

**T.C.  
KARABÜK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ**

**CME442 SENIOR PROJECT II**

**ANDROİD CİHAZLAR İLE KARGO TAKİP UYGULAMASI**

**METEHAN KAFADAR 2014210213004  
NURLAN GOJYALI -2016710213303**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**2019-2020 Bahar Dönemi**

Metehan KAFADAR ve Nurlan GOJYALI tarafından hazırlanan “Kargo Takip Uygulaması” başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Zafer ALBAYRAK

Bitime Projesi Danışmanı, Bilgisayar Mühendisliği

...../...../2020

Bilgisayar Mühendisliği bölümü, bu tez ile Bitirme Projesi Tezini onaylamıştır

Dr. Hakan KUTUCU

.....

Bölüm Başkanı

## ÖNSÖZ

Günümüzde, özellikle günümüz de internetin her eve girmesiyle ile birlikte internet üzerindeki alışverişin büyük ölçüde artması ve kargo alanındaki yoğunluğun artmasıyla , tüm sektörler içinde kargo-teknolojinin önemini arttırmıştır. Günümüzde teknolojinin bize sunmuş olduğu en büyük avantajlardan biri akıllı telefonlardır. Bu sebepten dolayı çalışmamı kargo taşıma ile teknolojiyi yani android cihazlar ile takip konusunda yaptım. Bu çalışmamız sırasında öncelikle tez konusunu seçerken isteklerimizi göz önünde bulundurup bize yardımcı olan, yol gösteren, emeği geçen değerli hocam Zafer ALBAYRAK, bu zamana kadar bizden hiçbir zaman maddi, manevi desteklerini esirgemeyen ailelerimize sonsuz teşekkürler.

Metehan KAFADAR

Nurlan GOJYALI

*“Bu projedeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Metehan KAFADAR

Nurlan GOJYALI

# İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
ÖZET.....	viii

## BÖLÜM 1.

GİRİŞ.....	1
1.1.Projenin Tanımı.....	1
1.1.1. Projenin Amacı .....	2
1.2. Kargo Nedir ? Teknoloji neden bu kadar önemlidir?.....	3
1.2.1 Android Kullanım Yararları.....	4
1.2.2 Google Maps Teknolojisi .....	4
1.3.Google Map API.....	5
1.3.1 API nedir?.....	5
1.3.2 Google API nedir?.....	6
1.3.2 Google Directions API nedir?.....	7
1.4. Mobil Uygulama geliştirme.....	8

## BÖLÜM 2.

PROJEDE KULLANILAN TEKNOLOJİLER.....	9
2.1. Android İşletim Sistemi.....	9
2.1.1 Android Mimarisi.....	9
2.1.2 Android Sürümleri.....	11
2.1.3 Bir Android Projesinin Genel Yapısı.....	12
2.2 Firebase Platformu.....	15
2.2.1 Firebase Tarihi.....	15
2.2.2. Firebase Hizmetleri.....	16

2.2.3. Firebase Real Time Database.....	16
2.2.4. Firebase Authentication.....	17
2.2.5. Firebase Storage.....	19
2.2.6. Firebase Functions.....	19
BÖLÜM 3.	
GELİŞTİRİLEN YAZILIM.....	20
3.1. Projenin Arayüzleri.....	20
3.2. Projenin akış Şemaları.....	22
3.2.1. Kullanıcı için Akış Şeması.....	22
3.2.2. Kargocu rota İçin Akış Şeması.....	23
3.2.3. Alıcı-Kargocu İlişkisi için Akış Seması.....	24
3.3.Veritabanı Şemaları.....	25
3.3.1. Kullanıcılar Veritabanı Şeması.....	25
3.3.2. Kargo Veritabanı Şeması.....	26
3.3.3. Kargocu Puanlama Veritabanı Şeması.....	26
3.4.Proje Use Case Diyagramları.....	27
3.5.Proje Kodlarının Açıklanması.....	28
3.5.1. Main Sayfası.....	28
3.5.2. Kayıt Sayfası.....	28
3.5.3. Kargo Gönder Sayfası.....	29
3.5.4. Kargo QR Sayfası.....	30
3.5.5. Kargo Onaylama butonu Sayfası.....	31
3.5.6. Kargocu Sayfası.....	32
3.5.7. Map Sayfası.....	32
3.5.8. Kargocu Puanlama Sayfası.....	33
3.5.9. Kargocu Puan Görmesi	34
BÖLÜM 4.	
SONUÇLAR VE ÖNERİLER.....	35
KAYNAKLAR.....	31
ÖZGEÇMİŞ.....	32

BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	33
---	----

## **SİMGELER VE KISALTMALAR LİSTESİ**

API	: Application Programing Interface
FB	: Firebase
SD	: Secure Digital Memory Card
SDK	: Software Development Kit
UI	:User Interface
QR	:Quick Response
XML	:Extensible Markup Language

## ŞEKİLLER LİSTESİ

Şekil 1.1.	Uygulama Giriş Ekranı.....	1
Şekil 1.2.	Kargo İletişim.....	3
Şekil 1.3.	Google Maps.....	4
Şekil 1.4.	API'ler.....	6
Şekil 1.5.	Routing Haritası.....	7
Şekil 2.1.	Android Mimarisi.....	10
Şekil 2.2.	Proje Yapısı.....	12
Şekil 2.3.	Firebase Platformu.....	16
Şekil 2.4.	Realtime Database.....	17
Şekil 2.5.	Firebase Authentication.....	18
Şekil 2.6.	Firebase Storage.....	19
Şekil 2.7	Firebase Functions.....	20
Şekil 3.1.	Program Use Case.....	28



## ÖZET

Anahtar kelimeler: Mobil Uygulama, Firebase, Kargo, Android

Bu program android tabanlı bir kargo takip uygulamasıdır.

Uygulama androidde java dili tabanlı olarak yazılmıştır. Veritabanı olarak Firebase veritabanı (bulut tabanlı bir veritabanı) kullanılmıştır. Firebase platformunda ise kullanıcı kimlik doğrulama, gerçek zamanlı veritabanı, fonksiyon özelliği ve depolama (storage) özellikleri kullanılmıştır.

Android telefonlar ile uyumlu olan uygulama; Kullanıcının anlık olarak kargolarını takip etmeyi , hızlı şekilde kargo göndermeyi , kargoyu getiren kişinin puanlamasını yapmaya olanak sağlar.

Projenin amacı android uygulama sayesinde, üye olarak, listeye eklenen kişiler ile QR kod yardımıyla gizlilik ,kolaylık ve pratiklik sağlayarak hızlı şekilde kargo gönderip , alınacak kargoları da anlık olarak nerede olduğunu takip ederek insanlara zaman kazandırmayı hedeflemektedir

## **BÖLÜM 1. GİRİŞ**

Bu bölümde projenin tanımı, amacı gibi proje hakkında temel tanımlamaları hakkında bilgiler yer almaktadır.

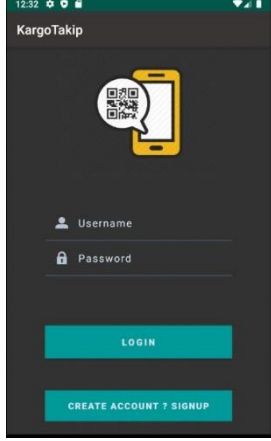
### **1.1 Projenin Tanımı**

Projede android işletim sistemi ve firebase platformu kullanılarak kullanıcıların kargo takibi yapması sağlanmıştır. Firebase platformu gerçek zamanlı olarak depolama yapabildiğimiz bulut (cloud) tabanlı bir platformdur.

Android telefonlarıyla uyumlu olan uygulama; QR kod yardımıyla gizlilik ,kolaylık ve pratiklik sağlayarak hızlı şekilde kargo gönderip , alınacak kargoları da anlık olarak nerede olduğunu takip ederek insanlara zaman kazandırmayı hedeflemektedir.

### 1.1.1 Projenin Amacı

Projenin amacı android uygulama sayesinde, üye olarak, listeye eklenen kişilerle QR kod yardımıyla hızlı ve kolay şekilde kargo gönderip, alınacak kargoların ise anlık olarak nerede olduğunu takip ederek insanlara günümüzde en önemli olan zamanı kazandırmayı hedeflemektedir.



Şekil 1.1. Uygulama Giriş Ekranı

## 1.2 Kargo nedir ? Teknoloji neden bu kadar önemlidir ?

Günümüzde devlet eli (PTT) haricinde, özel şirketler tarafından yapılan büyük ve küçük çapta yük ve eşya taşımacılığı işidir.

Kargo günümüzde bir çok insan tarafından kullanılmaktadır. Bugün Türkiye Kargo, Kurye ve Dağıtım sektörü daha fazlasının da olduğu tahmin edilen 100 bin kişiye istihdam sağlamaktadır. 25 bin araç ile 10 bin sabit merkezde günde 7 Milyon adres ziyareti ve günlük 5 Milyon kilometre yol kat edilmektedir.

Günümüzde teknolojinin hızla gelişme ile birlikte mobil teknolojiler kişisel olduğu kadar kurumsal yaşamda da yerini almaya başlamıştır. Mobil cihazların internete bağlanması ve bu bağlantının gittikçe daha hızlı olması yeni imkanların önünü açmıştır. Günümüzde ‘nesnelerin interneti’ (internet of things) olarak nitelendirilen bir çok cihazın internete bağlanması örnek olarak gösterilebilir. Öyleki artık dizüstü bilgisayarların dışında, diş fırçaları, saatler hatta gözlükler bile internete erişebilmekte ve çeşitli veri alışverişi yapabilmektedir.



Şekil 1.2. Kargo İletişim

### 1.2.1 Android'in kullanım yararları

Kolaylık:Kullanıcı sayısı fazla olduğu için daha çok insana erişilebilmektedir.

Hareketlilik: Google'ın verdiği Apiler sayesinde çok fonksiyonlu uygulamalar üretilebilir.

Üretkenlik: Açık kaynak kodlu olmasından dolayı üretkenlik fazladır.

Kolay kurulum: SDKlar ile birlikte tek tık ile kolayca çalıştırılabilir.

Genişletilebilirlik: MVC yapısından dolayı uygulamanın hacmi ve kullanım alanı geliştirebilir.

Güvenlik: Google ve Yazılımcıların yazdığı sorunsuz kodlar ile sistem açığı en aza indirilir.

Maliyet: Açık kaynak avantajından faydalınarak bir çok framework ve apiye ücretsiz olarak erişilebilmektedir.[2]

### 1.2.2 Google Maps Teknolojisi



Şekil 1.3. Google maps

Google Maps API'ları internette açık kaynak kodlu olarak bulunmaktadır. Herhangi biri tarafından sistemine entegre edilebilir haldedir. Aylık \$200 değerinde kullanımı developer'ın deneyebilmesi için sağlamaktadır. Hazır kodlar bir API Key yardımıyla sorgulanabilir haldedir. Harita tabanlı bir çok çözüm bulunmaktadır. Şirketler yapısına göre Google Maps'in birçok farklı özelliğinden yararlanmaktadır.Fayda anlamında ise, örneğin saha ekiplerinin navigasyon kullanarak A noktasından B noktasına giderken kullanacağı karayolları haritası ile rotalama ile zaman, benzin yani maliyetten kazanç sağlamalarına yardımcı olabilir. Daha sonraki adımlarda ise birden çok durağın mesafe

hesaplama matrisi kullanarak birbirlerine olan uzaklıklarına göre bir sıralama yapılır ve optimizasyon işlemi gerçekleştirilebilir. Birçok firma altlık harita olarak Google Maps kullanmaktadır. Güncel uygu görüntüsü sayesinde araç takip sistemlerinde tercih nedenidir. Ayrıca daha önce gitmediğiniz ve bilmediğiniz bir adresin sokak görünümü (street view) ile incelemesini yapabilirsiniz.

Türkiye (TR) Haritası ve dünya haritası için kullanıcılar tarafından iletilen feedback'ler ile güncellenmekte ve kullanıcılara optimum kalitede bir hizmet verilmektedir. Geocoding API (Coğrafi Konum) X ve Y koordinatından açık adresi il, ilçe, mahalle, cadde, sokak ve kapı numarasına kadar belirleme özelliği vardır. Bunlara ek olarak, açık adresten de X ve Y koordinatını elde etme yeteneğine sahiptir.

## **1.3 Google Maps Api**

### **1.3.1. Api nedir ?**

API (Application Programming Interface) bizim dilimizde “Uygulama Programlama Arayüzü”, bir uygulamanın işlevlerine dışarıdan veya uzaktan erişilip bu işlevlerin kullanılmasını sağlayan arayüzdür. API, bir sunucunun üzerindeki uygulamaya farklı platformlardan ulaşılmasını ve response dönmesine olanak sağlar. Web API'lerinin tamamı REST (REpresentational State Transfer) mimarisi üzerinde dizayn edilir. Bundan dolayı platform bağımsız çalışır. Bu mimari GET, POST, PUT, DELETE metotlarının hepsini desteklemektedir. Web API çıktıları talebe göre JSON, XML gibi çeşitli çıktıları olabilir.

API kullanımının asıl amacı bir uygulamanın bütün veya bazı metotlarını diğer uygulamalara kullanıma açarak uzaktan gelecek veri ve bilgi taleplerini kolayca ve hızlıca karşılamaktır. Böylelikle tek bir uygulamada gerçekleşen işlemlerden izin verilen uzak kullanıcılar belirli parametreler sayesinde faydalanabileceklerdir. API genel olarak gerçek zamanlı veriyi tek tek işlemeye yarar. Sunucunun API üzerinden gönderdiği parametre içeren veya içermeyen girdiyi sunucu işler ve geriye bir sonuç kümesi veya sadece başarı bildirimi döner. Verinin sadece belli bir kısmında yapılacak güncellemeler bir parametre gerektirir. API ise bu işlemlerin hem hızlı hemde pratik olmasını sağlar. Entegrasyon tarafına gelecek olursak, karşıdaki sunucunun API üzerinden izin verdiği fonksiyonları kullanabilmek için öncelikle istemciyi tanıttak bir key gereklidir. Daha sonra bu key ile kullanılacak erişimi onaylayacak olan bir şifre almak gerekir. API hizmeti veren sunucu tarafındaki kurum, ilk olarak istemcinin başvurusunda key ve şifreyi ister. İstenen fonksiyonlar kullanılır ve istenen bilgiler karşı uygulamadan API'lerin döndüğü response'lar vasıtasıyla alınır. İsteğe göre API'lara erişim ile ilgili kısıt konulabilir, bunun nedeni API'leri aşırı meşgul olmasını istemeden kaynaklıdır.[1]

### 1.3.2. Google Api nedir?

Google’ın uygulamaları veya web sitelerinden veri almak için Google’ın bize sunduğu apilerdir. Google api ile analitik, adsense, Google search, Bing, youtube, youtube analitik vs Google’ın sağlamış olduğu hizmetleri kendi projenizden takip etmenizi sağlayabiliriz. Mesela hergün tek tek analitiğe,adsense ve diğer hizmetlere giriş yapmak yerine projenizde kullandığınız panele entegre edebilir ve bu sayede panelde işlem yaparken aynı zamanda sitenizdeki google üzerindeki gelişmeleri takip edebilirsiniz. Kullanma ihtiyacı duymanız halinde ilk olarak yapmanız gereken google api key almaktır. Api key gmail hesabınızla bir api key alıyorsunuz ve onca hizmetten faydalaniyorsunuz.Google’ın bize sağlamış olduğu bu apileri <https://console.developers.google.com> web sitesinden görebiliriz.



#### Google Haritalar API'leri

- Google Maps Android API
- Google Maps SDK for iOS
- Google Maps JavaScript API
- Google Places API for Android
- Google Places API for iOS
- Google Maps Roads API
- Google Static Maps API
- Google Street View Image API
- Google Maps Embed API
- Google Places API Web Service
- Google Maps Geocoding API
- Google Maps Directions API
- Google Maps Distance Matrix API
- Google Maps Geolocation API
- Google Maps Elevation API
- Google Maps Time Zone API

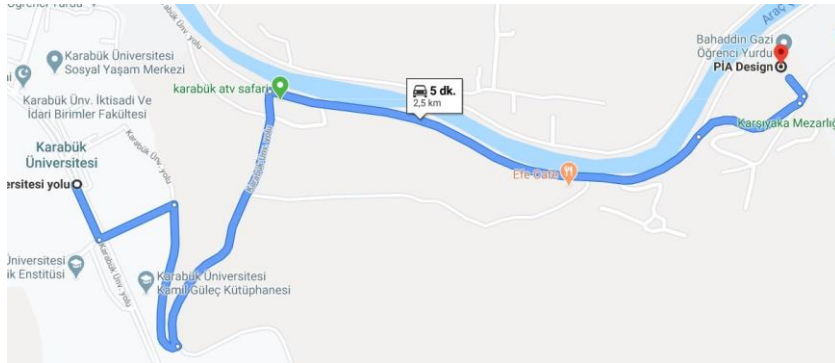
Şekil 1.4 API'ler

### 1.3.3 Google Directions API

```
<meta-data  
    android:name="com.google.android.geo.API_KEY"  
    android:value="AIzaSyAZe21_t6v5q88SuaU07-3GG-tAYtmq9vE" />
```

**Directions api** nedir? Direction yön, istikamet anlamına gelmektedir. Google'ın hizmeti olan Google mapin bize sunduğu bu api bizim başlangıç noktasını ve bitiş noktasını belirtmemiz takdirinde bizi yönlendiren bize en kısa, en verimli, en müsait yolu çizerek bu istikamette bize yol tarifi yapan Google'ın yol tarifi api dir. Aklımıza hemen şu soru gelecek illa Google'ın çizdiği yolu kullanmak zorunda mıyız ben Konya dan İstanbul'a Ankara üzerinden gitmek istiyorum bana yol tarifi yaparken bu güzergah üzerinden yapsın diyebiliyoruz. Veya sadece araba değil yürüme veya bisiklet ile gittiğimizi belirterek buna göre bir yol tarifi de çizdirebiliyoruz. Yani Google bize istediğimiz şekilde özelleştirilmiş bir yol tarifi imkanı sunuyor.[2]

Biz sistemimizde bu API'yı kargocunun en kısa sürede en iyi yolu bulması için kullanmayı hedefledik.



Şekil 1.5. Routing haritası



## 1.4 Mobil Uygulama Geliştirme

Teknolojinin ilerlemesi ve mobil cihaz kullanımının artmasıyla mobil uygulamaların sayısı da artıyor. Özellikle bankacılık, alışveriş vb. hayatı kolaylaştıran alanlarda mobil uygulamalar; basitçe yazılmış bir kod bütünü olmak yerine, kolay ve kullanışlı sistemler sunmak için vazgeçilmez hal alıyor. IOS işletim sistemiyle geliştirilenler App Store üzerinden, Android işletim sistemiyle geliştirilen mobil uygulamalara ise Google Play üzerinden ulaşım sağlanıyor.

IOS ve Android uygulamaları sadece büyük şirketler için değil; küçük ve orta ölçekli şirketler için de yaratılabilir uygulamaları kapsamaktadır. İnsanların bir kuruma telefon üzerinden ulaşması; güçlü bir iletişim yaratmanın yanında, günümüzde bir gereklilik halini almış durumda.

IOS ve Android uygulamalarıyla doğrudan ulaşabildiğiniz müşteriler sebebiyle; bütün kontrol sizin elinizde olur, iş takibini kolay ve sorunsuz yaparsınız. Mobil uygulamalar sayesinde; operasyonel maliyetlerden tasarruf eder, her an istediğiniz şekilde ödeme alabilir, müşteri beklentilerini doğru tespit edip rahatlıkla karşılayabilir, e-ticaret ve reklam kanalları vasıtasıyla gelirinizi arttırabilir; servisinizi ile hizmetinizi daha işlevsel ve kaliteli hale getirebilir, müşteri potansiyelinizi arttırabilirsiniz.

Mobil bir uygulamanın geliştirilebilmesi için belli adımlara dikkat edilmelidir. Mobil uygulamaların da aynı web siteniz gibi sürekli geliştirilip güncellenmesi gerekmektedir. Yine web sitelerinde olduğu gibi, sürdürülebilir olmaları çok önemlidir.

Genellikle hep IOS uygulamalarla ilgilenilse de; birçok telefon Android ve bir kısmı da Windows işletim sistemi kullanmakta. Eğer siz sadece uygulamanızı IOS üzerinden düşünürseniz bu çok büyük bir yanılgı olur. Uygulamanızın bütün işletim sistemlerinde geçerli olmasına ve kusursuz çalışmasına dikkat etmeniz gerekli.

Başlangıçta müşterilerinizden çok fazla izin ve bağlayıcılık istemeyin. Çünkü herhangi bir hacklenme durumunda, kurumsal kimliğiniz çok büyük zarar görür. Ne kadar güvenliğe önem verirsiniz verin, müşterilerinizin aldığınız kişisel bilgi ve izinleri olabildiğince minimumda tutun.

Aynı zamanda kullanıcı yorumlarını çok ama çok ciddiye alın. Çünkü mobil uygulamaların testleri her telefonda farklı olacağından, müşterileriniz aynı zamanda sizin AR-GE çalışanlarınız olacaktır. Yaptıkları yorumları dikkate alıp güncellemeleri ona göre yapmalısınız. İyi çalışan bir IOS ve Android uygulaması; müşterilerinizin memnuniyetini yüksek tutacaktır.[3]

## **BOLUM 2. PROJEDE KULLANILAN TEKNOLOJİLER**

Bu bölümde android işletim sistemi, firebase (cloud) veritabanı, node.js gibi kullanılan teknolojiler ve yazılım dilleri hakkında bilgi verilmiştir.

### **2.1 Android İşletim Sistemi**

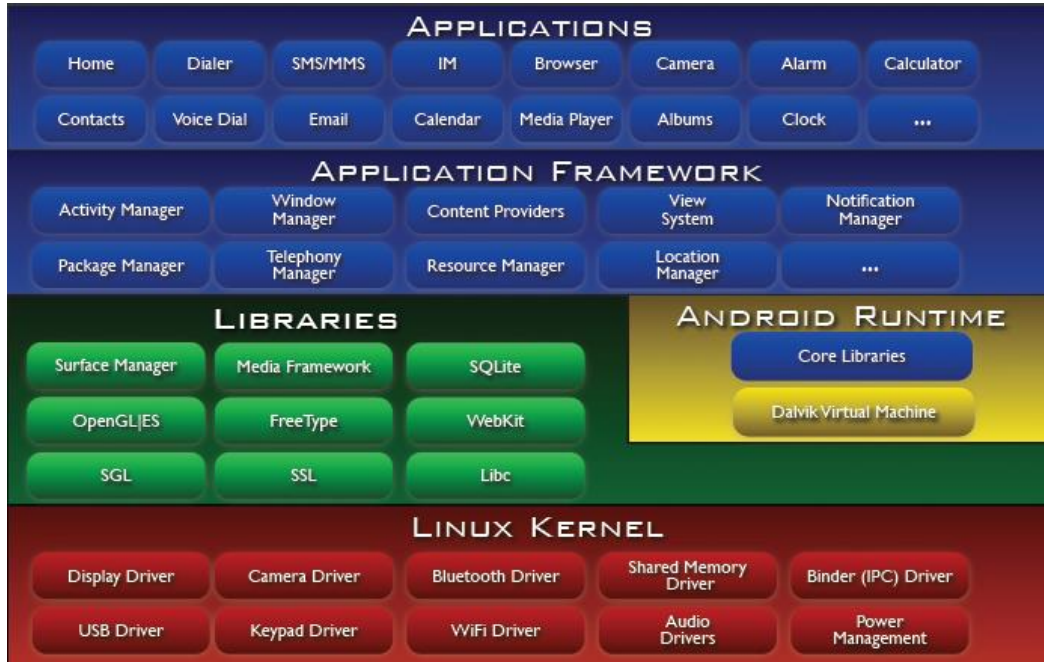
Android Google öncülüğündeki bir organizasyon (Open Handset Alliance) tarafından geliştirilen Linux türevi bir işletim sistemidir. Android; Open Handset Alliance tarafından mobil cihazlar için geliştirilmiş, Linux 2.6 çekirdeğine dayanan, açık kaynak kodlu özel bir Linux dağıtımıdır. Android'in yazılı tarihi Google' ın 2005 yılında Android Inc isimli firmayı satın almasıyla başlamıştır. Google Android'i açık kaynak kod esasına dayanarak geliştirmek istediği için 2007 yılında Open Handset Alliance isimli organizasyonu oluşturmuştur. Open Handset Alliance, Google dışında daha birçok donanım ve yazılım üreticisinin dahil olduğu bir organizasyondur. Örneğin; Samsung, HTC, LG, T-Mobile, Motorola, Intel, Dell, Texas Instruments, Nvidia, Broadcom gibi şuan otuzun üzerinde firma bu organizasyona dahildir.

2007 yılında pek çok yazılım, donanım, telekomünikasyon firmasının katkıda bulunduğu Open Handset Alliance adı verilen şirketler birliği sayesinde ortaya çıkan Android'in kaynak kodları iki farklı lisans kullanır. Kullandığı Linux çekirdeği GPL, diğer dış bileşenler ise Apache Lisansı ile dağıtılmaktadır. Android 2.6 çekirdeği üzerine geliştirilmiş özel bir Linux dağıtımıdır ve process, bellek yönetimi, dosya sistemi, güvenlik gibi temel sistem servisleri diğer Linux türevi sistemlerde olduğu gibidir. Android çekirdek kodlarını açık olarak yazılımcılara sunmuştur. Android açık kaynak kod olmasına rağmen GNU lisansı altında dağıtımı yapılmamaktadır.

Bu platformda uygulamalar Android Yazılım Geliştirme Kiti (SDK) kullanarak Java veya Kotlin dilinde yazılır. Bu SDK yazılımcıya hata ayıklayıcı, yazılım kütüphaneleri ve emülatör gibi yardımcı araçlar sunar.

Android güç kullanımını en aza indirmek ve hafızayı yeterli kullanabilmek için çeşitli yöntemler kullanır. Kullanılmayan uygulamaların bekleme moduna alınması; yetersiz hafıza durumunda uzun süredir aktif olmayan uygulamaların kapatılması bu yöntemlerden bazılarıdır. [4]

### 2.1.1 Android mimarisi



Şekil 2.1. Android Mimarisi

Android işletim sistemi 4 ana katmandan oluşur:

1. Linux Çekirdek (Linux Kernel)
2. Kütüphaneler (Libraries)
3. Android Çalışma Zamanı (Android Runtime)
  - 3.1.Çekirdek Kütüphaneleri (Core Libraries)
  - 3.2.Dalvik Sanal Makinesi (DVM – Dalvik Virtual Machine)
4. Uygulama Çatısı (Application Framework)
5. Uygulamalar (Applications) [4]

### **2.1.2 Android sürümleri**

Android 1.X: 2008’ deki 1.0 sürümünden sonra özellikle 2009’da çıkan Cupcake 1.5 ve Donut 1.6 sürümleri ile layout mantığı, Widget desteği sesle arama gibi özellikler gelmişti. Ayrıca Cupcake sürümü ticari olarak kullanılan ilk sürümdür.

Android 2.X.Y: Temel olarak donanım entegrasyonunun iyileştirilmesi, HTML 5 destekli tarayıcının eklenmesi, zoom özelliği Adobe Flash desteği gibi eklentiler gelmiştir. Pratikte Android’ in en stabil ilk sürümü diye de kabul edilen Froyo 2.2’de ise özellikle performans iyileştirmeleri, Flash eklentisi, SD kart ile uygulama çalıştırılabilmesi, JIT optimizasyonları gerçekleştirilmiştir. GingerBread 2.3 sürümü ise özellikle NFC (Near Field Communication) özelliğinin eklenmesiyle farklılaşır.

Android 3.X: HoneyComb ismiyle de tanınan bu sürümde özellikle telefonlar değil tabletler hedeflenmiştir. Çok çekirdekli işlemcilerin desteklenmesi ve çoklu görev (multitasking) geliştirmeleri bu sürümün belirgin özellikleridir.

Android 4.X.Y: Telefon ve tablet için farklı versiyonların kullanımından vazgeçilmiştir.

Android 5.X.Y: Lollipop

Android 6: Marshmallow versiyonu özellikle uygulama izinleri noktasında ciddi değişimlerin yaşandığı bir versiyon olmuştur.

Android 7.X.Y: Nougat

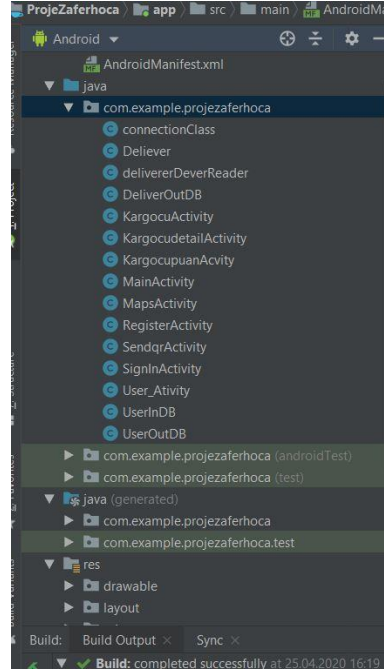
Android 8: Oreo

Android 9:Pie

Android 10

Android 11: Yakında çıkacak

### 2.1.3 Bir android projesinin genel yapısı



Şekil 2.2. Proje Yapısı

Proje oluşturulduğunda MVC yapısı otomatik olarak oluşur. Manifest,Java,Res ve Gradle Scripts dosyaları Android sekmesinde oluşur ve işlem yapmamıza olanak sağlar.

- build klasörü: Belirli derleme değişkenlerine göre derleme dosyalarını içerir. Ana uygulama modülü içinde bulunur.
- libs klasörü: Eğer uygulamamızda Android SDK içinde yer almayan bir Java kütüphanesi kullanıyorsak, ilgili JAR dosyalarını bu klasöre atmamız gerekir. JAR dosyası ekledikten sonra dosyaya sağ tuşla tıklayıp Add To Build Path seçeneğini seçmeyi unutmamalıyız. Aksi takdirde derleyici jar dosyasını göremez.
- src klasörü: Kaynak dosyaları burada yer alır. Bu klasörün içinde java dosyaları tutulur.
- src/androidTest klasörü: Araçlı test belgelerini içerir.

- src/main/java klasörü: Uygulama aktiviteleri için Java kaynak kodlarını içerir.
- src/main/gen/ klasörü: R.Java ya da AIDL dosyalarından üretilmiş arayüz belgeleri gibi Android Studio tarafından yaratılan belgeleri içerir.
- src/main/assets/ klasörü: Genelde boştur, raw değerli dosyalarınızı barındırmak için kullanabilirsiniz. Buraya koyduğunuz dosyalar .apk dosyasının içine derlenir.
- src/main/res klasörü: Uygulamayla ilgili her türlü kaynak bu klasörde tutulur. Burada tutulan dosya tipleri resim dosyaları, ekran tasarımları ve metin dosyaları olabilir. Bu klasörü biraz daha detaylı incelersek:
- drawable klasörleri: Bu klasörler uygulamada kullanılan resim dosyalarını içerir. Bu dosyalar PNG ya da JPEG formatında olabilir. Res klasörünün altında bulunan son sürümlere göre mdpi (medium dpi), hdpi (high dpi), xhdpi (extra high dpi), xxhpi ve xxxhdpi gibi çözünürlükteki klasörler cihaza özel ekran çözünürlüklerine göre dosya çağırmanızı sağlar. Örneğin uygulamanın çalıştığı cihaz eski modelse ve ekranı düşük çözünürlük destekliyorsa ldpi klasörü içindeki resim dosyaları kullanılacaktır. Ama uygulama yeni nesil geniş ekran bir cihazda çalışıyorsa ona uygun olarak yüksek çözünürlüklü bir klasörde bulunan dosyalar kullanılır.
- color klasörü: Renkleri tanımlayan XML dosyalarını içerir.
- mipmap klasörü: Uygulamanızın açılış logolarını içerir.
- layout klasörü: Burada ekranlara dair tasarım dosyaları bulunur. XML formatındaki bu dosyalar her ekrana ait tasarımları barındırır. Bir ekran Activity ile oluşturulduğunda onCreate metodu içinde ilgili layout çağırılır ve ekranda yer alacak öğeler oluşturulur.
- raw klasörü: İsteğe bağlı raw değerli dosyaları içerir. MP3 ve Ogg gibi medya dosyalarını kaydetmek için uygun bir klasördür.
- menü klasörü: Eğer cihazın Menü tuşuna basıldığında bir menü çıkmasını istiyorsak, menü elemanlarını bir xml dosyasında tanımlayarak bu klasör içinde saklayabiliriz.
- values klasörü: Uygulamada kullanılan sabit değişkenler burada saklanabilir. strings.xml dosyası uygulamada kullanılan ve ekranlarda kullanıcıya gösterilen her türlü metni saklar. Anahtar – veri mantığıyla saklanan bu değerler kod içinde ya da layout dosyalarında çağırılır. Aynı zamanda bu değerler R.java

dosyasında işaretlenir. Bunun dışında colors.xml dosyası layout dosyalarında kullanılacak önceden tanımlanmış renkleri içerir. styles.xml dosyası ise ekranlarda kullanılan ve yine layout dosyalarından çağrılan stilleri içerir. values klasörünün bir başka özelliği de cihazın ayarlanmış ana diline göre yerelleşebilmesidir.

- xml klasörü: aPreferenceScreen, AppWidgetProviderInfo, or Searchability Metadata gibi değerleri tanımlayan çok yönlü XML dosyalarını saklar.
- .gitignore dosyası: git tarafından reddedilen kayıt dışı dosyaları tanımlar.
- app.iml dosyası: IntelliJ IDEA modülü.
- build.gradle dosyası: Derleme sistemi için değiştirilebilir özellikleri içerir.
- AndroidManifest.xml dosyası: Uygulamayla ilgili her türlü özellik ve uygulamanın işletim sisteminden talep edeceği bütün izinler burada tanımlanır. Aynı zamanda uygulama içinde kullanılan her ekran burada kaydedilip tanımlanmak zorundadır.[6]
- **Gradle:** Gradle Android uygulaması geliştirme aşamalarını otomatize eden bir build sistemi olup test,debug ve release işlemlerini daha kolay hale getirmektedir.Android için tercih edilmesinin sebebi de içinde xml, resim ve font dosyaları gibi Android projesinde gereken diğer bileşenleri barındıran paketlerin oluşturulabilmesidir.Yani Maven projelerinde sadece Java içerebilen .jar çıktıları kullanılabilirken Gradle’da içerisinde Java, resim, font ve xml içeren .aar çıktıları kullanılabilirsiniz.
- **Android Plugin:** Android derleme sistemi olarak Gradle’ı temel alır .Android Plugin ise Gradle’a Android uygulamalarını oluşturmaya özel çeşitli özellikler ekler. Android Studio’yu güncellediğinizde, Android Plugin’i en son kullanılabilir sürüme otomatik olarak güncellemek için bir uyarı alabilirsiniz. Güncellemeyi kabul etmeyi seçebilir veya projenizin yapı gereksinimlerini temel alarak başka bir sürümü manuel olarak belirtebilirsiniz

Android Plugin’i, Android Studio’daki **File > Project Structure > Project** menüsünden veya Project build.gradle dosyasından güncelleyebilirsiniz. Eklenti sürümü, Android Studio projesinde oluşturulan tüm modüller için geçerlidir.

## 2.2 Firebase Platformu

Google tarafından yeni özelliklerin eklenmesiyle gelişen Firebase, bütün bu ihtiyaçlar karşılayabilme iddiasında ücretsiz kullanım da sunan bir platform. Herhangi bir platformda uygulama geliştirmeye bir sebeple başlayıp daha sonra bir kontrol paneli ve her durumda ulaşılabilir kullanıcı veri deposuna ihtiyacınız olduğunu hemen hissetmişsinizdir. Günümüzde uygulamalar platform farketmeksizin aynı veriye her cihazdan erişmek istiyor. Uygulaması, birçok kullanıcı tarafından yüklenen geliştiricilerin de kayıt - oturum bilgilerini tutma, uygulamaların kullanım verilerini analiz etme, yeni duyurular yapmak için aynı zamanda kullanıcıya bildirim gönderme, uygulamayı test etme gibi işlemleri rahatlıkla yönetebileceği bir yönetim paneli gerekiyor. İşte Google tarafından yeni özelliklerin eklenmesiyle sürekli kendini geliştiren Firebase, bütün bu ihtiyaçları karşılayabilmek için uygulama geliştiricilerine ücretsiz kullanım da sunan bir platformdur.

### 2.2.1 Firebase Tarihi

Firebase platformu 2011 senesinde iki startup tarafından kuruldu. O dönemki ilk adı ise Envolv idi. İlk çıktığında asıl amaç, uygulama üreten yazılımcıların zahmetsiz şekilde chat uygulamasını ekleyebilmeleriydi. Bu uygulamayı API şeklinde tanımlayabiliriz. Chat uygulaması konusunda kolaylık sağlaması pek çok kişinin ilgisini çekti. Buna rağmen tam anlamıyla bu noktaya tatmin olmadılar. Çünkü asıl ihtiyaç olan şey chat uygulaması değildi, bunu fark etmiş oldular. Anlık verilerin senkronizasyonu chat uygulamasından çok daha acil bir uygulama olarak görüldü. Dolayısıyla iki startup bu yönde gelişim sağlamaya karar verdiler. Anlık verilerin senkronuna örnek olarak multi oyunculu oyunlarda oyunlara ait verilerinin senkronize edilmesi gibi düşünülebilir. Bu nedenle hemen bir sene sonra Firebase isminde yepyeni bir firma kurdular ve burada ağırlıklı olarak data senkronu üzerine hizmet verdiler.

2014 senesinde Google, bu hizmeti iki girişimciden satın aldı. Asıl büyüme hikayesi de tam olarak bu noktada başladı. Google sürekli olarak yeni ürünler geliştirdi ve sistemi büyüttü. Bugün geline nokta yazılımla ilgilenen ve daha çok uygulama tasarlayan kişiler için olmazsa olmaz teknolojilerden biri haline gelmiştir. Sizin de kullanıcı girişleri bulunan ya da bazı verilerin depolandığı bir platformunuz varsa Google Firebase sisteminden dilediğiniz ölçüde yararlanabilirsiniz. Firebase bugün hem küçük hem de orta ölçekli pek çok uygulamanın ihtiyaçlarına yanıt verebiliyor. Bu nedenle de en başından bu yana tamamen ücretsiz olarak sunuldu.[7]

### 2.2.2 Firebase Hizmetleri

Google Firebase uygulama geliştirme noktasında çok sayıda ürünü ve hizmeti bir arada sunuyor. Bu hizmetlerden ve ürünlerden bazıları şu şekilde: Realtime Database: Veri tutmaya yarayan büyük bir veritabanı uygulaması diyebiliriz. Temel amacı anlık olarak verileri senkronize etmektir. Authentication: Bu serviste işlem yapmak adına



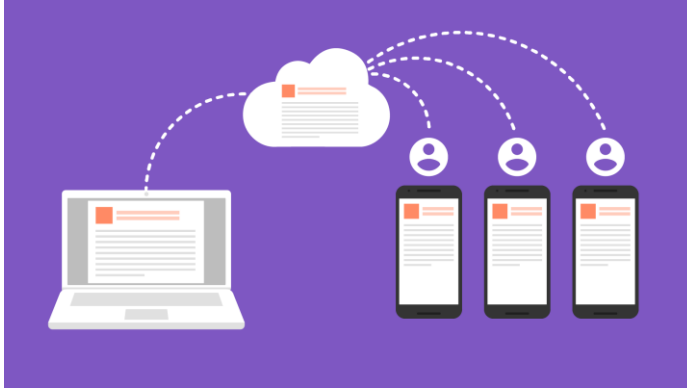
giriş yapmanız şarttır. Test Lab: Ürettiğiniz ya da tasarım aşamasında olduğunuz mobil uygulamalarla ilgili size ön bilgiler sunar. Uygulamada denediğiniz her şeyi hızlıca test edebilirsiniz. Crashlytic: Anlık olarak loglama yapmanızı sağlar. Ayrıca hatalarınızı anında deşifre eder ve bu sayede güncellemeler yapmanıza olanak sağlar. Firestore: Anlık senkronizasyonun daha ölçeklenebilir halidir. Realtime Database ile sık sık karşılaştırılır. Günümüzde beta sürümü olarak hizmet verir.



Şekil 2.3. Firebase Platformu

### 2.2.3 Firebase realtime database

Bir mimari tasarlıyoruz web mimaride birçok kullanıcı bulunmakta. Mobil , Web , Desktop vs. tüm istemciler aynı anda asenkron bir şekilde bilgi alışverişi yapabiliyor ve zaten ismi buradan geliyor. Bulutta çalışan bir NoSql veritabanıdır amma veritabanı olarak alışık olduğumuz tablo ve kolonlar yok. Bu yapı bir json ağacı şeklinde tasarlanmıştır. En önemli avantajlarından biri de şu; normalde veritabanını başka bir sunucuya aktarırken çoğunlukla yaptığımız şey dışarı bir sql dosyası aktarıp onu aktarmak istediğimiz sunucuya içeri aktar diyerek aktarmaktır. Burada Google bize şöyle bir kolaylık sunmakta veri aktarma işlemini json yapısı üzerinden yapabildiğiniz gibi harici bir json dosyasını da veritabanına eklemeniz veya var olan yapıyı dışarı aktarmanız söz konusu.



Şekil 2.4. Realtime Database

### 2.2.3.1 Firebase authentication

Firebase Authentication, uygulamanızın kullanıcılarının kimliğini doğrulamak için arka uç hizmetleri, kullanımı kolay SDK'lar ve hazır kullanıcı arayüzü kitaplıkları sağlayarak farklı platformlarda ve uygulamalar arasında daha ilgi çekici bir deneyim sunar. Şifre kullanarak ve Google, Facebook, Twitter gibi popüler birleşmiş kimlik sağlayıcıları kullanarak kimlik doğrulamayı da destekler. Böylece, kullanıcılar içeriğinize kolayca erişebilir ve uygulamalarınıza hızlı ve güvenli bir şekilde girebilirler.

#### TEMEL ÖZELLİKLER

- Diğer Firebase özellikleriyle **çok iyi entegre olur.**
- OAuth 2.0 ve OpenID Connect gibi **endüstri standartlarını kullandığı için** özel arka ucunuzla kolayca entegre edilebilir.
- **İki farklı geliştirme seçeneği sunar.** Bunlardan biri, eksiksiz bir kimlik doğrulama çözümü olan FirebaseUI, diğeri de bir veya birçok oturum açma yöntemini manuel olarak uygulamanıza entegre etmenizi sağlayan Firebase Authentication SDK'sıdır.
- Kullanıcılarınızın zaten Gmail, Google Play, Google+ ve diğer Google hizmetlerinde kullandıkları Google hesaplarını kullanarak kolayca oturum açmalarını sağlayan **güvenli bir kimlik doğrulama yöntemi sunar.** Ayrıca, şifre kullanarak ve Facebook, Twitter gibi popüler birleşmiş kimlik sağlayıcıları kullanarak kimlik doğrulamayı da destekler.
- Tek seferlik bir onayla güvenli bir şekilde **cihazlarda ve web sitenizde sorunsuz bir uygulama deneyimi sağlar.** Bu, kullanıcılarınız hangi cihazı kullanırlarsa kullansınlar onların ilgisini canlı tutar.

- **Kullanıcıları Google hizmetlerine güvenli bir şekilde bağlar.** Google kişileriyle paylaşımında bulunun, Drive'a dosyalar kaydedin, Takvim'e etkinlikler ekleyin ve daha pek çok işlem yapın.
- **Kullanıcıların Google Cüzdan hesapları ile ödeme yapmalarına olanak tanır.** Kullanıcılara Google Cüzdan hesaplarıyla ödeme yapma imkanı vererek uygulama içi satın alma işlemlerini basitleştirir.



Şekil 2.5. Firebase Authentication

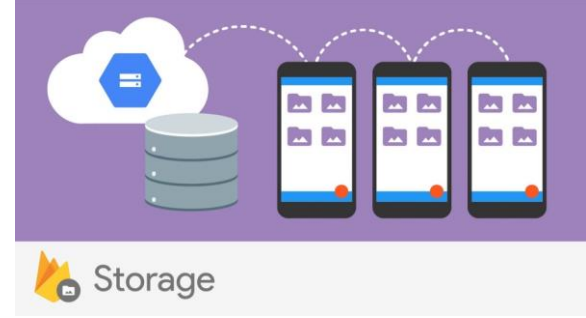
#### 2.2.4 Firebase storage

Storage ile kullanıcıların oluşturduğu resim, ses ve video gibi içerikler sunucu tarafı kodu olmadan depolanır ve erişilir. Temel kurulumda varsayılan olarak tüm okuma ve yazma işlemlerine izin verilir. Güvenlik amacıyla veri yapısı tamamlandıktan sonra özel verilerin güvenliğini sağlamak için yeni kurallar yazılabilir.

```

service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}

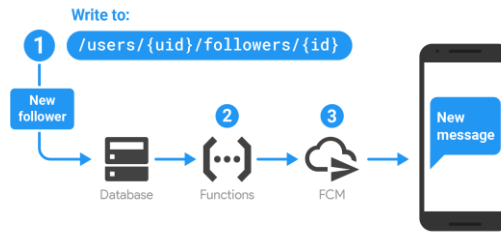
```



Şekil 2.6. Firebase Storage

### 2.2.5 Firebase functions

Firebase Cloud Functions, Google Cloud ve Firebase yapılarının birer parçasıdır. Cloud Functions temel olarak, belirtilen tetikleyiciler üzerindeki yazılı işlevlerinizi yürüten, belirlediğiniz Firebase projeniz için çalışan bir yapıdır. Bir veritabanı işlemi gerçekleştiğinde kullanıcıya bildirim gönderebiliriz. Örneğin kargo takip yapısının olduğu bir uygulamamızda önceki kişiye kargo verildiğinde sonraki kişiye bildirim gönderilebiliriz.

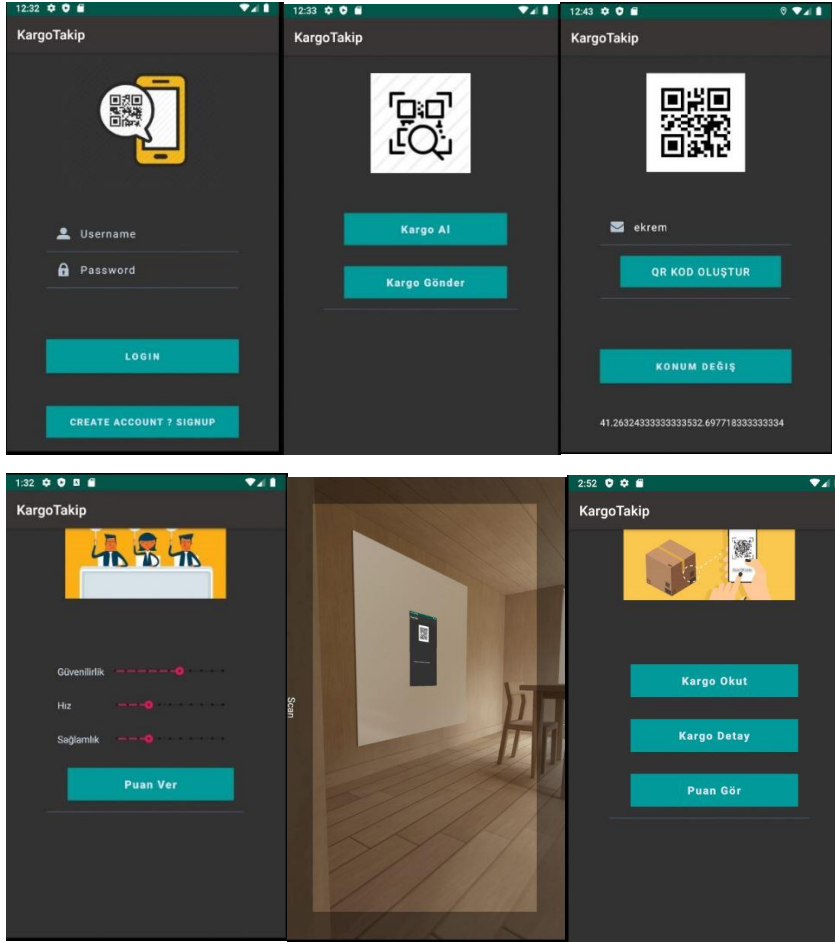


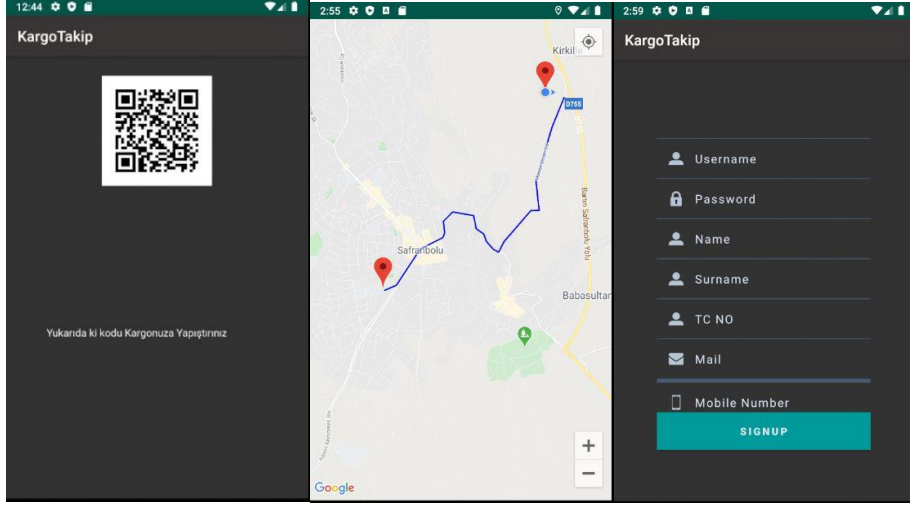
Şekil 2.7. Firebase Function

### 3 GELİŞTİRİLEN YAZILIM

Bu bölümde geliştirilen yazılım için kodların açıklanması, akış diyagramları, veritabanı şemaları, arayüz tasarımları açıklanmıştır.

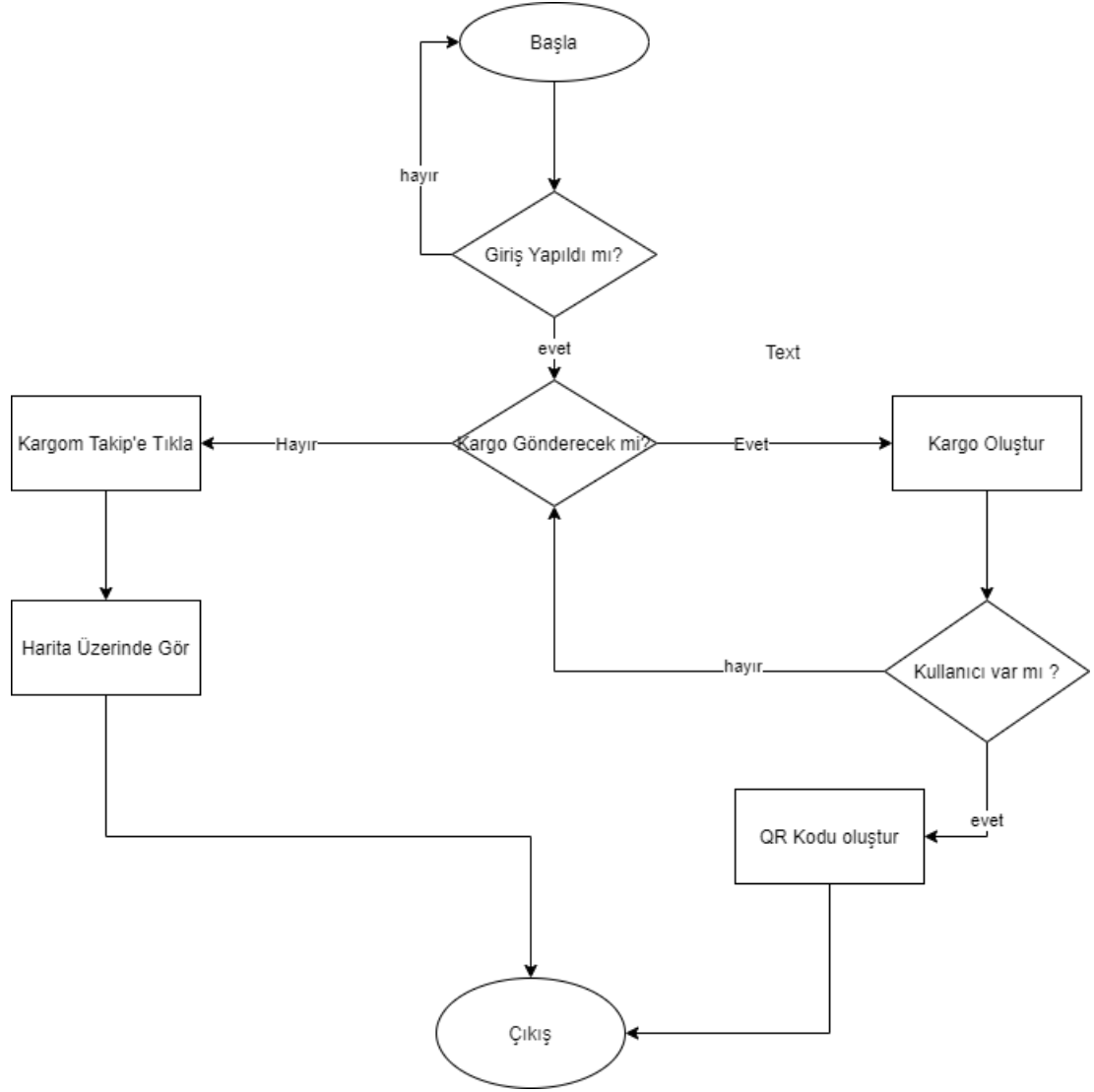
#### 3.1 Projenin Arayüzleri



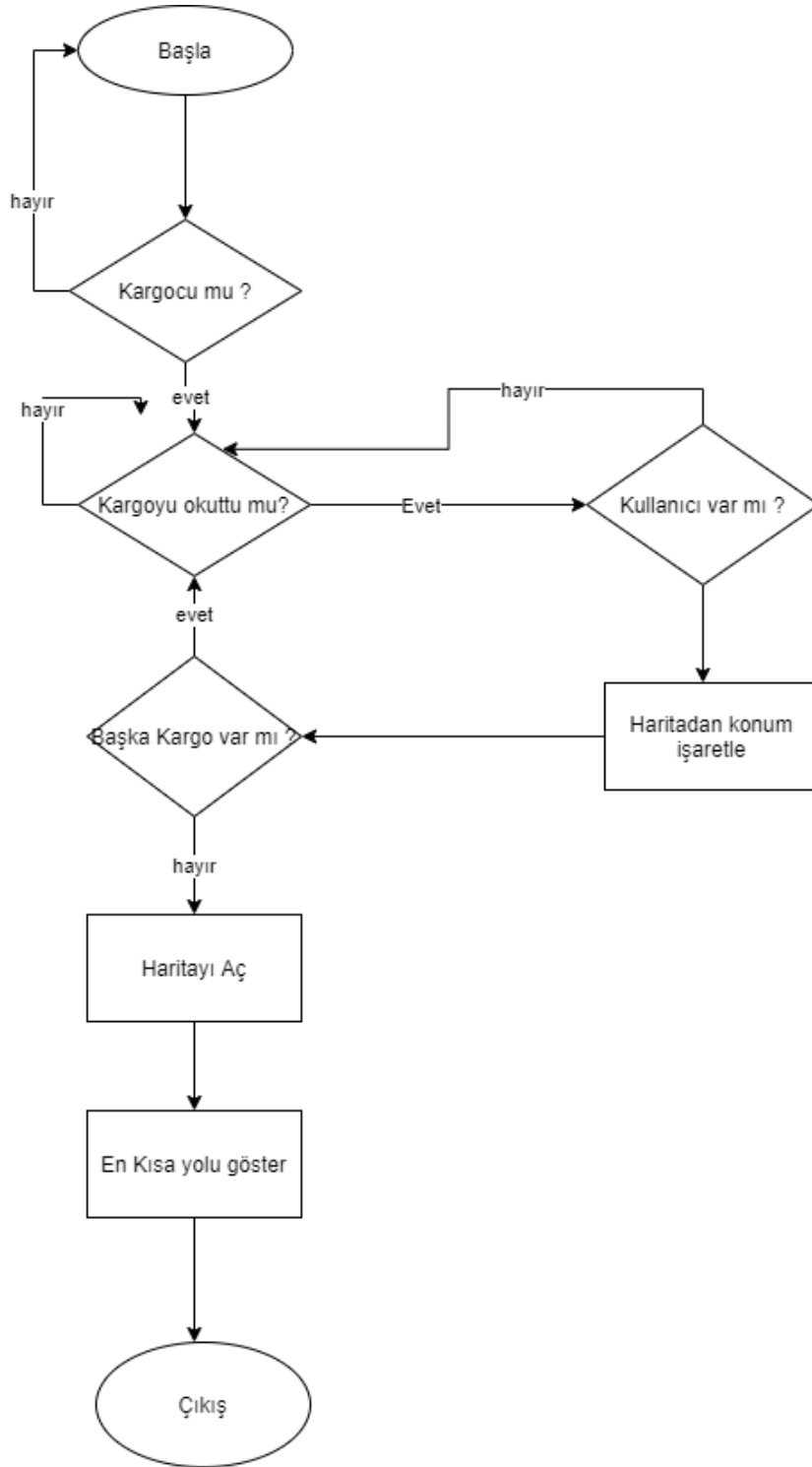


## 3.2 Projenin Akış Şemaları

### 3.2.1 Kullanıcı için Akış Şeması

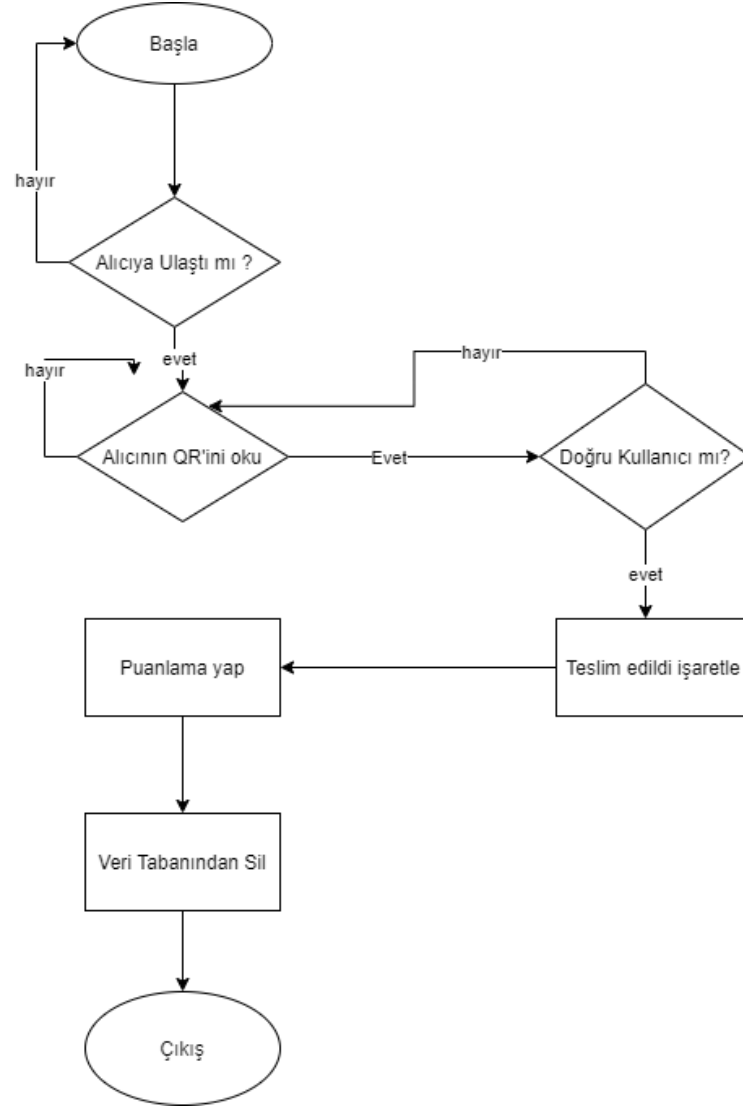


### 3.2.2 Kargocu rota için akış şeması



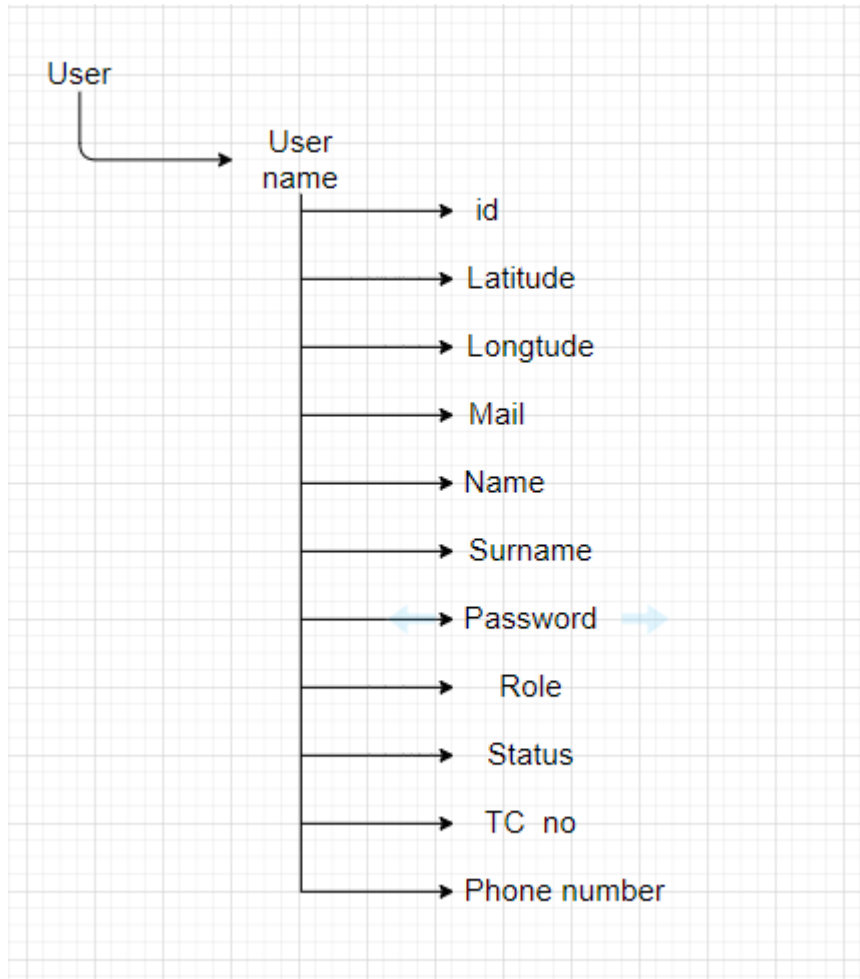


### 3.2.3 Alıcı – Kargocu İlişkisi

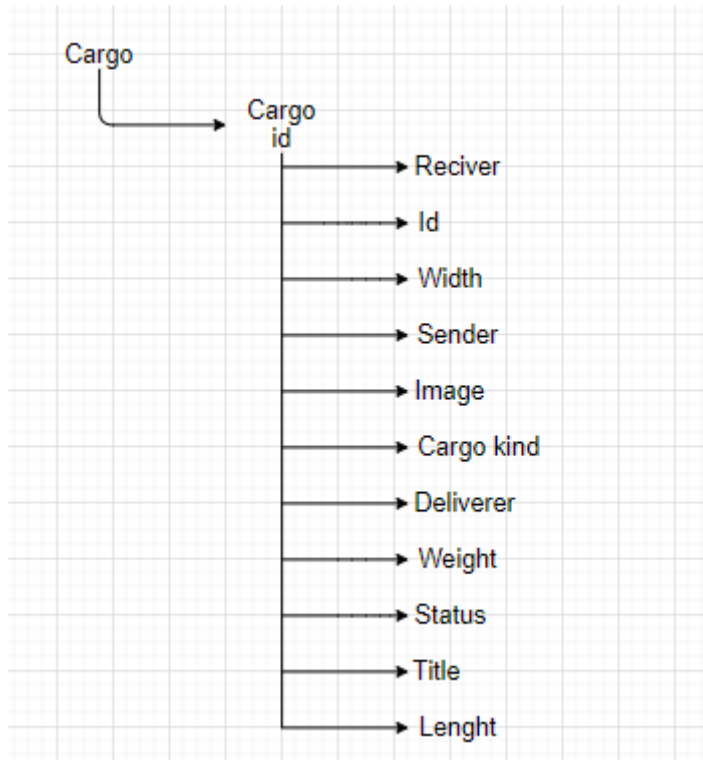


### 3.3 VeriTabanı Şeması

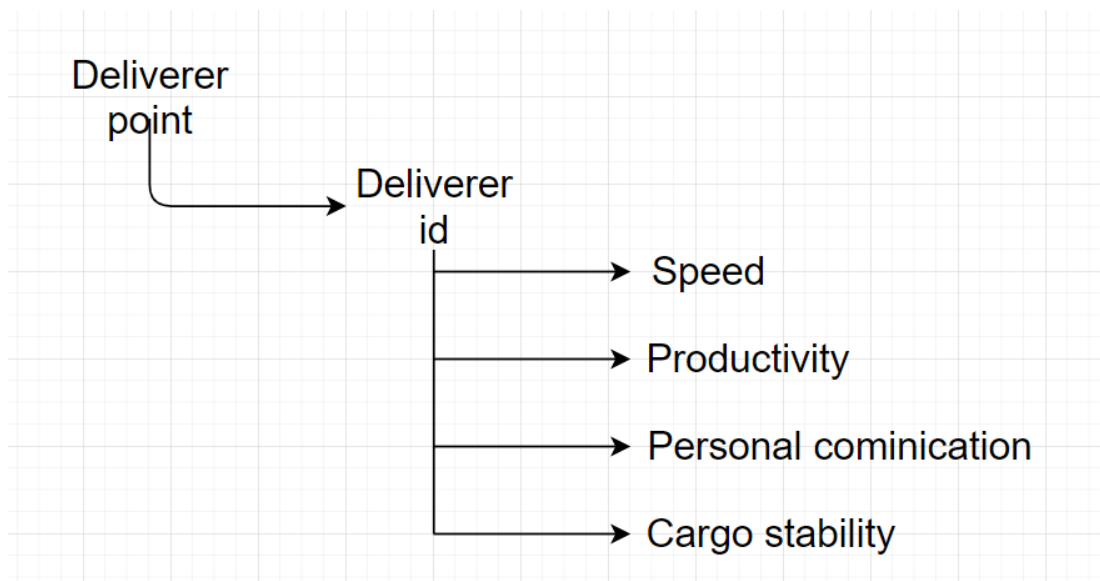
#### 3.3.1 Kullanıcılar veritabanı şeması



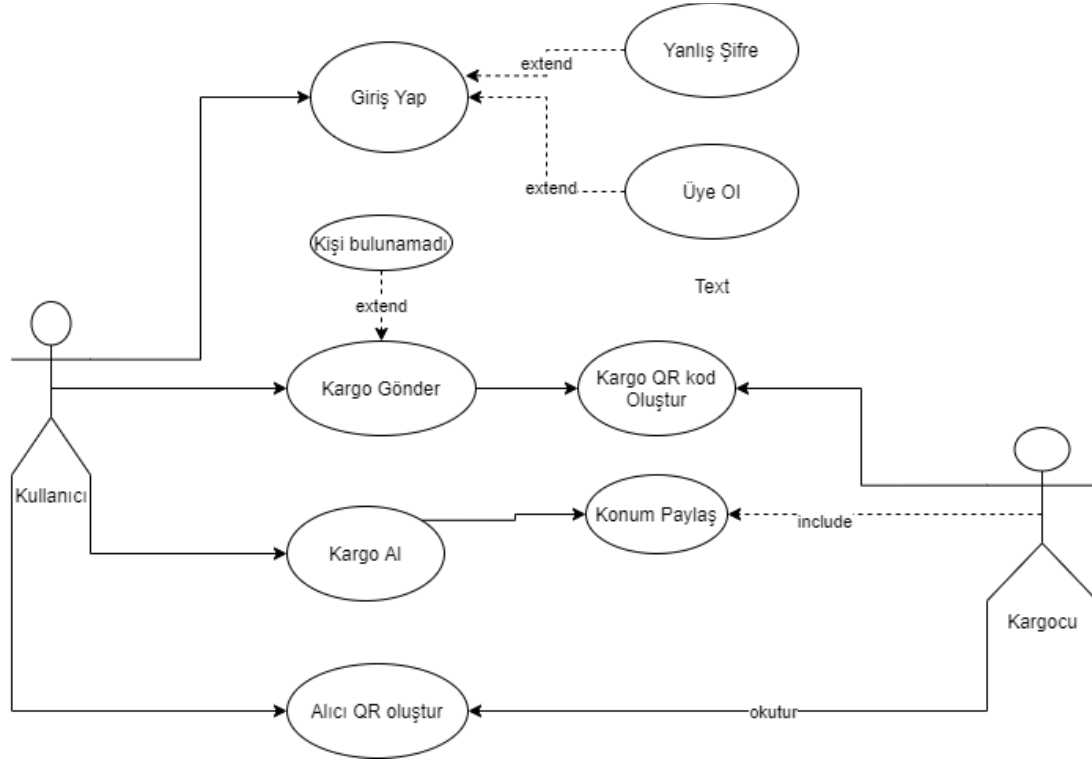
### 3.3.2 Kargo veritabanı şeması



### 3.3.3 Kargocu Puanlama veritabanı şeması



### 3.4 Proje Use Case Diyagramı



Şekil 3.1. Program Use Case

## 3.5 Proje Kodlarının Açıklanması

### 3.5.1 Main sayfası

```
btnmsign.setOnClickListener((v) -> {
    Intent intent = new Intent(getApplicationContext(), RegisterActivity.class);
    startActivity(intent);
});

btnlogin.setOnClickListener((v) -> {
    DatabaseReference database = firebaseDatabase.getReference().child("user").child(username.getText().toString());
    ValueEventListener vel = new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) throws RuntimeException {
            UserOutDB user;
            user = dataSnapshot.getValue(UserOutDB.class);

            if (password.getText().toString().equals(user.password) && username.getText().toString().equals(user.username)) {
                if (user.role.equals("1")) {
                    Toast.makeText(getApplicationContext(), "Merhaba kargocu", Toast.LENGTH_LONG).show();

                    Intent intent1 = new Intent(getApplicationContext(), KargocuActivity.class);
                    startActivity(intent1);
                } else {
                    Toast.makeText(getApplicationContext(), "Merhaba kullanıcı", Toast.LENGTH_LONG).show();
                }
            }
        }
    };
    database.addValueEventListener(vel);
});
```

- Başlangıç sayfasında 2 adet buton ve 2 adet edit text bulunmaktadır.
- Bu butonları MainActivity' de tanımladıktan sonra setOnClickListener dinleyicisi oluşturuldu ve butona tıklandığında tanımladığımız Intent nesnesi ile ilgili activity' e gitmesi sağlandı.
- Bit Intent farklı bileşenler arasında çalışma zamanında bağlama sağlayan nesnedir.
- Giriş doğrulaması yapıldıktan sonra Rolü 1 ise Kargoc Activitesine yollandı.
- Normal Kullanıcı ise UserActivitesine yollandı.

### 3.5.2 Kayıt sayfası

```
signup.setOnClickListener((view) -> {
    //Kullanıcı adında birisi varsa kullanıcı vardır mesajı döndür.

    savename=name.getText().toString();
    savesurmane=surname.getText().toString();
    saveusername=username.getText().toString();
    savepassword=password.getText().toString();
    savetchno=tcno.getText().toString();

    savetelno=telno.getText().toString();
    savemail=mail.getText().toString();

    //Toast.makeText(getApplicationContext(), "TAKİTİME "+savename, Toast.LENGTH_LONG).show();
    connection.addToDB(saveusername, savepassword, savename, savesurmane, savetchno, savemail, savetelno, getApplicationContext());
    connection.removeFromDB(getApplicationContext());

});
}

public void clearTVs(){
```

- Connections Classımızı çağırıldı ve Firebase ile bağlantıyı kuruldu.
- addToDb metodu ile birlite bilgiler veritabanına kaydedildi.

### 3.5.3 Kargo Gönder (SignInActivity)

```
//qr oluşturma kodu
girilendeger = takeun.trim();
if (girilendeger.length() > 0) {
    WindowManager manager = (WindowManager) getSystemService(WINDOW_SERVICE);
    Display display = manager.getDefaultDisplay();
    Point point = new Point();
    display.getSize(point);
    int width = point.x;
    int height = point.y;
    int smallerdimension = width < height ? width : height; //if width<height is true, width is returned, else height is returned
    smallerdimension = smallerdimension * 2 / 4;
    qrgEncoder = new QRLEncoder(girilendeger, bundle: null, QRGContents.Type.TEXT, smallerdimension);
    try {
        bitmap = qrgEncoder.encodeAsBitmap();
        ivsiQR.setImageBitmap(bitmap);
    } catch (WriterException e) {
        Log.v(TAG, e.toString());
    }
} else {
    tvsiWelcome.setError("Required");
}
btnsiChangeloc.setOnClickListener((v) -> {
    String lat=Double.toString(latitude);
    String lon=Double.toString(longitude);
    tvsiWelcome.setText(lat+" "+lon+" "+girilendeger);
    connection.update(girilendeger,lat,lon);
});
btnsiTakeqr.setOnClickListener((v) -> {
    burdaki=etsiSender.getText().toString();
    String deliverid = new connectionClass().kargo(girilendeger,burdaki);
    Intent gotosengr=new Intent( packageContext, SignInActivity.this,SendqrActivity.class);
    gotosengr.putExtra( name: "gonderilen",deliverid);
    startActivity(gotosengr);
});
```

- Kişinin QR kodu anlık Gösterildi.
- Kişi DB’den konumunu anlık olarak değiştirebiliyor.
- Kişi göndermek istediği kişinin kullanıcısı adını yazıyor.

### 3.5.4 Kargo QR sayfası(SendQRActivity)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sendqr);
    ivseQR=findViewById(R.id.iv_seQR);
    final String takeun = getIntent().getExtras().getString( key: "gonderilen",  defaultValue: "defaultKey");
    final String girilendegeer=takeun;

    if (girilendegeer.length() > 0) {
        WindowManager manager = (WindowManager) getSystemService(WINDOW_SERVICE);
        Display display = manager.getDefaultDisplay();
        Point point = new Point();
        display.getSize(point);
        int width = point.x;
        int height = point.y;
        int smallerdimension = width < height ? width : height; //if width<height is true, width is returned, else height is returned
        smallerdimension = smallerdimension * 2 / 4;
        qrgEncoder = new QRGEncoder(girilendegeer,  bundle: null, QRGContents.Type.TEXT, smallerdimension);
        try {
            bitmap = qrgEncoder.encodeAsBitmap();
            ivseQR.setImageBitmap(bitmap);
        } catch (WriterException e) {
            Log.v(TAG, e.toString());
        }
    } else {
        Toast.makeText( context: this, text: "Hataaa",Toast.LENGTH_SHORT).show();
    }
}
```

- Databaseden Kargo ebeveyni(parent) altında Kargo random ID ile yeni bir çocuk(child) oluşturulup içinde gönderenin ve alıcının bilgileri tutulur.
- Oluşturulan ID'nin QR kodu ekrana yansıtılır.
- Mesajlar listeye eklenir ve güncellenir.

### 3.5.5 Kargo Onaylama butonu (SinginActivity )

```
onay.setOnClickListener((v) -> {  
  
    IntentIntegrator integrator=new IntentIntegrator(activity);  
    integrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE_TYPES);  
    integrator.setPrompt("Scan");  
    integrator.setCameraId(0);  
    integrator.setBeepEnabled(false);  
    integrator.setBarcodeImageEnabled(false);  
    integrator.initiateScan();  
  
});  
gonder.setOnClickListener((v) -> {
```

```
@Override  
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
    String statu;  
    DeliverOutDB kargobilgi;  
    kargobilgi = dataSnapshot.getValue(DeliverOutDB.class);  
    kargocuadi =kargobilgi.kargocu;  
    kargoalici=kargobilgi.alici;  
    statu=kargobilgi.status;  
    // Toast.makeText(getApplicationContext(),kargoalici,Toast.LENGTH_LONG).show();  
  
    if (statu.equals("0")){  
        Toast.makeText(getApplicationContext(), text: "kargo zaten alınmış",Toast.LENGTH_LONG).show();  
        Intent puanlama = new Intent(getApplicationContext(), User_Activity.class);  
        startActivity(puanlama);  
    }else if (!getIntent().getExtras().getString( key: "un").equals(null)){  
  
        if ( getIntent().getExtras().getString( key: "un").equals(kargoalici) && statu.equals("1")){  
            Intent puanlama = new Intent(getApplicationContext(), KargocupuanAcvity.class);  
            puanlama.putExtra( name: "kargocuadi", kargocuadi);  
            puanlama.putExtra( name: "kargoid",kargoid);  
            startActivity(puanlama);  
        }else if(getIntent().getExtras().getString( key: "un").equals(kargoalici) && statu.equals("0")){  
            Toast.makeText(getApplicationContext(), text: "gidemezsin karagözlüm",Toast.LENGTH_LONG).show();  
            Intent puanlama = new Intent(getApplicationContext(), User_Activity.class);  
            startActivity(puanlama);  
        }else{  
            Toast.makeText(getApplicationContext(), text: "Sizin Kargonuz Değil",Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

- Kargo üzerindeki QR kullanıcı tarafından okutulduktan sonra okutulan kargonun veritabanında ‘status’üne bakılır ve ona göre işlemler gerçekleşir.



### 3.5.6 Kargocu sayfası(KargocuActivity)

```
final FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
DatabaseReference databaseReference = firebaseDatabase.getReference().child("Kargos").child(kargoid);

ValueEventListener vel = new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        DeliverOutDB deliever;
        deliever = dataSnapshot.getValue(DeliverOutDB.class);
        String alici=deliever.alici;
        DatabaseReference dbref = FirebaseDatabase.getInstance().getReference().child("user").child(alici);
        ValueEventListener valevent = new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                UserOutDB user;
                user=dataSnapshot.getValue(UserOutDB.class);
                String lat=user.lat;

                String lon=user.lon;
                Toast.makeText(getApplicationContext(), "lat+" +lon,Toast.LENGTH_LONG).show();
                Intent gonder=new Intent(getApplicationContext(),MapsActivity.class);
                gonder.putExtra("lat",lat);
                gonder.putExtra("lon",lon);
                startActivity(gonder);
            }
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
}
```

- Kargocu Qr kodu okuttuğunda MapsActivity'e okuttuğu kişinin konum bilgisini gönderir

### 3.6.7 Map sayfası(MapsActivity)

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    final String takelat = getIntent().getExtras().getString("lat", "defaultKey");
    final String takelon=getIntent().getExtras().getString("lon", "defaultKey");
    double lgsol=Double.parseDouble(takelat); //veri çekilip lgsol a alınacak
    double lgsag=Double.parseDouble(takelon); //veri çekilip lgsol a alınacak

    if(ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, Location_req);
    return;
}
mMap.setMyLocationEnabled(true);

final Location location = locationManager.getLastKnownLocation(provider);
if (location != null) {
    str= onLocationChanged(location);
    myhome = new LatLng(str.getLatitude(),str.getLongitude());
    mMap.addMarker(new MarkerOptions().position(myhome).title("my hme"));
} else {
    Toast.makeText( context: this, text: "bulunamadı",Toast.LENGTH_SHORT).show();
}

LatLng baskalokasyon = new LatLng(lgsol, lgsag);
mMap.addMarker(new MarkerOptions().position(baskalokasyon).title("istediğin yer"));
```

Burada LocationManager methodu kullanılarak burayı kullanan Kargocunun ve Kargo sahibinin konumu çekiliyor ve haritada gösteriliyor.

```

DirectionsApiRequest req = DirectionsApi.getDirections(context,lanlonchange,str1);
try {
    DirectionsResult res = req.await();

    //Loop through legs and steps to get encoded polyLines of each step
    if (res.routes != null && res.routes.length > 0) {
        DirectionsRoute route = res.routes[0];

        if (route.legs !=null) {
            for(int i=0; i<route.legs.length; i++) {
                DirectionsLeg leg = route.legs[i];
                if (leg.steps != null) {
                    for (int j=0; j<leg.steps.length;j++){
                        DirectionsStep step = leg.steps[j];
                        if (step.steps != null && step.steps.length >0) {
                            for (int k=0; k<step.steps.length;k++){
                                DirectionsStep step1 = step.steps[k];
                                EncodedPolyline points1 = step1.polyline;
                                if (points1 != null) {
                                    //Decode polyline and add points to list of route coordinates
                                    List<com.google.maps.model.LatLng> coords1 = points1.decodePath();
                                    for (com.google.maps.model.LatLng coord1 : coords1) {
                                        path.add(new LatLng(coord1.lat, coord1.lng));
                                    }
                                }
                            }
                        } else {
                            EncodedPolyline points = step.polyline;
                            if (points != null) {
                                //Decode polyline and add points to list of route coordinates
                                List<com.google.maps.model.LatLng> coords = points.decodePath();
                                for (com.google.maps.model.LatLng coord : coords) {
                                    path.add(new LatLng(coord.lat, coord.lng));
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Burada Kullanıcı ile Kargocu arasındaki yolun en kısa yolunu bulmak için yazıldı.

### 3.6.8 Kargocu Puanlama sayfası(KargocuPuanActivity)

```

DatabaseReference datatref = FirebaseDatabase.getReference().child("puanlar").child(kargocuid);

datatref.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

        kargocuPuanOut kargoout;
        kargoout=dataSnapshot.getValue(kargocuPuanOut.class);
        double hiz,z,guvenn,saglamlik;
        double hiz=Double.valueOf(kargoout.hiz);
        double guven=Double.valueOf(kargoout.guven);
        double saglamlik=Double.valueOf(kargoout.saglamlik);
        double sonpuan;
        int count=Integer.parseInt(kargoout.cnt);
        int intcount = count+1;
        sbguven.setProgress((int)guven);
        sbhiz.setProgress((int)hiz);
        sbsaglam.setProgress((int)saglamlik);

        hiz=hiz*count;
        guvenn=guven*count;
        saglamlik=saglamlik*count;

        hiz=(hiz+hiz1)/(count+1);
        guvenn=(guvenn+guven1)/(count+1);
        saglamlik=(saglamlik+saglam1)/(count+1);

        sonpuan=(hiz+guvenn+saglamlik)/3;

        kargocuPuanIn kargocupuanin = new kargocuPuanIn(String.valueOf(saglamlik),String.valueOf(guvenn),String.valueOf(hiz),String.valueOf(sonpuan));
        datatref.child(kargocuid).setValue(kargocupuanin);
    }
}

```

Önce veriler Firebase'den çekilerek sonra count ile çarpılıp sonra SeekBar'dan alınan veriler ile toplanarak count+1 'e bölünüyor. Count+1 Degerini yeni count'a olarak gösterilip puanlama hesabı yapılıyor.

### 3.6.9 Kargocu Puan Görmesi(KargocuPuanGor)

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_kargocupuanGor);
    tvhizpuan=findViewById(R.id.tv_hizpuan);
    tvguvenpuan=findViewById(R.id.tv_guvenpuan);
    tvsaglampuan=findViewById(R.id.tv_saglampuan);
    final String kargocu=getIntent().getStringExtra("un", "def");
    DatabaseReference dataRef = FirebaseDatabase.getInstance().getReference().child("puanlar").child(kargocu);

    dataRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            kargocuPuanOut kargoout;
            kargoout=dataSnapshot.getValue(kargocuPuanOut.class);
            double hiz=Double.valueOf(kargoout.hiz);
            double guven=Double.valueOf(kargoout.guven);
            double saglamlik=Double.valueOf(kargoout.saglamlik);
            tvhizpuan.setText(String.valueOf(hiz));
            tvguvenpuan.setText(String.valueOf(guven));
            tvsaglampuan.setText(String.valueOf(saglamlik));
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
```

Son Firebase'deki veriler çekilerek kargocuya gösterildi.

### 3 SONUÇLAR VE ÖNERİLER

Bu program android tabanlı mobil uygulamalar kapsamında oluşturulmuş Kargo Takip Uygulamasıdır.

Uygulama androidde java dili tabanlı olarak yazılmıştır. Veritabanı olarak Firebase veritabanı (bulut tabanlı bir veritabanı) kullanıldı. Firebase veritabanında ise kullanıcı kimlik doğrulama, gerçek zamanlı veritabanı, fonksiyon özelliği ve depolama (storage) özellikleri kullanıldı. Harita kullanımı için Android Studionun Google Maps'ı kullanılmıştır.Google Maps ve Google Directions API'leriyle birlikte yol ve rota bulunması sağlanmıştır.Program test edildiğinde kayıt olma, giriş yapma, QR kod oluşturma,oluşturulan QR kodun okutulması ve haritada işaretlenmesi,kargocunun bulunduğu konumdan işaretli yerleri haritada görme gibi işlemler test sonucunda başarıyla sonuçlanmıştır.

Uygulamanın hızlı, kısa bir sürede yapılması ve bireysel bir çalışma olmasından kaynaklanan tasarım ve kod düzeninin profesyonel olmaması projenin eksik yönleridir.Kişilerin kişisel bilgilerinin güvenliği için daha çok güvenlik tedbirleri alınabilir.Uygulama daha çok geliştirilebilir.

## KAYNAKLAR

- [1] <https://medium.com/@akanesen/api-nedir-entegrasyonu-nas%C4%B1l-yap%C4%B1l%C4%B1r-578f93578e46>
- [2] <http://www.apikutuphanesi.com/google-map/directions-api-nedir/>
- [3] <https://www.optimisthub.com/mobil-uygulama-gelistirme-nedir.html>
- [4] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-cihazlar-ve-android-isletim-sistemi-uzerine-genel-bilgiler>
- [5] AYKUT TAŞDELEN, Android Programlama Eğitimi, İstanbul, Pusula Yayınları, Ocak 2017
- [6] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/bir-android-projesinin-yapisi>  
(Erişim Tarihi 10.07.2018)
- [7] <https://www.mediatick.com.tr/blog/firebase-nedir>
- [8] <https://www.npmjs.com/package/firebase>  
(Erişim Tarihi 05.07.2018)