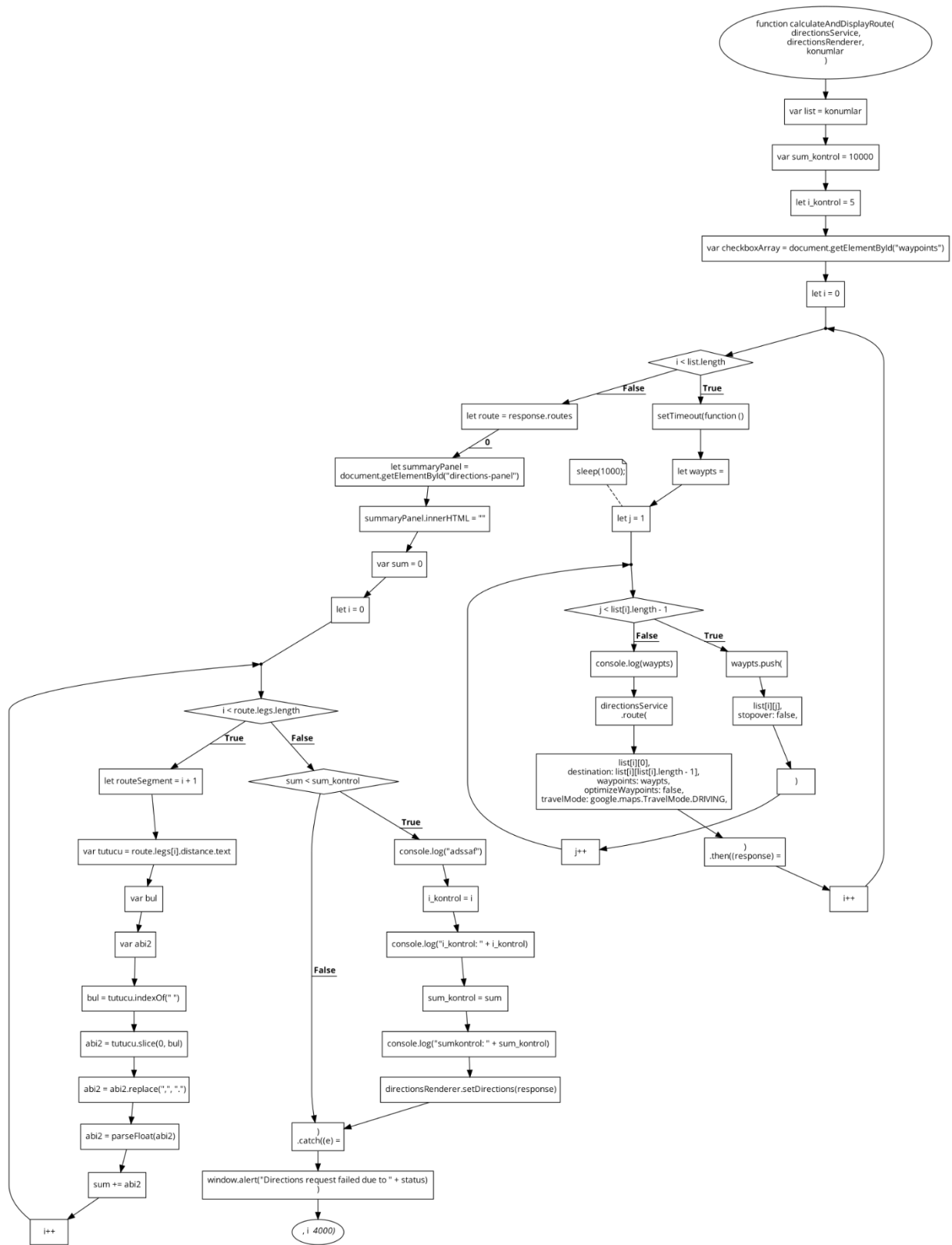
En kısa yolu bulmak için kullanılan algoritma heap algoritması.

Heap Algoritması : [Bilgisayar bilimlerinde](#) kullanılan karşılaştırmaya dayalı bir [sıralama algoritmasıdır](#). Uygulamada pek çok bilgisayarda [hızlı sıralama](#) algoritmasından daha yavaş çalışsa da en kötü durumda $O(n \log n)$ çalışma süresi vardır. Yığın sıralaması diziyi yerinde sıralar ancak kararlı bir sıralama algoritması değildir.


III.AKIŞ ŞEMALARI







IV.EKRAN GÖRÜNTÜLERİ

 GUI 2—□✕

Kullanici adi

Sifre

Giris

Kullanici_adi

Sifre

Kayıt Ol

Kullanici_adi

Eski sifre

Yeni sifre

Sifreni degistir

GUI 2

Autocomplete search

All establishment address geocode (cities) (regions)

☒ Bias to map viewport ☐ Strict bounds

Hide Markers Show Markers Delete Markers

Kargo konumu

musteri adı

Kargo St. Kargo Ekle

id	konum	musteri	durum
1	40.75598966488145, 29.946034566692727	false	
2	40.763393048499836, 29.959849947825585	false	
3	40.76717655577006, 29.95650438787671	false	
4	40.7672033427642, 29.949121387758364	false	
5	40.77333178815191, 29.9587342108674	false	
6	40.776141714264234, 29.97092546635093	false	

GUI 1

Map Satellite Terrain

IV.KAYNAKÇA

BrowserView. (2021). TeamDev. <https://jxbrowser-support.teamdev.com/docs/guides/browser-view.html>

READ, WRITE, UPDATE, DELETE Data | Cloud Firestore v9.1 | JavaScript. (2021, October 4).

YouTube. <https://www.youtube.com/watch?v=z-pyxOx3i1s&list=LL&index=4&t=168s>

Removing Markers | Maps JavaScript API |. (2021). Google Developers.

<https://developers.google.com/maps/documentation/javascript/examples/marker-remove>

Wikipedia contributors. (2021, October 3). *Heap (data structure)*. Wikipedia.

[https://en.wikipedia.org/wiki/Heap_\(data_structure\)](https://en.wikipedia.org/wiki/Heap_(data_structure))

GeeksforGeeks. (2021, September 15). *HeapSort*. <https://www.geeksforgeeks.org/heap-sort/>