



T.C.
KARADENİZ TEKNİK ÜNİVERSİTESİ
FAKÜLTE ÖĞRENCİ STAJ DOSYASI

ÖĞRENCİNİN



ADI SOYADI : Muhammet Emin Gündoğar
BÖLÜMÜ : Bilgisayar Mühendisliği
PROGRAMI : 1.Öğretim
SINIFI : 2.Sınıf
NUMARASI : 407596

STAJ DÖNEMİ: 04/07/2022 - 05/08/2022



04/07/2022 TARİHİNDEN 10/07/2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Uygulama Arayüzü Tasarımı	9
SALI	Uygulama Arayüz Tasarımı	10
ÇARŞAMBA	Tab layout kullanımı	11
PERŞEMBE	Recycler view kullanımı	12
CUMA		
CUMARTESİ		
ÖĞRENCİNİN İMZASI : 		
KONTROL EDENİN İMZASI : 		


11/07/2022 TARİHİNDEN 17/07/2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ		
SALI		
ÇARŞAMBA	Retrofit Kullanımı	13
PERŞEMBE	Retrofit Kullanımı	14
CUMA		
CUMARTESİ		
ÖĞRENCİNİN İMZASI : 		
KONTROL EDENİN İMZASI : 		

**18/07/2022 TARİHİNDEN 24/07/2022 TARİHİNE KADAR BİR HAFTALIK
ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Uygulama içi güncelleme	15
SALI	Mvvm Kullanımı	16
ÇARŞAMBA	Exoplayer ile video oynatma	17
PERŞEMBE	Recycler view kullanımı	18
CUMA	Bottom sheet kullanımı	19
CUMARTESİ		
ÖĞRENCİNİN İMZASI : 		
KONTROL EDENİN İMZASI : 		

**25/ 07/2022 TARİHİNDEN 31/07/2022 TARİHİNE KADAR BİR HAFTALIK
ÇALIŞMA**

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Popup menu kullanımı	20
SALI	Grafik Kullanımı	21
ÇARŞAMBA	Git Branch Kullanımı	22
PERŞEMBE	Kayıt Ekran Tasarımı	23
CUMA	Firestore Kullanımı	24
CUMARTESİ		
ÖĞRENCİNİN İMZASI : 		

KONTROL EDENİN İMZASI

:

01/08/2022 TARİHİNDEN 07/08/2022 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ	Uygulama İzinleri	25
SALI	Glide İle Görsel İndirme	26
ÇARŞAMBA	Bottom Navigation Kullanımı	27
PERŞEMBE	Mesaj Ekranı Tasarımı	28
CUMA	Mesajlaşma Algoritması	29
CUMARTESİ		

ÖĞRENCİNİN İMZASI

:

KONTROL EDENİN İMZASI

:

/ /20 TARİHİNDEN / /20 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ		
SALI		
ÇARŞAMBA		
PERŞEMBE		
CUMA		
CUMARTESİ		
ÖĞRENCİNİN İMZASI	:	

KONTROL EDENİN İMZASI :

/ /20 TARİHİNDEN / /20 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ		
SALI		
ÇARŞAMBA		
PERŞEMBE		
CUMA		
CUMARTESİ		
ÖĞRENCİNİN İMZASI :		
KONTROL EDENİN İMZASI :		

/ /20 TARİHİNDEN / /20 TARİHİNE KADAR BİR HAFTALIK ÇALIŞMA

GÜNLER	YAPILAN İŞ	İZAHATIN BULUNDUĞU YAPRAK NO'SU
PAZARTESİ		
SALI		
ÇARŞAMBA		
PERŞEMBE		
CUMA		
CUMARTESİ		
ÖĞRENCİNİN İMZASI :		

KONTROL EDENİN İMZASI :

Kontrol Sonucu		
KISIM	Kotlin İle Android Geliştirme	Yaprak No:1
YAPILAN İŞ	Uygulama Arayüzü Tasarımı	Tarih: 04/07 /2022

İlk staj günümde geç kalmamak için sabah erkenden kalkıp gittim.Gittiğimde daha işin başlamasına yarım saat vardı.Şirkete daha kimse gelmemişti.Heyecanlıydım.Benim gibi ilk günü olan bir çalışan gelince onunla muhabbet edip biraz rahatladım.Sonra stajımla ilgilenecek olan Android ekibi gelince onlarla tanıştım.Sonra onlardan task istedim.Yaptıkları avukatlık uygulamasının arayüzünden bir parça verdiler.Androidde kodlama konusunda iyiyimdir ancak tasarım konusunda gerçekten bir bilgim yoktu.Ve bu ilk tasarım beni biraz da zorladı.Ama çok şey öğrendim.bir textview'a elevation verip nasıl button gibi gözükmesini sağlayabiliriz.Bu ilk öğrendiğim şeydi.Sonra bir imageview içerisine başka bir imageview koymayı öğrendim.Bu da benim için güzel bir tecrübe oldu.Bunu yapmanın yolu da dıştaki imageview'ın width ve height'i kadar margin vermekmiş.Bu uygulama basit gözükse de bana tecrübe anlamında çok şey kattı ve tasarım anlamında bana bir özgüven verdi.Ama staj yapmamın en büyük katkısı ise internette çok uğraşıp bulacağım bu bilgilere kolayca tecrübeli yazılımcılar tarafından erişebilmek oldu.Sağolsunlar ilk günümde tüm sorularımda yardımcı oldular.Aşağıdaki uygulamada kullandığım widgetlar:Imageview,Textview,Edittext,Checkbox ve button oldu.Buttonun rengini de kendim seçtim ve hoşuma gitti.Umarım önümüzdeki günlerde daha güzel tasarımlar yaparım.

Profili Düzenle



Stajyer Avukat

Avukat

name

surname

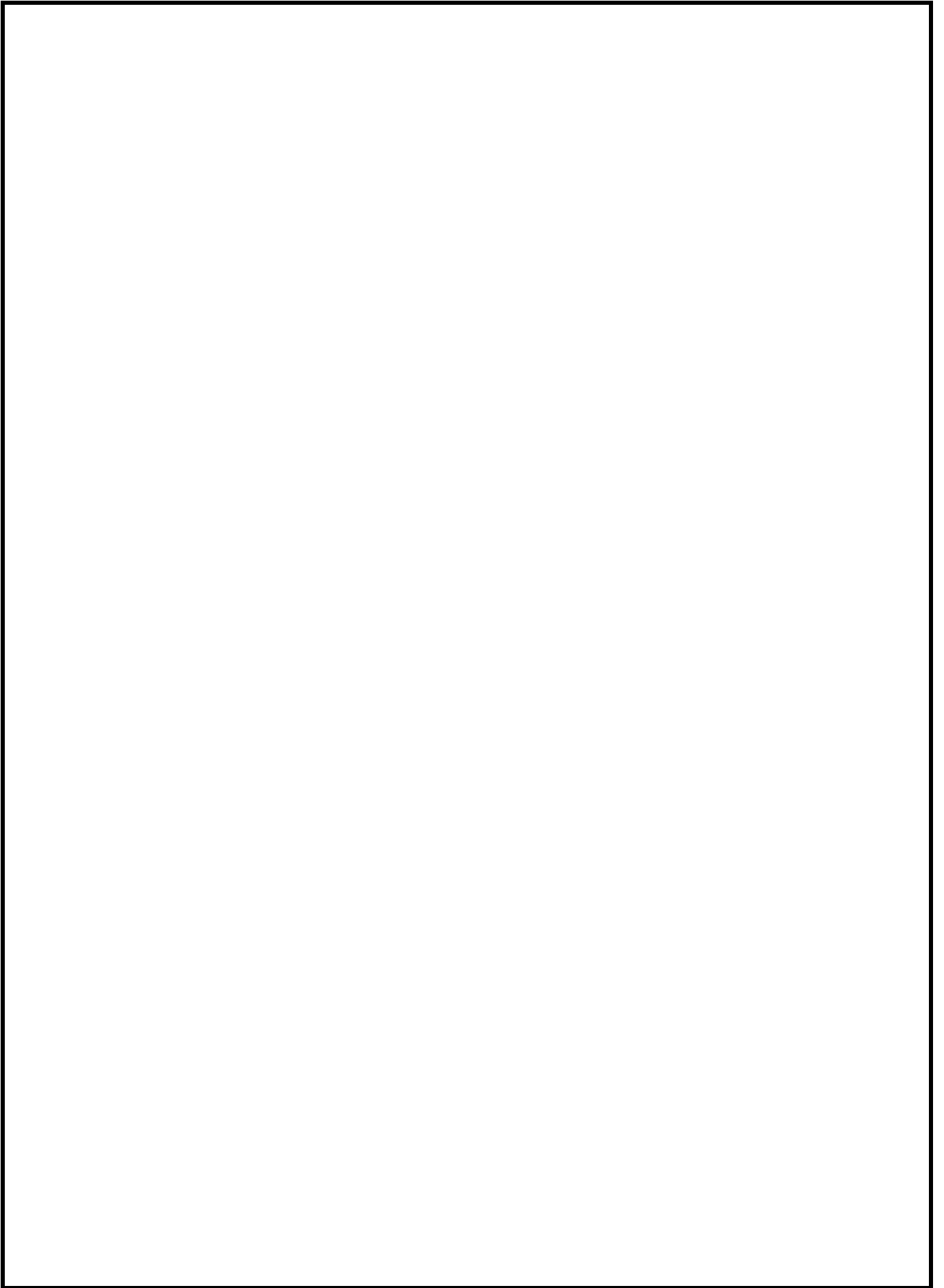
e-mail

Lawyer no

phone number

☐ Acil Avukatı listesinde görünmek istiyorum.

Güncelle



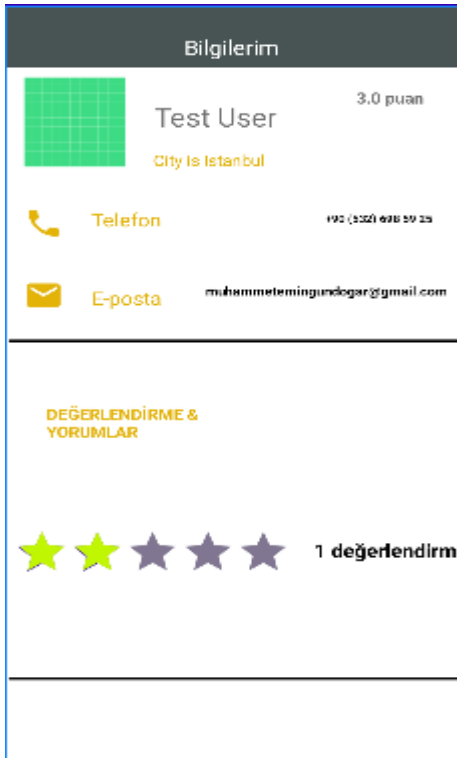
KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:2
YAPILAN İŞ	Uygulama Arayüz Tasarımı	Tarih: 05/07 /2022

Bugün şirkette 2.günüm.Geldim ve başka bir ekran tasarlamaya başladım.Bu ekranda kullanıcının bilgileri olacaktı.Ama üstteki action bar kaldırılıp yerine özel bir view koyulacaktı. Action bar kaldırdıktan sonra onun yerine view koymam gerekiyordu.

```
<View
    android:id="@+id/view"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:background="#485353"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

Yukarıdaki kodla da ekranın üstüne view yerleştirdim.Daha sonra svg iconlar oluşturdum.Renklerini de bana gösterilen renklerle aynı yapmaya çalıştım.Sayfanın içindeki parçalar arası geçişler için viewlar kullandım.En sonda da rating bar koydum.Bunu hayatımda ilk defa kullandım.



Böylece bir projeyi daha gerçekleştirmiş oldum.Bu projede tasarım konusunda yeni şeyler öğrenmiş oldum.Gözüme de güzel geldiği için hoşuma gitti.Şirketteki çalışanlara gösterdiğimde onlar da güzel olduğunu söylediler.

KONTROL SONUCU 		
KISIM	Kotlin ile Android geliştirme	Yaprak No:3
YAPILAN İŞ	Tab layout kullanımı	Tarih: 06 /07 /2022
<p>Bugün şirkette 3.günüm ve heyecanımı yenmeye başladım.Sabah gidip yeni bir task almak istediğimi söyledim.Developer arkadaşım bana kendi staj döneminde yaptığı getir uygulamasının bir arayüz sayfasını gösterdi ve bunu yapıp yapamayacağımı sordu.Ben de denemek istediğimi söyledim ve yapmaya başladım.Öncelikle kullanmam gereken yapıları öğrendim.Üstte bir tablayout kullanacaktım.Tablayout ise aşağıdaki görseldeki gibi bir yapı</p>  <p>Bu yapı sayesinde fragmentlar arası geçişi daha hızlıca ve güzel bir şekilde yapıp kullanıcının ux deneyimini arttırma imkanı elde ediyoruz.Tab layout ile ilgili videolar izledim ve dokümantasyonlar okudum.Tab layoutun kaydırma özelliğini de eklemek için viewpager adlı yapıyı eklemem gerektiğini de öğrendim.Viewpager kısaca açıklamam gerekirse kaydırma işlemleriyle başka görseller,ekranlar görüntülememizi sağlayan yapıdır.</p> <pre> <com.google.android.material.tabs.TabLayout android:id="@+id/tabLayout" android:layout_width="match_parent" android:layout_height="wrap_content" android:background="@color/teal_200" app:tabIconTint="@color/black" app:tabIndicatorColor="@color/white" app:tabSelectedTextColor="@color/white" app:tabTextColor="@color/black" /> <androidx.viewpager2.widget.ViewPager2 android:id="@+id/viewPager" android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1" /> </pre> <p>Yukarıdaki xml kod parçacığında tablayout ve viewpager kullanımını görebilirsiniz.Daha sonrasında getir sayfasının tasarımını yapmaya başladım.Bu tasarımda üstte bir view olacak getirlemek yazısı iki farklı renkten oluşması gerekiyor bunun için kod araştırmaya başladım ve tek satır kodla hallolduğunu görünce mutlu oldum.Html kodlarını kullanıp yapabiliyormuşuz.</p> <pre> val text = "getiryemek" binding.getirTitle.setText(Html.fromHtml(text)) </pre> <p>Html bilmenin burada işime yarayacağını düşünmemiştim.Daha sonrasında bir cardview oluşturup içine bir de imagebutton eklemem gerekiyordu.Bunu da daha önce yapmamıştım ve ilk defa yaptım ama zorlanmadım.Cardview'a biraz da elevation verdim 15 dp de Radius verip daha güzel gözükmelerini sağladım.Daha sonrasında ise recycler view oluşturmam gerekiyordu.Ancak saat 6 ya geliyordu ve işten çıkma vaktim geldiği için işe ara verdim ve son yarım saatte diğer developerları izlemeye karar verdim.Onları burda gördüğüm sürece öğrenmem gereken şeyler olduğu için sürekli onlara sektörle ilgili merak ettiklerimi soruyorum.Böylece bir günün daha sonuna geldim.</p>		

KONTROL SONUCU 		
KISIM	Kotlin ile Android geliştirme	Yaprak No:4
YAPILAN İŞ	Recycler view kullanımı	Tarih: 07 /07 /2022
<p>Bugün iş yerine geldim ve getir uygulamama devam ettim.En son tab layout ve view pager oluşturup ekrandan da birkaç yapıyı oluşturmıştım.Bugün de recycler view ve geri kalan kısımları yapacaktım.Öncelikle recycler viewden başladım.Recycler view için bir adapter sınıfı bir item layout bir adet de bu itemi barındıracak bir layout gerekiyor.Öncelikle item view oluşturmayla başladım.Android projelerde öncelikle layoutlar oluşturulur,ona göre kodda işlemler yapılır.Item olarak aşağıdaki gibi bir şey tasarladım.</p> <div data-bbox="529 734 1002 965" data-label="Image"></div> <p>Yapı kısaca şu şekilde.Üstte internetten bulduğum bir imageview aşağıda iki textviewden oluşuyor.Getirin kullandığın renklerin benzerini kullanmaya çalıştım.Daha sonrasında bu layoutun anlam kazanacağı adapter class'ını yazmaya başladım.Adapter class Recycler view adapter classından miras alması gerekiyor.Daha sonrasında inner class olarak bir de view holder yazmam gerekiyordu.Bu da görünüm tutucu olarak görev görüyor yani layoutumuzun barındığı sınıf diyebiliriz.Sonra da bir abstract class olan Recycler View.Adapter'dan miras aldığımız için 3 tane gerekli methodu override etmemiz gerekiyor.</p> <p>getItemCount,onCreateViewHolder,onBindViewHolder getItemCount methodunda recycler view'da kaç eleman olacaksa onu döndürüyoruz. onCreateViewHolder'da ViewHolder objesi döndürüyoruz.</p> <p>onBindViewHolder'da her eleman için bize verilen position parametresi sayesinde her elemana hangi değerin atanacağını belirliyoruz.Adapter classını da yazdıktan sonra verileri daha düzenli ve kodun daha okunabilir olması için model class oluşturmaya karar verdim.Modelde bitmap yani görsel,title ve description aldım.Daha sonra recycler view'in ait olduğu fragmenta girip recycler view'ı kullanabiliriz.Bu arada fragment içerisinde view binding kullandım.Çünkü view binding eskiden kullanılan Android-extensions aksine bize daha güvenli bir şekilde layout elementlerine erişme imkanı veriyor.Böylece başka classtan alakasız bir layouta erişip uygulamasının çökmesi gibi sorunların önüne geçebiliyoruz.Daha önce activitylerde kullanmıştım ancak fragmentlarda biraz daha farklı kullanılıyormuş onDestroyView'da binding'i null yapmazsak memory leak'e yol açıyormuş.Fragmentta recycler view kullanımını bitirdikten sonra.View pager için de bir adapter yazıyoruz.createFragment methodunu override edip position'a göre göstereceğimiz fragment seçiyoruz.main activity'de tablayout mediator sayesinde viewpager ile tablayout birbirine bağlanıyor.Böylece app kullanıma hazır hale geldi.Developer arkadaşla uygulamamı gösterdiğimde beğendiğini söyledi.Ben de teşekkür ettim.Sonraki gün başka uygulama vereceğini söyledi.Böylece bir günün daha sonuna geldim.</p>		

KONTROL SONUCU



KISIM	Kotlin ile Android geliştirme	Yaprak No:5
YAPILAN İŞ	Retrofit Kullanımı	Tarih: 13/07 /2022

Şirkete geldiğimde developer arkadaştan bu sefer retrofit adlı kütüphaneyi kullanacağım bir proje istedim.Retrofit adlı kütüphane ile ilgili birçok tutorial izlemiştim.Bilgim vardı ancak daha önce hiçbir kişisel projemde kullanmamıştım.Kısaca açıklamak gerekirse retrofit http requestleri yapmamızı sağlayan bir Android kütüphanesi,REST API'lerden veri çekmek için kullanılıyor.Bana key gerektirmeyen elindeki apilerden birini verdi.Api içerisinde filmler vardı.Bu apiyi Postman adlı api aracıyla test ettim.Ve 200 sonucunu alınca Ben de Retrofit film Project diye bir proje açıp çalışmaya başladım.Öncelikle retrofiti ve diğer kütüphaneleri implemente ettim.Bu kütüphane bazı yan kütüphaneleri de barındırıyor.Moshi,Gson gibi bunlar da JSON objelerini kotlin objelerine dönüştüren kütüphaneler.Ben Gson seçip kullanmaya başladım.Ama bu projede diğer projelerimden farklı olarak bitirdikten sonra değil de çalışırken githuba adım adım commit atmaya karar verdim.Çünkü şirkette gördüğümde bu şekilde yapıyorlardı.Ben tek çalışacağım için branch açmayacaktım,tek bir branch üzerinden gitmeye karar verdim.Android studio'ya github hesabımı entegre ettikten sonra çalışmaya başladım.Commit ve push kavramlarının ne olduğunu öğrendim.Commit local branch'e değişiklik yollamaya,push ise remote branch'e değişiklikleri yollamaya verilen admış.Bunlar şimdilik yeterli olacaktı.Projeyi açtıktan sonra ilk başta recycler view'dan bilindiği üzere item layout oluşturdum.



Filmin adı,yönetmeni,yayın tarihi olacak ve solda da silme için bir button olacak.Bunu ilk başta eklememişim.Ancak mentörüm bu şekilde yapsan daha iyi olur diyince silmeyi de ekledim.Listeden eleman silip bir de toast mesaj göstermenin kodu aşağıdadır.

```
movieList.removeAt(position)
notifyItemRemoved(position)
notifyItemRangeChanged(position, movieList.size)
holder.itemView.visibility = View.GONE
val deletedItem = movieList[position].title
Toast.makeText(
    holder.itemView.context,
    text: "Deleted film is $deletedItem ",
    Toast.LENGTH_LONG
).show()
```

Daha sonra her zamanki gibi recycler view adapter'imi yazdım.Geçen projeden aşına olduğum için bu sefer çok daha hızlı yazdım.İlk öğrendiğimde recycler view konusunun beni ve birçok kişiyi zorladığını itiraf etmeliyim.Daha sonrasında modelimi yazmaya başladım.Bu modeldeki isimlerin jsondaki verilerle aynı isimde vermek daha kolay çalışmamızı sağlıyor.Daha sonra api isteğinin nasıl yapıldığıyla ilgili tutorial izlerken saat 6'ya geldi ve bilgisayarımı kapatıp işten ayrıldım.Sonraki gün Api isteğini yapıp devamında da navigation sayesinde details fragment işlemlerini halletmeyi düşünüyorum.

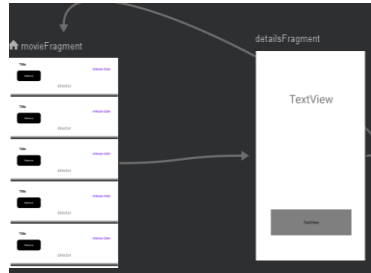
KONTROL SONUCU**KISIM****Kotlin ile Android geliştirme**

Yaprak No:6

YAPILAN İŞ**Retrofit Kullanımı**

Tarih: 14 /07 /2022

Bugün şirkete gelip projeme geçen gün kaldığım yerden devam ettim.Öncelikle Apiden http request için gerekli tutoriali izledim ve çok kolay olduğunu fark ettim.Bir interface oluşturuyoruz.Bu interface içerisinde companion object oluşturup base url adında bir değişken oluşturuyoruz.base url parametrelerden bağımsız endpoint'imizin adresi diyebiliriz.Companion object ise java,c++ dillerinden aşına olduğumuz static keywordünün kotlin karşılığı,class adıyla sınıf değişkenine erişebiliyoruz bu sayede,Daha sonra da get annotation oluşturup api parametrelerini veriyoruz.Parametre dediğimiz şey de linkin base url'den arta kalan kısmı gibi bir şey.Biz tek bir şekilde sorgu yapacağımız için bir tane get annotation oluşturduk.Temel olarak rest apilerde get,post,update,delete işlemleri vardır.Diğer 3 ü sunucu tarafında gerçekleşen işlemler.Bir gün onlarla da çalışmak istiyorum.Test ettiğimde verilerin gelmediğini fark ettim.Logcatte hatanın sebebini baktığımda.No adapter attached hatası gördüm.Breakpointler koyup debug yapmayı developerlardan görmüştüm.Ben de bunun faydalı olabileceğini düşündüm.Verilerin geldiği yere baktığımda null geldiğini gördüm.Ama tekrardan apiyi postmande denediğimde 200 yani success geliyordu.Açıkçası şaşırmıştım.Çünkü her şeyi dokümantasyona tutoriallara uygun şekilde yaptığımı düşünüyordum.Denemeye devam ettim.2 saat zaman ayırdıktan sonra şirketteki en tecrübeli developer ömer abinin yanına gittim ve yardım istedim.O da sağolsun beni kırmadı ve yardım etti.O da bir debug yaptı ve çok geçmeden hatanın sebebini anladı.Gelen veriyi ben tek bir model olarak alıyormuşum,ancak gelen veri modellerin bir listesiymiş.Bunun sebebi de benim JSON bilmememden kaynaklıydı.Sorunum hallolduktan sonra filme tıklandığında o filmin detaylarının gösterildiği fragment yapımına geçtim.Bir layout file açıp tasarımımlı oluşturudum.Bu tasarımda filmin adı ile yönetmeni göstermek istedim.Bunun için iki textview kullandım.Daha sonrasında navigation kullanarak fragmentlar arası geçişi yapacaktım.Ama tıklanan filmin bilgisini de diğer fragmenta aktarmam gerekiyordu.Bunun için de actionlara yani fragmentlar arası geçişlerdeki oklara argument vermemiz gerekiyor.



Normalde bir string veya int yollayacağımız zaman direkt ekleyebiliyoruz,ancak ekleyeceğimiz bir objeyse bunun için o sınıfı parcelable interface implement etmesi gerekiyor.Bunu da yaptıktan sonra Movie modelini karşı fragmenta yollamayı başardım.Ancak bu sefer de gönderilen filmin adının sayfanın üstünde gözükmesini istedim.Genelde gerçek hayattaki applerde bu şekilde olduğu için yapmam gerektiğini düşündüm.Ancak bunun stringi label alanına yazmaktan farklı olduğunu gördüm.Movie.title yazdığımda app çöküyordu.Movie yazdığımda garip bir yazı gözüküyordu.Araştırdım ancak bulamadım.Ben de ömer abiye sorma kararı aldım.Yanına gittiğimde bir iki baktı.Sonra Model classa gidip toString methodunu

override etti ve return `Movie.title` yazınca anladım ki biz labelda yazdırıldığımızda `toString` methodunu override ediyor.Sadece movie yazdığımda Hash adresini döndürüyordu.Ömer abi de `toString` methodunun davranışını değiştirip `Movie`'nin title döndürmesini sağladı.Kendisine teşekkür edip masama döndüm.Böylece app bitmişti ve çok mutluydum.

KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:7
YAPILAN İŞ	Uygulama içi güncelleme	Tarih: 18 /07 /2022

Bugün şirkete geldim.Ömer abiye bana yaptığı işlerden bir tane vermesini istedim.Force update ile ilgili projede bir kısım olduğunu söyledi.Bana yapabileceğimi söyledi.Force Update kısaca açıklamak gerekirse,uygulama içerisinde yeni bir güncelleme geldiğinde ekrana güncellenmeniz gerekiyor diye bir uyarı gelir ya ama güncellemeden uygulamaya devam edemeyiz.İşte buna force update Android framework'te Immediate update deniyor.Bir de flexible update kısmı var Bu da kullanıcı isterse uygulamayı güncelleyebilir,isterse güncellemeden devam edebilir.Ömer abi daha önce hiç böyle bir projede çalışmadığını onun için de ilk olacağını söyledi.O yüzden heyecanlıydım biraz.Gidip developer android'den medium'dan makaleler bakmaya başladım.Developer android'de kodlar buldum

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val appUpdateManager = AppUpdateManagerFactory.create(this)
    appUpdateManager.appUpdateInfo.addOnSuccessListener { it: AppUpdateInfo?
        if (it.updateAvailability() == UpdateAvailability.UPDATE_AVAILABLE &&
            it.isUpdateTypeAllowed(IMMEDIATE)
        ) {
            appUpdateManager.startUpdateFlowForResult(
                it,
                IMMEDIATE,
                this,
                1
            )
        }
    }
}
```

Bu kod app başlatıldığında kullanıcıya immediate update yaptırıyor.Ancak iptal edilebildiğinden

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == 1) {
        if (resultCode != RESULT_OK) {
            Log.e(tag: "MY_APP", msg: "Update flow failed! Result code: $resultCode")
            showUpdateFlow()
        }
    }
}
```

Onactivityresultta eğer RESULT_OK dönmüyorsa tekrar update için flow göstermemiz gerekiyor.Bunu halletikten sonra Ömer abiye kodları yolladım ve başka görev istedim.O da bana teşekkür etti ve immediate update'i projeye ekledi.Bana in app review yapacağını söyledi.Bana deneyebileceğimi söyledi.Kendisi daha önce denemişti,ancak değerlendirme ekranı gelmiyordu.In app review ise bir uygulamada bizi değerlendirmek ister misiniz gibi yazılar oluyor ya onlara verilen genel ad.Bir fonksiyon oluşturup review manager oluşturup daha sonra da requestReviewFlow methodunu çağırmaya çalıştım,ancak benim de ekranıma gelmedi.Bu durum garibime gitti çünkü her şeyi dokümantasyona uygun şekilde yaptığımı düşündüm.Biraz daha uğraşacaktım ancak yorulduğumu fark ettim.Biraz ömer abinin ve diğer arkadaşların çalışmalarına bakmaya karar verdim.Ömer abinin kod blokları arasına region init,region variables gibi yazılar koyduğunu gördüm.Sonra bunun kodları daha düzenli birimlere ayırmak için yapıldığını anladım.Ben de kullanmaya karar verdim.Kodumuzu daha güzelleştirmek için neler yapabileceğimizi sordum.Bana her zaman MVVM ile çalışmamı önerdi.Test edilmesi daha kolay olduğu için.Değişken isimlerine özellikle vurgu yaptı.Değişken isimleri kendini açıklayan olmalıdır.Kodu o dili az bilen biri bile anlamalıdır.Ben de bunları yapmaya çalışıyorum projelerimde.Ömer abiye bu değerli tavsiyeleri için teşekkür ettim ve masama geçip eşyalarımı topladım.Ve bir günü daha bu şekilde kapattık.

KONTROL SONUCU 		
KISIM	Kotlin ile Android geliştirme	Yaprak No:8
YAPILAN İŞ	Mvvm Kullanımı	Tarih: 19 /07 /2022
<p>Bugün şirkete geldiğimde In app review ile uğraşacaktım ki ömer abi bana o işi hallettiğini söyledi.Ben de nasıl olduğunu sorduğumda reviewmanager'i fonksiyonun dışından initialize ettiğini ve daha önce çağrıldığı için olduğunu söyledi.Nasıl aklına geldiğini sorduğumda içgüdüsel olarak yaptığını söyledi.Anladım ki tecrübeyle insan birçok şeyi halledebiliyor.Projenin in app review özelliğinin çalışması beni mutlu etti.Geçen gün ömer abinin dediği MVVM meselesine daha çok önem vermem gerektiğini fark ettim ve içinde mvvm olmayan bir adım sayar dersi bulup,buna mvvm eklemem gerektiğini düşündüm ve çalışmaya başladım. Bugün youtube'da gördüğüm bir adım sayar projesini takip etmek istedim.Proje sayesinde sistem servislerine hakimiyetim artacaktı.Bunun için yeni proje açtım ve projeyi izlemeye başladım.Kısaca mantık şu: Sisteme erişip android cihazımızda bulunan adım sayma donanımını kullanıyoruz.Projeyi kısa sürede izleyip ben de yaptım.Ama proje sonunda öğrendim ki gerçek adım sayarlar daha net ölçüm yapmak için iki yer arası mesafeyi alıp ona göre adımı belirliyormuş.Ben de android cihazımda test ettiğimde telefonu kendi adım sayarıyla bu işin gerçekten çok da verimli olmadığını gördüm.</p> <p>Daha sonra bu projeye MVVM mimarisini eklemem gerektiğini düşündüm.Çünkü mvvm sayesinde uygulama hem daha organize oluyor hem de konfigürasyon değişiklikleri(telefonu yan çevirme,düşük bellek gibi)şeylerden etkilenmiyor.Kısaca açıklamak gerekirse mesela telefonu yan yatırdığımızda uygulama önce kapatılıp sonra tekrar açılıyor lifecycle methodları onpause,onstop,oncreate,onresume sırayla çağrılıyor.Bundan dolayı da verileri livedataya yazmazsak kaybolup uygulama resetleniyor.</p> <p>Ama viewmodelda sadece lifceycle olarak oncleared methodu var ve uygulama kapatıldığında çalışıyor ve konfigürasyon değişikliklerinden etkilenmiyor.</p> <p>Ve ben bu faydasını çok kullanmayacak olsam da view model test etmeyi daha da kolaylaştırıyor.Ben de android dokümantasyonuna girip mvvm kullanımına baktım. Verileri live data adlı değişkende tutmamın işimi göreceğini düşündüm,flow daha çok statik olmayan veriler için kullanılıyor.Ben zaten her adım atıldığında main activityde veriyi observe edeceğim için anlık olarak takip edebilecektim.observe methodu bir lifecycle owner istiyor.Burda aktiviteyi verebiliyoruz.Bu projeye de regionlar ekledim kodu daha düzenli tutmak ve karışıklığı azaltmak için kullanıyoruz.Ve viewmodel classımı yazmaya başladım.Elimde 3 veri var bu proje için toplam adım önceki toplam adım ve şuanki adım sayısı,proje bu şekilde uygulanmıştı ama mvvm eklenmemişti ben de kendim ekledim ve mvvm mimarisi için gereken kodu aşağıya koydum.Main activity'de observe ettikten sonra ömer abiye gidip viewmodel'ı uygun şekilde implemente edip etmediğimi sordum.İnceleyip onayladı ve tebrik etti.Bunu ileriki projelerimde de kullanmayı düşünüyorum.</p> <pre> class StepsViewModel : ViewModel() { var totalSteps : MutableLiveData<Float> = MutableLiveData() fun setTotalSteps(liveDataValue: Float) { totalSteps.value = liveDataValue } val previousTotalSteps: MutableLiveData<Float> = MutableLiveData() fun setPreviousTotalSteps(liveDataValue: Float) { previousTotalSteps.value = liveDataValue } val currentSteps: MutableLiveData<Float> = MutableLiveData() fun setCurrentSteps(liveDataValue: Float) { currentSteps.value = liveDataValue } } </pre>		

KONTROL SONUCU 		
KISIM	Kotlin ile Android geliştirme	Yaprak No:9
YAPILAN İŞ	Exoplayer ile video oynatma	Tarih: 20/07 /2022
<p>Bugün şirkete geldiğimde yeni bir proje yapmaya başlayacaktım.Şirkette yapılan projeleri incelediğimde video oynatılan bir kısım gördüm.Ben de kendim bir tane video oynatıcı yapmaya karar verdim.Kütüphanelerden exoplayer'in bu iş için en uygunu olduğunu öğrendim.Codelabs'a girip adımları izlemeye başladım.Codelabs kısaca kod üzerinde kısa bir proje yaptırılan dokümantasyon gibi düşünülebilir.Baktıktan sonra adımları izlemeye başladım.Dokümantasyon kolay olduğu için rahatça anladım.</p> <pre><VideoView android:id="@+id/videoView" android:layout_width="288dp" android:layout_height="396dp" android:layout_marginStart="68dp" android:layout_marginTop="108dp" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" /></pre> <p>Yukarıda görüldüğü gibi xmlde videoview oluşturuyoruz.Sonra resoruce altında raw klasörü oluşturuyoruz.İçine de oynatmak istediğimiz videoyu atıyoruz.Ben de tamino-indigo night şarkısını attım.sonra koddaki viewbindingi aktif ettikten sonra aşağıdaki kodlarla videoyu çalıştırıyoruz.</p> <pre>val video = Uri.parse("android.resource://" + packageName + "/" + R.raw.indigonight) binding.videoView.setVideoURI(video) binding.videoView.start()</pre> <p>Projeyi çalıştırdığımda videonun çalıştığını ancak ses gelmediğini fark ettim.Sesi farklı bir şekilde aktif etmemiz gerektiğini düşündüm biraz araştırdım ancak bir şey bulamadım.Sonrasında acaba emülatörde bir sorun olabileceğini düşündüm.Gidip ses ayarlarına girdim ve sesi yüklettim bundan sonra ses gelmeye başladı.Bu da beni mutlu etti.Exoplayer gelecekte de Android alanında çok kullanılacak gibi gözüküyor.Çünkü Google tarafından çıkarıldı ve yeri sağlam gözüküyor,dokümantasyonda,Android sdk'de olan media player api'den daha fazla özelliği olduğu yazıyor.Dokümantasyonu okumaya devam ettiğimde ilginç bir şey dikkatimi çekti.Biz mesela spotifyde müzik dinliyoruz,daha sonra youtubeda video açtığımızda spotifydaki müziğin durma sebebi Spotify uygulamasının videoyu durdurmasıymış.Önceden Android sistemi kendisi bu işlemi hallediyor diye düşünürdüm.O yüzden müzik veya video uygulaması yaparsak arka planda başka video veya müzik başlatılıp başlatılmadığını kontrol etmemiz gerekiyor.Bu projeyi şirkette diğer çalışanlara gösterdiğimde güzel olduğunu söylediler.Belki bu konu hakkında ileride makale yazabilirim.</p> <p>Sonra boş durmadan yeni bir işe geçiş yapmak istedim.Mediumda gezerken timber adlı bir şey gördüm.Bunun bir kütüphane olduğunu anladım.Githubda projenin linkine girince log işlemi için kullanılacağını öğrendim.Merak ettiğim şey ise şuydu:Zaten Android sdk ile gelen bir log aracımız var neden başka bir şekilde log yapalım.Oturup makaleler okumaya başladım.Bize kattığı 2 büyük artı var.Birincisi normal log işleminde tag eklememiz gerekiyor.Hatta genelde bu bulunduğumuz aktivitenin ismi olur.İkincisi ise uygulama release edilirken yani play store'da yayınlanırken log kodlarını silmemiz gerekiyor.Bu iki şeyi önlemek için Timber hayat kurtarıyor.Öncelikle implementasyonları yaptıktan sonra yapmamız gereken şey bir application sınıfı oluşturup Timberi aktif hale getirmek.Bunun sayesinde timber'ı tüm uygulamamızda kullanır hale gelebiliyoruz.Daha sonra ise Timber.d("Logging") Yazıp debug'da Logging yazısını görebiliriz.Diğer log modları ise verbose,info,warn,error ve</p>		

assert.Bu modlardan ihtiyacımıza uygun olanı alıp kullanabiliriz.Kalan vaktimde ise oturup timber ile ilgili makale yazdım linki de burada.Makale yazmak öğrenmeye çok faydalı.
<https://medium.com/@muhammetemingundogar53/timber-nedir-b4adef8b92c3>

KONTROL SONUCU



KISIM	Kotlin ile Android geliştirme	Yaprak No:10
YAPILAN İŞ	Recycler view kullanımı	Tarih: 21 /07 / 2022

Bugün şirkete geldiğimde developer arkadaştan yeni bir proje almaya karar verdim.O da bana şuanda üzerinde çalıştıkları bir projeyi gösterdi.Ekranda bir adet şirket çalışanlarının olduğu ekran vardı.Bu ekrandan başka ekranlar da açılacaktı.Bottom sheet gibi,sonra özelleştirilmiş alert dialog gibi...Kullanmam gereken renkleri ondan aldıktan sonra projeyi kendim yapmaya başladım.Bu projede çok şey öğrenebileceğimi fark ettim.Ve projeye git'i entegre ettikten sonra layoutları yapmaya başladım.Ana ekran tasarımımda textview,rounded imageview,imageview,view ve recycler view olacaktı.Bunları yapmaya başladım.Ancak istediğim tasarım tam olarak ortaya çıkmadı özellikle büyük ekranda.Developer arkadaştan destek aldım o da bana constraintleri parent yani en dıştaki layouta değil de diğer layoutlara verirsem daha responsive bir tasarım olacağını söyledi.Onun dediğini yaptım.Ekranın en sağındaki tasarımın start constraintini parenta verdim.En sağdaki tasarımın end constraintini de parenta verdim.Daha sonra diğer tasarımları sırayla birbirine constarint verince birisi aşağıya kayınca diğeri de otomatik olarak onunla birlikte aşağıya kaymaya başladı büyük ekranda.Böylece responsive tasarım elde ettim.Textviewların birinde farklı bir syntax denedim.Daha önceleri resource dosyasından R.string.name diyerek dosya alabiliyordum ancak bu string dosyasının içerisinde olan string statik bir string oluyordu.Bunu dinamik yapmanın yollarını araştırdım ve sounda buldum.

```
<string name="member">%1$d Member</string>
```

Yukarıdaki gibi yapınca daha sonra da kotlin dosyamızda aşağıdaki gibi çağırınca 14 member yazısı gelecek.

```
val resources = resources
val text = resources.getString(R.string.member, ...formatArgs: 14)
binding.textViewMembers.text = text
```

Proje biraz gözümü korkutmaya başlamıştı.Çünkü şimdi bildiğim şeylerden çok bilmediğim şeylere geçiş yapacaktım.Bottom sheet,custom alert dialog gibi,bunlar şimdilik gözümü korkutuyordu ancak bir süre uğraşınca bunların üstesinden gelebileceğimi düşündüm.Projede diğer önemli bir kısım olan recycler view yapısına geçiş yaptım.Recycler view'da bir adet rounded image view 3 tane text view bir tane de bottom sheet için sağa doğru ok olacak.Textviewların bir tanesi bazen gözüküp bazen gözükmeyecek diye birisi invisible olarak set edildi.Default tasarım aşağıdaki gibi olacak.



İsim

Bilgisayar Mühendisi

Company Manager



Company manager yazısı ise bazı durumlarda görünür,bazı durumlarda invisible olacak.Bu tasarımı beğendim ve recycler view için adapter yazmaya başlayacaktım ki yorulduğumu ve saatin 6 ya yaklaştığını fark ettim.Şirketteki diğer arkadaşlardan kendimi geliştirmek için tavsiyeler almaya gittim.Aynı zamanda onları izlemek bana çok zevk veriyor.Umarım ben de ileride onlar gibi böyle büyük projelerde yer alabilirim.Ömer abiye timber ile ilgili makale yazdığımı söyledim ve beni tebrik etti.Böyle şeyleri yapmamın çok faydalı olduğunu söyledi.Başarılı bir Android developer olmak için çok zaman harcamak gerektiğini söyledi.Kendisi Türk olmadığı için genelde kendisiyle ingilizce anlaşıyoruz.Bu da yabancı dil açısından iyi bir pratik oluyor.Çünkü ileride yurtdışında çalışırsam orda da ekipteki arkadaşlarımla ingilizce konuşmak zorunda olacağım.Mesai saati bitti ve eve gittim.

KONTROL SONUCU 		
KISIM	Kotlin ile Android geliştirme	Yaprak No:11
YAPILAN İŞ	Bottom sheet kullanımı	Tarih: 22/07 /2022

Bugün şirkete geldiğimde üzerinde çalıştığım projeye devam ettim.En son recycler view için item layout oluşturmıştım ancak adapter yazmamıştım.Adapteri açıp çalışmaya başladım.Adapter recycler view için her bir itemda olacak davranışları belirlediğimiz yer diyebiliriz.Daha önce de bahsettiğim gibi oncreateViewholder,onbindviewholder,getitemcount methodlarını override ettim.Bunlardan sonra aynı zamanda bottom sheet açılması için bir fonksiyon yazdım.

```
private fun onItemClick(item: DialogModel, position: Int,context: Context) {
    val bottomSheet = BottomSheet(context = context)

    bottomSheet.showBottomDialog(item)
}
```

Itemview.setonclick listenerda bu fonksiyonu çağırdım.

```
holder.itemView.setOnClickListener { it: View?
    onItemClick(item,position,context)
}
```

Verdiğimiz item ise o sırada tıklanan item bize onbindviewholder methodunda veriliyor ve onu kullanıyoruz.

```
val item = membersList[position]
```

Bottom sheet kullanımını da araştırma sonucu öğrendim.Bir class açmamız gerekiyor ve bottom sheet açıldığında nasıl davranacağımızı yazıyoruz.

```
class BottomSheet(val context: Context) {
    private lateinit var binding: BottomSheetDialogBinding
    fun showBottomDialog(item: DialogModel) {
        binding = BottomSheetDialogBinding.inflate(LayoutInflater.from(context))
        binding.textViewName.text = item.name
        binding.textViewJob.text = item.job

        item.resId?.let { binding.imageView.setImageResource(it) }
        val bottomSheetDialog = BottomSheetDialog(context)
        bottomSheetDialog.setContentView(binding.root)
        bottomSheetDialog.show()

        binding.closeButton.setOnClickListener { it: View?
            bottomSheetDialog.dismiss()
        }
    }
}
```

Bottom sheet recycler viewda her sağ oka tıklandığında o kullanıcıya ait bottom sheet aşağıdan çıkacak.Bence bu gayet güzel bir yöntem.Ui/UX açısından da güzel duruyor.Daha sonra bu projeyi daha güzel hale getirmek için <https://fonts.google.com/> sitesinden Google tarafından desteklenen fontları indirmek istedim.Anton ve squarepeg adlı iki tane güzel font bulup projeye ekledim.Resource klasörü altında font adlı bir klasör açıp onun içine attım ve xml'de fontFamily attribute olarak bu ikisini iki yerde kullandım.Default olarak gelen yazı tipinden çok daha güzel gözüktüğünü söylemeliyim.Yavaş yavaş projede sona yaklaşırken menü ekleme adımına geçiyordum.Normal menü çok sefer kullanmıştım ancak bu sefer pop up menü kullanacaktım.Pop up menü ise default olarak gelen menüden farklı olarak benim projemde imageview'a tıklandığında açılacak.Araştırmayla işe başladım ve normal menu kullanımından biraz farklı olduğunu gördüm.Normal menüyü projeye eklerken menu xml oluşturup daha sonra onCreateoptionsmenu ve onOptionsItemSelected methodlarını override edip kullanıyoruz ancak

Pop up menude deęişen bir şeyler olduğunu gördüm.Bugün de mesaiyi bitirdik ve eve gittim.

KONTROL SONUCU



KISIM	Kotlin ile Android geliştirme	Yaprak No:12
YAPILAN İŞ	Popup menu kullanımı	Tarih: 25/07 /2022

Bugün şirkete geldim ve projeye kaldığım yerden devam ettim.Pop up menuyu araştırdıktan sonra yazmaya başladım.Öncelikle menu xml oluşturdum.Menu xmlde her bir menu elemanı için bir adet item tagi açıyoruz.Bu tag içinde title,icon,id gibi şeyleri yazıyoruz.

```
<item
    android:id="@+id/define"
    android:icon="@drawable/arrow_right"
    android:title="Define New Manager" />
```

Bunun üzerine main activity geçip kodlarımı yazmaya başladım.Pop up menuyu ayrı bir methodda onclicklerini farklı bir methodda verecektim.

```
popupMenu = PopupMenu(this, binding.imageViewMenu)
popupMenu.inflate(R.menu.menu)
```

popup menuyu bu şekilde oluşturduk.Daha sonra da aşağıdaki kodu try catch bloğuna aldım.

```
val popup = PopupMenu::class.java.getDeclaredField("mPopup")
popup.isAccessible = true
val menu = popup.get(popupMenu)
menu.javaClass.getDeclaredMethod("setForceShowIcon", Boolean::class.java)
    .invoke(menu, true)
```

Yukarıdaki kod ile pop up menuyu kullanılabilir hale getirdik.Imageview tıklandığında sonuçta ekran görüntüsü şeklindeki gibi oldu.

```
varare
> Define New Manager
> Edit
> delete
> report
```

Bu tasarımı beğendim ve define new manager yazısına tıklandığında custom alert dialog oluşturacaktım.Bunun mantığını da araştırdım ve kısaca bir layoutu inflate edip

```
builder.setView(customLayout)
```

ile view olarak geçmek.

☒


Recycler view'da her element şu şekilde tasarlandı kişinin name,job,rounded image view ve bir adet checbkox.Çıkan alert dialogda define buttonuna tıklandığında checkbox seçili olan kullanıcılar ana ekranda geçen gün bahsettiğim yazı görünür halde olacak.Bunun mantığını şu şekilde kurdum.Data class'ta bir adet checked adlı değişken koydum.Zaten main activity ile alert dialogda aynı itemler olduğu için direk onbindviewholderda pozisyona göre boolean değer ataması yapıp kolayca daha sonra diğer adapterda aynı veriyi kullanabiliriz.Öncelikle listemi main activityde veriyordum.Ama her seferinde objeler baştan oluşturulduğundan bu yöntem çalışmıyordu.Bunun yerine bir obje açıp orada tüm işlemleri yaptığımda ise sorunsuz çalıştı.Bu işlemten sonra son adımda önceden seçili olan itemler menu açıldığında tekrardan seçili olması gerekiyordu.Bunu yapmaya uğraştım.Bu diğerine göre çok daha kolay oldu.

```
holder.binding.checkbox.isChecked = position.checked
```

Proje resmen bitmişti.Ancak birkaç test yapmam gerekiyordu.Bir hatamı buldum verilen position parametresini 1 den başlıyor zannettim ve bu yüzden son itemde uygulama

çöküyordu.Bunun 0 dan başladığını anlayınca o hatayı da çözdüm ve proje bitti.

KONTROL SONUCU

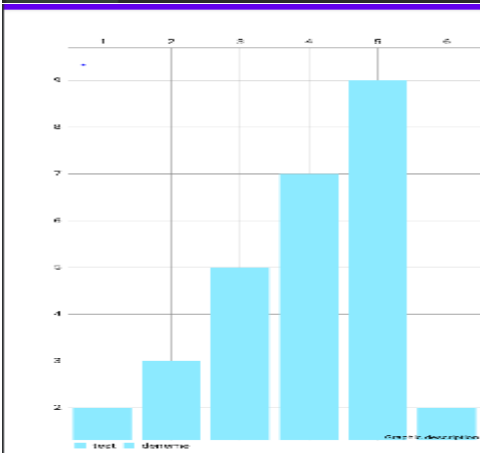
KISIM	Kotlin ile Android geliştirme	Yaprak No:13
YAPILAN İŞ	Grafik Kullanımı	Tarih: 26/07 /2022

Bugün şirkete geldim.Developer arkadaş bana grafikler konusunu öğrenmemi önerdi.Bunlar için [MpAndroidChart](#) adlı bir kütüphaneyi öğrenecektim.Bu kütüphane sayesinde Linechart yani çizgili grafik,barchart yani sütun grafiği,piechart yani dairesel grafiklerini öğrenecektim.Aslında bu kütüphane ile çok daha fazla grafik çeşidi çizabiliyoruz.Ancak ben en çok kullanılanları öğrenmek istedim.Dokümantasyonu gayet anlaşılır olduğundan direkt açıp okumaya başladım.Bir yandan da Android studioda proje açtım.İlk adım olarak view binding aktif ettim ve çalışmaya başladım.Kısaca mantık şu bar entryler yani tabloadaki veriler için liste oluşturuyoruz.

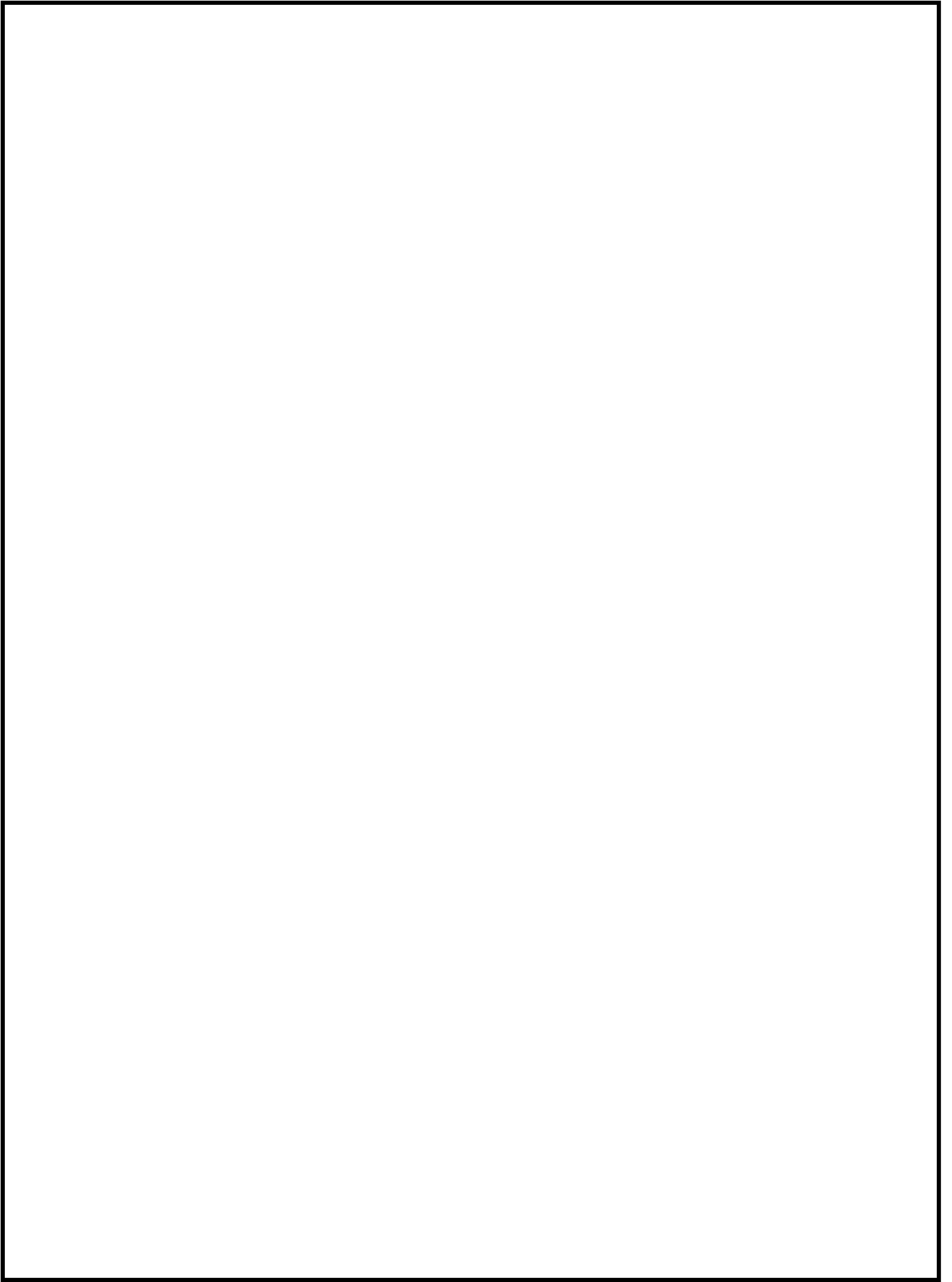
```
fun getBarEntries() {  
    entryList.add(BarEntry( x: 1f, y: 2f))  
    entryList.add(BarEntry( x: 2f, y: 3f))  
    entryList.add(BarEntry( x: 3f, y: 5f))  
    entryList.add(BarEntry( x: 4f, y: 7f))  
    entryList2.add(BarEntry( x: 5f, y: 9f))  
    entryList2.add(BarEntry( x: 6f, y: 2f))  
}
```

Bu listeleri oluşturduktan sonra,bu değerlerin ve label olan bir data set oluşturuyoruz.Daha sonra da Bar data objesi oluşturup içine argüman olarak elimizdeki setleri veriyoruz.Bu argüman var arg olarak tanımlanmış.Bunun amacı ise istediğimiz kadar set verebilmek.Var arg gelecek argümandan kaç tane olacağını bilmediğimiz zamanlarda kullanılıyor. Daha sonra bu grafiğin görünümünü bulup ona setdata olarak hazırladığımız bar data objesini vermemiz gerekiyor.

```
binding.barChart.data = barData
```



Daha sonra elimde böyle bir grafik oluştu.Bu grafiğin başka kullanımlarına da baktım ve gerçekten çok iyi bir kütüphane olduğuna kanaat getirdim.Çünkü içindeki grafik türleri oldukça fazlaydı.Githubda 35 bin yıldız almasının sebebi de bu olmalı.Bu grafiği ileride de kullanabileceğime kanaat getirdim ve saat 6 ya yaklaşırken eşyalarımı toplayıp işten çıktım



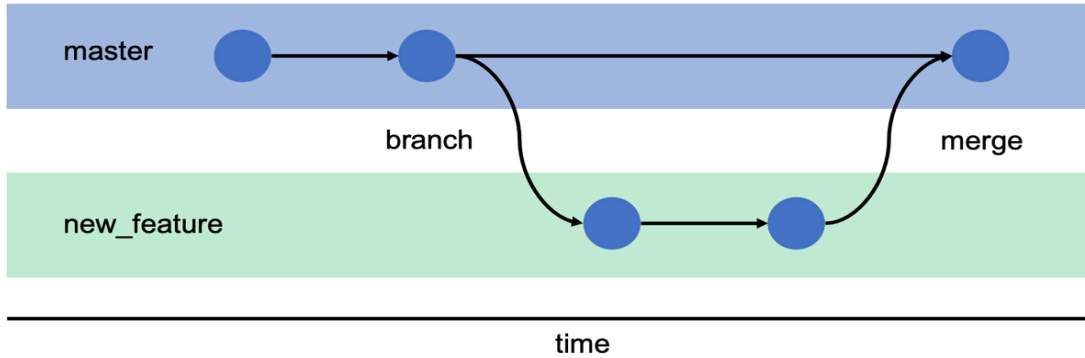
KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No: 14
YAPILAN İŞ	Git Branch Kullanımı	Tarih: 27 /07 /2022

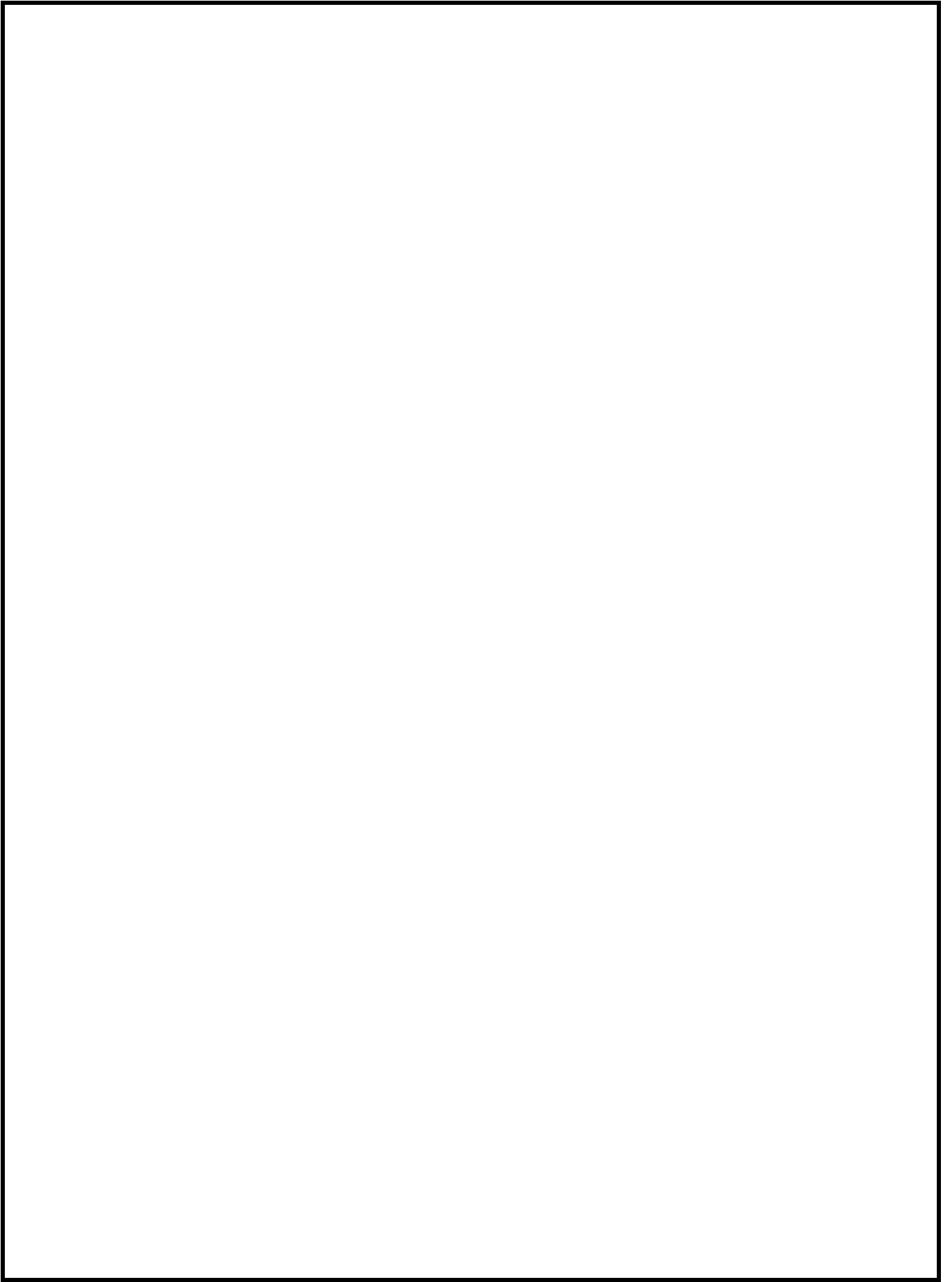
Bugün işe geldim ve diğer stajyer arkadaşla ortak bir mesaj uygulaması geliştirme kararı aldık.Bunu ekiptekilere söyledik ve bizi tebrik ettiler.Gerekli desteği vereceklerini söylediler.Biz de öncelikle bunu profesyonel bir iş gibi düşünüp öncelikle kağıt üzerinde tasarlamamız gerektiğini düşündük.Yani kısaca bir gereksinim analizi yapıyorduk.Gereksinim analizi olarak maddelerimiz şöyleydi:

- 1-Kullanıcı kayıt ekranı olacak ve görsel opsiyonel olarak seçilip kayıt olunacak.
- 2-O kullanıcıya ait mesajların gözüktüğü bir arayüz olacak.
- 3-Kullanıcıların mesajlaştığı bir arayüz olacak.

Şimdilik temelde bu gereksinimleri belirledik.Çok büyük bir proje olmadığı ve işler birbirinden bağımsız olduğu için gereksinimlere ileride ekleme yapabiliriz diye düşündük.Bu projenin bize en büyük katkısı iki kişilik ekip halinde bir proje geliştirme deneyimi olacaktı.Böylece aslında git kullanımına daha da hakim olacaktık.Çünkü artık tek bir branch commit-push şeklinde değil de ayrı branchler üzerinden çalışacaktık.



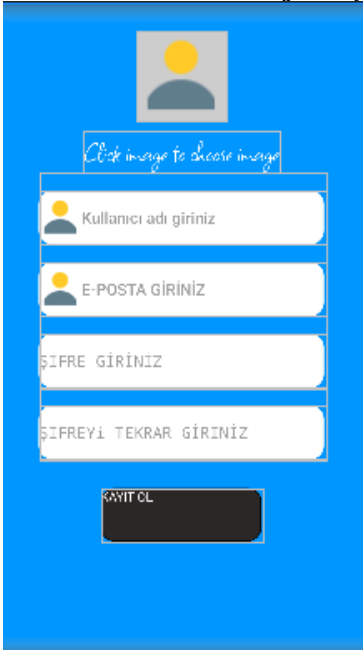
Yukarıdaki görseldeki gibi branch projenin dallara ayrılması ve bunun sonucunda herkesin ilerlemesini diğerinden bağımsız şekilde yapabilmesine olanak sağlıyor.Daha sonra da merge işlemi ile iki branch birleştiriliyor.Ekipteki diğer arkadaşlardan branch sistemi ile ilgili de bilgi aldıktan sonra kendimiz de üzerinde araştırmalar yaptık.Daha sonra kullanıcıların mesajlarının ve bilgilerinin tutulduğu online bir veritabanı gerekiyordu.Bunun için firebase adlı eklentiye kullanacaktık.Firebase Google tarafından sonradan satın alınmış bir eklenti.Bu eklentiyle kullanıcı kaydı,giriş işlemi,dosya depolama,real time veri depolama gibi işlemler yapılabilmektedir.Şirkettekilere firebase kullanıp kullanmadıklarını sorduğumuzda uygulama analizi için firebase kullandıklarını öğrendik.Daha sonrasında firebase ile ilgili mediumda makaleler okuduk.Aslında back end bilen birinin kullanmaması gereken bir şey olduğunu öğrendik.Çünkü normal bir veritabanına göre eksileri varmış.Ama biz back end bilmediğimiz için mecburen firebase'e muhtaç kaldık.Kafamızda son olarak projede gerekli şeyleri konuştuktan sonra internetten ui design örnekleri inceledik.Özellikle login ekranı için çok güzel tasarımlar gördük.Biz de elimizden geldiğince bu tasarımlarla alakalı şeyler ortaya çıkarmaya karar verdik.Saat 6 ya geldiğinde projenin taslağını oluşturmuştuk ve güzelce planladığımız için bu bizi mutlu etmişti.Umarım her şey planladığımız gibi iyi gider.



KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:15
YAPILAN İŞ	Kayıt Ekran Tasarımı	Tarih: 28/07 /2022

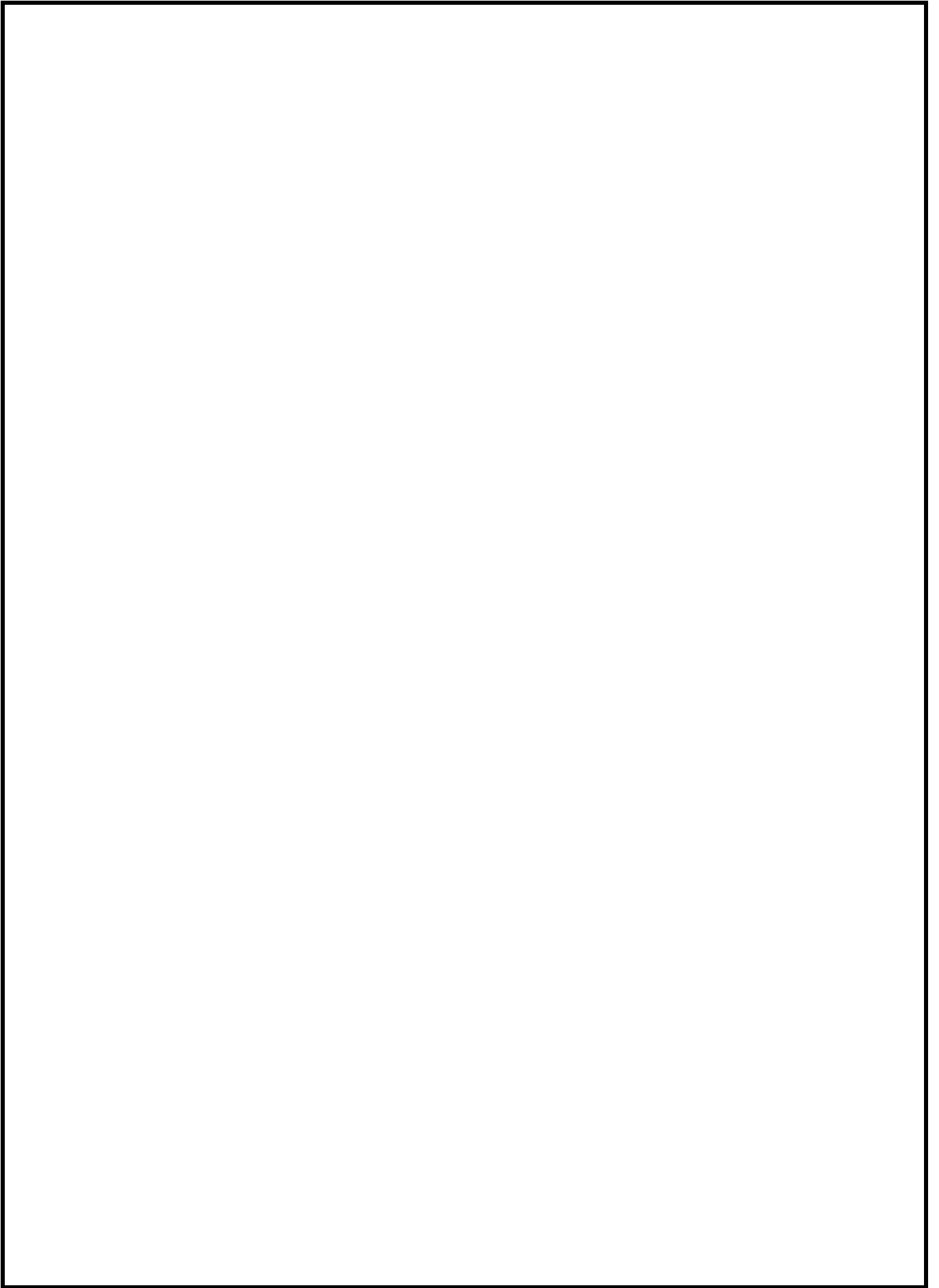
Bugün işe geldik ve arkadaşım ile projeye başlama kararı aldık. Planlamaya koca bir günü ayırmıştık. Umarız her şey beklediğimiz gibi gider. Öncelikle ben boş bir Android projesi açıp bu projeyi de githuba yükledim. Arkadaşıma da yetki verdim. Ona da bir branch açtık ve iki ayrı branch üzerinden çalışmalarımızı ilerletecektik. İlk iş bölümü olarak ben register ekranını o da login ekranını yapacaktı. İnternette güzel bir renk buldum arka planı o renk yaptım. Sonra internette kullanıcı profil resmini seçme ekranı için bir icon bulduk. Onu kullanmaya karar verdik. Sonrasında bir de uygulamaya font ekleme kararı aldık. Bir adet font bulup indirdim ve projeye ekledim. Kullanıcı adı ile e posta kısmına da icon ekledim. Gözüme hoş gelen bir tasarım oldu. Özellikle bu staja başladığımda tasarım anlamında kötü olduğumu düşünürsek.



Tasarım genel olarak hazırdu ancak kullanıcı adı kısmına uzun bir kullanıcı adı girdiğinde kullanıcı adı genel olarak aşağıya kayıyordu. Bunu bir türlü çözemedim. Ömer abiye gidip editText'in büyümesini istemiyorum sadece yana kaymasını istiyorum dedim. Biraz uğraştıktan sonra editText'e wrap content yerine 100 dp yaptığımızda bu sorunun ortadan kalktığını gördüm. Genel anlamda tatmin edici bir sonuçtu. Sonrasında Firebase kayıt kodlarını yazmam gerekiyordu. Ama daha firebase console'da proje oluşturmamıştım. Firebase implementasyonlarını yaparken hata aldım. Çok uğraştım ancak bir türlü çözemedim. İnternette de arattım ancak bir çözüm bulamıyordum. Daha sonra stackoverflowda az kabul edilen bir cevabı denedim. Gradle.properties dosyasında

```
android.enableJetifier=true
```

satırını ekleyip sync now'a tıkladığımda hata almamıştım. Bu hata bugün bayağı vaktimi çalmıştı. Ama hatanın sebebi firebase dokümantasyonunun güncel olmamasıydı. Hatanın sebebinin ben olmaması biraz mutlu etse de çok uğraştırması canımı sıktı. Ama bizim mesleğimiz de hatalarla boğuşmak zaten. Bu hatadan dolayı bugün kodlama kısmına geçemedim ve gün bitti. Yarın firebase authentication kısmına bakacaktık. Bugün de böylece bitti.



KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:16
YAPILAN İŞ	Firestore Kullanımı	Tarih: 29 /07 /2022

Bugün işe geldiğimde firebase authentication kısmını yapacaktım.Bu iş biraz da olsa beni zorlayacaktı.Çünkü daha önce hiç uğraşmamıştım.

Firestore Authentication



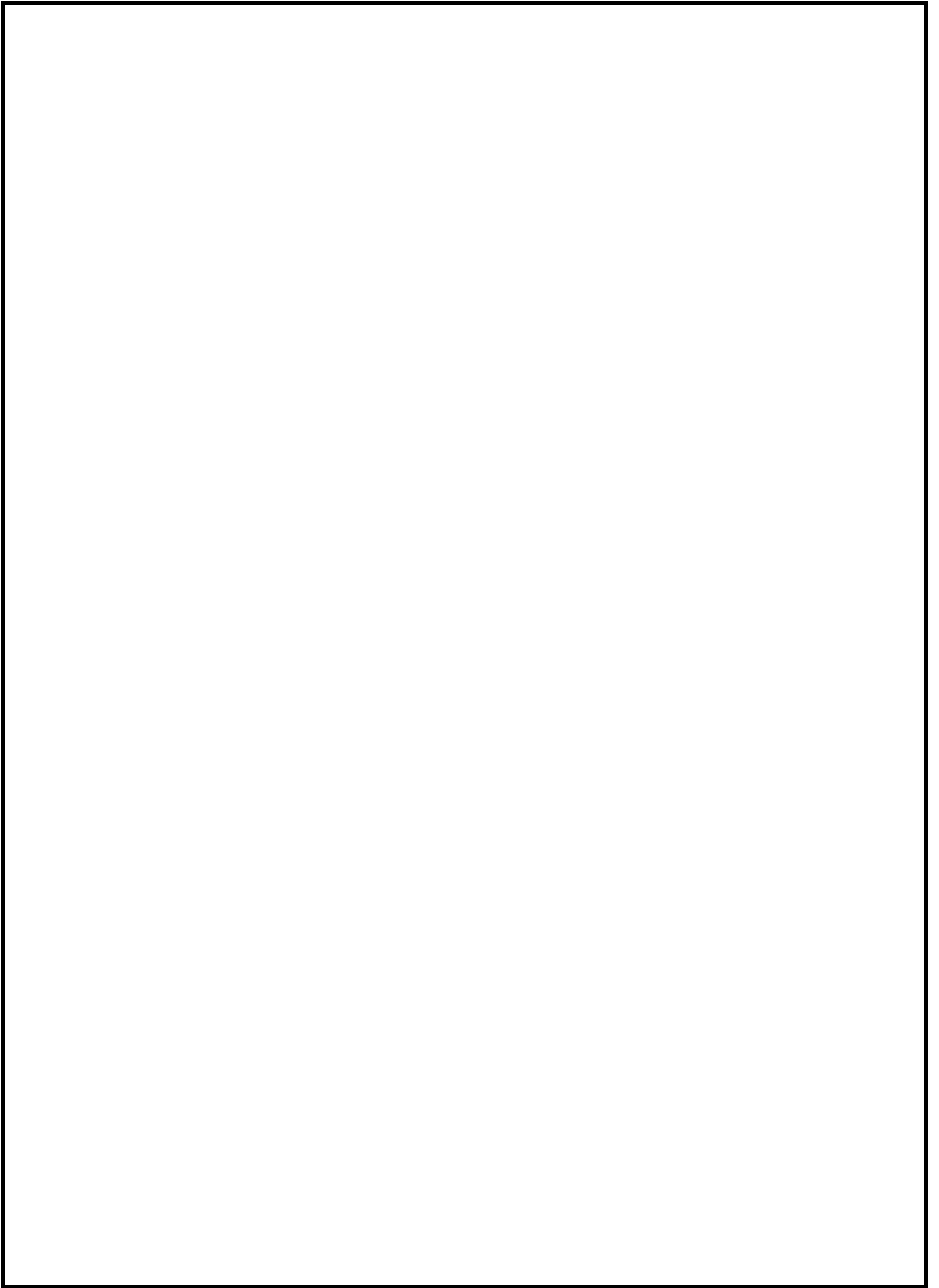
Şansıma dokümantasyon anlaşılırdı.Ve ek bir kaynağa ihtiyaç duymadan bu işlemi halledebileceğimi düşündüm.Firebase auth objesi oluşturup bunun üzerinden çalışacaktım.Firebase auth class'ı singleton bir class olduğunu gördüm.Singleton tüm projede sadece tek bir objesi oluşturulan ve her yerde aynı objesi kullanılan class demek.

```
auth = FirebaseAuth.getInstance()
```

Fragmenta özel binding kodunu da yazdım.Ve kullanıcıyı sisteme kayıt etme kodlarını yazdım.Kayıt olurken email ve parola ile kayıt özelliğini kullanacaktım.Aslında başka kayıt yöntemleri de varmış firebase içerisinde.Github ile kayıt ol gibi.Ancak bu projede bu kadar komplike bir şey yapmaya gerek olmadığını düşündük.

```
auth.createUserWithEmailAndPassword(e_posta, sifree)
    .addOnCompleteListener { it: Task<AuthResult!>
        if (it.isSuccessful) {
            val uid = auth.currentUser!!.uid
            registerHashMap.put("UserID", uid)
            val update = UserProfileChangeRequest.Builder()
                .setDisplayName(kullaniciAdi)
                .build()
            auth.currentUser?.updateProfile(update)
            firestore.collection(collectionPath: "USERS").add(registerHashMap).addOnCompleteListener { it: Task<DocumentReference!>
                if (it.isSuccessful) {
                    Log.e(tag: "Data", msg: "Data Başarıyla Eklendi")
                }
            }
            val intent = Intent(activity, MainActivity::class.java)
            startActivity(intent)
            showSuccesfullToast(title: "Başarılı", kullaniciAdi, requireContext())
        }
    }
```

Kullanıcının,kullanıcı adı ana ekranda gözükeceği için display name olarak onu da kaydettim. Daha sonrasında bu kayıt bloğuna girmesi için kullanıcı adı ve şifrenin boş olup olmaması kontrolünü de yaptım.Bunun için de isEmpty methodunu kullandım.Bundan sonra telefonumda uygulamayı test ettim.Firebase console'dan baktığımda uygulamanın sorunsuz kayıt işlemi yaptığını gördüm,arkadaşım da bu sırada firebase login işlemini halletmişti,ve böylece bir günün daha sonuna geldik.



KONTROL SONUCU

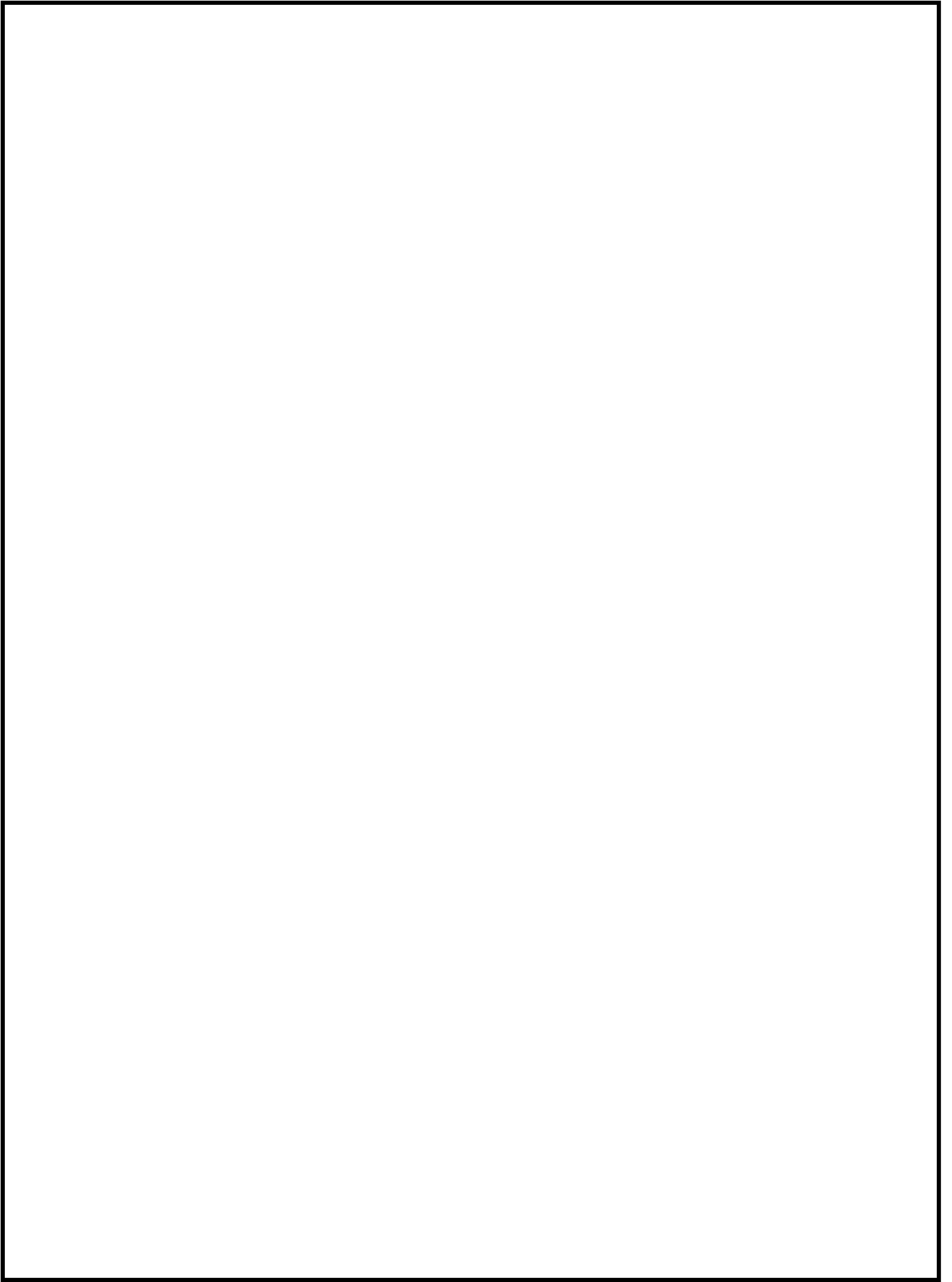


KISIM	Kotlin ile Android geliştirme	Yaprak No:17
YAPILAN İŞ	Uygulama İzinleri	Tarih: 01/08 /2022

Bugün işe geldim ve kayıt olma ekranındaki son aşama olan kullanıcının firebase firestore yani real time database kaydedilmesi ve firebase storage kısmına bakacaktım.Real time database kısaca sunucuda bir güncelleme olduğunda kullanıcıların direkt ekranına düşmesini sağlayan bir yapıdır.Firebase storage kısmına da kullanıcı görsel eklerse onu eklemeye default resim kaydedilecek.Görsel ekleme için izin almamız gerekiyor.Kullanıcı izin verirse galeriye erişebileceğiz.

```
binding.imageViewUser.setOnClickListener { it: View!
    if (ContextCompat.checkSelfPermission(
        requireContext(),
        Manifest.permission.READ_EXTERNAL_STORAGE
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        ActivityCompat.requestPermissions(
            requireActivity(),
            arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE),
            requestCode: 1
        )
    } else {//izin verilmiş galeriye git
        val galeriIntent =
            Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI)
        startActivityForResult(galeriIntent, requestCode: 2)
    }
}
```

Bu kod ile izin verilmişse galeriye gidiyoruz.Ancak kullanıcı izin vermemişse izin istiyoruz.Daha sonra fotoğrafı storage kısmına ordaki url'i de firebase firestore kısmına kaydediyoruz.Çünkü ilerde glide kütüphanesiyle görseli indirmemiz gerekecek.Ancak tam projeyi githubda merge yapmaya çalıştık ancak kodun bir kısmının kaybolduğunu gördük.Bu bizi çok strese soktu.Anladığımız kadarıyla birimiz son değişikliklerini pushlamamıştı.Bundan dolayı proje bir süre yavaşlayacaktı.Commitleri geri alıp tekrar birleştirmeyi düşündük ancak işleri daha da kötü yapabileceğimizi düşünüp bu karardan vazgeçtik.Kaybettiğimiz kısımları tekrar yazmaya başladık.Bazı görseller de silinmişti onları da internetten indirdik.Bundan sonra ayrı branchlerde çalışmanın bizim için zor olduğunu düşünüp tek bir branch üzerinden sırasıyla commit-pull-push yaparak çalışmaya karar verdik.Herkes sırasıyla işlemleri doğru yaparsa bir sorun çıkmayacaktı.Böylece tekrardan çalışmaya başlayacaktık ki yine bir sorun çıktı.Arkadaşım bu sefer yeni repository oluşturdu ve ben projeyi klonlayıp çalışmaya devam etmek istedim.Ancak projeye pushlayamıyordum.İnternette uzun süre hatayı arattım.Bi sitede ssh key oluşturmamı önerdiler.Onu yapmayı öğrendim.Sonra git bash açıp bazı komutlar yazdım ama çalışmadı hiçbir şey.ssh key bir tür özel şifre gibi düşünülebilir.Bu biraz moralimi bozmuştu.Saatlerimi aldı ama daha sonra hesabıma girdiğimde arkadaşımın attığı invite kabul etmediğimi gördüm.Bu kadar basit bir hatanın beni ne kadar uğraştırdığını görünce sinirlendim.Ama sonra sakinleştim.Çünkü yazılımda böyle şeyler gayet doğal ve böyle şeylere alışmamız gerekiyor.Saat zaten 5'i bulmuştu.Bu hatayla uğraşmaktan pek bir şey bakamadım ve günü verimsiz bir şekilde noktaladım.



KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:18
YAPILAN İŞ	Glide İle Görsel İndirme	Tarih: 02/08 /2022

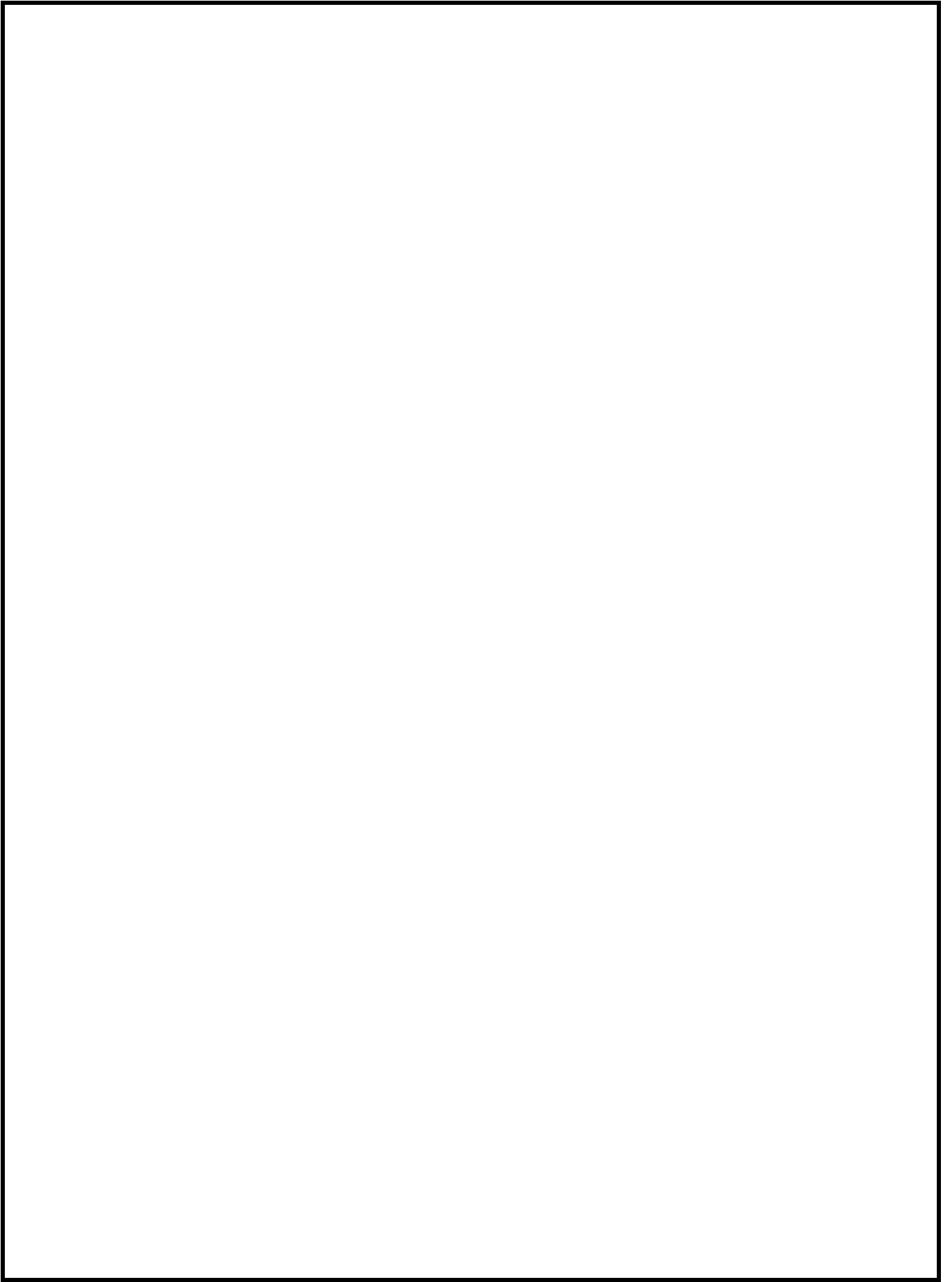
Bugün işe geldim ve dünü unutup verimli bir gün geçirmeyi umarak işe başladım.User recycler view kısmında,diğer kullanıcıların görseli ve kullanıcı adı gibi bilgileri gösterilecek.

```
override fun onBindViewHolder(holder: UsersViewHolder, position: Int) {  
    holder.binding.itemTextViewUserName.text = userList[position].userName  
    holder.binding.itemTextViewState.text = userList[position].userState  
    Glide.with(context).load(userList[position].userImage.toUri()).into(holder.binding.itemCircleImageView)
```

Bunu yapmak için öncelikle bir itemview oluşturmam gerekiyordu.Circle image view kütüphanesini kullandım.Yuvarlak görseller kare görsele göre daha güzel bir görünüm sağlıyor.Altta da bir toast mesajı göstermek için githubdan arkadaşımın bulduğu güzel bir toast kütüphanesi olan Motion Toast kullandık.Recycler view adapter yazdıktan sonra ortaya böyle bir sonuç çıktı.



Bunu da halletikten sonra sırada mesajlaşma algoritması ve aşağıda gözüken home,settings,exit yazan bottom navigation ile fragment değiştirme kalmıştı.Bunu araştırmaya başladım.Ve gayet kolay olduğunu gördüm.Yapmamız gereken şeyin menu.xml dosyasındaki home,settings ve exit fragmentlarındaki id ile navigation.xml dosyasındaki idlerle aynı olması ve sonrasında bottom navigation'a nav controller'ı parametre olarak vermek.Bunun sonrasında geçişler istediğimiz gibi olacak.Bunu tam yapacaktım ki işten çıkış saati geldi bilgisayarımı kapattım ve iş yerinden çıktım.



KONTROL SONUCU



KISIM	Kotlin ile Android geliştirme	Yaprak No:19
YAPILAN İŞ	Bottom Navigation Kullanımı	Tarih: 03 /08 /2022

Bugün işe geldim ve dün kaldığım yerden projeye devam ettim.menu xml dosyasını aşağıdaki gibi yazdım.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/homeFragment"
        android:icon="@drawable/ic_baseline_home_24"
        android:title="Home" />
    <item
        android:id="@+id/settingsFragment"
        android:icon="@drawable/ic_baseline_settings_24"
        android:title="Setting" />
    <item
        android:id="@+id/exitFragment"
        android:icon="@drawable/ic_baseline_exit_to_app_24"
        android:title="Exit" />
</menu>
```

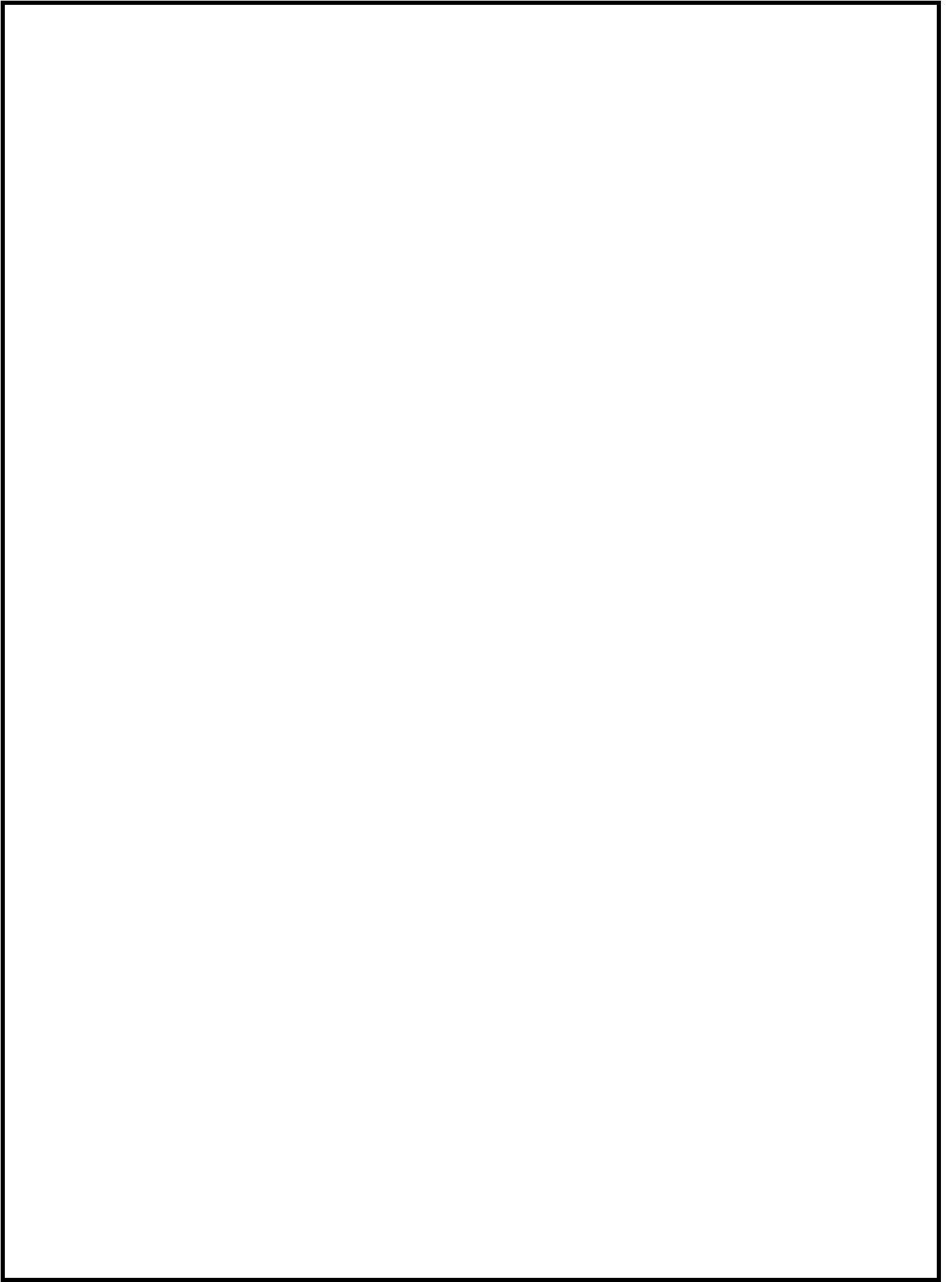
Daha sonra da bottom navigation view kısmına menu attribute olarak bu menu dosyasını vermemiz gerekiyor.

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    app:menu="@menu/bottom_navigation"
    android:id="@+id/bottomNav"
```

Son adım olarak da kod kısmında bunu belirtirsek bottom navigation view kısmını hallediyoruz.

```
val navHostFragment = supportFragmentManager.findFragmentById(R.id.navHostFragment)
val navController = navHostFragment?.findNavController()
if (navController != null) {
    binding.bottomNav.setupWithNavController(navController)
}
```

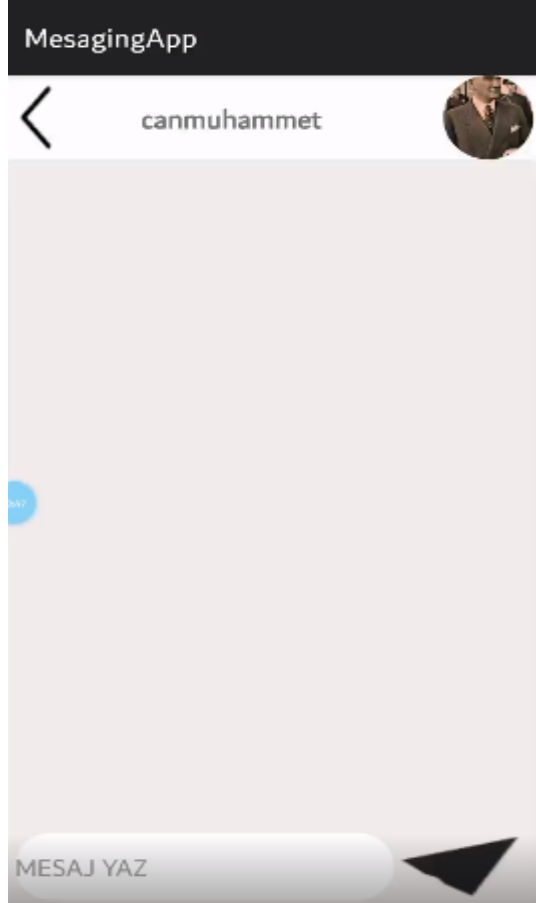
Yukarıdaki kod sayesinde navigation ile bottom navigation view birleştirme işini hallettik.Bundan sonra mesaj kısmına sıra geldi.Oturup bunu nasıl yapabileceğimizi konuştuk.Bunun için özel bir id sistemi tasarlamamız gerekiyordu,çünkü aslında her iki kullanıcı arasındaki mesajlaşma ekranında sadece o iki kullanıcı arasındaki mesajlar gösterilmeli.Bunun için de özel bir id sistemi oluşturmamız gerekiyordu.Bunu aramızda tartıştık ama tam bir sonuca varamadık.İnternette yapılan örneklere bakmaya karar verdik.Videoları izlerken bir sistem bulduk ve bunu uygulamaya karar verdik.



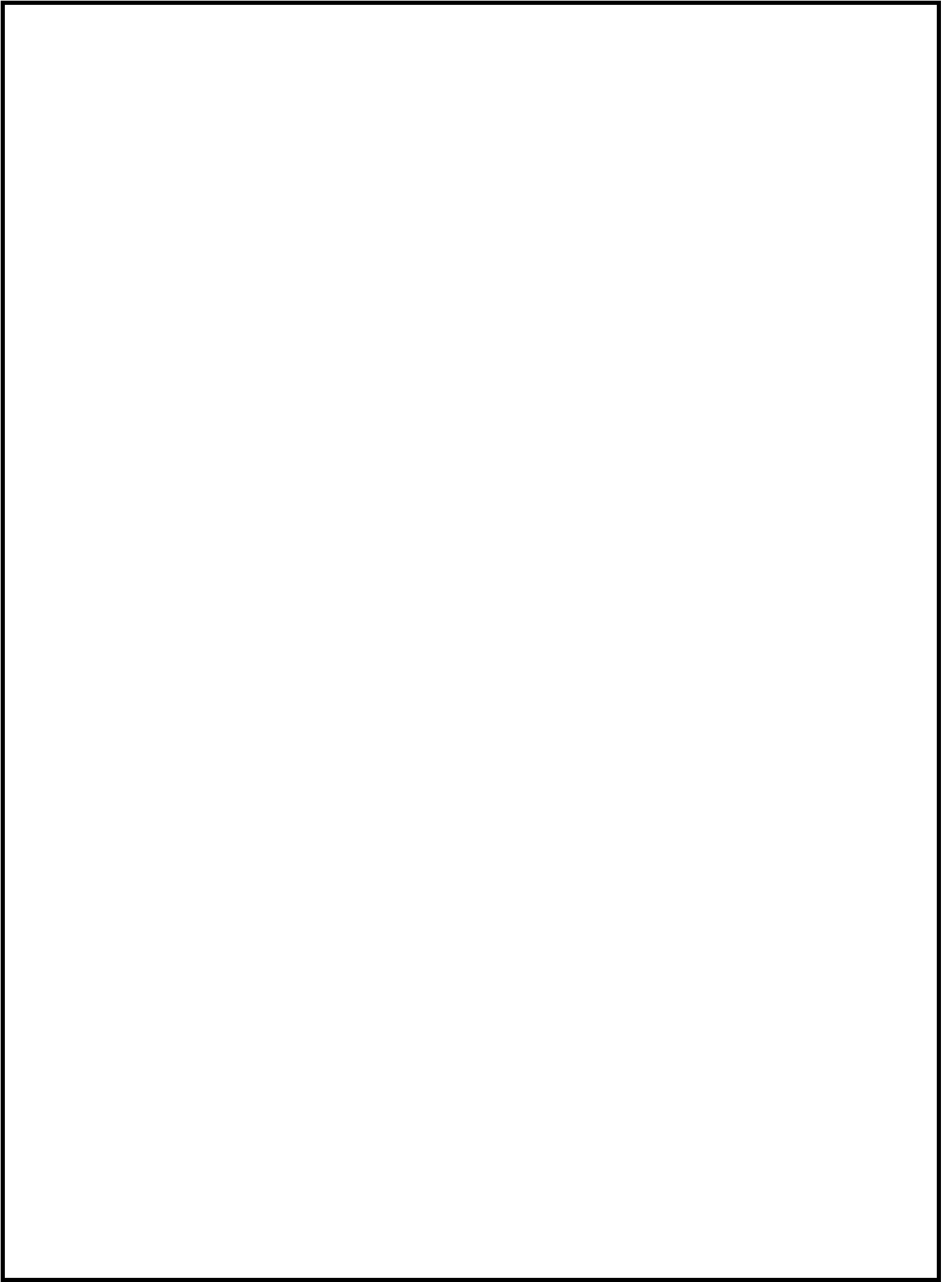
KONTROL SONUCU

KISIM	Kotlin ile Android geliştirme	Yaprak No:20
YAPILAN İŞ	Mesaj Ekranı Tasarımı	Tarih: 04/08 /2022

Bugün işe geldim ve mesajlaşma kısmını yapmaya başladık.Öncelikle mesajlaşma kısmının tasarlanması vardı.Bunun için öncelikle tasarıma karar verdik.Whatsapp benzeri bir tasarım olacaktı.Çalışmaya başladım.Üstte mesajlaşılana kişinin görseli,kullanıcı adı altta da mesajlaşma kısmı olacaktı.Tasarım aşağıdaki gibi oldu.



Aramızda tartıştıktan sonra tasarımın güzel olduğuna karar verdik.Mesajlaşma kutusu için de internette hazır bubble baktık.Bir tane deneme amaçlı bulup çalışmaya başladık.Daha sonra mesajlaşma algoritmasına geçtik.İzlediğimiz videoda kullanıcının id'si ile mesajlaştığı kişinin id'sini toplayıp özel bir oda id'si oluşturuyordu.Ancak bunun biraz karmaşık olduğunu düşündük.Kahvemizi içip konuşmaya devam ettik.Tam bir karara varamadık.Şirketteki diğer çalışanlara mesajlaşma algoritmasını nasıl yazmamız gerektiğini sorduk.Şirkette yazılan algoritma bize biraz uzaktı.Çünkü şirkette hazır bir sunucuya yazıldığı için her id hazır olarak geliyordu.Bizim böyle bir şansımız yoktu.Daha sonrasında şöyle bir karara vardık.Firebase firestore'da her user için current user adında bir document oluşturalım.Bu dokümanın içinde de mesajlaşılana kişinin kullanıcı adını argument olarak mesajlaşma fragemnt'a geçirirsek kaydetme yöntemimiz başarılı olacak diye düşündük.Ancak tam çalışmaya başlayacaktık ki işten çıkma saatinin geldiğini gördük.Yarın işte son günüm umarım bu projeyi bitirip ayrılırım.



KONTROL SONUCU



KISIM	Kotlin ile Android geliştirme	Yaprak No:21
YAPILAN İŞ	Mesajlaşma Algoritması	Tarih: 05/08 /2022

Bugün işte son günümüdü ve artık projeyi bitirmemiz gerekiyordu.Dün üzerinde düşündüğümüz algoritma üzerinde çalışmaya başladık.Kodu yazmaya başladık.

```

firebaseFirestore.collection( collectionPath: "MESSAGE").document(firebaseAuth.currentUser!!.email!!)
.collection(args.user.email).orderBy( field: "time", Query.Direction.ASCENDING)
.addSnapshotListener { value, error ->
    if (error != null) {
        showWarningToast( message: "Hata", title: "Hata Var", requireContext())
    } else {
        if (value != null) {
            if (!value.isEmpty) {
                val doc = value.documents
                chatsList.clear()
                for (i in doc) {
                    val message = i.get("message") as String
                    val from = i.get("from").toString()
                    val receive = i.get("receiver").toString()
                    val receiveUid = i.get("receiverUid").toString()
                    val senderUid = i.get("senderUid").toString()
                    val mesajModel =
                        MessageData(message, from, receive, receiveUid, senderUid)
                    Log.e( tag: "mesajlarr", mesajModel.toString())
                    chatsList.add(mesajModel)

                    adapter.chats = chatsList
                    binding.chatRecyclerView.smoothScrollToPosition( position: chatsList.size -1)
                }
            }
        }
    }
}

```

Bu kod sayesinde tüm mesajları firebaseden çekebiliyoruz.Daha sonrasında recycler view'da gelen mesajın kimden geldiğine bağlı olarak mesajı sağa ve sola ayırma işlemini yapmamız gerekiyordu.Bunun için getItemviewtype methodunu kullanacağız.

```

override fun getItemViewType(position: Int): Int {

    val chat=chats.get(position)

    if(chat.senderUid==FirebaseAuth.getInstance().currentUser!!.uid){
        return SENT_MESSAGE
    }else {
        return RECEIVED_MESSAGE
    }
    return super.getItemViewType(position)
}

```

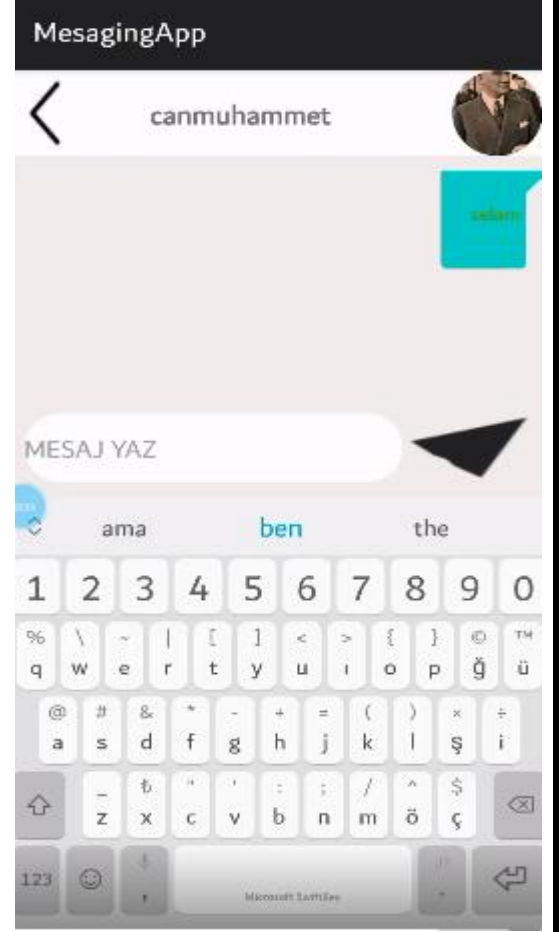
Daha sonra gelen view type göre layout inflate edeceğiz.

```

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ChatHolder {
    if(viewType==SENT_MESSAGE){
        val view=LayoutInflater.from(parent.context).inflate(R.layout.item_chat_right,parent, attachToRoot: false)
        return ChatHolder(view)
    }else{
        val view=LayoutInflater.from(parent.context).inflate(R.layout.item_chat_left,parent, attachToRoot: false)
        return ChatHolder(view)
    }
}

```


Bunun sonucunda mesaj istediğimiz gibi geliyordu,ancak bubble yani mesaj kutusunun görünümü biraz kötüydü ve mesajlar kayıyordu.Bu görünüm hoşumuza gitmediği için başka bubble aradık.Ve başka bir bubble denediğimizde hoşumuza gitti.Projeyi sonuçlandırmıştık.



Sonuç ikimiz için de tatmin ediciydi.İleride bu projeyi daha da geliştirip play store'a yükleyebiliriz.İşimiz bittiğinde şirketten ayrılma vakti gelmişti ve şirketteki diğer insanlarla vedalaştım,umarım ileride bir gün burada çalışırım.

