

# İÇİNDEKİLER

<b>ÖZET</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TEŞEKKÜR</b>	<b>iv</b>
<b>ŞEKİL LİSTESİ</b>	<b>vi</b>
<b>1 GİRİŞ</b>	<b>1</b>
<b>2 PROJEDE KULLANILAN ARAÇLAR</b>	<b>2</b>
2.1 Android Studio . . . . .	2
2.2 Firebase . . . . .	4
2.3 Android Picasso Kütüphanesi . . . . .	6
<b>3 NEREDESİN BAKALIM UYGULAMASI NASIL ÇALIŞIR?</b>	<b>7</b>
<b>4 SONUÇLAR VE ÖNERİLER</b>	<b>13</b>
<b>5 EKLER</b>	<b>14</b>
5.1 Android Studio Kodları . . . . .	14
5.1.1 MainActivity.java . . . . .	14
5.1.2 GirişKayıtActivity.java . . . . .	20
5.1.3 Konum.java . . . . .	25
5.1.4 KonumServisi.java . . . . .	26
5.1.5 Takip.java . . . . .	28
5.1.6 MapsActivity.java . . . . .	28
<b>KAYNAKLAR</b>	<b>30</b>
<b>ÖZGEÇMİŞ</b>	<b>31</b>

## ŞEKİL LİSTESİ

1	Android Studio . . . . .	2
2	Android Studio Genel Görünümü . . . . .	3
3	Firebase . . . . .	4
4	Firebase Özellikleri . . . . .	6
5	Neredesin Bakalım İzin İstekleri . . . . .	7
6	Neredesin Bakalım Kayıt Sayfası (Kayıt İşlemi) . . . . .	8
7	Firebase Authenfication Hizmetine Kayıt İşlemi . . . . .	8
8	Neredesin Bakalım Kayıt Sayfası (Profil Kaydetme İşlemi) . . . . .	9
9	Firebase Realtime Datebase Hizmetine Kullanıcılar Tablosu . . . . .	9
10	Neredesin Bakalım Konum Bilgisi Açma Switch . . . . .	10
11	Firebase Realtime Database Konumlar Tablosu . . . . .	10
12	"bm2020" adlı kullanıcıyı takip edilenler listesine ekleme . . . . .	11
13	Firebase Realtime Database Takipleşenler Tablosu . . . . .	11
14	Kullanıcı Harita Konumu . . . . .	12

# 1 GİRİŞ

Günümüzde Android sistem oldukça gelişmektedir bunun yanı sıra mobil cihazlarda her geçen gün daha da gelişmektedir. Bunlar göz önüne alındığında bu platformda uygulama yaparak insanların hayatlarını kolaylaştırmak neredeyse zorunlu hale gelmektedir. Bunlara en güzel örnekler alışveriş uygulamaları, görüntülü ve görüntüsüz haberleşme uygulamaları ve bunlar gibi bir çok uygulamadır. Tabiki sosyal medya mecralarını atlamak olmaz.

Günlük yaşamımızda cep telefonlarımız vazgeçilmez araçlarımız haline gelmiştir. Her geçen gün işlem gücü olarak daha da gelişmektedir. Bunun yanında telefonlar ve tablet bilgisayarlar neredeyse diğer bilgisayar satışlarını geçmiş durumdadır. Buradan anlıyoruz ki insanlar daha çok taşınabilir ve mobil cihazlar tercih etmektedir.

Bunları göz önünde bulundurarak bende insanların mobil cihazlar üzerinde kullanabileceği android bir uygulama geliştirdim. Bu uygulama sayesinde insanlar kolay bir şekilde birbirlerinin konumlarını görecektir ve bu uygulama sosyalleşmeyi veya kaybolan insanların birbirlerini bulmasını sağlayacaktır.

Projemde kullanmış olduğum programlar:

Android Studio(Android kısmının kodlanması)

Firebase(Kullanıcıların verilerinin tutulması)

## 2 PROJEDE KULLANILAN ARAÇLAR

### 2.1 Android Studio



Şekil 1: Android Studio

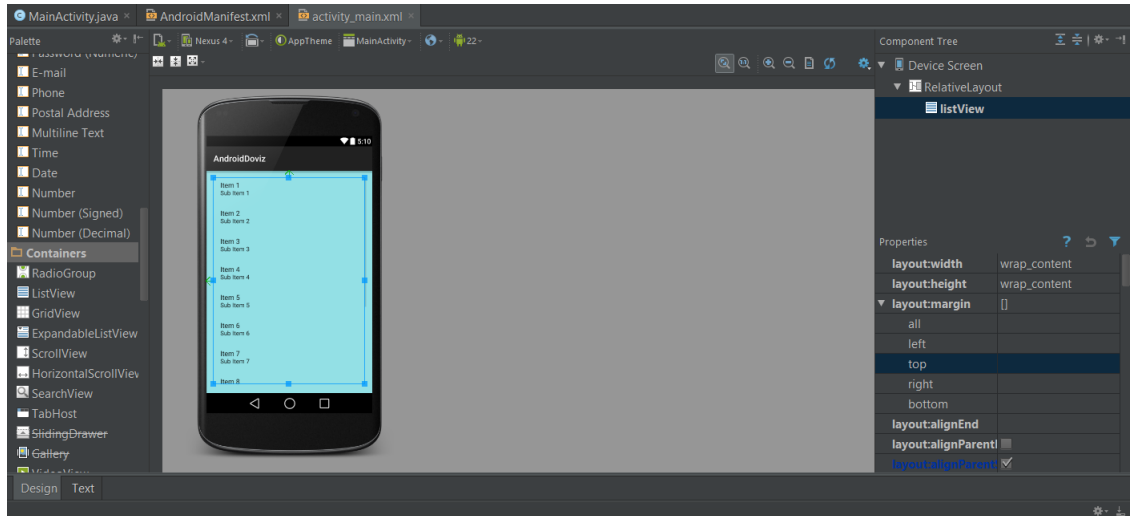
Android Studio, Google tarafından 16 Mayıs 2013 tarihinde Google I/O etkinliğinde tanıtılan ve o tarihten bu yana Android uygulama geliştirmede kullanılan resmi Entegre Geliştirme Ortamı (IDE) olarak bilinir. Her türlü Android cihazda çalışır. Yüksek kalite, verimli sonuçlar elde edilen uygulamalar geliştirmek için en hızlı araçları sağlar.

Android Studio, IntelliJ IDEA'ya dayalı olup Android uygulamaları geliştirmek için özel olarak tasarlanmıştır. Android Studio, akıllı bir kod düzenleyici ve hata ayıklayıcı, performans analiz araçları, emülatörler ve daha pek çok araç dahil olmak üzere uygulama geliştirmek için kullanıcıların ihtiyacını karşılayabilecek birçok özellik barındırır.

Google Android Studio'nun temel özellikleri:

- Google Android Studio, güçlü bir kod düzenleyici ve geliştirici araçları sunan IntelliJ IDEA'ya dayanmaktadır.

- Tüm Android platformları destekler niteliktedir.
- Kullanıcılara kolaylık sağlayan akıllı kod düzenleyiciye sahiptir.
- Anında çalıştırma özelliğiyle emülatör üzerinde çalışan uygulamanızın hızlı bir şekilde yansıtılmasını sağlayarak düzenleme, derleme ve çalıştırma süreçlerini hızlandırır.
- Kapsamlı test araçları ve çerçeveleri sunar.
- Google Cloud Platform ile Google Cloud Messaging ve Google App Engine'i entegre etmeyi kolaylaştıran yerleşik destek sunar. Sık kullanılan uygulama özelliklerini oluşturmanıza yardımcı olacak kod şablonları ve Github entegrasyonu.
- C ++ ve NDK desteği içerir.
- Karmaşıklıktan hoşlanmayanlar için sadeleştirilmiş bir hiyerarşiye göre tasarlanan Yeni Sınırlama Düzeni yöneticisi (Android API seviye 9'dan önceki sürümlerle uyumludur).



Şekil 2: Android Studio Genel Görünümü

## 2.2 Firebase



Şekil 3: Firebase

Firebase, Google'un kullanıcılara armağan ettiği ücretsiz platformlardan birisi durumundadır. Özellikle pek çok yeni özelliğin eklendiği Firebase, şu anda internet kullanan kişilerin hemen hemen tüm ihtiyaçlarını karşılamaktadır. Örneğin herhangi bir platform üzerinde uygulama geliştirildi. Bu tip durumlarda kontrol panellerine ya da veri depolama alanlarına ihtiyaç duyulabilir. Bugün ortaya çıkarılan uygulamaların hemen her türlü cihazda ya da işletim sisteminde sorunsuz şekilde yer alması istenilmektedir. Yönetim panellerinde genelde kayıt, oturum, veri analizi, uygulama testleri gibi işlemleri yapmak mümkün hale gelir.

İşte dünyanın en büyük arama motoru olan Google tam olarak bu ihtiyaçtan hareketle Firebase'i piyasaya sürmüştür. Firebase sisteminin ücretsiz sunulması ve yüksek performans ve verimlilik sunması, onu mucizevi kılıyor. Kullanıcılar Firebase teknolojisinden şimdilik oldukça memnun görünüyor. Normalde uygulama yönetimini ya da bildirim gönderme süreçlerini ayrı sunucular sayesinde yapmak mümkün hale geliyordu. Firebase teknolojisinde ise sunucu taraflı kod yazmaya gerek yoktur.

### Google Firebase ne işe yarar?

Google Firebase yalnızca mobil uygulamalarda işe yarayan bir teknoloji değildir; aynı zamanda bilgisayar gibi cihazlarda çalışan programlar için de yararlanılabilecek bir platformdur. Bugün üretilen projelerin kendilerine özgü yazılım dilleri olsa bile tüm pazara

hitap edecek bir yapıya sahip olmak zorundadırlar. Arka plan verilerinin tutulması ve uygulamaların her türlü platformda kullanılabilmesi, Firebase'in sağladığı faydalar arasındadır. Bu anlamda pek çok uygulamanın ortak sorunlarına karşı çözüm üretebildiğini söylenebilir.

Bugün yazılımcılar genelde uygulama süreçlerinde hangi backend teknolojisinden yararlanmaları gerektiğini tartışır. Aynı zamanda hangi veri tabanından yararlanmaları gerektiği de kritik bir sorudur. Firebase sayesinde yazılım uzmanlarının işi bir hayli kolaylaşmıştır, çünkü artık Firebase ile beraber daha hızlı geliştirme yapılabilmektedir. Firebase ile beraber yazılımcılar özellikle mobil uygulama konusunda daha büyük bir özgüvene sahip olmuşlardır ve bu durum niceliksel anlamda üretime doğrudan yansımıştır.

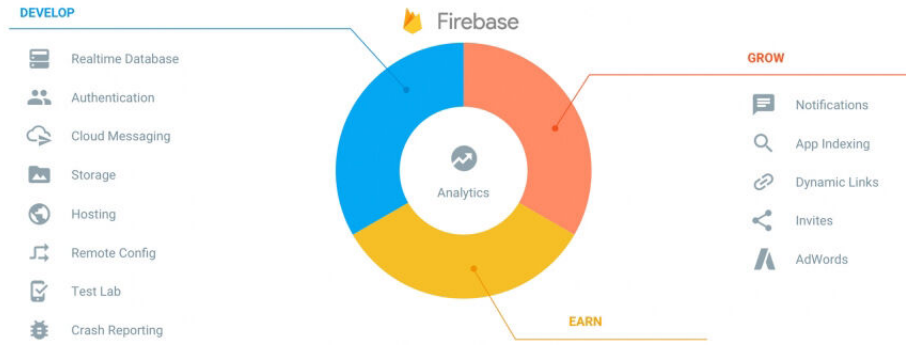
#### Google Firebase'in temel özellikleri:

Günümüzde uygulama yapanların kurtarıcısı olan Google Firebase, kendine has pek çok özelliğe sahiptir. Ayrıca bu özelliklerini her geçen sene daha da geliştirdiği için ayakta kalmayı başarılı şekilde sürdürmektedir.

- Eşzamanlı veritabanı
- Kullanıcı girişlerinde yetkilendirme
- Depolama
- Makine öğrenme kiti
- Performans test ortamı
- Platform arası ortak biçimde uygulama analizleri
- Ortak fonksiyon
- Reklam materyalleri
- Bildirim gönderimi
- Genel uygulama yönetimi

Firestore ürünleri ve hizmetleri: Google Firebase uygulama geliştirme noktasında çok sayıda ürünü ve hizmeti bir arada sunar. Bu hizmetlerden ve ürünlerden bazıları şu şekildedir:

- 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 denenene her şey hızlıca test edilebilmektedir.
- Crashlytic: Anlık olarak loglama yapılmasını 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.



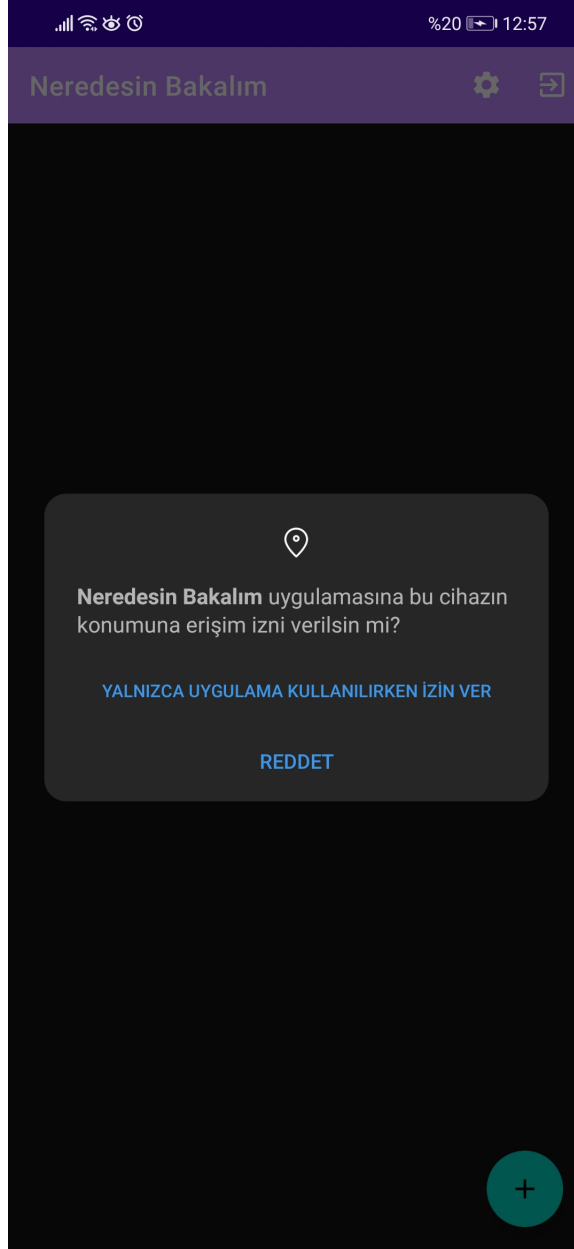
Şekil 4: Firebase Özellikleri

## 2.3 Android Picasso Kütüphanesi

Picasso Android uygulamalarında image yüklenmesi ve işlenmesi için geliştirilmiş bir kütüphanedir. Kullanımı kolaydır. İmagelerin cache'lenmesinide sağlar.



### 3 NEREDESİN BAKALIM UYGULAMASI NASIL ÇALIŞIR?



Şekil 5: Neredesin Bakalım İzin İstekleri

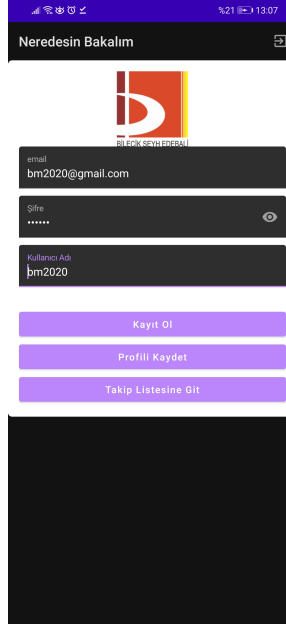
Önce şekil 5’te görüldüğü gibi uygulamayı telefonumuzda ilk çalıştırdığımızda bizde konum izinleri isteyecektir. Bu izinler bizi takip eden kişilerin bizim konumumuzu görebilmesi için gereklidir. Bunlara izin verilmelidir.

Şekil 6: Neredesin Bakalım Kayıt Sayfası (Kayıt İşlemi)

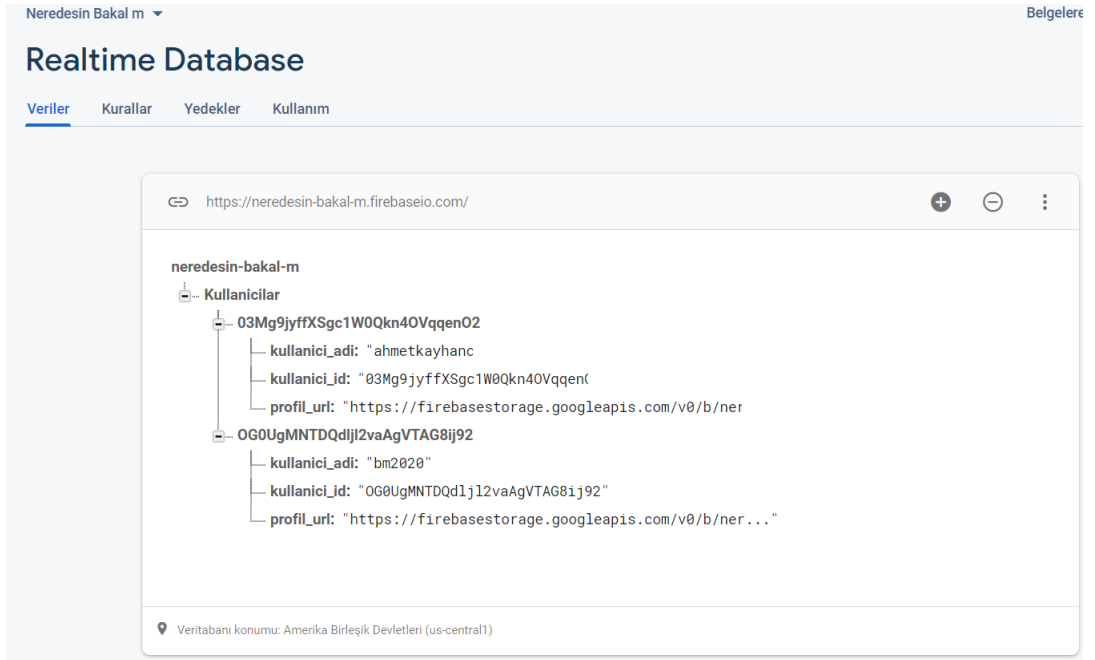
Tanımlayıcı	Sağlayıcılar	Oluşturulma tarihi	Oturum Açma Tarihi	Kullanıcı UID'si
ahmetkayhanc@gmail.com		31 Ara 2020	31 Ara 2020	03Mg9jyffXSgc1W0Qkn40Vqqen02
bm2020@gmail.com		31 Ara 2020	31 Ara 2020	OG0UgMNTDQdljI2vaAgVTAG8lj92

Şekil 7: Firebase Authenfication Hizmetine Kayıt İşlemi

Şekil 6’da “bm2020” adlı kullanıcılarımızın email adresini ve şifresini “kayıt ol “ butonu ile şekil 7’de gösterilen firebase veritabanımızın authentication bölümüne kaydediyoruz.

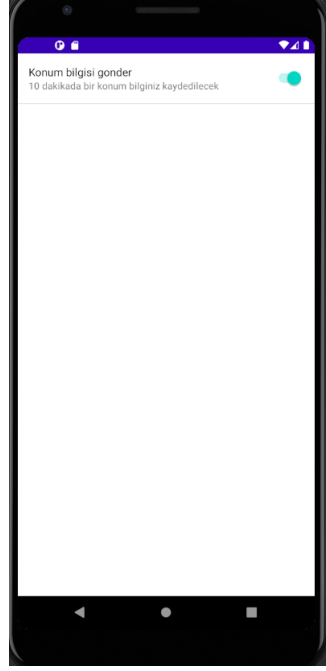


Şekil 8: Neredesin Bakalım Kayıt Sayfası (Profil Kaydetme İşlemi)

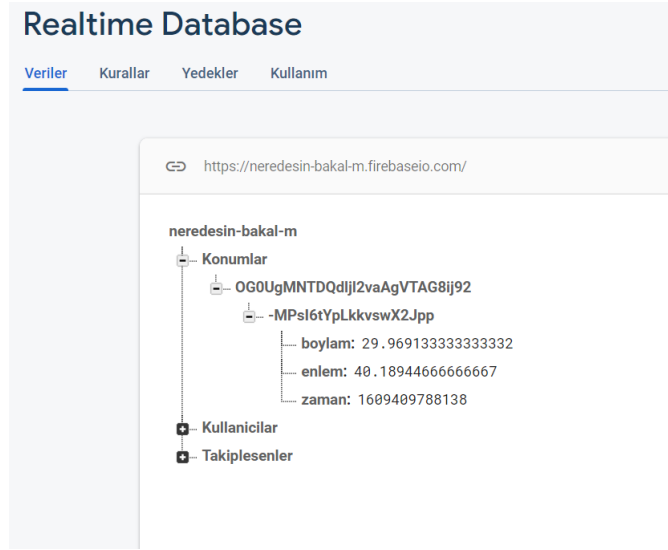


Şekil 9: Firebase Realtime Database Hizmetine Kullanıcılar Tablosu

Şekil 8’de “bm2020” adlı kullanıcımızın, kullanıcı adını ve profil resmini “profil kaydet” butonu ile şekil 9’da gösterilen firebase veritabanımızın realtime-database bölümündeki "Kullanıcılar" tablosuna alıyoruz. Burada görünen "profil\_url" kısmı picasso resim kütüphanesini kullanarak profil fotoğrafımızı firebase’nin storage hizmetine kaydediyor.

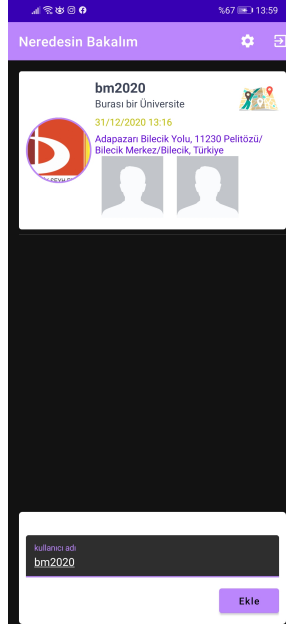


Şekil 10: Neredesin Bakalım Konum Bilgisi Açma Switch



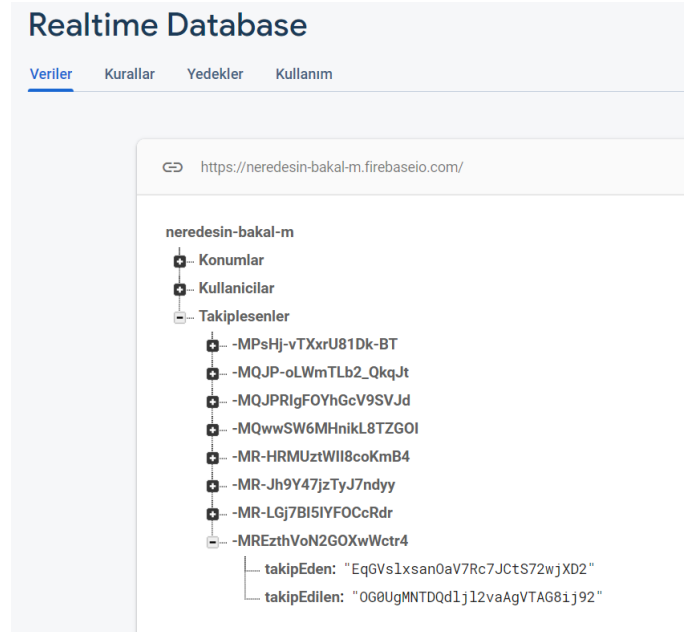
Şekil 11: Firebase Realtime Database Konumlar Tablosu

Şekil 10’da “bm2020” adlı kullanıcımız profilinin ayarlar bölümünden konumunun ve konum alma zamanının (enlem ve boylam bilgileri), şekil 11’de ki firebase veritabanının realtime-database hizmet sayfasında "Konumlar" adlı tablo altına kayıt olmasına izin vermektedir.

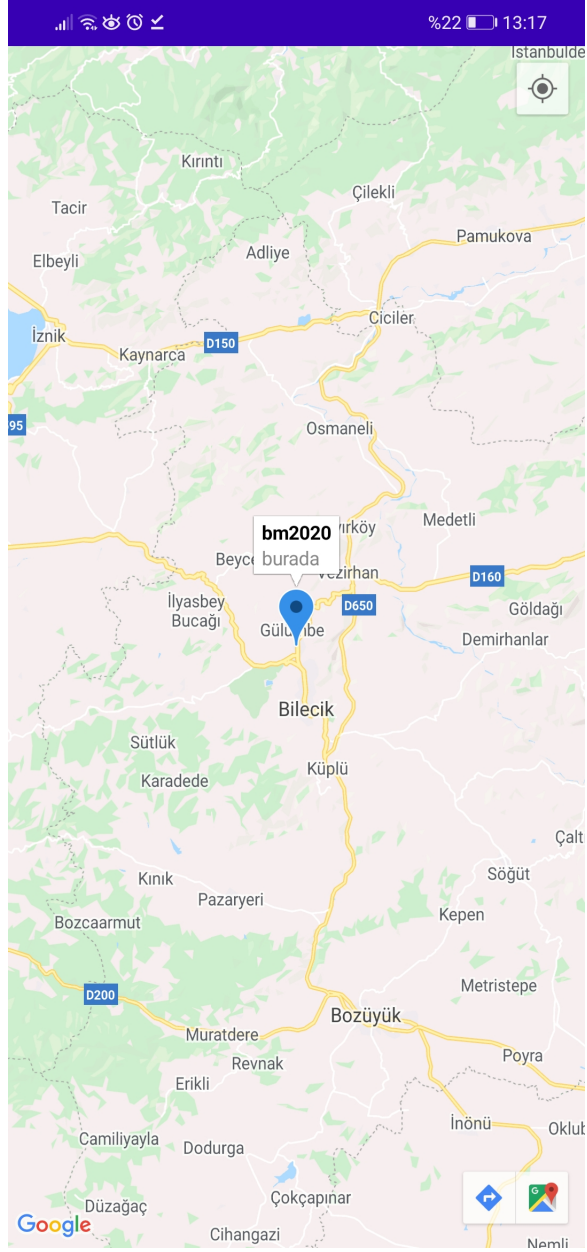


Şekil 12: "bm2020" adlı kullanıcıyı takip edilenler listesine ekleme

Şekil 12’de başka bir kullanıcımızın profilinden “bm2020” adlı kullanıcıyı takip ediyoruz. Takip işleminin firebase veritabanındaki işleyişini şekil 13’te görebilirsiniz.



Şekil 13: Firebase Realtime Database Takiplesenler Tablosu



Şekil 14: Kullanıcı Harita Konumu

Şekil 12’de görülen "bm2020" kullanıcısının üzerindeki harita ikonuna tıklayarak, şekil 14’te görüldüğü gibi konumunu harita üzerinde görülebilir.

## 4 SONUÇLAR VE ÖNERİLER

Bu projede ilk kez Android studio, Firebase ve Picasso kütüphanesini kullandığım için biraz zorlandım. Çünkü projeye başlamadan önce neyi nasıl yapacağıma dair bir taslağım yoktu. Bu sebeplerden dolayı bazı eksiklerimi sonradan farkettim bu yüzden geri dönmem hem zor hem de çok uğraştırıcı oldu. Bu projeyi yapacaklara tavsiyem, projeye başlamadan önce kullanacakları araç ve gereçleri iyi tanımaları ve projeyi yaparken uyulması gereken bir taslak hazırlamalarıdır. Örneğin benim hatalarımdan biri giriş sayfasındadır. Bu projeyi tekrar yapmak isteyenler giriş sayfasını geliştirebilir. Takip edilenlerin gösterildiği sayfaya birkaç ekleme yaparak daha da güzelleştirebilirler.

## 5 EKLER

### 5.1 Android Studio Kodları

#### 5.1.1 MainActivity.java

```
package com.lyadirga.neredesinbakalm;
```

```
import android.Manifest; import android.app.Activity; import android.content.Context;
import android.content.Intent; import android.content.pm.PackageManager; import and-
roid.location.Address; import android.location.Geocoder; import android.os.Build; im-
port android.os.Bundle;
```

```
import com.google.android.gms.tasks.OnFailureListener; import com.google.android.gms.tasks.OnSu-
import com.google.android.material.bottomsheet.BottomSheetBehavior; import com.google.android.m-
import com.google.android.material.snackbar.Snackbar; import com.google.android.material.textfield-
import com.google.android.material.textfield.TextInputLayout; import com.google.firebase.auth.Fireb-
import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.auth.FirebaseAuthUser-
import com.google.firebase.database.ChildEventListener; import com.google.firebase.database.Data-
import com.google.firebase.database.DataSnapshot; import com.google.firebase.database.DatabaseEr-
import com.google.firebase.database.DatabaseReference; import com.google.firebase.database.Fireba-
import com.google.firebase.database.Query; import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
```

```
import androidx.annotation.NonNull; import androidx.annotation.Nullable; import and-
roidx.appcompat.app.AppCompatActivity; import androidx.appcompat.widget.Toolbar; im-
port androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat;
```

```
import android.text.TextUtils; import android.util.Log; import android.view.View;
```

```
import android.view.Menu; import android.view.MenuItem; import android.view.ViewGroup;
import android.widget.BaseAdapter; import android.widget.ImageView; import android.widget.Linea-
import android.widget.ListView; import android.widget.QuickContactBadge; import and-
roid.widget.TextView; import android.widget.Toast;
```



```

import java.io.IOException; import java.text.SimpleDateFormat; import java.util.ArrayList;
import java.util.Date; import java.util.List; import java.util.Locale;

public class MainActivity extends AppCompatActivity public static final int ALARM_MUHRU =
1453; private BottomSheetBehavior bottomSheetBehavior; private DatabaseReference userDa
Takip > takipList; private TakipListesiAdapter adapter;

@Override public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) super.onRequestPermissionsResult(requestCode,
permissions, grantResults);

if (grantResults.length > 0 grantResults[0] == PackageManager.PERMISSION_GRANTED)

if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED)

Toast.makeText(this, "izin alındı", Toast.LENGTH_SHORT).show();

@Override protected void onCreate(Bundle savedInstanceState) super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

Toolbar toolbar = findViewById(R.id.toolbar); setSupportActionBar(toolbar); textInput-
Layout = findViewById(R.id.til); kullaniciAdi = findViewById(R.id.takip_kullaniciadi); listView =
findViewById(R.id.takipListesi); View view = findViewById(R.id.bottom_sheet); bottomSheetBehavior
BottomSheetBehavior.from(view); bottomSheetBehavior.setState(BottomSheetBehavior.STATE_OPEN);

takipList = new ArrayList<>(); mAuth = FirebaseAuth.getInstance(); user = mAuth.getCurrentUser();
adapter = new TakipListesiAdapter(this, takipList); listView.setAdapter(adapter);

if (user == null) startActivity(new Intent(this, GirisKayitActivity.class)); finish(); return;

userDatabase = FirebaseDatabase.getInstance().getReference().child("Kullanıcılar"); ta-
kipDatabase = FirebaseDatabase.getInstance().getReference().child("Takipleyenler"); ko-
numlaraDatabase = FirebaseDatabase.getInstance().getReference().child("Konumlar");

//Android M ve üzeri için lokasyon izin alma if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

```

```

//Takip edilenlerin listesi takipDatabase.orderByChild("takipEden").equalTo(user.getId())
.addValueEventListener(new ValueEventListener() @Override public void onDataChange(@NonNull
DataSnapshot snapshot)

takipList.clear(); for (DataSnapshot snapshot1 : snapshot.getChildren()) Takip takip =
snapshot1.getValue(Takip.class); takipList.add(takip); findViewById(R.id.progress).setVisibility(View
adapter.notifyDataSetChanged());

@Override public void onCancelled(@NonNull DatabaseError error)

);

FloatingActionButton fab = findViewById(R.id.fab); fab.setOnClickListener(new View.OnClickListener()
@Override public void onClick(View view) bottomSheetBehavior.setState(BottomSheetBehavior.STATE
);

bottomSheetBehavior.setBottomSheetCallback(new BottomSheetBehavior.BottomSheetCallback()
@Override public void onStateChanged(@NonNull View bottomSheet, int newState) switch
(newState) case BottomSheetBehavior.STATE_COLLAPSED : //bottomSheetBehavior.setState
90

break; case BottomSheetBehavior.STATE_EXPANDED : fab.hide(); break; case BottomSheetBehavior
fab.show(); break; case BottomSheetBehavior.STATE_SETTING :

break;

@Override public void onSlide(@NonNull View bottomSheet, float slideOffset)

);

@Override public boolean onCreateOptionsMenu(Menu menu) // Inflate the menu; this
adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.menu_main, menu); return

@Override public boolean onOptionsItemSelected(MenuItem item) // Handle action bar
item clicks here. The action bar will // automatically handle clicks on the Home/Up button,

```

```

so long // as you specify a parent activity in AndroidManifest.xml. int id = item.getItemId();

//noinspection SimplifiableIfStatement if (id == R.id.action_ettings)startActivity(newIntent(Main

if (id == R.id.Exit) mAuth.signOut(); startActivity(new Intent(MainActivity.this, Giriş-
KayıtActivity.class)); finish(); return true;

return super.onOptionsItemSelected(item);

public void clickEkle(View view)

textInputLayout.setError(null);

if (TextUtils.isEmpty(kullaniciAdi.getText())) textInputLayout.setError("kullanıcı adınızı
giriniz"); return;

String kullaniciadi = kullaniciAdi.getText().toString(); Query queryRef = userDatabase.orderByChild

if (snapshot.exists()) String ad = snapshot.child("kullaniciadi").getValue(String.class); Stringid =
snapshot.child("kullaniciid").getValue(String.class); Stringurl = snapshot.child("profilurl")
" kullanıcısı bulundu", Toast.LENGTH_SHORT).show(); takipEkle(user.getUid(), snapshot.

@Override public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable
String previousChildName)

@Override public void onChildRemoved(@NonNull DataSnapshot snapshot)

@Override public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable
String previousChildName)

@Override public void onCancelled(@NonNull DatabaseError error)

);

private void takipEkle(String takipEden, String takipEdilen) Takip takip = new Takip(takipEden,
takipEdilen); takipDatabase.push().setValue(takip).addOnSuccessListener(new OnSuccess-
Listener<Void>() @Override public void onSuccess(Void aVoid) Toast.makeText(MainActivity.this,

```

```
"Başarılı", Toast.LENGTH_SHORT).show();).addOnFailureListener(new OnFailureListener()
```

```
private class TakipListesiAdapter extends BaseAdapter private final Context context; private final List<Takip> takipList;
```

```
public TakipListesiAdapter(Context c, List<Takip> takipList) this.context = c; this.takipList = takipList;
```

```
@Override public int getCount() return takipList.size();
```

```
@Override public Object getItem(int position) if (takipList.size() == 0) return null;
```

```
return takipList.get(position);
```

```
@Override public long getItemId(int position) return 0;
```

```
@Override public View getView(int position, View convertView, ViewGroup parent) if (takipList.size() == 0) return null;
```

```
LinearLayout container = (LinearLayout) ((Activity) context).getLayoutInflater().inflate(R.layout.takip,
```

```
Takip takip = takipList.get(position);
```

```
final TextView ad = container.findViewById(R.id.ad); final TextView zaman = container.findViewById(R.id.zaman); final TextView adresler = container.findViewById(R.id.adresler);  
//final TextView uzaklık final ImageView profil = container.findViewById(R.id.profilItem);
```

```
final Konum[] konum = new Konum[1]; final String[] url = new String[1];
```

```
//kullanıcı adı ve profil resmini set etme
```

```
userDatabase.child(takip.getTakipEdilen()).addValueEventListener(new ValueEventListener() @Override public void onDataChange(@NonNull DataSnapshot snapshot) if (snapshot.exists()) String kullanicadi = snapshot.child("kullanicadi").getValue(String.class); url[0] = snapshot.child("profilurl").getValue(String.class); ad.setText(kullanicadi); Picasso.get().load
```

```
@Override public void onCancelled(@NonNull DatabaseError error) Toast.makeText(context,
```

```

error.getMessage(), Toast.LENGTH_SHORT).show();

);

//Tarih saat ve adres bilgisini set etme işlemi
konumlaraDatabase.child(takip.getTakipEdilen()).orderBy
ValueEventListener() @Override public void onDataChange(@NonNull DataSnapshot
snapshot) Log.d("Konumlar",snapshot.toString());

for (DataSnapshot snapshot1:snapshot.getChildren()) konum[0] = snapshot1.getValue(Konum.class);
setKonumBilgisi (konum[0]);

private void setKonumBilgisi(Konum konum)

SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy HH:mm"); Date date
= new Date(konum.getZaman()); zaman.setText(format.format(date));

String adres = ""; Geocoder geocoder = new Geocoder(context, Locale.getDefault());

try List<Address> addressList = geocoder.getFromLocation(konum.getEnlem(), konum.getBoylam()
1); for (Address adr : addressList) Log.d("Adres", "setKonumBilgisi: " + adr.toString());
adres += adr.getAddressLine(0); for (int i = 0; i < adr.getMaxAddressLineIndex(); i++)
//adres+=adr.getAddressLine(i)+" , "; catch (IOException e) e.printStackTrace();

adresler.setText(adres);

/*//Uzaklık bilgisi hesaplama
Location location = new Location(""); location.setLatitude(konum.getEnlem());
location.setLongitude(konum.getBoylam());

Location mevcutKonum = new Location(""); location.setLatitude(enlem); location.setLongitude(boylam);
Log.d("Konumlar", "setKonumBilgisi: "+enlem+" "+boylam+" "+position+" konum"+konum.getEnlem()
"+konum.getBoylam());

float[] distance = new float[2]; Location.distanceBetween(enlem, boylam, konum.getEnlem(),
konum.getBoylam(), distance); int m = (int) distance[0]; //uzaklik.setText(m + " metre");

try int mesafe = (int) mevcutKonum.distanceTo(location); uzaklik.setText(mesafe + "

```

```

metre");

catch (Exception e) Log.e("Hata", "setKonumBilgisi: " + e.getMessage());

*/

@Override public void onCancelled(@NonNull DatabaseError error) //Hata );

container.setOnClickListener(new View.OnClickListener() @Override public void onC-
lick(View v) //MapsActivity açılacak Intent intent = new Intent(context,MapsActivity.class);
intent.putExtra("enlem",konum[0].getEnlem()); intent.putExtra("boylam",konum[0].getBoylam());
intent.putExtra("profil_url",url[0]); intent.putExtra("kullaniciadi",ad.getText().toString()); con
);

return container;

```

### 5.1.2 GirişKayıtActivity.java

```

package com.lyadirga.neredesinbakalm;

import androidx.annotation.NonNull; import androidx.annotation.Nullable; import and-
roidx.appcompat.app.AppCompatActivity;

import android.content.ContentResolver; import android.content.Intent; import android.graphics.Bitmap
import android.net.Uri; import android.os.Bundle; import android.provider.MediaStore;
import android.text.TextUtils; import android.util.Log; import android.view.Menu; im-
port android.view.MenuItem; import android.view.View; import android.webkit.MimeTypeMap;
import android.widget.Button; import android.widget.ImageView; import android.widget.ProgressBar
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener; import com.google.android.gms.tasks.On
import com.google.android.gms.tasks.Task; import com.google.android.material.textfield.TextInputE
import com.google.android.material.textfield.TextInputLayout; import com.google.firebase.auth.Auth
import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.auth.FirebaseUser;

```

```

import com.google.firebase.database.DataSnapshot; import com.google.firebase.database.DatabaseEr
import com.google.firebase.database.DatabaseReference; import com.google.firebase.database.Fireba
import com.google.firebase.database.ValueEventListener; import com.google.firebase.storage.Firebase
import com.google.firebase.storage.StorageReference; import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

public class GirisKayitActivity<valid> extends AppCompatActivity

private TextInputLayout tilEmail, tilSifre, tilKullaniciAdi; private TextInputEditText email,
sifre, kullaniciadi; private ProgressBar progresBarCircle; private ImageView profilPhoto;
private Uri profilPhotoUri = null; private static final int RESIM_SEC = 1;

private FirebaseAuth mAuth; private StorageReference mStorageRef; private Database-
Reference databaseReference; private ValueEventListener databaseEventListener; private
boolean profildegisecek = false;

@Override protected void onStart() super.onStart();

FirebaseUser user = mAuth.getCurrentUser(); if (user != null) progresBarCircle.setVisibility(View.VI
updateUI(user);

@Override protected void onDestroy() super.onDestroy();

if (databaseReference!=null databaseEventListener!=null) databaseReference.removeEventListener(c

private void updateUI(FirebaseUser user) if (user == null) kullaniciadi.setText(null); pro-
filPhoto.setImageResource(R.drawable.avatar); email.setText(null); sifre.setText(null); re-
turn;

if (databaseReference == null) databaseReference = FirebaseDatabase.getInstance().getReference().c
.child(user.getUid());

((Button) findViewById(R.id.buttonGiris)).setText("Takip Listesine Git");

if (profildegisecek)

```

```
progresBarCircle.setVisibility(View.GONE); return;
```

```
databaseEventListener = databaseReference.addValueEventListener(new ValueEventListener() @Override public void onDataChange(@NonNull DataSnapshot snapshot) progresBarCircle.setVisibility(View.GONE);
```

```
String ad = snapshot.child("kullaniciadi").getValue(String.class); Stringurl = snapshot.child("p
```

```
kullaniciadi.setText(ad); Picasso.get().load(url).into(profilPhoto);
```

```
@Override public void onCancelled(@NonNull DatabaseError error) progresBarCircle.setVisibility(  
);
```

```
@Override protected void onCreate(Bundle savedInstanceState) super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_giris_kayit);
```

```
mAuth = FirebaseAuth.getInstance(); mStorageRef = FirebaseStorage.getInstance().getReference();
```

```
tilEmail = findViewById(R.id.tilEmail); email = findViewById(R.id.email);
```

```
tilKullaniciAdi = findViewById(R.id.tilKullaniciAdi); kullaniciadi = findViewById(R.id.kullaniciadi)
```

```
tilSifre = findViewById(R.id.tilSifre); sifre = findViewById(R.id.sifre);
```

```
progresBarCircle = findViewById(R.id.progresBarCircle); profilPhoto = findViewById(R.id.profil);
```

```
if (mAuth.getCurrentUser() != null) databaseReference = FirebaseDatabase.getInstance().getReference(  
.child(mAuth.getCurrentUser().getUid());
```

```
profilPhoto.setOnClickListener(new View.OnClickListener() @Override public void onC-  
lick(View v) //Intent intent2 = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); //startActiv  
newIntent(Intent.ACTION_PICK); intent.setType("image/*"); intent.setAction(Intent.AC  
true;  
);
```

```
@Override protected void onActivityResult(int requestCode, int resultCode, @Nullable
```



```

Intent data) super.onActivityResult(requestCode, resultCode, data);

if (requestCode == RESIM_5E_CresultCode == RESULT_OK data != null data.getData() !=
null) //databaseReference.removeEventListener(databaseEventListener); profilPhotoUri =

//Bitmap bitmap = (Bitmap) data.getExtras().get("data"); //profilPhoto.setImageBitmap(bitmap);

public boolean onCreateOptionsMenu(Menu menu) menu.add(0, 1, Menu.NONE, "Çıkış").setIcon(R

@Override public boolean onOptionsItemSelected(MenuItem item) mAuth.signOut();
updateUI(null); return super.onOptionsItemSelected(item);

public void profilKaydet(View view)

tilKullaniciAdi.setError(null);

if (profilPhotoUri == null || TextUtils.isEmpty(kullaniciadi.getText()))

if (profilPhotoUri == null) Toast.makeText(this, "Lütfen profil fotoğrafınızı seçiniz...",
Toast.LENGTH_SHORT).show();

if (TextUtils.isEmpty(kullaniciadi.getText())) tilKullaniciAdi.setError("Lütfen kullanıcı adı-
nızı giriniz");

return;

//Firebase profil bilgisi kayıt progressBarCircle.setVisibility(View.VISIBLE);

String kullaniciAdi = kullaniciadi.getText().toString(); String uzanti = getFileExtension(profilPhotoUri

StorageReference childRef = mStorageRef.child("KullaniciProfil").child(mAuth.getCurrentUser().get

.child(mAuth.getCurrentUser().getUid()).child(kullaniciAdi + "." + uzanti); childRef.putFile(profilPhoto

OnSuccessListener<UploadTask.TaskSnapshot>() @Override public void onSuccess(UploadTask.TaskSnap
taskSnapshot)

Task<Uri> result = taskSnapshot.getMetadata().getReference().getDownloadUrl(); result.addOnSuccess

```

```

OnSuccessListener<Uri>() @Override public void onSuccess(Uri uri) databaseReference.child("kull
profildegisecek=false; );

Toast.makeText(GirisKayitActivity.this, "Yuppi", Toast.LENGTH_SHORT).show(); progressBarCi

private String getFileExtension(Uri uri) ContentResolver contentResolver = getContent-
Resolver(); MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton(); return mi-
mimeTypeMap.getExtensionFromMimeType(contentResolver.getType(uri));

public void kayitOl(View view) if (!validateForm()) return;

progressBarCircle.setVisibility(View.VISIBLE);

 mAuth.createUserWithEmailAndPassword(email.getText().toString(), sifre.getText().toString())
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() @Override pub-
lic void onComplete(@NonNull Task<AuthResult> task) progressBarCircle.setVisibility(View.GONE
if (task.isSuccessful()) // Sign in success, update UI with the signed-in user's information
Toast.makeText(GirisKayitActivity.this, "Kayıt başarılı", Toast.LENGTH_SHORT).show(); ((Button

else Toast.makeText(GirisKayitActivity.this, "Authentication failed. " + task.getException(),
Toast.LENGTH_SHORT).show(); Log.w("TAG", "createUserWithEmail : failure", task.getE

);

public void girisYap(View view) if (mAuth.getCurrentUser() != null) startActivity(new
Intent(this, MainActivity.class)); finish(); return;

if (!validateForm()) return;

progressBarCircle.setVisibility(View.VISIBLE);

 mAuth.signInWithEmailAndPassword(email.getText().toString(), sifre.getText().toString()).addOnC
new OnCompleteListener<AuthResult>() @Override public void onComplete(@NonNull
Task<AuthResult> task)

progressBarCircle.setVisibility(View.GONE);

```

```

if (task.isSuccessful()) FirebaseUser user = mAuth.getCurrentUser(); updateUI(user);
Toast.makeText(GirisKayitActivity.this, "Giriş başarılı", Toast.LENGTH_SHORT).show(); ((Button

private boolean validateForm() boolean valid = true; tilEmail.setError(null); tilSifre.setError(null);

if (TextUtils.isEmpty(email.getText()) || TextUtils.isEmpty(sifre.getText())) if (TextU-
tils.isEmpty(email.getText())) tilEmail.setError("Lütfen mail adresinizi giriniz"); valid
= false; else if (!email.getText().toString().contains("@")) tilEmail.setError("Lütfen ge-
çerli bir mail adresi giriniz"); valid = false;

if (TextUtils.isEmpty(sifre.getText())) tilSifre.setError("Lütfen şifrenizi giriniz"); valid =
false;

return valid;

```

### 5.1.3 Konum.java

```

package com.lyadirga.neredesinbakalm;

class Konum

private Double enlem,boylam; private Long zaman;

public Konum()

public Double getEnlem() return enlem;

public void setEnlem(Double enlem) this.enlem = enlem;

public Double getBoylam() return boylam;

public void setBoylam(Double boylam) this.boylam = boylam;

public Long getZaman() return zaman;

public void setZaman(Long zaman) this.zaman = zaman;

```

#### 5.1.4 KonumServisi.java

```
package com.lyadirga.neredesinbakalm;

import android.Manifest; import android.annotation.SuppressLint; import android.app.AlarmManager;
import android.app.PendingIntent; import android.app.Service; import android.content.Context;
import android.content.Intent; import android.content.pm.PackageManager; import and-
roid.location.Criteria; import android.location.Location; import android.location.LocationListener;
import android.location.LocationManager; import android.os.Build; import android.os.Build.VERSION;

import androidx.annotation.NonNull; import androidx.annotation.Nullable; import and-
roidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat;

import com.google.android.gms.tasks.OnFailureListener; import com.google.android.gms.tasks.OnSu-
import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.database.DatabaseReferen-
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap; import java.util.Map;

public class KonumServisi extends Service

private LocationManager locationManager; private LocationListener locationListener; pri-
vate boolean lokasyonBilgisiAlindi = false; private FirebaseAuth mAuth; private Databa-
seReference konumDatabase;

@Nullable @Override public IBinder onBind(Intent intent) return null;

@Override public void onCreate() super.onCreate(); mAuth = FirebaseAuth.getInstance();
konumDatabase = FirebaseDatabase.getInstance().getReference().child("Konumlar").child(mAuth.ge

locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE); loca-
new LocationListener()@Override public void onLocationChanged(Location location) if (!lokasy-

lokasyonBilgisiKaydet(location.getLatitude(),location.getLongitude());

;
```

```

private void lokasyonBilgisiKaydet(double enlem, double boylam)

lokasyonBilgisiAlindi = true;

Map<String, Object> map = new HashMap<>(); map.put("enlem", enlem); map.put("boylam", boylam);
map.put("zaman", System.currentTimeMillis());

konumDatabase.push().setValue(map).addOnSuccessListener(new OnSuccessListener<Void>()
@Override public void onSuccess(Void aVoid) alarmAyarla(); stopSelf();

).addOnFailureListener(new OnFailureListener() @Override public void onFailure(@NonNull
Exception e) lokasyonBilgisiAlindi = false; Log.e("Hata", e.getMessage());

);

private void alarmAyarla() AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
new Intent(this, KonumServisi.class); PendingIntent pi = PendingIntent.getService(this, M
if (Build.VERSION.SDK_INT >= VERSION_CODES.KITKAT) alarmManager.setExact(AlarmManager.ALARM_TYPE_RINGTONE, pi,
else alarmManager.set(AlarmManager.RTC_WAKEUP, System.currentTimeMillis() +
600000, pi);

@Override public void onDestroy() super.onDestroy(); if (locationManager != null) lo-
cationManager.removeUpdates(locationListener); locationManager = null;

@SuppressLint("MissingPermission") @Override public int onStartCommand(Intent in-
tent, int flags, int startId)

Criteria criteria = new Criteria(); criteria.setPowerRequirement(Criteria.POWER_MEDIUM); criteria.setAccuracyRequirement(Criteria.ACCURACY_FINE);
locationManager.getBestProvider(criteria, true);

if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) // Konum isteğinden önceki versiyonlar için
else

if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)

```

*PackageManager.PERMISSION\_GRANTED)/izin alınmamış.Modifikasyon mesajı gönder*

```
return super.onStartCommand(intent, flags, startId);
```

### **5.1.5 Takip.java**

```
package com.lyadirga.neredesinbakalm;
```

```
import com.google.firebase.database.Exclude;
```

```
public class Takip { private String takipEden,takipEdilen,key;
```

```
public Takip() //boş yapıcı metod
```

```
public Takip(String takipEden, String takipEdilen) {this.takipEden = takipEden; this.takipEdilen  
= takipEdilen;
```

```
public String getTakipEden() {return takipEden;
```

```
public void setTakipEden(String takipEden) {this.takipEden = takipEden;
```

```
public String getTakipEdilen() {return takipEdilen;
```

```
public void setTakipEdilen(String takipEdilen) {this.takipEdilen = takipEdilen;
```

```
@Exclude
```

```
public String getKey() {return key;
```

```
public void setKey(String key) {this.key = key;
```

### **5.1.6 MapsActivity.java**

```
package com.lyadirga.neredesinbakalm;
```

```
import androidx.fragment.app.FragmentActivity;
```

```

import android.annotation.SuppressLint; import android.graphics.Color; import android.os.Bundle;
import android.view.Gravity; import android.view.View; import android.widget.ImageView;
import android.widget.LinearLayout; import android.widget.TextView; import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory; import com.google.android.gms.maps.
import com.google.android.gms.maps.OnMapReadyCallback; import com.google.android.gms.maps.
import com.google.android.gms.maps.model.BitmapDescriptorFactory; import com.google.android.g
import com.google.android.gms.maps.model.Marker; import com.google.android.gms.maps.model.M
import com.squareup.picasso.Callback; import com.squareup.picasso.Picasso;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback

private GoogleMap mMap; private double enlem, boylam; private String kullaniciAdi,
profilUrl;

@Override protected void onCreate(Bundle savedInstanceState) super.onCreate(savedInstanceState);
setContentView(R.layout.activity_maps); //ObtaintheSupportMapFragmentandgetnotifiedwhen
(SupportMapFragment)getSupportFragmentManager().findFragmentById(R.id.map); m
getIntent().getDoubleExtra("enlem", 0); boylam = getIntent().getDoubleExtra("boylam", 0)
getIntent().getStringExtra("kullaniciadi"); profilUrl = getIntent().getStringExtra("profi

/** * Manipulates the map once available. * This callback is triggered when the map is
ready to be used. * This is where we can add markers or lines, add listeners or move the
camera. In this case, * we just add a marker near Sydney, Australia. * If Google Play
services is not installed on the device, the user will be prompted to install * it inside the
SupportMapFragment. This method will only be triggered once the user has * installed
Google Play services and returned to the app. */ @SuppressWarnings("MissingPermission")
@Override public void onMapReady(GoogleMap googleMap) mMap = googleMap;

LatLng takipEdilen = new LatLng(enlem, boylam); mMap.addMarker(new MarkerOptions().position

mMap.setMapType(GoogleMap.MAP_TY_PENORMAL); mMap.setMyLocationEnabled(true);

```

## KAYNAKLAR

- [1] <https://developer.android.com/reference/android/app/Activity/>
- [2] [https://www.bilgigunlugum.net/prog/android/android\\_aktivite](https://www.bilgigunlugum.net/prog/android/android_aktivite)
- [3] <https://github.com/firebase/quickstart-android>
- [4] <https://developers.google.com/maps/documentation/android-sdk/start>
- [5] <https://www.mehmetkirazli.com/android-google-maps-api-kullanimi>
- [6] <https://github.com/android>
- [7] <https://umiitkose.com/2016/09/android-kutuphaneleri-1-picasso/>
- [8] <https://medium.com/android-t%C3%BCrkiye/etkili-layout-se%C3%A7imi-ve-kullan%C4%B1m%C4%B1-a2cf833ac85d>