

**Bölüm Adı: Elektrik ve Otomasyon**

**Program Adı: Mobil Teknolojileri**

**Ders Adı: Android Uygulama Geliştirme**

**Proje Konusu: Personel bilgileri kayıt sistemi**

**19MY93004**

**Bilal Mercan**

İçindekiler

[Uygulama Senaryosu 3](#_Toc62029652)

[İzinler 4](#_Toc62029653)

[Manifest Dosyası 4](#_Toc62029654)

[Gradle Dosyası 5](#_Toc62029655)

[Button 6](#_Toc62029656)

[Button dizaynı 6](#_Toc62029657)

[Sayfa geçişleri 7](#_Toc62029658)

[Alert Dialog 8](#_Toc62029659)

[Kamera 9](#_Toc62029660)

[Gridview 10](#_Toc62029661)

[Media Player ve Progressbar 11](#_Toc62029662)

[Local Notification 12](#_Toc62029663)

[Not Kaydetme 13](#_Toc62029664)

[Api 14](#_Toc62029665)

[Sqlite Veri Tabanı Crud (Oluşturma, Okuma, Güncelleme, Silme) İşlemleri 16](#_Toc62029666)

[Tablo Oluşturma 16](#_Toc62029667)

[Tabloya veri ekleme 18](#_Toc62029668)

[Tabloda veri silme 19](#_Toc62029669)

[Tabloda veri listeleme 20](#_Toc62029670)

[Tabloda veri güncelleme 21](#_Toc62029671)

[Veri tabanı sorguları testi 22](#_Toc62029672)

[Ekle Butonu 23](#_Toc62029673)

[Listele Butonu 24](#_Toc62029674)

[Güncelle ve Sil Butonu 25](#_Toc62029675)

# Uygulama Senaryosu

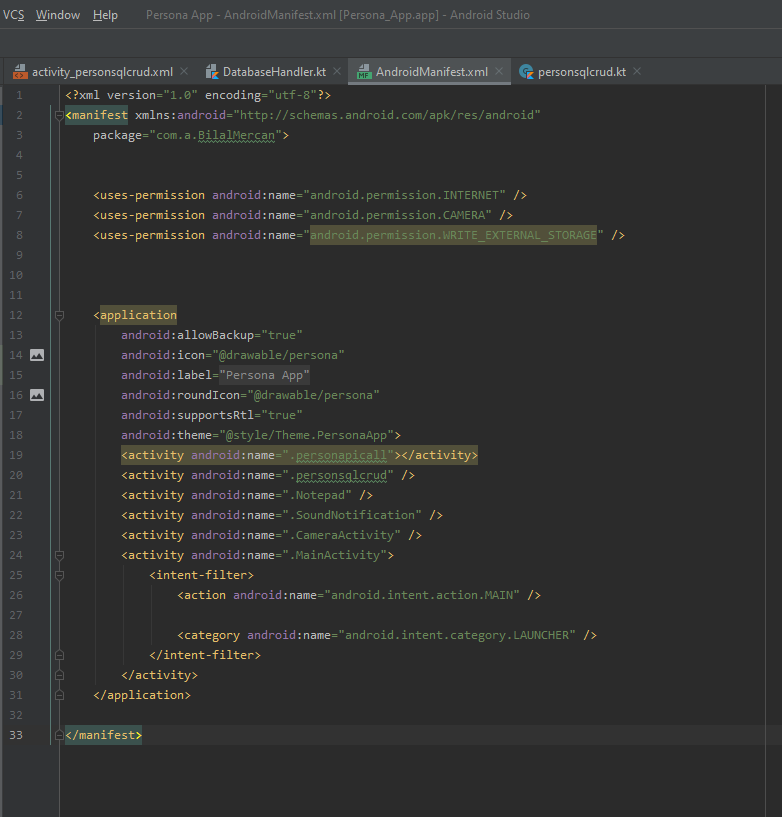
Uygulamada kullanıcı öncelikle kamerayı yüz kontrol için kullanabilir, çektiği fotoğrafları görebilir. Ses kontrol bölümünde uygulamaya özel bir sesi oynatabilir, isterse bildirim butonuna basarak uygulama bilgilerini görüntüleyebilir. Notlarını kaydedebilir. Hava durumunu (İstanbul) anlık olarak görebilir. Basit veri tabanı işlemlerini yapabilir.

Not: Uygulama Android Pie 9.0 sürümünde test edilmiştir.

# İzinler

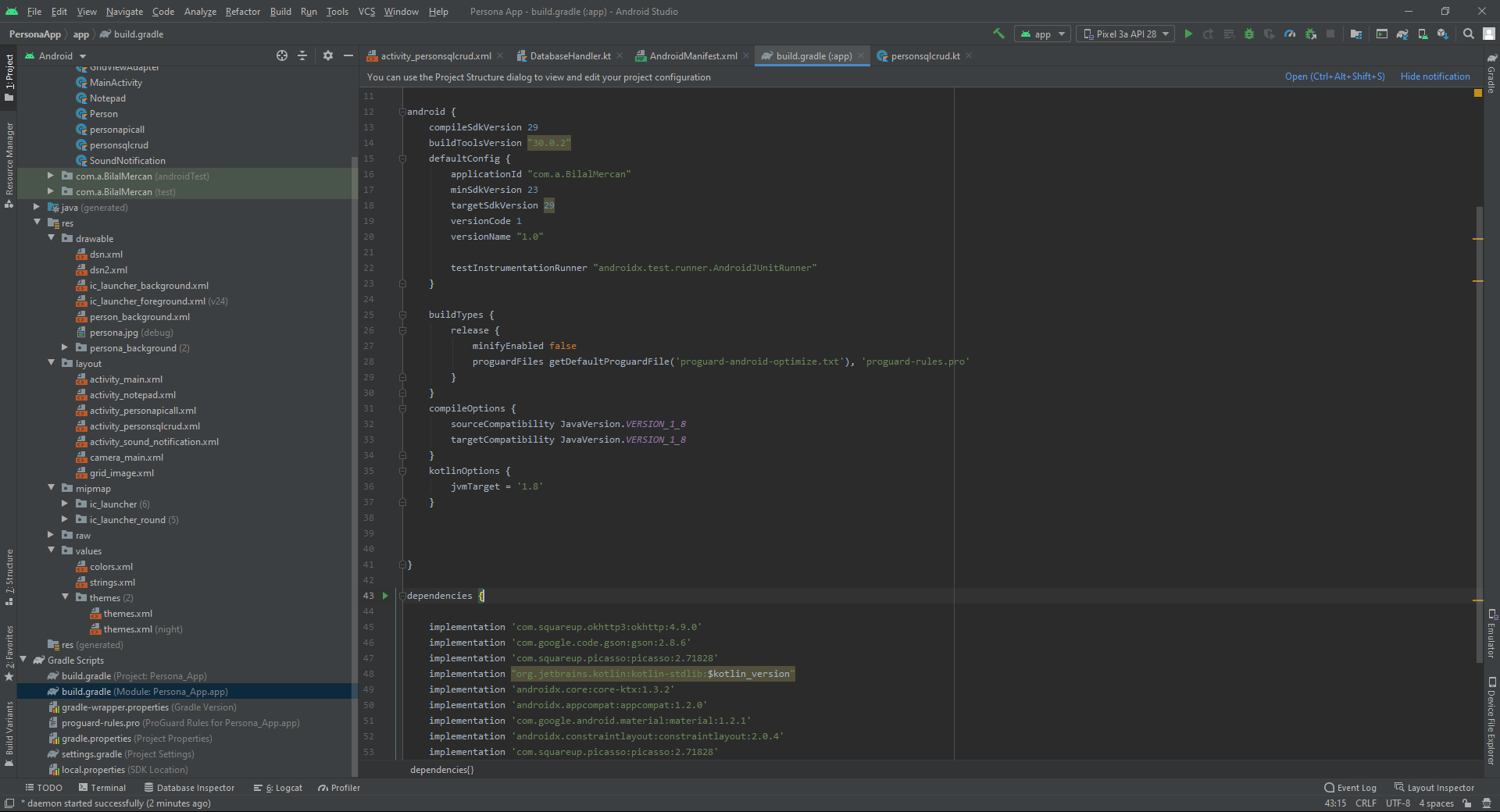
## Manifest Dosyası

Uygulamada internet, kamera ve depolama kullanılacağı için bu özelliklerin kullanılması için izinler alınıyor.



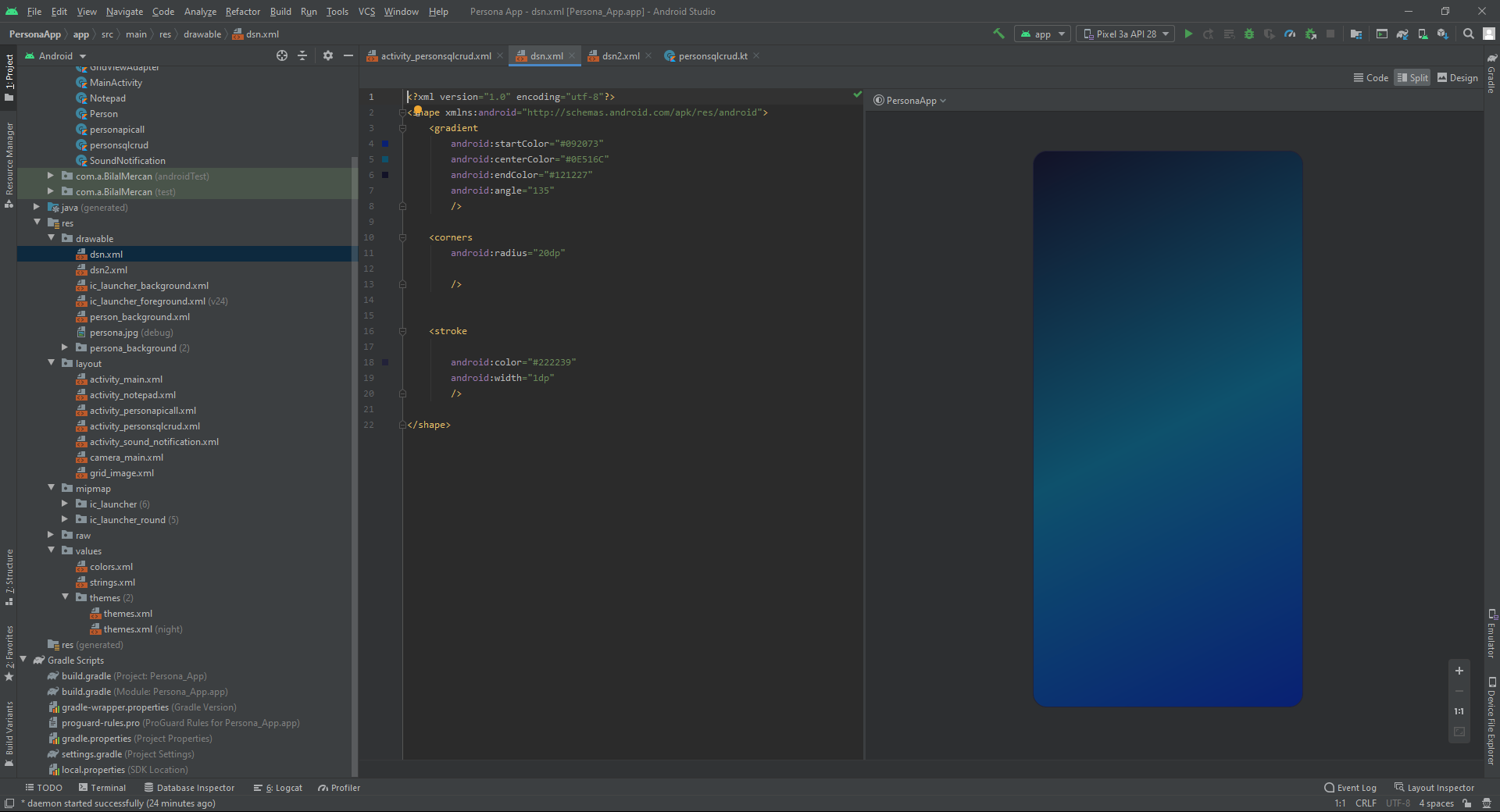
## Gradle Dosyası

İnternet için Gradle dosyasına okhttp, gson ve picasso bağımlılıkları ekleniyor. İcon ve round icon ile uygulama ikonunu ayarlanıyor. Gson kütüphanesi hava durumu bilgilerinin alınması için kullanılıyor.



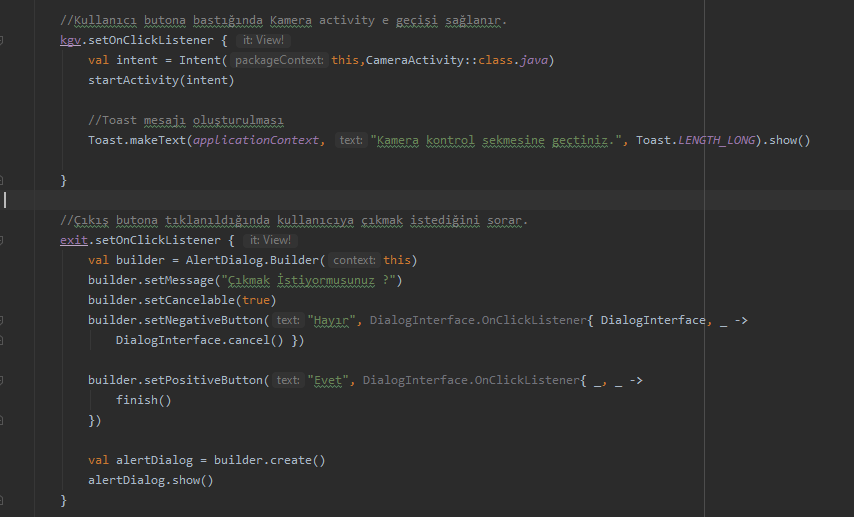
# Button

## Button dizaynı

Dizayn xml dosyaları drawable klasörüne yükleniyor. Xml dosyasında yer alan shape ile butona şekil verileceği belirtiliyor. Gradient tanımı renk geçişleri için kullanılıyor corner radius ile butonların bütün kenarlarından kırpılarak daha estetik gözükmesi sağlanıyor.

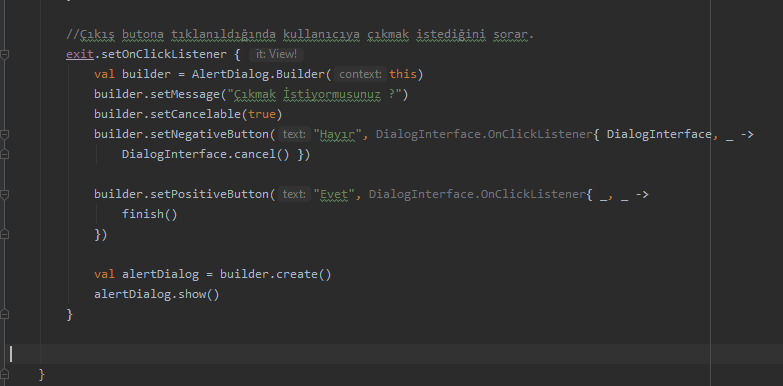
## Sayfa geçişleri

Çıkış butonu için ekstra Alertdialog ile kullanıcıya “çıkmak istiyor musunuz” sorusu yöneltiliyor.

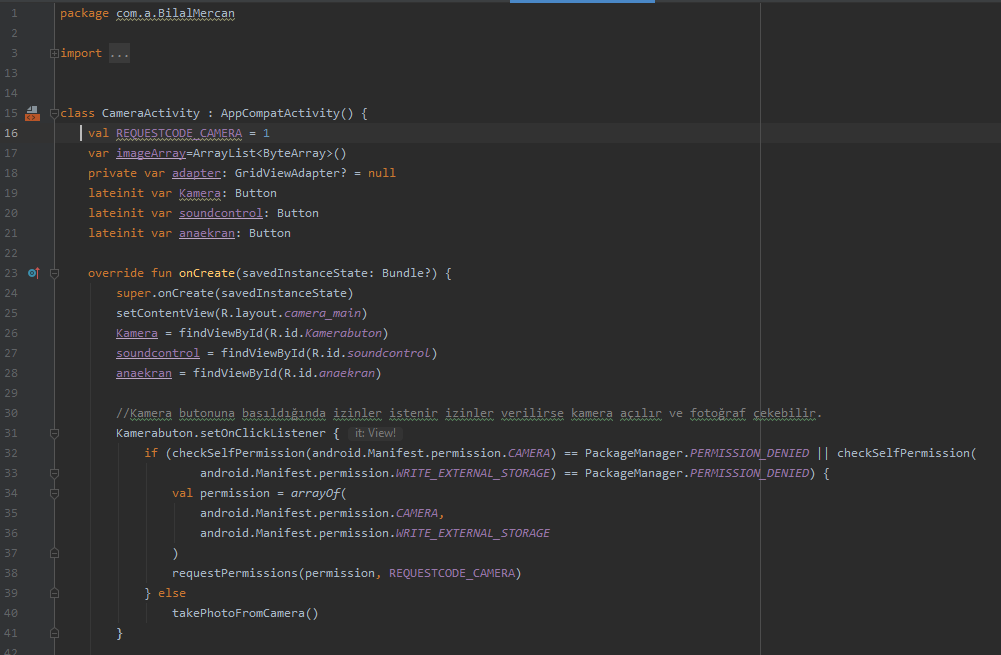


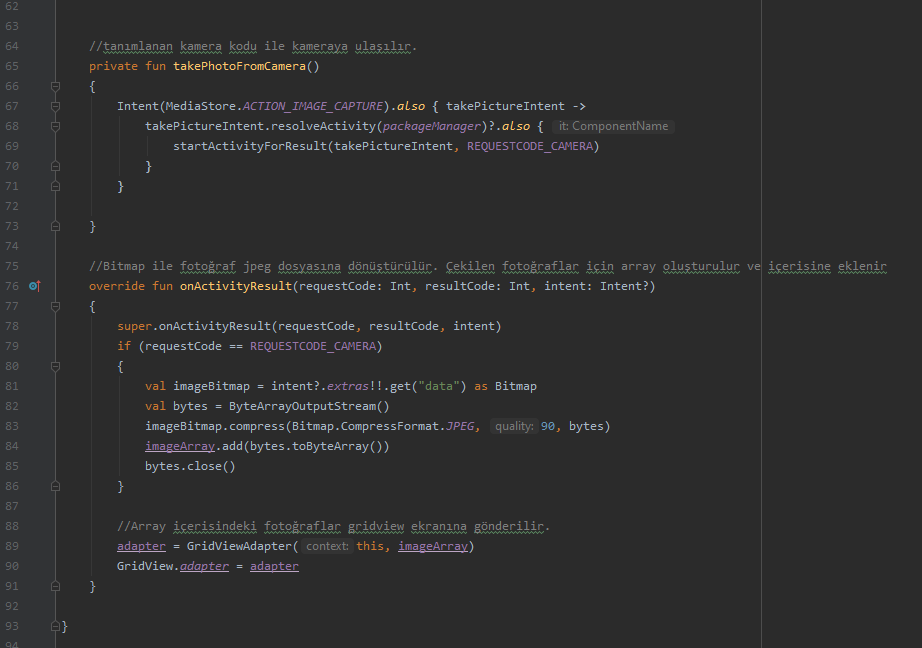
## Alert Dialog

Çıkış butonuna basıldığında kullanıcıya çıkmak istiyor musunuz sorusu yöneltilir. Hayır butonuna basınca dialog penceresi kapanır ve uygulamada kalır evet butonuna basınca finish ile uygulamadan çıkış yapar.



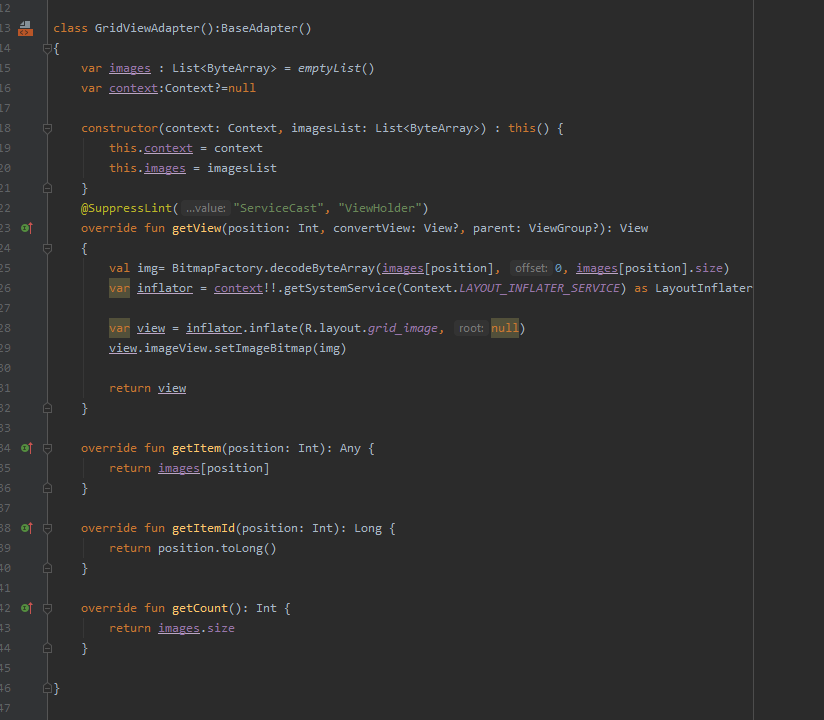
# Kamera

Kamerayı kullanabilmek için öncelikle kamera kodu oluşturulması ve arraylist oluşturulması gerekmektedir. Oluşturulan kod ve arraylist ile kod ile kameraya ulaşıp arraylist ile çekilen fotoğrafların array içerisine aktarılması sağlanır. Gridview eklenerek çekilen fotoğrafların düzenli gösterilmesi sağlanmıştır.



## Gridview

Kamerada çekilen fotoğrafların görüntülenmesi için kullanılmıştır.



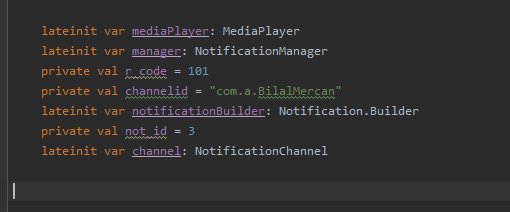
# Media Player ve Progressbar

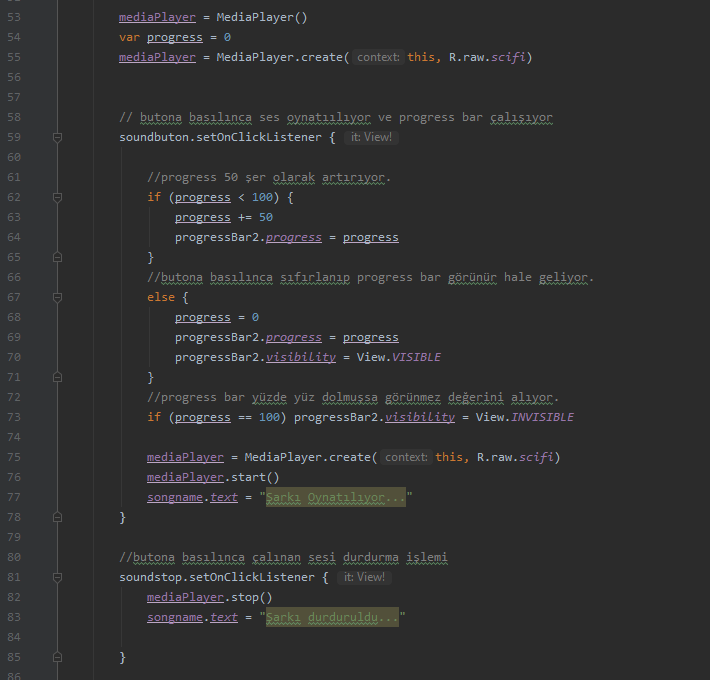
Media player ile birlikte Progressbar kullanılmakta.

Media player müzik, ses, video gibi medyaların oynatılmasını sağlar.

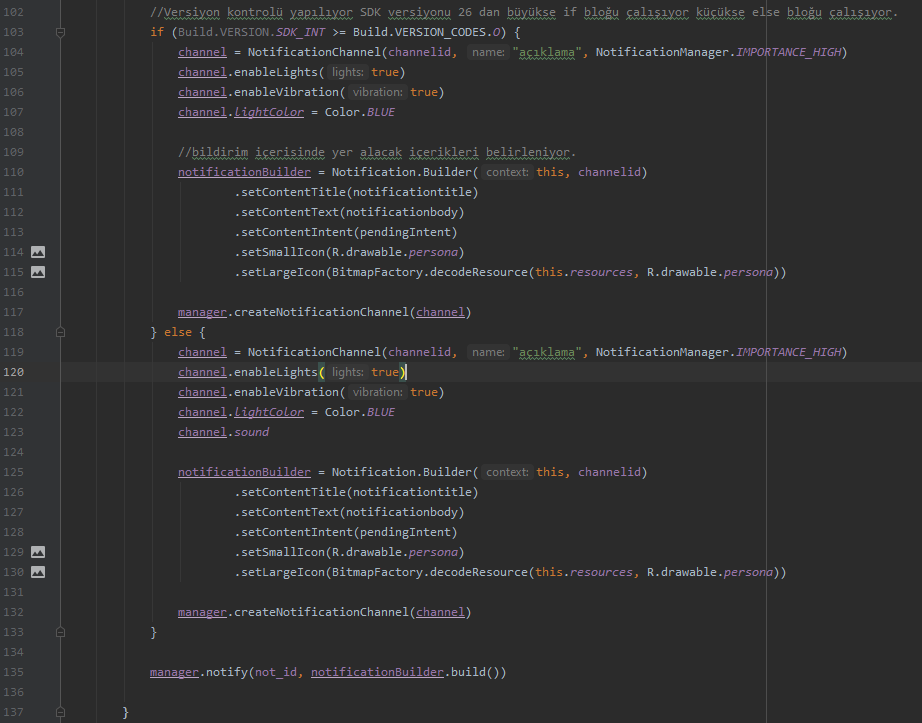
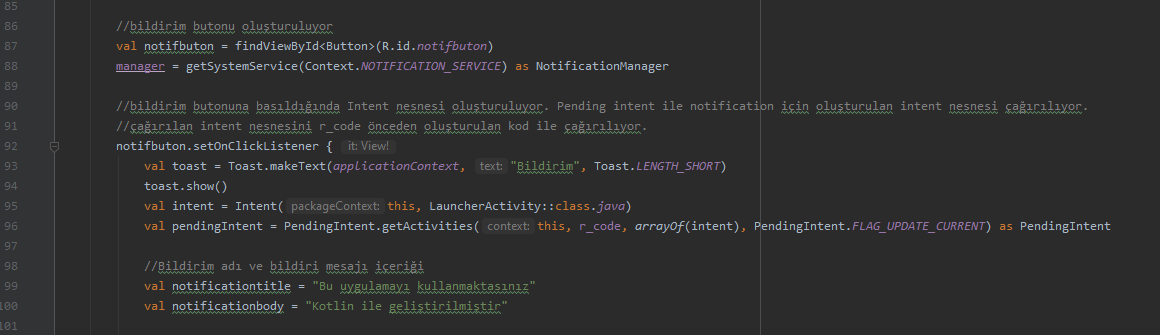
Progress bar ise, bir işlem yaparken tetiklenen, kullanıcıya bilgi verilmesi sağlanır.

Oluşturulan raw dosyasını kaynak dosyası olarak alıyor ve içerisindeki şarkıyı ulaşıyor. Butona basıldığında şarkının oynatılması ve Progressbarın tetiklenerek çalışması sağlanıyor.

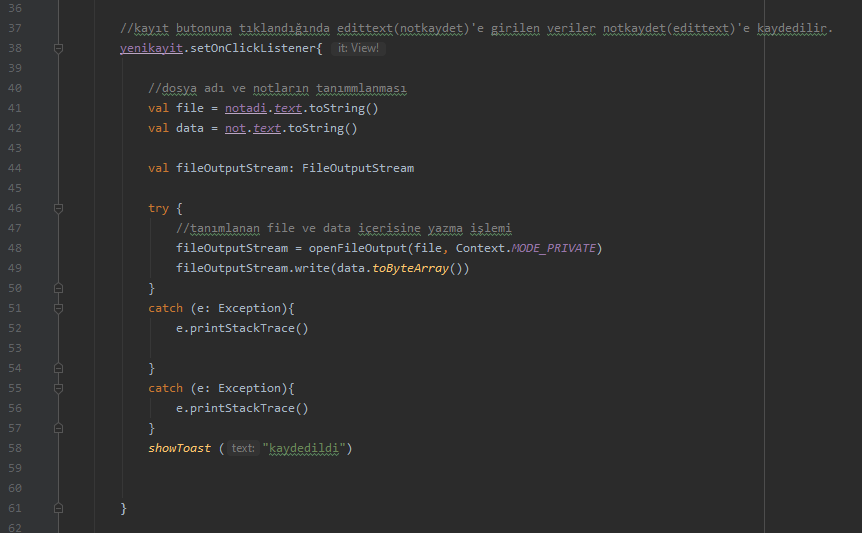


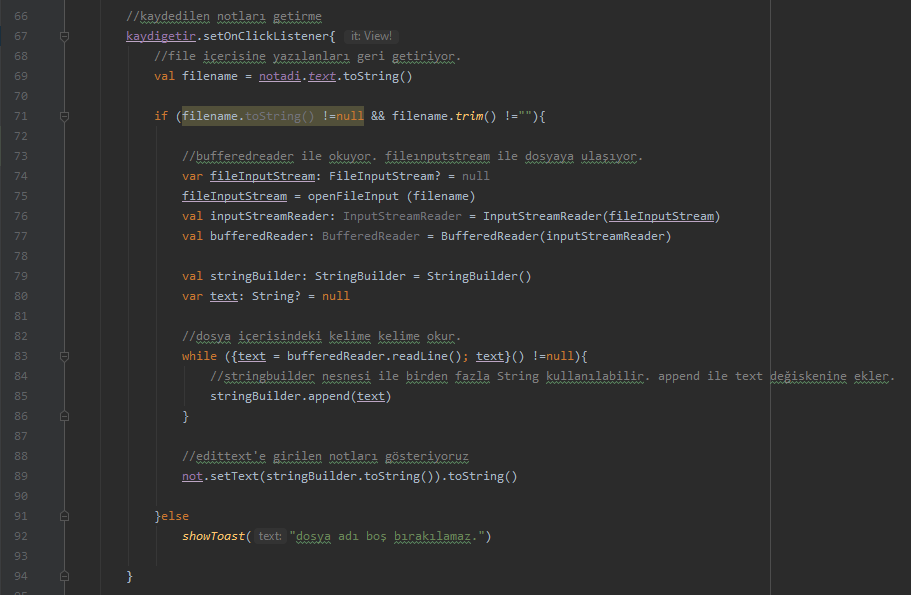


# Local Notification

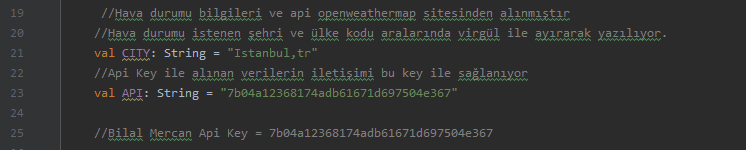
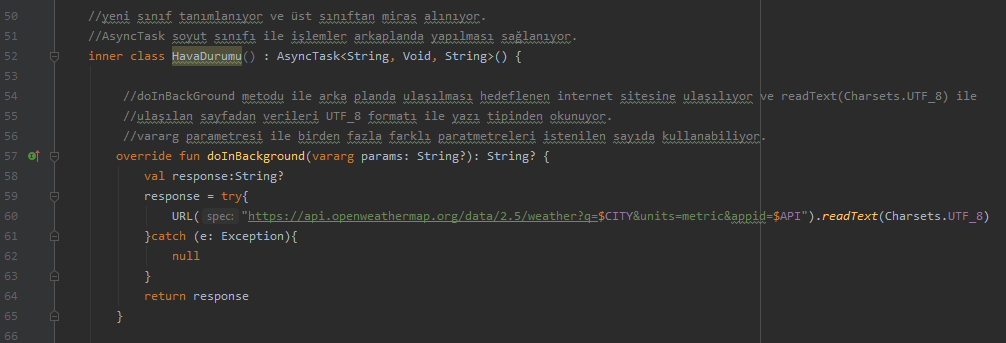
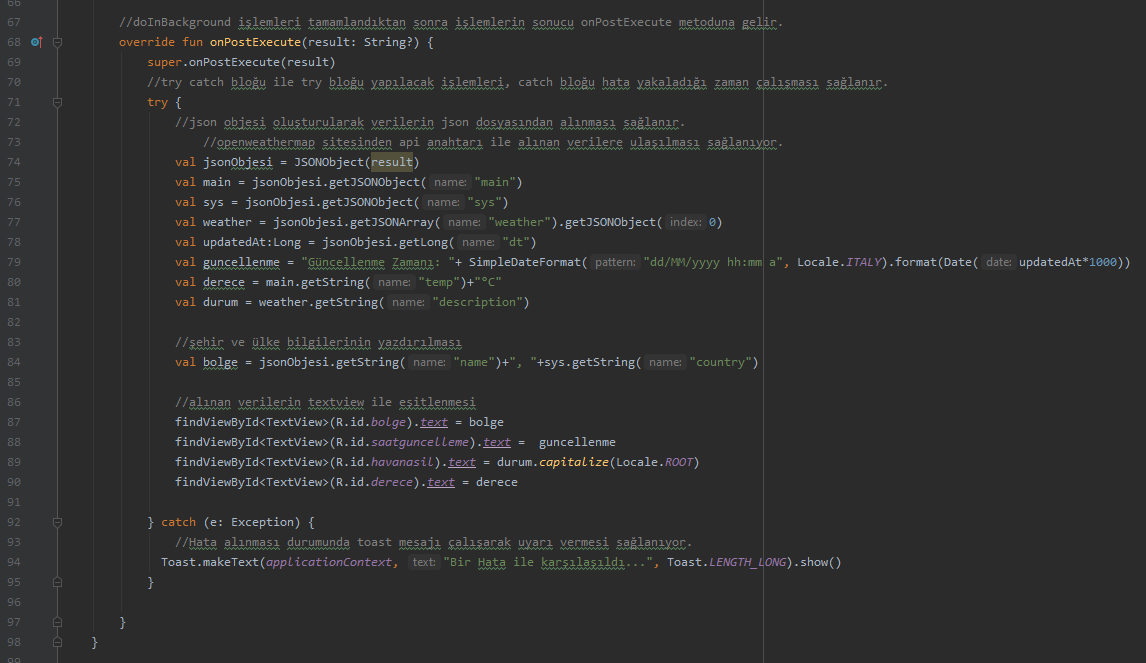
Uygulama içerisinde bildirim verilmesi isteniliyor ise kullanılır.

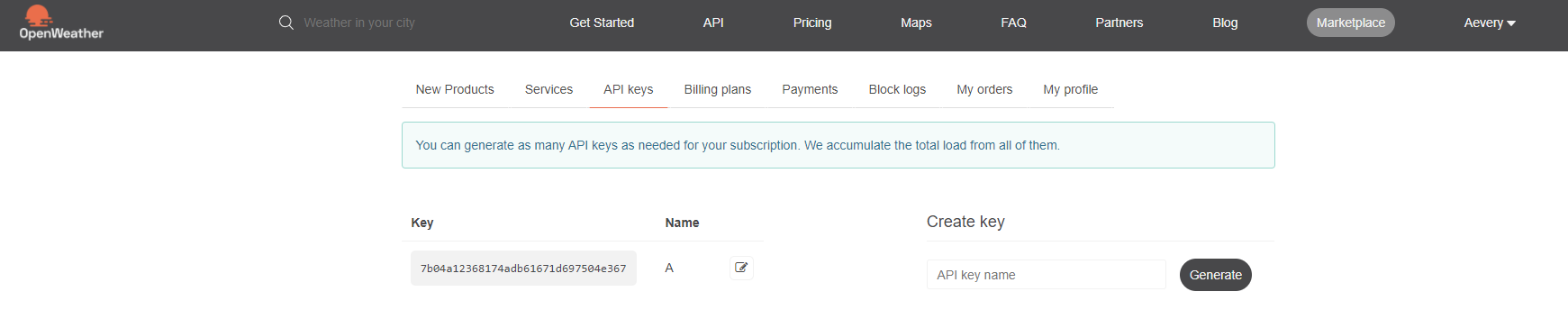
# Not Kaydetme

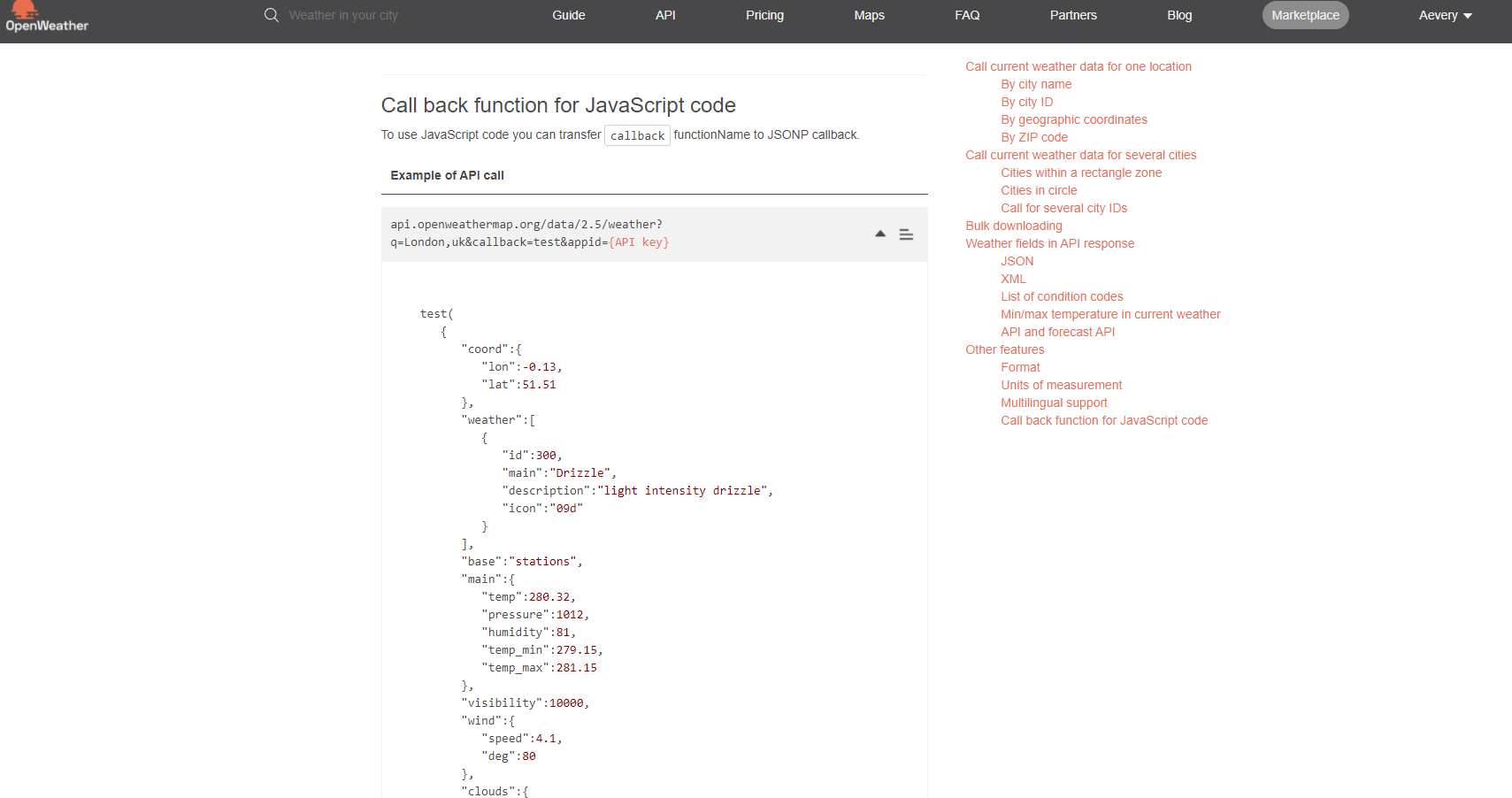
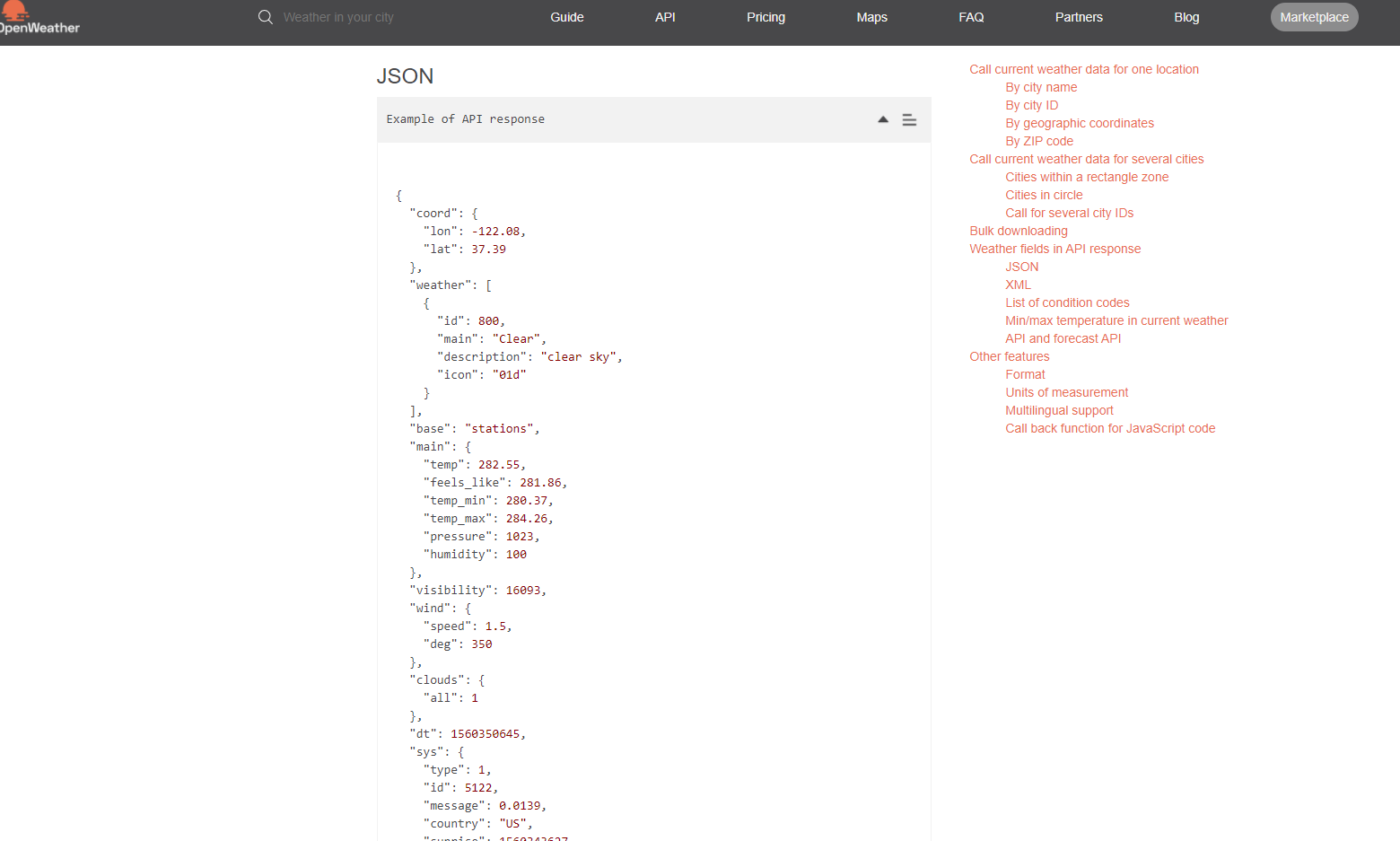
Not kaydedilmesi için dosya adı verilmesi gerekiyor. Not adı ile birlikte kaydedilen notlar görüntülenebiliyor ve kaydedilebiliyor.



# Api

Api ile Hava durumu bilgileri <https://openweathermap.org/> adresinden alınıyor.

Api key ile verilere oluşturulan api keyler ile ulaşılması sağlanıyor. Uygulamada kullanılan api key:

Alınan Json verilerin yapısı

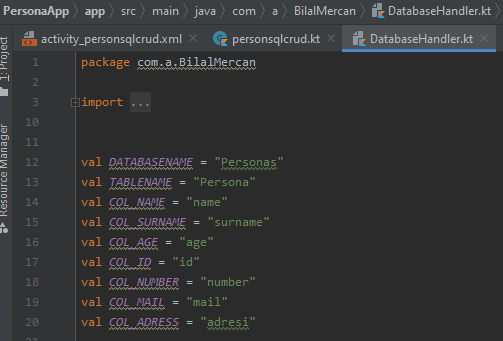
# Sqlite Veri Tabanı Crud (Oluşturma, Okuma, Güncelleme, Silme) İşlemleri

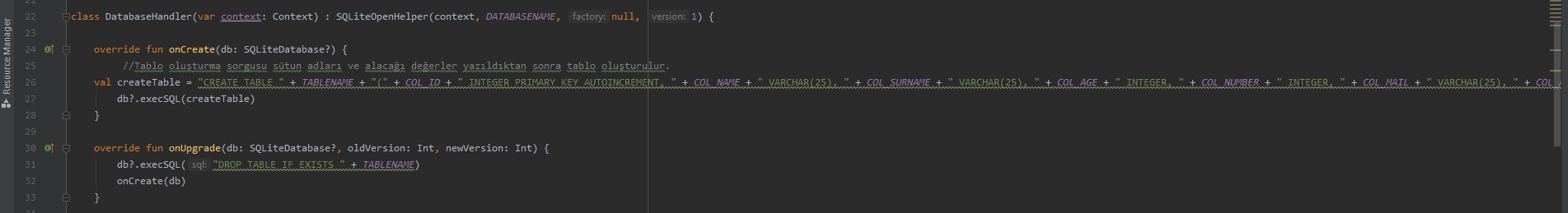
Sqlite ile basit veri tabanı işlemleri yapılabilmekte. Bu uygulamadaki senaryosu; personel bilgilerini veritabanına aktaran bir sistem tasarlamak.

## Tablo Oluşturma

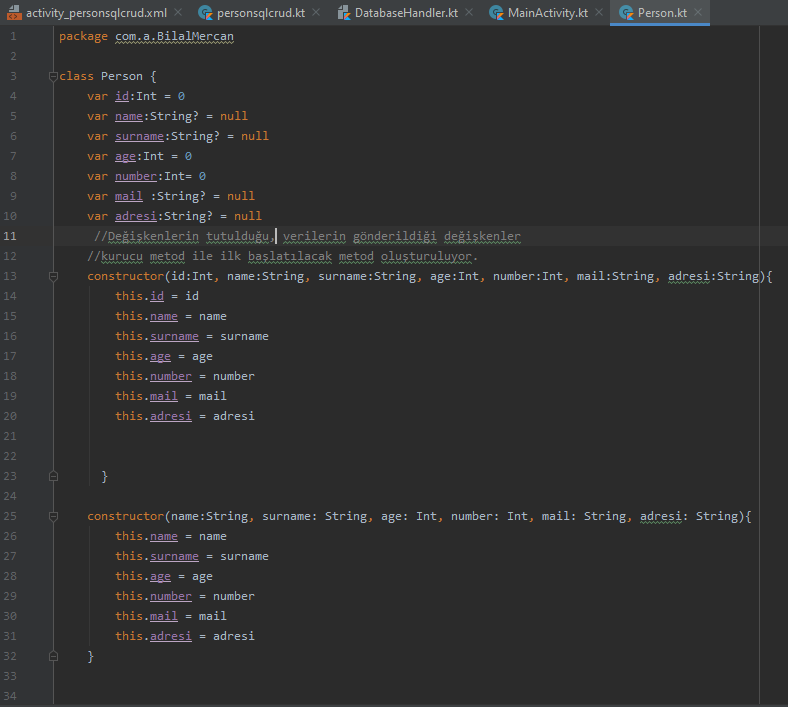
Veri tabanı oluşturmak için öncelikle Database Handler dosyası oluşturuluyor. Bu dosya içerisinde sütunlar, verilerin hangi değişkende tutulacağı, sql sorguları ve oluşturma, silme, güncelleme ve okuma fonksiyonları tanımlanıyor.

Öncelikle değiştirilemez değişkenler olarak veri tabanı adı, tablo adı, sütun adları ayarlanıyor.



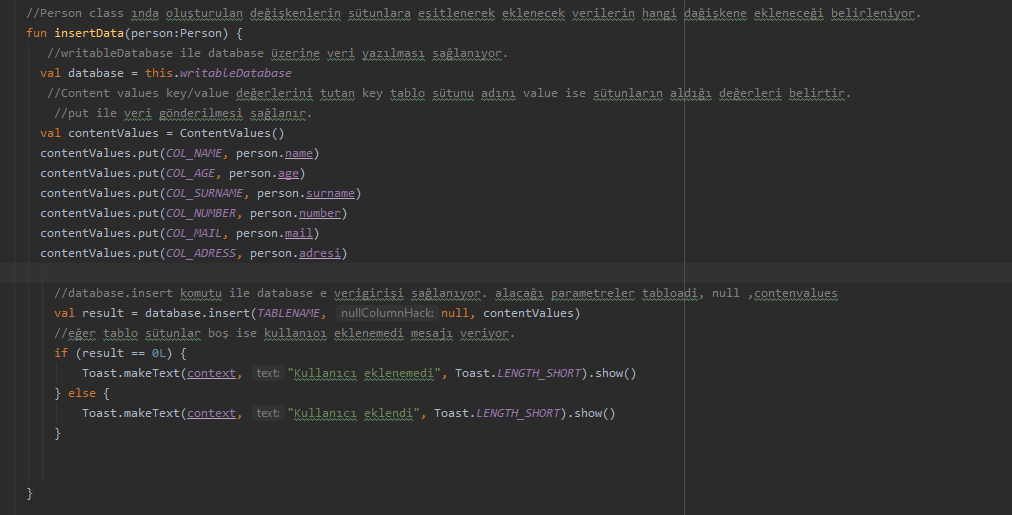
Oncreate metodu içinde ilk çalıştırılacak işlem yani tablo oluşturma işlemi yapılıyor. Sql sorgusu ile ve veri tipleri ayarlanarak tablo oluşturuluyor ve değişkene atanıyor. execSql ile oluşturulan sorgu çalıştırılıyor.

Verilerin oluşturulması, silinmesi, güncellenmesi ve listelenmesi için verilerin değişkenlerde tutulması gerekiyor. Person adında Sınıf oluşturulup içerisine değişkenler tanımlanıyor. Var ile değiştirilebilir değişken (var) ile değişken adları ve değişken türleri belirleniyor. Constructor metodu içerisinde daha önceden oluşturula değişkenler metod içerisinde tanımlanıyor ve eşleştiriliyor.



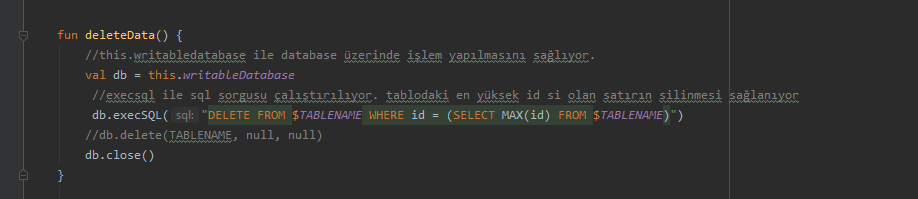
## Tabloya veri ekleme

İnsertData adında metod oluşturuluyor. Parametresi Person sınıfı olarak belirleniyor.



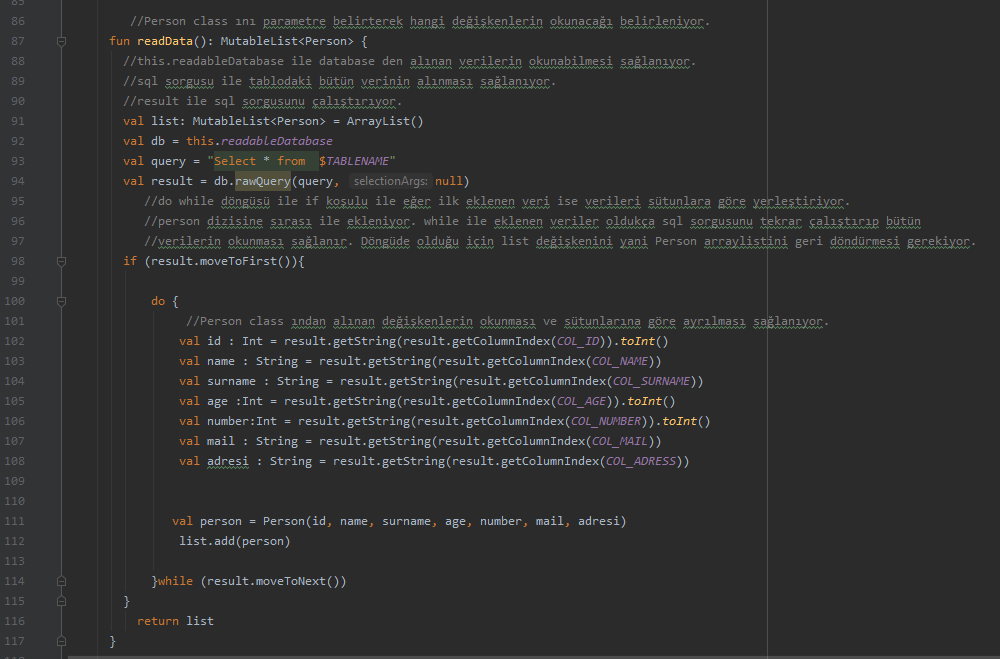
## Tabloda veri silme

deleteData adında metod oluşturuluyor. Sql sorgusu ile tablodaki eklenen en son verinin (en son id) ile satırın silinmesi işlemini gerçekleştirir. Db.close ile veri tabanıyla iş bittiğinde veri tabanı kapatılır böylece arka planda sürekli çalışılmaması sağlanır.



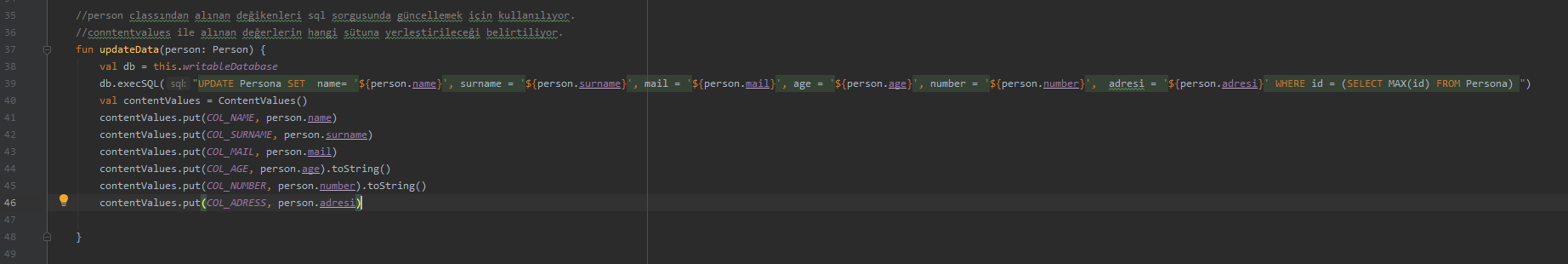
## Tabloda veri listeleme

readData adında metod oluşturuluyor. Mutablelist değiştirilebilir değişkenleri okuyabilen, yazabilen, silebilen ve güncelleyebilen collection. Person parametresi ile işlem yapılacak verilerin yolu belirleniyor. İf koşulu ile eğer tabloya daha önce veri eklenmemişse do while döngüsüne girilerek do bloğunda değişkenler ve türleri belirtiliyor ve tablodaki sütun indexine yerini belirterek sütun adları ile değişken adlarının eşleştirilmesi sağlanıyor. Person adında değişken tanımlanıyor. Tanımlanan bu değişken ile sütun adları ile eşleştiriliyor. Daha önceden tanımlanan list değişkeninin add komutu ile person değişkenine eklenmesi sağlanıyor. While ile döndürülen değerin yani ilk veri eklendiğinde (.moveToNext) ile eklenecek verilerin aynı işlemleri yapılması sağlanıyor.

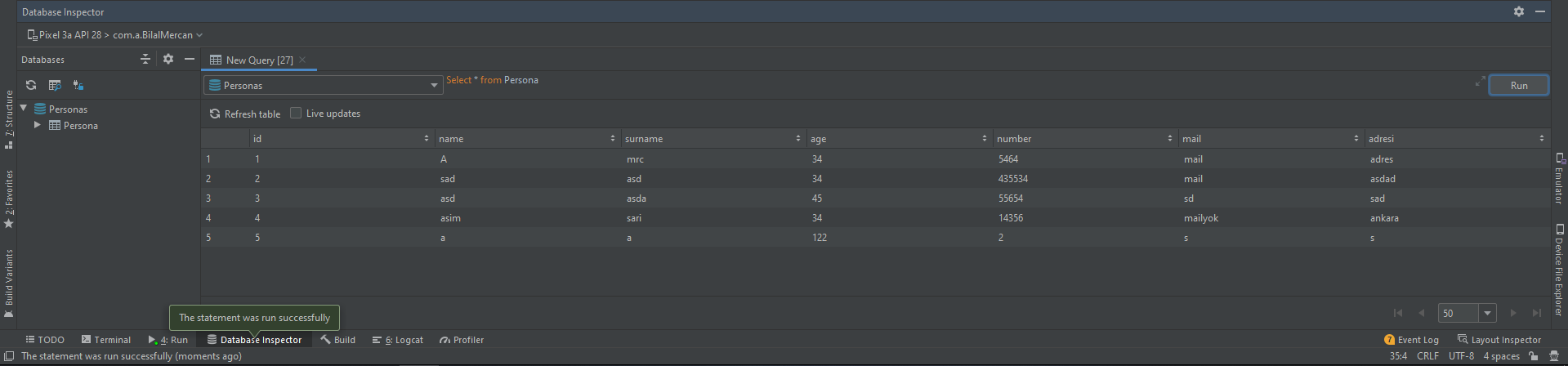


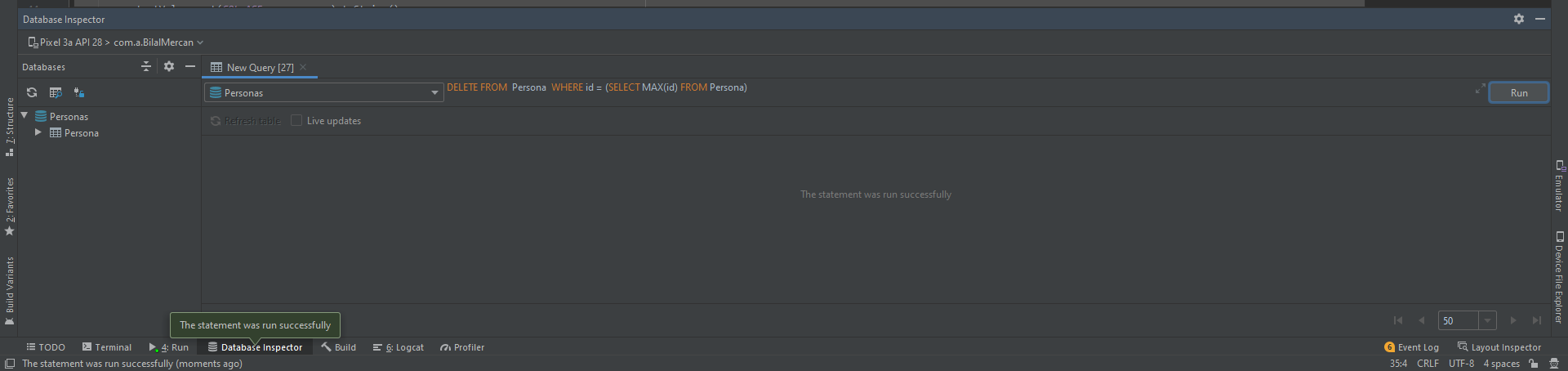
## Tabloda veri güncelleme

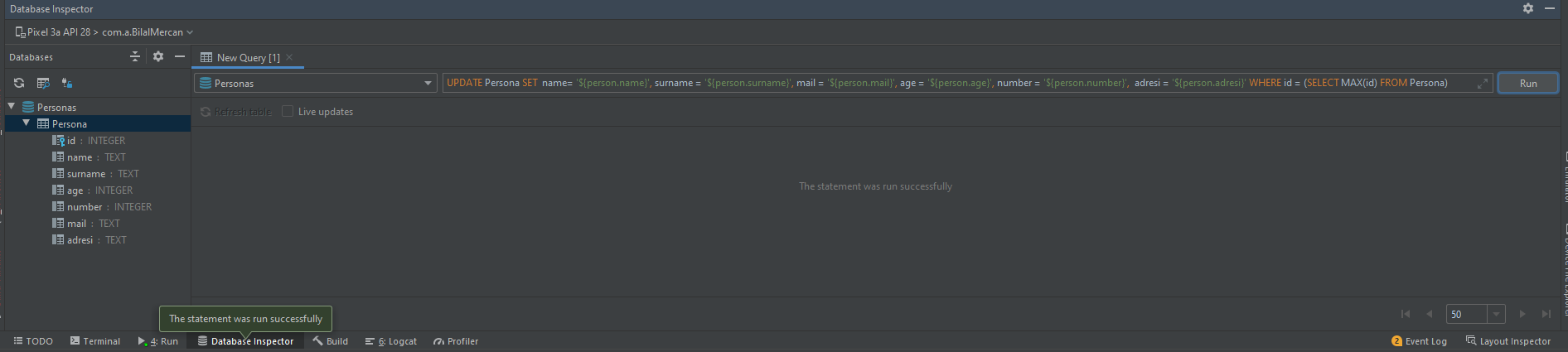
Updatedata adında fonksiyon oluşturuluyor. Person class parametre olarak belirleniyor. Sql sorgusu ile Tablo güncelleniyor. Tablodaki güncelleme koşulu en yüksek id ile güncelleme sağlanıyor. Content Values ile sütun ve person class ında ki veriler eşleştiriliyor. Put ile verilerin gönderilmesi sağlanıyor.



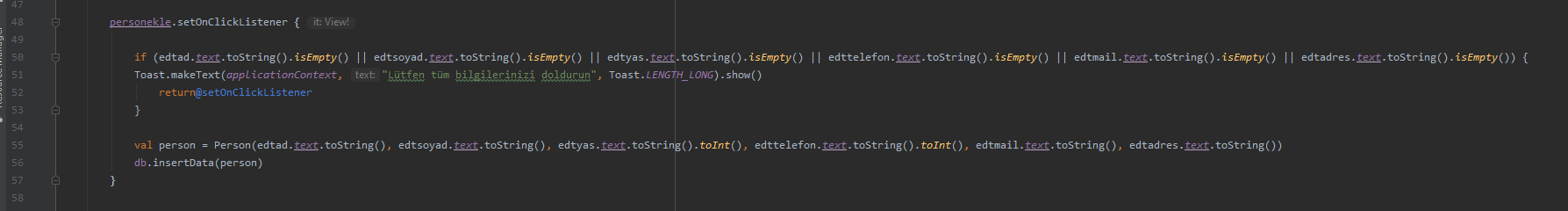
## Veri tabanı sorguları testi

Tablodaki bütün verileri listeleme sorgusu

Delete sorgusu testi 

 Update sorgusu testi

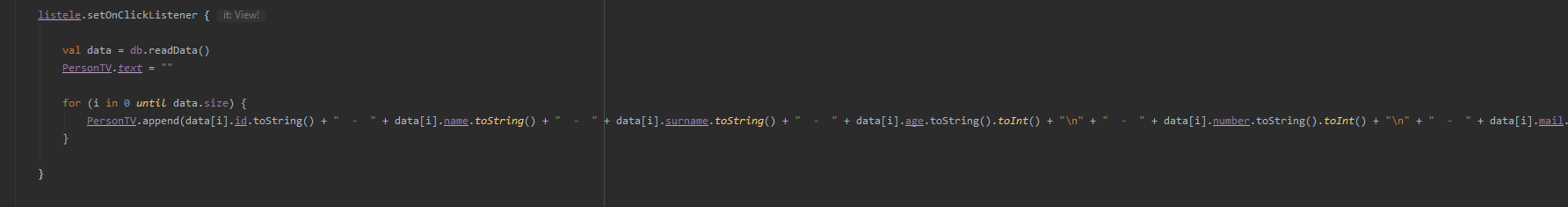
## Ekle Butonu

Person adında değişken oluşturuluyor. İf koşuluna edittext ten girilen verilerin boş olması halinde toast mesajı veriliyor. Oluşturulan bu değişkende Person class’ındaki veriler edittext e girilen verileri sütunlara yazılması sağlanıyor. (. ToString) ile verilerin yazılmasını, (. ToInt) ile sayısal verilerin tanımlanmasını sağlıyor. Daha önceden tanımlanmış olan Sqlite Db değişkeninde bulunuyor. İnsertdata ile Sql komutu çalışıyor. Person değişkenine veriler ekleniyor.

## Listele Butonu

ReadData ile DatabaseHandler sayfasında verilerin okunmasını sağlayan metod çağırılıyor.

Textview’ a .text yazılması sağlanır.

 For döngüsü ile veri boyutu 0’dan büyükse PersonTV adındaki textview’ a append komutu ile eklenmesi sağlanıyor. Data değişkeni person class ında ki değişkenler ile eşleştiriliyor ve yazıya dönüştürülerek verilerin görüntülenmesi sağlanıyor.

## Güncelle ve Sil Butonu

Güncelle butonunda person değişkenine edittext ten alınan verilerin yazıya dönüştürülmesi sağlanıyor. Güncelle ve sil butonunda DatabaseHandler sayfasında tanımlanmış olan metodlar çalıştırılıyor.

