RAPOR BAŞLANGIÇ

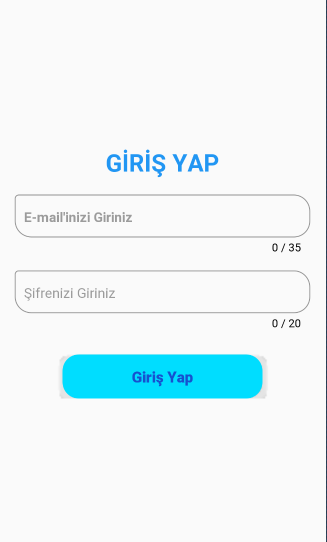
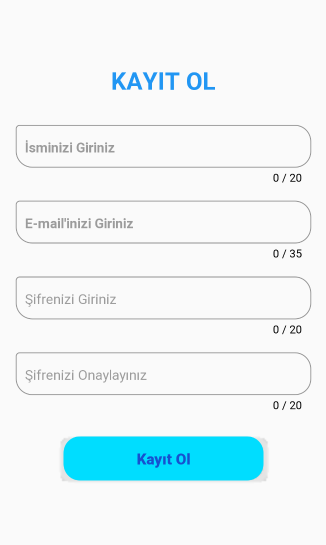
Chat Programı

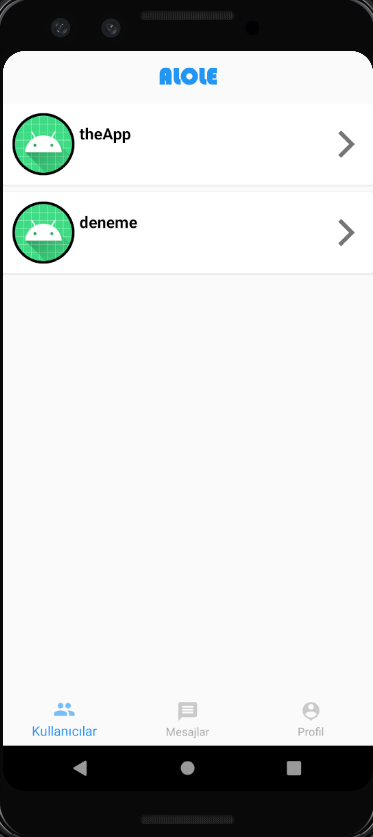
Proje Grup Üyeleri

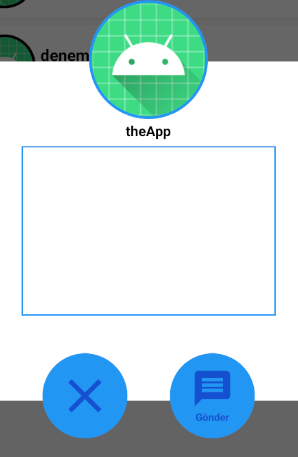
Berrin Korkutata

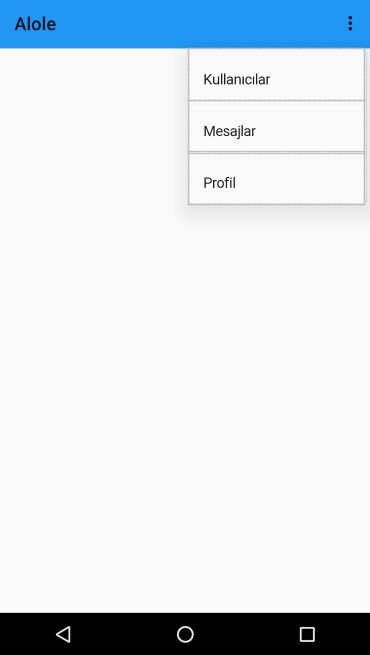
Süleyman Özdemir

1. EKRAN TASARIMLARI

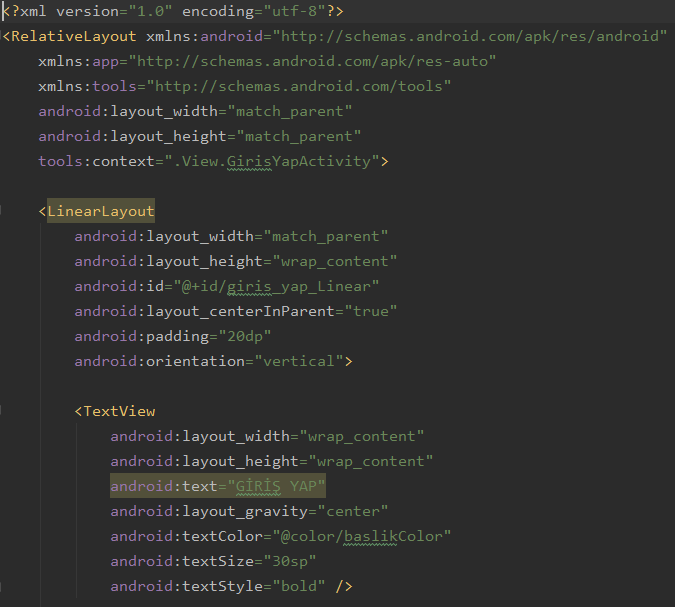




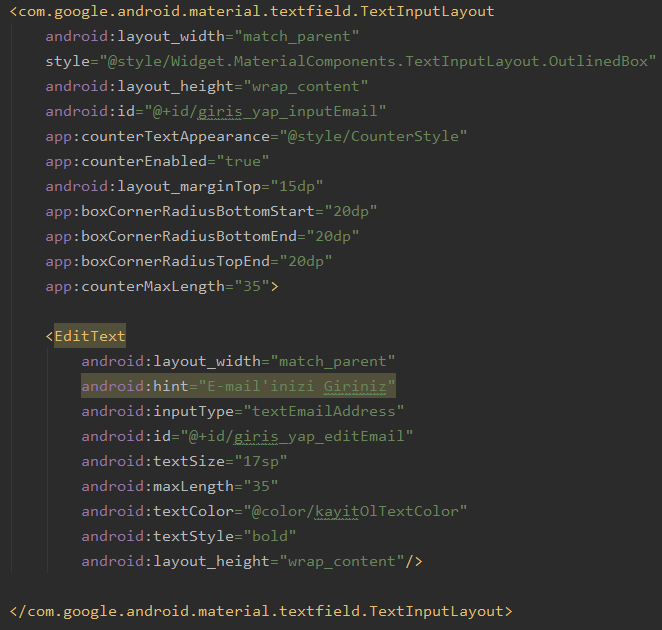
1. Giriş Ekranı Ara Yüz Düzen Kodlamaları

 Giriş ekranı ana kısmı

Burada Giriş Ekranının,

boyutu ve konumu ayarlandı. Burada Ekranın Başlığı, başlığın ismi, başlık

ismini, şekli ve boyutu ayarlandı.

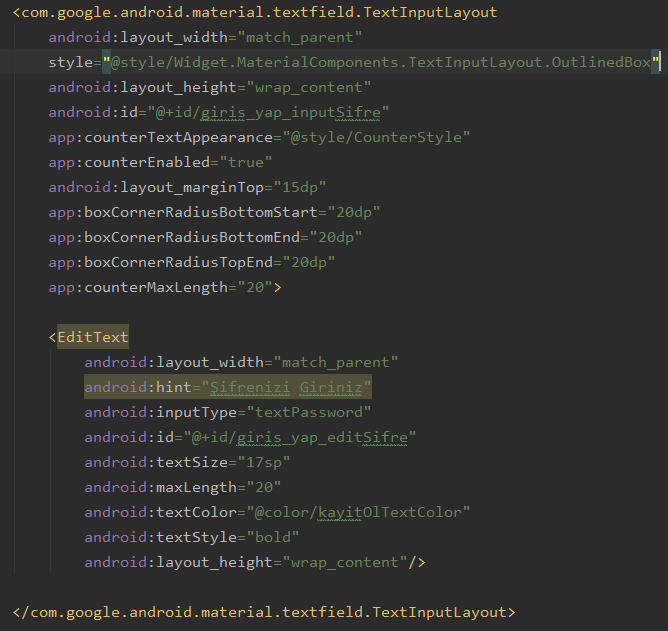
 Giriş Ekranındaki e-maili

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.

Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.

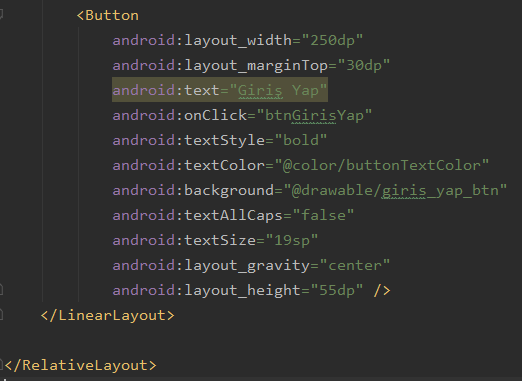
 Giriş Ekranındaki şifreyi

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.

Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.

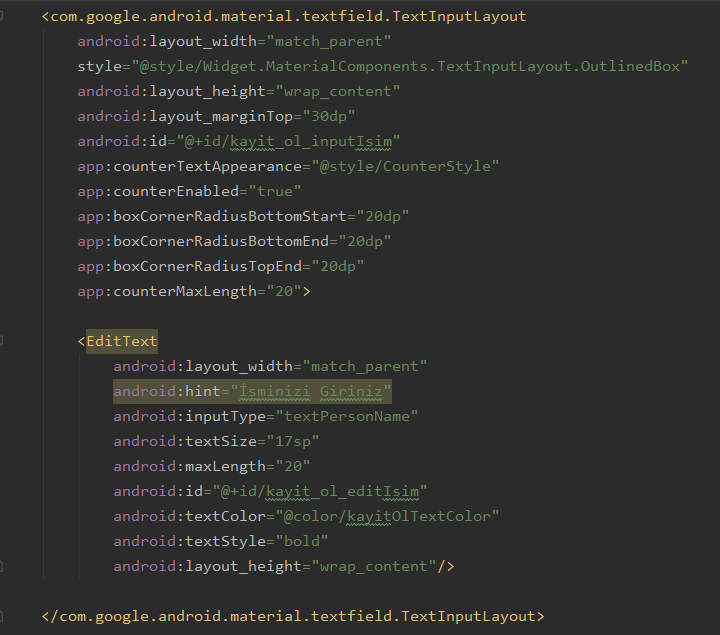


Giriş Ekranında giriş yapılması için gerekli olan butonun rengi, üzerindeki yazısı, şekli, yazı boyutu, konumu, yazının stili ayarlandı.

1. Kayıt Ol Ekranı Ara Yüz Düzen Kodlamaları

 Kayıt ol Ekranı ana kısımlar

Burada Ekranın Başlığı, başlığın ismi, başlık ismini, şekli ve boyutu ayarlandı.



Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.

Kayıt ol Ekranındaki adımızı

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.



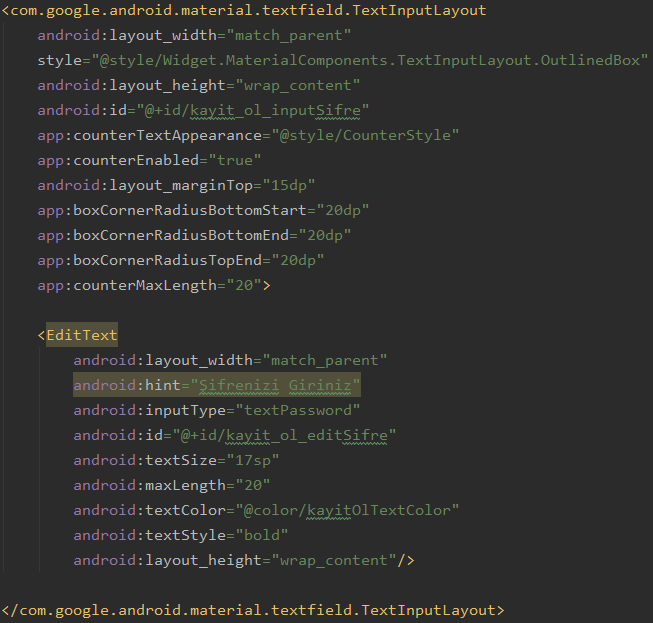
Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.

Kayıt ol Ekranındaki e-maili

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.

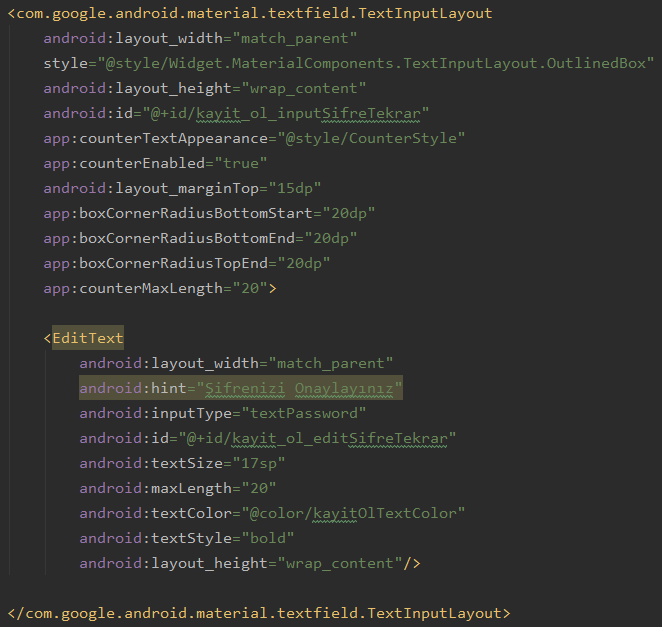
 Kayıt ol Ekranındaki şifreyi

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.

Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.



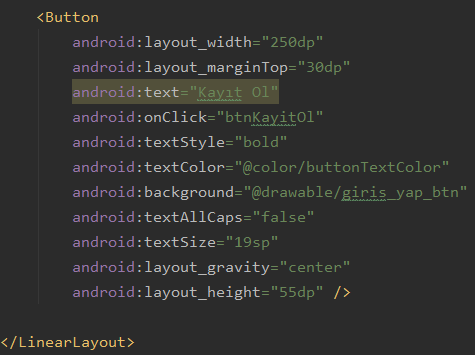
Textfield‘ın içine yazacağımız yazının boyutu, şekli, rengi

Ve boyutu ayarlandı.

Kayıt ol Ekranındaki şifreyi tekrar

Yazacağımız textfield’ın boyutu, şekli,

Maximum yazı uzunluğu ayarlandı.



Giriş Ekranında giriş yapılması için gerekli olan butonun rengi, üzerindeki yazısı, şekli, yazı boyutu, konumu, yazının stili ayarlandı.

1. Ana Ekran Ara Yüz Düzen Kodlamaları

 Ana ekran ana kısmı

Ana ekrana önceden hazırlanmış toolbar’ın eklenmesi

ve gerekli düzenleme ayarları yapıldı.

Ana ekrana eklenilen navigasyon

Tuşlarının düzeni ayarlandı.

 Ana ekrana eklenilecek kullanıcılar

Ekranı ikon düzenlemesi.

Ana ekrana eklenilecek mesajlar ekranı

ikon düzenlemesi Ana ekrana eklenilecek profil ekranı ikon

düzenlemesi.

1. Gelen Mesaj İstekleri Ara yüz Düzen Kodlamaları

 Mesaj İstekleri ara yüz düzeni

Ana kısımlar.

Gelecek olan mesaj sayfasının kapatılmasında Ara yüzün Başlık kısımlarının

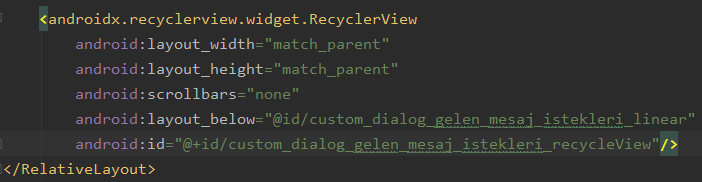
Gerekli olan ikonun düzenlenmesi. Düzeni.

Başlık yazısının boyutu ve stilinin

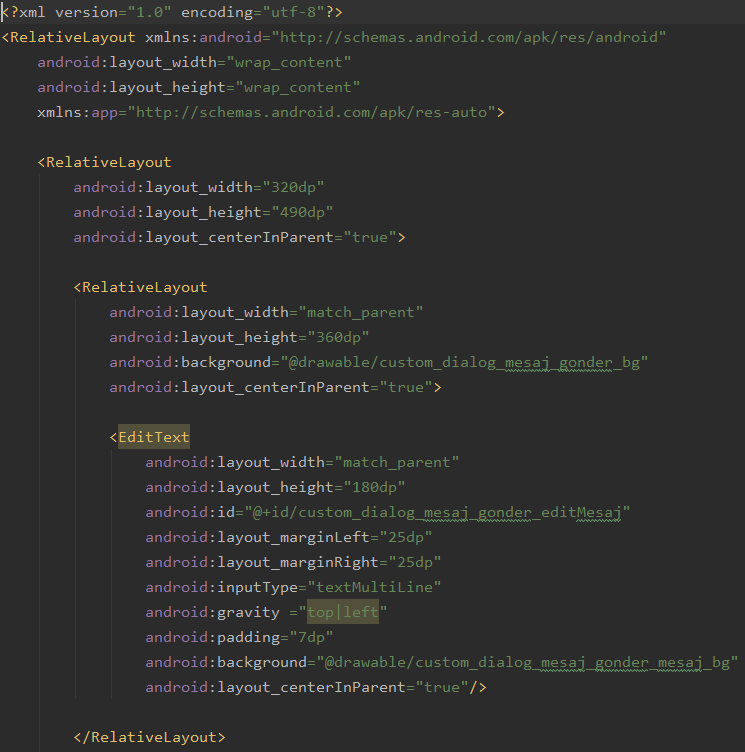
Düzeni.

 Toolbar ayarları düzenlenmesi

Toolbar’ın içindeki yazının şekil ayarlamaları.

 Mesajlaşma kısmının ayarlanması.

1. Mesaj Gönderme Ara yüz Düzen Kodlamaları

 Mesaj gönderme ana ayar.

Mesaj gönderme sayfasının görünüm düzenlenmesi.

Mesajlaşma alanının düzeni,

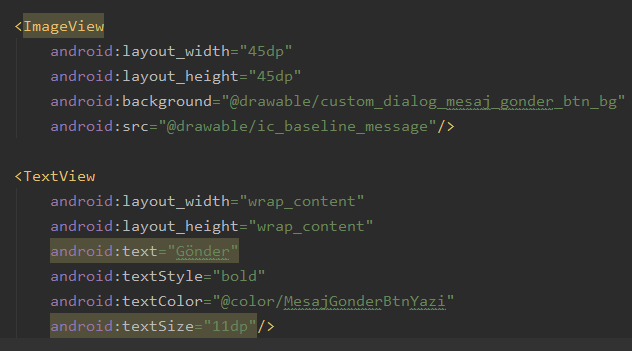
Konumu ve stilinin ayarlanması.

 Linear Layout’un düzeni

Mesajlaşma kutusunu kapatacak ikonun düzenlenmesi.

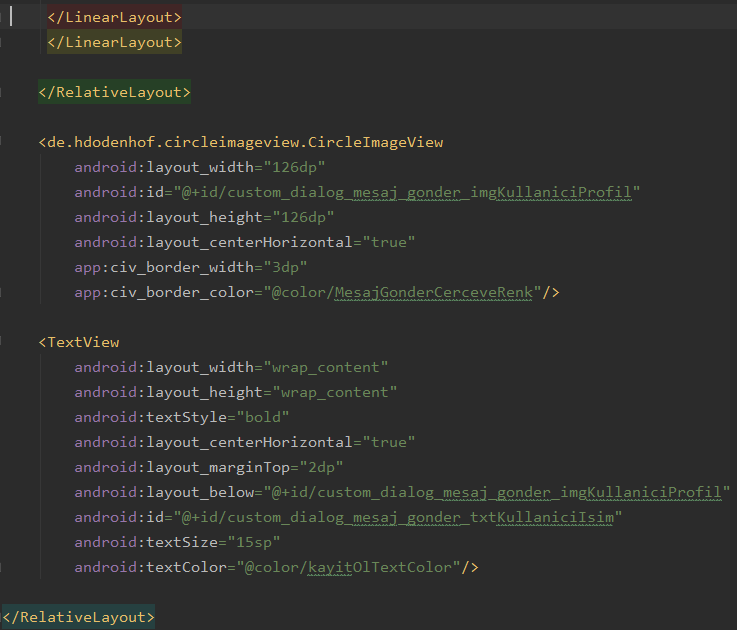
İkinci Linear Layout’un düzeninin

ayarlanması.

 Mesaj gönderme ikonunun düzen

ayarlanması.

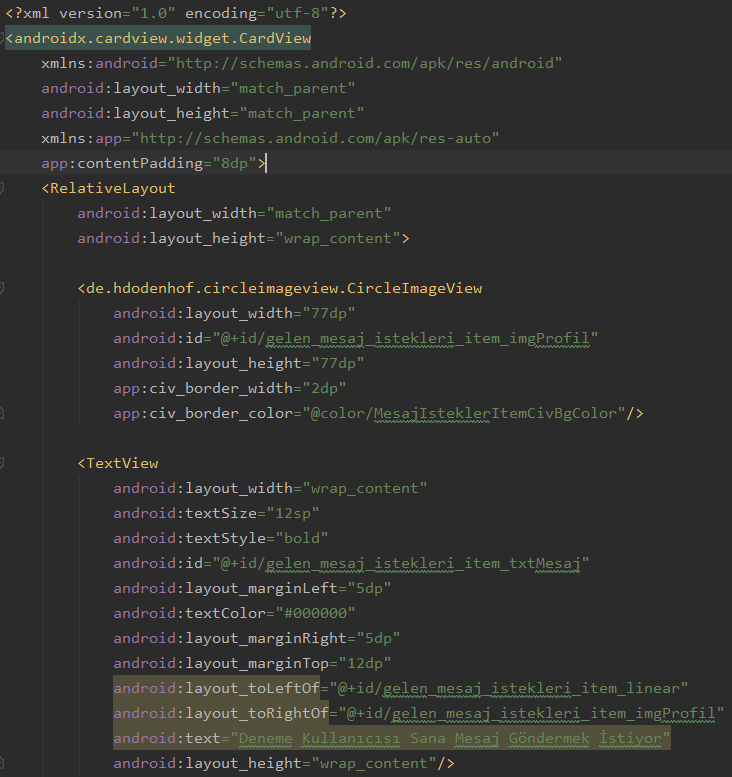
Mesaj gönderme tuşunun yazısını ayarlaması.

 Mesajlaşma kutusunun renk ve boyutunun

ayarlanması.

Mesajlaşma kutusunun yazı şekli, boyutunu ve stilinin ayarlanması.

1. Mesaj İstekleri Kabul veya Reddetme Ara Yüz Kodlamaları

 İstek Gönderme Ekranının ana kısmı

İstek gönderen kişinin profil resminin görüneceği dairenin düzen ayarlamaları.

Mesaj isteğinin gönderen kişi hakkında

verecek bilginin yazı stili, boyutu ve

konumunun ayarlanması.

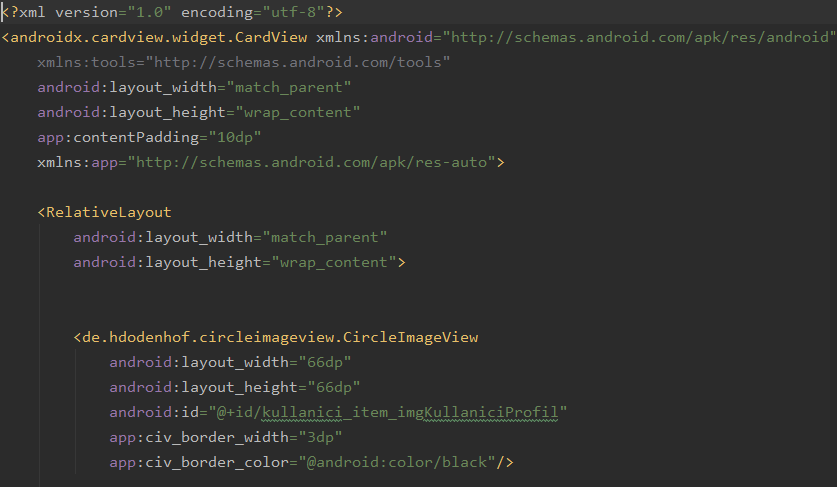
 Linear Layout’un ana düzenlemeleri

Mesaj isteğini reddedecek ikonun düzen ayarlaması.

Mesaj isteğini onaylayacak ikonun düzen

ayarlanması.

1. Kullanıcı İsimleri Ara Yüz Düzen Kodlamaları

 Tasarımın ana ara yüz

düzenlenmesi.

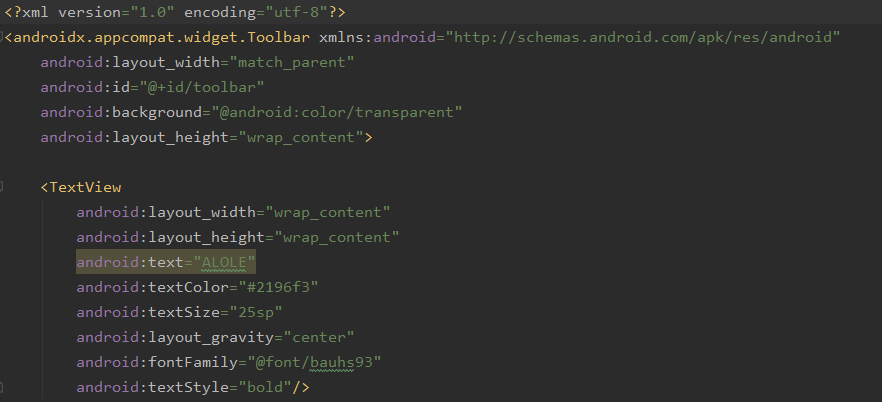
Görünecek olan kullanıcıların profil resimlerinin görünüm ayarları düzenlendi.

 Kullanıcı adının görüneceği

Textfield’ın düzen ayarları.

Üzerine tıkladığımızda kullanıcı bilgilerini gösterecek tuşun ayarları.

1. Yan Ara Yüz Düzenlemeleri

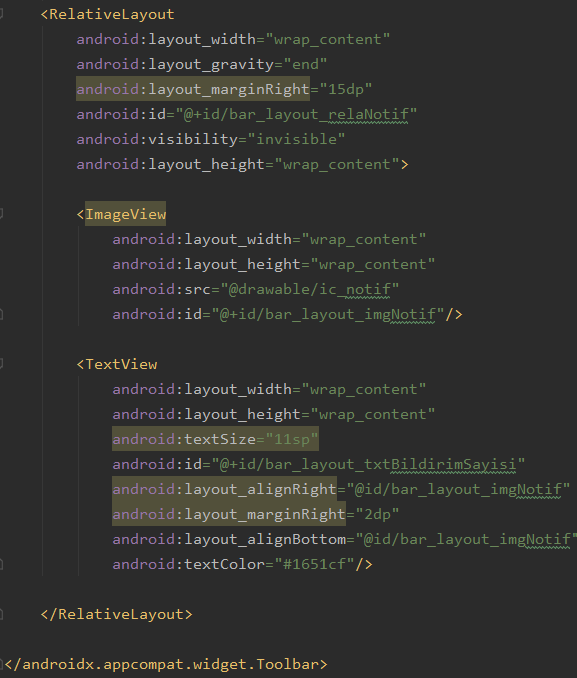


Tasarımın ana ara yüz

düzenlemesi.

Toolbar’ın başlığında bulunacak olan başlığın

ana düzenleme ayarları.

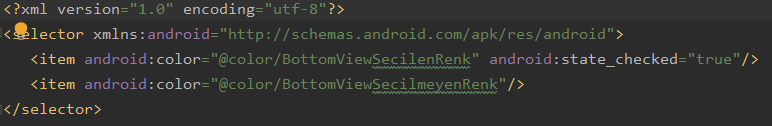
 Mesajlar ekranında görünecek

Olan bildirim işaretinin düzen ayarı.

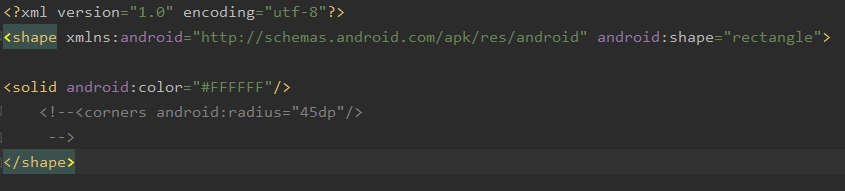
Bildirim işaretinin düzen ayarları.

Bildirim işaretinin boyut, şekil, renk ve

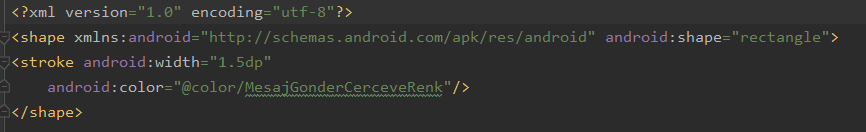
Konum düzeni ayarlandı.



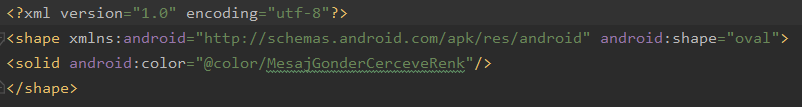
Sayfa geçişleri sırasında seçilecek olan sayfa tuşlarının renk ayarlamaları.



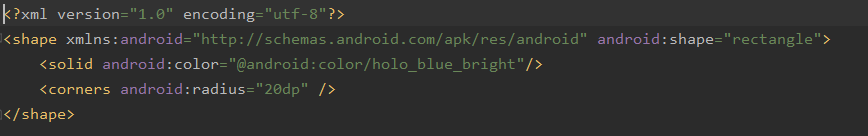
Mesajlaşma kutusunun şekil ve renk ayarları.



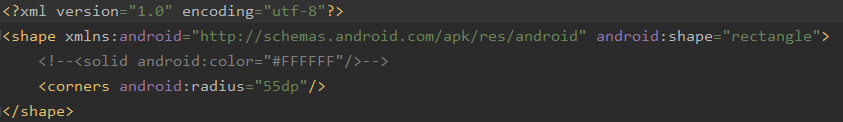
Mesaj kutusunda buluna çerçevenin şekil ve renk ayarları.



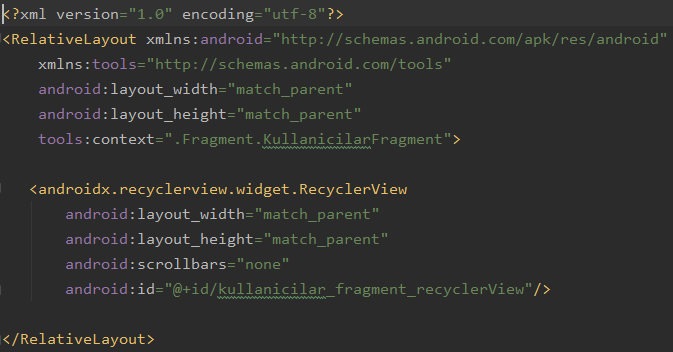
Mesaj kutusunun gönderme ve iptal etme tuşunun şekil ve renk ayarları.



Giriş Ekranında bulunan giriş yap tuşunun şekil ve renk ayarları.

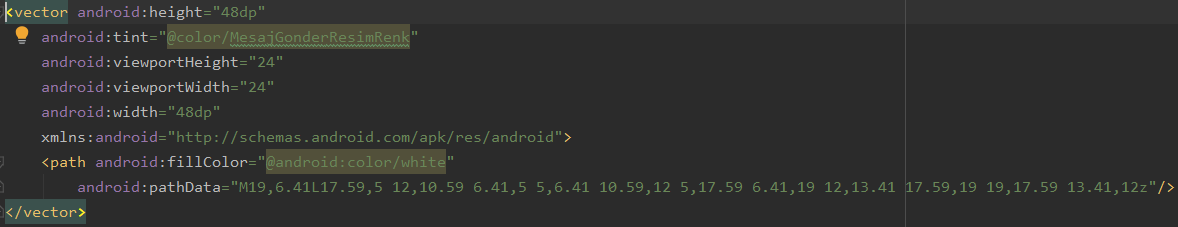


Kayıt ol Ekranında bulunan kayıt ol tuşunun renk ve şekil ayarları.

 Kullanıcıların

Adlarının görüneceği

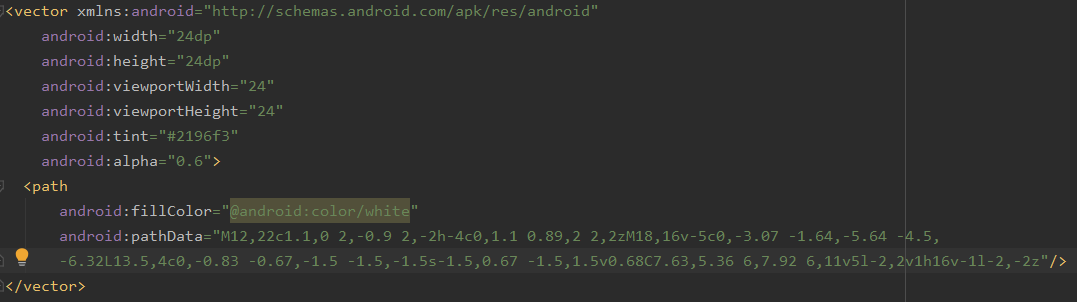
Sıranın düzen ayarları.



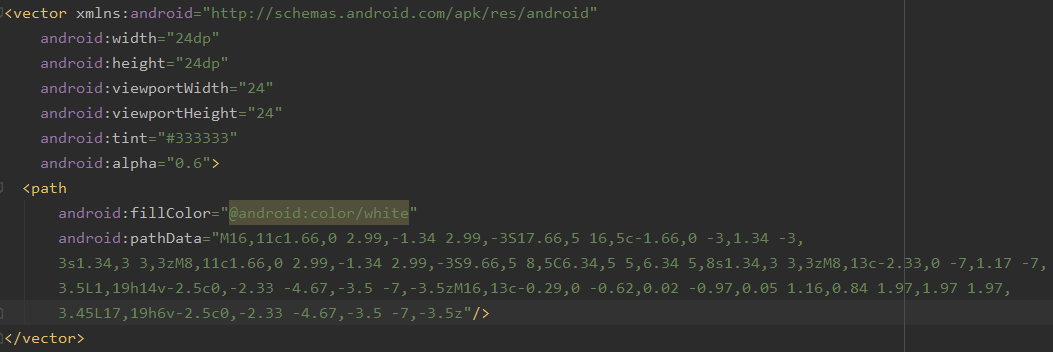
Belirli sayfaları kapatacak olan kapatma tuşunun genel tasarım ayarları.



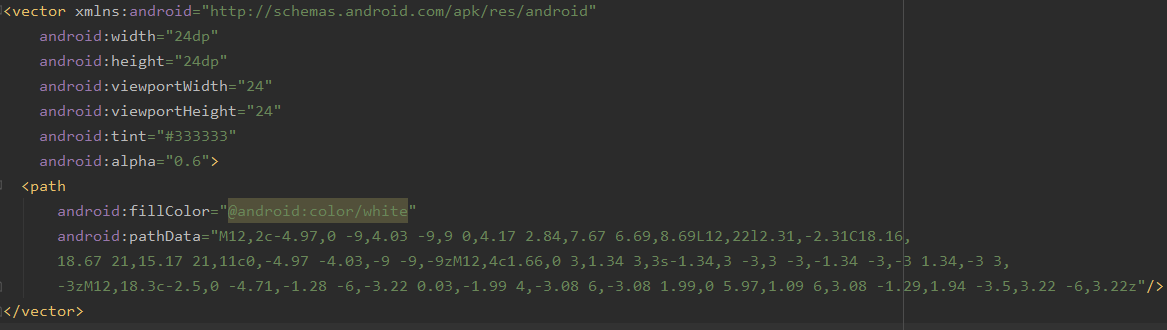
Mesajlar sayfasını açacak olan tuşun genel tasarım ayarları.



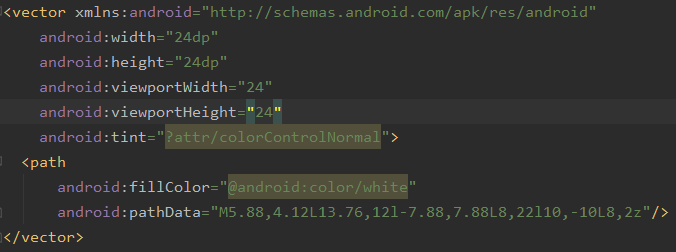
Mesaj bildirimlerini gösterecek olan tuşun genel tasarım ayarları.



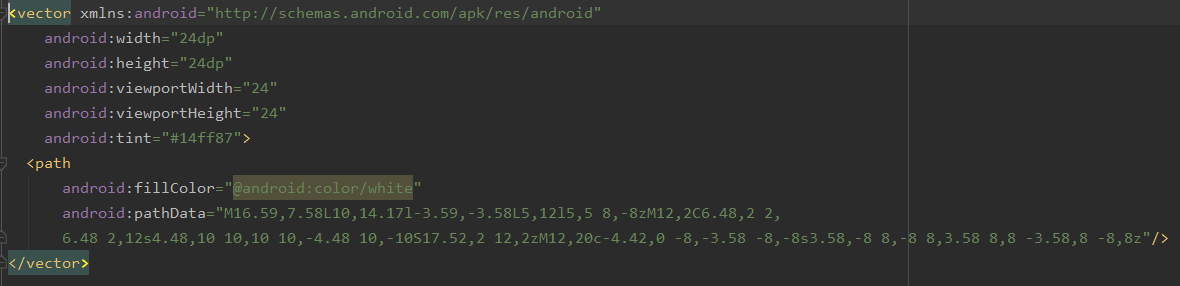
Chat programına kayıt olan kullanıcıların sayfasını açacak olan tuşun genel tasarım ayarları.



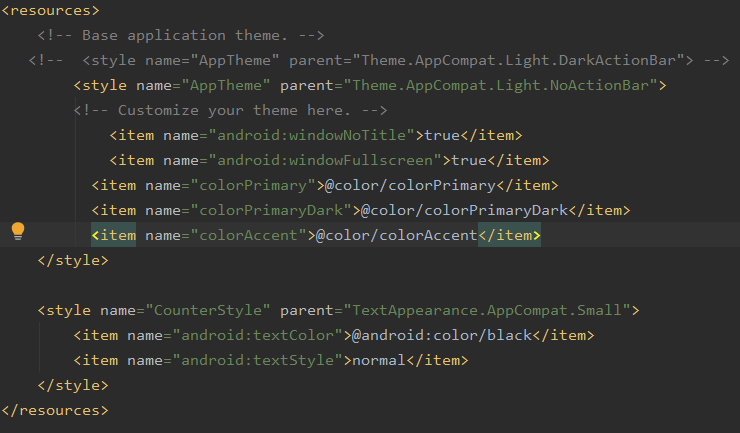
Kullanıcın kendi bilgilerinin bulunduğu sayfanın genel tuş tasarımının ayarları.



Kullanıcı bilgilerini gösterecek olan ekranı açan tuşun genel tasarımın ayarları.



Gönderilecek olan mesaj isteğinin kabul edilmesi için kullanılacak tuşun genel tasarım ayarları.

 Ekran tasarımlarının tamamında

kullanılacak olan düzen ayarlarının tamamı.

 Programın ana renk ayarlaması.

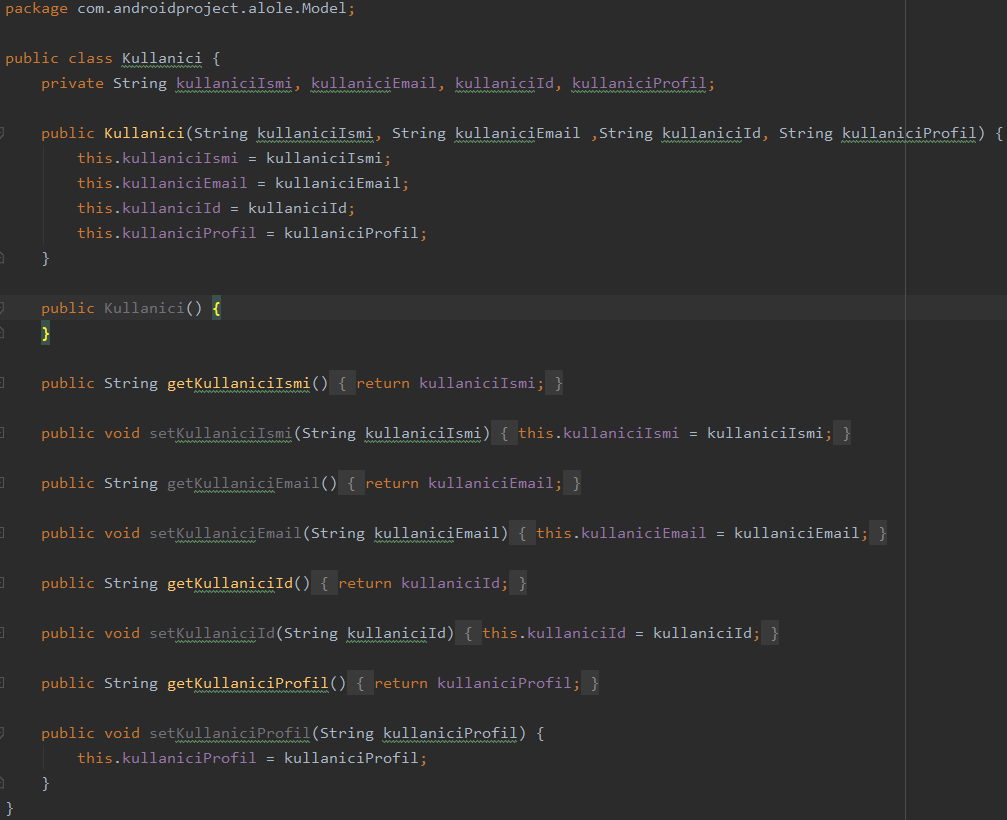
Kayıt yapma ekranındaki renklerin ayarlaması.

Ana Sayfadaki renklerin ayarlaması.

Mesajlaşma ekranındaki renklerin ayarlaması.

Mesaj isteklerinde kullanılacak olan tuşların renk ayarlaması.

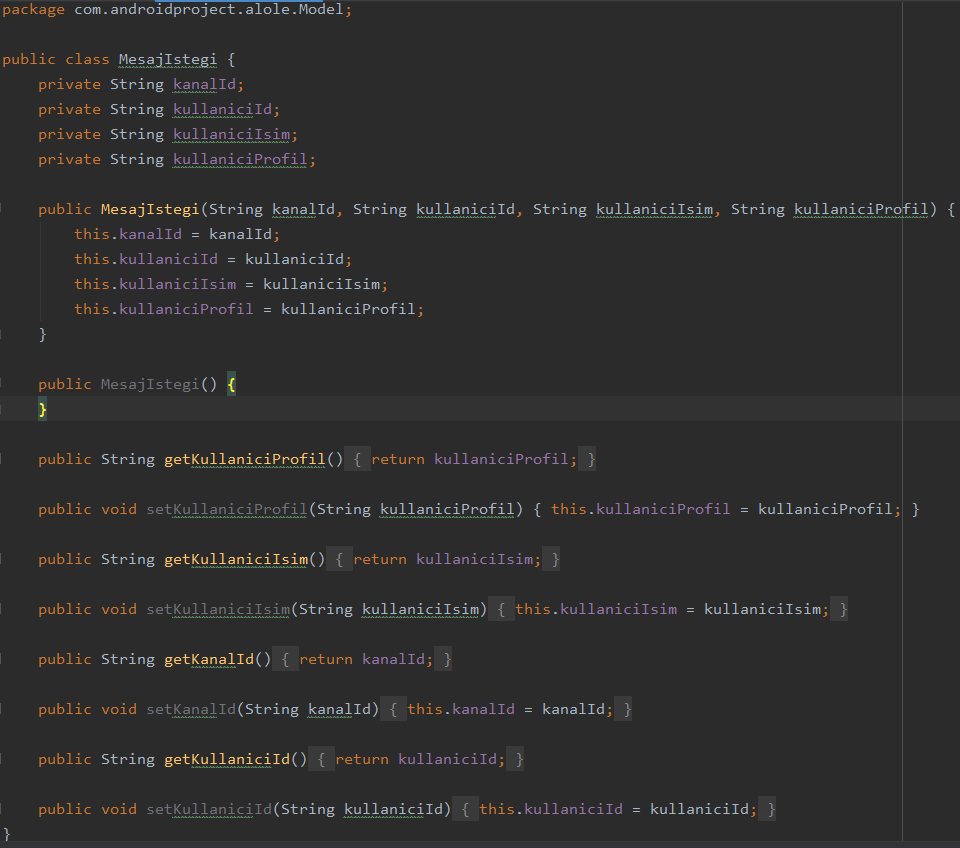
1. Kullanıcı İçin Gerekli Olan Bilgilerin Kodlamaları

 Gerekli parametreler ile

Parametrelerin Constructor tanımı.

Gerekli olan bilgilerin getter ve setter’larının tanımı.

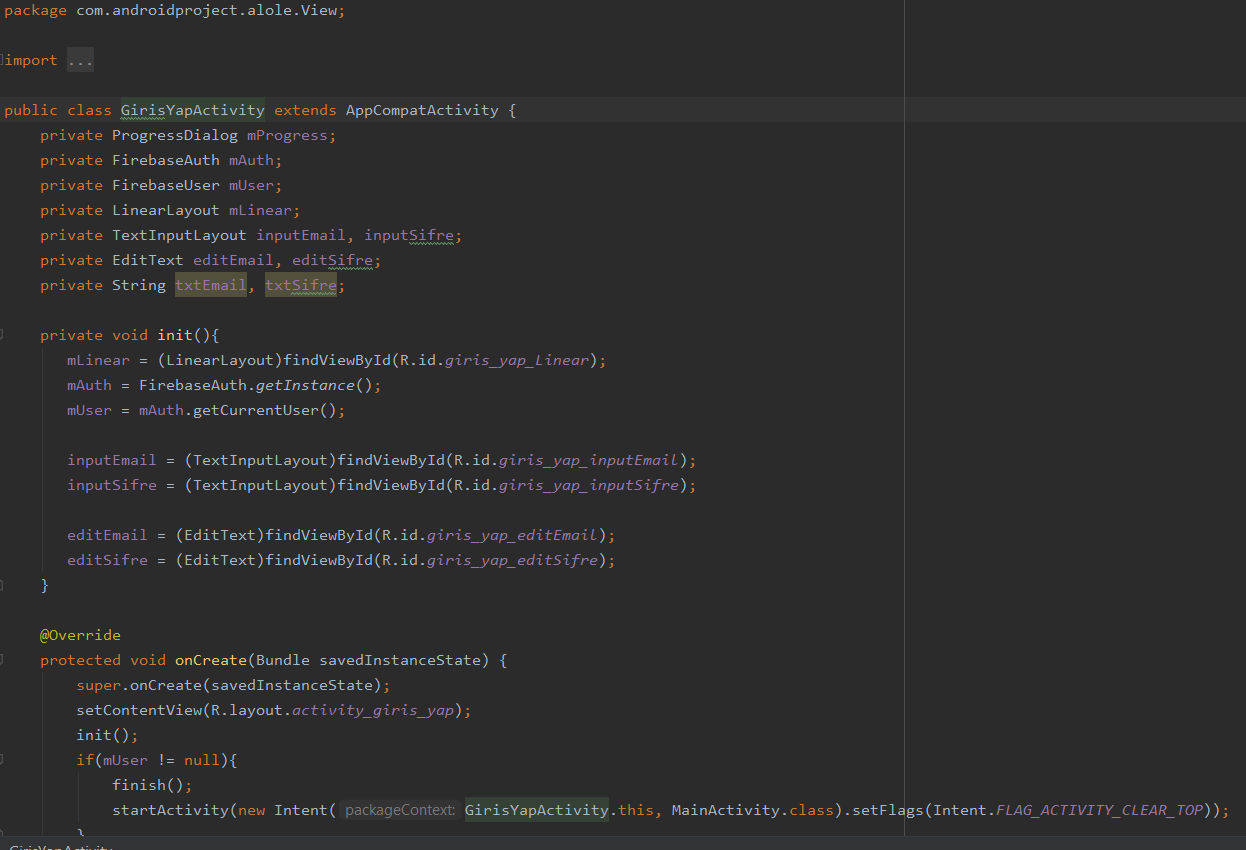
1. Mesaj İsteği İçin Gerekli Olan Bilgilerin Kodlamaları

 Gerekli parametreler ile

parametrelerin Constructor tanımı.

Gerekli olan bilgilerin getter ve setter’larının tanımı.

1. Giriş Yapma Ekranı Ana Kodlamalar

 Parametrelerin tanımı.

Parametreleri initialize ederek gerekli

yerlerde kullanacağız.

Burada daha önce giriş yapan kullanıcı var mı yok mu dite kontrol edilir. Var ise

Ana ekrana geçiş sağlanır.



e-mail ve şifre girildikten sonra e-mail ve şifre alanının boş olup olmadığını kontrol eder.

Girilen bilgiler doğru olup olmadığı kontrol edildikten sonra direk ana ekranı gidecek. Eğer yanlış ise neden girmediği hatasını gösterecek. Girilen e-mail ve şifrenin yerinin boş olması durumda

Karşımıza çıkacak olan uyarıları oluşturduk.

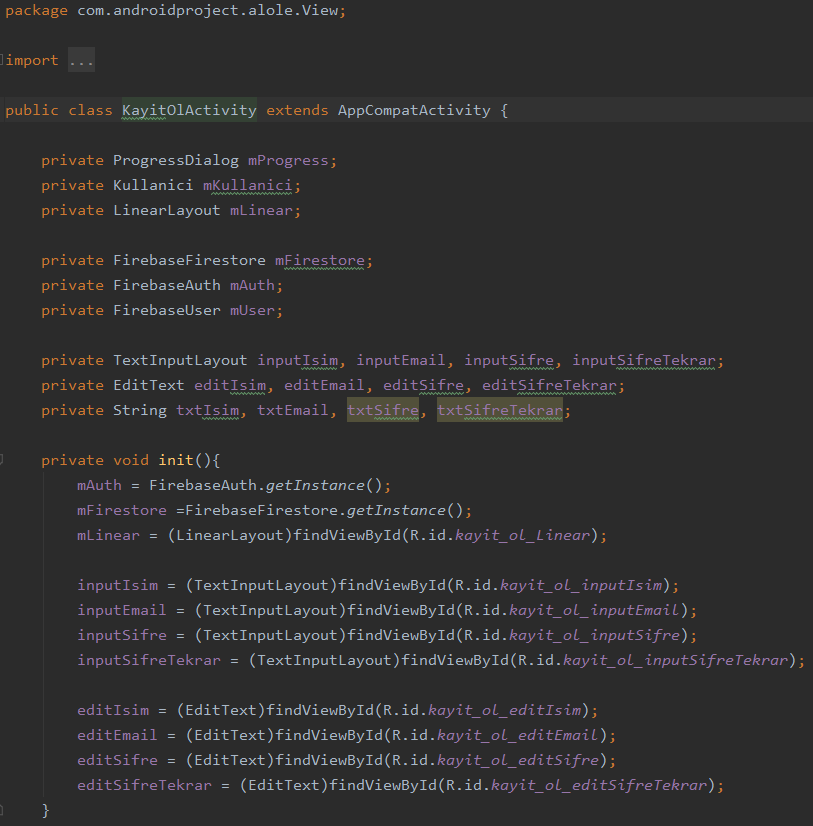
Giriş yap ekranında oluşturulan kayıt ol tuşuna basınca

Kayıt ol ekranı açılacak. Ayrıca Giriş yapıldıktan sonra

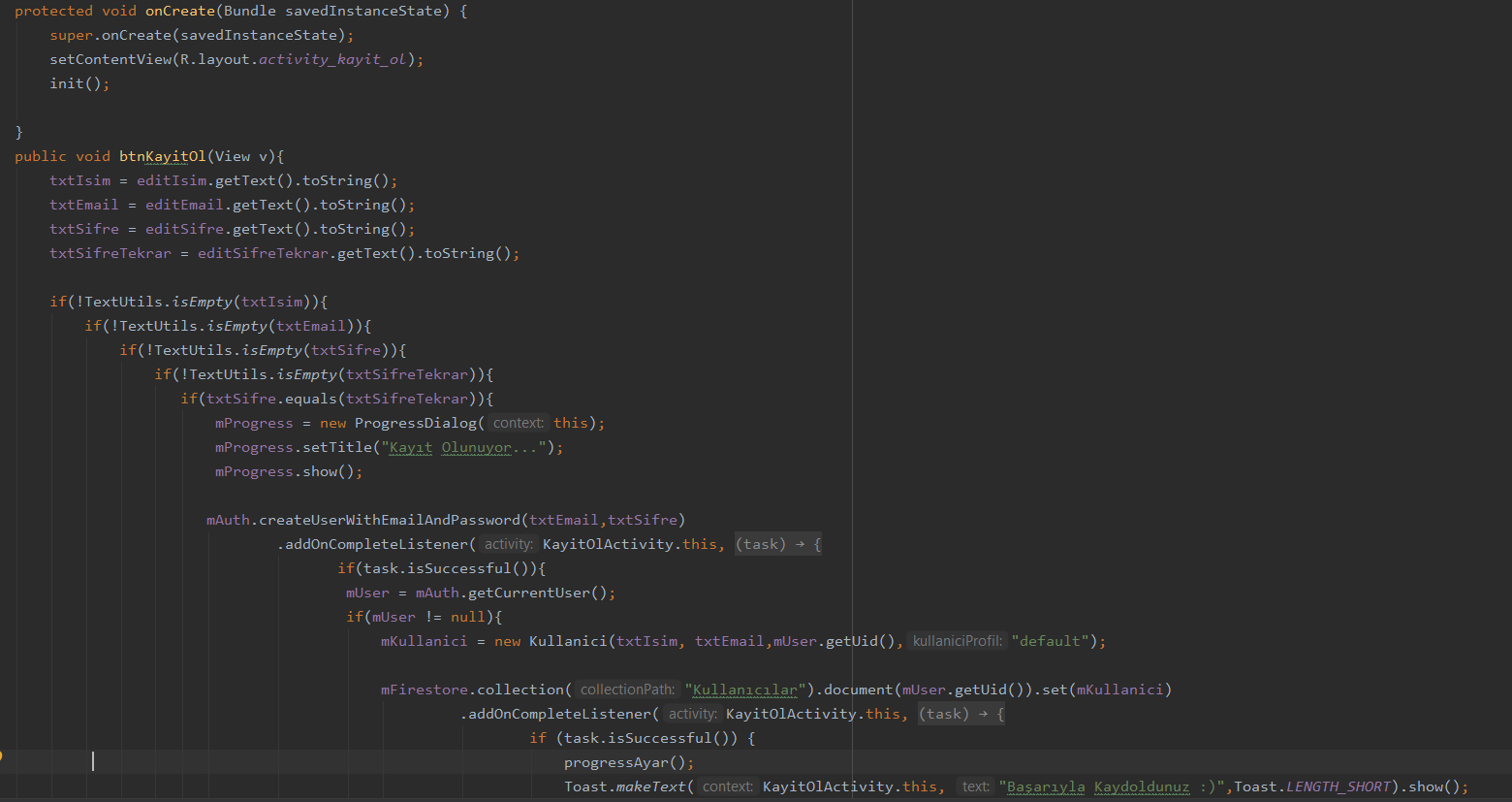
Ayarlanan Giriş yapılıyor yazısı görünecek. Sonra

Dismiss fonksiyonu ile kapatılacak.

1. Kayıt Yapma Ekranı Ana Kodlamalar

 Gerekli parametrelerin tanımı.

Parametreleri initialize ederek gerekli yerlerde kullanacağız.

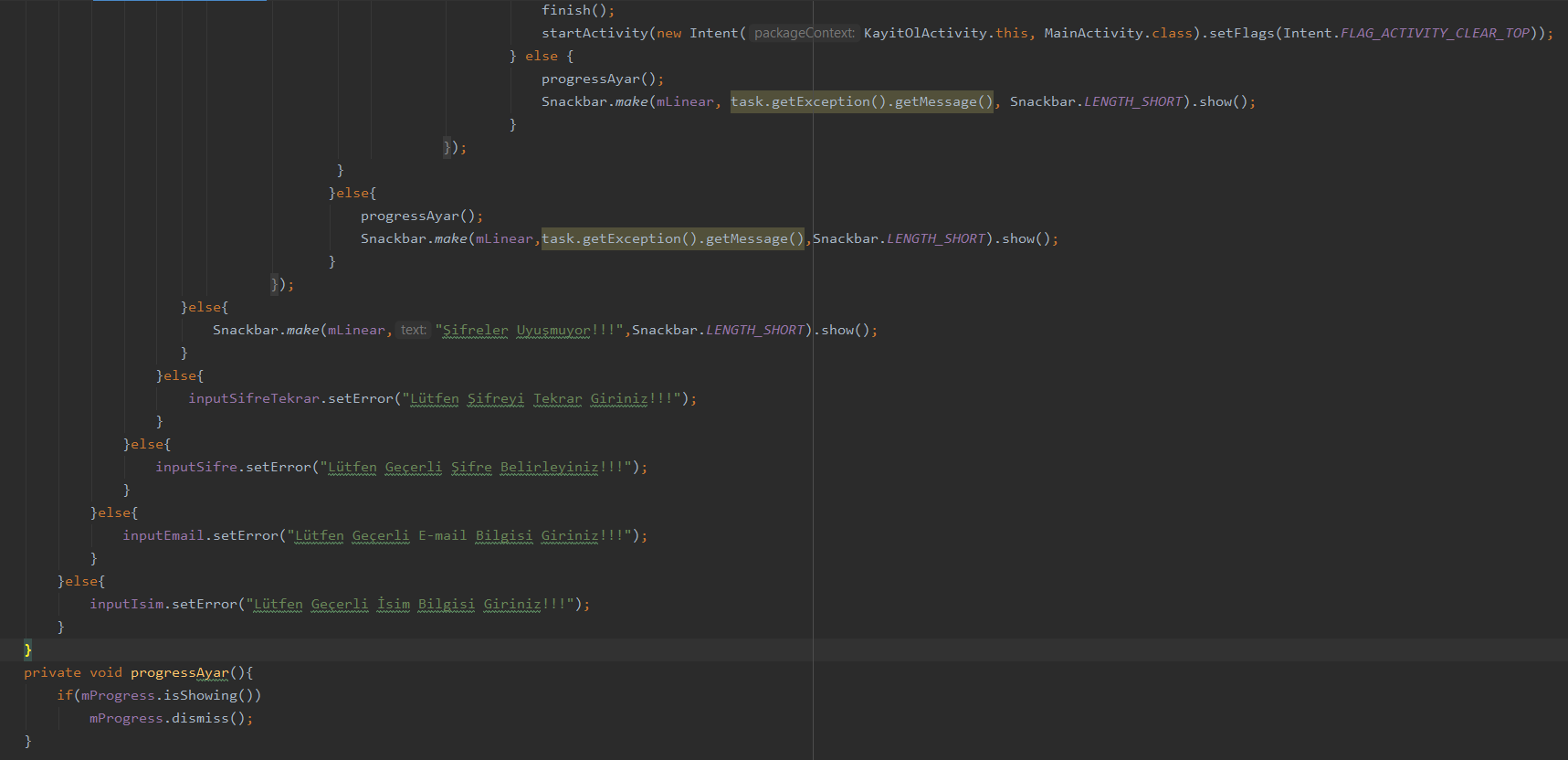


On create fonksiyonunda tüm yazılmış fonksiyonlar çalıştırılır. btnKayitOl fonksiyonun başında da

Girilecek olan isim, e-mail, şifre ve şifre tekrar alanlar tanımlanır.

İsim, e-mail, şifre ve şifre tekrar kısmı boş olmadığında ve şifre, şifre tekrar alanlarındaki şifreler aynı olduğunda kayıt olacak.

Şartlar sorunsuz çalıştığı sürece girilen bilgiler doğru olup olmadığı kontrol edildikten sonra girilen bilgiler ile Kullanıcılar adlı veri tabanına kayıt yapılacak.

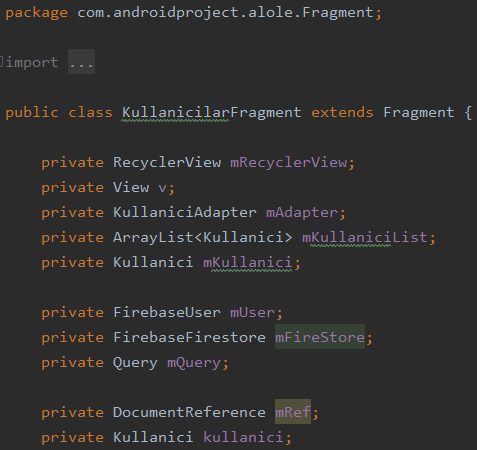


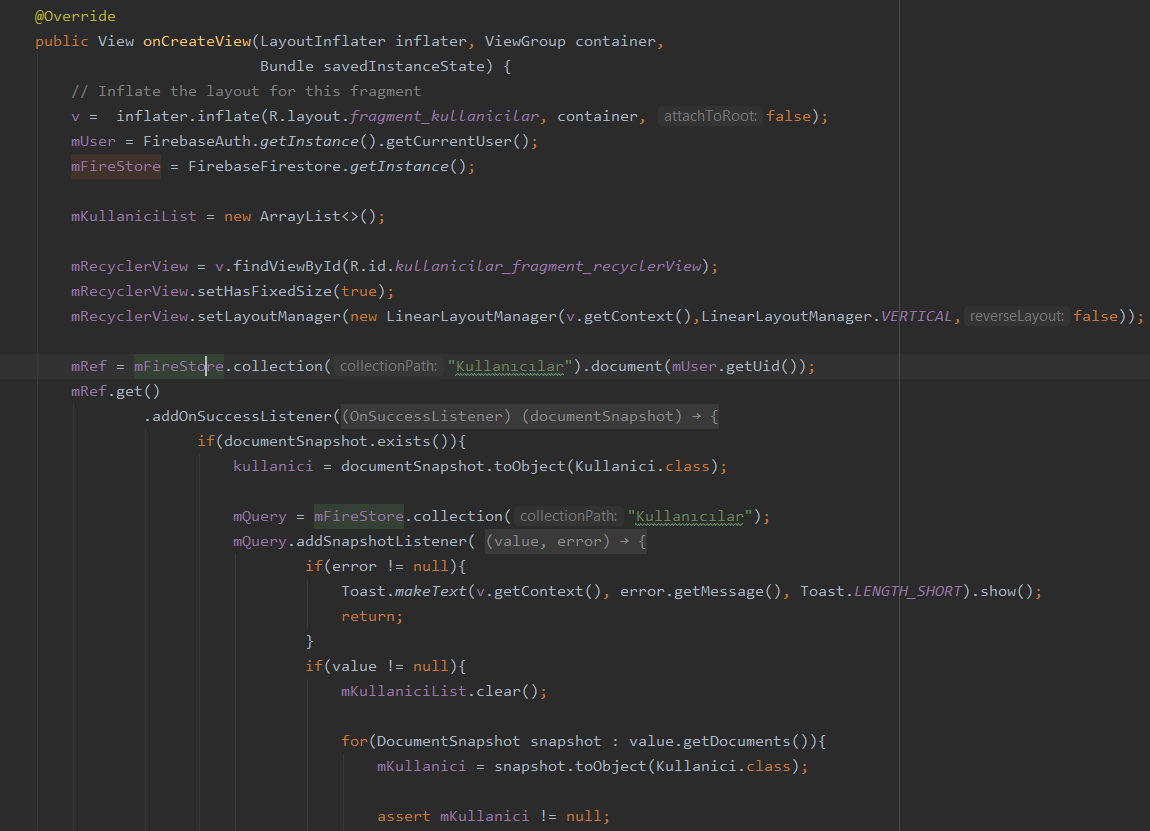
Kayıt yapma işlemi tamamlandıktan sonra direk ana

ekrana geçiş yapılacak.

İsim, e-mail, şifre alanları boş olup ve şifre alanlarındaki şifreler aynı değilse bu uyarılar karşımıza çıkacak. Ayrıca Kayıt olduktan sonra progressAyar fonksiyonunda yazılan kayıt yapılıyor yazısı karşımıza çıkacak. Sonra dismiss() fonksiyonu ile kapıtılacak.

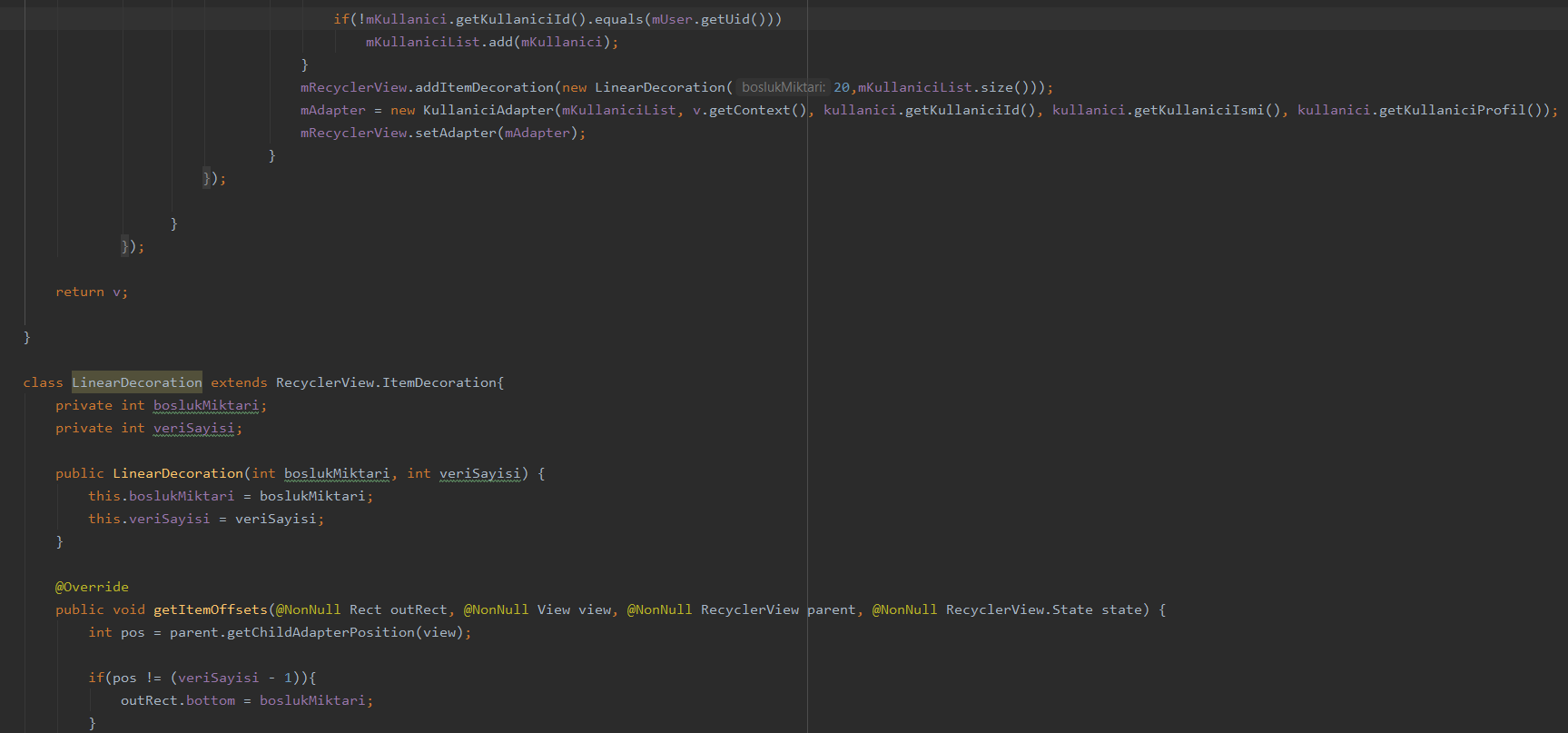
1. Kullanıcılar İle İlgili Ana Kodlamalar

 Gerekli olan parametrelerin tanımı.



Tanımladığımız parametreleri initialize ettik. V’ye dosyamın adını eklemek istediğim üst düzeni söylemek ve aslında henüz eklemek istemediğimi söylemekti. RecycleView da Layout Manager’ı kullanarak kullanıcıları belirli bir düzende sıralayacak. ArrayList() kullanılarak FireBase’den kullanıcıları çekecek.

Kullanıcılar veri tabanından kullanıcılar çekilecek. addSuccessListener içerisinde kullanılan documentSnapshot ile kullanıcı listesindeki verilere tek tek ulaşacağız. Mquery ile FireBase’deki tüm verilere ulaşacağız. Mesela uygulamaya girdiğimiz anda yeni bir kullanıcı eklenebilir. Anlık olarak görebilmemiz için addSnapShotListener kullandık. Bunun sayesinde biz uygulamayı kullanacağımız sırada yeni kaydolan tüm kullanıcıları görebileceğiz. Birde hatalar için gerekli hata blokları oluşturduk.



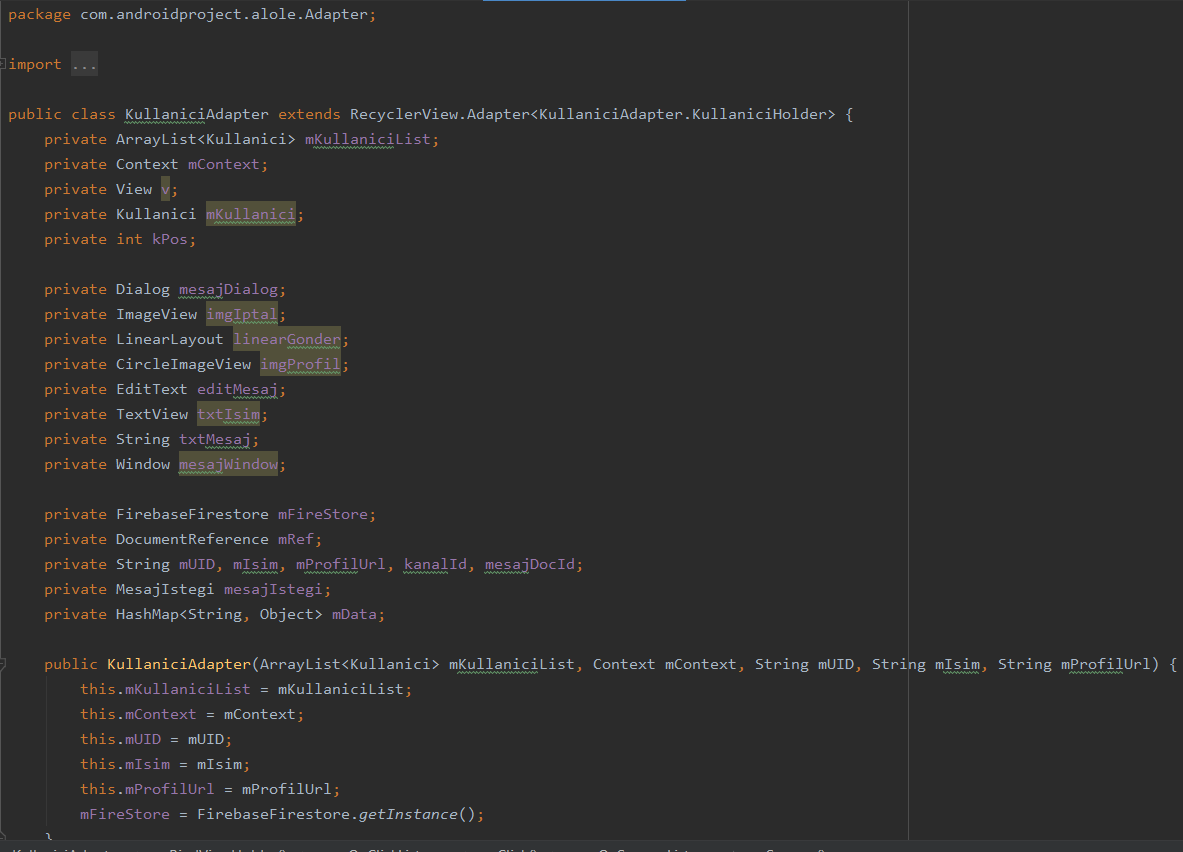
Bu class içerisinde tanımlanan fonksiyon ile boşluk miktarı ve veri sayısı kontrolü yapılabilecek parametreler oluşturuldu.

Burada kullanıcıların sayısı belirlenip ona göre sıralamaya yardımcı olacak.

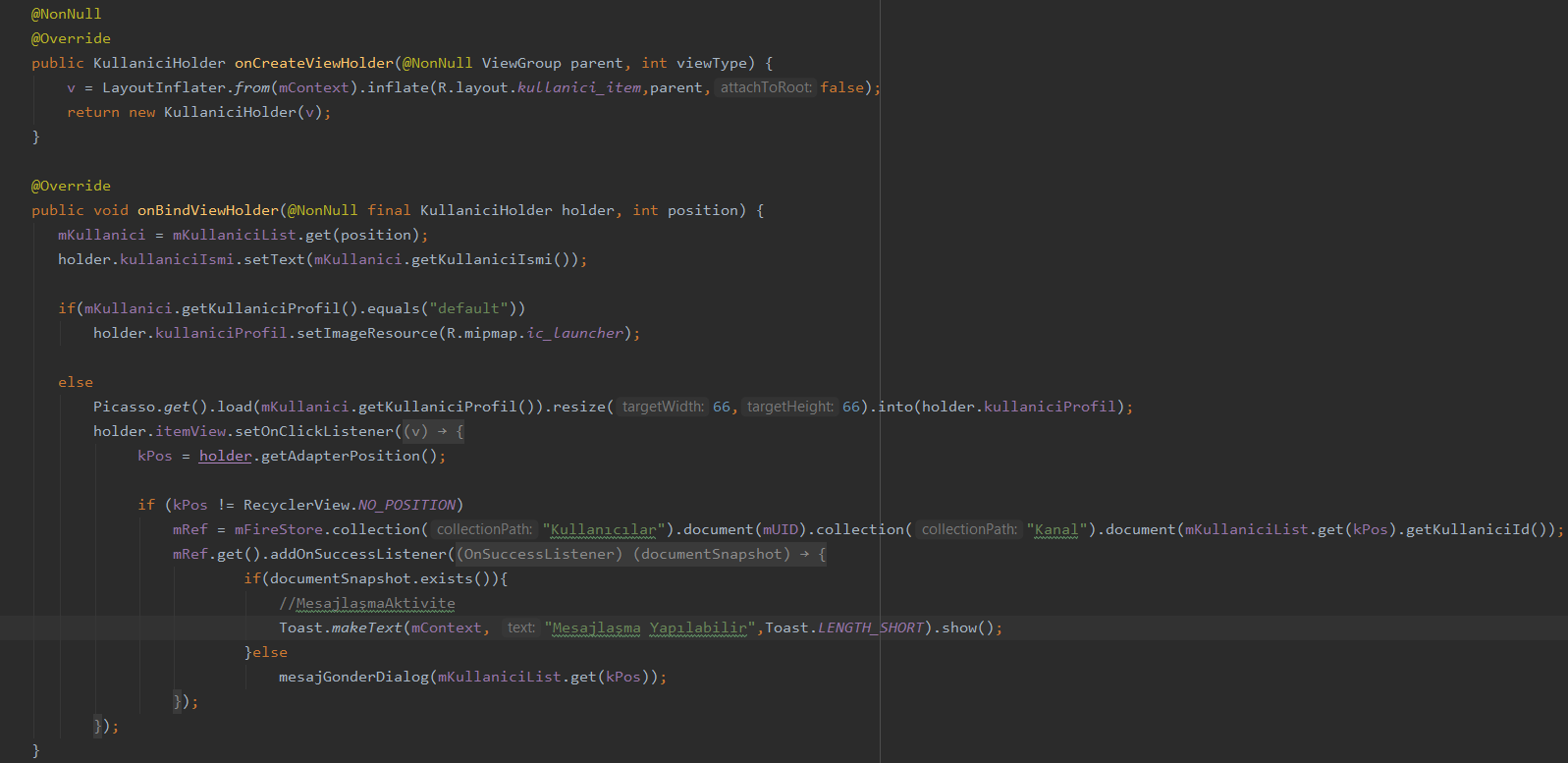
Giriş yapan kullanıcı dışında kullanıcılar gösterilecek,

RecycleView ile Arraylist’te tanımlanan parametre ile

Verilere gösterilecek.

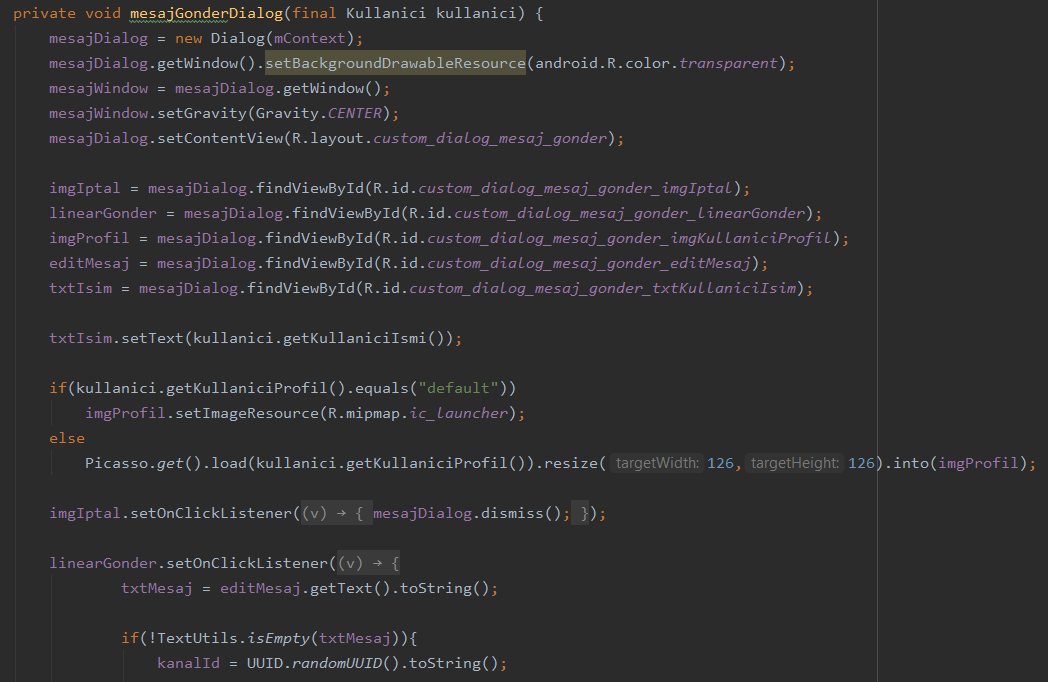


Gerekli olan tüm parametreler tanımlandı. Daha sonra bu parametrelerin constructorları oluşturuldu.



Burada kullanacağımız tasarımın bağlanımını yaptık. Buradaki v bağlantı kurmamızı sağlıyor.

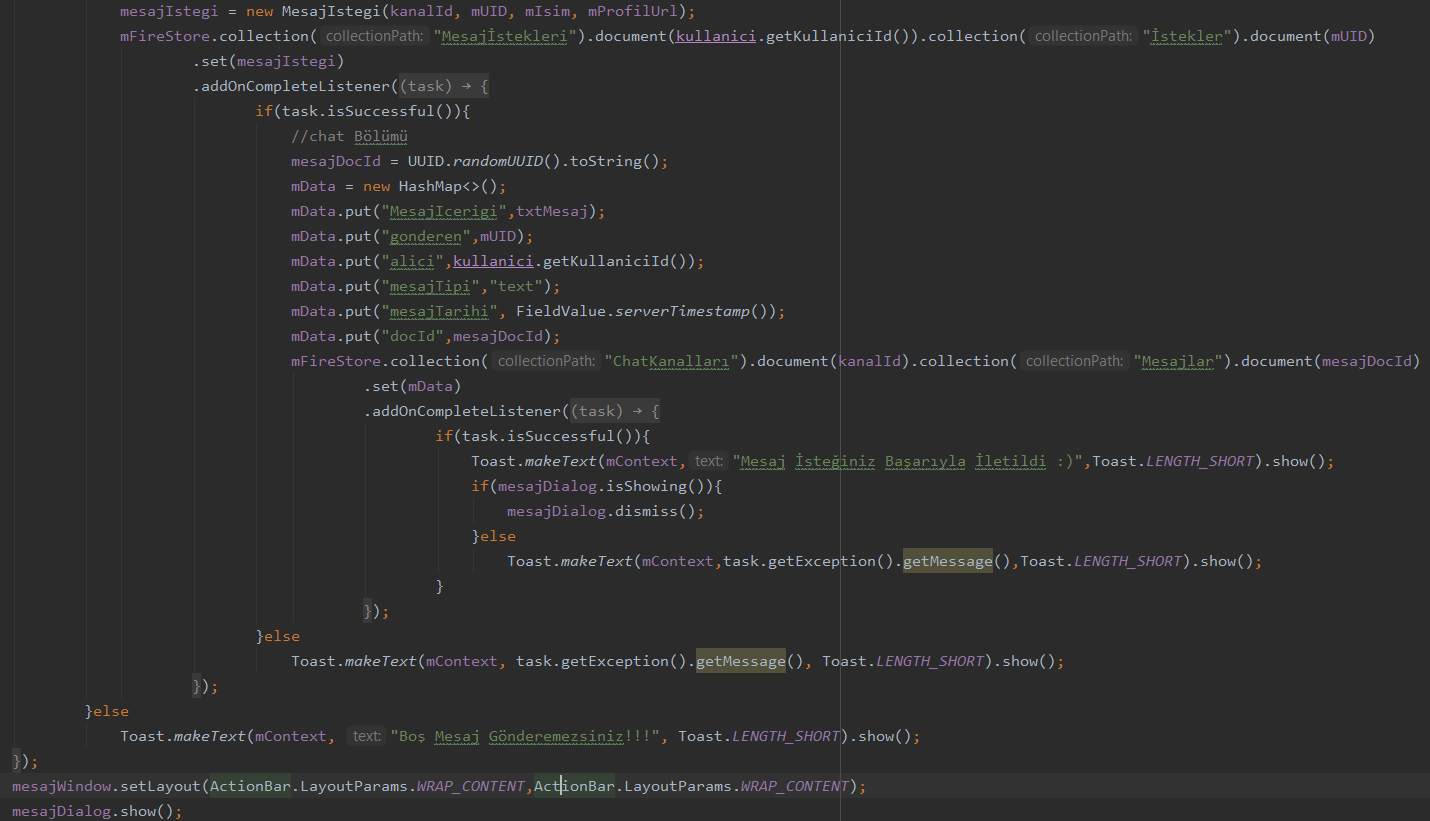
Burada gelen verileri aktarma işlemi gerçekleştiriyoruz. Yaptığımız kontrol içerisinde default ile eşit olduğu sürece profil resmini set edebiliyoruz. Değil ise resim yüklüyoruz. Else kısmında ise internetten resim yükleyebilmek için gradle kısmına picasso kütüphanesini ekledik. Yüklenen resim boyutları büyük olduğunda program kasabileceğinden resmin genişliğini ve yüksekliğini 66, 66 dp şeklinde ayarladık. Daha sonra bunu kullanıcı profilinin içine yükledik.

 Gerekli parametreleri

İnitialize ettik.

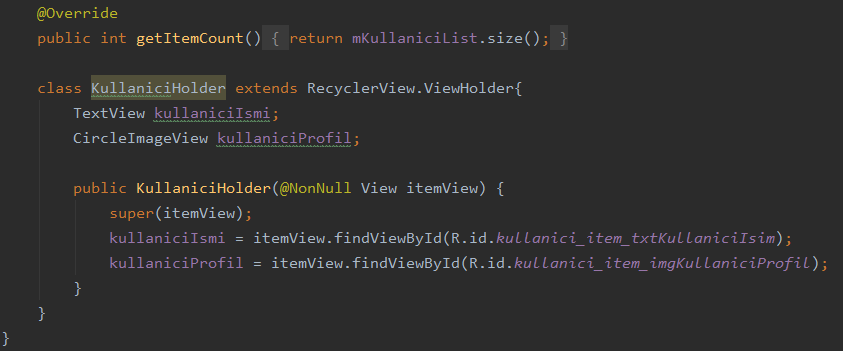
Mesajlaşma için gerekli olan kullanıcı bilgileri default ile eşit olduğu sürece profil resmini set ediyoruz. . Değil ise resim yüklüyoruz. Else kısmında ise internetten resim yükleyebilmek için gradle kısmına picasso kütüphanesini ekledik. Yüklenen resim boyutları büyük olduğunda program kasabileceğinden resmin genişliğini ve yüksekliğini 126, 126 dp şeklinde ayarladık. Daha sonra bunu mesajı gönderen kullanıcı profilinin içine yükledik.

SetOnClickListener ile açılan mesajlaşma penceresinin kapatma işlemi ve mesajlaşma işlemini gerçekleştirdik.



Mesaj İsteği Sınıfından gerekli bilgileri kullanarak. Firebase de Mesaj istekleri koleksiyonu içerisinde istekler kanalı oluşturarak hashmap ile gönderilen mesajların gizliliği sağladık. Mesajlaşma da mesajın içeriği, mesajı gönderen kişi, mesajın alıcısı, mesajın tipi, tarihi ve id’sini kullanarak Chat Kanalı oluşturduk. Oluşturduğumuz kanal içerisinde Mesajlar koleksiyonunda mesajları kaydedecek. Mesaj isteği iletildiğinde başarıyla gönderildiğine dair yazı karşımıza çıkacak. Mesajlaşma kutusu kapanacak.

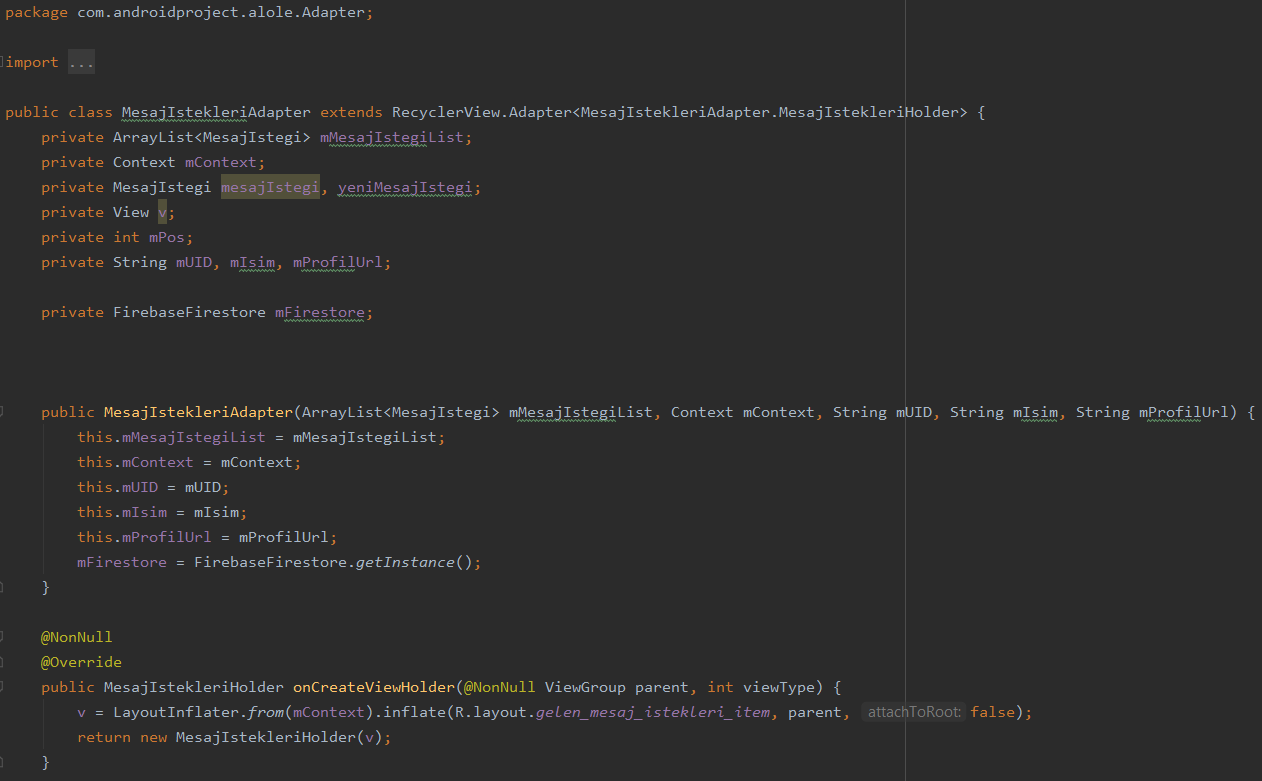
Mesaj isteği iletilmediği takdir de hata bloğu karşımıza çıkacak.



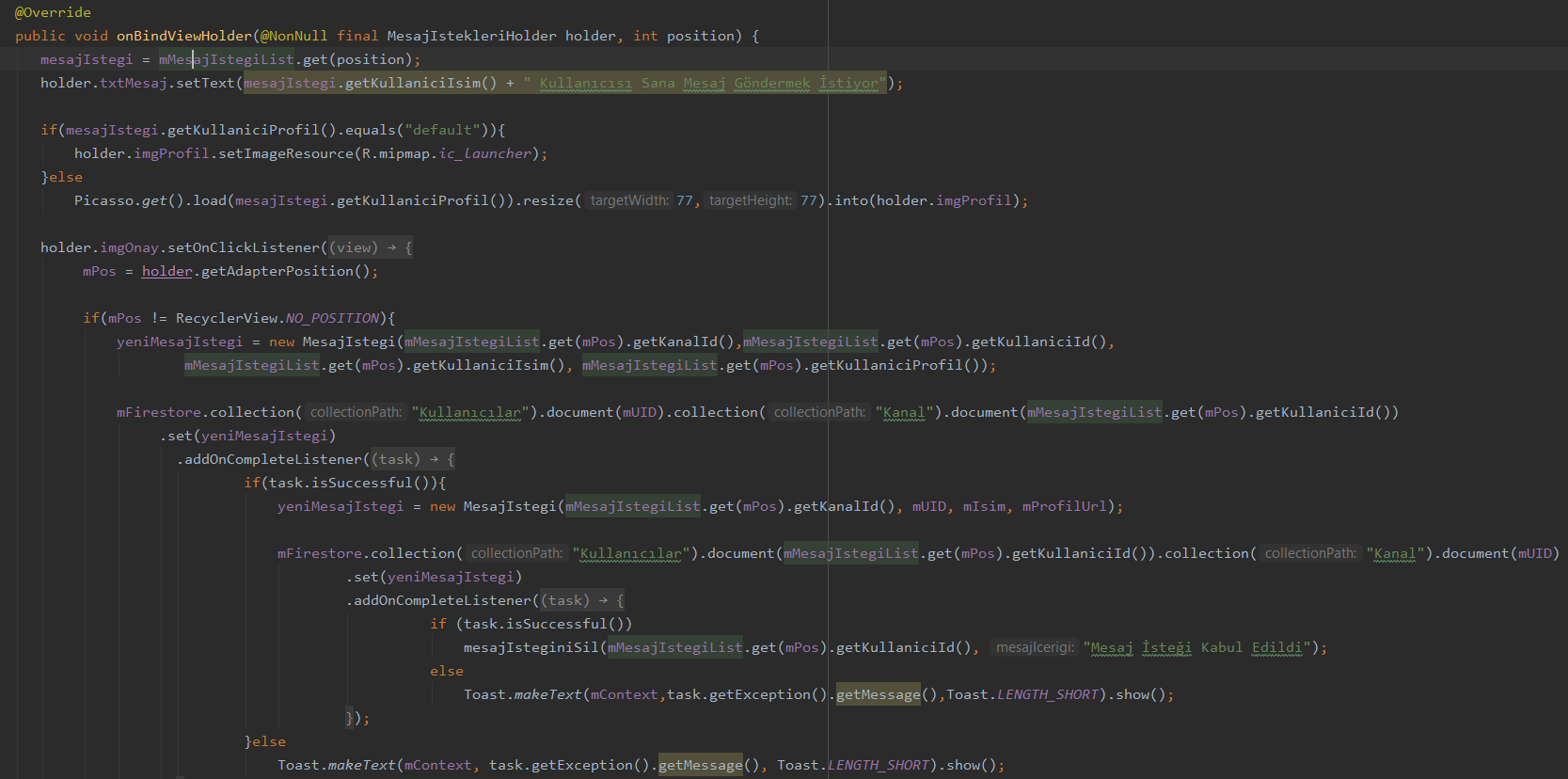
GetItemCount ile geri kullanıcı sayısı döndürecek.

Kullanıcı bir önceki kısımda kullanıcı holder içerisinde v’yi kullandık. V buradaki itemView’e denk oluyor buda bağlantı kurmamızı sağlıyor.

1. Mesaj İle İlgili Ana Kodlamalar

 Gerekli olan tüm parametreler tanımlandı. Daha sonra bu parametrelerin constructorları oluşturuldu.

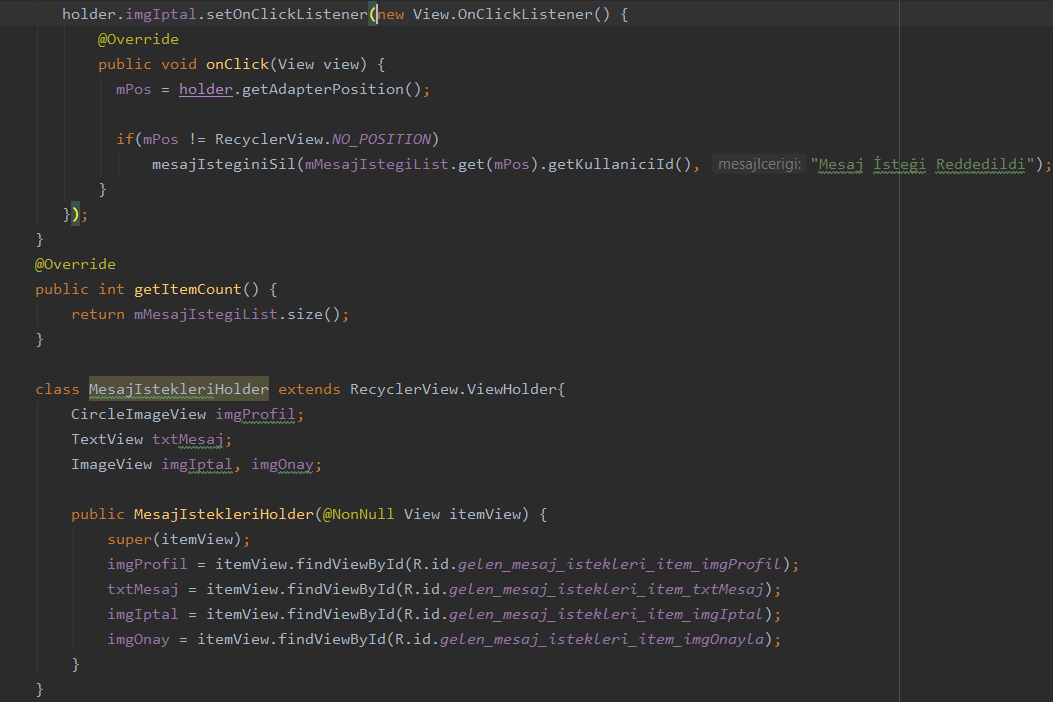
Burada kullanacağımız tasarımın bağlanımını yaptık. Buradaki v bağlantı kurmamızı sağlıyor.



Burada gelen verileri aktarma işlemi gerçekleştiriyoruz. Yaptığımız kontrol içerisinde default ile eşit olduğu sürece profil resmini set edebiliyoruz. Değil ise resim yüklüyoruz. Else kısmında ise internetten resim yükleyebilmek için gradle kısmına picasso kütüphanesini ekledik. Yüklenen resim boyutları büyük olduğunda program kasabileceğinden resmin genişliğini ve yüksekliğini 77, 77 dp şeklinde ayarladık. Daha sonra bunu kullanıcı profilinin içine yükledik.

Burada gelen mesaj isteğinin kanalıyla bağlantı kurup daha sonra gelen isteği kabul etme işlemi

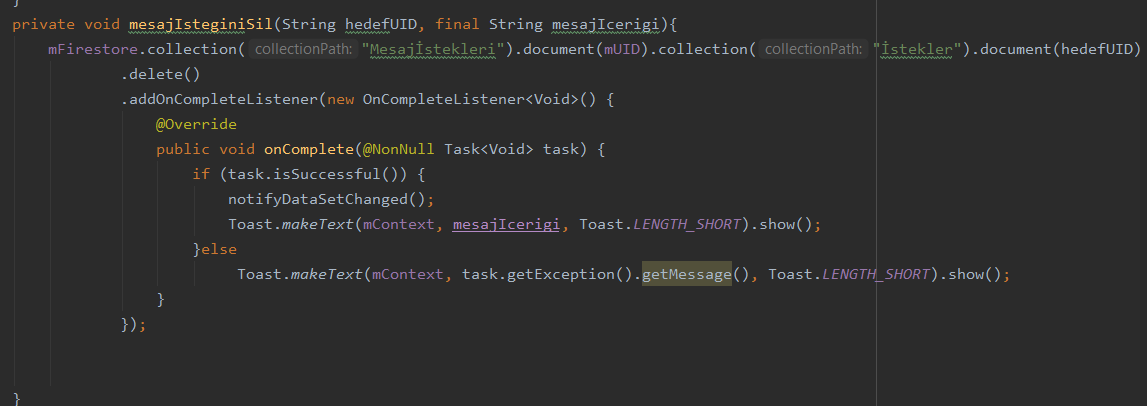
Kodlaması yapıldı.



Tasarlanmış mesaj isteğini iptal etme tuşunun çalışma mekanizması yapıldı.

GetItemCount ile geri kullanıcı sayısı döndürecek.

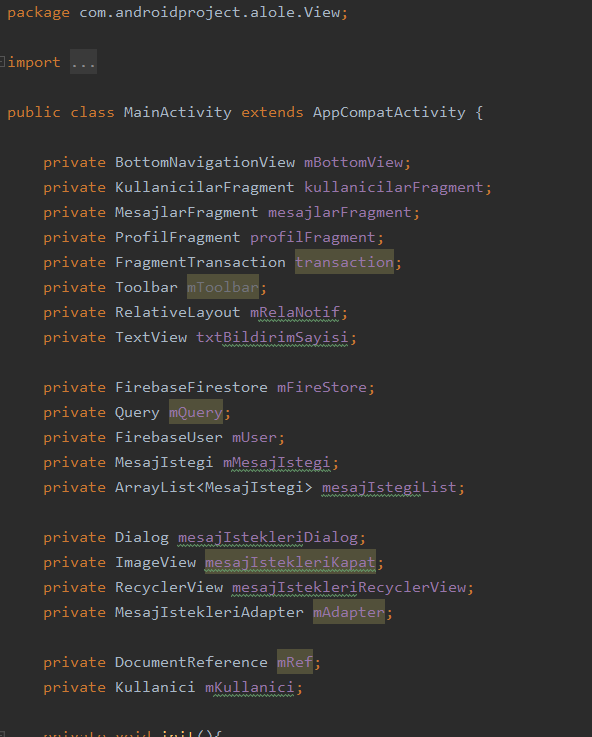
Bir önceki kısımda mesaj istekleri holder içerisinde v’yi kullandık. V buradaki itemView’e denk oluyor buda bağlantı kurmamızı sağlıyor.



Burada gelen mesaj istekleri silme metodunu yazdık. mFirestore ile Mesaj İstekleri koleksiyonundan istekler dokümanına bağlandık. Delete() metodu ile gönderilen isteği silme işlemi gerçekleştirilecek.

İşlem gerçekleşmezse gerekli hata bloğu önümüze çıkacak.

1. Ana Ekranın Ana Kodlamaları

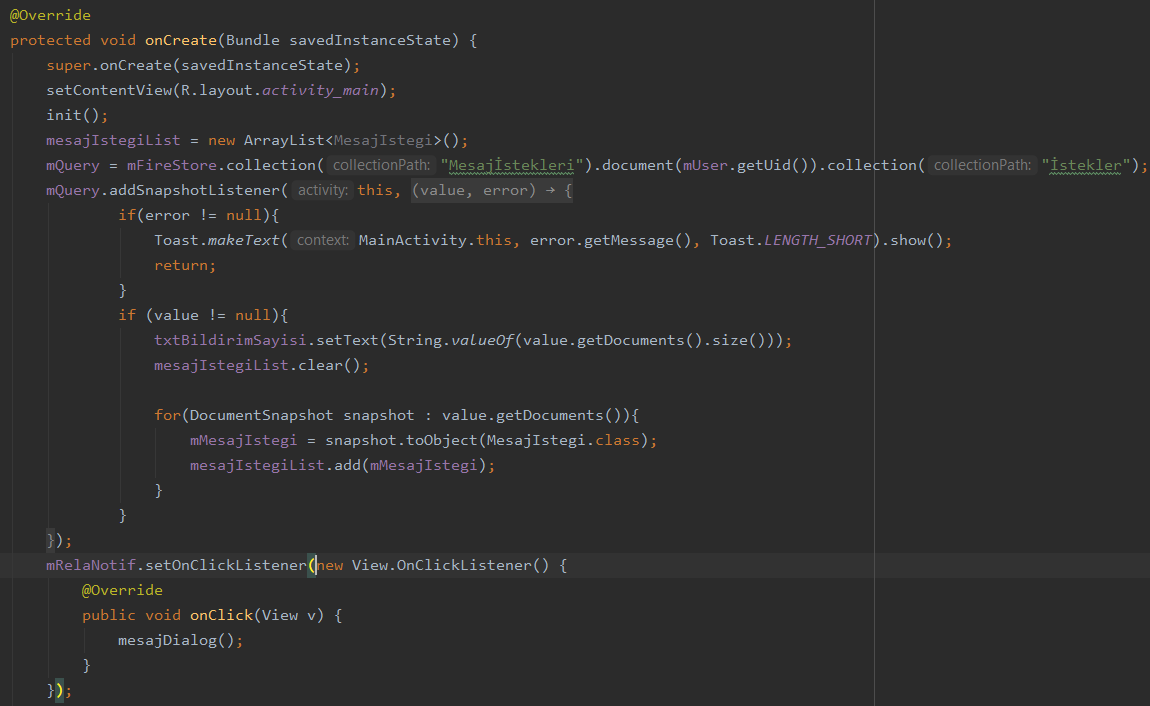
 Gerekli olan tüm parametrelerin tanımını

yaptık.



Burada Firebase’deki verilere bağlantıyı sağladık.

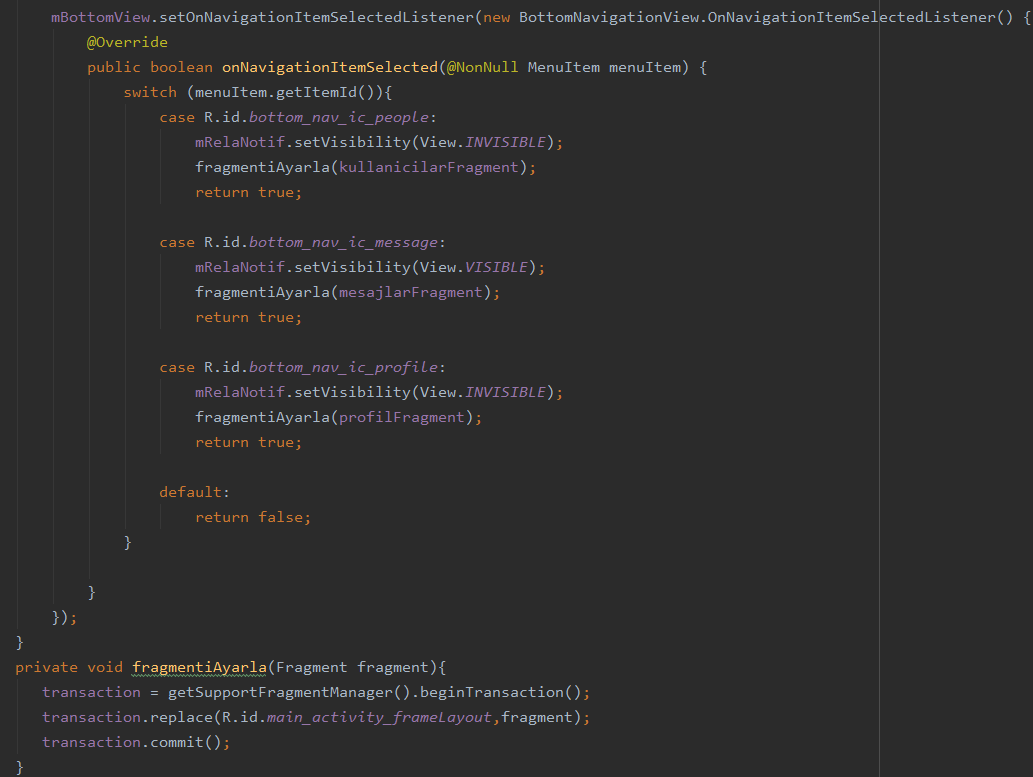
Başlangıçta çalışması için her bir class ve her bir değişken init() metodunda çağırıldı. değişkenler id'lere göre baz alındı. Burada kullanıcılar, mesajlar ve profil fragmenti bulunuyor.



Bir mesaj isteğinin olup olmadığını kontrol edebilmek için onCreate içinde mQuery = mFireStore.collection() komutunu kullandık. "Mesajİstekleri" adında koleksiyon adını verdik sonra document'in içine kullanıcı id ve koleksiyon adını ekledik. Sonra mQuery. AddSnapShotListener komutunu yazdığımızda bizden yeni bir aktivite istedi onEvent içinde if bloğunda hata null değilse bir hata vermiş anlamına gelir. Toast ile hangi hata ise o hatanın mesajını veriyoruz. Tekrar dönmesi içinde return komutu eklendi. Bir diğer if bloğunda ise değer boş değilse bildirim sayısını alıyoruz. mesajIstegi sınıfından bir ArrayList oluşturduk, bildirimi her aldığımızda, önceki bildirimleri tekrar görmemek için mesajIstegiList.clear() ile daha önceki mesaj istekleri temizlenmiş oluyor.

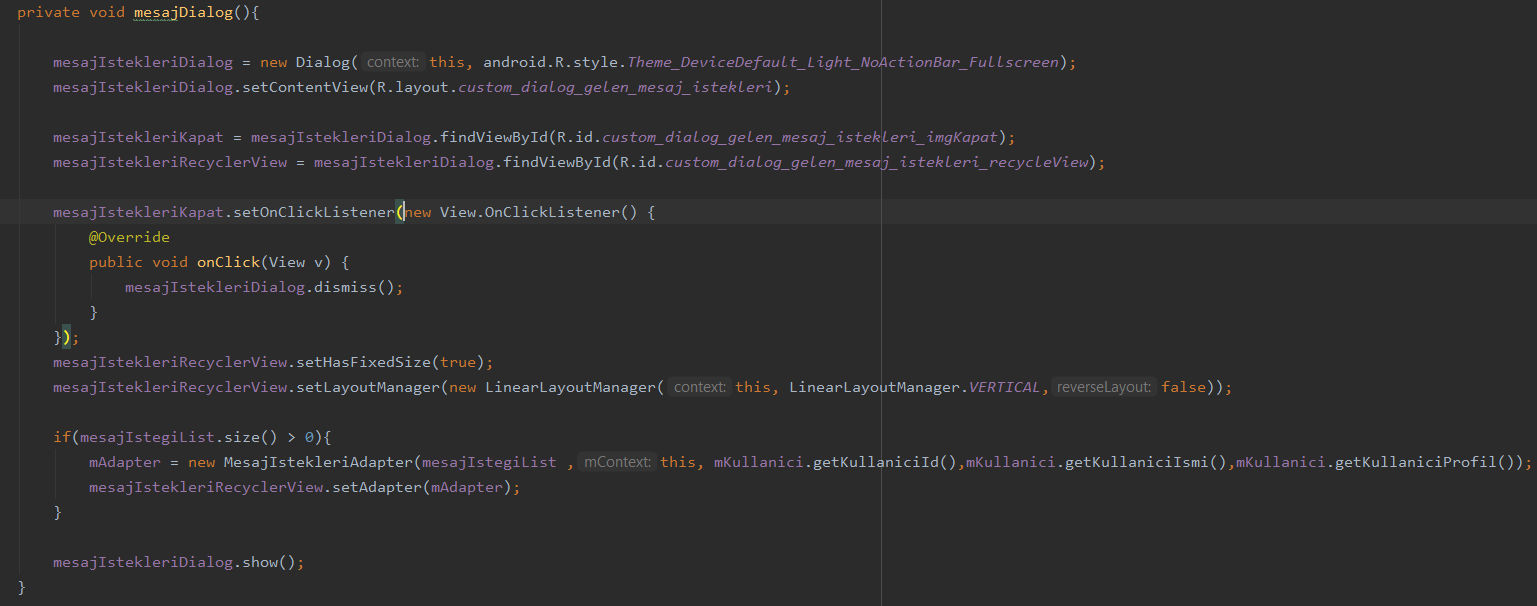
Ondan sonra veriler ekleniyor. Veriler üzerinde gezinebilmek için bir foreach döngüsü oluşturduk, gelen verilerimiz QuerySnapshot olduğundan bu verileri almak için DocumentSnapshot olarak almamız gerekiyor. mMesajIstegi değişkeni, MesajIstegi sınıfından gelen veriler eklenmiş oluyor.

Burada Mesajlaşma için gerekli olan metodu çağırdık.



Switch case komutunda ise kullanıcı hangi case durumunu seçerse, diğer durumlar invisible duruma gelir yani görünmezler.

Burada Fragment sayfalarını ayarlayacak metodu oluşturduk.



mesajDialog() adında bir metot oluşturuldu. Diyalog burada yeniden initialize edildi. Stil bölümünden Theme\_DeviceDefault\_Light\_NoActionBar\_Fullscreen olarak seçildi böyle olunca açılan Bar, tam ekran olarak açılır. OnClick içerisinde Mesaj isteği penceresinin kapatma işlemi gerçekleştirdik. İf bloğunun içerisinde mesaj isteğini gönderecek olan kişinin bilgilerini çağırdık.

1. FireBase Ekranları



