

# Android Eğitimi

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Senior Android – IOS Developer and Trainer

# Kotlin

# Değişken Oluşturma ve Print() Metodu Kullanım Örnek

```
var ogrenciAdi = "Ahmet"  
var ogrenciYas = 23  
var ogrenciBoy = 1.98  
var ogrenciBasharf = 'A'  
var ogrenciDevamEdiyorMu = true
```

```
println(ogrenciAdi)  
println(ogrenciYas)  
println(ogrenciBoy)  
println(ogrenciBasharf)  
println(ogrenciDevamEdiyorMu)
```

```
var urun_id:Int = 3416  
var urun_adi:String = "Macbook Pro"  
var urun_adet:Int = 100  
var urun_fiyat:Double = 34.999  
var urun_tedarikci:String = "Apple"  
  
println("Ürün id : $urun_id")  
println("Ürün adı : $urun_adi")  
println("Ürün adet : $urun_adet")  
println("Ürün fiyat : $urun_fiyat ₺")  
println("Ürün tedarikçi : $urun_tedarikci")
```

```
var a = 10  
var b = 20
```

```
println("$a x $b : ${a * b}")
```

# Constant - Sabitler

```
//Sabitler  
var sayi = 30  
    println(sayi)  
sayi = 100  
    println(sayi)
```

```
val numara = 90  
    println(numara)
```

# Tür Dönüşümü

```
//Sayısalдан sayısala dönüşüm  
val i = 42  
val d = 78.95  
  
val sonuc1 = i.toDouble()  
println(sonuc1)  
  
val sonuc2 = d.toInt()  
println(sonuc2)
```

```
//Metinden sayısala dönüşüm  
val yazı = "67.54"  
  
val sonuc5 = yazı.toDoubleOrNull()  
  
if (sonuc5 != null){  
    println(sonuc5)  
}else{  
    print("Dönüşümde hata oluştu ve sayınızı kontrol ediniz")  
}  
  
//Yöntem 3  
var yazı3 = "48"  
val z = yazı2.toIntOrNull()  
  
z?.let { it: Int  
    print(z)  
}
```

# Konsol Girdisi

```
println("Adınızı giriniz")  
  
val girdi = Scanner(System.`in`)  
  
val ad = girdi.next()  
  
println("Adınız : $ad")
```

Tüm türlerde girdi olabilir.

Bunu sağlamak için girdiye uygun tür metodu seçilmelidir.

Örn : int -> nextInt() , double -> nextDouble() vb.

# Karşılaştırma Operatörleri

```
val a = 40  
val b = 50
```

```
val x = 90  
val y = 80
```

```
println("a == b : ${a == b}")  
println("a != b : ${a != b}")  
println("a > b : ${a > b}")  
println("a >= b : ${a >= b}")  
println("a < b : ${a < b}")  
println("a <= b : ${a <= b}")
```

```
println("a > b || x > y : ${a > b || x > y}")//OR ( VEYA ) : false - false : false olur , diğer durumlarda hep true  
println("a > b && x > y : ${a > b && x > y}")//AND ( VE ) : true - true : true olur , diğer durumlarda hep false
```

# İf çalışma yapısı

```
val yas = 17
val isim = "Mehmetx"

if(yas >= 18){
    println("Reşitsiniz")
}else{
    println("Reşit değilsiniz")
}

if (isim == "Ahmet"){
    println("Merhaba Ahmet")
}else if (isim == "Mehmet"){
    println("Merhaba Mehmet")
}else{
    println("Tanınmayan kişi")
}
```

```
val ka = "admin"
val s = 123456

if (ka == "admin" && s == 12345){
    println("Hoşgeldiniz")
}else{
    println("Hatalı giriş")
}

val sayı = 11

if(sayı == 10 || sayı == 20){
    println("Sayı 10 veya 20 dir")
}else{
    println("Sayı 10 veya 20 değildir")
}
```

# Alan Hesabı

```
val girdi = Scanner(System. `in`)

println("Dikdörtgen Alanı (1)")
println("Çember alanı (2)")

val secim = girdi.nextInt()

println("Seçiminiz : $secim")

if ( secim == 1 ) {
    println("Kısa kenar giriniz")
    val kisiKenar = girdi.nextInt()

    println("Uzun kenar giriniz")
    val uzunKenar = girdi.nextInt()

    println("Dikdörtgen Alanı")
    val alan = kisiKenar * uzunKenar
    println("Sonuç : $alan")
}

if ( secim == 2 ) {
    println("Yarı çap giriniz")
    val yarıCap = girdi.nextInt()

    println("Çember Alanı")
    println("Yarı Çap : $yarıCap")
    val alan = 3.14 * yarıCap * yarıCap
    println("Sonuc : $alan")
}
```

# When Yapısı

```
val gun = 9

when ( gun ) {
    1 ->
        println("Pazartesi")
    2 ->
        println("Salı")
    3 ->
        println("Çarşamba")
    4 ->
        println("Perşembe")
    5 ->
        println("Cuma")
    6 ->
        println("Cumartesi")
    7 ->
        println("Pazar")
else ->
    println("Böyle bir gün yok")
}
```

# Dört İşlem

```
val girdi = Scanner(System.`in`)

println("Toplama (1)")
println("Çıkarma (2)")
println("Çarpma (3)")
println("Bölme (4)")

val tercih = girdi.nextInt()

println("Birinci sayıyı gir")
val sayi1 = girdi.nextInt()

println("İkinci sayıyı gir")
val sayi2 = girdi.nextInt()

when (tercih) {
    1 ->
        println("Toplama : ${sayi1+sayi2}")
    2 ->
        println("Çıkarma : ${sayi1-sayi2}")
    3 ->
        println("Çarpma : ${sayi1*sayi2}")
    4 ->
        println("Bölme : ${sayi1/sayi2}")
    else ->
        println("Böyle bir işlem yok")
}
```

# Menu İşlem

```
val girdi = Scanner(System.`in`)

println("Si (1)")
println("Ekle (2)")

val tercih = girdi.nextInt()

when (tercih) {
    1 ->
        println("Veri Silindi")
    2 ->
        println("Veri Eklendi")
    else ->
        println("Böyle bir işlem yok")
}
```

# Döngüler

```
for (i in 1..3){//1,2,3
    println("Döngü 1 : $i")
}

//10 ile 20 arasında 5 er arasıın
for (a in 10..20 step 5){
    println("Döngü 2 : $a")
}

//20 ile 10 arasında 5 er azalsın
for (b in 20 downTo 10 step 5){
    println("Döngü 3 : $b")
}

var sayac = 1

while (sayac<6){
    println("Döngü 4 : $sayac")
    //sayac+=1//sayac = sayac + 1
    sayac++
}
```

```
for (i in 1..5){
    if (i == 3){
        break
    }
    println("Döngü 5 : $i")
}

for (i in 1..5){
    if (i == 3){
        continue
    }
    println("Döngü 6 : $i")
}
```

## Döngü : İsim Tekrarla 1

```
val girdi = Scanner(System.`in`)  
println("Tekrar sayısı giriniz")  
val tekrar = girdi.nextInt()  
val isim = "Ahmet"  
for ( i in 1..tekrar){  
    println("$i. $isim")  
}
```

## Döngü : İsim Tekrarla 2

```
val girdi = Scanner(System.`in`)

println("İsim giriniz")

val isim = girdi.next()

println("Tekrar sayısı giriniz")

val tekrar = girdi.nextInt()

for ( i in 1..tekrar){
    println("$i. $isim")
}
```

# Döngü : Veri İşleme

```
val girdi = Scanner(System.`in`)

println("İşlenecek veri miktarı giriniz")

var veri = girdi.nextInt()

while (veri > 0) {
    println("$veri. veri")
    veri-=1
}
```

# NESNE TABANLI PROGRAMLAMA

# Araba Analojisi

```
data class Araba(var renk:String,var hiz:Int,var calisiyorMu:Boolean) {  
    init {  
        println("Araba sınıfından nesne oluşturuldu")  
    }  
    fun calistir(){  
        calisiyorMu = true  
        hiz = 5  
    }  
    fun durdur(){  
        calisiyorMu = false  
        hiz = 0  
    }  
    fun hizlan(kacKm:Int){  
        hiz+=kacKm//hiz = hiz + kacKm  
    }  
    fun yavasla(kacKm:Int){  
        hiz-=kacKm//hiz = hiz - kacKm  
    }  
    fun bilgiAl(){  
        println("*****")  
        println("Renk : ${renk}")  
        println("Hız : ${hiz}")  
        println("Çalışıyor mu : ${calisiyorMu}")  
    }  
  
    //this veya self : Bulunduğumuz sınıf  
    //super : Kalıtım varsa üst sınıfı temsil  
}
```

```
val bmw = Araba( renk: "Kırmızı", hiz: 0, calisiyorMu: false)  
//Okuma işlemi  
bmw.bilgiAl()  
//Veri atama  
bmw.hiz = 10  
bmw.calisiyorMu = true  
  
bmw.bilgiAl()  
bmw.durdur()  
bmw.bilgiAl()  
bmw.calistir()  
bmw.bilgiAl()  
bmw.hizlan( kacKm: 100)  
bmw.bilgiAl()  
bmw.yavasla( kacKm: 50)  
bmw.bilgiAl()  
  
val sahin = Araba( renk: "Beyaz", hiz: 100, calisiyorMu: true)  
sahin.bilgiAl()  
sahin.durdur()  
sahin.bilgiAl()  
sahin.calistir()  
sahin.bilgiAl()  
sahin.hizlan( kacKm: 70)  
sahin.bilgiAl()  
sahin.yavasla( kacKm: 50)  
sahin.bilgiAl()
```

isim ADALAN

# Primary Constructor için `init` kullanımı

- Nesne oluşturulurken değişkenlere değer atamanın haricinde başka bir işlem yapmak için kullanılır.
- Nesne oluşturulduğu anda çalışır.

```
class Kisiler(var ad:String,var yas:Int) {  
    init {  
        println("Primary Constructor Çalıştı")  
    }  
}
```

```
val kisi = Kisiler(ad: "Ahmet", yas: 18)  
println(kisi.ad)      Nesneoluştugu anda  
println(kisi.yas)    init çalışır.
```

ÇIKTI :

```
Primary Constructor Çalıştı  
Ahmet  
18
```

# Fonksiyonlar

```
class Fonksiyonlar {
    fun selamla1(){
        val sonuc = "Merhaba Ahmet"
        println(sonuc)
    }

    fun selamla2() : String {
        val sonuc = "Merhaba Ahmet"
        return sonuc
    }

    fun selamla3(isim:String){
        val sonuc = "Merhaba $isim"
        println(sonuc)
    }

    fun toplama1(){
        val toplam = 30 + 40
        println(toplam)
    }

    fun toplama2(sayı1:Int,sayı2:Int) : Int {
        val toplam = sayı1 + sayı2
        return toplam
    }
}

fun main() {
    val f = Fonksiyonlar()
    f.selamla1()

    val gelenSonuc = f.selamla2()
    println("Gelen Sonuç : $gelenSonuc")

    f.selamla3( isim: "Zeynep" )

    f.toplama1()

    val gelenToplam = f.toplama2( sayı1: 100, sayı2: 200 )
    println("Gelen Toplam : $gelenToplam")

    f.carp( sayı1: 5, sayı2: 4, isim: "Ece" )
}

//Overloading
fun carp(sayı1:Int,sayı2:Int){
    println("Çarpma : ${sayı1 * sayı2}")
}

fun carp(sayı1:Double,sayı2:Int){
    println("Çarpma : ${sayı1 * sayı2}")
}

fun carp(sayı1:Int,sayı2:Int,isim:String){
    println("Çarpma : ${sayı1 * sayı2} - İşlem yapan : $isim")
}
```

# Extension Fonksiyon

- Extension fonksiyonlar kotlin içinde bulunan sınıfları genişletmek için kullanılan pratik bir yöntemdir.
- Örneğin int sınıfına özellik katabiliriz.

```
fun Int.carpi(sayı: Int) : Int{  
    return this * sayı  
    //this metodun isminin önündeki sayıyı ifade eder.  
}
```

```
val sonuc = 5.carpi( sayı: 2)  
println(sonuc)//10
```

# infix Fonksiyon

- Extension fonksiyonları daha pratik kullanmamızı sağlar.

```
infix fun Int.carpi(sayı: Int) : Int{  
    return this * sayı  
    //this metodun isminin önündeki sayıyı ifade eder.  
}  
  
extension karşılığı  
val sonuc = 5 carpi 2  
val sonuc = 5.carpi( sayı: 2)  
  
println(sonuc)//10
```

```
fun main() {  
    val sonuc = 5 carpi 2  
    println("Gelen Sonuç : $sonuc")  
}  
  
infix fun Int.carpi(sayı:Int):Int{  
    return this * sayı  
}
```

# Nullable

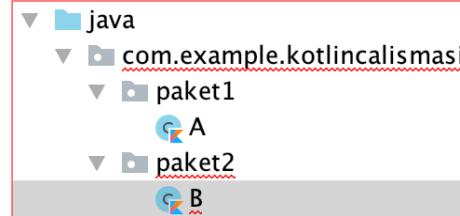
```
var str1:String? = null //Yöntem 3 :  
//str1 = " merhaba "  
  
if (str1 != null){  
    println("Sonuç 3 : ${str1.trim()}")  
}  
else{  
    println("Sonuç nulldur")  
}  
  
//Yöntem 1 :  
println("Sonuç 1 : ${str1?.trim()}")  
  
//Yöntem 2 :  
//println("Sonuç 2 : ${str1!!.trim()}")  
  
//Yöntem 4:  
str1?.let { it: String  
    println("Sonuç 4 : ${str1.trim()}")  
}
```

# lateinit

```
class MainActivity : AppCompatActivity() {  
  
    //lateinit kullanımı  
    //Sadece referans tiplere eklenebilir. Primitif tiplerde eklenemez.  
  
    //Boş bir değişken oluşturamayız  
    var str1:String  
  
    //Nullable ile null değer verebiliriz ve boş bir tanımlama olabilir.  
    var str2:String? = null  
  
    //lateinit kullanarak boş bir değişken oluşturabiliriz.  
    lateinit var str3:String  
  
    //Primitif tiplerde eklenemez.  
    lateinit var sayı:Int  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

# Paketler & Import

- **Paketler** birden fazla sınıfı kümeyelediğimiz yapılardır.
  - **Paketler** ile daha düzenli projeler oluşturulabilir.
- 
- **Import** bir sınıfı başka bir sınıf içinde kullanıcaksa o sınıfı **import** etmeliyiz.
  - Bir sınıfın tüm uzantısı kullanıldığında **import'a** gerek yoktur.



```

package com.example.kotlincalismasi.paket2

import com.example.kotlincalismasi.paket1.A

class B {

    fun fonksiyon() {
        val nesne = A()

        println(nesne.varsayılanDegişken)
        println(nesne.publicDegişken)
        println(nesne.privateDegişken) //Hata : Aynı sınıf içinde olmadığımız için
        println(nesne.internalDegişken)
        println(nesne.protectedDegişken)//Hata : Aynı sınıf veya bulunduğu sınıfın从中 türemediği için

    }
}
  
```

# Visibility Modifier

# Visibility Modifiers ( Erişim )

|  |   |   |   |
|--|---|---|---|
| <ul style="list-style-type: none"><li>• <b>Public</b><ul style="list-style-type: none"><li>• Sınıf</li><li>• Metod</li><li>• Değişken</li></ul></li><li>• (Bütün sınıflardan ve paketlerden ulaşılabilir.)</li></ul> | <ul style="list-style-type: none"><li>• <b>Internal</b><ul style="list-style-type: none"><li>• Metod</li><li>• Değişken</li></ul></li><li>• (Yalnızca bulunduğu modül içinde yer alan sınıflar ulaşılabilir.)</li><li>• Paket ile modül aynı kavram değildir.</li></ul> | <ul style="list-style-type: none"><li>• <b>Private</b><ul style="list-style-type: none"><li>• Metod</li><li>• Değişken</li></ul></li><li>• (Yalnızca tanımlandıkları sınıfın içinden ulaşılabilir.)</li></ul> | <ul style="list-style-type: none"><li>• <b>Protected</b><ul style="list-style-type: none"><li>• Metod</li><li>• Değişken</li></ul></li><li>• (Bulundukları sınıfın veya bulundukları sınıfın türetildiklerinde ulaşılabilir.)</li></ul> |
|--|---|---|---|

**Not :** Hiçbir access modifier kullanılmıyorsa public anlamına gelir.

# Visibility Modifier Örnek

```
class A {  
    var varsayılanDegisken = 1  
    public var publicDegisken = 2  
    private var privateDegisken = 3  
    internal var internalDegisken = 4  
    protected var protectedDegisken = 5  
}
```

```
class B {  
  
    fun fonksiyon() {  
  
        val nesne = A()  
  
        println(nesne.varsayılanDegisken)  
        println(nesne.publicDegisken)  
        println(nesne.privateDegisken) //Hata : Aynı sınıf içinde olmadığımız için  
        println(nesne.internalDegisken)  
        println(nesne.protectedDegisken)//Hata : Aynı sınıf veya bulunduğu sınıfın türemediği için  
    }  
}
```

# Visibility Modifier Örnek

Eğer çalıştığımız sınıfı protected değişkenin bulunduğu sınıfın dışından türetirsek protected hatası kalkar.

```
open class A {  
    var varsayılanDegisen = 1  
    public var publicDegisen = 2  
    private var privateDegisen = 3  
    internal var internalDegisen = 4  
    protected var protectedDegisen = 5  
}
```

```
class B : A() {  
  
    //Kalıtım olduğu için A sınıfı içindeki değişkenlere direkt erişim sağlanabilir.  
  
    fun fonksiyon() {  
  
        println(varsayılanDegisen)  
        println(publicDegisen)  
        println(privateDegisen) //Hata : Aynı sınıf içinde olmadığıımız için  
        println(internalDegisen)  
        println(protectedDegisen)  
    }  
}
```

# Visibility Modifier Örnek

Eğer değişkenleri kendi sınıfı içinden çağrırsak hiç sorun yaşamayız.

```
class A {  
    var varsayılanDegisken = 1  
    public var publicDegisken = 2  
    private var privateDegisken = 3  
    internal var internalDegisken = 4  
    protected var protectedDegisken = 5  
  
    fun fonksiyon() {  
        println(varsayılanDegisken)  
        println(publicDegisken)  
        println(privateDegisken)  
        println(internalDegisken)  
        println(protectedDegisken)  
    }  
}
```

# Data Class

# Data Class

- Java üzerinde çalışma yaparken oluşturduğumu model sınıflarında dolu constructor,setter getter>equals,hascode gibi yapıları kendimiz koymak zorundaydık.
- Kotlin üzerinde sınıfı data keyword'u eklendiğinde bu özelliklerin hepsine sahip olmuş oluyor.
- Genelde veritabanı tablolarının modelleri için kullanılır.

```
data class Kisiler(var ad:String) {  
}
```

# Java ile Data Class Karşılaştırması

JAVA

```
public class Person {  
    private String ad;  
  
    public Person(String ad) {  
        this.ad = ad;  
    }  
  
    public String getAd() {  
        return ad;  
    }  
  
    public void setAd(String ad) {  
        this.ad = ad;  
    }  
  
    //toString  
    //hashCode  
    //equals  
}
```

Kotlin

```
data class Kisiler(var ad:String) {  
}
```

# Static Değişkenler ve Metodlar

# Static Değişkenler ve Metodlar

- Bir değişkenin veya metodun, bulunduğu sınıfın ( class ) nesne oluşturmaya gerek kalmadan erişilmek istenirse kullanılır.

```
class Asinifi {  
  
    companion object {  
        //Direk erişim için bu alan içine kodlama yapılır.  
        var x = 10  
  
        fun metod(){  
            println("Merhaba")  
        }  
    }  
}
```

```
println(Asinifi.x)  
Asinifi.metod()
```

# Enumeration

# Enumeration

- `enum` ifadesi gösterilir.
- Parametrelerde kullanılır.
- Verilerin eşleşmesi sonucunda bir işlem yapılır.
- Kodlama yapan yazılımcıyı detaydan kurtarmaktadır.

```
enum class Renkler {  
    Beyaz, Siyah  
}
```

```
val renk = Renkler.Beyaz  
  
when (renk) {  
    Renkler.Beyaz -> print("#FFFFFF")  
    Renkler.Siyah -> print("#000000")  
}
```

# Enumeration - Örnek

```
fun main() {
    ucretHesapla(adet: 100, KonserveBoyut.ORTA)
}

fun ucretHesapla(adet:Int, boyut:KonserveBoyut){
    when(boyut){
        KonserveBoyut.KUCUK ->
            println("Toplam maliyet : ${30 * adet} ₺")
        KonserveBoyut.ORTA ->
            println("Toplam maliyet : ${80 * adet} ₺")
        KonserveBoyut.BUYUK ->
            println("Toplam maliyet : ${140 * adet} ₺")
    }
}

enum class KonserveBoyut {
    KUCUK, ORTA, BUYUK
}
```

# Composition

# Composition

- Başka sınıflardan ( class ) olmuş nesneler bir sınıfın değişkeni olabilir.

```
class Adres(var il:String,var ilce:String) {  
}
```

```
class Kisiler(var ad:String,var yas:Int,var adres:Adres) {  
}
```

```
val adres = Adres( il: "Bursa", ilce: "Osmangazi")  
  
val kisi = Kisiler( ad: "Ahmet", yas: 18,adres)  
  
println("Kişi ad    : ${kisi.ad}")  
println("Kişi yaş   : ${kisi.yas}")  
println("Adres il   : ${kisi.adres.il}")  
println("Adres ilçe : ${kisi.adres.ilce}")
```

# Composition

Kategoriler Tablosu

| kategori_id | kategori_ad |
|-------------|-------------|
| 1           | Dram        |
| 2           | Komedi      |
| 3           | Bilim Kurgu |

Yonetmenler Tablosu

| yonetmen_id | yonetmen_ad       |
|-------------|-------------------|
| 1           | Nuri Bilge Ceylan |
| 2           | Quentin Tarantino |
| 3           | 2013              |

Filmler Tablosu

| film_id | film_ad   | film_yil | kategori_id | yonetmen_id |
|---------|-----------|----------|-------------|-------------|
| 1       | Django    | 2013     | 1           | 2           |
| 2       | Inception | 2006     | 3           | 3           |

```

data class Filmler(
    var film_id:Int,
    var film_ad:String,
    var film_yil:Int,
    var kategori: Kategoriler,
    var yonetmen: Yonetmenler)
}

data class Yonetmenler(
    var yonetmen_id:Int,
    var yonetmen_ad:String) {
}

data class Kategoriler(
    var kategori_id:Int,
    var kategori_ad:String) {
}

val k1 = Kategoriler( kategori_id: 1, kategori_ad: "Dram")
val k2 = Kategoriler( kategori_id: 2, kategori_ad: "Bilim Kurgu")

val y1 = Yonetmenler( yonetmen_id: 1, yonetmen_ad: "Quentin Tarantino")
val y2 = Yonetmenler( yonetmen_id: 2, yonetmen_ad: "Christopher Nolan")

val f1 = Filmler( film_id: 1, film_ad: "Django", film_yil: 2013,k1,y1)

println("Film id : ${f1.film_id}")
println("Film ad : ${f1.film_ad}")
println("Film yıl : ${f1.film_yil}")
println("Film kategori : ${f1.kategori.kategori_ad}")
println("Film yönetmen : ${f1.yonetmen.yonetmen_ad}")

```

Veritabanı modeli olduğu  
için data class olması iyi  
olacaktır.

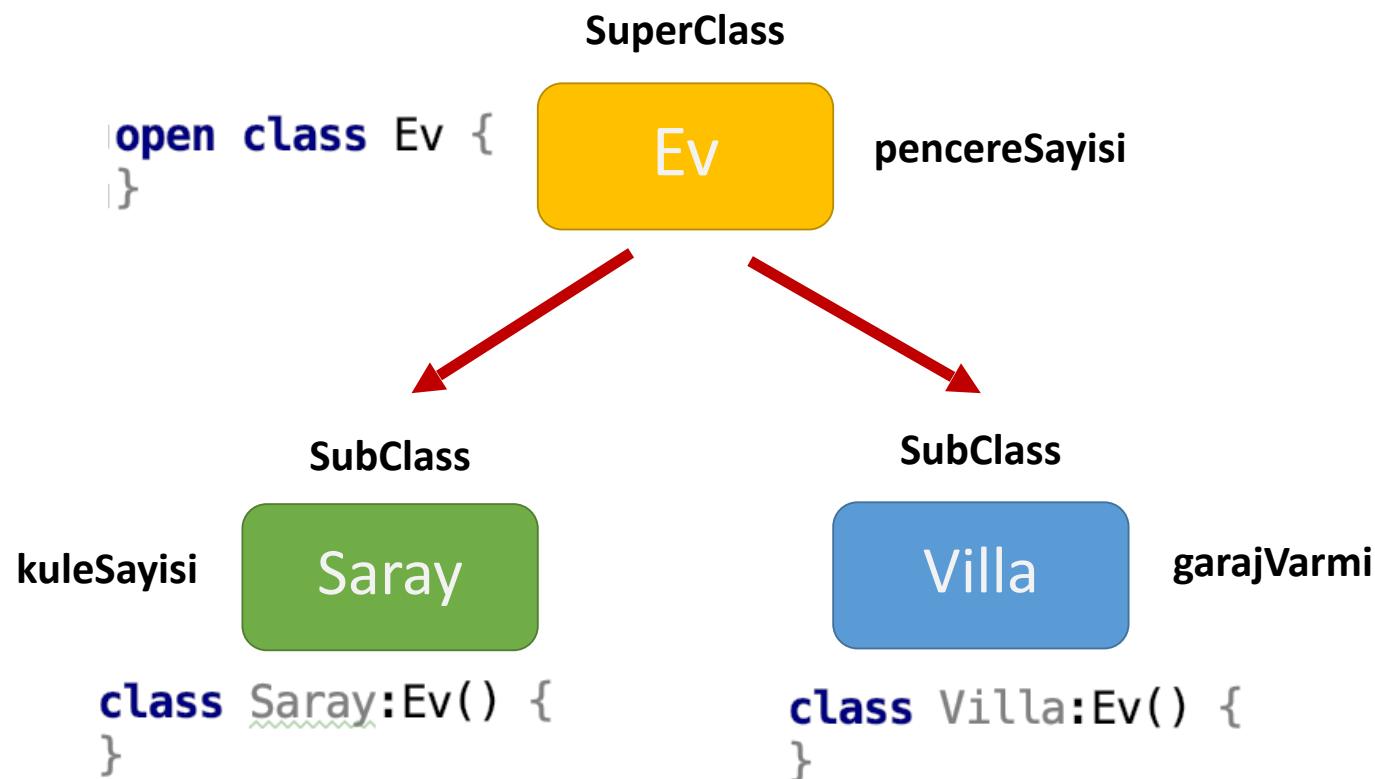
# OOP Kuralı – Kalıtım ( Inheritance )

- Mevcut bir sınıfın başka bir sınıf türetmek için kullanılır.
- Kodun tekrar kullanabilirliğini artırır.
- Sadece **class** için geçerlidir.
- Super class **open** kelimesi ile tanımlanır.
- : işaretleri ile tanımlanır.
- Bir sınıfın tek kalıtımı olabilir.
- Bir sınıfa birden fazla sınıf kalıtım yolu ile bağlanamaz.
- Üst sınıfa **superclass** denir.
- Alt sınıfa **subclass** denir.

```
open class Arac {  
}  
  
open class Araba : Arac() {  
}  
  
class Nissan:Araba() {  
}
```



# Kalıtım Hiyerarşisi Örnek



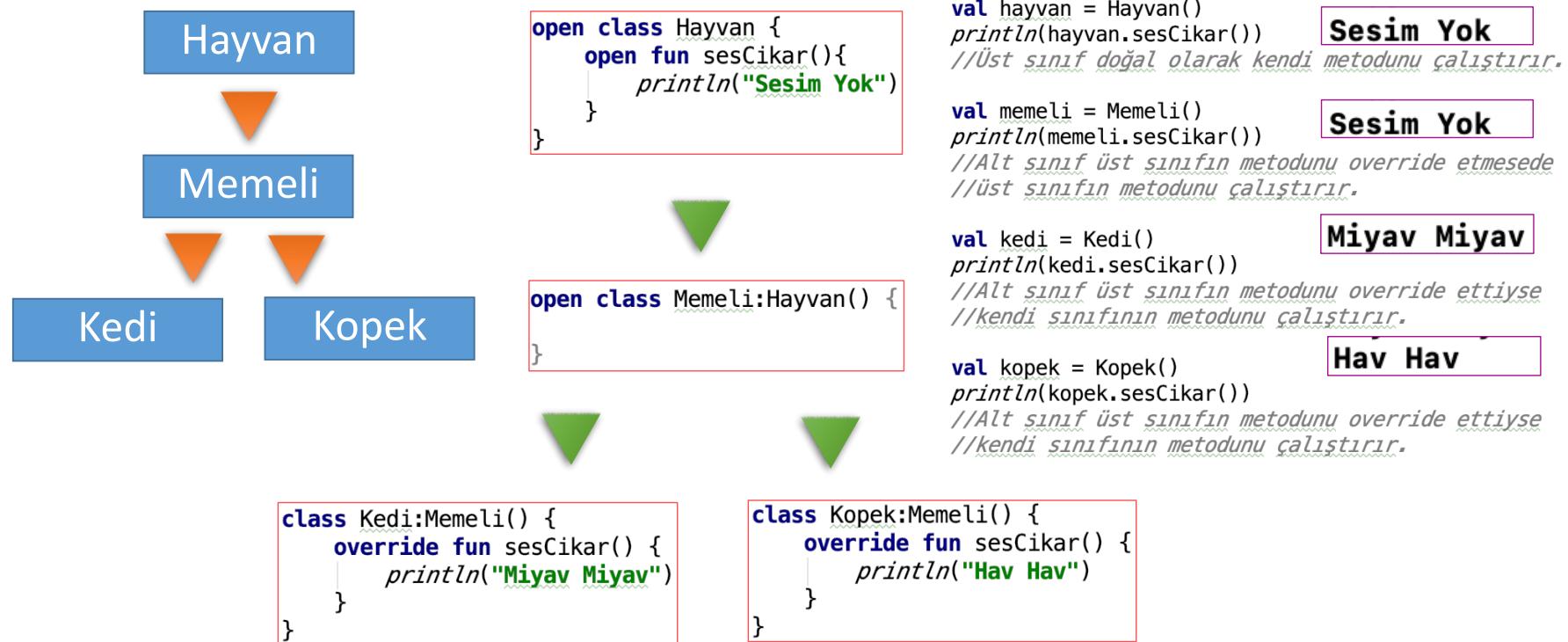
# Kalıtım Hiyerarşisi Örnek

```
open class Ev(var pencereSayisi:Int) {  
}  
  
class Saray(var kuleSayisi:Int,  
            pencereSayisi:Int) : Ev(pencereSayisi) {  
}  
  
class Villa(var garajVarmi:Boolean,  
            pencereSayisi:Int) : Ev(pencereSayisi) {  
}  
  
val topkapiSarayı = Saray( kuleSayisi: 5, pencereSayisi: 300)  
val bogazVilla = Villa( garajVarmi: true, pencereSayisi: 10)  
  
println(topkapiSarayı.kuleSayisi)  
println(topkapiSarayı.pencereSayisi)  
  
println(bogazVilla.garajVarmi)  
println(bogazVilla.pencereSayisi)
```

# Override

# Metodları Ezme : Overriding

- Kalıtım ilişkisinde üst sınıfın metodlarının alt sınıf tarafından tekrar kullanılmasıdır.
- override edilecek metod'a **open** kelimesi eklenmelidir.



# Nesnelerin Tip Dönüşümü

## Tip Kontrolü - **is**

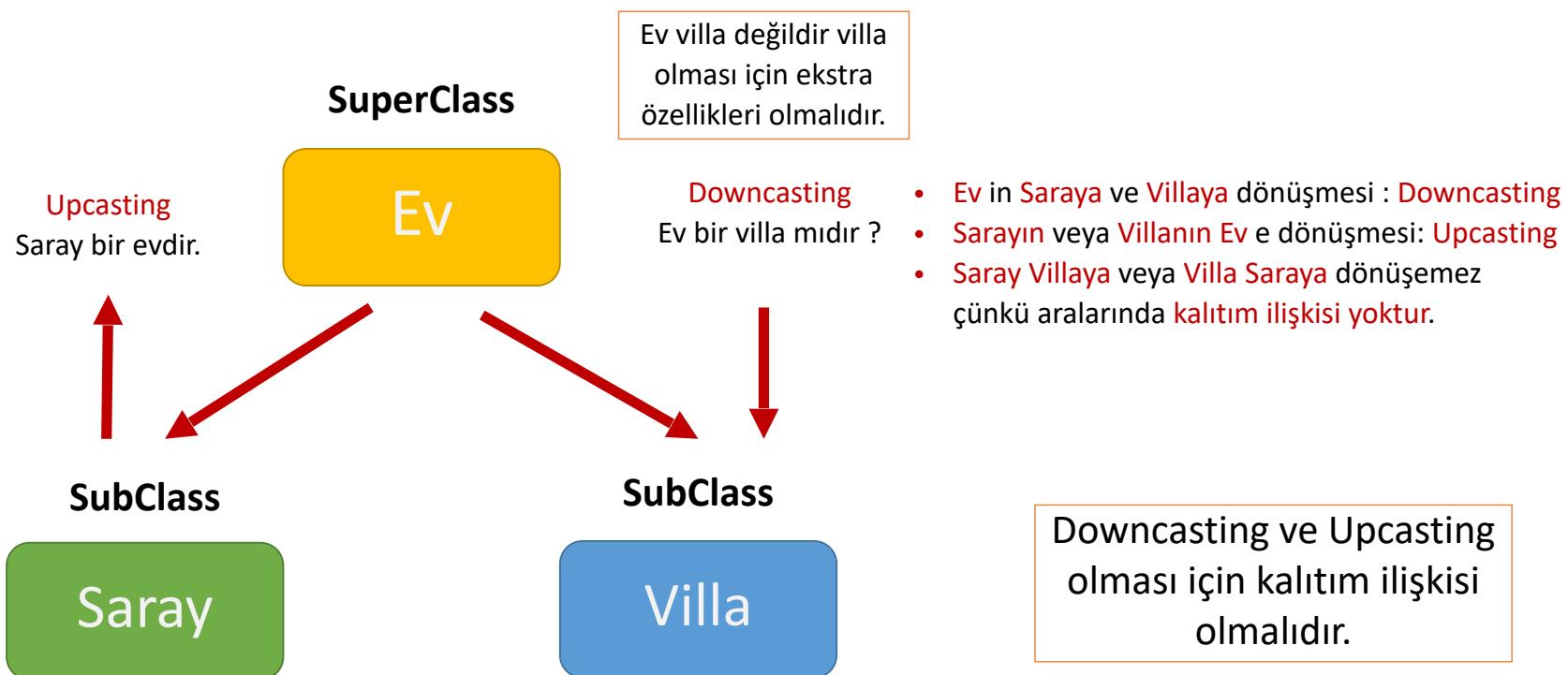
- Tip kontrolü **is** ile yapılabilir. `is true` `false` şeklinde bilgi verir.

```
val topkapiSarayı = Saray( pencereSayısı: 100, kuleSayısı: 10)  
if (topkapiSarayı is Saray){  
    println("Saraydır")  
}else{  
    println("Saray Değildir")  
}
```

ÇIKTI :

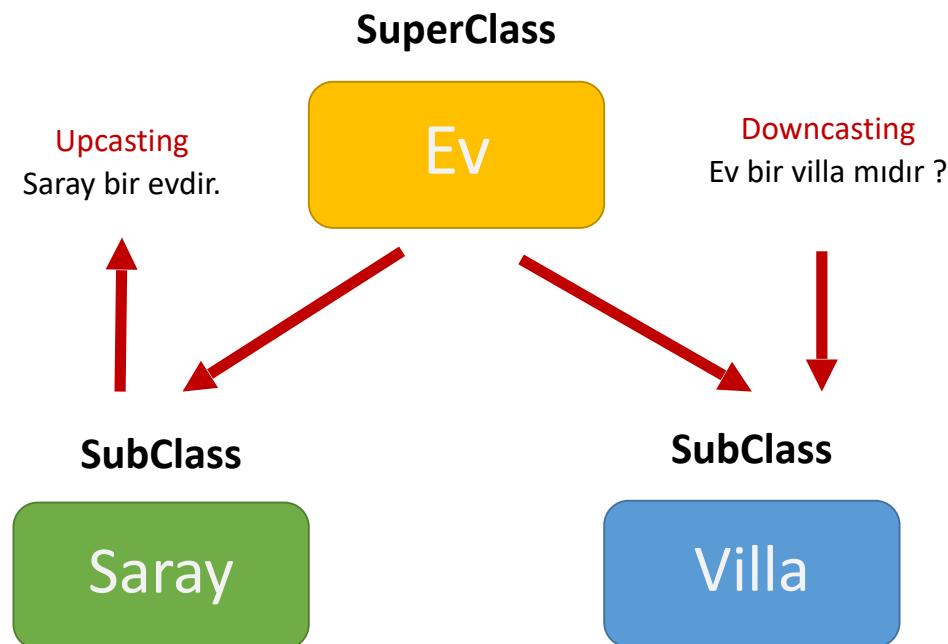
Boxed:  
Bu saraydır!

# Downcasting - UpCasting



# Upcasting

```
val ev1:Ev = Saray( pencereSayisi: 5, kuleSayisi: 3)  
//Upcasting
```



## Downcasting - as

```
val ev2 = Ev( pencereSayisi: 2)
```

```
val saray1 = ev2 as Saray  
//Downcasting
```

SuperClass

Upcasting  
Saray bir evdir.

Ev

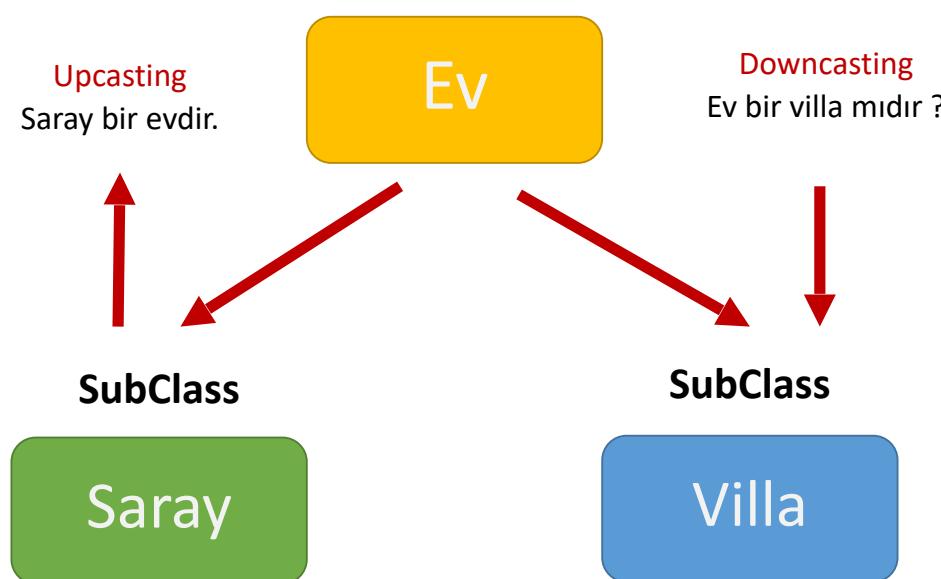
Downcasting  
Ev bir villa midir ?

SubClass

Saray

SubClass

Villa

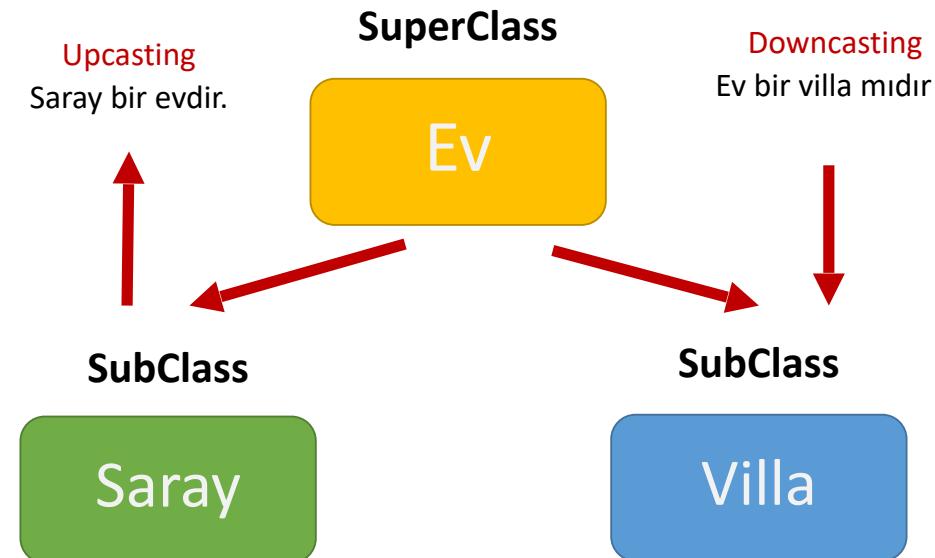


# Downcasting - as? - (nullable) cast

```
val ev2 = Ev( pencereSayisi: 2)
```

```
val saray2 = ev2 as? Saray  
//Downcasting  
//? işaretini ile güvenli castin olmuş olur.  
//Eğer dönüşümde sorun olursa  
//saray2 değişkeni içine null aktarılır.
```

? işaretini ile güvenli casting olmuş olur.  
Dönüşüm olurken hata oluşması engellenir.  
Eğer dönüşümde sorun olursa saray2 değişkeni içine null aktarılır.



# Interface

# Interface

- Class yapısında kullanılabilir.
- Bir birden fazla interface alabilir.
- : ile eklenirler.
- Kalıtım gibi sınıfın sonuna () konulmaz.
- Hazır taslaklar gibi düşünebilirsiniz.
- Interface'ler sınıflara özellik katar.

```
interface MyInterface {  
    val degisken:Int  
    fun metod1()  
  
    fun metod2() : String  
}  
  
class ClassA : MyInterface {  
    override val degisken: Int = 10  
  
    override fun metod1() {  
        println("Metod 1 çalıştı")  
    }  
  
    override fun metod2(): String {  
        return "Metod 2 çalıştı"  
    }  
}
```

# Interface Örnek

```
interface MyInterface {  
    val degisken:Int  
  
    fun metod1()  
  
    fun metod2() : String  
}  
  
val a = ClassA()  
println(a.degisken)  
a.metod1()  
println(a.metod2())
```

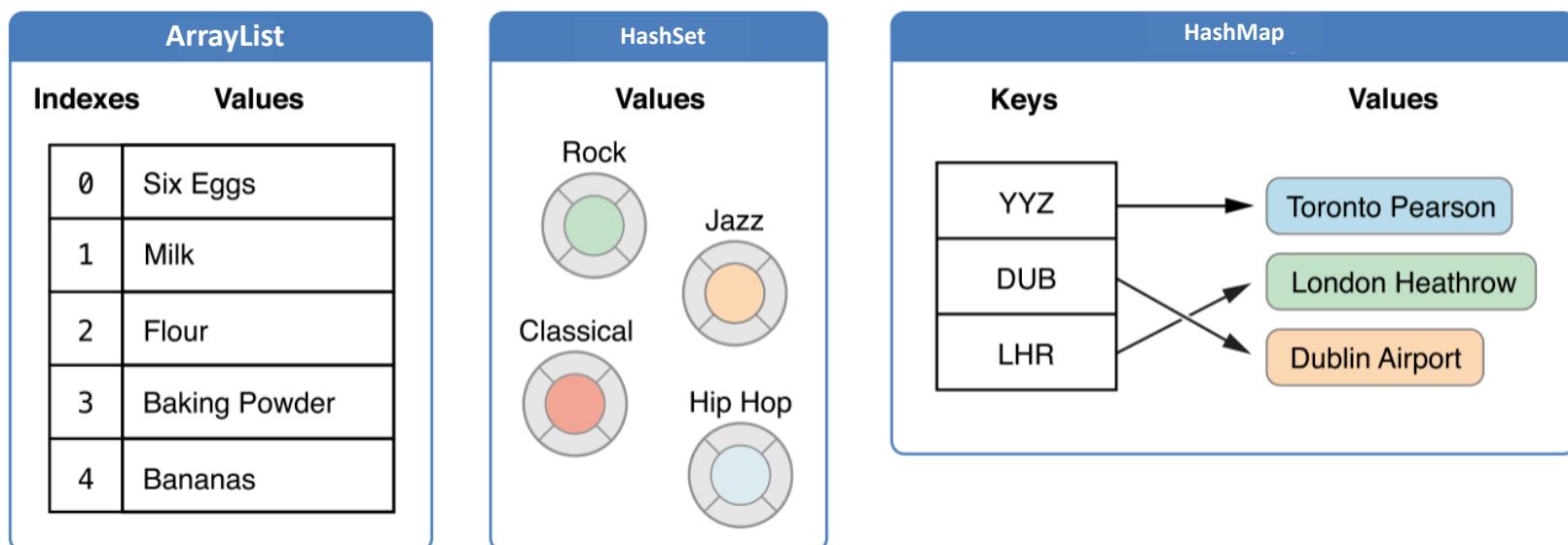
```
class ClassA : MyInterface {  
    override val degisken: Int = 10  
  
    override fun metod1() {  
        println("Metod 1 çalıştı")  
    }  
  
    override fun metod2(): String {  
        return "Metod 2 çalıştı"  
    }  
}
```

# Collections

# Mutable Collections

- Mutable özelliği diziler hem yazılabilir hem okunabilir.
- Immutable diziler sadece okunabilir.
- Böyle bir ayırım yapmalarının sebebi kodlama yaparken büyük çoğunlukla veri tabanlarından alınan dizileri okuma işlemi yapmamız diyebiliriz.
- Performansı artırmak amaçlanmıştır.

# Collection Types



# ArrayList İşlemleri

```

val sayilar = ArrayList<Int>()

val meyveler = ArrayList<String>()

//Veri ekleme
meyveler.add("Elma")//0.
meyveler.add("Muz")//1.
meyveler.add("Kiraz")//2.

println(meyveler.toString())

meyveler[1] = "Yeni Muz"
println(meyveler.toString())

meyveler.add( index: 1, element: "Portakal")
println(meyveler.toString())

//Okuma işlemi
println(meyveler.get(2))
println(meyveler[3])

println("Boyut : ${meyveler.count()}")
println("Boyut : ${meyveler.size}")
println("Boş kontrol : ${meyveler.isEmpty()}")
println("İçeriyor mu : ${meyveler.contains("Kirazx")}") Kasım ADALAN

meyveler.reverse()
println(meyveler.toString())

meyveler.sort()
println(meyveler.toString())

for (meyve in meyveler){
    println("Sonuç 1 : $meyve")
}

for ((indeks,meyve) in meyveler.withIndex()){
    println("$indeks. -> $meyve")
}

meyveler.removeAt( index: 2)
println(meyveler)

meyveler.clear()
println(meyveler)

```

# Nesne Tabanlı Kullanım

```
data class Ogrenciler(var no:Int, var ad:String, var sinif:String) {  
}  
  
    val o1 = Ogrenciler( no: 200, ad: "Zeynep", sinif: "9C")  
    val o2 = Ogrenciler( no: 300, ad: "Ahmet", sinif: "11Z")  
    val o3 = Ogrenciler( no: 100, ad: "Beyza", sinif: "12A")  
  
    val ogrencilerListesi = ArrayList<Ogrenciler>()  
  
    ogrencilerListesi.add(o1)  
    ogrencilerListesi.add(o2)  
    ogrencilerListesi.add(o3)  
  
    for (o in ogrencilerListesi){  
        println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")  
    }
```

# Sıralama

```
println("Sayısal : Küçükten -> Büyüge")
val siralama1 = ogrencilerListesi.sortedWith(compareBy { it.no })
for (o in siralama1){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}

println("Sayısal : Büyüktten -> Küçüğe")
val siralama2 = ogrencilerListesi.sortedWith(compareByDescending{ it.no })
for (o in siralama2){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}

println("Harfsel : Küçükten -> Büyüge")
val siralama3 = ogrencilerListesi.sortedWith(compareBy{ it.ad })
for (o in siralama3){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}

println("Harfsel : Büyüktten -> Küçüğe")
val siralama4 = ogrencilerListesi.sortedWith(compareByDescending{ it.ad })
for (o in siralama4){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}
```

# Filtreleme

```
println("Filtreleme 1")
val filtreleme1 = ogrencilerListesi.filter { (it.ad).contains( other: "y" ) }
for (o in filtreleme1){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}

println("Filtreleme 2")
val filtreleme2 = ogrencilerListesi.filter { it.no >= 200 }
for (o in filtreleme2){
    println("No : ${o.no} - Ad : ${o.ad} - Sınıf : ${o.sinif}")
}
```

# Set İşlemleri

```
val meyveler = HashSet<String>()

meyveler.add("Elma")
meyveler.add("Kiraz")
meyveler.add("Muz")
println(meyveler.toString())

meyveler.add("Amasya Elması")
println(meyveler.toString())

println(meyveler.elementAt( index: 1))
println(meyveler.size)
println(meyveler.isEmpty())

for (m in meyveler){
    println("Sonuç : $m")
}

for ((i,m) in meyveler.withIndex()){
    println("$i. -> $m")
}

meyveler.remove(element: "Elma")
println(meyveler.toString())

meyveler.clear()
println(meyveler.toString())
```

# Map İşlemleri

```
val iller = HashMap<Int, String>()
iller.put(16, "BURSA")
iller.put(34, "İSTANBUL")
iller[6] = "ANKARA"

println(iller.toString())

println(iller.get(16))

iller.put(16, "YENİ BURSA")
println(iller.toString())

println(iller.size)
println(iller.isEmpty())

for((anahtar,deger) in iller){
    println("$anahtar -> $deger")
}

iller.remove(key: 34)
println(iller.toString())

iller.clear()
println(iller.toString())
```

# ileri Android

# try catch blogu

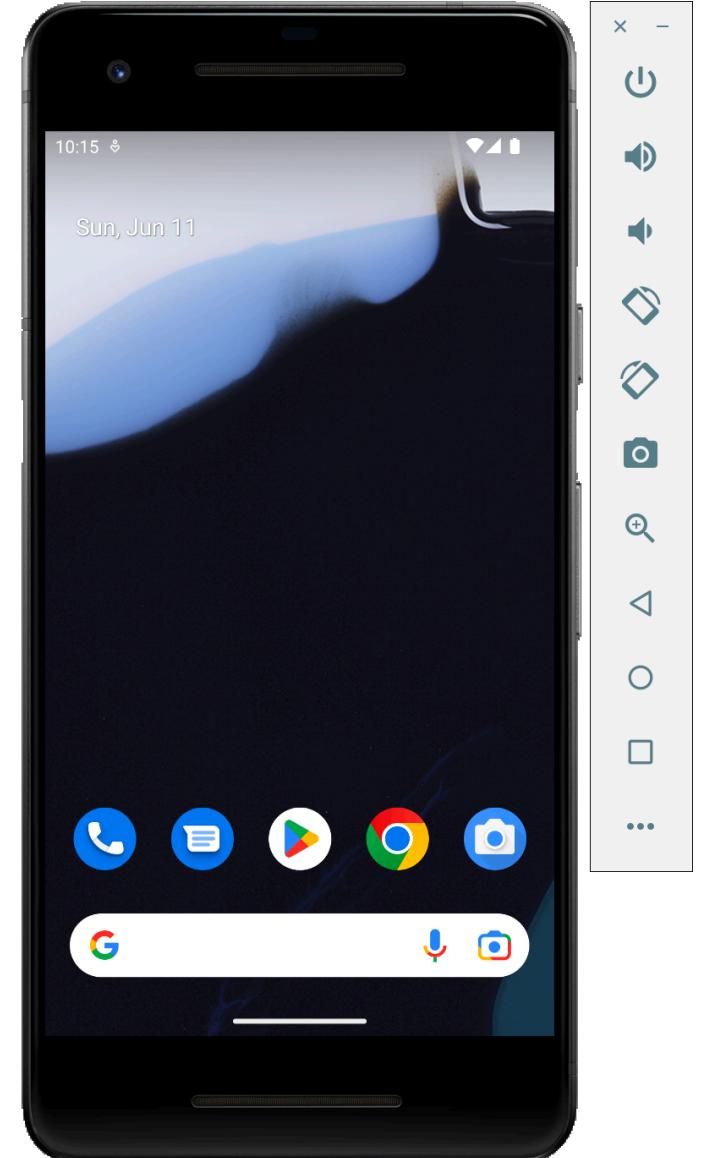
```
val x = 10
val y = 0

try{
    println("Sonuç : ${x/y}")
    println("İşlem Tamam")
}catch (e:Exception){
    println("İkinci sayı sıfır olamaz")
}
```

# Android Giriş

# AVD Kurulumu

- **AVD** ( Android Virtual Device )(Android Sanal Cihaz)
- Geliştirdiğimiz uygulamaların çalışmasını görebilmek için kullandığımız sanal cihazlardır.
- Bütün Android sürümlerine ait gerçek cihazlara sahip olmak zor olacağından böyle bir sanal cihaz oluşturulmuştur.

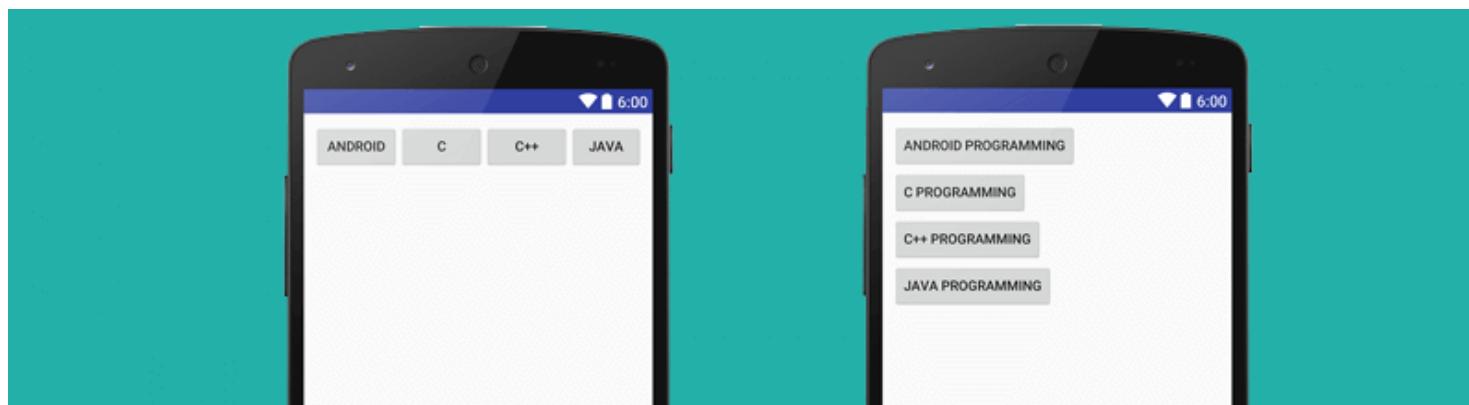


# İlk Uygulamayı Çalıştırma - Proje Dizini

# Android Tasarım

# Layout

- Layout, android uygulamaların her bir ekranında bulunması gereklidir.
- Ayrıca özelleştirilmiş tasarımlarda yapılmaktadır.

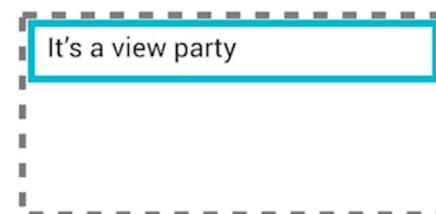


# View Width/Height

width = wrap\_content  
height = wrap\_content



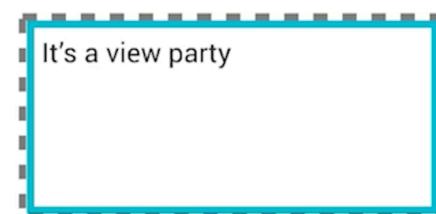
width = match\_parent  
height = wrap\_content



width = wrap\_content  
height = match\_parent



width = match\_parent  
height = match\_parent



# Boyutlandırma Birimleri

# dp – density pixels

- Farklı ekranlarda aynı oranları yakalayabilmek için tasarlanmıştır.
- **Not : Ekran dokunuşlarında yeterli alanı sağlamak için önerilen parmak genişliği 50dp ' dir.**

```
<androidx.cardview.widget.CardView  
    android:layout_width="match_parent"  
    android:layout_height="50dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent">
```

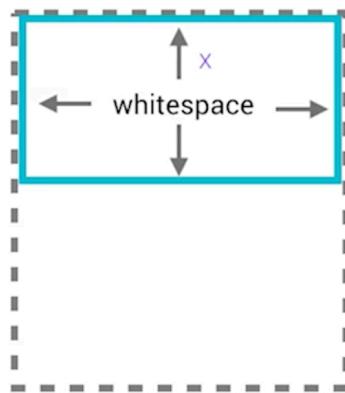
## sp – scale pixels

- Ölçekten bağımsız piksel yani cihaz ekran boyutu değişse de sabit kalan boyutlandırma birimidir.
- Genellikle font boyutu tercihinde kullanılır.
- Ekran boyutu artıkça kullanıcılar daha fazla yazı okumak isterler.

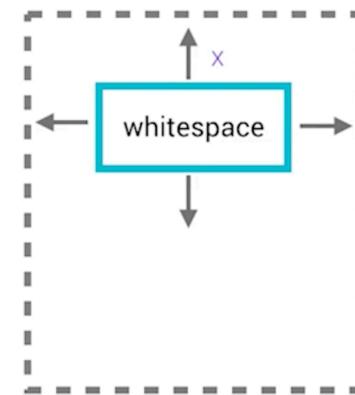
```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="İlkProje"  
    android:textColor="@color/colorPrimary"  
    android:textSize="30sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

# Padding vs Margin

padding = x



layout\_margin = x





10:34 ⓘ

Pizza

## Beef Cheese



Cheese   Sausage   Olive   Pepper

20 min

**Delivery**

Meat lover , get ready to meet your pizza !

**\$ 5.98**

**ADD TO CART**



10:34 ⓘ

Pizza

## Beef Cheese



Cheese   Sausage   Olive   Pepper

20 min

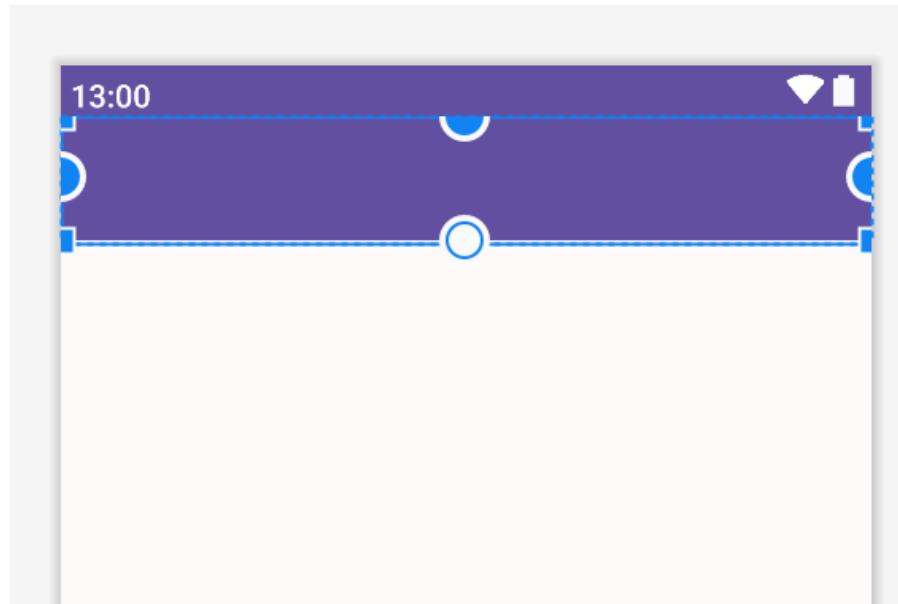
**Delivery**

Meat lover , get ready to meet your pizza !

**\$ 5.98**

**ADD TO CART**

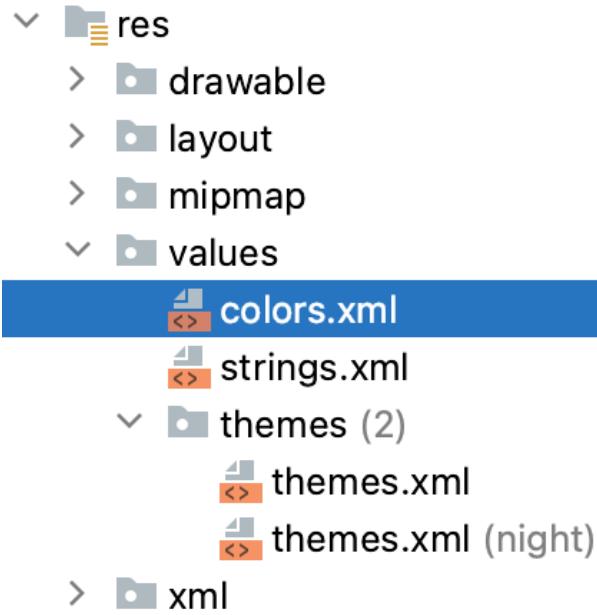
# Toolbar Ekleme



Component Tree

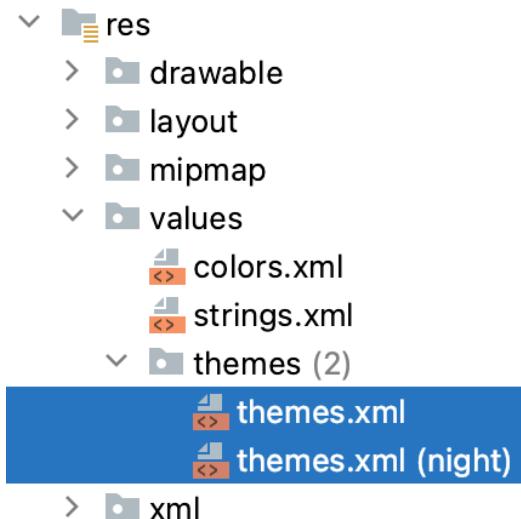
.ConstraintLayout  
.toolbar

# Kullanılacak Renkleri Oluşturma



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <!-- Renkler -->
    <color name="anaRenk">#CB2020</color>
    <color name="anaRenkKoyu">#BF2020</color>
    <color name="yaziRenk1">#FFFFFF</color>
    <color name="yaziRenk2">#636363</color>
</resources>
```

# Status Bar Rengi Değiştirme



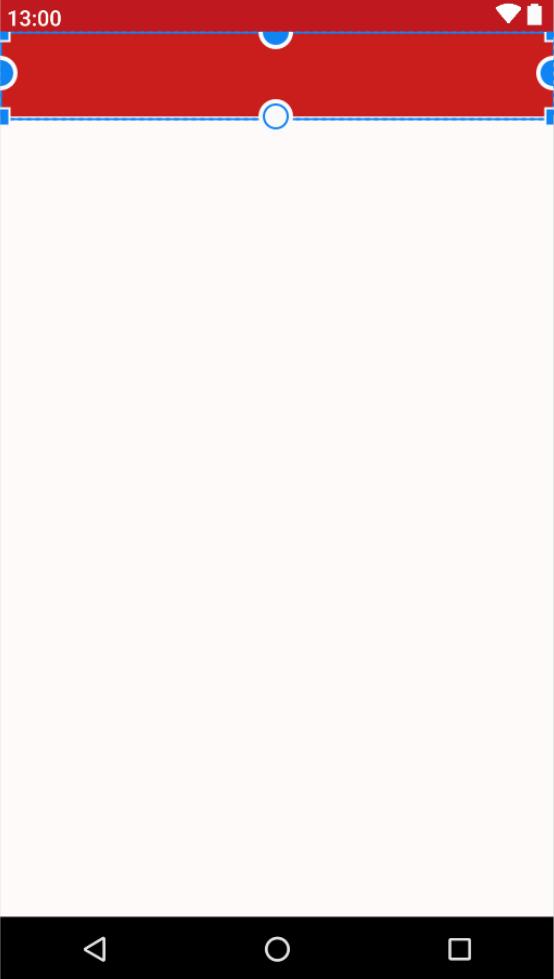
```
<resources xmlns:tools="http://schemas.android.com/tools"> Normal Mode
    <!-- Base application theme. -->
    <style name="Base.Theme.MyApplication" parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
        <item name="android:statusBarColor">@color/anaRenkKoyu</item>
    </style>

    <style name="Theme.MyApplication" parent="Base.Theme.MyApplication" />
</resources>
```

```
<resources xmlns:tools="http://schemas.android.com/tools"> Night Mode
    <!-- Base application theme. -->
    <style name="Base.Theme.MyApplication" parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your dark theme here. -->
        <!-- <item name="colorPrimary">@color/my_dark_primary</item> -->
        <item name="android:statusBarColor">@color/anaRenkKoyu</item>
    </style>
</resources>
```

# Toolbar Rengi Değiştirme



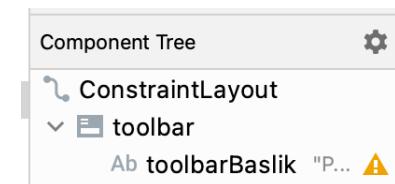
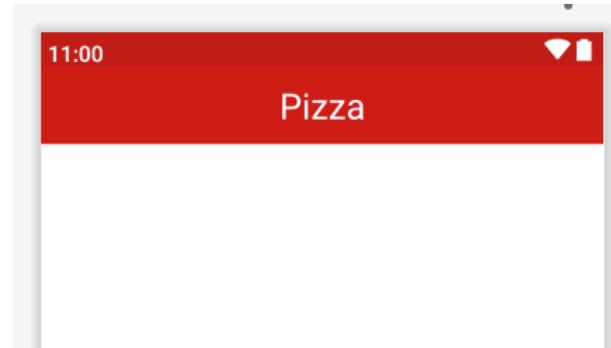
The screenshot shows the Android Studio interface with the layout editor on the left and the properties panel on the right.

**Properties Panel (Right Side):**

- Transformations:
  - y: 0
  - z: 0
- Common Attributes:
  - theme: ?attr/actionBarTheme
  - background: @color/anaRenk (highlighted in pink)
  - navigationIcon:
  - popupTheme:
  - minHeight: ?attr/actionBarSize
  - elevation:
  - alpha:

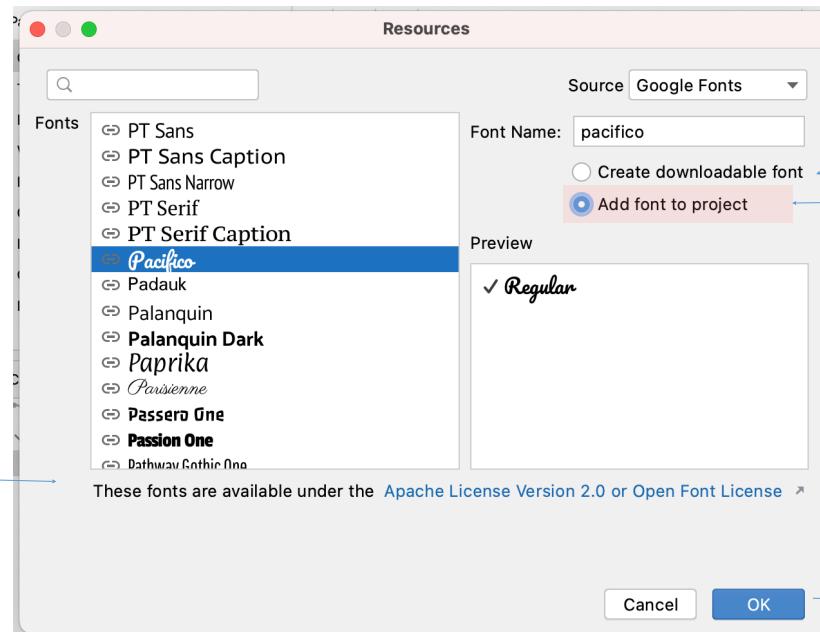
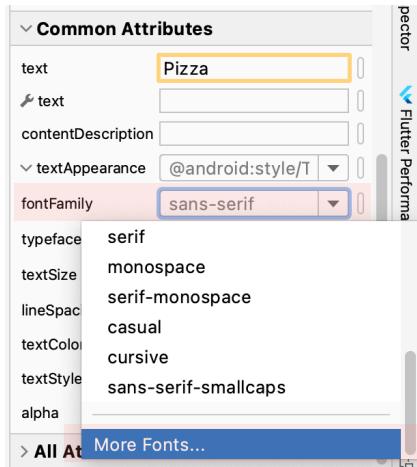
# Toolbar Başlık Ortalama

```
<androidx.appcompat.widget.Toolbar  
    android:id="@+id/toolbar"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:background="?attr/colorPrimary"  
    android:minHeight="?attr/actionBarSize"  
    android:theme="?attr/actionBarTheme"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent">  
  
<TextView  
    android:id="@+id/toolbarBaslik"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  Başlığı ortalar  
    android:text="Pizza"  
    android:textColor="@color/yaziRenk1"  
    android:textSize="26sp"  
    tools:layout_editor_absoluteX="189dp"  
    tools:layout_editor_absoluteY="18dp" />  
  
</androidx.appcompat.widget.Toolbar>
```



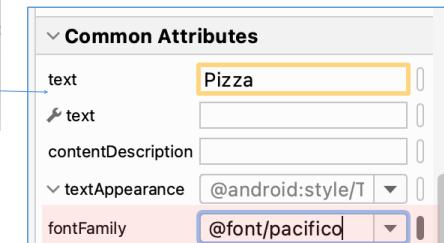
# Toolbar Başlık Font Değişimi

TextView seçildikten sonra



Fontu gerektiğinde internetten  
alarak kullanır.

Fontu projeye indirir.





chipBackgroundColor @color/anaRenk

@color/anaRenk

chipStrokeColor @color/anaRenk

@color/anaRenk

textColor @color/yaziRenk1

*Yatayda ortala , genişlik 0dp*

textSize 22sp

22sp

lineSpacingExtra

16

textColor

@color/yaziRenk2

textSize 44sp

44sp

lineSpacingExtra

16

textColor

@color/anaRenk

textStyle

B I Tr

13:00

Pizza

Beef Cheese



Cheese

Sausage

Olive

Pepper

20 min

Delivery

Meat lover , get ready to meet your  
pizza !

\$ 5.98

ADD TO CART

textSize 36sp

36sp

lineSpacingExtra

16

textColor

@color/anaRenk

B I Tr

textSize 22sp

22sp

lineSpacingExtra

16

textColor

@color/yaziRenk2

B I Tr

textSize 22sp

22sp

lineSpacingExtra

16

textColor

@color/anaRenk

B I Tr

backgroundTint

@color/anaRenk

textSize 18sp

18sp

lineSpacingExtra

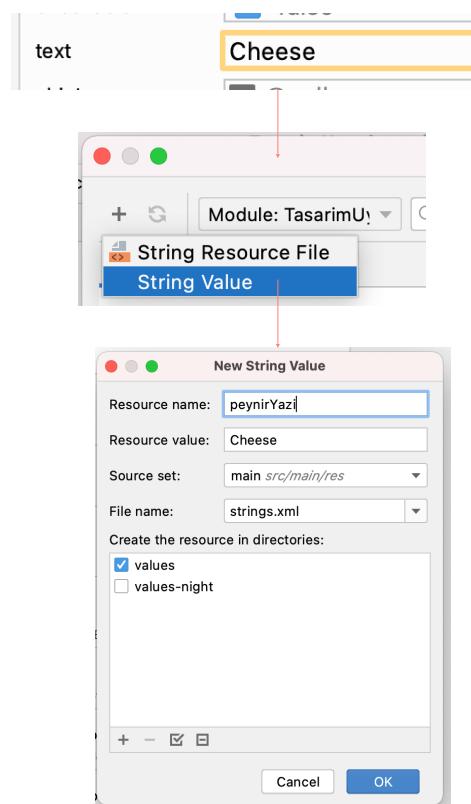
16

textColor

@color/yaziRenk1

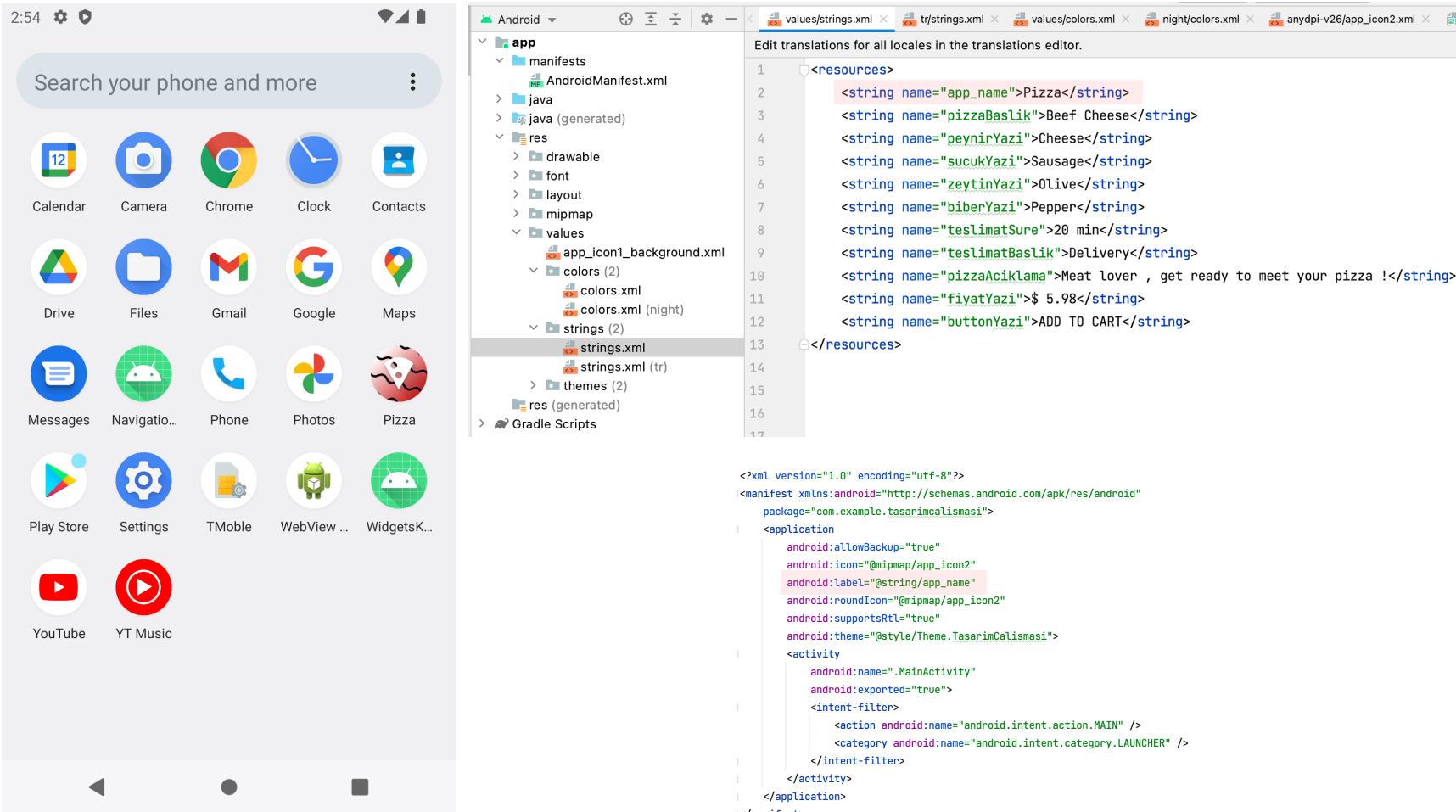
B I Tr

# String Dosyası İçeriğinin Oluşturulması

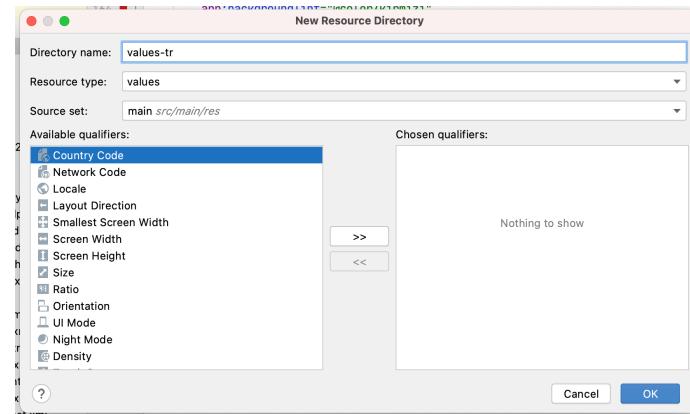
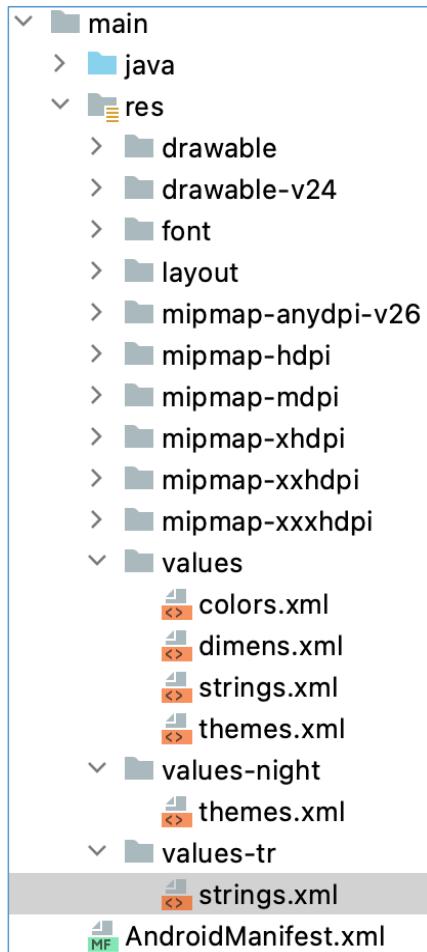


```
<resources>
    <string name="app_name">TasarimUygulaması</string>
    <string name="pizzaBaslik">Beef Cheese</string>
    <string name="peynirYazi">Cheese</string>
    <string name="sucukYazi">Sausage</string>
    <string name="zeytinYazi">Olive</string>
    <string name="biberYazi">Pepper</string>
    <string name="teslimatSure">20 min</string>
    <string name="teslimatBaslik">Delivery</string>
    <string name="pizzaAciklama">"Meat lover , get ready to meet your pizza ! "</string>
    <string name="fiyatYazi">$ 5.98</string>
    <string name="buttonYazi">ADD TO CHART</string>
</resources>
```

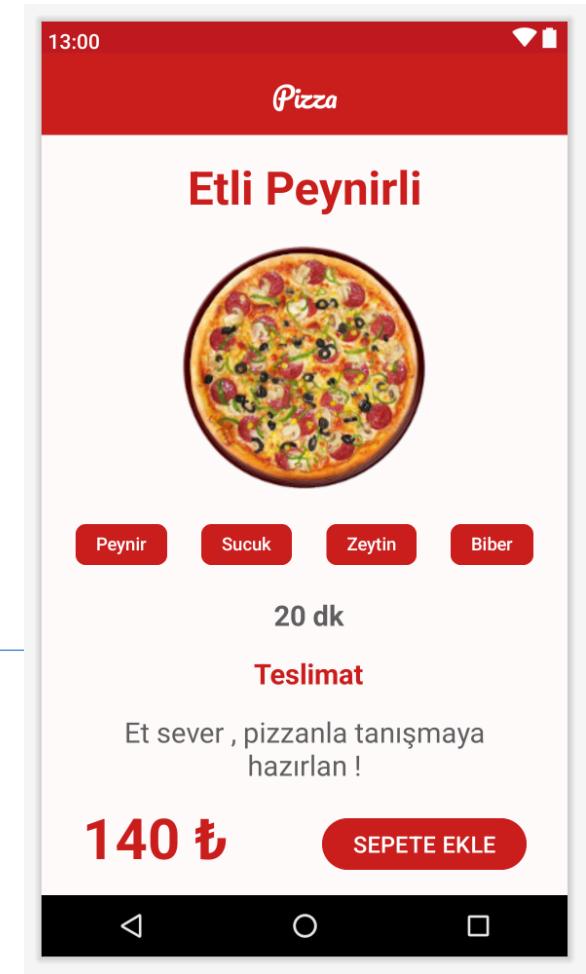
# Uygulama Adı Değiştirme



# Çoklu Dil Desteği



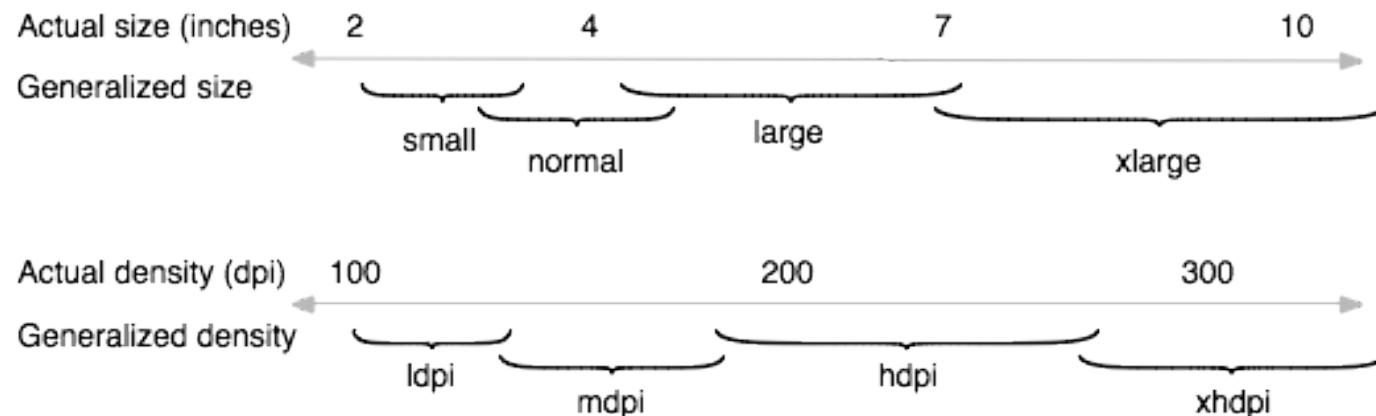
```
<resources>
    <string name="app_name">TasarimUygulaması</string>
    <string name="pizzaBaslik">Etli Peynirli</string>
    <string name="peynirYazi">Peynir</string>
    <string name="sucukYazi">Sucuk</string>
    <string name="zeytinYazi">Zeytin</string>
    <string name="biberYazi">Biber</string>
    <string name="teslimatSure">20 dk</string>
    <string name="teslimatBaslik">Teslimat</string>
    <string name="pizzaAciklama">"Et sever , pizzanla tanışmaya hazırlan ! "</string>
    <string name="fiyatYazi">48.99 ₺</string>
    <string name="buttonYazi">SEPETE EKLE</string>
</resources>
```



# Çoklu Ekran Desteği

# Çoklu Ekran Desteği

- Android eko sistemi farklı ekran boyutlarına sahip cihazlardan oluşmaktadır.
- Bir uygulama tüm ekranlara göre tasarlanmalıdır.

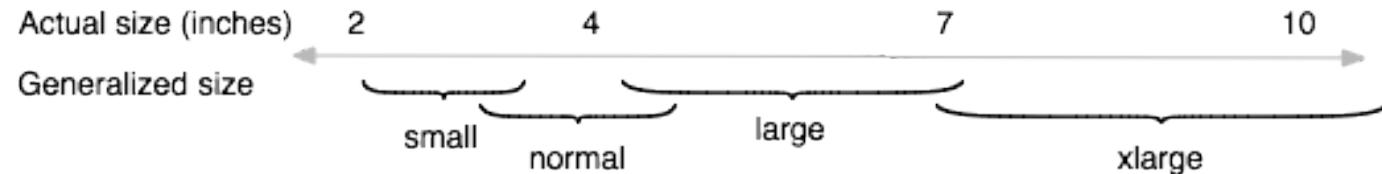


dpi : her bir inch teki piksel yoğunluğu

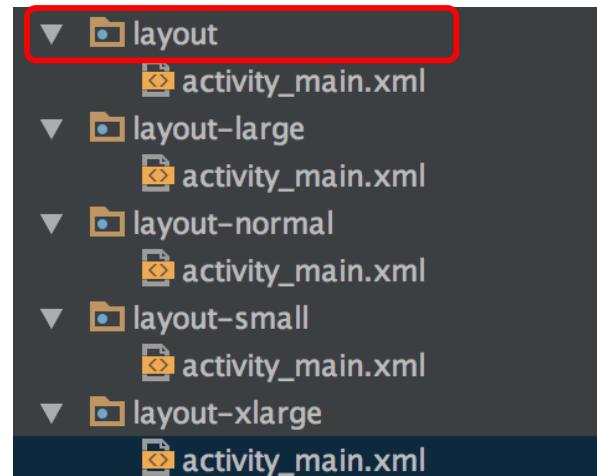
Kasım ADALAN

94

# Ekran boyutuna göre sınıflandırma



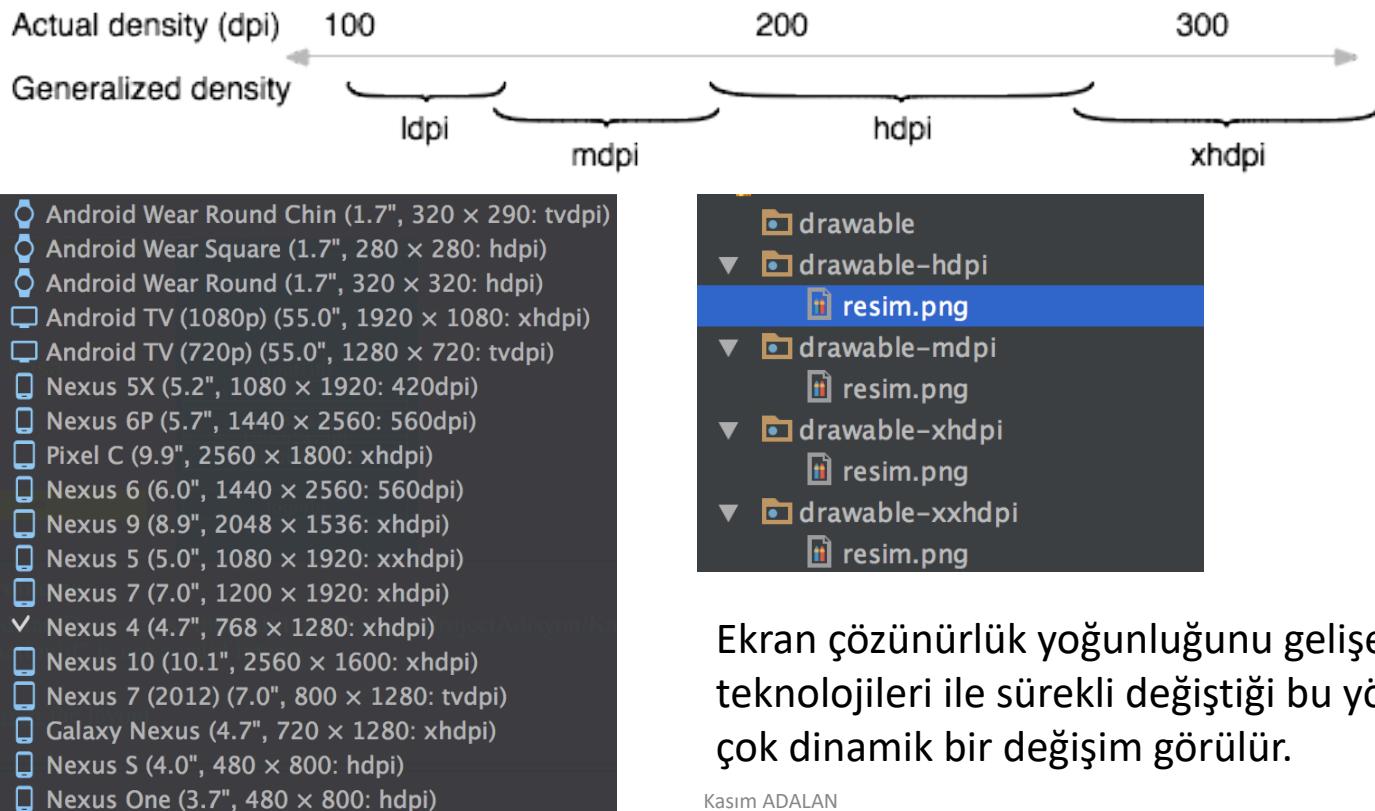
- ✗ Android Wear Round Chin → small/activitymain.xml
- ✗ Android Wear Square → small/activitymain.xml
- ✗ Android Wear Round → small/activitymain.xml
- ✗ Android TV (1080p) → xlarge/activitymain.xml
- ✗ Android TV (720p) → xlarge/activitymain.xml
- ✗ Nexus 5X → normal/activitymain.xml
- ✗ Nexus 6P → normal/activitymain.xml
- ✗ Pixel C → xlarge/activitymain.xml
- ✗ Nexus 6 → normal/activitymain.xml
- ✗ Nexus 9 → xlarge/activitymain.xml
- ✓ Nexus 5 → normal/activitymain.xml
- ✗ Nexus 7 → large/activitymain.xml
- ✗ Nexus 4 → normal/activitymain.xml
- ✗ Nexus 10 → xlarge/activitymain.xml
- ✗ Nexus 7 (2012) → large/activitymain.xml
- ✗ Galaxy Nexus → normal/activitymain.xml
- ✗ Nexus S → normal/activitymain.xml
- ✗ Nexus One → normal/activitymain.xml



Not : Ek almayan dosyamızı silmeyin varsayılan olarak durması sistemin tanımlanamayan durumlarda çalışmasına yardım eder.

Ekran boyutları daha evrensel olduğu için yaygın kullanım bu yöntemdir.

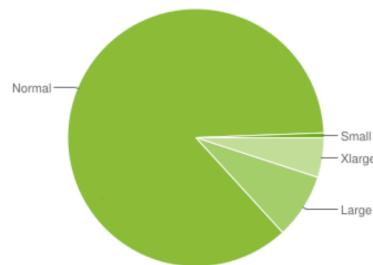
# Ekran çözünürlük yoğunluğuna ( dpi ) göre sınıflandırma



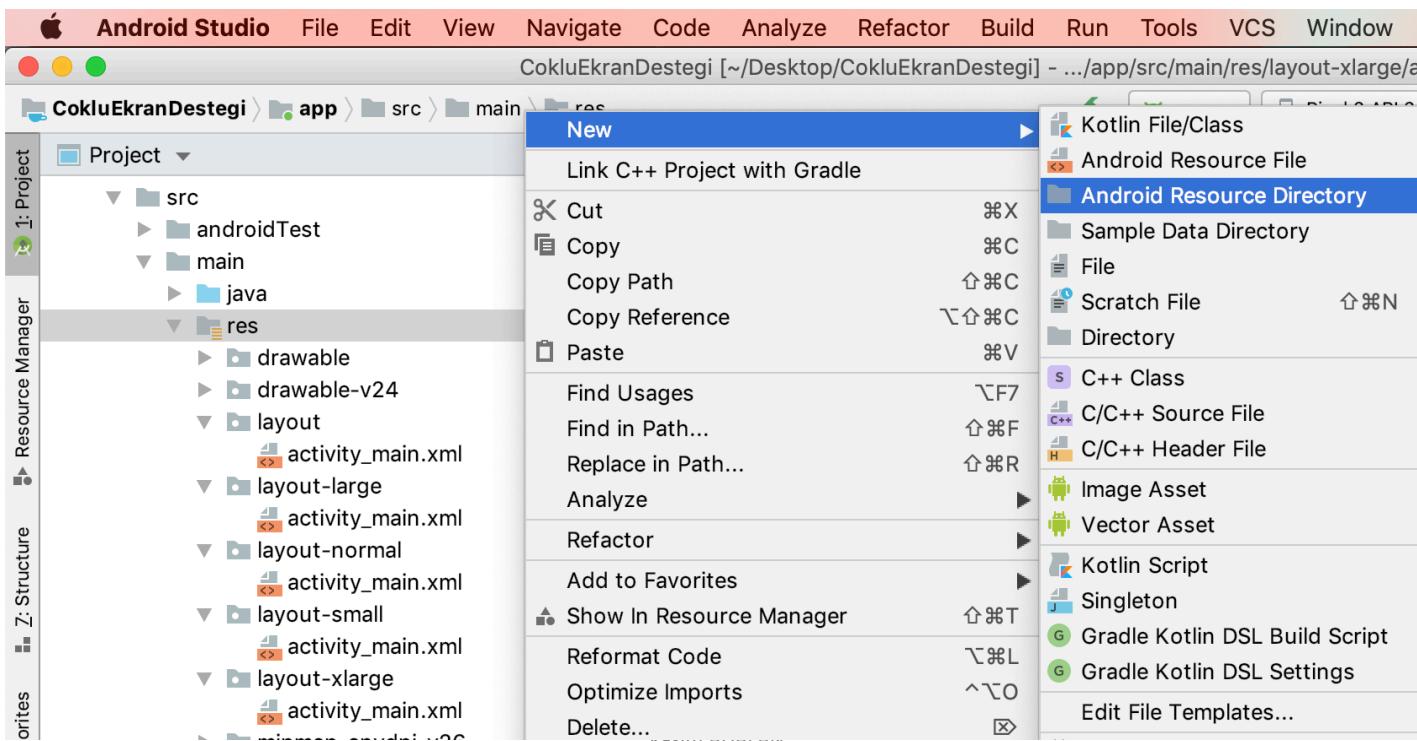
Ekran çözünürlük yoğunluğunu gelişen ekran teknolojileri ile sürekli değiştiği bu yöntemde çok dinamik bir değişim görülür.

# Ekran Boyutu Kullanım Dağılımı

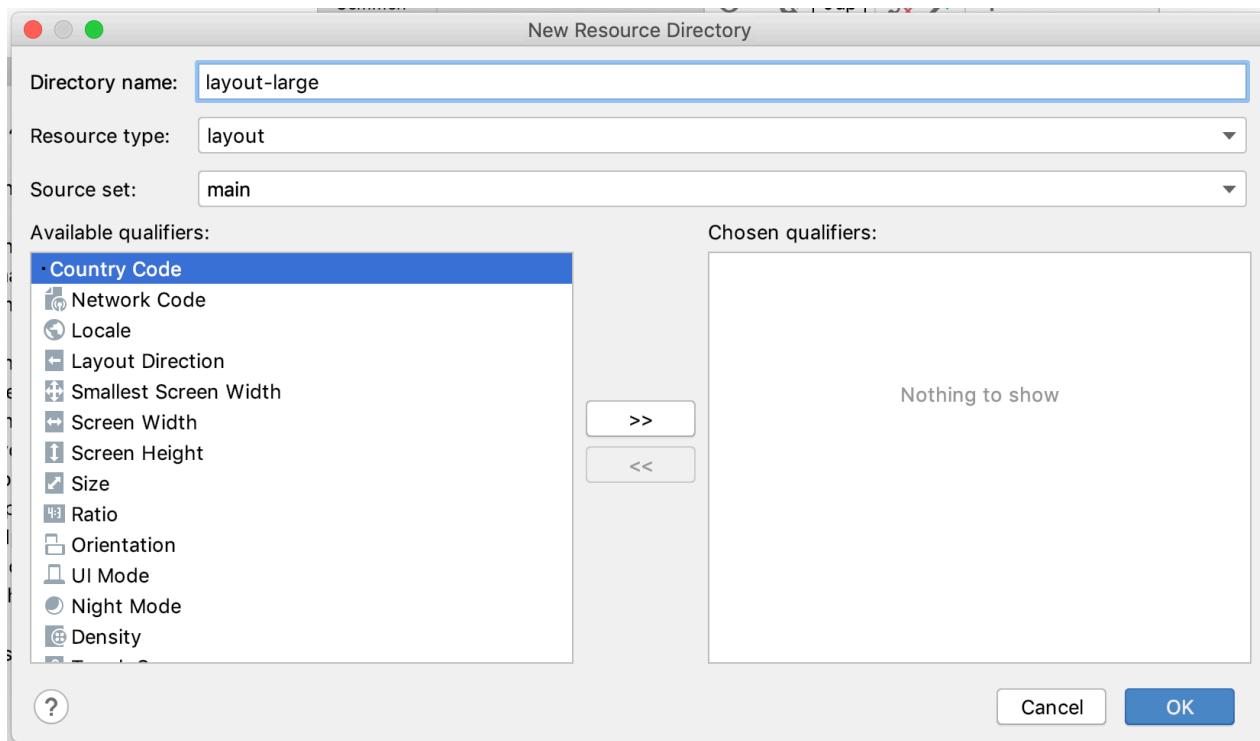
|        | ldpi | mdpi | tvdpi | hdpi  | xhdpi | xxhdpi | Total |
|--------|------|------|-------|-------|-------|--------|-------|
| Small  | 0.4% |      |       |       | 0.1%  | 0.1%   | 0.6%  |
| Normal |      | 0.9% | 0.3%  | 24.0% | 37.7% | 23.6%  | 86.5% |
| Large  |      | 2.4% | 1.9%  | 0.6%  | 1.6%  | 1.7%   | 8.2%  |
| Xlarge |      | 3.1% |       | 1.3%  | 0.6%  |        | 5.0%  |
| Total  | 0.4% | 6.4% | 2.2%  | 25.9% | 40.0% | 25.4%  |       |



# Çoklu Ekran Desteği Dosyaları oluşturma

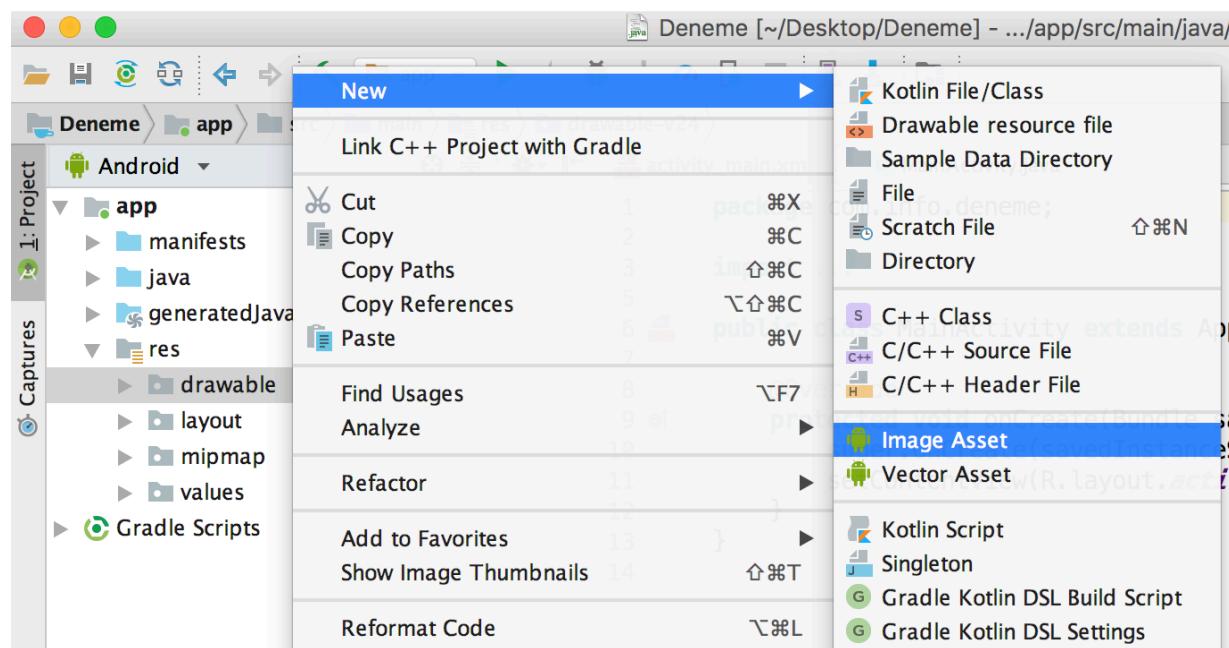


# Çoklu Ekran Desteği Dosyaları oluşturma

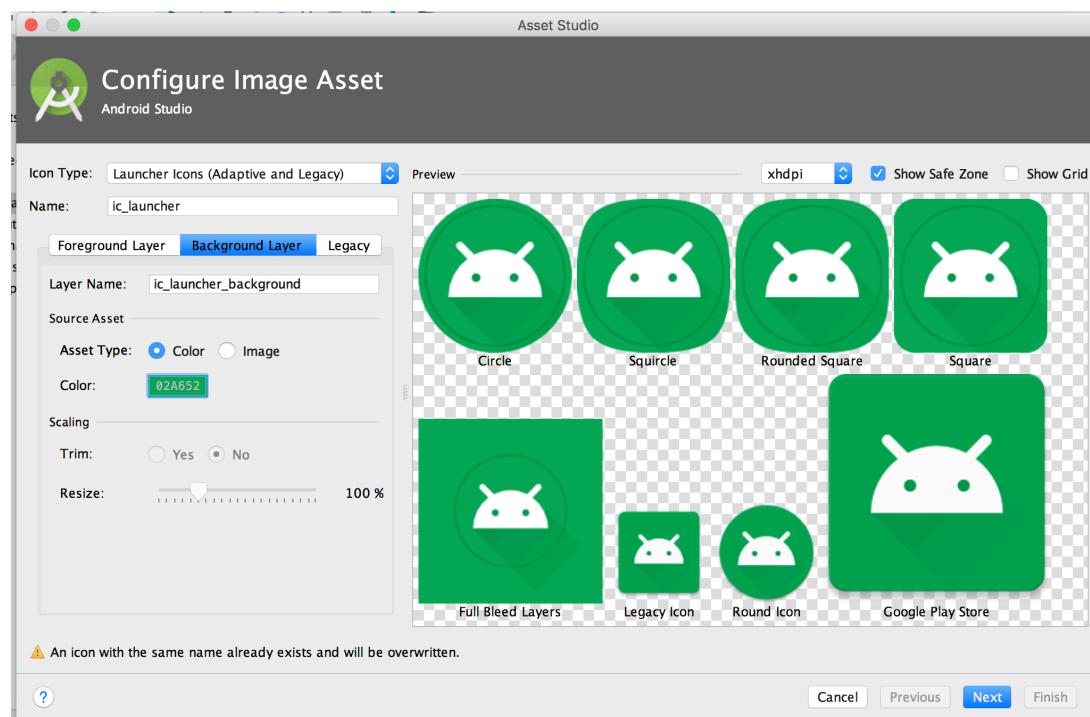


# Icon Oluşturma

# Icon Oluşturma



# Icon Oluşturma



# Dark Mode



10:34 ⓘ

Pizza

## Beef Cheese



Cheese   Sausage   Olive   Pepper

20 min

**Delivery**

Meat lover , get ready to meet your pizza !

**\$ 5.98**

**ADD TO CART**



10:34 ⓘ

Pizza

## Beef Cheese



Cheese   Sausage   Olive   Pepper

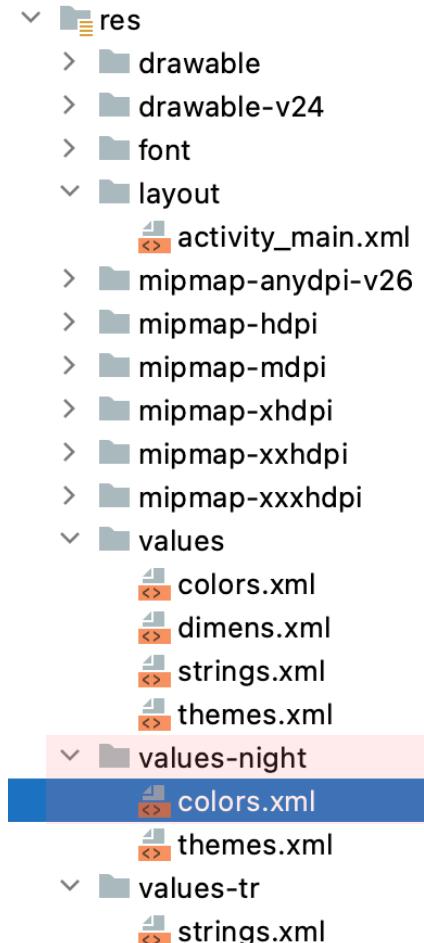
20 min

**Delivery**

Meat lover , get ready to meet your pizza !

**\$ 5.98**

**ADD TO CART**



Color dosyasını kopyala

Gece moduna uygun renkler oluştur.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
    <!-- Renklerimiz -->
    <color name="anaRenk">#EF6C00</color>
    <color name="koyuAnaRenk">#E65100</color>
    <color name="yaziRenk1">#FFFFFF</color>
    <color name="yaziRenk2">#9F9F9F</color>
</resources>
```

# Çalışma Yapısı

# Sayfa Oluşturma

*MainActivity*

1:06 ☰ ▶◀

Anasafya

Detay

*DetayActivity*

1:11 ☰ ▶◀

Merhaba

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3

*Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.*

# Görsel Nesne Bağlama

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        binding.buttonDetay.setOnClickListener { it: View!  
            val intent = Intent(packageContext: this@MainActivity, DetayActivity::class.java)  
            startActivity(intent)  
        }  
    }  
}  
  
class DetayActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityDetayBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityDetayBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        binding.textViewBilgi.text = "Merhaba"  
    }  
}
```

# Sayfalar Arası Geçiş

*MainActivity*

1:06 ⓘ

```
binding.buttonDetay.setOnClickListener { it: View!  
    val intent = Intent(packageContext: this@MainActivity, DetayActivity::class.java)  
    startActivity(intent)  
}
```

Anasafya

*DetayActivity*

1:11 ⓘ

Merhaba

Geri tuşu veya  
kaydırma ile bir  
önceki sayfaya  
dönüş

Detay

# Activity'ler arası veri taşıma

- String, Int , Double etc. gönderilebilir.
- Bir sınıfından oluşturulmuş bir nesnede gönderilebilir.

## Verinin Gönderilmesi

```
val yeniIntent = Intent(packageContext: this@MainActivity, ActivityB::class.java)  
yeniIntent.putExtra(name: "mesaj", value: "merhaba")  
startActivity(yeniIntent)
```

## Verinin Alınması

key

```
val gelenMesaj = intent.getStringExtra(name: "mesaj")
```

## Veri Gönderimi

```
binding.buttonDetay.setOnClickListener { it: View!  
    val intent = Intent(packageContext: this@MainActivity, DetayActivity::class.java)  
    intent.putExtra(name: "ad", value: "Ahmet")  
    intent.putExtra(name: "yas", value: 23)  
    intent.putExtra(name: "boy", value: 1.78)  
    intent.putExtra(name: "bekar", value: true)  
    startActivity(intent)  
}
```

## Veri Alımı

```
val gelenAd = intent.getStringExtra(name: "ad")  
val gelenYas = intent.getIntExtra(name: "yas", defaultValue: 0)  
val gelenBoy = intent.getDoubleExtra(name: "boy", defaultValue: 0.0)  
val gelenBekar = intent.getBooleanExtra(name: "bekar", defaultValue: false)  
  
binding.textViewBilgi.text = "$gelenAd - $gelenYas - $gelenBoy - $gelenBekar"
```

# Navigation Component

# Navigation Component

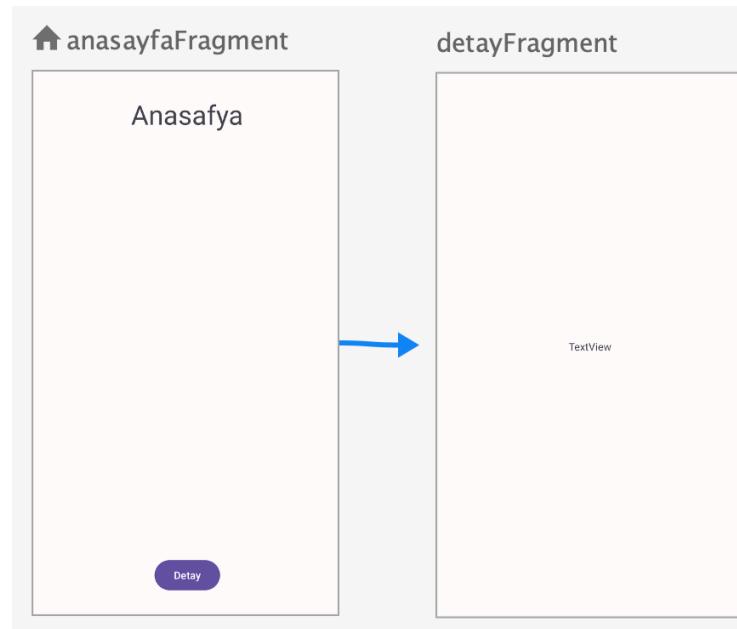
Sayfa geçişlerini daha görsel ve pratik hale getiren bir yapıdır.

Activity üzerinde istediğimiz şekilde fragment geçişleri yapabiliriz.

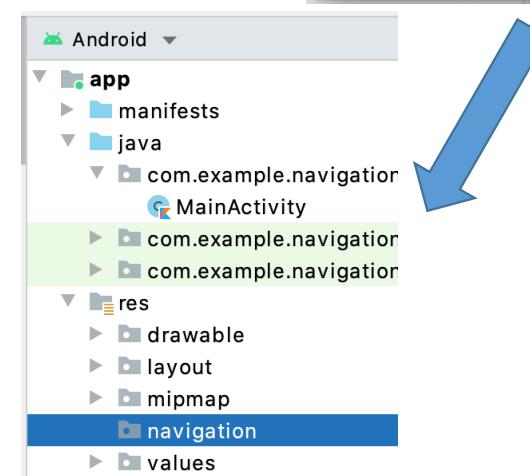
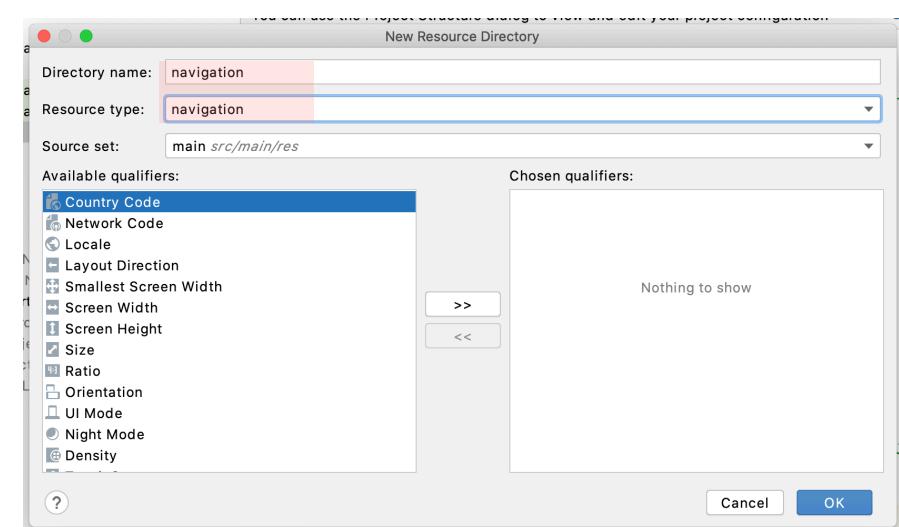
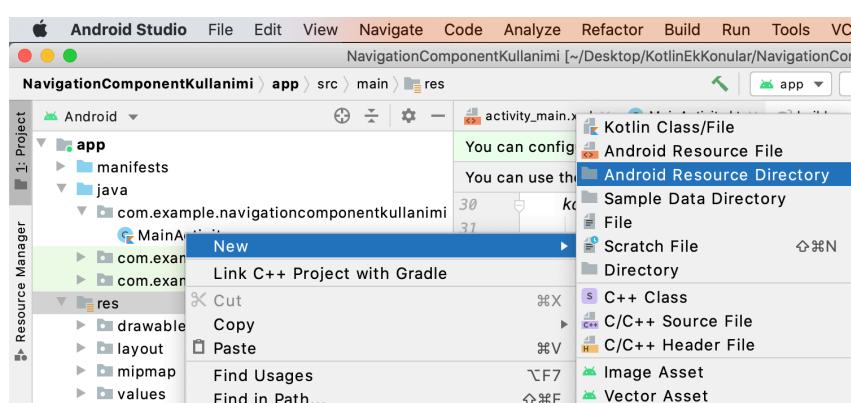
Geçişlerde veri transferi yapabiliriz.

Geçişlere animasyon ekleyebiliriz.

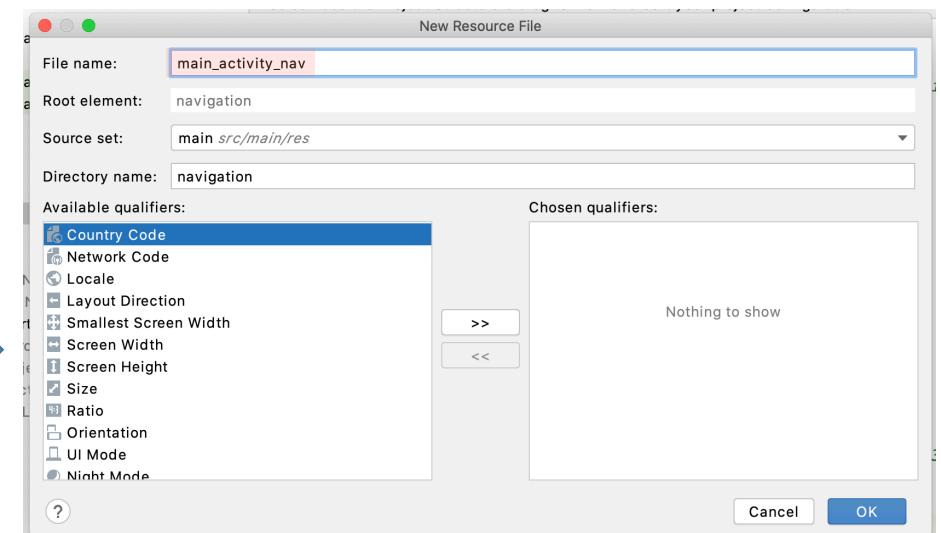
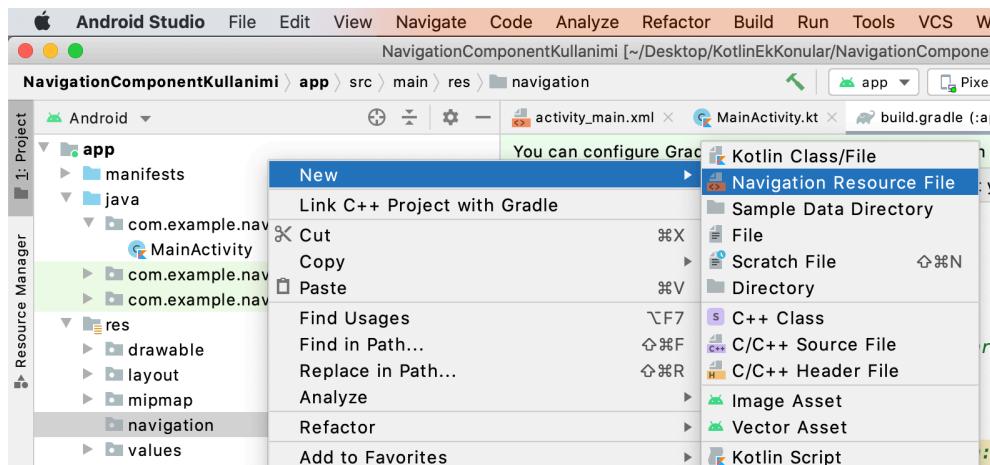
Birden fazla activityde kullanılabilir ama her activity için ayrı navigation dosyası oluşturmalıyız.



# Navigation Klasörü Oluşturma



# Navigation Dosyası Oluşturma



Eğer birden fazla activity için navigation dosyası gerekliyse , her activity için ayrı ayrı dosya oluşturulur.

# Uyarı

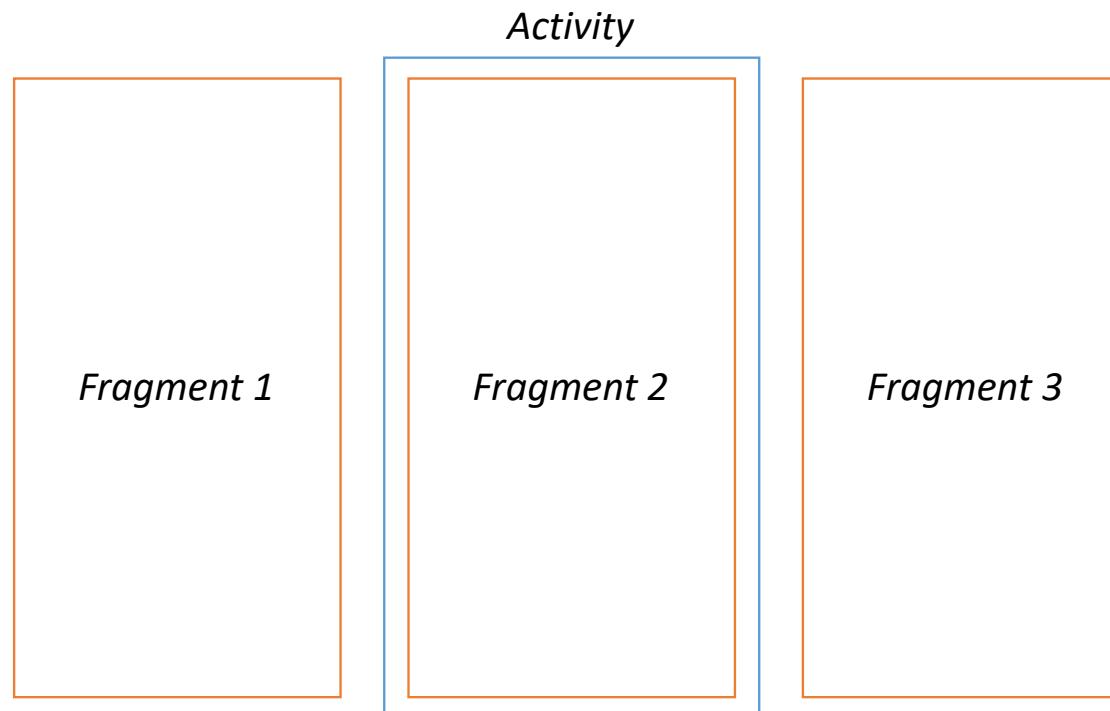
- Navigation Component kütüphane sürümü ile compile sdk uyumlu değilse kütüphane sürümünü düşürebilirsiniz.

```
dependencies { this: DependencyHandlerScope

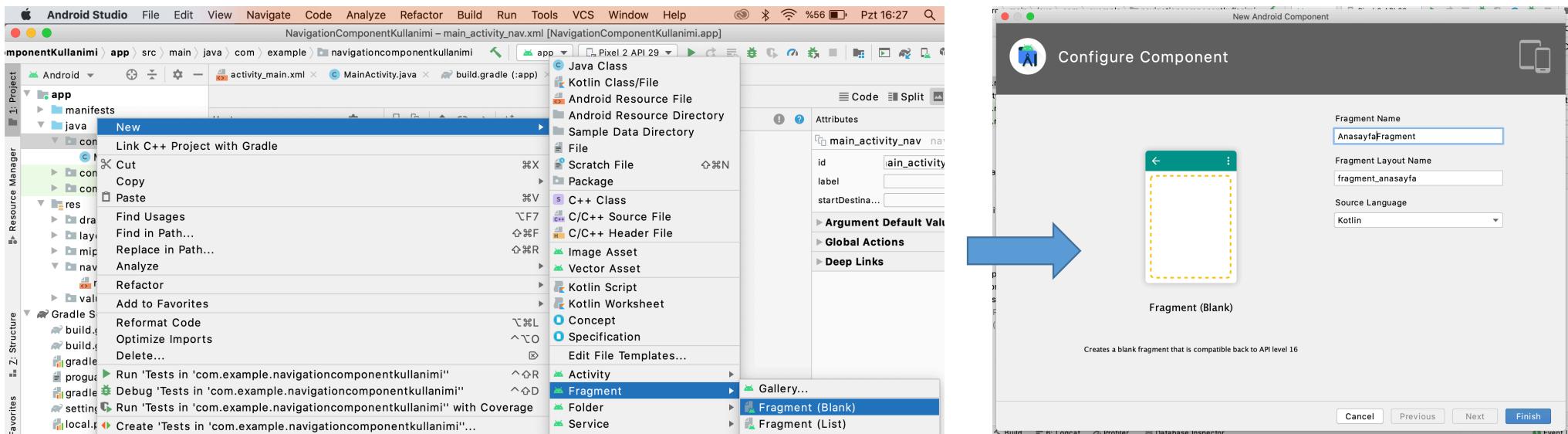
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.navigation:navigation-fragment:2.5.0")
    implementation("androidx.navigation:navigation-ui:2.5.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

# Activity & Fragment İlişkisi

- Fragmentlar Activity olmadan çalışmazlar.
- Bir Activity üzerinde birden fazla Fragment çalışabilir.
- Fragmentlar Activity tasarımları içinde çalışırlar ve görüntülenirler.



# Fragment Oluşturma



```
class AnasayfaFragment : Fragment() {  
  
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?  
        , savedInstanceState: Bundle?): View? {  
  
        return inflater.inflate(R.layout.fragment_anasayfa, container, attachToRoot: false)  
    }  
}
```

# ViewBinding Kullanımı : Fragment

Build.gradle/module

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

fragment\_blank.xml

Component Tree

- ConstraintLayout
  - button "Button"

1

```
class BlankFragment : Fragment() {
```

```
    private lateinit var tasarim: FragmentBlankBinding
```

```
    override fun onCreateView(inflater: LayoutInflater,  
                             container: ViewGroup?, savedInstanceState: Bundle?): View? {
```

```
        tasarim = FragmentBlankBinding.inflate(inflater, container, attachToParent: false)
```

```
        tasarim.buttonny.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }
```

```
        return tasarim.root
```

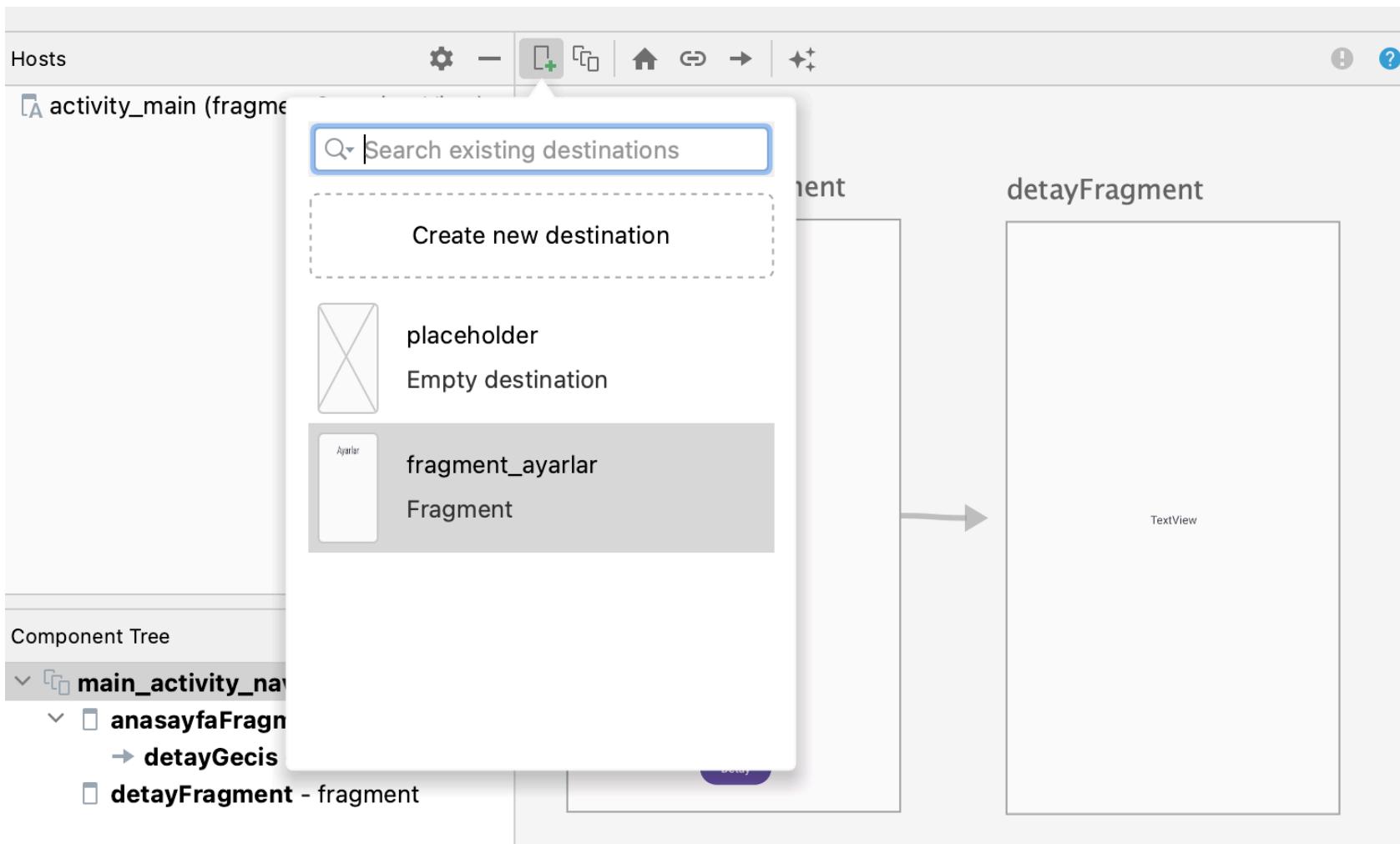
}

2

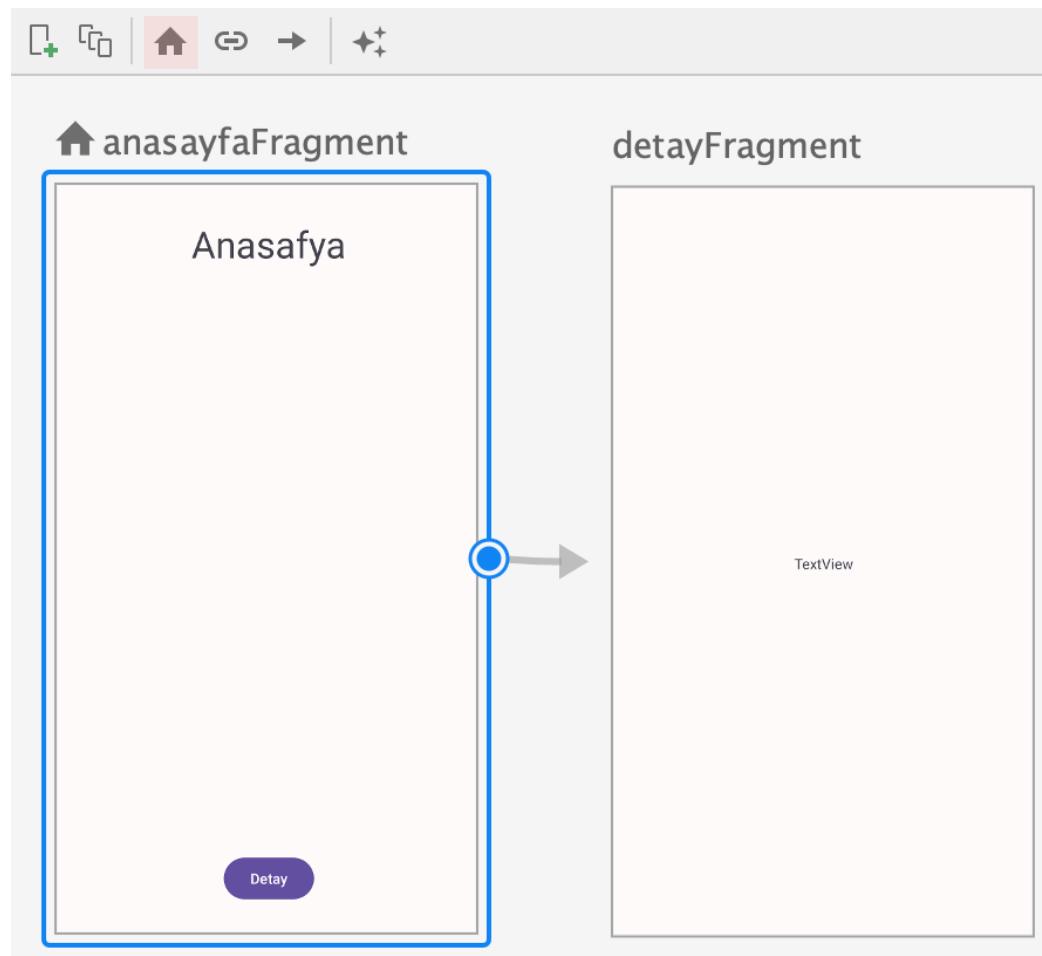
3

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

# Navigation Dosyasına Fragment Ekleme



# Anasayfanın Değiştirilmesi



# Sayfalar Arası Geçiş

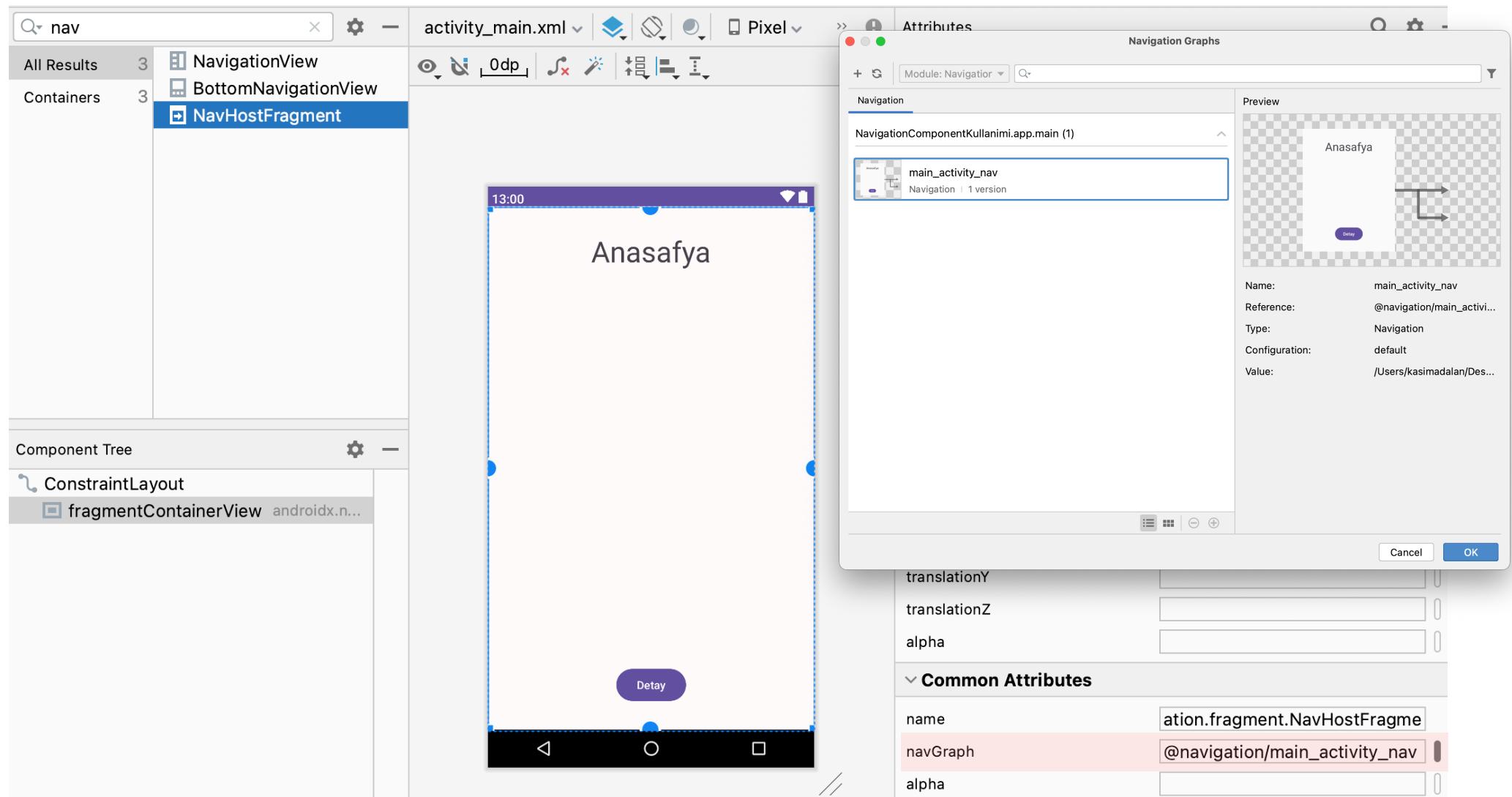
The screenshot shows the Android Studio Layout Editor with the following components:

- Hosts:** activity\_main (fragmentContainerView)
- Component Tree:**
  - main\_activity\_nav - navigation
    - anasayfaFragment - fragment
    - detayGecis - action
  - detayFragment - fragment
  - ayarlarFragment - fragment
- Layout Preview:** Displays two fragments: anasayfaFragment (containing "Anasafya" and a "Detay" button) and detayFragment (containing a "TextView"). A blue arrow indicates the transition from anasayfaFragment to detayFragment.
- Attributes (for detayGecis action):**
  - action: detayGecis
  - id: detayGecis
  - destination: detayFragment
  - Animations:** (Empty fields)
  - Argument Default Values:**

|       |               |               |
|-------|---------------|---------------|
| ad    | string        | default value |
| yas   | integer       | default value |
| boy   | float         | default value |
| bekar | boolean       | default value |
| urun  | com.example.n | default value |
  - Pop Behavior:** (Empty field)
- Java Code (in binding file):**

```
binding.buttonDetay.setOnClickListener { it: View!
    Navigation.findNavController(it).navigate(R.id.detayGecis)
}
```

# Navigation Dosyasını Activity Tasarımına Ekleme



# Uyarı

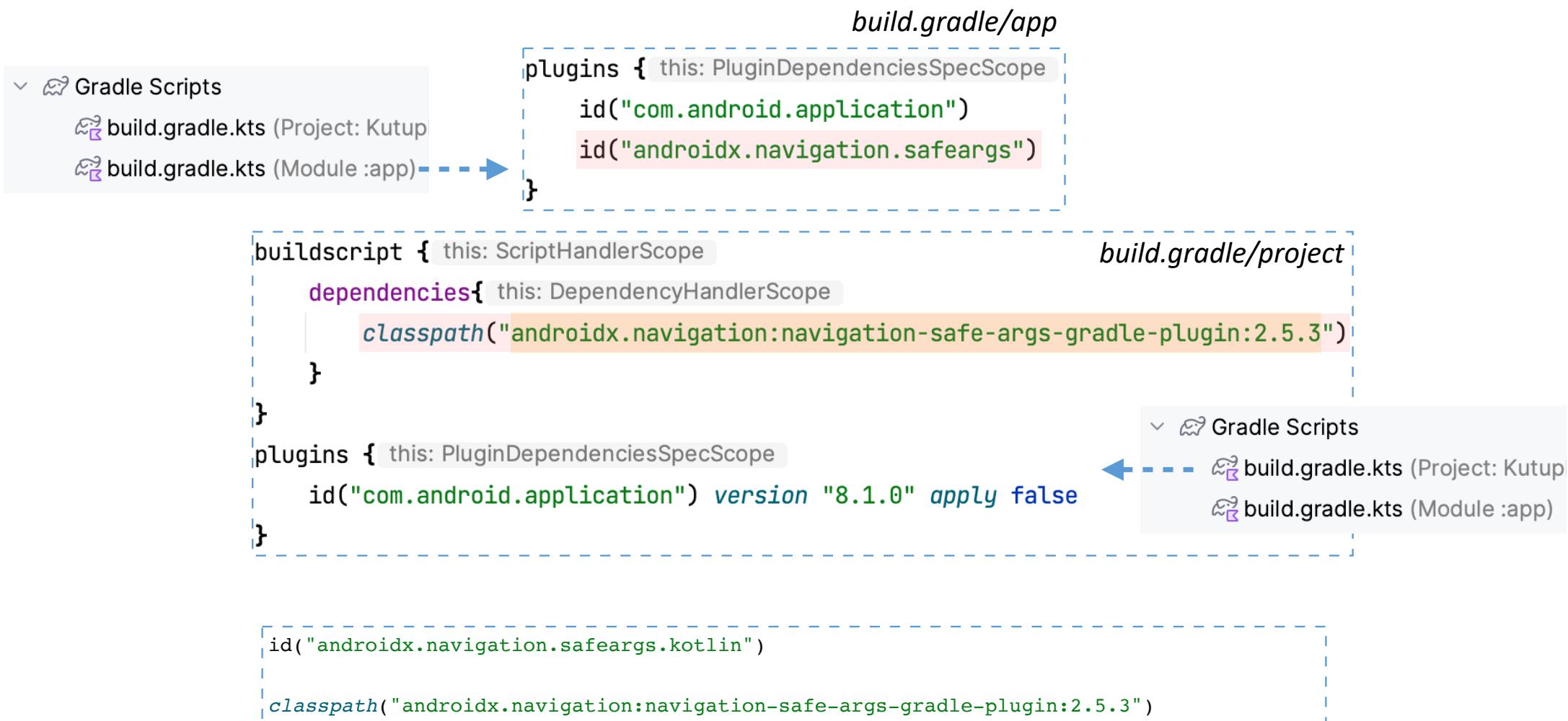
- Navigation Component kütüphane sürümü ile compile sdk uyumlu değilse kütüphane sürümünü düşürebilirsiniz.

```
dependencies { this: DependencyHandlerScope

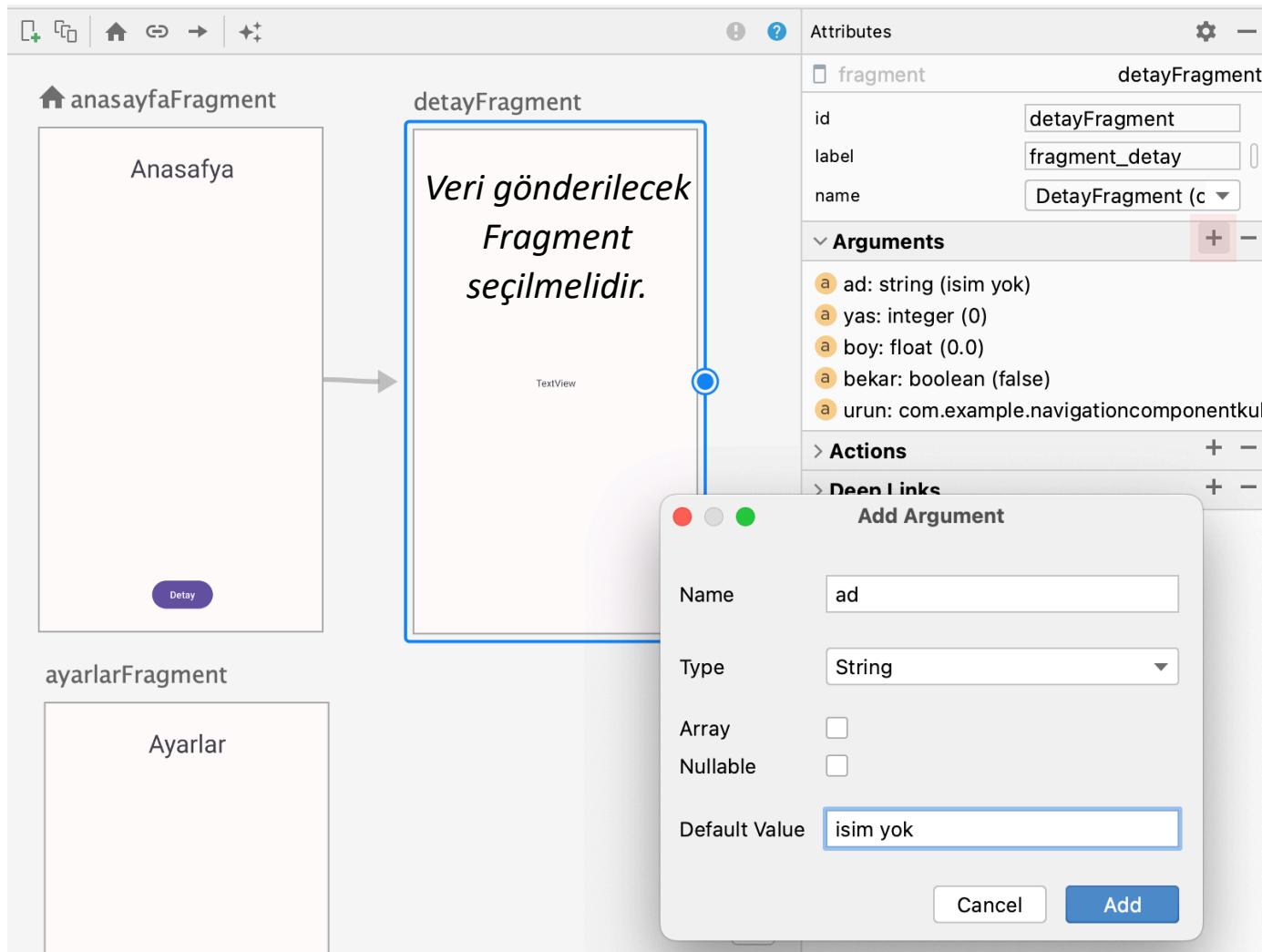
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.navigation:navigation-fragment:2.5.0")
    implementation("androidx.navigation:navigation-ui:2.5.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

# Sayfalar Arası Veri Transferi

# Kurulum



# Gönderilecek Veri Oluşturma



## Veri Gönderimi

```
binding.buttonDetay.setOnClickListener { it: View!  
    val urun = Urunler( id: 100, ad: "TV")  
    val gecis = AnasayfaFragmentDirections  
        .detayGecis(urun = urun,ad = "Ahmet",yas = 23,boy = 1.78f,bekar = true)  
    Navigation.findNavController(it).navigate(gecis)  
}
```

```
data class Urunler(var id:Int,  
                    var ad:String) : Serializable
```

## Veri Alımı

```
val bundle:DetayFragmentArgs by navArgs()  
  
val gelenAd = bundle.ad  
val gelenYas = bundle.yas  
val gelenBoy = bundle.boy  
val gelenBekar = bundle.bekar  
  
val urunId = bundle.urun.id  
val urunAd = bundle.urun.ad  
  
binding.textViewBilgi.text = "$gelenAd - $gelenYas - $gelenBoy - $gelenBekar - $urunId - $urunAd"
```

# Yaşam Döngüsü

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    Log.e( tag: "Yaşam Döngüsü", msg: "onCreate")  
    //Uygulama çalıştırıldığında bir kere çalışır.  
}  
  
override fun onPause() {  
    super.onPause()  
    Log.e( tag: "Yaşam Döngüsü", msg: "onPause")  
    //Sayfa görünmez oldu  
}  
  
override fun onResume() {  
    super.onResume()  
    Log.e( tag: "Yaşam Döngüsü", msg: "onResume")  
    //Sayfa göründü  
}
```

# Geri Tuşu Kullanımı

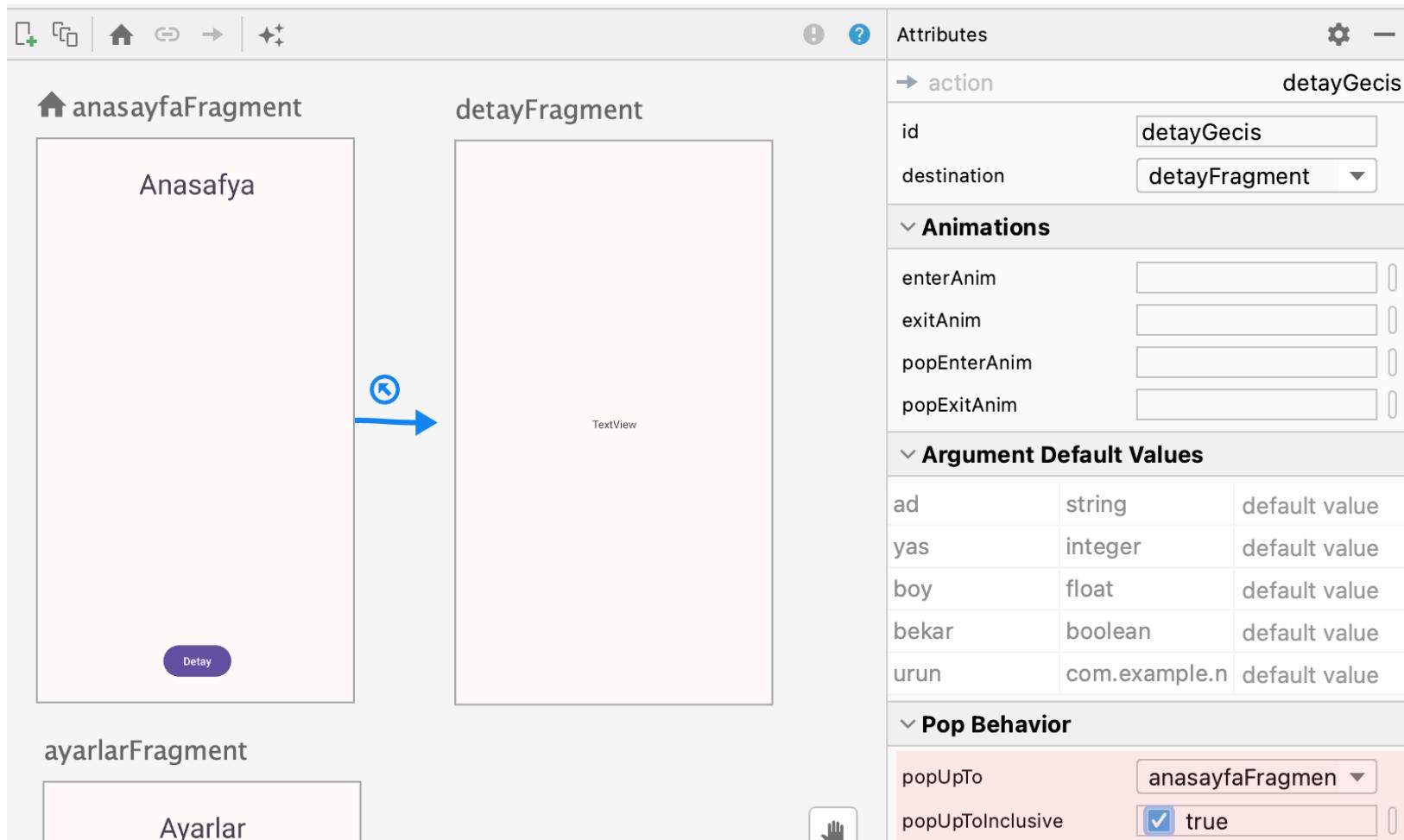
```
binding.textViewBilgi.text = "$gelenAd - $gelenYas - $gelenBoy - $gelenBekar - $urunId - $urunAd"

val backPress = object : OnBackPressedCallback(enabled: true) { //Geri dönüş aktif değil
    override fun handleOnBackPressed() {
        Snackbar.make(binding.textViewBilgi, text: "Geri dönmek istiyor musunuz?", Snackbar.LENGTH_SHORT)
            .setAction(text: "EVET") { it: View!
                isEnabled = false //Geri dönüş aktif
                requireActivity().onBackPressedDispatcher.onBackPressed() // Geri dönüş kodlaması
            }.show()
    }
}
requireActivity().onBackPressedDispatcher.addCallback(viewLifecycleOwner, backPress)

return binding.root
```

# Back Stack Kullanımı

- Geri tuşu ile bir önceki sayfaya dönmemizi sağlayan yapıdır.



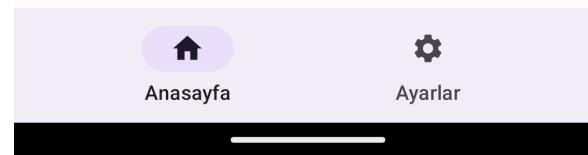
# Bottom Navigation Kullanımı

# Bottom Navigation

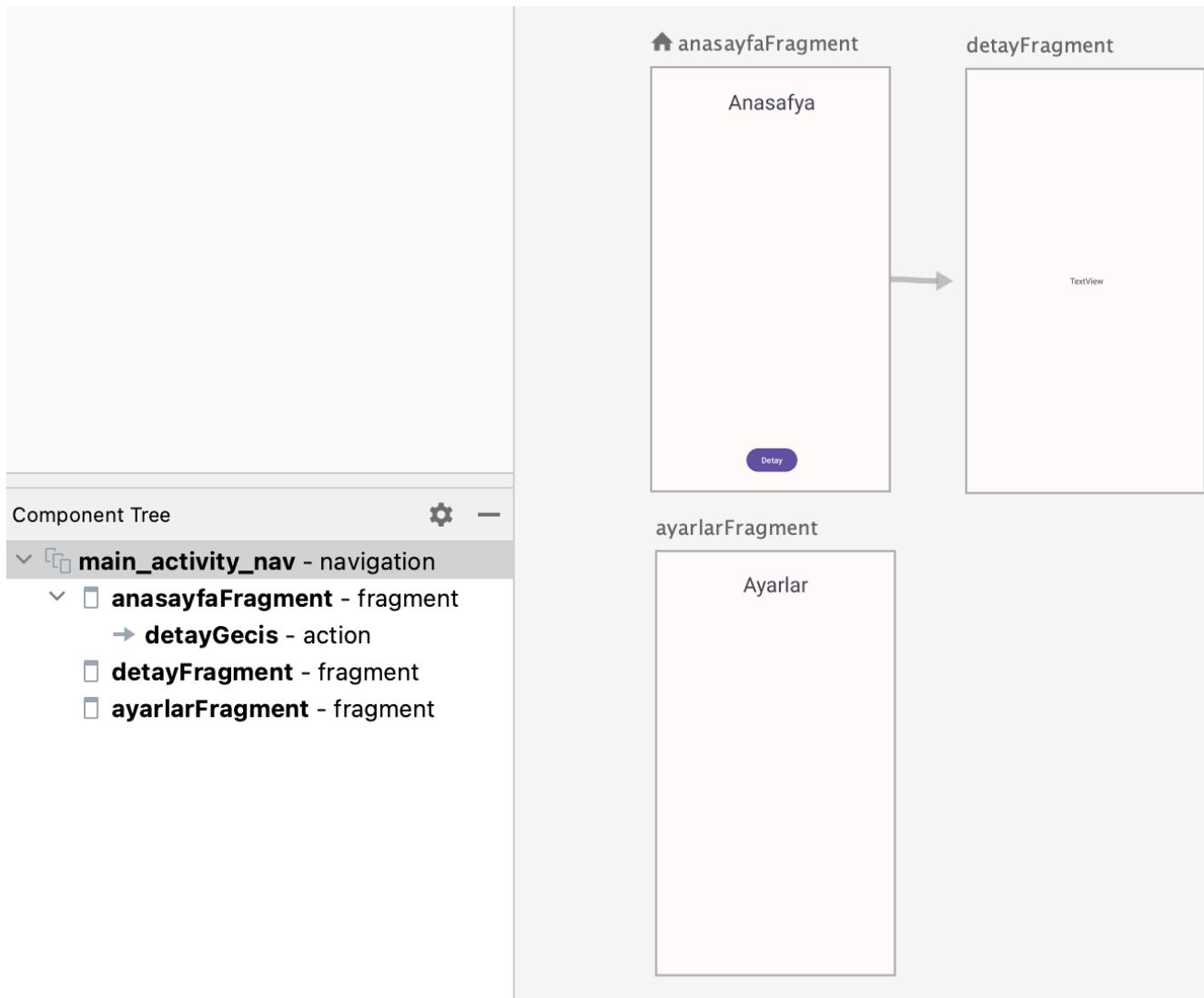


Anasafya

Detay



# Fragment Oluşturma



# Bottom Navigation için Menu Oluşturma

- Menu item idleri ile Fragment idleri aynı olmalıdır.
- Hangi item tıklanıldığı ve açılacak Fragment bu şekilde belirlenir.

The screenshot shows the Android Studio interface. On the left, the XML code for `menu_tasarim.xml` is displayed:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

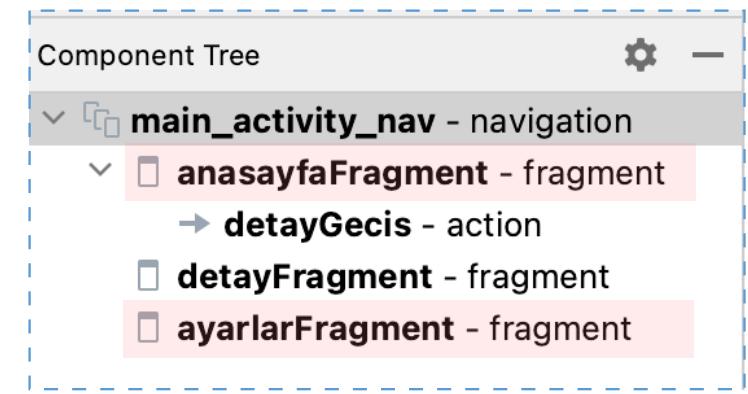
    <item
        android:id="@+id/anasayfaFragment"
        android:icon="@drawable/anasayfa_resim"
        android:title="Anasayfa" />

    <item
        android:id="@+id/ayarlarFragment"
        android:icon="@drawable/ayarlar_resim"
        android:title="Ayarlar" />

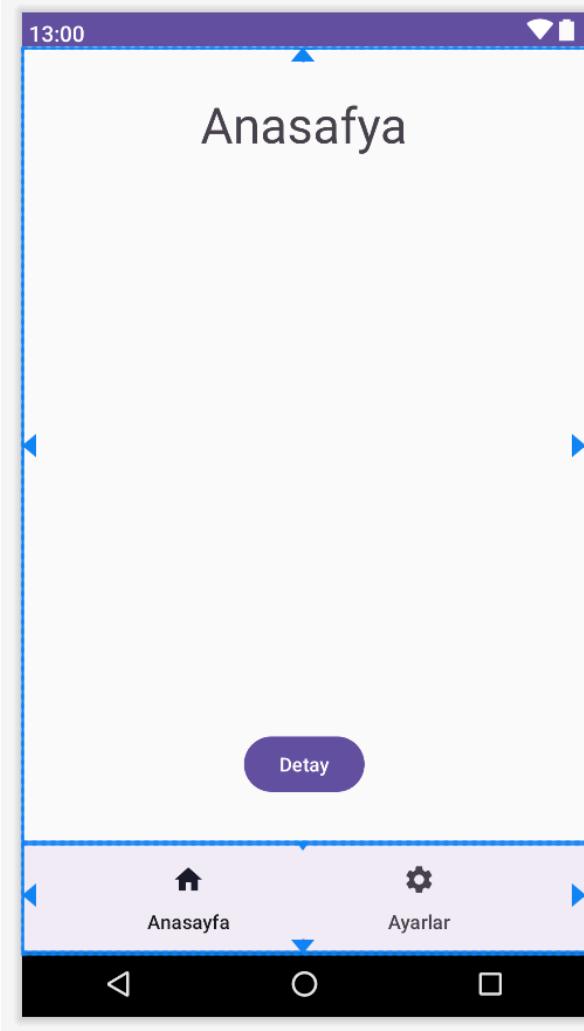
</menu>
```

On the right, the Component Tree shows the navigation structure:

- `main_activity_nav - navigation`
  - `anasayfaFragment - fragment` (selected)
  - `detayGecis - action`
  - `detayFragment - fragment`
  - `ayarlarFragment - fragment`



# Activity Sayfasi



The screenshot shows an Android activity titled "Anasafya". At the top, there is a purple header bar with the time "13:00" and signal/battery icons. Below the header is a white content area containing a purple rounded rectangular button with the text "Detay". At the bottom is a navigation bar with a light blue header. On the left side of the navigation bar is a house icon with the text "Anasayfa". On the right side is a gear icon with the text "Ayarlar". The bottom of the screen features a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

**Component Tree**

- ConstraintLayout
  - navHostFragment1 androidx.navigation...
  - bottomNavigationView

**Common Attributes**

|                             |                        |
|-----------------------------|------------------------|
| style                       | @style/Widget.Mat      |
| itemHorizontalTranslatio... | false                  |
| labelVisibilityMode         | auto                   |
| itemIconTint                | @color/m3_navigatio... |
| menu                        | @menu/menu_tasarim     |
| itemBackground              | @null                  |
| itemTextColor               | @color/m3_navigatio... |
| elevation                   | 3dp                    |
| alpha                       |                        |

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
- buttonx "Button"

1

```
class MainActivity : AppCompatActivity() {
```

3

```
    private lateinit var tasarim: ActivityMainBinding
```

```
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)
```

```
        tasarim.buttonx.setOnClickListener { it: View!
```

```
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }
```

}

2

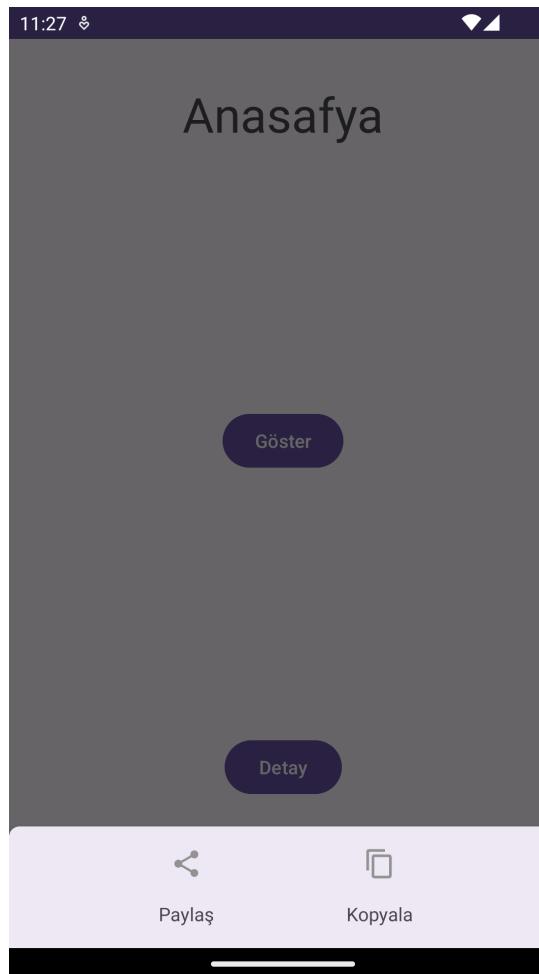
# Activity Kodlamasi

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        val navHostFragment1 =  
            supportFragmentManager.findFragmentById(R.id.navHostFragment1) as NavHostFragment  
  
        NavigationUI.setupWithNavController(binding.bottomNavigationView, navHostFragment1.navController)  
    }  
}
```

# Bottom Sheet Dialog

# Bottom Sheet Dialog

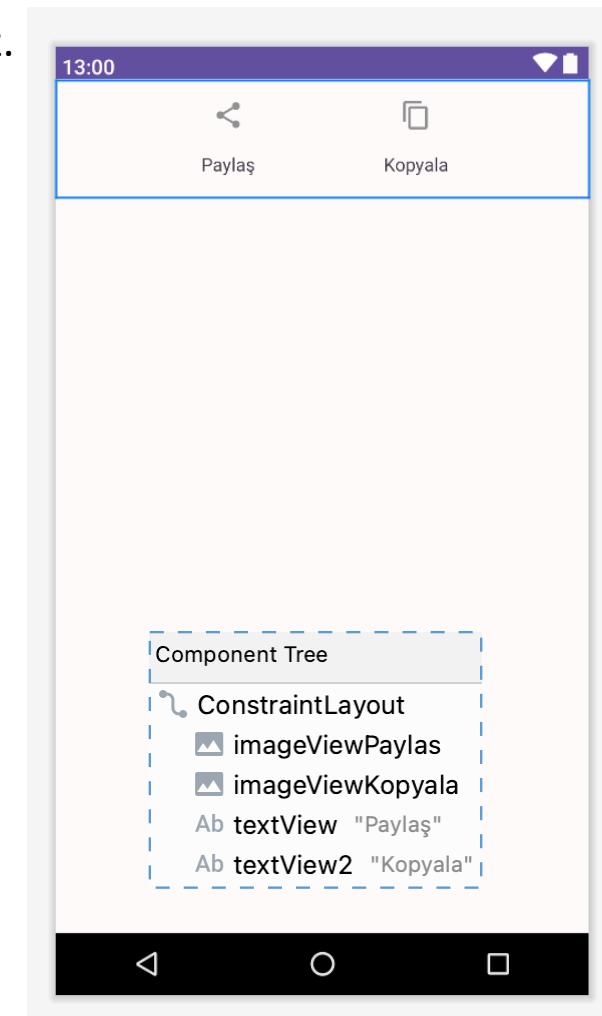
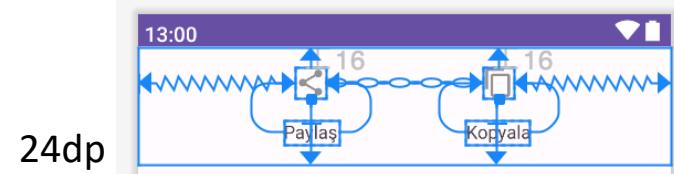
- Sayfanın altında yardımcı bir alan olarak açılan yapıdır.



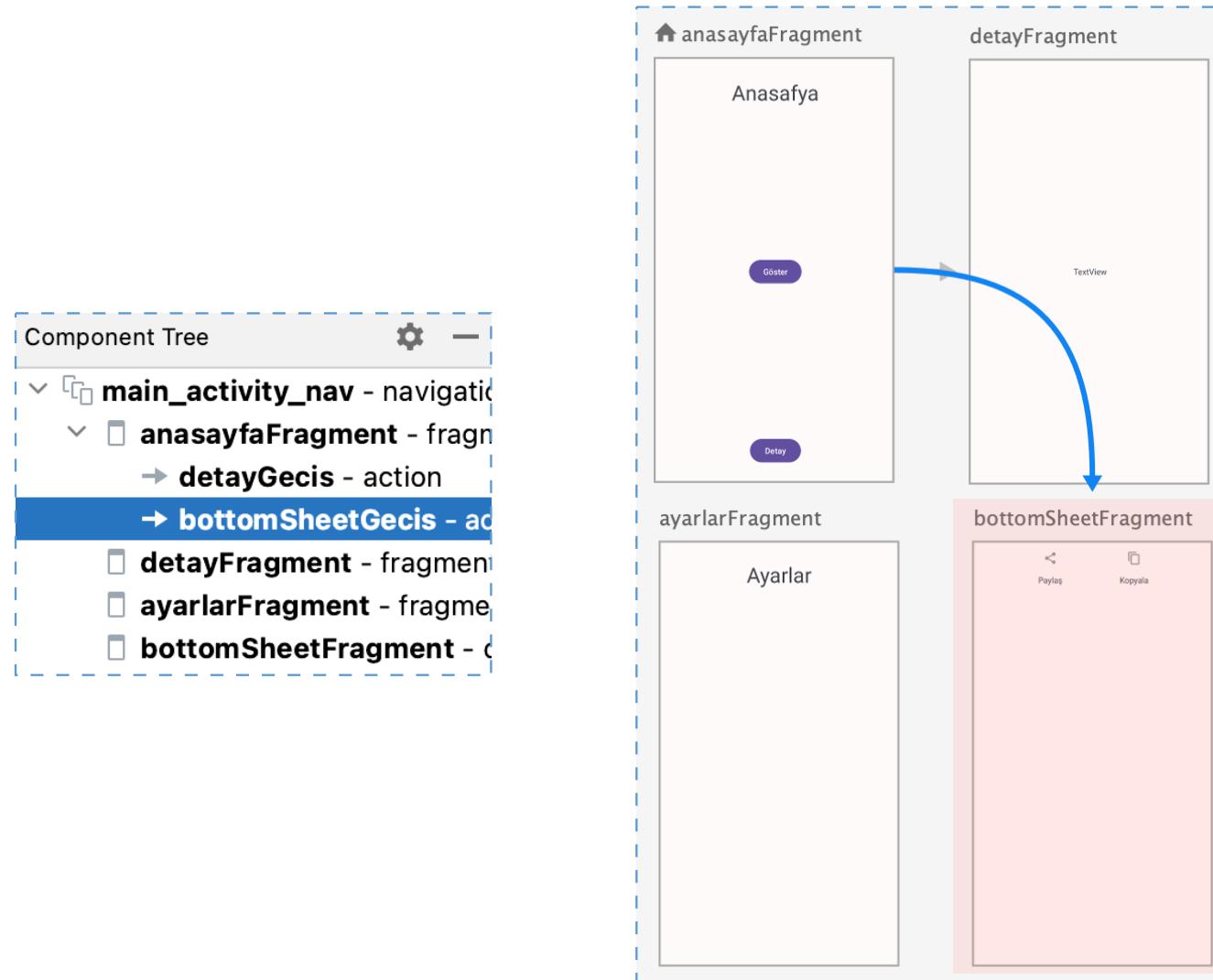
# Bottom Sheet Dialog Tasarımı

- Tasarım oluşturmak için Fragment oluşturmamız gereklidir.
- Fragment tasarımı içeriği sayfa altında görebiliriz.

```
class BottomSheetFragment : BottomSheetDialogFragment() { Fragment bu özellikle olmalıdır.  
    private lateinit var binding: FragmentBottomSheetBinding  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        binding = FragmentBottomSheetBinding.inflate(inflater, container, attachToParent: false)  
  
        binding.imageViewPaylas.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Paylaşıldı", Snackbar.LENGTH_SHORT).show()  
        }  
  
        binding.imageViewKopyala.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Kopyalandı", Snackbar.LENGTH_SHORT).show()  
        }  
  
        return binding.root  
    }  
}
```

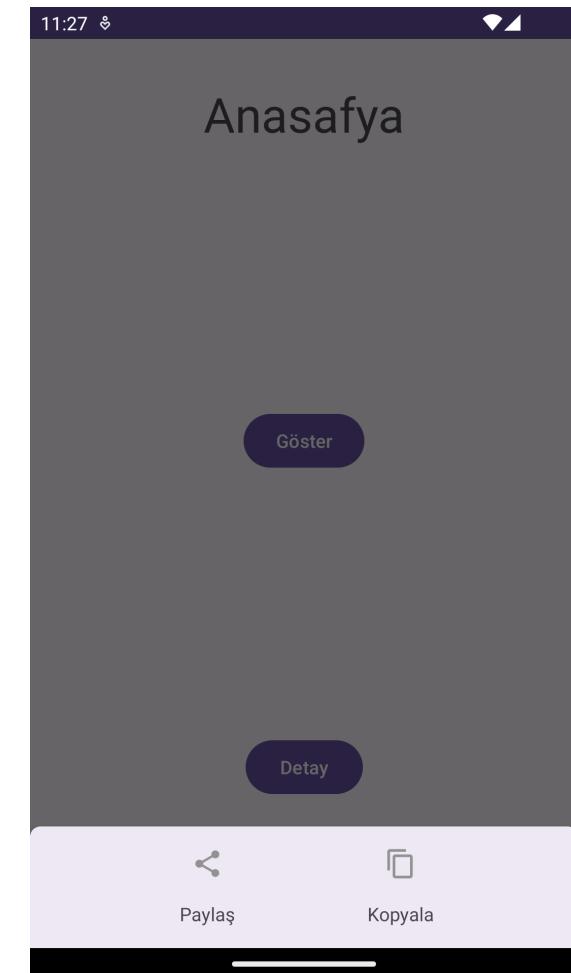


# Navigation Component Kullanımı



# Bottom Sheet Dialog Görüntüleme

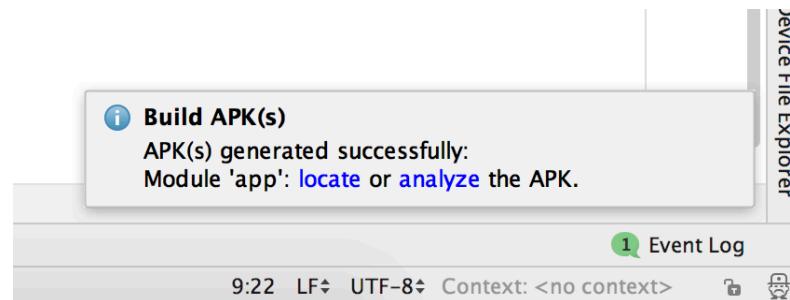
```
class AnasayfaFragment : Fragment() {  
    private lateinit var binding: FragmentAnasayfaBinding  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        binding = FragmentAnasayfaBinding.inflate(inflater, container, attachToParent: false)  
  
        binding.buttonDetay.setOnClickListener { it: View!  
            val urun = Urunler(id: 100, ad: "TV")  
            val gecis = AnasayfaFragmentDirections  
                .detayGecis(urun = urun, ad = "Ahmet", yas = 23, boy = 1.78f, bekar = true)  
            Navigation.findNavController(it).navigate(gecis)  
        }  
  
        binding.buttonGoster.setOnClickListener { it: View!  
            Navigation.findNavController(it).navigate(R.id.bottomSheetGecis)  
        }  
    }  
  
    return binding.root  
}
```



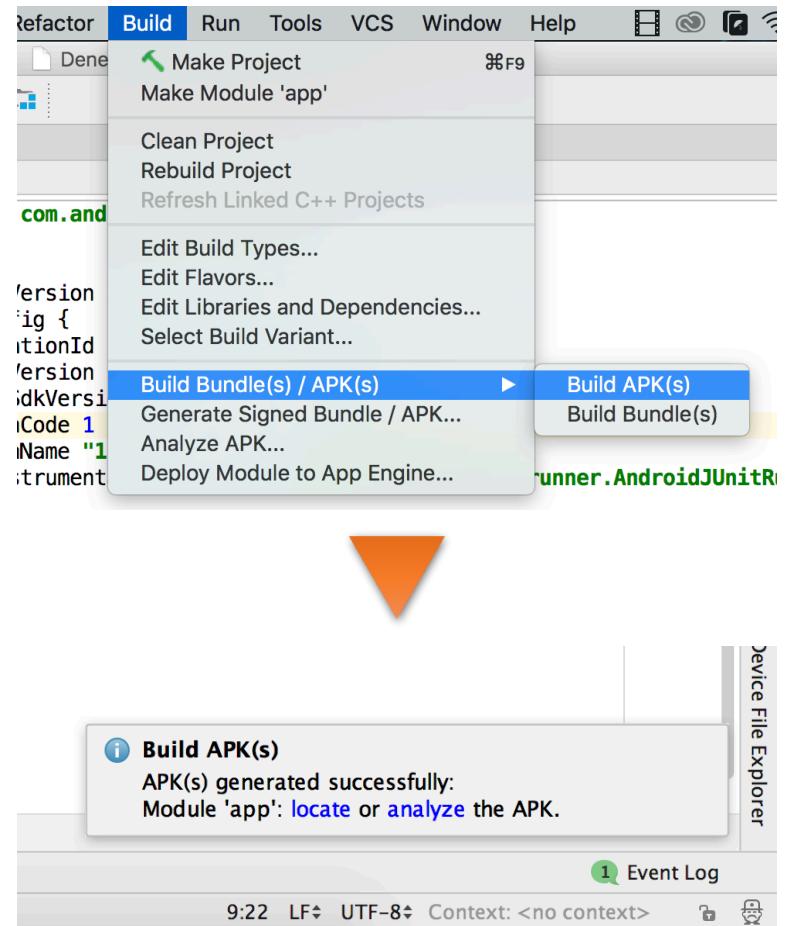
# Uygulama Çıktısı Alma

# Apk Oluşturma

- **İmzalı apk** : Google play'e yüklemek için gerekli olan apk sürümü.
- **Normal apk** : Google play haricinde kullanmak için kullanılır.



# Normal APK



# İmzalı ( Signed ) APK

```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 27
5     defaultConfig {
6         applicationId "com.info.xxx.aaaa.deneme"
7         minSdkVersion 16
8         targetSdkVersion 27
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
17        }
18    }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:27.1.1'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28 }
```

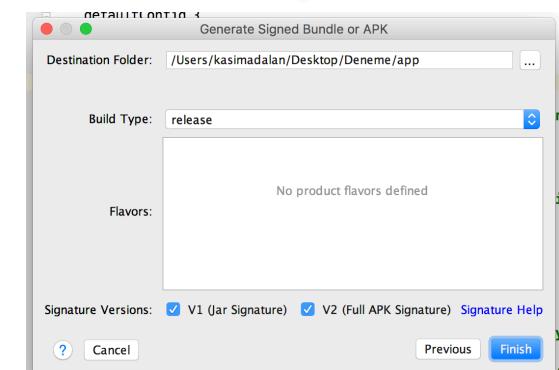
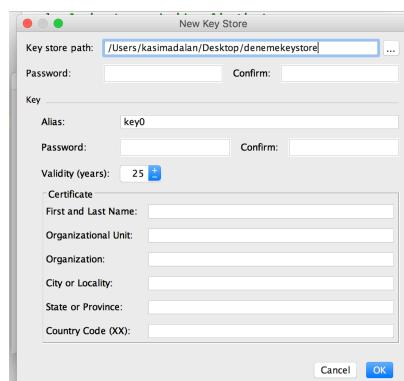
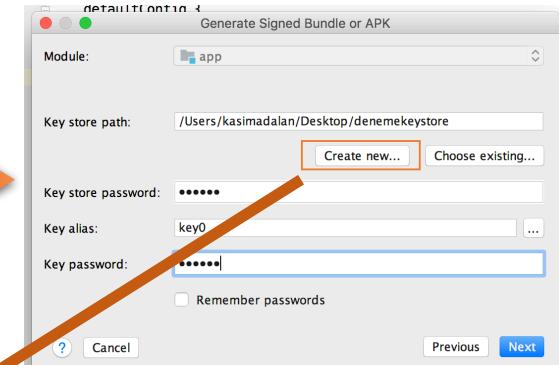
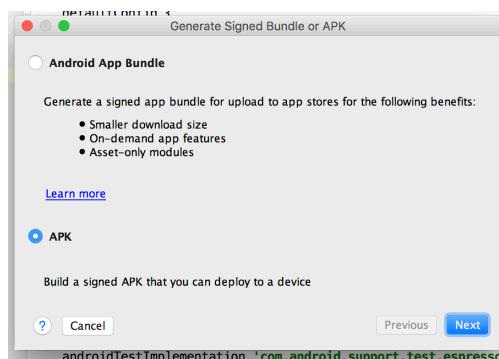
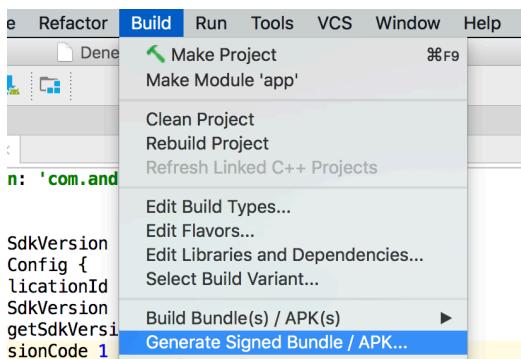
applicationId uygulama paket ismi ile aynıdır ve google play'de kullanılmamış olmalıdır.

versionCode yazılım sırasıdır.

versionName ise kullanıcının göreceği versiyodur.

Büyük güncellemelerde büyük artış yapılır.

# İmzalı APK Oluşturma



Kasım ADALAN

# Uygulamanın Google Play Yüklenmesi



# Uygulamanın Google Play Yüklenmesi

The screenshot shows the Google Play Console interface. On the left, there's a sidebar with various navigation options like 'Tüm uygulamalar', 'Hesap özeti', and 'Mağaza giriş'i, which is highlighted with a blue border. The main area is titled 'Mağaza giriş'i and shows 'Ürün bilgileri' (Product information). It features a placeholder for a product image labeled 'Deneme Taslak'. At the top right, it says 'TÜRKÇE – TR-TR' and 'Çevirileri yönet'. Below that, there's a text input field for 'Tam açıklama \*' (Full description \*) with a character limit of '15/80'. A red warning message is displayed in the center: 'Google Play Console sayfasından uygulama paylaşmak için her yıl 25 dolar verilmelidir.' (Sharing an app from the Google Play Console page requires paying \$25 per year.) At the bottom, there's a note about meta data policies and a large blue button labeled 'TASLAĞI KAYDET'.

# Mağaza için Görsellerin Hazırlanması

Ürün bilgileri TÜRKÇE – TR-TR Çeviri

2/8 ekran görüntüsü DOSYALARA

|  |  |  |
|--|--|--|
| <b>Yüksek çözünürlüklü simge *</b><br>Varsayılan – Türkçe – tr-TR<br><b>512 x 512</b><br>32 bit PNG (alfa kanallı) | <b>Öne Çıkan Grafik *</b><br>Varsayılan – Türkçe – tr-TR<br><b>1024 g x 500 y</b><br>JPG veya 24 bit PNG (alfa kanalsız)                         | <b>Tanıtım Grafiği</b><br>Varsayılan – Türkçe – tr-TR<br><b>180 g x 120 y</b><br>JPG veya 24 bit PNG (alfa kanalsız) |
|                                  |  | <br>+                             |
| <b>TV Bannerı</b><br>Varsayılan – Türkçe – tr-TR<br><b>1280 g x 720 y</b><br>JPG veya 24 bit PNG (alfa kanalsız)   | <b>Daydream 360 derecelik stereoskopik resim</b><br>Varsayılan – Türkçe – tr-TR<br><b>4096 g x 4096 y</b><br>JPG veya 24 bit PNG (alfa kanalsız) |                                  |

# İmzalı APK'nın yüklenmesi

Hesap özet  
Uygulama sürümleri  
Android Hazır Uygulamalar  
Yapı kitaplığı  
Cihaz kataloğu  
Uygulama imzalama  
Mağaza giriş  
İçerik derecelendirme  
Fiyatlandırma ve dağıtım  
Uygulama içi ürünler

← Yeni üretim sürümü

1 Sürümü hazırlayın 2 İnceleyin ve kullanıma sunun

Google Play'den uygulama imzalama özelliği  
Etkin.

Eklenecek Android Uygulama Paketleri ve APK'lar  
Bu uygulama paketleri ve APK'lar bu sürümün kullanıma sunulmasından sonra Google Play Store'da sunulacak.  
KİTAPLIKTAN EKLE

Uygulama paketlerinizi ve APK'larınızı buraya bırakın veya bir dosya seçin.

DOSYALARA GÖZ AT

# Widgets

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

```
class MainActivity : AppCompatActivity() {
```

3

```
    private lateinit var tasarim: ActivityMainBinding
```

```
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)
```

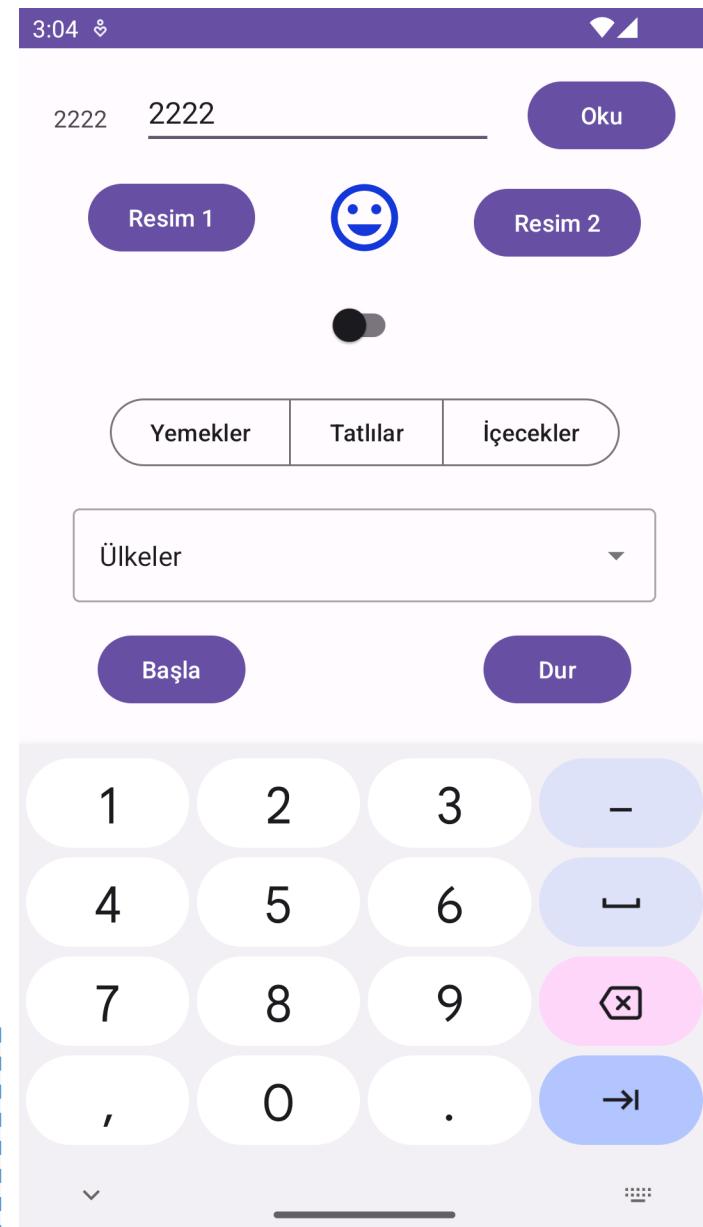
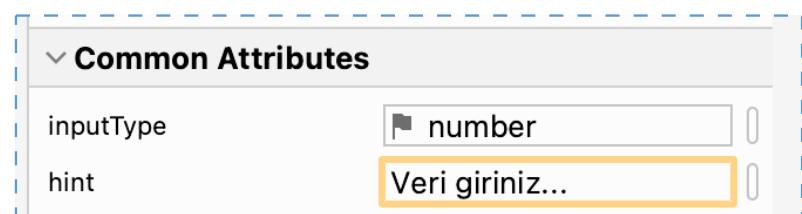
```
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }
```

*Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.*

# TextView - Button - EditText

```
binding.buttonOku.setOnClickListener { it: View!  
    val alınanVeri = binding.editTextGirdi.text.toString()  
    binding.textViewSonuc.text = alınanVeri  
}
```

*Input Type klavye türünü değiştirmektedir.*

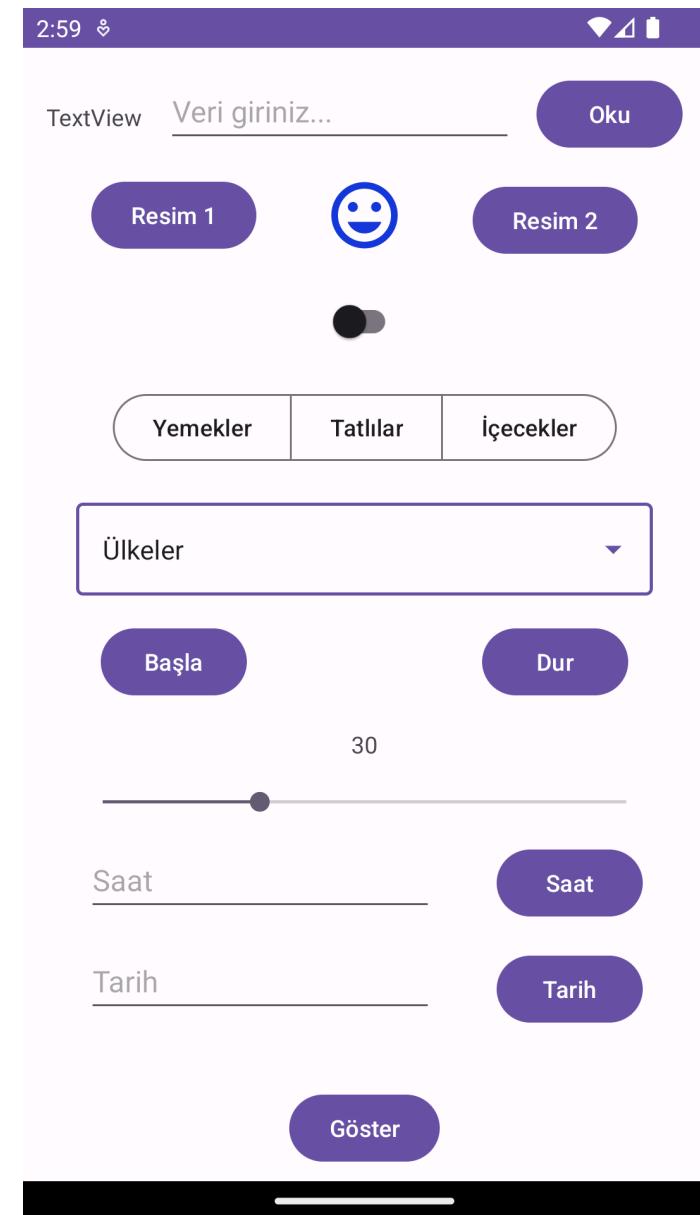


# ImageView

```
binding.buttonResim1.setOnClickListener { it: View!
    binding.imageView.setImageResource(R.drawable.resim1)
}

binding.buttonResim1.setOnClickListener { it: View!
    binding.imageView.setImageResource(
        resources.getIdentifier( name: "resim2", defType: "drawable", packageName))
}
```

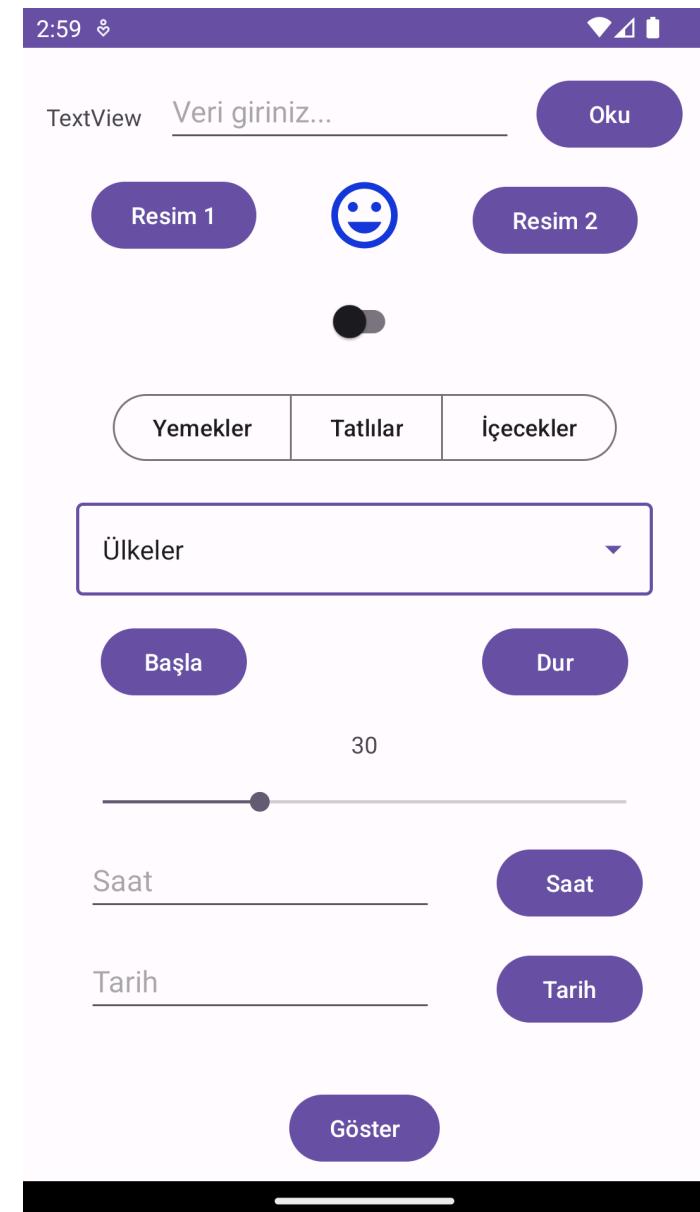
Resim boyutu : 48dp x 48dp



# Switch Material

```
binding.switch1.setOnCheckedChangeListener { buttonView, isChecked ->
    if(isChecked){
        Log.e( tag: "Widgets", msg: "Switch : ON")
    }else{
        Log.e( tag: "Widgets", msg: "Switch : OFF")
    }
}
-----
binding.buttonGoster.setOnClickListener { it: View!
    Log.e( tag: "Widgets", msg: "Switch durum : ${binding.switch1.isChecked}")
}

<com.google.android.material.switchmaterial.SwitchMaterial
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```



# Toggle Buttons

7:16



Yemekler Tatlılar İçecekler

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    private var kontrol = false  
  
    binding.toggleButton.setOnButtonCheckedListener { group, checkedId, isChecked ->  
        kontrol = isChecked  
        if(kontrol){  
            val secilenButton = findViewById<Button>(binding.toggleButton.checkedButtonId)  
            val buttonYazi = secilenButton.text.toString()  
            Log.e( tag: "Sonuc",buttonYazi)  
        }  
  
        binding.buttonGoster.setOnClickListener { it: View!  
            Log.e( tag: "Sonuc", msg: "Switch Durum : ${binding.switch1.isChecked}")  
            if (kontrol){  
                val secilenButton = findViewById<Button>(binding.toggleButton.checkedButtonId)  
                val buttonYazi = secilenButton.text.toString()  
                Log.e( tag: "Sonuc", msg: "Toggle Durum : $buttonYazi")  
            }  
        }  
    }  
}
```



```
<com.google.android.material.button.MaterialButtonToggleGroup  
    android:id="@+id/toggleButton"      Component panelde direk yer  
    android:layout_width="wrap_content"  olmadığı xml için elimizle  
    android:layout_height="wrap_content"  yazmamızı.  
    app:singleSelection="true">   Buttonlar arasında tek seçim  
                                    yapmayı aktif eder.  
        <Button  
            android:id="@+id/buttonYemekler"  
            style="?attr/materialButtonOutlinedStyle"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Yemekler" />  
        <Button  
            android:id="@+id/buttonTatlilar"  
            style="?attr/materialButtonOutlinedStyle"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Tatlılar" />  
        <Button  
            android:id="@+id/buttonIcecekler"  
            style="?attr/materialButtonOutlinedStyle"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="İçecekler" />  
    </com.google.android.material.button.MaterialButtonToggleGroup>
```

Buttonu çerçeveli gösterir.

# Expose Dropdown Menu

- Kullanıcıdan girdi almamızı sağlar.
- Aynı anda girdi , seçim ve otomatik tamamlama özelliği sunmaktadır.

```
<com.google.android.material.textfield.TextInputLayout  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.ExposedDropdownMenu"  
    android:layout_width="match_parent"                      Çerçevevi modern görünüm oluştur.  
    android:layout_height="wrap_content"  
    android:layout_marginStart="32dp"  
    android:layout_marginTop="32dp"  
    android:layout_marginEnd="32dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/toggleButton">  
  
<AutoCompleteTextView  
    android:id="@+id/autoCompleteTextView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:text="Ülkeler" /> Açıklama yazısı  
</com.google.android.material.textfield.TextInputLayout>
```

Türkiye ▾

Türkiye

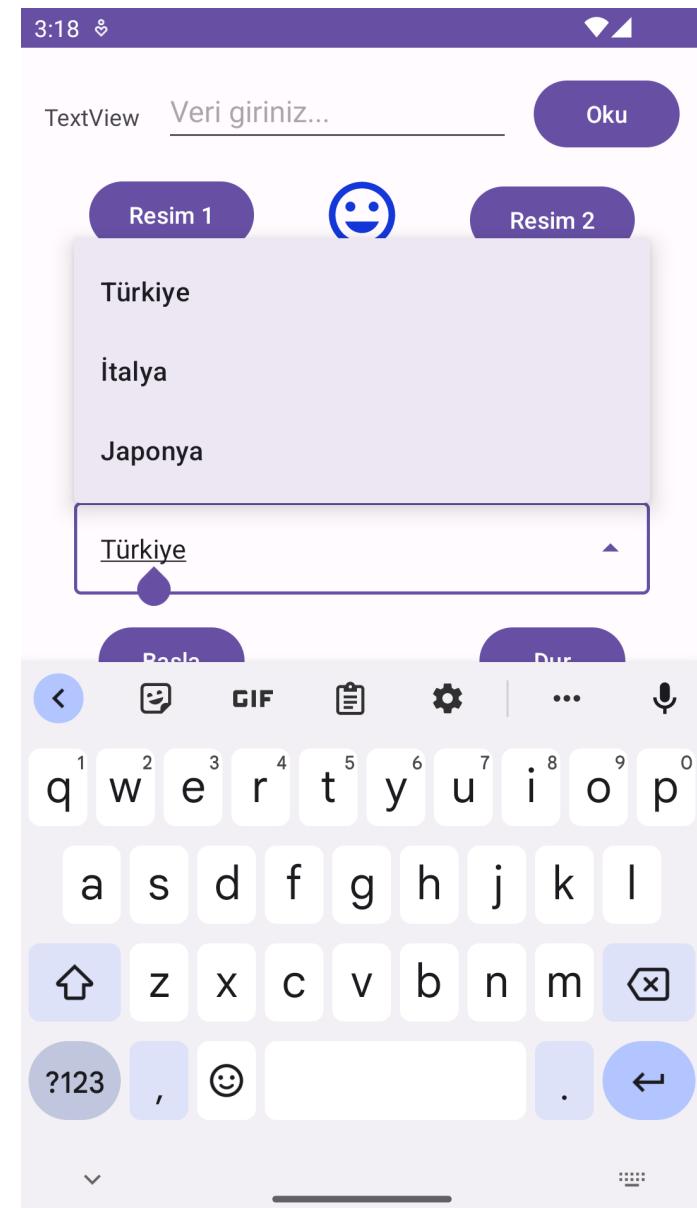
İtalya

Japonya

# Expose Dropdown Menu

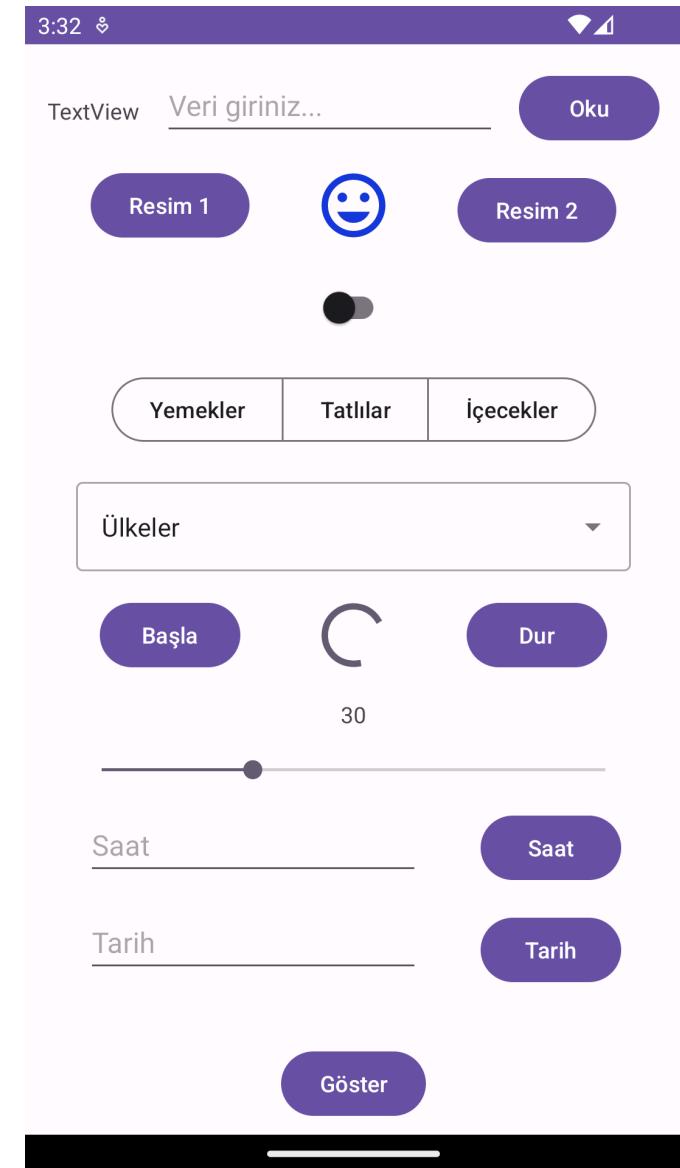
```
val ulkeler = ArrayList<String>()
ulkeler.add("Türkiye")
ulkeler.add("İtalya")
ulkeler.add("Japonya")
val arrayAdapter = ArrayAdapter(context: this, android.R.layout.simple_list_item_1, ulkeler)
binding.autoCompleteTextView.setAdapter(arrayAdapter)
```

```
binding.buttonGoster.setOnClickListener { it: View!
    val ulke = binding.autoCompleteTextView.text.toString()
    Log.e(tag: "Widgets", msg: "Ülke : $ulke")
}
```



# Progress Bar

```
binding.buttonBasla.setOnClickListener { it: View!  
    binding.progressBar.visibility = View.VISIBLE  
}  
  
binding.buttonDur.setOnClickListener { it: View!  
    binding.progressBar.visibility = View.INVISIBLE  
}
```

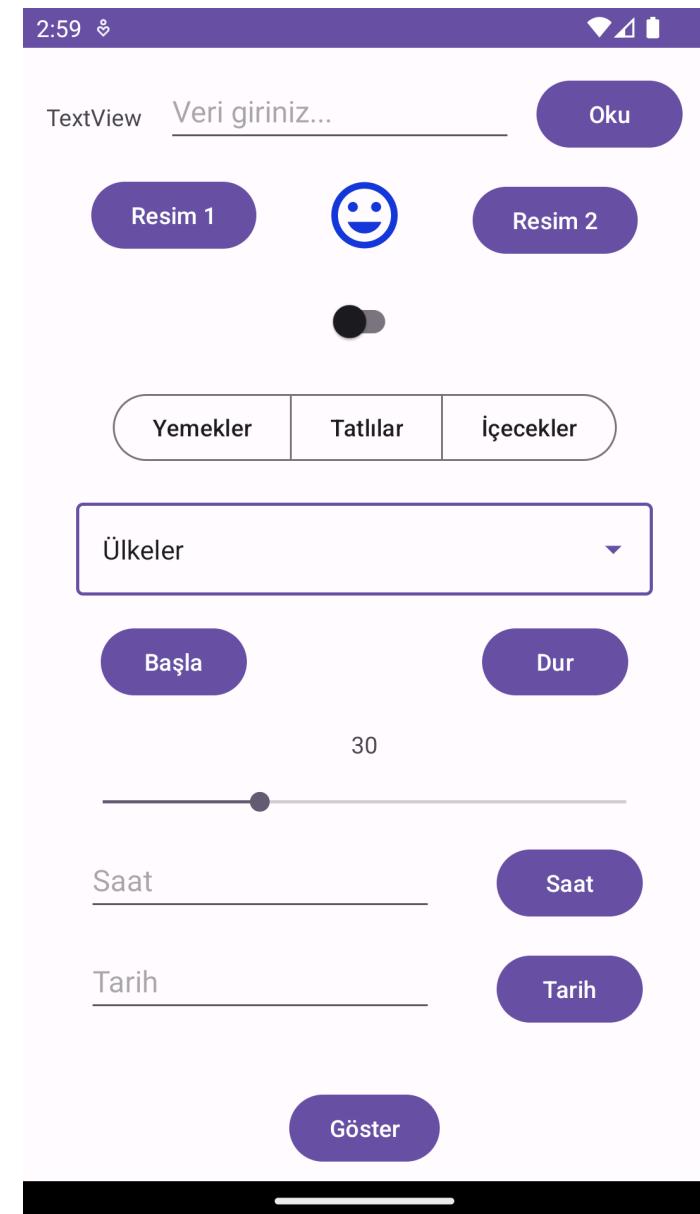


# Slider

```
binding.textViewSlider.text = binding.slider.progress.toString()

binding.slider.setOnSeekBarChangeListener(object : OnSeekBarChangeListener{
    override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
        binding.textViewSlider.text = progress.toString()
    }
    override fun onStartTrackingTouch(seekBar: SeekBar?) {}
    override fun onStopTrackingTouch(seekBar: SeekBar?) {}
})

binding.buttonGoster.setOnClickListener { it: View!
    Log.e( tag: "Widgets", msg: "Slider durum : ${binding.slider.progress}")
}
```

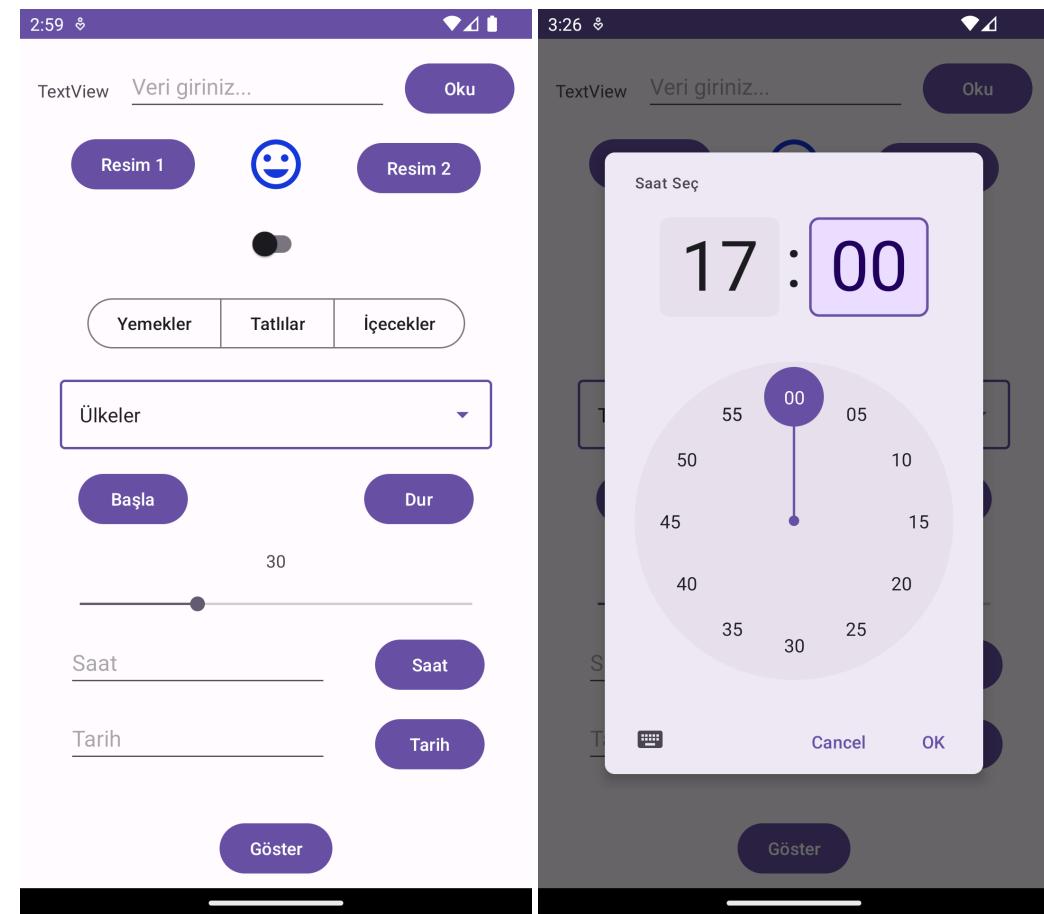


# Material Time Picker

```
binding.buttonSaat.setOnClickListener { it: View!
    val tp = MaterialTimePicker.Builder()
        .setTitleText("Saat Seç")
        .setTimeFormat(TimeFormat.CLOCK_24H)
        .build()

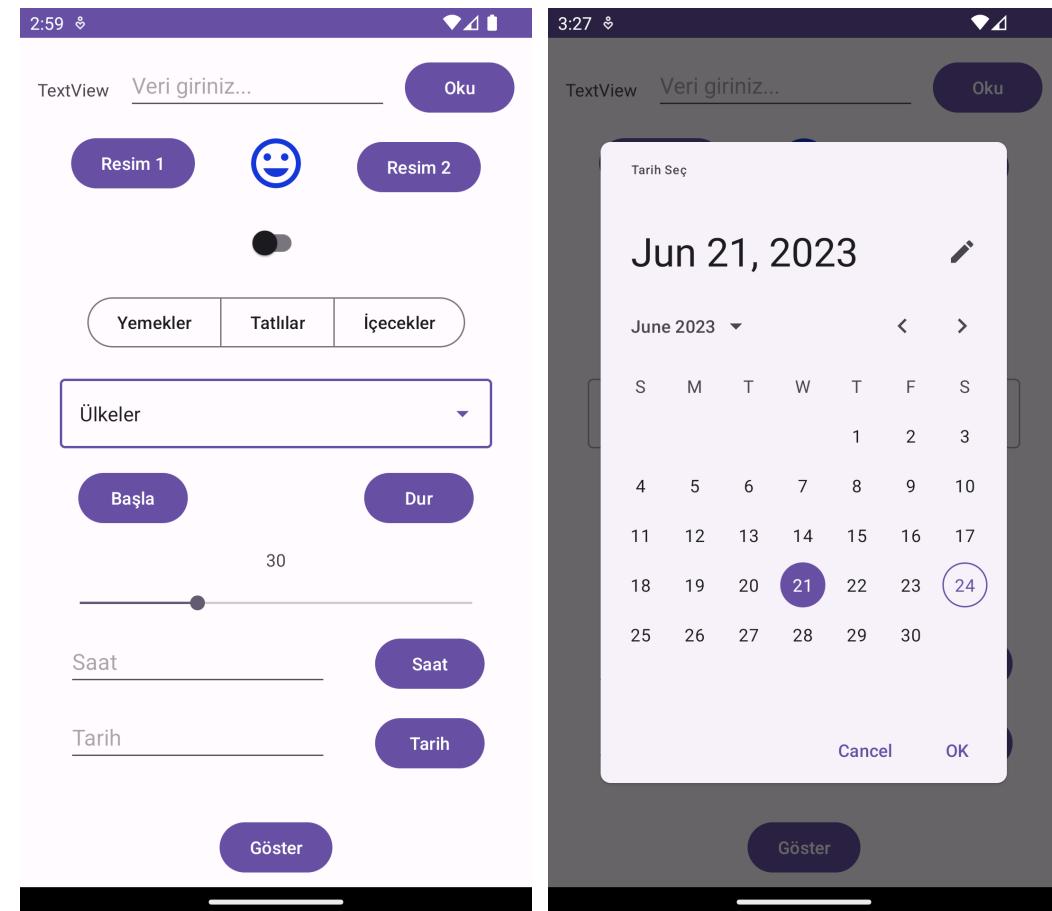
    tp.show(supportFragmentManager, tag: "Saat")

    tp.addOnPositiveButtonClickListener { it: View!
        binding.editTextSaat.setText("${tp.hour} : ${tp.minute}")
    }
}
```



# Material Date Picker

```
binding.buttonTarih.setOnClickListener { it: View!  
    val dp = MaterialDatePicker.Builder.datePicker()  
        .setTitleText("Tarih Séç")  
        .build()  
  
    dp.show(supportFragmentManager, tag: "Tarih")  
  
    dp.addOnPositiveButtonClickListener { it: Long!  
        val df = SimpleDateFormat(pattern: "dd/MM/yyyy", Locale.getDefault())  
        val tarih = df.format(it)  
        binding.editTextTarih.setText(tarih)  
    }  
}
```



# ScrollView

- ScrollView, Sayfa içerisindeki içerik ekran boyutundan daha fazla yer kaplıyorsa aşağı ve yukarı sayfayı hareket ettirmemize yardım eder.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context=".MainActivity">

    <Button ...>
    <ImageView ...>
    <Switch ...>

```

*Constraint Layout  
yüksekliği wrap  
content olmalıdır.*

<CheckBox ...>

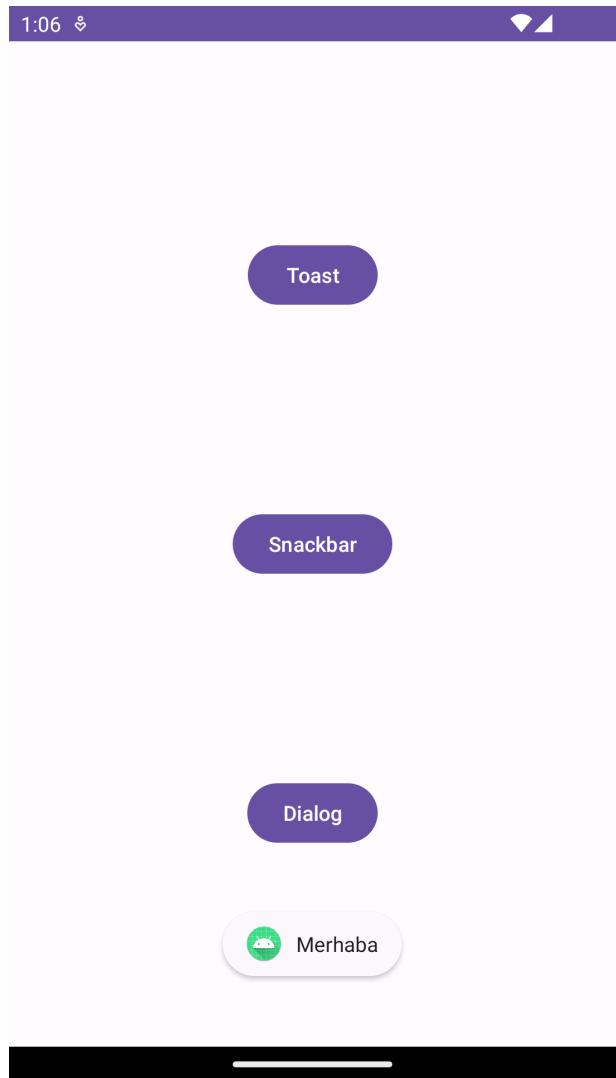
<RadioGroup ...>

<Button ...>

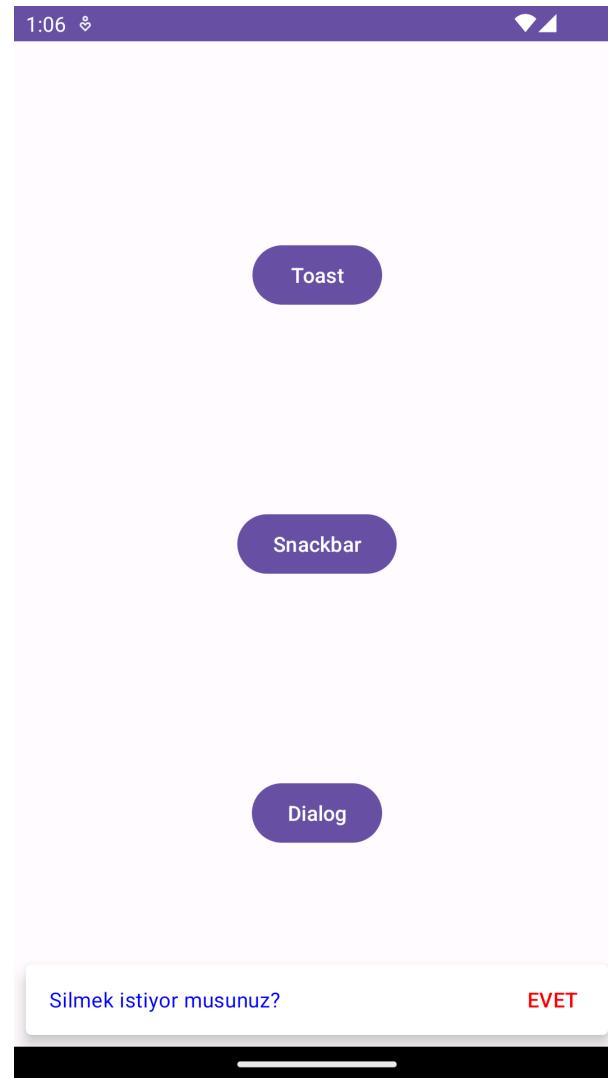
</androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>

# Kullanıcı Etkileşimi

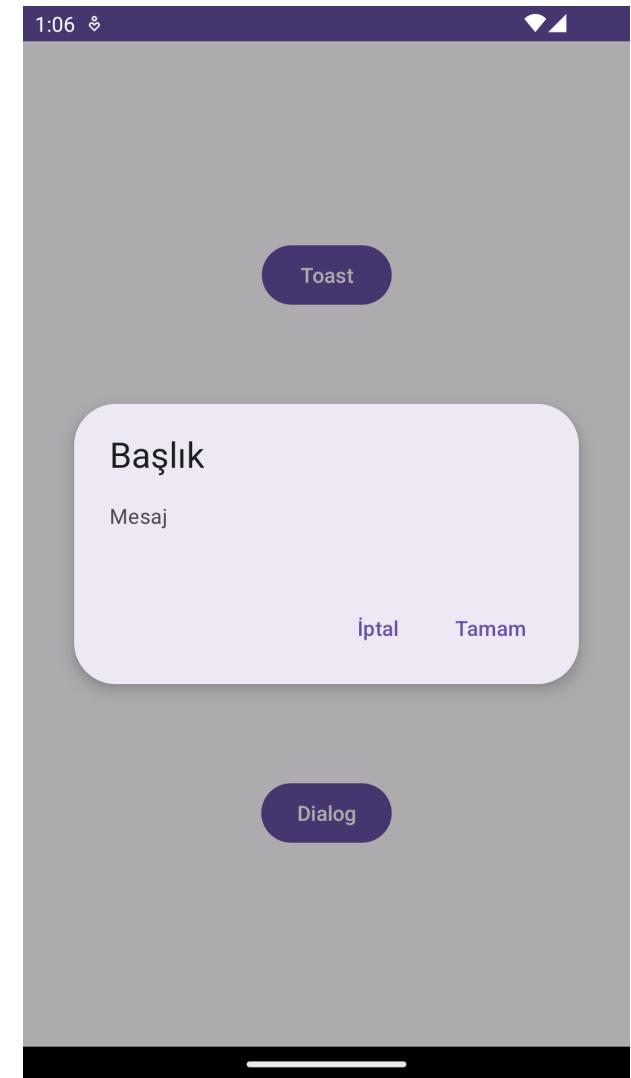
## *Toast*



## *Snackbar*



## *Dialog*



# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
  
    activity_main.xml
```

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

```
class MainActivity : AppCompatActivity() {
```

3

```
    private lateinit var tasarim: ActivityMainBinding
```

```
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)
```

```
        tasarim.buttonx.setOnClickListener { it: View!
```

```
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }
```

```
}
```

*Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.*

1:06 ☼



# Toast

Toast

Snackbar

Dialog



Merhaba

```
binding.buttonToast.setOnClickListener { it: View!
    Toast.makeText( context: this, text: "Merhaba", Toast.LENGTH_SHORT).show()
}
```

# Snackbar

Toast

Snackbar

Dialog

```
binding.buttonSnackbar.setOnClickListener { it: View!
    Snackbar.make(it, text: "Silmek istiyor musunuz?", Snackbar.LENGTH_LONG)
        .setAction( text: "EVET") { it: View!
            Snackbar.make(it, text: "Silindi", Snackbar.LENGTH_SHORT)
                .setBackgroundTint(Color.WHITE)
                .setTextColor(Color.BLUE)
                .show()
        }
        .setBackgroundTint(Color.WHITE)
        .setTextColor(Color.BLUE)
        .setActionTextColor(Color.RED)
        .show()
}
```

Silmek istiyor musunuz?

EVET

1:06

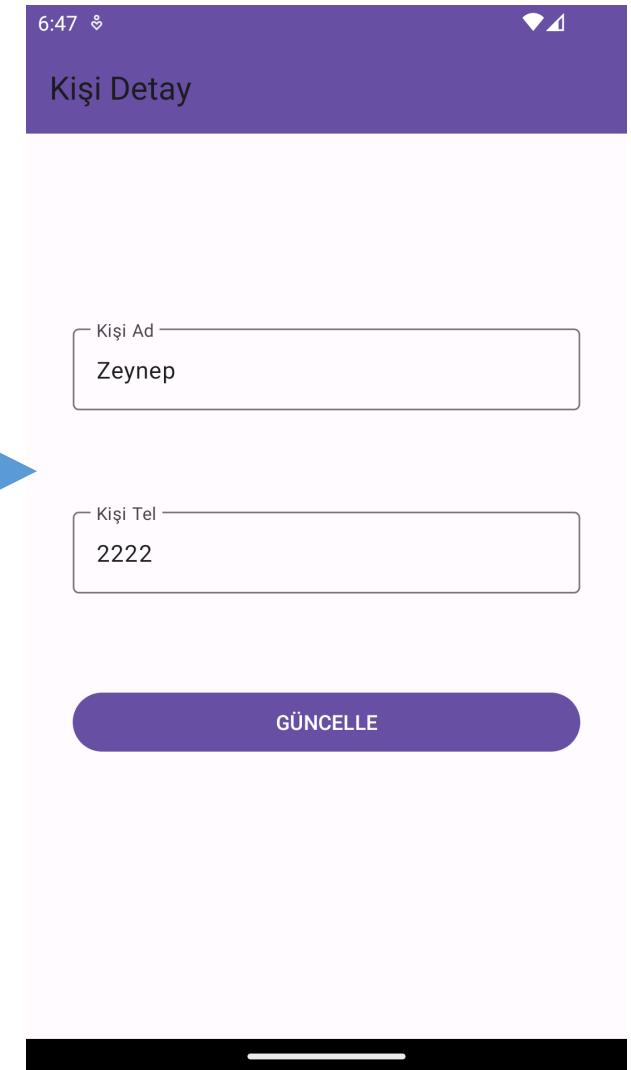
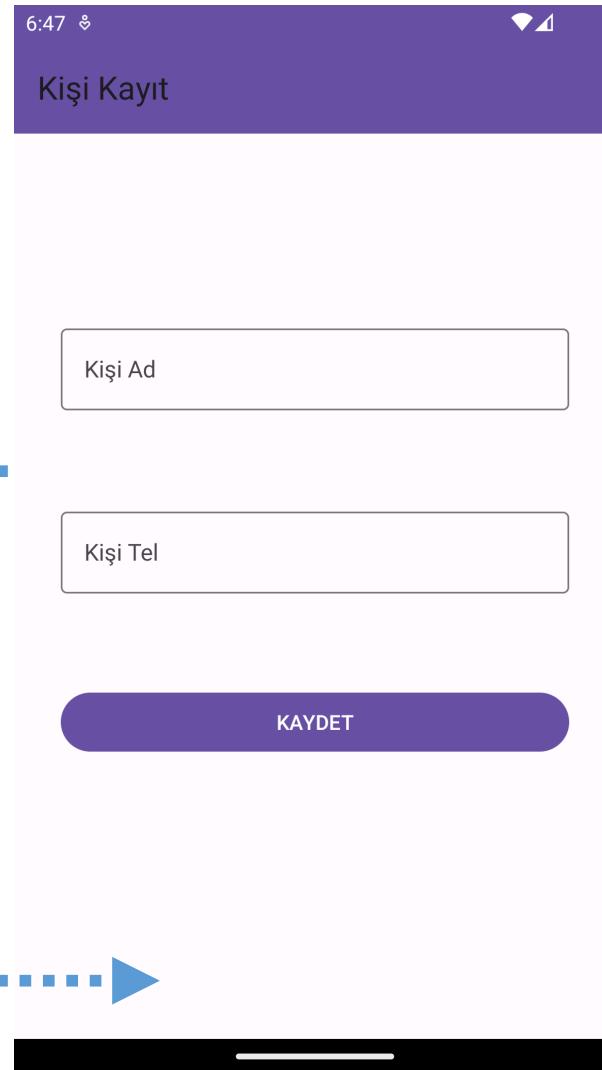
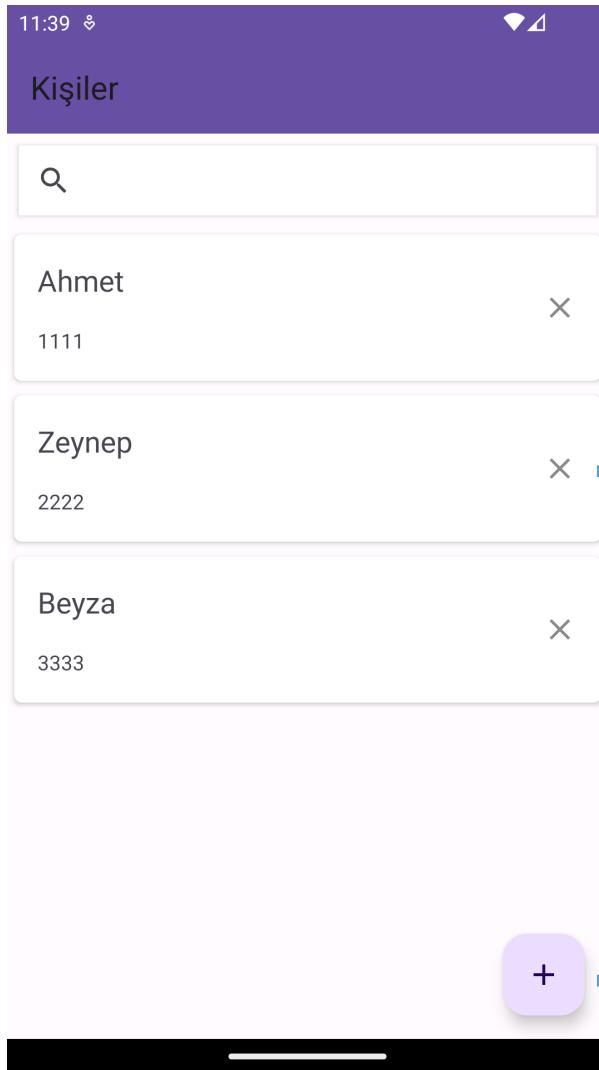


# Dialog

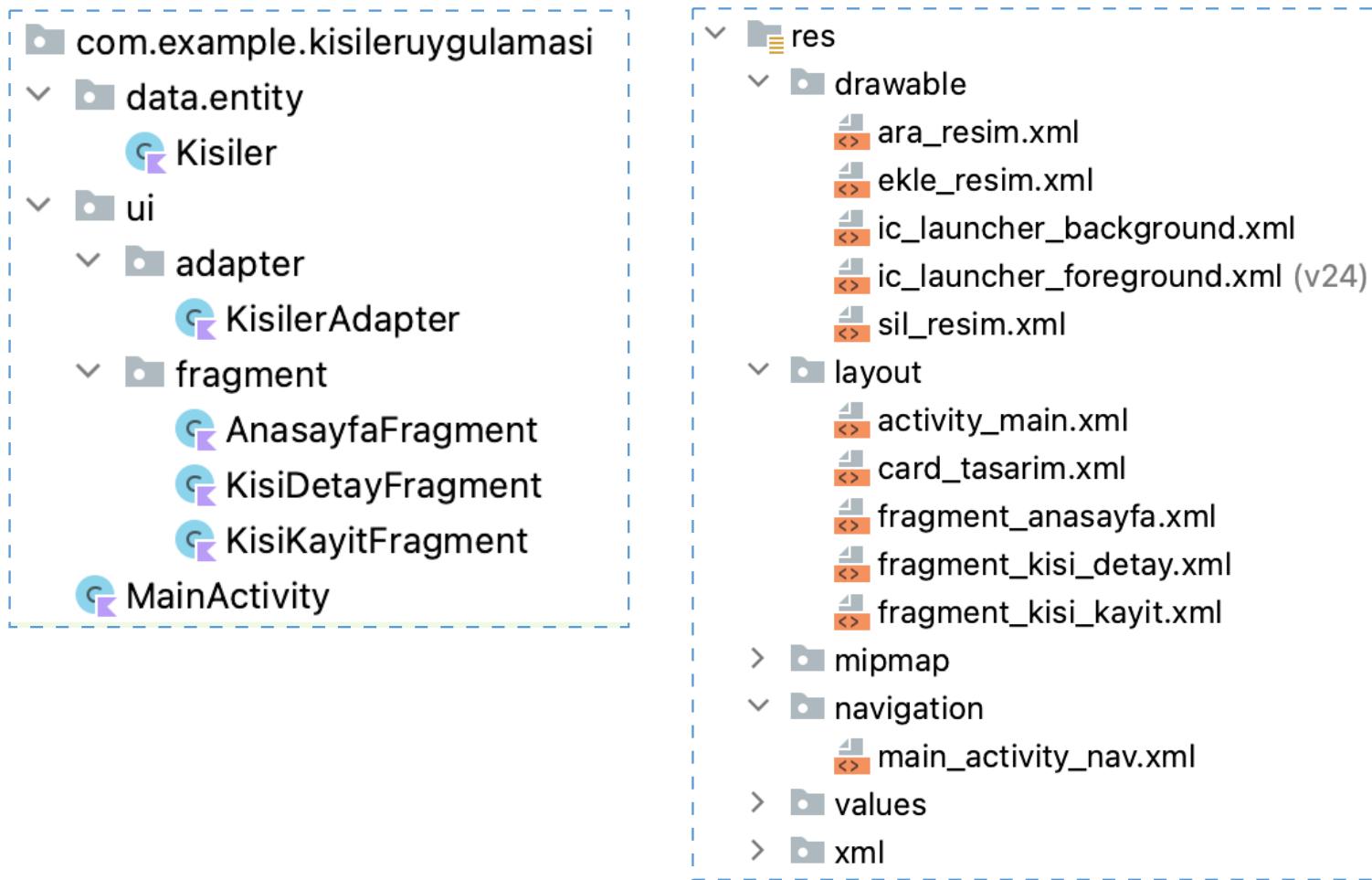
```
binding.buttonDialog.setOnClickListener { it: View!  
    MaterialAlertDialogBuilder(context: this)  
        .setTitle("Başlık")  
        .setMessage("Mesaj")  
        .setPositiveButton(text: "Tamam") { d, i ->  
            Toast.makeText(context: this, text: "Tamam Seçildi",  
                Toast.LENGTH_SHORT).show()  
        }  
        .setNegativeButton(text: "İptal") { d, i ->  
            Toast.makeText(context: this, text: "İptal Seçildi",  
                Toast.LENGTH_SHORT).show()  
        }  
    .show()  
}
```

# Material Design

# Kişiler Uygulaması

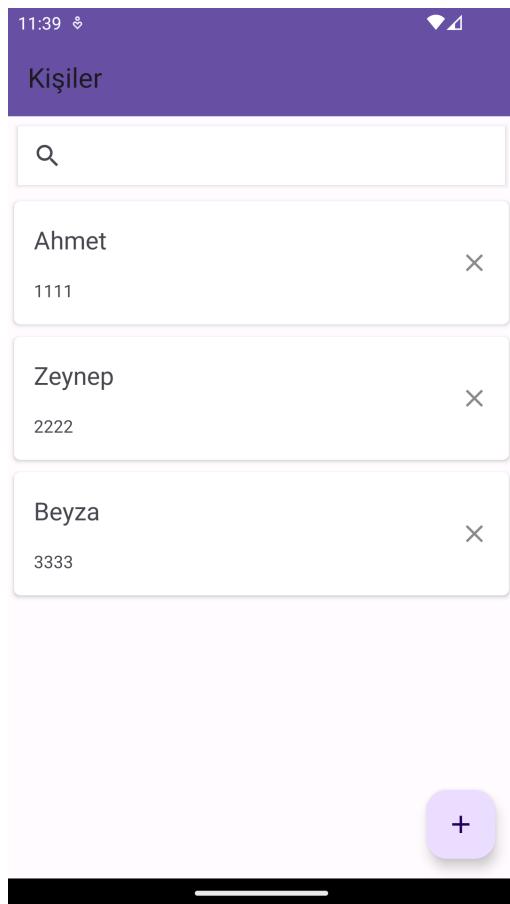


# Proje Genel Yapısı

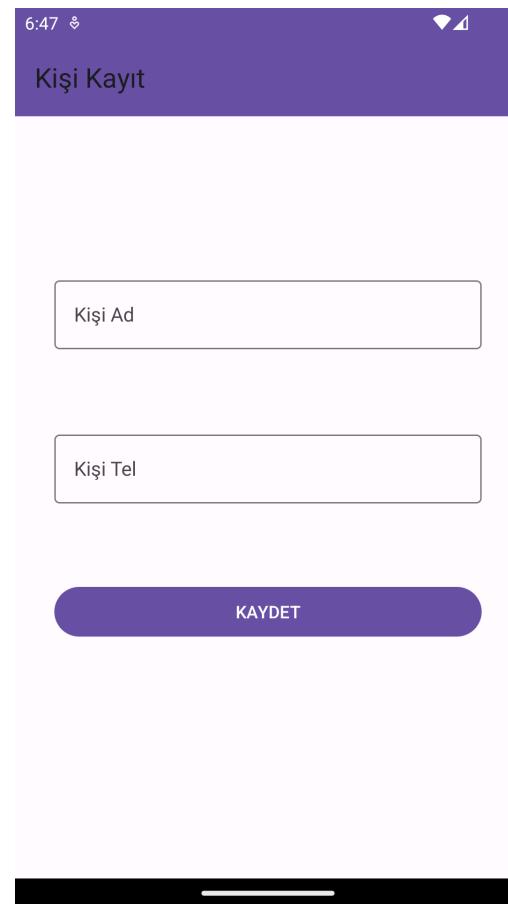


# Fragment Oluşturma

*AnasayfaFragment*



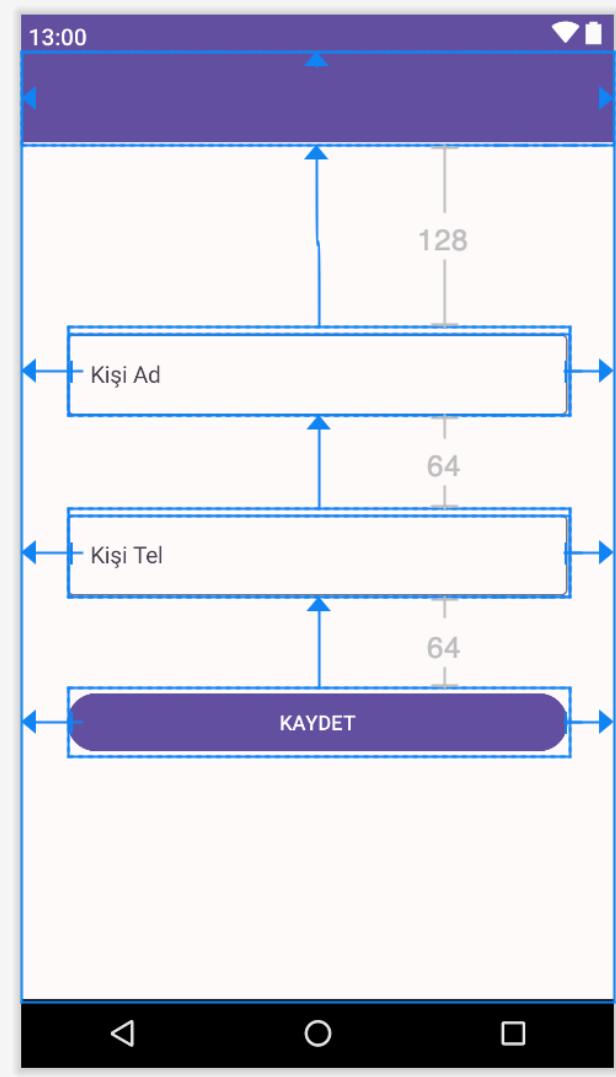
*KisiKayitFragment*



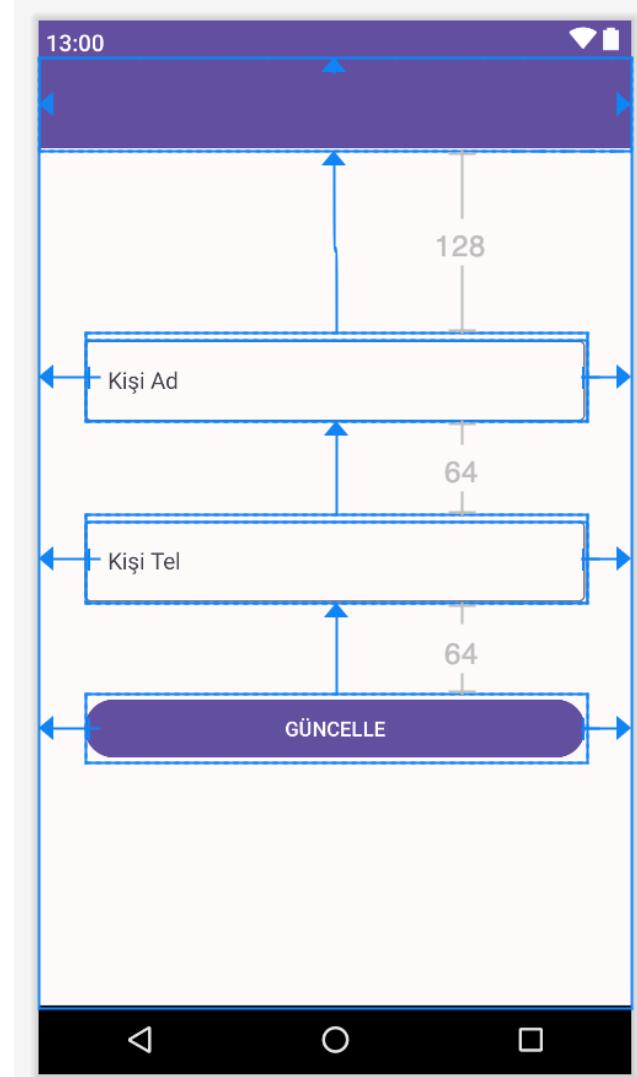
*KisiDetayFragment*



*KisiKayitFragment*



*KisiDetayFragment*



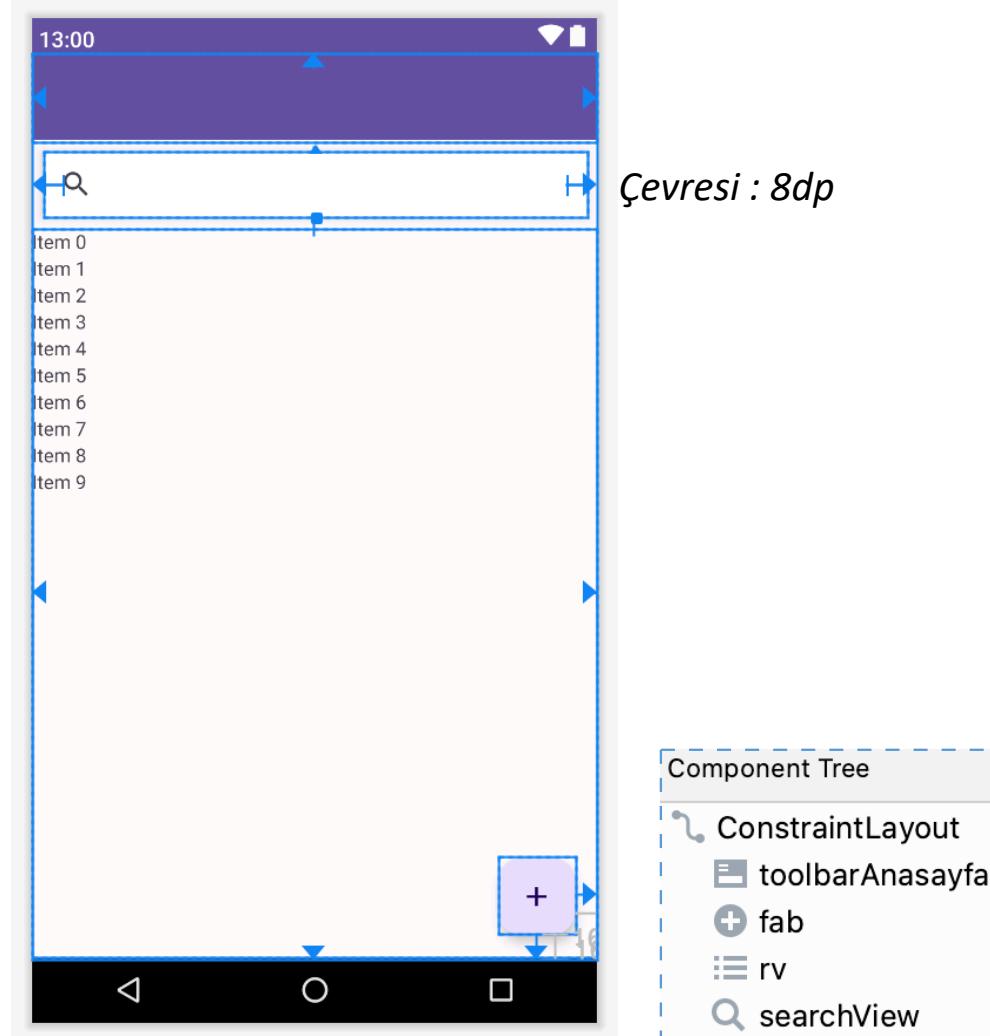
Component Tree

```
ConstraintLayout
└ Ab textInputLayout
  └ editTextKisiAd
└ Ab textInputLayout2
  └ editTextKisiTel
  └ buttonGuncelle "GÜNCELLE"
  └ toolbarKisiDetay
```

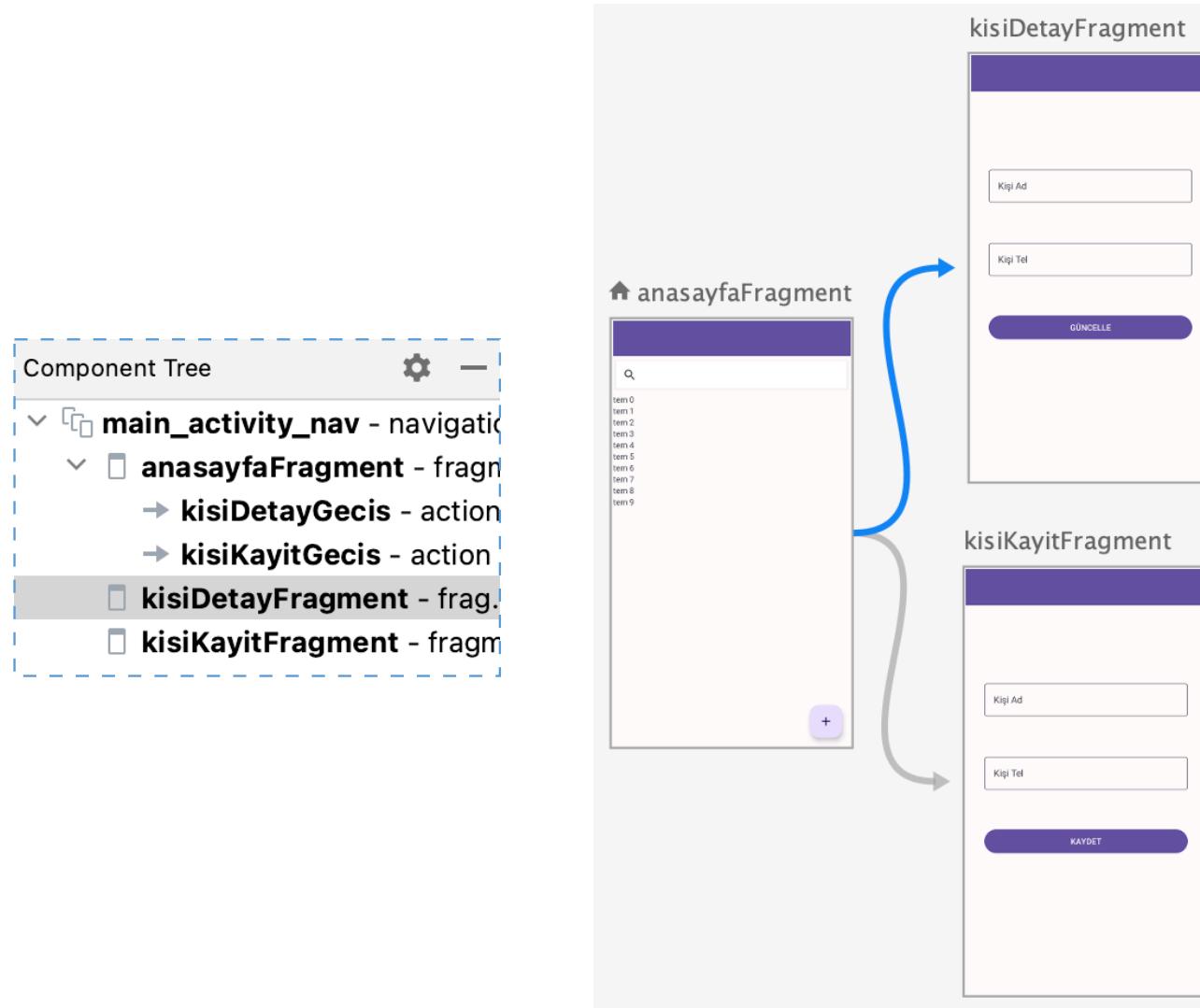
Component Tree

```
ConstraintLayout
  └ toolbarKisiKayit
  └ Ab textInputLayout
    └ editTextKisiAd
  └ Ab textInputLayout2
    └ editTextKisiTel
    └ buttonKaydet "KAYDET"
```

## AnasayfaFragment



# Navigation Component Kullanımı



# ViewBinding Kullanımı : Fragment

Build.gradle/module

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

fragment\_blank.xml

Component Tree

- ConstraintLayout
  - button "Button"

1

```
class BlankFragment : Fragment() {
```

```
    private lateinit var tasarim: FragmentBlankBinding
```

```
    override fun onCreateView(inflater: LayoutInflater,  
                             container: ViewGroup?, savedInstanceState: Bundle?): View? {
```

```
        tasarim = FragmentBlankBinding.inflate(inflater, container, attachToParent: false)
```

```
        tasarim.buttonny.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }
```

```
        return tasarim.root
```

}

2

3

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

6:47 ☼



## Kişi Kayıt

Kişi Ad

Kişi Tel

KAYDET

```
class KisiKayitFragment : Fragment() {
    private lateinit var tasarim: FragmentKisiKayitBinding
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                               savedInstanceState: Bundle?): View? {
        tasarim = FragmentKisiKayitBinding.inflate(inflater, container, attachToParent: false)

        tasarim.toolbarKisiKayit.title = "Kişi Kayıt"

        tasarim.buttonKaydet.setOnClickListener { it: View!
            val kisi_ad = tasarim.editTextKisiAd.text.toString()
            val kisi_tel = tasarim.editTextKisiTel.text.toString()
            kayit(kisi_ad,kisi_tel)
        }

        return tasarim.root
    }

    fun kayit(kisi_ad:String,kisi_tel:String){
        Log.e( tag: "Kişi Kayıt", msg: "$kisi_ad - $kisi_tel")
    }
}
```

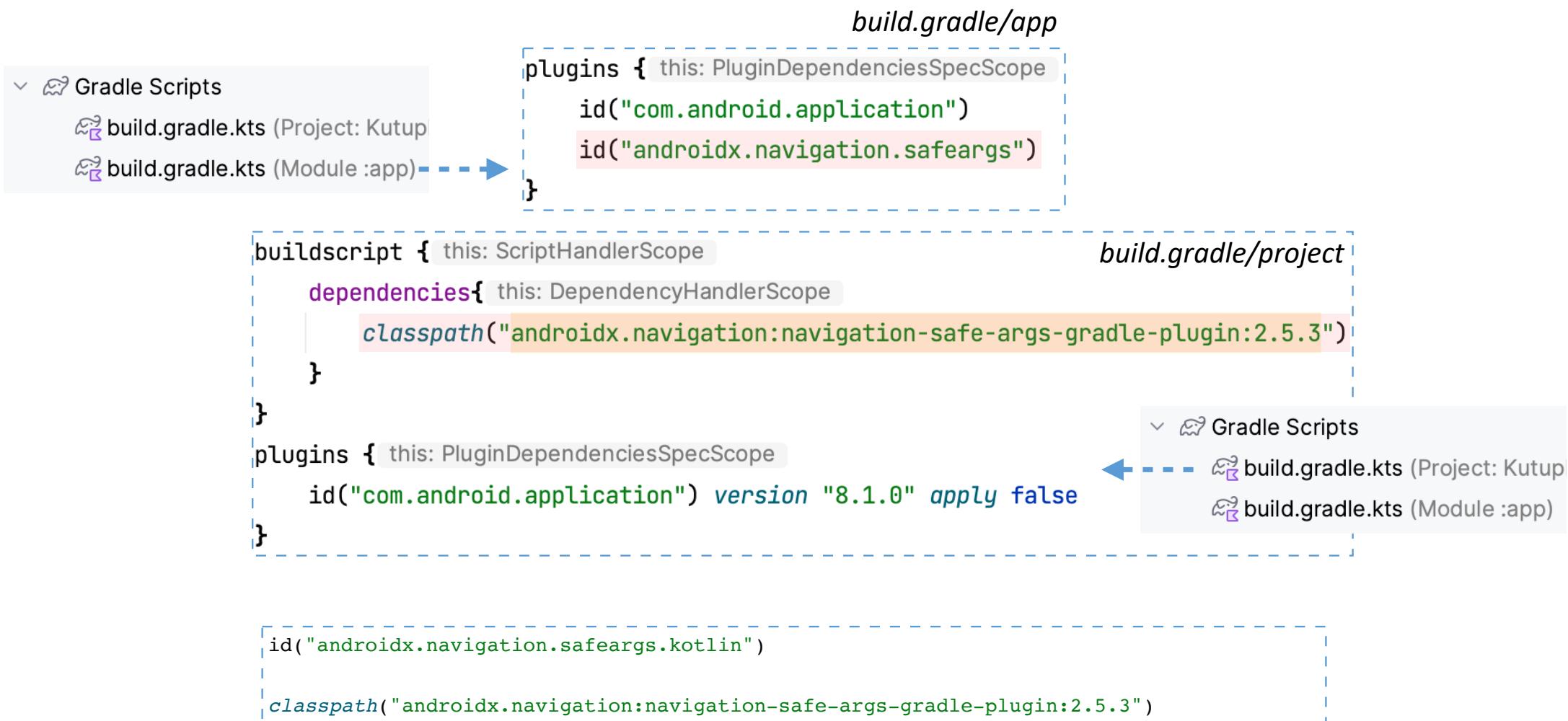
# Uyarı

- Navigation Component kütüphane sürümü ile compile sdk uyumlu değilse kütüphane sürümünü düşürebilirsiniz.

```
dependencies { this: DependencyHandlerScope

    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.navigation:navigation-fragment:2.5.0")
    implementation("androidx.navigation:navigation-ui:2.5.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

# Kurulum



# Gönderilecek Veri Oluşturma

The screenshot shows the Android Studio navigation graph editor. On the left, there are three fragments: 'anasayfaFragment' (Anasafya), 'detayFragment' (Detay), and 'ayarlarFragment' (Ayarlar). An arrow points from 'anasayfaFragment' to 'detayFragment'. The 'detayFragment' screen contains a TextView with the text: *Veri gönderilecek Fragment seçilmelidir.* (The fragment to send data must be selected.)

The right side shows the 'Attributes' panel for 'detayFragment'. It includes:

- Attributes**:
  - fragment: detayFragment
  - id: detayFragment
  - label: fragment\_detay
  - name: DetayFragment (c)
- Arguments**:
  - + ad: string (isim yok)
  - + yas: integer (0)
  - + boy: float (0.0)
  - + bekar: boolean (false)
  - + urun: com.example.navigationcomponentkull
- Actions**: + -
- Deep Links**: + -

A modal dialog titled 'Add Argument' is open, showing the configuration for the 'ad' argument:

- Name: ad
- Type: String
- Array:
- Nullable:
- Default Value: isim yok

At the bottom right, a portion of the generated Java code is visible:

```
data class Kisiler(var kisi_id:Int,  
                    var kisi_ad:String,  
                    var kisi_tel:String) : Serializable {  
}
```

6:47



## Kişi Detay

Kişi Ad

Zeynep

Kişi Tel

2222

GÜNCELLE

```
class KisiDetayFragment : Fragment() {
    private lateinit var tasarim: FragmentKisiDetayBinding
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?): View? {
        tasarim = FragmentKisiDetayBinding.inflate(inflater, container, attachToParent: false)
        tasarim.toolbarKisiDetay.title = "Kişi Detay"

        val bundle: KisiDetayFragmentArgs by navArgs()
        val gelenKisi = bundle.kisi
        tasarim.editTextKisiAd.setText(gelenKisi.kisi_ad)
        tasarim.editTextKisiTel.setText(gelenKisi.kisi_tel)

        tasarim.buttonGuncelle.setOnClickListener { it: View!
            val kisi_ad = tasarim.editTextKisiAd.text.toString()
            val kisi_tel = tasarim.editTextKisiTel.text.toString()
            guncelle(gelenKisi.kisi_id, kisi_ad, kisi_tel)
        }
        return tasarim.root
    }
    fun guncelle(kisi_id: Int, kisi_ad: String, kisi_tel: String) {
        Log.e( tag: "Kişi Güncelle", msg: "$kisi_id - $kisi_ad - $kisi_tel")
    }
}
```

11:39 ☀



## Kişiler



Ahmet



1111

Zeynep



2222

Beyza



3333



```
class AnasayfaFragment : Fragment() {
    private lateinit var tasarim: FragmentAnasayfaBinding
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                               savedInstanceState: Bundle?): View? {
        tasarim = FragmentAnasayfaBinding.inflate(inflater, container, attachToParent: false)

        tasarim.toolbarAnasayfa.title = "Kişiler"

        tasarim.fab.setOnClickListener { it: View!
            Navigation.findNavController(it).navigate(R.id.kisiKayitGecis)
        }

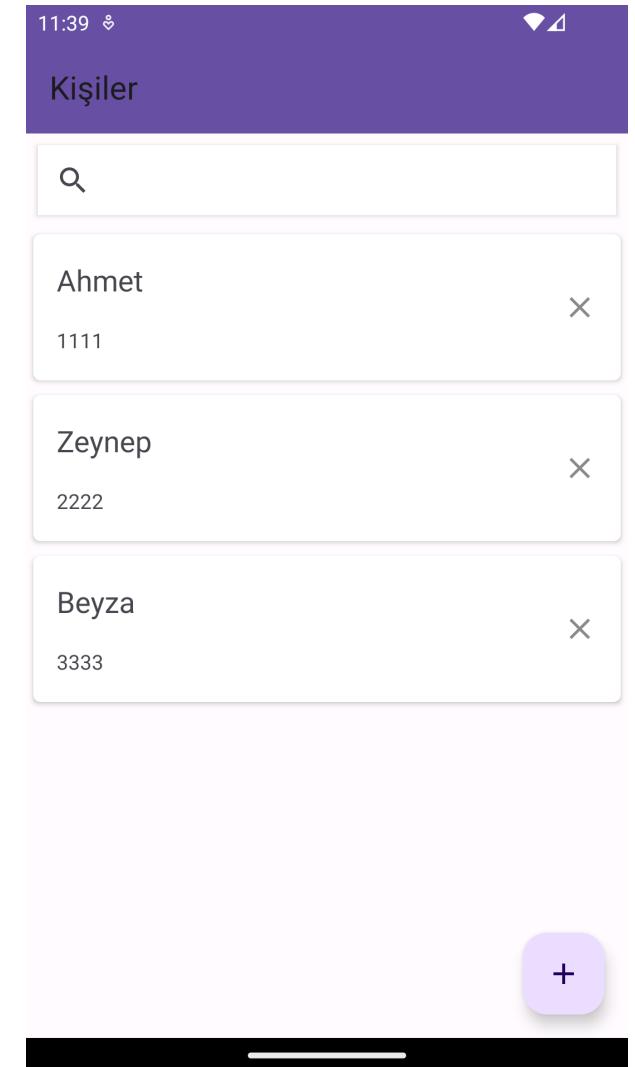
        return tasarim.root
    }

    override fun onResume() {
        super.onResume()
        Log.e( tag: "Kişi Anasayfa", msg: "Dönüldü")
    }
}
```

# SearchView : Arama Özelliği

```
<SearchView
```

```
    android:id="@+id/searchView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:background="@color/white"
    android:elevation="2dp"
    android:queryHint="Ara"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/toolbarAnasayfa" />
```

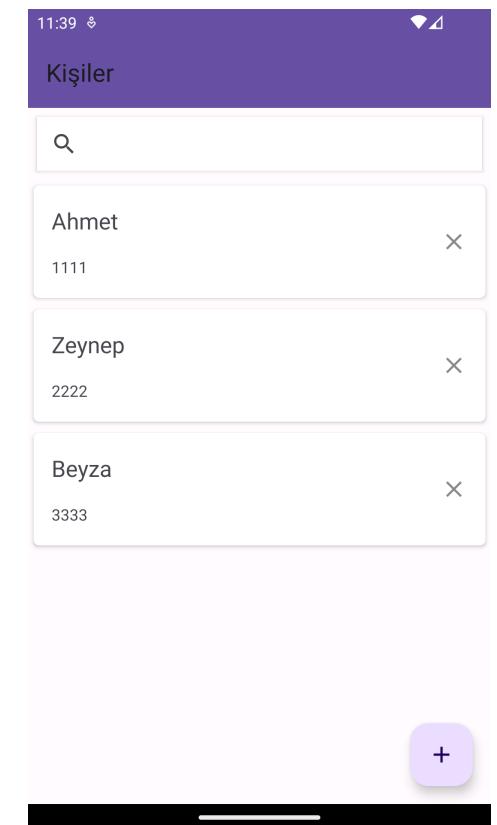
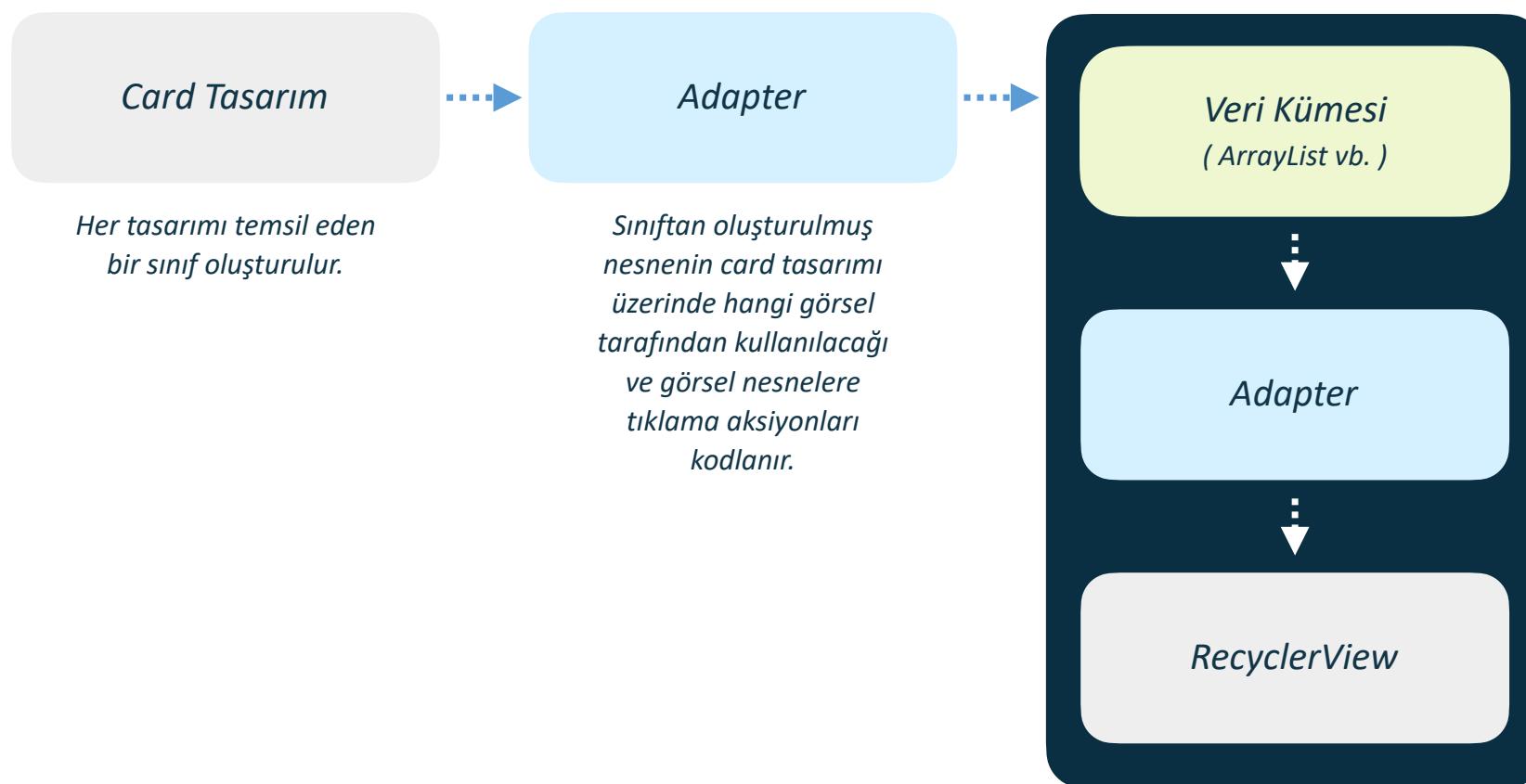


# SearchView : Arama Özelliği

```
binding.searchView.setOnQueryTextListener(object :  
    android.widget.SearchView.OnQueryTextListener {  
        override fun onQueryTextChange(newText: String): Boolean {  
            ara(newText)  
            return true  
        }  
        override fun onQueryTextSubmit(query: String): Boolean {  
            ara(query)  
            return true  
        }  
    })  
  
    return binding.root  
}  
  
fun ara(aramaKelimesi:String){  
    Log.e( tag: "Kişi Ara", aramaKelimesi)  
}  
}
```

# Listeleme

# Listeleme



11:39 ☼



Kişiler



Ahmet

1111



Zeynep

2222

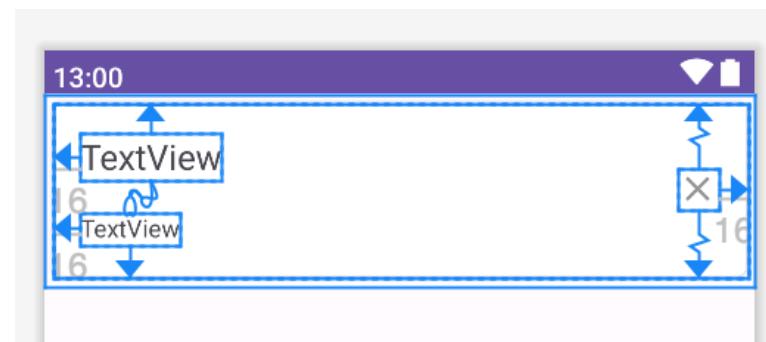


Beyza

3333



# Card Tasarım



|                         |  |
|-------------------------|--|
| layout                  |  |
| activity_main.xml       |  |
| card_tasarim.xml        |  |
| fragment_anasayfa.xml   |  |
| fragment_kisi_detay.xml |  |
| fragment_kisi_kayit.xml |  |

|                         |              |
|-------------------------|--------------|
| Component Tree          |              |
| LinearLayout (vertical) |              |
| cardViewSatir           |              |
| ConstraintLayout        |              |
| imageViewSil            |              |
| textViewKisiTel         |              |
| textViewKisiAd          |              |
| layout_width            | match_parent |
| layout_height           | wrap_content |
| cardCornerRadius        | 5dp          |
| layout_margin           | 5dp          |
| layout_width            | match_parent |
| layout_height           | match_parent |
| textSize                | 20sp         |

# RecyclerView Adapter

```
class KisilerAdapter(var mContext: Context, var kisilerListesi:List<Kisiler>)
    : RecyclerView.Adapter<KisilerAdapter.CardTasarimTutucu>() {
    inner class CardTasarimTutucu(var tasarim: CardTasarimBinding) : RecyclerView.ViewHolder(tasarim.root)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CardTasarimTutucu {
        val layoutInflater = LayoutInflater.from(mContext)
        val tasarim = CardTasarimBinding.inflate(layoutInflater, parent, attachToParent: false)
        return CardTasarimTutucu(tasarim)
    }

    override fun onBindViewHolder(holder: CardTasarimTutucu, position: Int) {
        val kisi = kisilerListesi.get(position)
        val t = holder.tasarim
        t.textViewKisiAd.text = kisi.kisi_ad
        t.textViewKisiTel.text = kisi.kisi_tel
        t.cardViewSatir.setOnClickListener { it: View!
            val gecis = AnasayfaFragmentDirections.kisiDetayGecis(kisi = kisi)
            Navigation.findNavController(it).navigate(gecis)
        }
        t.imageViewSil.setOnClickListener { it: View!
            Snackbar.make(it, text: "${kisi.kisi_ad} silinsin mi?", Snackbar.LENGTH_LONG)
                .setAction( text: "EVET"){ Log.e( tag: "Kişi Sil",kisi.kisi_id.toString() ) }.show()
        }
    }

    override fun getItemCount(): Int {
        return kisilerListesi.size
    }
}
```

# Fragment Kodlamasi

```
class AnasayfaFragment : Fragment() {
    private lateinit var binding: FragmentAnasayfaBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentAnasayfaBinding.inflate(inflater, container, attachToParent: false)

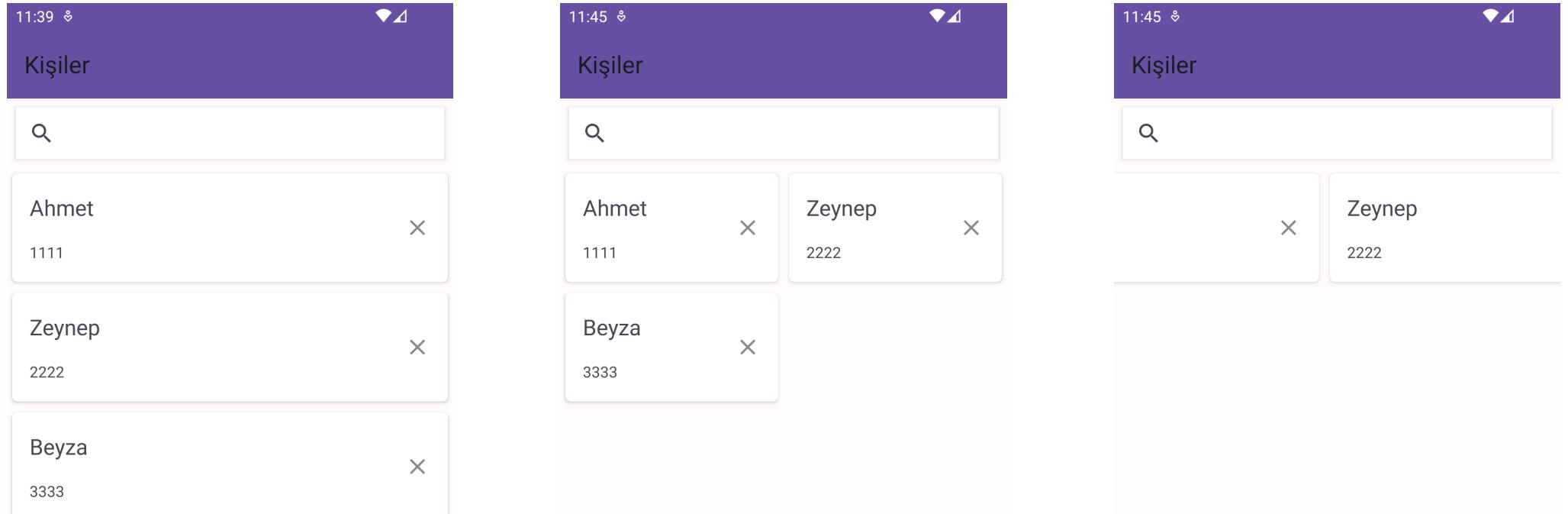
        binding.toolbarAnasayfa.title = "Kişiler"

        tasarim.rv.layoutManager = LinearLayoutManager(requireContext())

        val kisilerListesi = ArrayList<Kisiler>()
        val k1 = Kisiler( kisi_id: 1, kisi_ad: "Ahmet", kisi_tel: "1111")
        val k2 = Kisiler( kisi_id: 2, kisi_ad: "Zeynep", kisi_tel: "2222")
        val k3 = Kisiler( kisi_id: 3, kisi_ad: "Beyza", kisi_tel: "3333")
        kisilerListesi.add(k1)
        kisilerListesi.add(k2)
        kisilerListesi.add(k3)

        val adapter = KisilerAdapter(requireContext(),kisilerListesi)
        tasarim.rv.adapter = adapter
    }
}
```

# RecyclerView Görünümleri



```
tasarim.rv.setLayoutManager =  
    LinearLayoutManager(requireContext())
```

+

```
tasarim.rv.setLayoutManager =  
    StaggeredGridLayoutManager(spanCount: 2,  
                                StaggeredGridLayoutManager.VERTICAL)
```

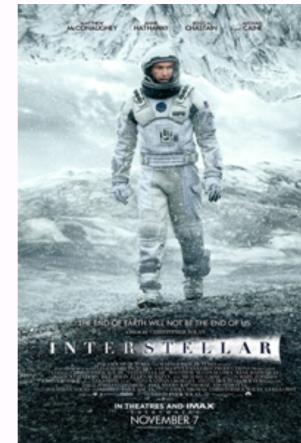
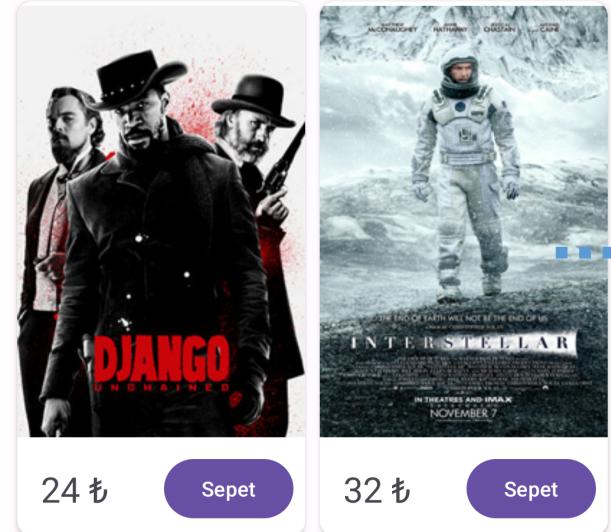
+

```
tasarim.rv.setLayoutManager =  
    StaggeredGridLayoutManager(spanCount: 1,  
                                StaggeredGridLayoutManager.HORIZONTAL)
```

+

# Düzenli RecyclerView

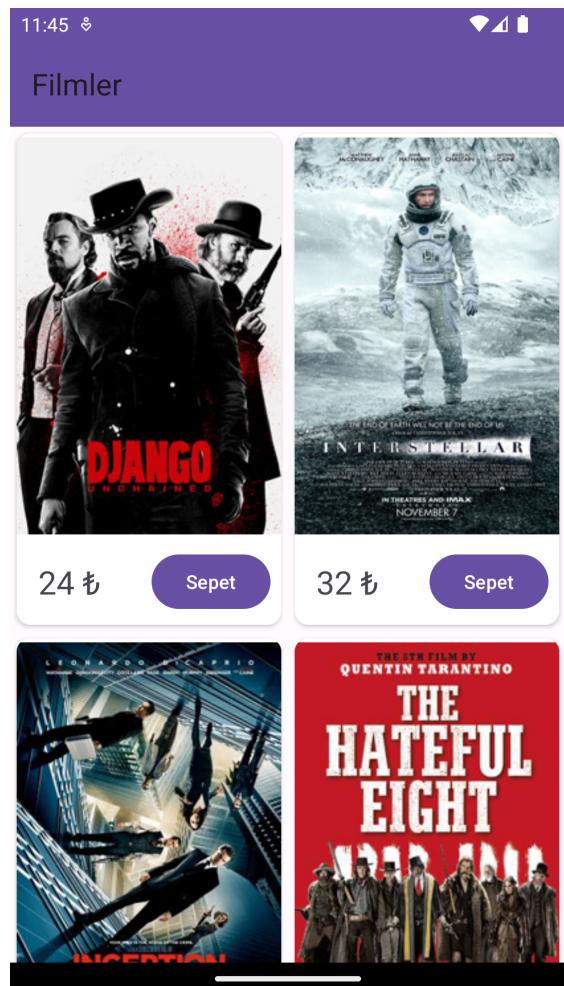
# Film Uygulaması



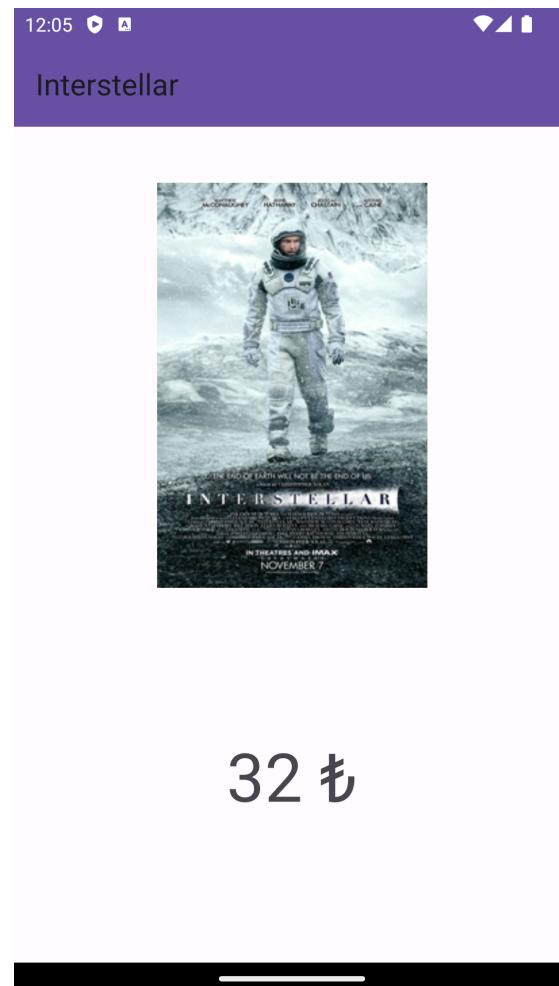
32 ₺

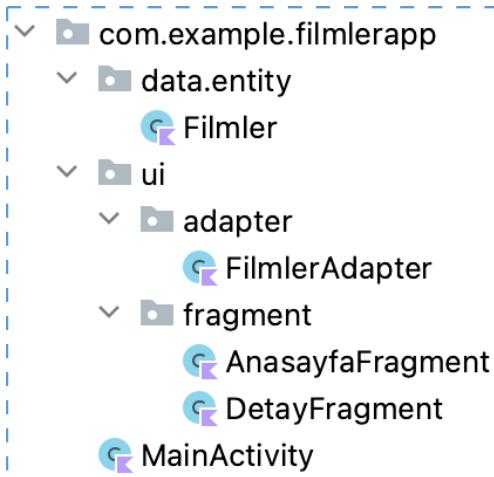
# Fragment Oluşturma

*AnasayfaFragment*



*DetayFragment*

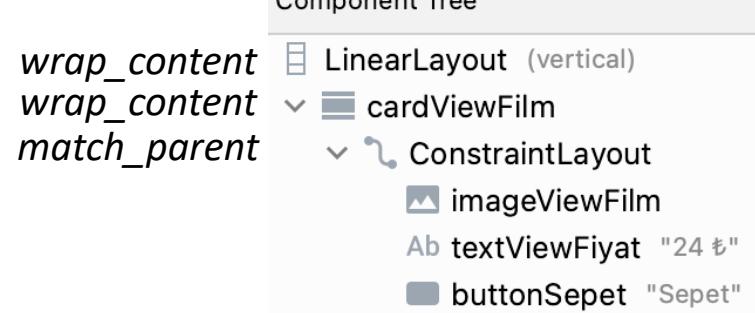
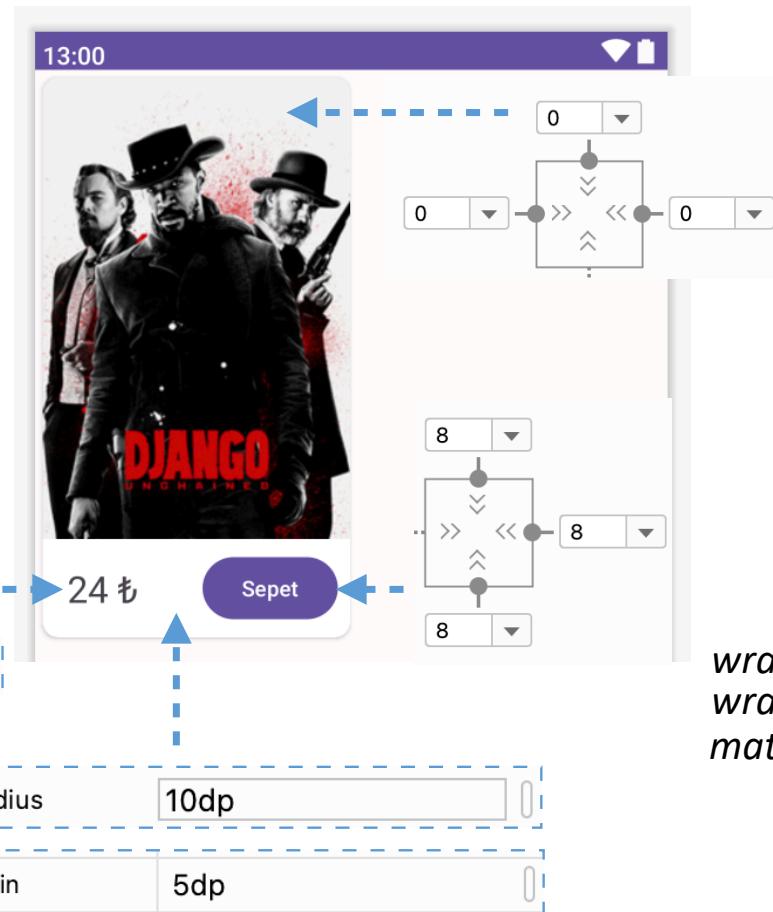




```

data class Filmler(var id:Int,
                   var ad:String,
                   var resim:String,
                   var fiyat:Int) : Serializable

```



```
class FilmlerAdapter(var mContext: Context, var filmlerListesi: List<Filmler>)
    : RecyclerView.Adapter<FilmlerAdapter.CardTasarimTutucu>() {
    inner class CardTasarimTutucu(var tasarim: CardTasarimBinding) : RecyclerView.ViewHolder(tasarim.root)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CardTasarimTutucu {
        val tasarim = CardTasarimBinding.inflate(LayoutInflater.from(mContext), parent, attachToParent: false)
        return CardTasarimTutucu(tasarim)
    }

    override fun onBindViewHolder(holder: CardTasarimTutucu, position: Int) {
        val film = filmlerListesi.get(position)
        val t = holder.tasarim
        t.imageViewFilm.setImageResource(
            mContext.resources.getIdentifier(film.resim, defType: "drawable", mContext.packageName))

        t.textViewFiyat.text = "${film.fiyat} ₺"
        t.cardViewFilm.setOnClickListener { it: View!
            val gecis = AnasayfaFragmentDirections.detayGecis(film = film)
            Navigation.findNavController(it).navigate(gecis)
        }
        t.buttonSepet.setOnClickListener { it: View!
            Snackbar.make(it, text: "${film.ad} sepete eklendi", Snackbar.LENGTH_LONG).show()
        }
    }

    override fun getItemCount(): Int { return filmlerListesi.size }
}
```

```

class AnasayfaFragment : Fragment() {
    private lateinit var binding: FragmentAnasayfaBinding
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?): View? {
        binding = FragmentAnasayfaBinding.inflate(inflater, container, attachToParent: false)
        binding.toolbarAnasayfa.title = "Filmler"

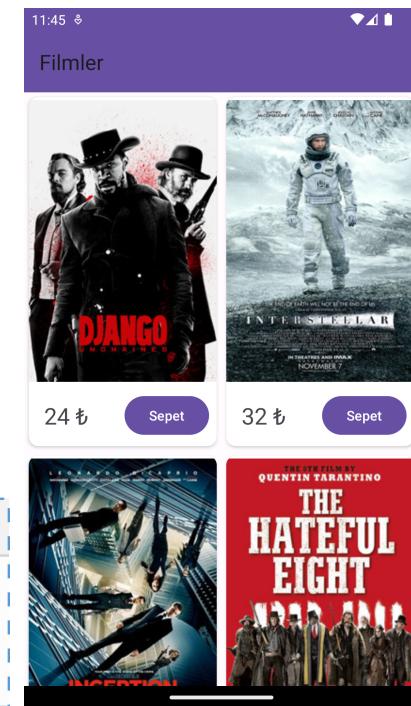
        binding.filmRv.layoutManager =
            StaggeredGridLayoutManager(spanCount: 2, StaggeredGridLayoutManager.VERTICAL)

        val filmlerListesi = ArrayList<Filmler>()
        val f1 = Filmler(id: 1, ad: "Django", resim: "django", fiyat: 24)
        val f2 = Filmler(id: 2, ad: "Interstellar", resim: "interstellar", fiyat: 32)
        val f3 = Filmler(id: 3, ad: "Inception", resim: "inception", fiyat: 16)
        val f4 = Filmler(id: 4, ad: "The Hateful Eight", resim: "thehatefuleight", fiyat: 28)
        val f5 = Filmler(id: 5, ad: "The Pianist", resim: "thepianist", fiyat: 18)
        val f6 = Filmler(id: 6, ad: "Anadoluda", resim: "anadoluda", fiyat: 10)
        filmlerListesi.add(f1)
        filmlerListesi.add(f2)
        filmlerListesi.add(f3)
        filmlerListesi.add(f4)
        filmlerListesi.add(f5)
        filmlerListesi.add(f6)

        val adapter = FilmlerAdapter(requireContext(), filmlerListesi)
        binding.filmRv.adapter = adapter

        return binding.root
    }
}

```



```

val filmlerListesi = ArrayList<Filmler>()
val f1 = Filmler(1,"Django","django",24)
val f2 = Filmler(2,"Interstellar","interstellar",32)
val f3 = Filmler(3,"Inception","inception",16)
val f4 = Filmler(4,"The Hateful Eight","thehatefuleight",28)
val f5 = Filmler(5,"The Pianist","thepianist",18)
val f6 = Filmler(6,"Anadoluda","anadoluda",10)
filmlerListesi.add(f1)
filmlerListesi.add(f2)
filmlerListesi.add(f3)
filmlerListesi.add(f4)
filmlerListesi.add(f5)
filmlerListesi.add(f6)

```

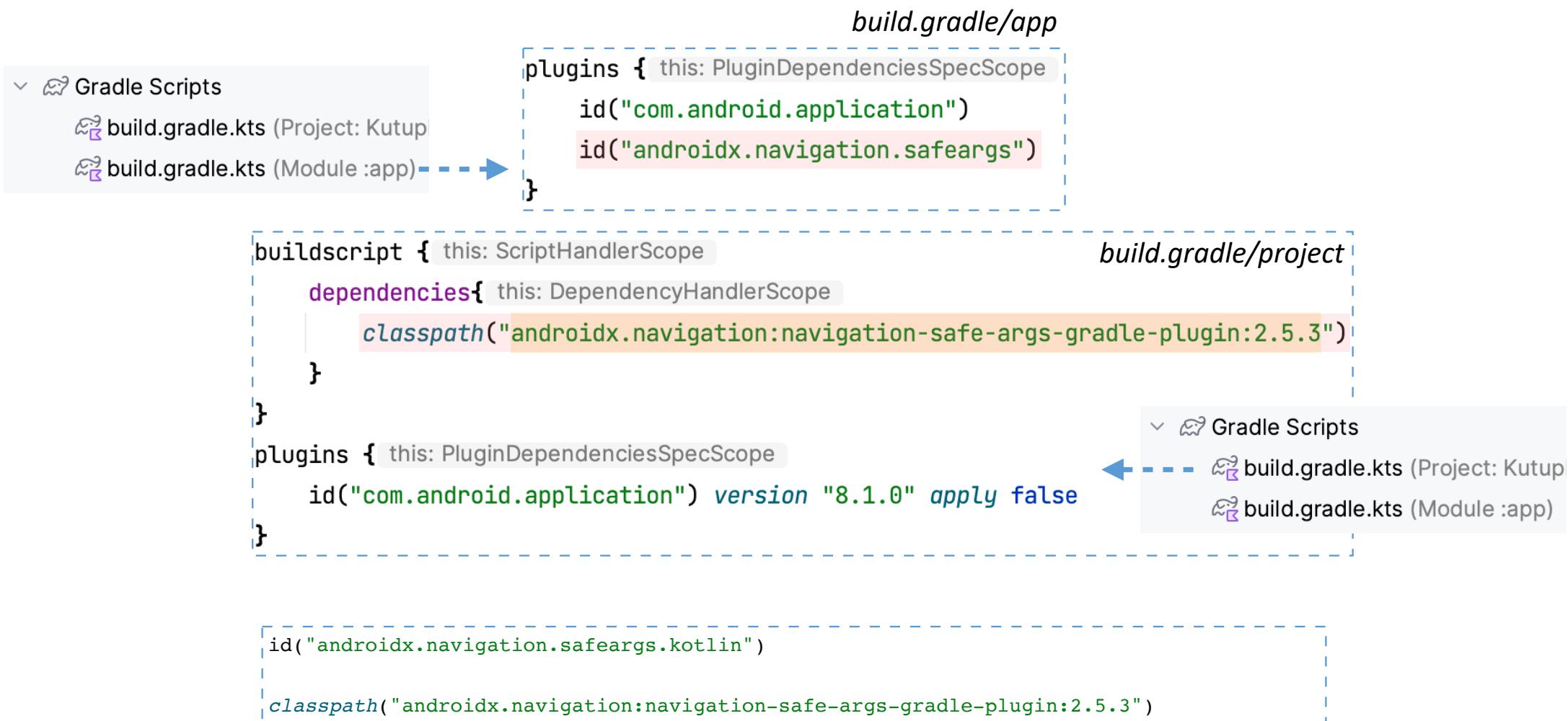
# Uyarı

- Navigation Component kütüphane sürümü ile compile sdk uyumlu değilse kütüphane sürümünü düşürebilirsiniz.

```
dependencies { this: DependencyHandlerScope

    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.navigation:navigation-fragment:2.5.0")
    implementation("androidx.navigation:navigation-ui:2.5.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

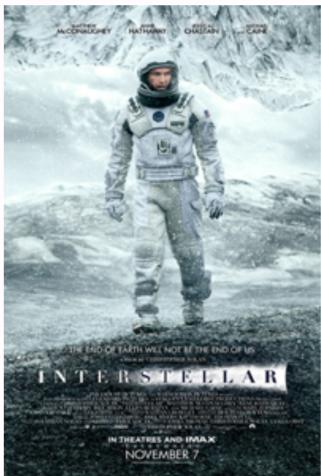
# Kurulum



12:05



Interstellar



32 ₺

```
class DetayFragment : Fragment() {
    private lateinit var binding: FragmentDetayBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentDetayBinding.inflate(inflater, container, attachToParent: false)

        val bundle: DetayFragmentArgs by navArgs()
        val film = bundle.film

        binding.toolbarDetay.title = film.ad

        binding.ivFilm.setImageResource(
            resources.getIdentifier(film.resim, defType: "drawable", requireContext().packageName))

        binding.tvFiyat.text = "${film.fiyat} ₺"

        return binding.root
    }
}
```

Component Tree

- ConstraintLayout
  - toolbarDetay
  - ivFilm
  - tvFiyat "50 ₺" textSize 50sp

# Uygulama Mimarisi

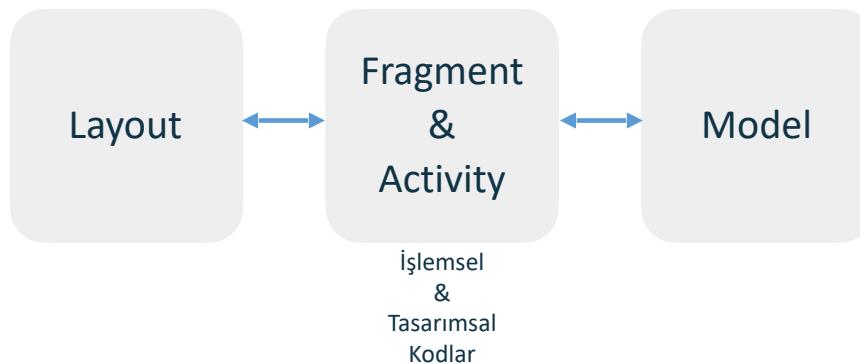
# MVVM Clean Architecture

# MVVM Clean Architecture

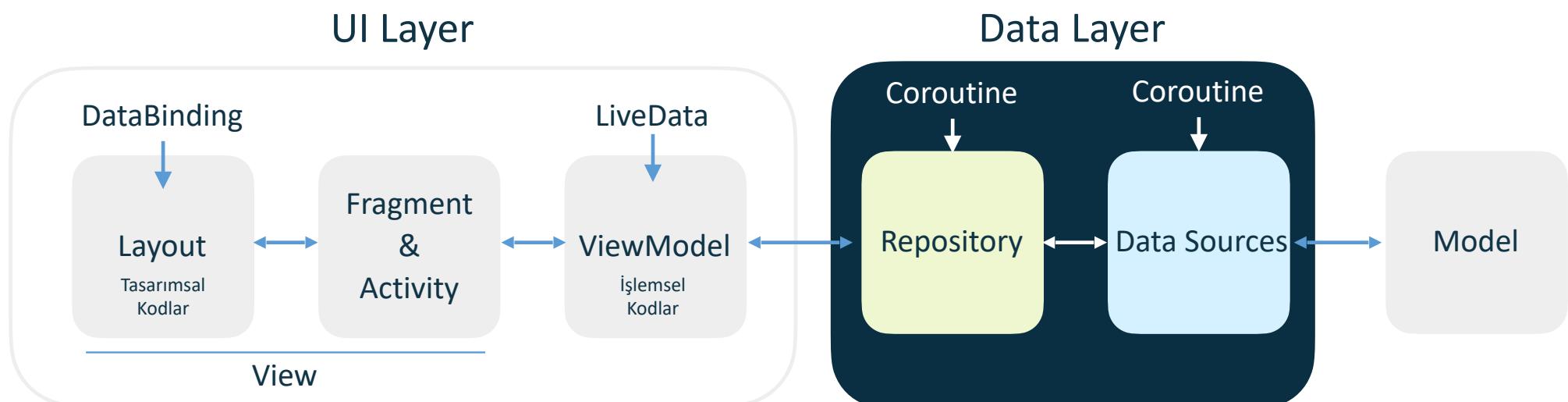
Bu mimari sayesinde daha profesyonel kodlamalar yapabiliriz.  
Özellikle büyük projelerde oluşan kodlama karmaşasını azaltır.

Mimarinin amacı proje içerisindeki işlemsel kodları ve tasarımsal kodları ayırip daha modüller ve düzenli hale getirmektir.

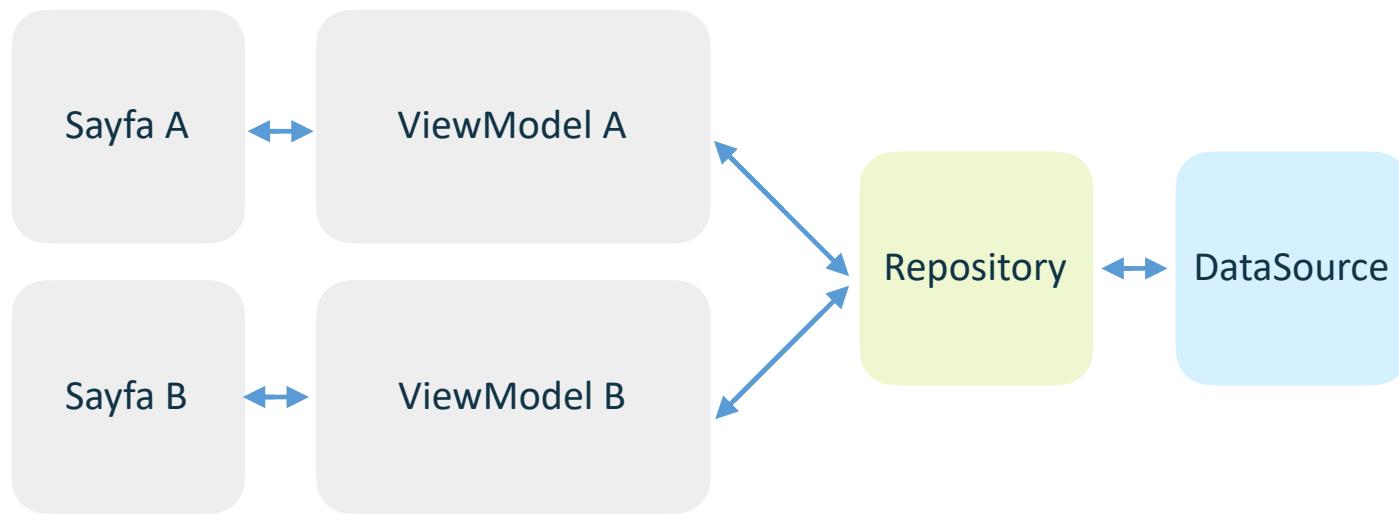
# Klasik Yöntem



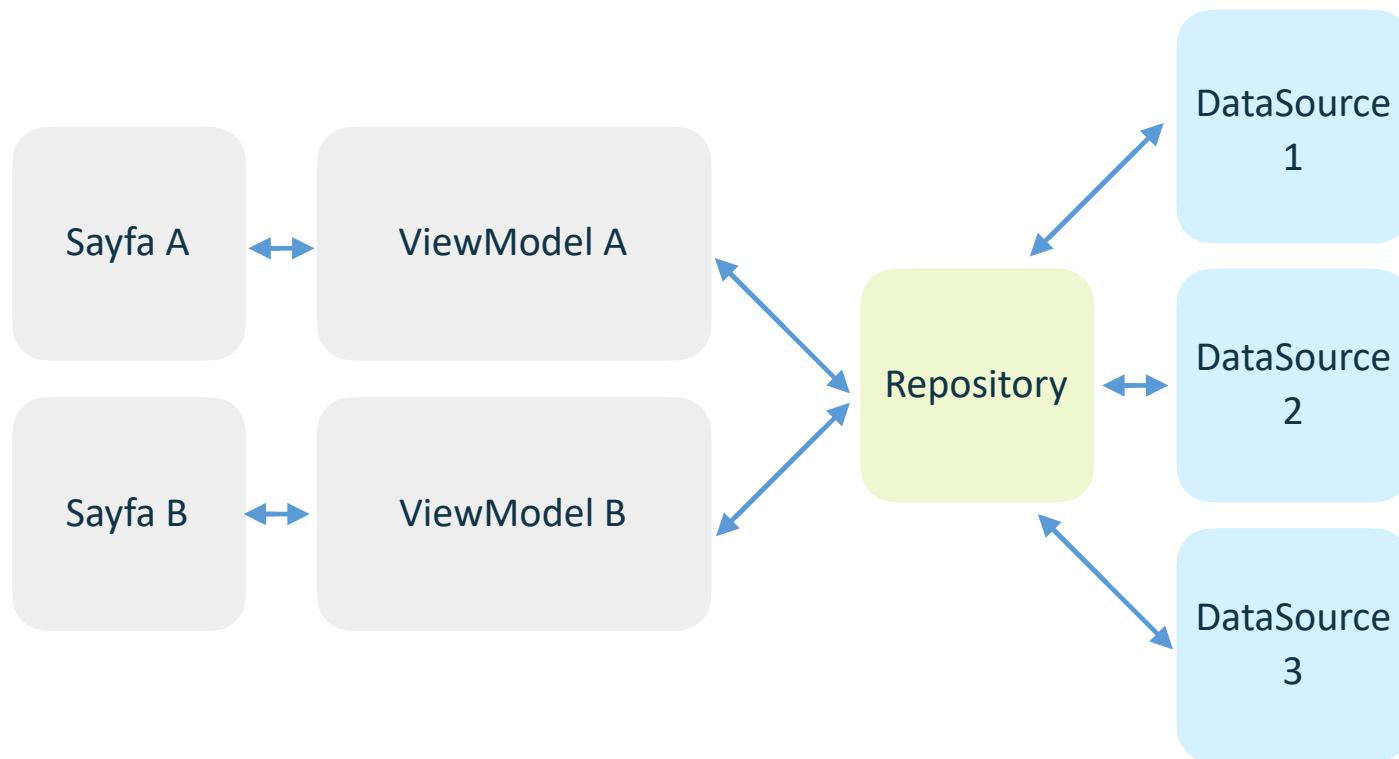
# MVVM Clean Architecture



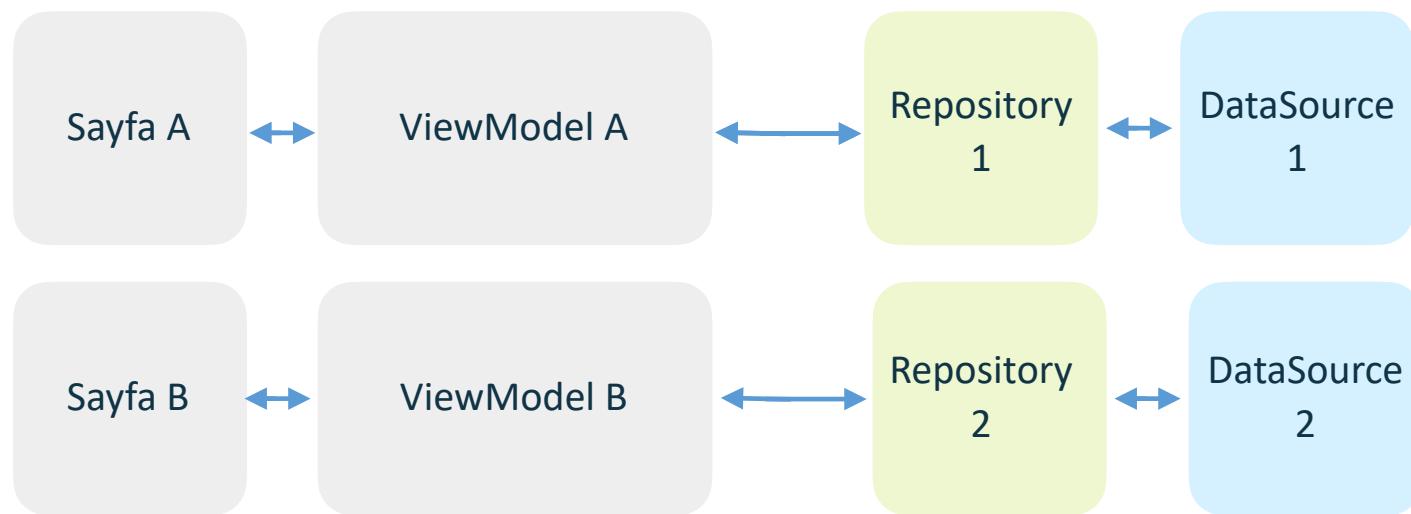
# Farklı Senaryolar



# Farklı Senaryolar



# Farklı Senaryolar



# Data Binding

# DataBinding

Temel olarak tasarım alanına veri göndermektir.

Bu veriler değişken veya nesne olabilir.

Sınıftan tasarım alanına değişken veya nesne göndererek arayüzde bunları kullanabiliriz.

Daha az kodlama ile tasarımını ve sınıf modelini ile yönetebiliriz.

Tasarım üzerindeki görsel nesneleri yönetmek için kolaylık sağlar.

Kullanıcı arayüzü manuel müdahale olmadan otomatik olarak güncellenmesini sağlayan bir kütüphanedir.

Klasik yöntemlere göre daha hızlı çalışır ve kod kalabalığından kurtarır.

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

*Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.*

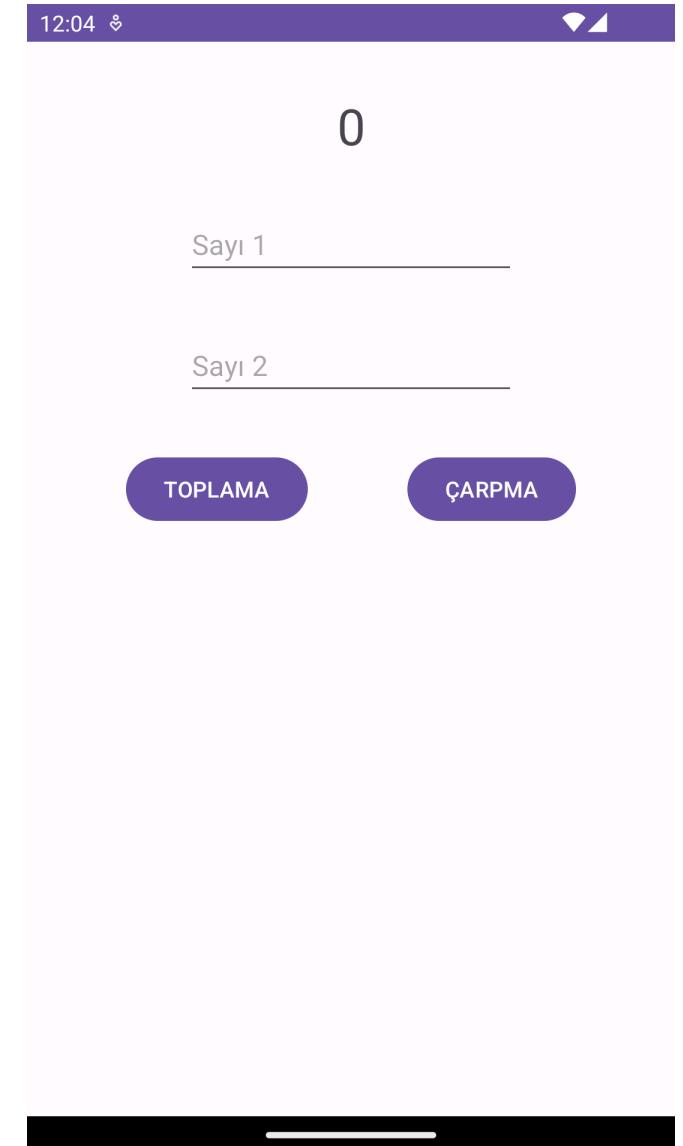
```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(LayoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3

# Klasik Yöntem

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        binding.textViewSonuc.text = "0"  
  
        binding.buttonToplama.setOnClickListener { it: View!  
            val alinanSayi1 = binding.editTextSayi1.text.toString()  
            val alinanSayi2 = binding.editTextSayi2.text.toString()  
  
            val sayi1 = alinanSayi1.toInt()  
            val sayi2 = alinanSayi2.toInt()  
            val toplam = sayi1 + sayi2  
  
            binding.textViewSonuc.text = toplam.toString()  
        }  
  
        binding.buttonCarpma.setOnClickListener { it: View!  
            val alinanSayi1 = binding.editTextSayi1.text.toString()  
            val alinanSayi2 = binding.editTextSayi2.text.toString()  
  
            val sayi1 = alinanSayi1.toInt()  
            val sayi2 = alinanSayi2.toInt()  
            val carpma = sayi1 * sayi2  
  
            binding.textViewSonuc.text = carpma.toString()  
        }  
    }  
}
```



# Kütüphane Kurulumu

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the **app** module. It includes **manifests**, **java** (containing **MainActivity**), **res**, and **Gradle Scripts** (containing **build.gradle** files for the project and module, and **gradle-wrapper.properties**).
- Build Scripts:** The **build.gradle** file for the module is open in the editor. The code is as follows:

```
plugins {
    id 'com.android.application'

}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    buildFeatures {
        dataBinding true
    }
}
```

```
buildFeatures {
    dataBinding true
}
```

# Tasarımsal Kurulum

- Tasarım alt yapımızın dışına layout tag'i eklenmelidir ve tasarım alt yapısının sahip olduğu xmlns özellikleri bu tag'e aktarılmalıdır.

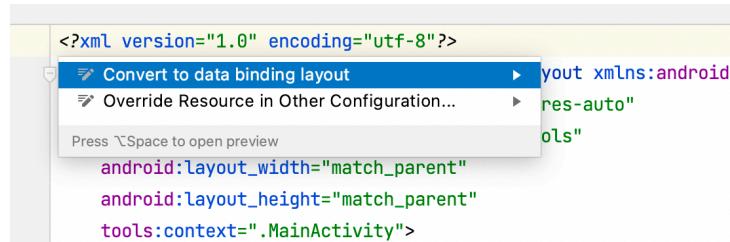
```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

        <TextView
            android:id="@+id/textViewSonuc"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="32dp"
            android:text="TextView"
            android:textSize="34sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <Button
            android:id="@+id/buttonCarpma"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="32dp"
            android:text="ÇARPMA"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toEndOf="@+id/buttonToplama"
            app:layout_constraintTop_toBottomOf="@+id/editTextSayi2" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</layout>
```

KISAYOL : Tasarımın herhangi bir noktasında alt + enter yaparak seçim yapılınrsa otomatik layout tagleri eklenir.



# DataBinding ile Tasarıma Erişim : Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)
```

*Önemli : Tasarım Sınıfı ismi tasarım ismine göre otomatik oluşturulur.*

*activity\_main olduğu için ActivityMainBinding olur.*

Tasarımı kodlama olarak sınıfı aktardık ve bu nesne sayesinde tasarımındaki görsel nesnelere erişebiliriz.

# DataBinding ile Tasarıma Erişim : Fragment

```
class KisiKayitFragment : Fragment() {  
  
    private lateinit var tasarim:FragmentKisiKayitBinding  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
  
        tasarim = DataBindingUtil.inflate(inflater,  
            R.layout.fragment_kisi_kayit, container, attachToParent: false)  
  
        tasarim.kisiKayitFragment = this  
        tasarim.kisiKayitToolbarBaslik = "Kişi Kayıt"  
  
        return tasarim.root  
    }  
}
```

# Görsel Nesnelere Erişim

Tasarım nesnesi ile  
tasarımda yer alan  
görsel nesnelere  
idleri ile erişebiliriz.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim:ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)  
  
        → tasarim.textViewSonuc.text = "0"  
  
        tasarim.buttonToplama.setOnClickListener { it: View!  
            val alinanSayi1 = tasarim.editTextSayi1.text.toString()  
            val alinanSayi2 = tasarim.editTextSayi2.text.toString()  
  
            val sayi1 = alinanSayi1.toInt()  
            val sayi2 = alinanSayi2.toInt()  
  
            val toplam = sayi1 + sayi2  
  
            tasarim.textViewSonuc.text = toplam.toString()  
    }  
}
```

# Event Handle

- Button'a tıklanılma gibi işlemleri daha pratik hale getirebiliriz.
- Tasarım Alanından Sınıf Metodlarını Tetikleme

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable name="mainActivityNesnesi" type="com.example.mvvmkullanimi.MainActivity" />
    </data>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

        <TextView
            android:id="@+id/textViewSonuc"
            android:layout_width="wrap_content"
            ...>
```

Tasarım alanında  
kullanmak için nesne  
isimlendirmesi

Hangi sınıftaki metodu çalıştırırmak  
istiyorsak o sınıfın paketi ve adı  
Nesne yardımıyla sınıftaki  
metoda erişiyoruz ve tıklanılma  
özellikine ekliyoruz.

Not : Tasarım alanında variable oluşturduysak bu  
değere tasarımın kullanıldığı yerden activity veya  
sınıftan veri aktarmamız şart. Aksi halde veriyi  
kullanamayız.

```
<Button
    android:id="@+id/buttonToplama"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="TOPLAMA"
    android:onClick="@{() -> mainActivityNesnesi.buttonToplamaTikla()}"
    app:layout_constraintEnd_toStartOf="@+id/buttonCarpma"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextSayi2" />

<Button
    android:id="@+id/buttonCarpma"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="ÇARPMA"
    android:onClick="@{() -> mainActivityNesnesi.buttonCarpmaTikla()}"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/buttonToplama"
    app:layout_constraintTop_toBottomOf="@+id/editTextSayi2" />
```

# Event Handle

- Buttona tıklanılma gibi işlemleri daha pratik hale getirebiliriz. Tasarım Alanından Sınıf Metodlarını Tetikleme

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim:ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)  
        tasarim.mainActivityNesnesi = this  
  
        tasarim.textViewSonuc.text = "0"  
    }  
  
}
```

```
fun buttonToplamaTikla(){  
    val alinanSayi1 = tasarim.editTextSayi1.text.toString()  
    val alinanSayi2 = tasarim.editTextSayi2.text.toString()  
  
    val sayi1 = alinanSayi1.toInt()  
    val sayi2 = alinanSayi2.toInt()  
  
    val toplam = sayi1 + sayi2  
  
    tasarim.textViewSonuc.text = toplam.toString()  
}
```

Bu tanımlama ile tasarımında oluşturduğumuz activity nesnesine tasarımından activity'e erişim yetkisi veriyoruz. Bu tanımlama olmaz ise buttonlar metodları çalıştırılamaz.

metodun ismi tasarımında oluşturulan nesne ismi ile aynı olmalıdır.

```
fun buttonCarpmaTikla(){  
    val alinanSayi1 = tasarim.editTextSayi1.text.toString()  
    val alinanSayi2 = tasarim.editTextSayi2.text.toString()  
  
    val sayi1 = alinanSayi1.toInt()  
    val sayi2 = alinanSayi2.toInt()  
  
    val carpma = sayi1 * sayi2  
  
    tasarim.textViewSonuc.text = carpma.toString()  
}
```

# Event Handle : Parametre ile Veri Gönderme

- Tasarım alanından activity'e veri göndermek isteniyorsa kullanılabilir.

```
<Button  
    android:id="@+id/buttonToplama"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="TOPLAMA"  
    android:onClick="@{()> mainActivityNesnesi.buttonToplamaTikla(editTextSayi1.getText().toString(),editTextSayi2.getText().toString())}"  
    app:layout_constraintEnd_toStartOf="@+id/buttonCarpma"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/editTextSayi2" />  
  
<Button  
    android:id="@+id/buttonCarpma"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="ÇARPMA"  
    android:onClick="@{()> mainActivityNesnesi.buttonCarpmaTikla(editTextSayi1.getText().toString(),editTextSayi2.getText().toString())}"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toEndOf="@+id/buttonToplama"  
    app:layout_constraintTop_toBottomOf="@+id/editTextSayi2" />  
/>/androidx.constraintlayout.widget.ConstraintLayout>  
</layout>
```

TASARIM  
AŞAMASI

1

Tasarım alanından  
gönderilen parametre

Tasarım alanından  
gönderilen parametre

*Not : getText() metodu kullanılmalı kotlin metodunda  
hata oluşuyor.*

# Event Handle : Parametre ile Veri Gönderme

- Tasarım alanından activity'e veri göndermek isteniyorsa kullanılabilir.

```
fun buttonToplamaTikla(alinanSayi1:String,alinanSayi2:String){  
    val sayi1 = alinanSayi1.toInt()  
    val sayi2 = alinanSayi2.toInt()  
  
    val toplam = sayi1 + sayi2  
  
    tasarim.textViewSonuc.text = toplam.toString()  
}
```

KODLAMA  
AŞAMASI

2

Tasarım alanından gönderilen parametreyi alma

```
fun buttonCarpmaTikla(alinanSayi1:String,alinanSayi2:String){  
    val sayi1 = alinanSayi1.toInt()  
    val sayi2 = alinanSayi2.toInt()  
  
    val carpma = sayi1 * sayi2  
  
    tasarim.textViewSonuc.text = carpma.toString()  
}
```

# Tasarım Alanına Veri ( Değişken ) Gönderme

- Data Binding yapısının temel işlemi tasarım alanına veri göndermektir.
- Bunun en basit yolu ise değişken gönderme işlemidir.

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)  
        tasarim.mainActivityNesnesi = this  
        tasarim.hesaplamaSonucu = "0"  
    }  
  
    fun buttonToplamaTikla(alinanSayi1:String,alinanSayi2:String){  
        val sayi1 = alinanSayi1.toInt()  
        val sayi2 = alinanSayi2.toInt()  
        val toplam = sayi1 + sayi2  
        tasarim.hesaplamaSonucu = toplam.toString()  
    }  
  
    fun buttonCarpmaTikla(alinanSayi1:String,alinanSayi2:String){  
        val sayi1 = alinanSayi1.toInt()  
        val sayi2 = alinanSayi2.toInt()  
        val carpma = sayi1 * sayi2  
        tasarim.hesaplamaSonucu = carpma.toString()  
    }  
}
```

Tasarım alanından gönderilen değişken set metodu ile aktarılır.  
Bu metodun ismi tasarım alanındaki değişken adına göre otomatik olur.

```
<?xml version="1.0" encoding="utf-8"?>  
<layout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        xmlns:tools="http://schemas.android.com/tools">  
  
    <data>  
        <variable name="mainActivityNesnesi" type="com.example.mvvmkullanimi.MainActivity" />  
        <variable name="hesaplamaSonucu" type="String" />  
    </data>  
  
    <androidx.constraintlayout.widget.ConstraintLayout  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        tools:context=".MainActivity">  
  
        <TextView  
            android:id="@+id/textViewSonuc"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_marginTop="32dp"  
            android:text="@{hesaplamaSonucu}"  
            android:textSize="34sp" />  
    </androidx.constraintlayout.widget.ConstraintLayout>
```

Tasarım alanında değişken  
oluşturma işlemi.  
name : Değişken adı  
type : Değişken türünü belirtir.

Değişkenin arayüzde  
kullanılması

Sınıf içinde değişkene set  
metodu ile veri aktarıldığı  
anda arayüzde görülmüş olur.

# Tasarım Alanında Parantez İçinde Kodlama Yapma

- Tür dönüşümü , koşul gibi kodlamaları tasarım alanında yapabiliriz.
- Görsel nesnenin her özelliğinde kodlama yapılabilir.

The screenshot shows the Android Studio interface with two code snippets side-by-side.

**Left Snippet (activity\_main.xml):**

```
<TextView  
    android:id="@+id/textViewSonuc"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="@{hesaplamaSonucu}"  
    android:textColor="@{Integer.parseInt(hesaplamaSonucu) > 10 ? @color/kirmizi : @color/mavi }"  
    android:visibility="@{Integer.parseInt(hesaplamaSonucu) > 0 ? View.VISIBLE : View.INVISIBLE}"  
    android:textSize="@{hesaplamaSonucu.length() > 1 ? @dimen/boyut2 : @dimen/boyut1 }"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

**Right Snippet (colors.xml):**

```
1  <?xml version="1.0" encoding="utf-8"?>  
2  <resources>  
3  ■ <color name="black">#FF000000</color>  
4  ■ <color name="white">#FFFFFF</color>  
5  ■ <color name="mavi">#2196F3</color>  
6  ■ <color name="kirmizi">#F44336</color>  
7  </resources>
```

# Tasarım alanına istediğimiz sınıfı import edebiliriz.

- Örnek görsel nesnenin görünürlüğü için View nesnesi import etmek gibi

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable name="mainActivityNesnesi" type="com.example.mvvmkullanimi.MainActivity" />
        <variable name="hesaplamaSonucu" type="String" />
        <import type="android.view.View"/>
    </data>    Import işlemi

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

        <TextView
            android:id="@+id/textViewSonuc"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="32dp"
            android:text="@{hesaplamaSonucu}"
            android:textColor="@{Integer.parseInt(hesaplamaSonucu) > 10 ? @color/purple_500 : @color/teal_700}"
            android:visibility="@{Integer.parseInt(hesaplamaSonucu) > 0 ? View.VISIBLE : View.INVISIBLE}"Import edilen nesnenin kullanımı
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
```

# Boyutsal Çalışma

- dp veya sp birimleri ile boyut çalışması yapabiliriz.

The screenshot shows the Android Studio project structure on the left and the XML code editor on the right. In the project tree, the 'values' folder is expanded, showing 'colors.xml', 'dimens.xml' (selected), and 'strings.xml'. Below the project tree, the XML code editor displays the following code:

```
<resources>
    <dimen name="boyut1">50sp</dimen>
    <dimen name="boyut2">100sp</dimen>
</resources>
```

In the XML code editor, the 'dimens.xml' file contains the following code:

```
<TextView
    android:id="@+id/textViewSonuc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="@{hesaplamaSonucu}"
    android:textColor="@{Integer.parseInt(hesaplamaSonucu) > 10 ? @color/purple_500 : @color/teal_700}"
    android:visibility="@{Integer.parseInt(hesaplamaSonucu) > 0 ? View.VISIBLE : View.INVISIBLE}"
    android:textSize="@{hesaplamaSonucu.length() > 1 ? @dimen/boyut2 : @dimen/boyut1 }"
    app:layout_constraintEnd_toEndOf="parent"   Sonuç boyutu 1 karakterden büyükse
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

The code editor highlights several parts of the XML code, including the 'dimens.xml' file path, the 'dimen' elements, and the conditional logic in the TextView attributes. A red box surrounds the 'textSize' attribute's value expression. A red arrow points from the explanatory text 'Sonuç boyutu 1 karakterden büyükse' to the value expression '@{hesaplamaSonucu.length() > 1 ? @dimen/boyut2 : @dimen/boyut1 }'.

# ViewModel - Kotlin

# ViewModel

- Viewmodel'in asıl amacı arayüzü besleyecek verileri organize etmektir.
- Ayrıca sayfa rotasyonlarında ( Dikey ve Yatay konuma geçişte ) veriyi korumaktadır.
- Arayüzü ( Activity ) verilerden ayırarak daha kontrollü yapı oluşturabiliriz.
- ViewModel içinde sayfa üzerinde yapılacak işlemleri bulundurabiliriz.
- Activity'de verileri arayüze aktarma işlemi yaparız.
- View Modelde veritabanından veri alma , arayüzde matematiksel işlem vb. gibi şeyler yapılır.
- Verilerdeki değişimleri gözlemlemek için *LiveData* yapısını kullanabiliriz.
- DataBinding ile karışmaması lazım , Data binding tasarım alanına verileri kolay şekilde aktarmamızı sağlar.
- View Model ise Activity içinde yapılan arayüz ile ilgili veri işlemlerini kontrol eder.
- Genel kullanım her sayfaya özgü ViewModel oluşturmaktır.

# Kurulum

The screenshot shows the `build.gradle/app` file in the Android Studio editor. The code is as follows:

```
dependencies { this: DependencyHandlerScope
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation("androidx.lifecycle:lifecycle-viewmodel:2.5.1")
    implementation("androidx.activity:activity-ktx:1.6.1")
}
```

A pink rectangular highlight covers the last two lines of the code. A blue dashed arrow points from the right side of the screen towards this highlighted area, indicating a navigation target. On the right side of the screen, there is a navigation bar titled "Gradle Scripts" which includes "build.gradle.kts (Project: Kutup)" and "build.gradle.kts (Module :app)".

# ViewModel Sınıfı Oluşturma

- Arayüzde kullanacağımız veriyi bu sınıfta tanımlarız ve yönetiriz.

```
class MainActivityViewModel : ViewModel() { ViewModel özelliği aktarılır.  
Yönetilecek ve arayüzde  
kullanılacak veri → var sonuc = "0"  
  
Yönettiğimiz değere  
matematiksel işlemi aktardık. →  
    fun toplamaYap(alinanSayi1:String, alinanSayi2:String){  
        val sayı1 = alinanSayi1.toInt()  
        val sayı2 = alinanSayi2.toInt()  
        val toplam = sayı1 + sayı2  
        sonuc = toplam.toString()  
    }  
  
    fun carpmaYap(alinanSayi1:String, alinanSayi2:String){  
        val sayı1 = alinanSayi1.toInt()  
        val sayı2 = alinanSayi2.toInt()  
        val carpma = sayı1 * sayı2  
        sonuc = carpma.toString()  
    }  
}
```

*Buradaki metodların amacı arayüzü değiştirecek veya besleyecek verilere değer üretmektir.*

*Bundan dolayı return demeden yönettiğimiz veriye elde ettiğimiz değer aktarılır. Aktarılma işlemi olduğunda yönettiğimiz verinin içeriği değişmiş olur.*

# ViewModel Kullanımı

- Activity içinde oluşturduğumuz ViewModel sınıfını kullanabiliriz.

Oluşturduğumuz ViewModel sınıfını Activity'e bağlıyoruz.

View model içinde yönettiğimiz veriye hesaplanan değeri aktarmak için View model içindeki metod çalıştırılır.

Bu metod yönettiğimiz değerin içeriğini değiştirir.

```
class MainActivity : AppCompatActivity() {
    private lateinit var tasarim:ActivityMainBinding

    private val viewModel:MainActivityViewModel by viewModels()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)
        tasarim.mainActivityNesnesi = this
        tasarim.hesaplamaSonucu = viewModel.sonuc
    }

    fun buttonToplamaTikla(alinanSayi1:String,alinanSayi2:String){
        viewModel.toplamaYap(alinanSayi1,alinanSayi2)
        tasarim.hesaplamaSonucu = viewModel.sonuc
    }

    fun buttonCarpmaTikla(alinanSayi1:String,alinanSayi2:String){
        viewModel.carpmaYap(alinanSayi1,alinanSayi2)
        tasarim.hesaplamaSonucu = viewModel.sonuc
    }
}
```

View model içinde yönettiğimiz veriye erişim.

Yukardaki metod sayesinde yönettiğimiz değerin içeriğinin değişmiş en son halini alırız ve arayüzde gösteririz.

# ViewModel İçinde Context Kullanımı

ViewModel içinde veritabanı gibi çalışmalarda context gereklili olabilir.

1

```
class MainActivityViewModelFactory(private val application: Application)
    : ViewModelProvider.NewInstanceFactory() {

    override fun <T : ViewModel?> create(modelClass: Class<T>): T {
        return MainActivityViewModel(application) as T
    }

}
```

2

```
class MainActivityViewModel(application: Application) : AndroidViewModel(application) {

    private val kdaor = KisilerDaoRepository(application)
    var kisilerListesi = MutableLiveData<List<Kisiler>>()
```

3

```
class MainActivity : AppCompatActivity(), SearchView.OnQueryTextListener {
    private lateinit var adapter: KisilerAdapter
    private val viewModel: MainActivityViewModel by viewModels() {MainActivityViewModelFactory(application)}
```

# LiveData - Kotlin

# LiveData

- View Model ile kullanılır.
- LiveData View Model'in kullanımı kolaylaştırır.
- Temelde yaptığı işlem ViewModel içinde yönetilen verinin tetiklenmesini sağlamak ve değişimi dinlemektir.
- Bu tetikleme ve değişimi dinleme işlemi kodlama yapımızı sadeleştirir.
- Örnek : TextView içeriği çok fazla yerde değiştiriliyorsa , normalde her değişim kodlamasında textView kodlamasını yazmamız gereklidir.
- LiveData sayesinde dinleme yapısı kurularak tek bir textView tanımlaması yapılarak , değişen değeri dinleriz ve değeri textView içeriğine aktarırız.
- Bu işlem gereksiz textView kodlamalarından bizi kurtarır.

# LiveData – ViewModel içinde Kullanımı

- ViewModel yapısını güçlendirmek için kullanılır.

```
Yönetilecek veri live data → var sonuc = MutableLiveData( value: "0")  
türüne dönüştürülür.  
Verimize varsayılan değer  
atanır.  
  
class MainActivityViewModel : ViewModel {  
    fun toplamaYap(alinanSayi1:String,alinanSayi2: String){  
        val sayi1 = alinanSayi1.toInt()  
        val sayi2 = alinanSayi2.toInt()  
        val toplam = sayi1 + sayi2  
        sonuc.value = toplam.toString() ←  
    }  
  
    fun carpmaYap(alinanSayi1:String,alinanSayi2: String){  
        val sayi1 = alinanSayi1.toInt()  
        val sayi2 = alinanSayi2.toInt()  
        val carpma = sayi1 * sayi2  
        sonuc.value = carpma.toString()  
    }  
}
```

value metodu ile hesaplama sonucunda  
oluşan değeri verimize atadık ve dinleme  
işlemiini tetikledik.

# LiveData – Activity İçinde Kullanımı

Yönettiğimiz verinin  
değişimini dinleme.  
view model içindeki  
value metodunu bu  
dinlemeyi tetikler.

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    private val viewModel:MainActivityViewModel by viewModels()  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = DataBindingUtil.setContentView( activity: this,R.layout.activity_main)  
        tasarim.mainActivityNesnesi = this  
  
        viewModel.sonuc.observe( owner: this) { it: String!  
            tasarim.hesaplamaSonucu = it  
        }  
    }  
}
```

value metodunu ile veride değişim  
olduğu anda en son değer buraya  
gelir ve arayüzde kullanılabilir.

```
fun buttonToplamaTikla(alinanSayi1:String,alinanSayi2:String){  
    viewModel.toplamaYap(alinanSayi1,alinanSayi2)  
}  
  
fun buttonCarpmaTikla(alinanSayi1: String,alinanSayi2: String){  
    viewModel.carpmaYap(alinanSayi1,alinanSayi2)  
}  
}
```

Bu metod ile ViewModel  
içindeki verimizin içeriğini  
hesaplama sonucu ile  
değiştirebiliriz ve içinde  
value metodunu olduğu için  
dinleme işlemini tetikler.

# Data Layer

## Repository - Data Source

# Repository - Data Source Sınıfı

- İşlemsel kodların çalıştığı yer Data Source'dur.
- Repository , View Modelden gelen isteği Data Source'a iletir.
- Ortak kullanım için oluşturduğumuz metodların yer aldığı sınıfıtır.
- Veritabanı işlemlerinde bazı metodları bir çok sayfa kullanabilir.
- Ortak erişebilecek bir sınıf oluşturup kodlama tekrarlarını azaltırız.
- Repository ve Data Source içinde veri iletişimini sağlamak için Coroutine kullanılır.

# Data Source Oluşturma

```
class MatematikDataSource {  
    suspend fun toplamaYap(alinanSayi1:String, alinanSayi2: String) : String =  
        withContext(Dispatchers.IO){ this: CoroutineScope  
            val sayi1 = alinanSayi1.toInt()  
            val sayi2 = alinanSayi2.toInt()  
            val toplam = sayi1 + sayi2  
            return@withContext toplam.toString()  
        }  
  
    suspend fun carpmaYap(alinanSayi1:String, alinanSayi2: String) : String =  
        withContext(Dispatchers.IO){ this: CoroutineScope  
            val sayi1 = alinanSayi1.toInt()  
            val sayi2 = alinanSayi2.toInt()  
            val carpma = sayi1 * sayi2  
            return@withContext carpma.toString()  
        }  
}
```

*İşlemin yapımı ve coroutine altyapısı ile veriyi return etme*

# Repository Oluşturma

```
class MatematikRepository {  
    var mds = MatematikDataSource()  
  
    suspend fun toplamaYap(alinanSayi1:String, alinanSayi2: String) : String  
    = mds.toplamaYap(alinanSayi1, alinanSayi2)  
  
    suspend fun carpmaYap(alinanSayi1:String, alinanSayi2: String) : String  
    = mds.carpmaYap(alinanSayi1, alinanSayi2)  
}
```

*Data source üzerinden gelen veriyi  
view modele iletmek için metod  
oluşturulur.*



# ViewModel İçinde Kullanımı

```
class MainActivityViewModel : ViewModel() {  
    var sonuc = MutableLiveData(value: "0")  
    var mrepo = MatematikRepository()  
  
    fun toplamaYap(alinanSayi1:String, alinanSayi2: String){  
        CoroutineScope(Dispatchers.Main).launch { this: CoroutineScope  
            sonuc.value = mrepo.toplamaYap(alinanSayi1, alinanSayi2)  
        }  
    }  
  
    fun carpmaYap(alinanSayi1:String, alinanSayi2: String){  
        CoroutineScope(Dispatchers.Main).launch { this: CoroutineScope  
            sonuc.value = mrepo.carpmaYap(alinanSayi1, alinanSayi2)  
        }  
    }  
}
```

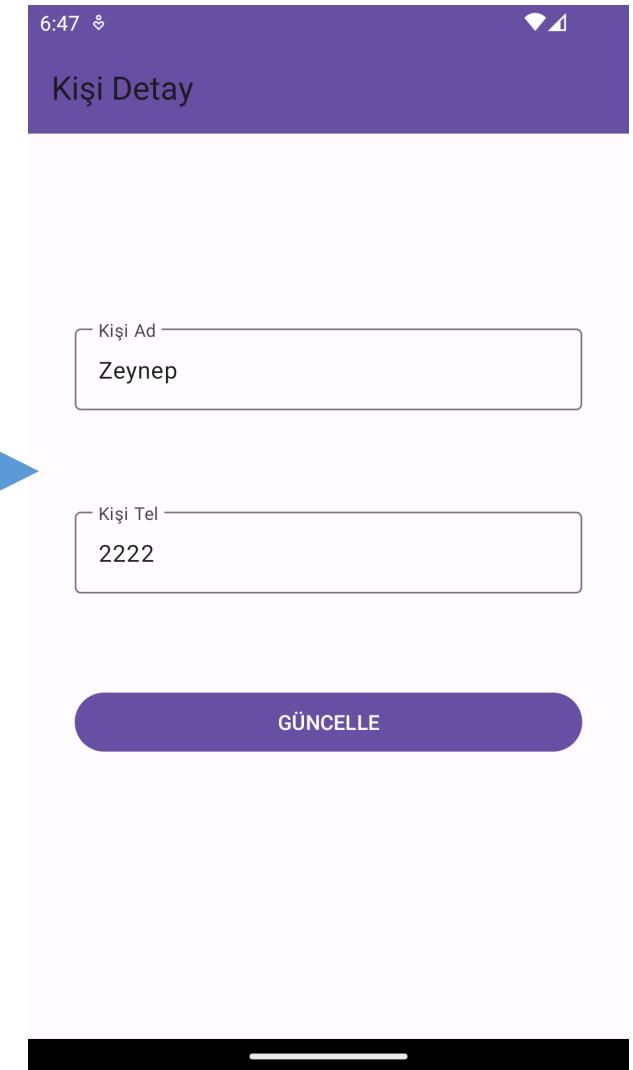
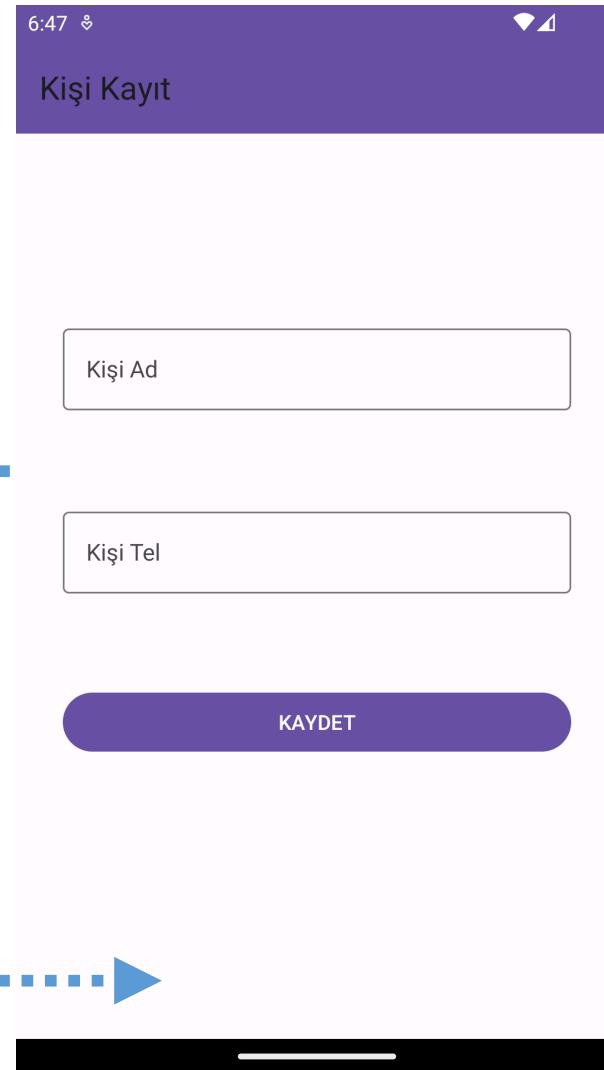
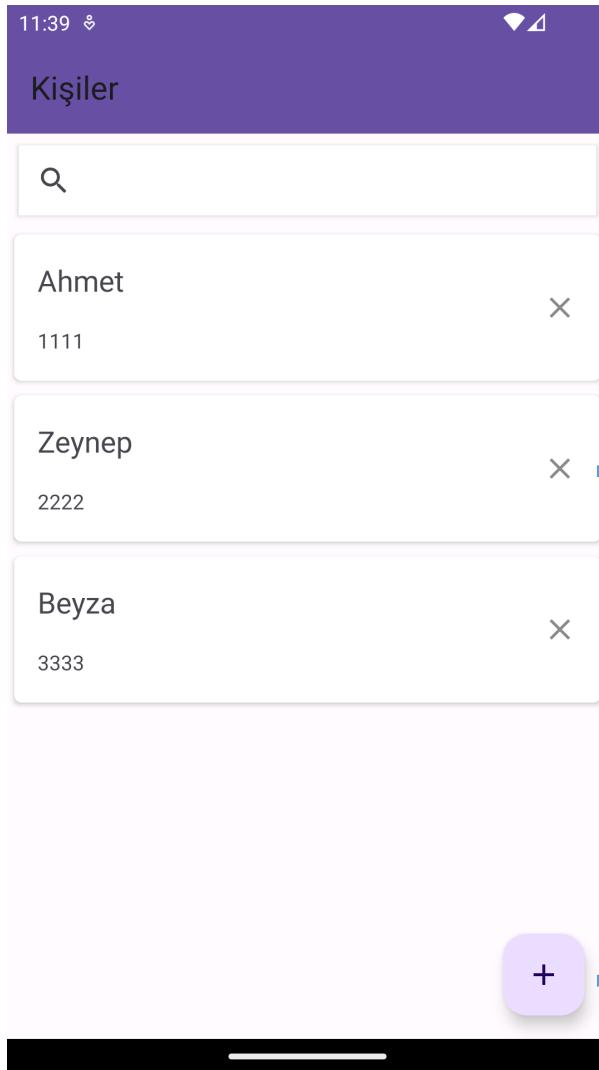
*Repository üzerinden gelen veriyi  
live dataya iletiliyor  
ve  
arayüz tetikleniyor.*



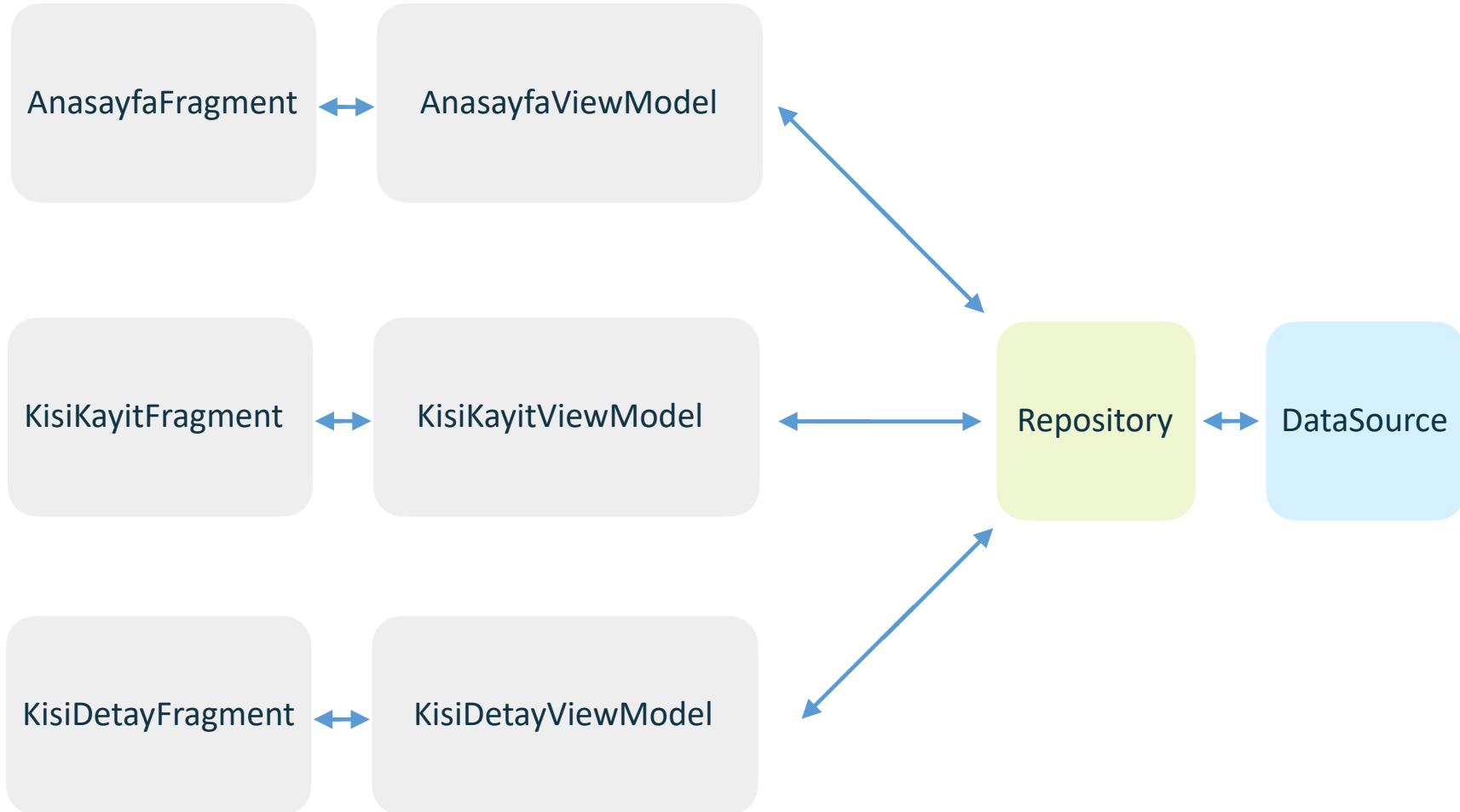
# Kişiler Uygulaması

MVVM

# Kişiler Uygulaması



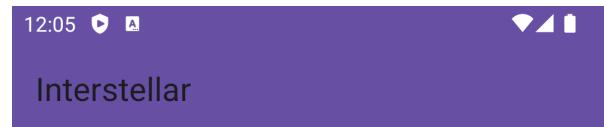
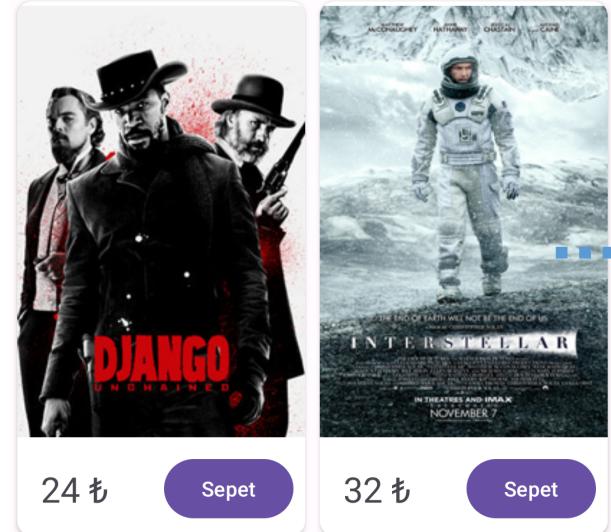
# Kişiler Uygulaması



# Filmler Uygulaması

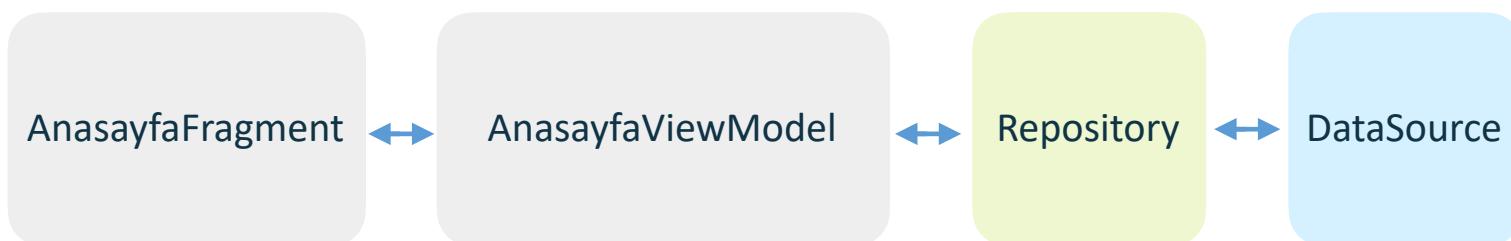
MVVM

# Film Uygulaması



32 ₺

# Filmler Uygulaması



# Extension Kullanımı

# Extension

- Extension sık kullanılan kodları kendimizce metodlaşdırırmak anlamına gelmektedir.

```
package com.example.kisileruygulamasi.utils

import ...

fun Navigation.gecisYap(v:View,id: Int) {
    findNavController(v).navigate(id)
}

fun Navigation.gecisYap(v:View,id: NavDirections) {
    findNavController(v).navigate(id)
}
```



The image shows a portion of the Android Studio project structure. A 'utils' folder is expanded, revealing an 'Extensions.kt' file. This indicates where the extension functions defined in the code above are implemented.

# Kullanımı

Eski Kodlama

```
fun fabTikla(v:View){  
    Navigation.findNavController(v).navigate(R.id.kisiKayitGecis)  
}
```

Yeni Kodlama

```
fun fabTikla(v:View){  
    Navigation.gecisYap(v,R.id.kisiKayitGecis)  
}
```

---

Eski Kodlama

```
t.satirCard.setOnClickListener { it: View!  
    val gecis = AnasayfaFragmentDirections.kisiDetayGecis(kisi = kisi)  
    Navigation.findNavController(it).navigate(gecis)  
}
```

Yeni Kodlama

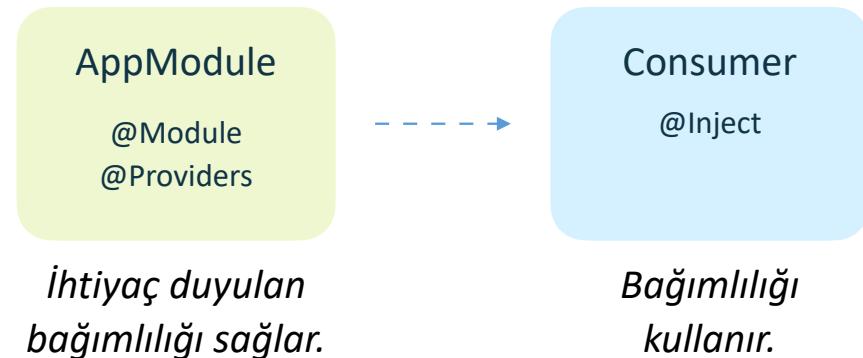
```
t.satirCard.setOnClickListener { it: View!  
    val gecis = AnasayfaFragmentDirections.kisiDetayGecis(kisi = kisi)  
    Navigation.gecisYap(it,gecis)  
}
```

# Dependency Injection

# Dagger Hilt

# Dagger Hilt

- Dependency Injection , bağımlılık demektir.
- Bir yapıyı kullanırken başka bir yapı kullanmak zorunda kalınıyorsa bağımlılık var anlamına gelmektedir.
  - Örn : Bir sınıfın Context istemesidir.
  - Büyük çaplı projelerde kodlamayı sadeleştirmektedir.
  - Dagger Hilt , android için bu işlemleri yaptığımiz kütüphanedir.



# Klasik Kullanım

```
class MainActivity : AppCompatActivity() {  
  
    lateinit var kargo: Kargo  
    lateinit var internet: Internet  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        kargo = Kargo()  
        kargo.gonder()  
  
        internet = Internet()  
        internet.basvuruYap()  
    }  
  
    class Adres(var adresBilgisi:String) {  
    }  
  
    class Kargo {  
        val adres = Adres(adresBilgisi: "Kadıköy / İSTANBUL")  
  
        fun gonder(){  
            Log.e(tag: "Sonuç", msg: "Kargo ${adres.adresBilgisi} adresine gönderildi.")  
        }  
    }  
  
    class Internet {  
        val adres = Adres(adresBilgisi: "Kadıköy / İSTANBUL")  
  
        fun basvuruYap(){  
            Log.e(tag: "Sonuç", msg: "Internet başvurusu ${adres.adresBilgisi} adresine yapıldı.")  
        }  
    }  
}
```

# Kurulum

```
buildscript { this: ScriptHandlerScope
    dependencies{ this: DependencyHandlerScope
        classpath ("com.google.dagger:hilt-android-gradle-plugin:2.44")
    }
}
plugins { this: PluginDependenciesSpecScope
    id("com.android.application") version "8.1.0" apply false
}

dependencies { this: DependencyHandlerScope
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}

implementation("com.google.dagger:hilt-android:2.44")
kapt("com.google.dagger:hilt-android-compiler:2.44")
}

classpath ("com.google.dagger:hilt-android-gradle-plugin:2.44")
id("kotlin-kapt")
id("dagger.hilt.android.plugin")

implementation("com.google.dagger:hilt-android:2.44")
kapt("com.google.dagger:hilt-android-compiler:2.44")
```

build.gradle/project

build.gradle/app

build.gradle/app

Gradle Scripts

build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

Gradle Scripts

build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

```
id("com.android.application")
id("org.jetbrains.kotlin.android")
id("kotlin-kapt")
id("dagger.hilt.android.plugin")
```

build.gradle/app

# Kapt Kurulum Hatası Çözümü

```
compileOptions { this: CompileOptions
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
kotlinOptions { this: KotlinJvmOptions
    jvmTarget = "1.8"
}

dependencies {
    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation ("androidx.room:room-runtime:2.5.0-beta02")
    kapt("androidx.room:room-compiler:2.5.0-beta02")
    implementation("androidx.room:room-ktx:2.4.3")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.5.1")
}
```

build.gradle/app

*Not : Hata alırsanız bunu ekleyebilirsiniz.*

```
compileOptions {
    sourceCompatibility = JavaVersion.VERSION_1_7
    targetCompatibility = JavaVersion.VERSION_1_7
}
kotlinOptions {
    jvmTarget = "17"
}
```

Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

# Hilt Sınıfı Oluşturma

- Bütün uygulamayı temsil eden bir sınıf olmalıdır.

```
@HiltAndroidApp
```

```
class HiltApplication : Application() {  
}
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.kisileruygulaması">
```

```
<application  
        android:name=".di.HiltApplication"  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="KisilerUygulaması"  
        android:roundIcon="@mipmap/ic_launcher_round"
```



*di : dependency injection*

2

```
class Adres(var adresBilgisi:String) {  
}
```

3

```
class Kargo @Inject constructor(var adres:Adres) {  
    fun gönder(){  
        Log.e( tag: "Sonuç", msg: "Kargo ${adres.adresBilgisi} adresine gönderildi.")  
    }  
}
```

4

```
@Module  
@InstallIn(SingletonComponent::class)  
class AppModule {  
    @Provides  
    @Singleton  
    fun provideAdres() : Adres {  
        return Adres( adresBilgisi: "Kadıköy / İSTANBUL")  
    }  
}
```

5

```
@AndroidEntryPoint  
class MainActivity : AppCompatActivity() {  
    @Inject  
    lateinit var kargo: Kargo  
    @Inject  
    lateinit var internet: Internet  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        kargo.gönder()  
        internet.basvuruYap()  
    }  
}
```

# Kişiler Uygulaması

## Hilt

# Hilt

- Uygulamamızda yer alan bağımlılıklar tespit edilmesi.

```
class AnasayfaViewModel : ViewModel() {  
    var krepo = KisilerRepository()  
  
class KisiKayitViewModel : ViewModel() {  
    var krepo = KisilerRepository()  
  
class KisiDetayViewModel : ViewModel() {  
    var krepo = KisilerRepository()  
  
class KisilerRepository {  
    var kds = KisilerDataSource()
```

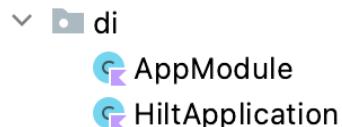
# Hilt Sınıfı Oluşturma

- Bütün uygulamayı temsil eden bir sınıf olmalıdır.

```
@HiltAndroidApp
```

```
class HiltApplication : Application() {  
}
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.kisileruygulaması">  
  
<application  
        android:name=".di.HiltApplication"  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="KisilerUygulaması"  
        android:roundIcon="@mipmap/ic_launcher_round"
```



*di : dependency injection*

# Bağımlıkları Kullanıacak Sınıfları Düzenleme

```
@HiltViewModel  
class AnasayfaViewModel @Inject constructor(var krepo:KisilerRepository) : ViewModel() {  
    var kisilerListesi = MutableLiveData<List<Kisiler>>()
```

```
@HiltViewModel  
class KisiDetayViewModel @Inject constructor(var krepo:KisilerRepository) : ViewModel()
```

```
@HiltViewModel  
class KisiKayitViewModel @Inject constructor(var krepo:KisilerRepository) : ViewModel()
```

```
class KisilerRepository(var kds:KisilerDataSource) {
```

# Module Oluşturma

- Bağımlılıkları besleyecek sınıfın ve metodların oluşturulması

```
@Module
@InstallIn(SingletonComponent::class)
class AppModule {
    @Provides
    @Singleton
    fun provideKisilerRepository(kds:KisilerDataSource) : KisilerRepository {
        return KisilerRepository(kds)
    }

    @Provides
    @Singleton
    fun provideKisiDetayViewModel(@Inject constructor(var krepo:KisilerRepository) : ViewModel() {
        return KisiDetayViewModel(krepo)
    }
}
```



# Kullanım

- @AndroidEntryPoint kullanıcı sınıfının kodlanacağı Fragment ve Activity'e eklenmelidir.
- Eğer Fragment ile çalışıyorsa Fragmentların çalışmasını sağlayan Activity'de eklenmelidir

```
@AndroidEntryPoint
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

```
@AndroidEntryPoint
class AnasayfaFragment : Fragment(), SearchView.OnQueryTextListener {
    private lateinit var tasarim:FragmentAnasayfaBinding
    private lateinit var viewModel: AnasayfaFragmentViewModel
```

```
@AndroidEntryPoint
class KisiDetayFragment : Fragment() {
    private lateinit var tasarim:FragmentKisiDetayBinding
    private lateinit var viewModel: KisiDetayFragmentViewModel
```

```
@AndroidEntryPoint
class KisiKayitFragment : Fragment() {
    private lateinit var tasarim:FragmentKisiKayitBinding
    private lateinit var viewModel: KisiKayitFragmentViewModel
```

# Filmler Uygulaması

## Hilt

# Hilt

- Uygulamamızda yer alan bağımlılıklar tespit edilmesi.

```
class AnasayfaViewModel : ViewModel() {  
    var frepo = FilmlerRepository()  
  
class FilmlerRepository {  
    var fds = FilmlerDataSource()
```

# Hilt Sınıfı Oluşturma

- Bütün uygulamayı temsil eden bir sınıf olmalıdır.

```
@HiltAndroidApp
```

```
class HiltApplication : Application() {  
}
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.kisileruygulaması">  
  
<application  
        android:name=".di.HiltApplication"  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="KisilerUygulaması"  
        android:roundIcon="@mipmap/ic_launcher_round"
```



*di : dependency injection*

2

## Bağımlıkları Kullanıacak Sınıfları Düzenleme

```
@HiltViewModel  
class AnasayfaViewModel @Inject constructor(var frepo:FilmlerRepository) : ViewModel() {
```

```
class FilmlerRepository(var fds:FilmlerDataSource) {
```

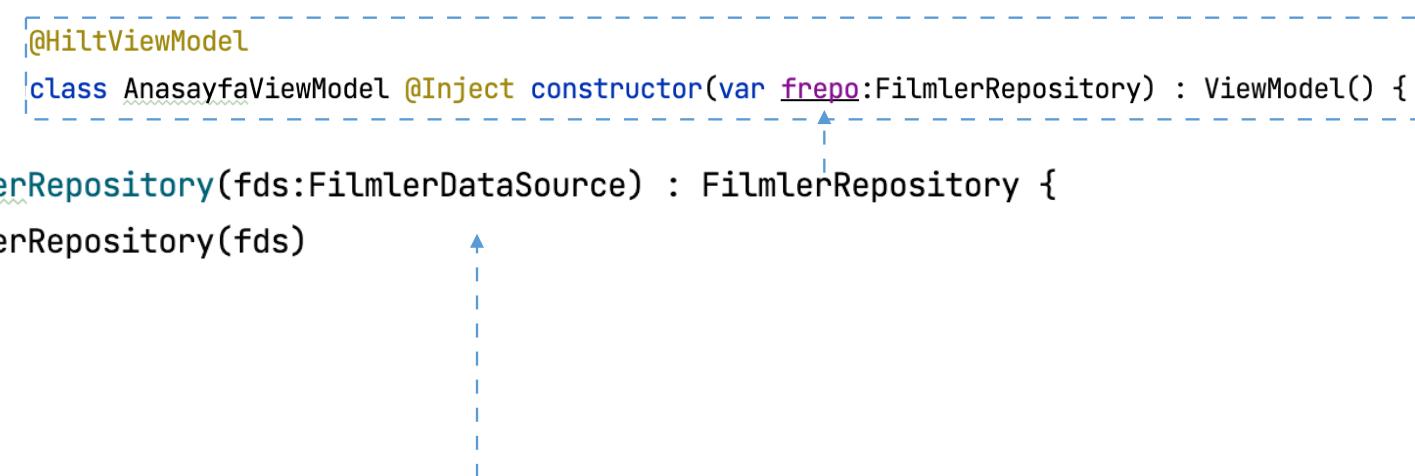
# Module Oluşturma

- Bağımlılıkları besleyecek sınıfın ve metodların oluşturulması

```
@Module
@InstallIn(SingletonComponent::class)
class AppModule {
    @Provides
    @Singleton
    fun provideFilmllerRepository(fds:FilmllerDataSource) : FilmllerRepository {
        return FilmllerRepository(fds)
    }

    @Provides
    @Singleton
    fun provideFilmllerDataSource() : FilmllerDataSource {
        return FilmllerDataSource()
    }
}

class AnasayfaViewModel @Inject constructor(var frepo:FilmllerRepository) : ViewModel()
```



# Kullanım

- @AndroidEntryPoint kullanıcı sınıfının kodlanacağı Fragment ve Activity'e eklenmelidir.
- Eğer Fragment ile çalışıyorsa Fragmentların çalışmasını sağlayan Activity'de eklenmelidir

```
@AndroidEntryPoint
```

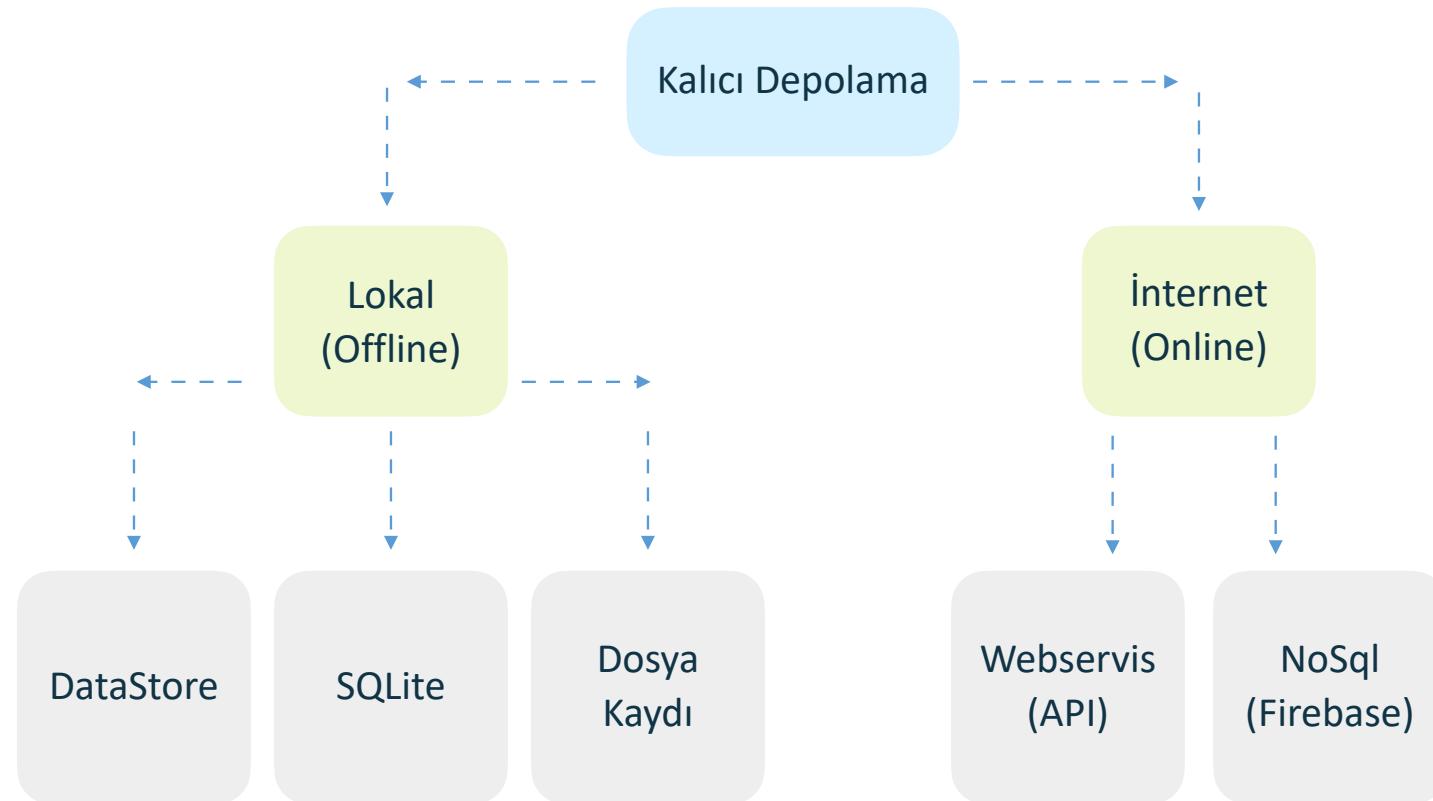
```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

```
@AndroidEntryPoint
```

```
class AnasayfaFragment : Fragment() {  
    private lateinit var binding: FragmentAnasayfaBinding  
    private lateinit var viewModel: AnasayfaViewModel
```

# Depolama İşlemleri

# Depolama İşlemleri



# Preferences DataStore

## Preferences DataStore

- Key - Value ilişkisi ile basit verileri kalıcı olarak depolayabiliriz.
- Kullanılan bazı veri türleri : *String,int,double,bool,Set<String>*
- Uygulama silindiğinde veriler silinmektedir.
- Veritabanı üzerindeki gibi detaylı verileri kayıt etmiceksek hızlı bir çözüm için kullanılabilir.

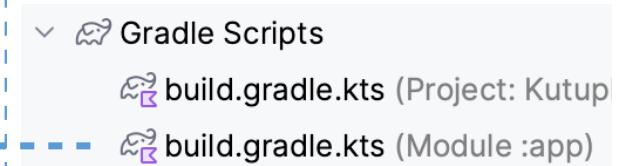
# Kurulum

```
dependencies { this: DependencyHandlerScope           build.gradle/app

    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation("androidx.datastore:datastore-preferences:1.0.0")
    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.5.1")
    api("org.jetbrains.kotlinx:kotlinx-coroutines-core:1.6.4")
    api("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.6.4")
}

implementation("androidx.datastore:datastore-preferences:1.0.0")
implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.5.1")
api("org.jetbrains.kotlinx:kotlinx-coroutines-core:1.6.4")
api("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.6.4")
```



# Preferences DataStore Sınıfı Oluşturma

Preferences  
DataStore'a erişim  
için ds isimli değişken  
oluşturuldu.

Kayıt İşlemi

Okuma İşlemi

Silme İşlemi

```
class AppPref(var context:Context) {  
    → val Context.ds : DataStore<Preferences> by preferencesDataStore( name: "bilgiler")  
  
    companion object {  
        val AD_KEY = stringPreferencesKey( name: "AD")  
    }  
  
    suspend fun kayitAd(ad:String){  
        context.ds.edit { it: MutablePreferences  
            → it[AD_KEY] = ad  
        }  
    }  
  
    suspend fun okuAd():String {  
        val p = context.ds.data.first()  
        return p[AD_KEY] ?: "isim yok"  
    }  
  
    suspend fun silAd(){  
        context.ds.edit { it: MutablePreferences  
            → it.remove(AD_KEY)  
        }  
    }  
}
```

Kayıt edilecek dosya adı

Kayıt için kullanılacak KEY tanımlaması

Okuma işlemi yapılmırken varsayılan değer oluşturulabilir.

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(LayoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3

# Kullanımı

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

        val ap = AppPref(context: this)

        val job = CoroutineScope(Dispatchers.Main).launch {
            ap.kayitAd(ad: "Ahmet")

            ap.silAd()

            val gelenAd = ap.okuAd()
            Log.e(tag: "Gelen Ad", gelenAd)
        }
    }
}
```

# Final Kod :

```
class AppPref(var context:Context) {
    val Context.ds : DataStore<Preferences> by preferencesDataStore(name: "bilgiler")

    companion object {
        val AD_KEY = stringPreferencesKey(name: "AD")
        val YAS_KEY = intPreferencesKey(name: "YAS")
        val BOY_KEY = doublePreferencesKey(name: "BOY")
        val BEKAR_KEY = booleanPreferencesKey(name: "BEKAR")
        val ARKADAS_LISTE_KEY = stringSetPreferencesKey(name: "ARKADAS_LISTE")
        val SAYAC_KEY = intPreferencesKey(name: "SAYAC")
    }

    suspend fun kayitAd(ad:String){
        context.ds.edit { it: MutablePreferences
            it[AD_KEY] = ad
        }
    }

    suspend fun okuAd():String {
        val p = context.ds.data.first()
        return p[AD_KEY] ?: "isim yok"
    }

    suspend fun silAd(){
        context.ds.edit { it: MutablePreferences
            it.remove(AD_KEY)
        }
    }

    suspend fun kayitYas(yas:Int){
        context.ds.edit { it: MutablePreferences
            it[YAS_KEY] = yas
        }
    }

    suspend fun okuYas():Int {
        val p = context.ds.data.first()
        return p[YAS_KEY] ?: 0
    }

    suspend fun kayitBoy(boy:Double){
        context.ds.edit { it: MutablePreferences
            it[BOY_KEY] = boy
        }
    }

    suspend fun okuBoy():Double {
        val p = context.ds.data.first()
        return p[BOY_KEY] ?: 0.0
    }

    suspend fun kayitBekar(bekar:Boolean){
        context.ds.edit { it: MutablePreferences
            it[BEKAR_KEY] = bekar
        }
    }

    suspend fun okuBekar():Boolean {
        val p = context.ds.data.first()
        return p[BEKAR_KEY] ?: false
    }

    suspend fun kayitArkadasListe(liste:Set<String>){
        context.ds.edit { it: MutablePreferences
            it[ARKADAS_LISTE_KEY] = liste
        }
    }

    suspend fun okuArkadasListe():Set<String>? {
        val p = context.ds.data.first()
        return p[ARKADAS_LISTE_KEY]
    }

    suspend fun kayitSayac(sayac:Int){
        context.ds.edit { it: MutablePreferences
            it[SAYAC_KEY] = sayac
        }
    }

    suspend fun okuSayac():Int {
        val p = context.ds.data.first()
        return p[SAYAC_KEY] ?: 0
    }
}
```

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

        val ap = AppPref( context: this)

        val job = CoroutineScope(Dispatchers.Main).launch {
            //Kayıt İşlemleri
            ap.kayitAd( ad: "Ahmet")
            ap.kayitYas( yas: 23)
            ap.kayitBoy( boy: 1.78)
            ap.kayitBekar( bekar: true)

            val liste = HashSet<String>()
            liste.add("Mehmet")
            liste.add("Zeynep")

            ap.kayitArkadasListe(liste)
        }

        val gelenAd = ap.okuAd()
        val gelenYas = ap.okuYas()
        val gelenBoy = ap.okuBoy()
        val gelenBekar = ap.okuBekar()

        Log.e( tag: "Gelen Ad",gelenAd)
        Log.e( tag: "Gelen Yaş",gelenYas.toString())
        Log.e( tag: "Gelen Boy",gelenBoy.toString())
        Log.e( tag: "Gelen Bekar",gelenBekar.toString())

        val gelenListe = ap.okuArkadasListe()

        if (gelenListe != null){
            for(a in gelenListe){
                Log.e( tag: "Gelen Arkadaş",a)
            }
        }
    }
}
```



# Uygulama : Açılış Sayısı

```
class AppPref(var context: Context) {  
    val Context.ds : DataStore<Preferences> by preferencesDataStore(name: "bilgiler")  
  
    companion object {  
        val SAYAC_KEY = intPreferencesKey(name: "SAYAC")  
    }  
  
    suspend fun kayitSayac(sayac:Int){  
        context.ds.edit { it: MutablePreferences  
            it[SAYAC_KEY] = sayac  
        }  
    }  
  
    suspend fun okuSayac():Int {  
        val p = context.ds.data.first()  
        return p[SAYAC_KEY] ?: 0  
    }  
}
```

Açılış Sayısı : 2



# Uygulama : Açılış Sayısı

Açılış Sayısı : 2

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        val ap = AppPref( context: this)  
  
        val job = CoroutineScope(Dispatchers.Main).launch { this: CoroutineScope  
            var gelenSayac = ap.okuSayac()  
  
            ap.kayitSayac(++gelenSayac)  
  
            tasarim.textViewSayac.text = "Açılış Sayısı : $gelenSayac"  
        }  
    }  
}
```

# Veritabanı Giriş

# Tablo Yapısı

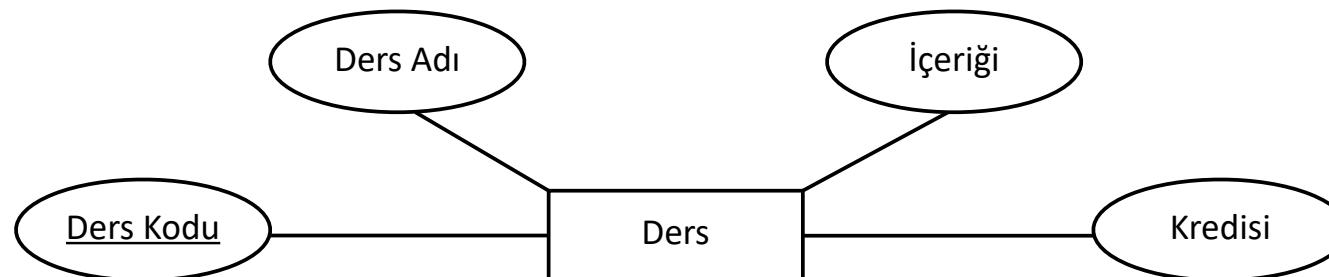
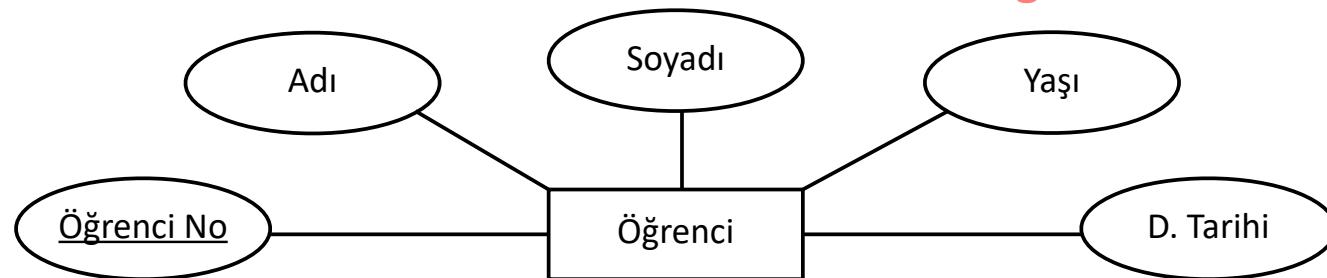
- Tüm Veritabanları içerisinde bilgiler tablolarda Alan

Veri

| not_id | ogrenci_adi  | ders_adi | not1 | not2 |   |
|--------|--------------|----------|------|------|---|
| 1      | Mehmet ERSOY | Tarih    | 50   | 60   |  Kayıt |
| 2      | Mehmet ERSOY | Fizik    | 70   | 80   |   |
| 3      | Zeynep GÜR   | Tarih    | 70   | 90   |   |
| 4      | Cemal GELİR  | Tarih    | 30   | 50   |   |

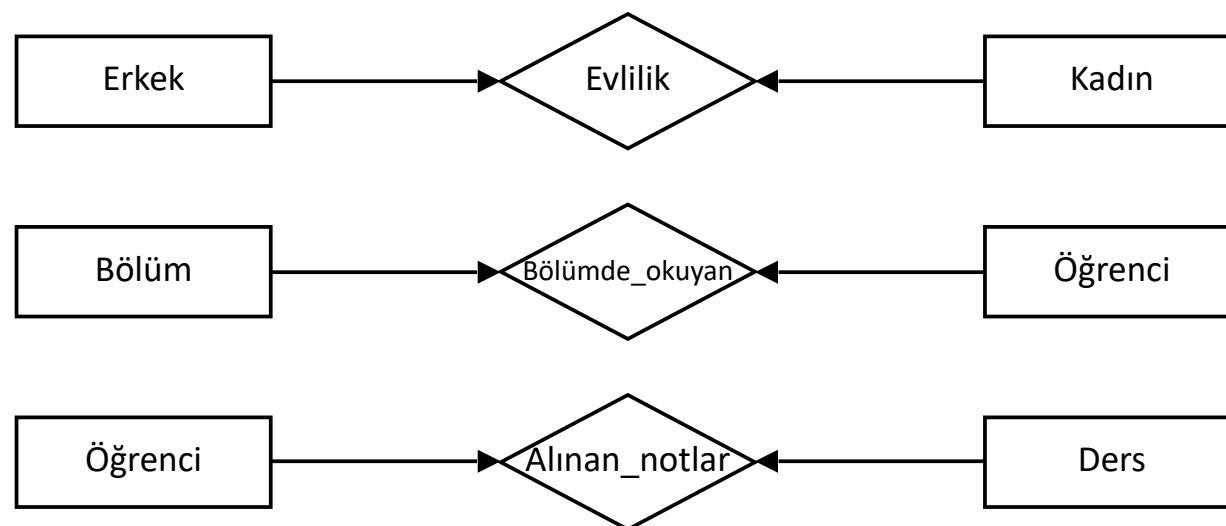
# Tablo Tasarlama

- **Öncelikle her tablo için uygun alanların belirlenmelidir aynı sınıflardaki özelliklerin belirlenmesi gibi.**

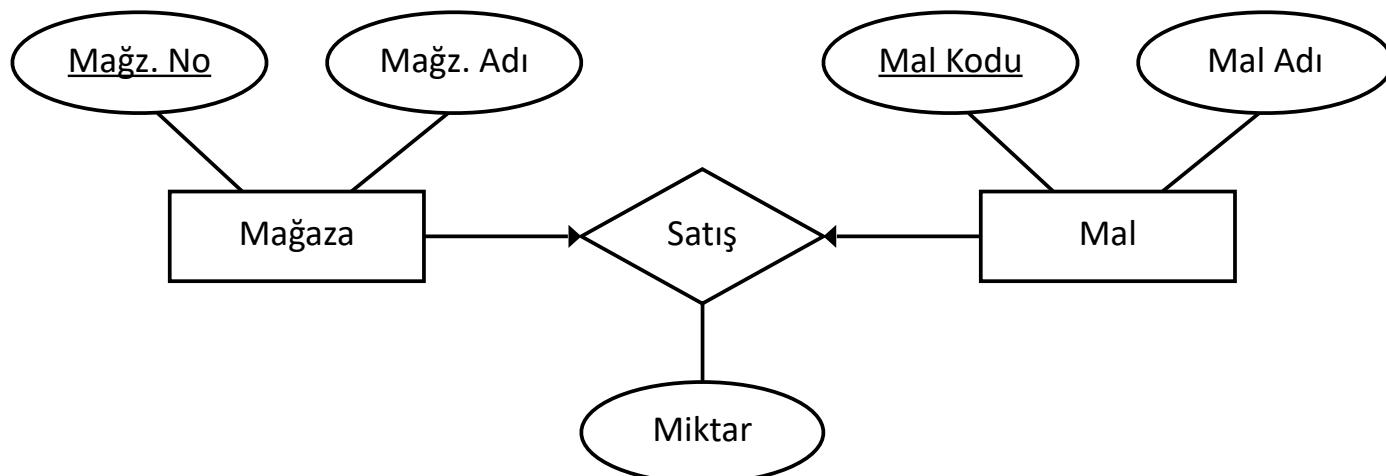


# İlişkisel Tablo Modeli

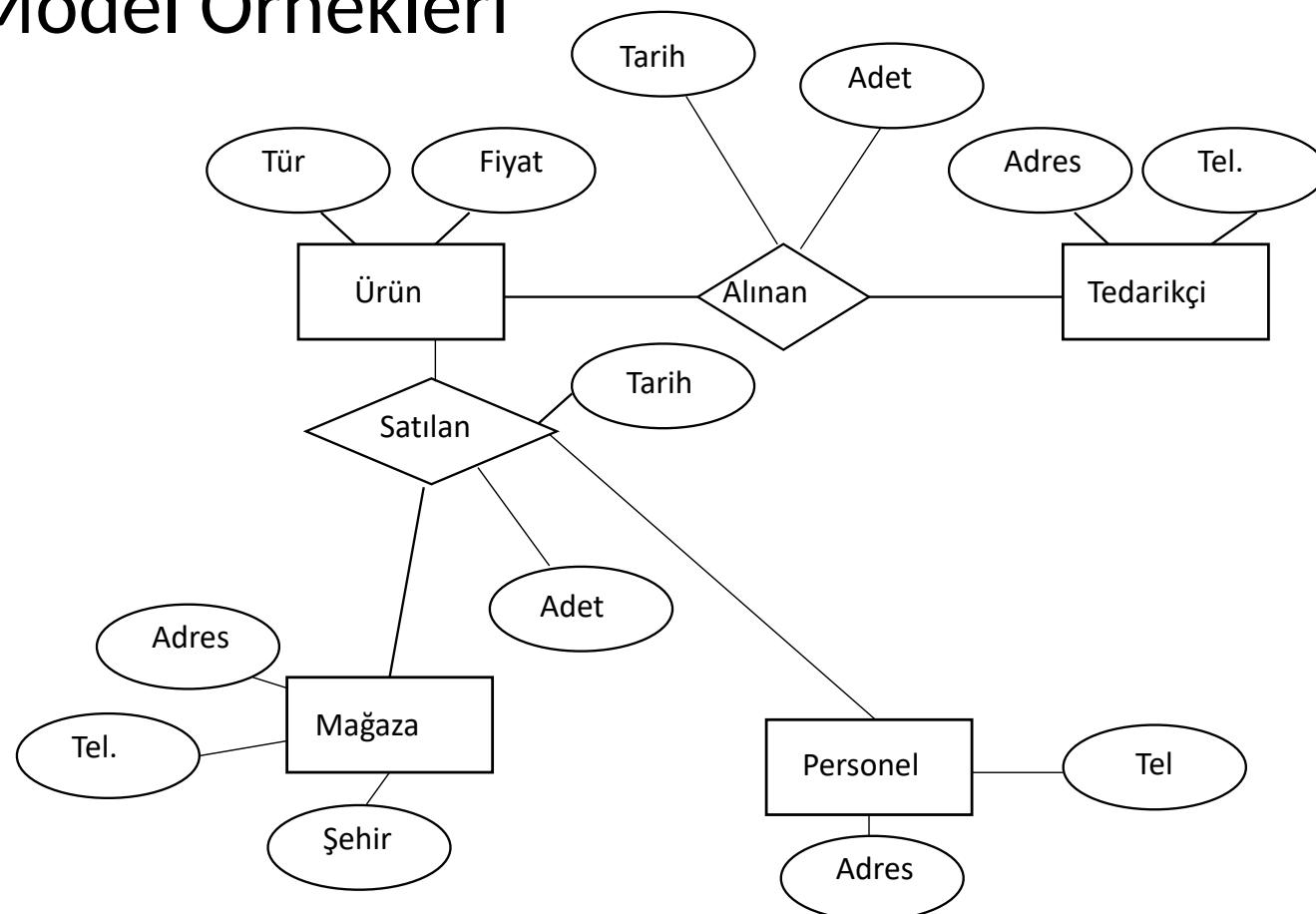
- Tablolardaki gereksiz veri tekrarlarını önlemek için kullanılan bir yapıdır.



# İlişkisel Model Örnekleri



# İlişkisel Model Örnekleri



## Birincil Anahtar – PRIMARY KEY

- İlişkisel veri tabanında bir tablonun benzer değerler içermeyen (tekil : unique) bir sütunu ya da birkaç sütunu birlikte **birincil anahtar** (primary key - PK) olarak tanımlanabilir.
- Birincil anahtar bir aday anahtardır ve söz konusu varlığın kayıtlarını en iyi biçimde karakterize eden anahtardır.
- Birincil anahtar tanımlandığında şu şekilde bir sınırlama konulmuş olacaktır:
  - Birincil anahtar NULL (boş) değerleri veya birbirinin aynı değerleri içeremez.

## Dış Anahtar – FOREIGN KEY

- Bir dış anahtar (foreign key - FK) bir sütun veya çok sayıdaki sütunların birleşiminden oluşur.
- Dış anahtar aynı tablo ya da başka bir tablodaki bir birincil anahtar ile eşleştirilir.

| PK |       |         | FK |  | PK      |          |
|----|-------|---------|----|--|---------|----------|
| No | Adı   | BölümNo |    |  | BölümNo | BölümAdı |
| 25 | Burak | 10      |    |  | 10      | Personel |
| 13 | Begüm | 10      |    |  | 20      | Muhasebe |
| 28 | Dilay | 20      |    |  | 30      | Satış    |

# Filmler Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

| film_id | film_adi          | film_yil | film_resim      | kategori_ad | yonetmen_ad  |
|---------|-------------------|----------|-----------------|-------------|--------------|
| 3       | Inception         | 2010     | inception       | Bilim kurgu | C. Nolan     |
| 5       | Django            | 2008     | django          | Dram        | Q. Tarantino |
| 9       | The Hateful Eight | 2010     | thehatefuleight | Dram        | Q. Tarantino |

# Filmler Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

| film_id | film_ad           | film_yil | film_resim      | kategori_id | yonetmen_id |
|---------|-------------------|----------|-----------------|-------------|-------------|
| 3       | Inception         | 2010     | inception       | 11          | 88          |
| 5       | Django            | 2008     | django          | 22          | 90          |
| 9       | The Hateful Eight | 2010     | thehatefuleight | 22          | 90          |

| kategori_id | kategori_adi |
|-------------|--------------|
| 11          | Bilim kurgu  |
| 22          | Dram         |
| 44          | Komedи       |

| yonetmen_id | yonetmen_adi   |
|-------------|----------------|
| 88          | C. Nolan       |
| 90          | Q. Tarantino   |
| 130         | Yılmaz Erdoğan |

# Okul Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

| not_id | ogrenci_adi  | ders_adi | not1 | not2 |
|--------|--------------|----------|------|------|
| 1      | Mehmet ERSOY | Tarih    | 50   | 60   |
| 2      | Mehmet ERSOY | Fizik    | 70   | 80   |
| 3      | Zeynep GÜR   | Tarih    | 70   | 90   |
| 4      | Cemal GELİR  | Tarih    | 30   | 50   |

# Okul Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

| not_id | ogrenci_id | ders_id | not1 | not2 |
|--------|------------|---------|------|------|
| 1      | 1          | 1       | 50   | 60   |
| 2      | 1          | 2       | 70   | 80   |
| 3      | 2          | 1       | 70   | 90   |
| 4      | 4          | 1       | 30   | 50   |

| ogrenci_id | ogrenci_adi  |
|------------|--------------|
| 1          | Mehmet ERSOY |
| 2          | Zeynep GÜR   |
| 4          | Cemal GELİR  |

| ders_id | ders_adi |
|---------|----------|
| 1       | Tarih    |
| 2       | Fizik    |
| 3       | Kimya    |

# Sipariş Girme Uygulaması

- Kategorilere ayrılmış ürünlerden adet girerek sipariş oluşturma.

| siparis_id | siparis_adet | urun_adi | kategori_adi |
|------------|--------------|----------|--------------|
| 1          | 2            | Pizza    | Yiyecekler   |
| 2          | 1            | Baklava  | Tatlılar     |
| 3          | 1            | Sütlaç   | Tatlılar     |
| 4          | 3            | Ayran    | İçecekler    |

# Sipariş Girme Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

| siparis_id | siparis_adet | urun_adi |
|------------|--------------|----------|
| 1          | 2            | 1        |
| 2          | 1            | 2        |
| 3          | 1            | 3        |
| 4          | 3            | 5        |

| kategori_id | kategori_adi |
|-------------|--------------|
| 1           | Yiyecekler   |
| 2           | Tatlılar     |
| 3           | İçecekler    |

| urun_id | urun_adi | kategori_id |
|---------|----------|-------------|
| 1       | Pizza    | 1           |
| 2       | Baklava  | 2           |
| 3       | Sütlaç   | 2           |
| 5       | Ayran    | 3           |

# Hastane Uygulaması

- Bu senaryo için temel bir veri tabanı modeli düşünelim.

| randevu_id | bolum_adi   | doktor_adi   | hasta_adi    | randevu_tarihi   |
|------------|-------------|--------------|--------------|------------------|
| 1          | Dermotoloji | Sedat AK     | Mehmet ERSOY | 12/02/2018 10:30 |
| 2          | Kardiyoloji | Ceyda MERMER | Kemal ALTAY  | 10/04/2018 12:30 |
| 3          | Dermotoloji | Sedat AK     | Zeynep GÜR   | 07/01/2018 16:00 |
| 4          | Dermotoloji | Sedat AK     | Cemal GELİR  | 26/02/2018 9:30  |

# Hastane Uygulaması

- Bu senaryo için olması gereken veri tabanı modeli.

| randevu_id | doktor_id | hasta_adi    | randevu_tarihi   |
|------------|-----------|--------------|------------------|
| 1          | 1         | Mehmet ERSOY | 12/02/2018 10:30 |
| 2          | 2         | Kemal ALTAY  | 10/04/2018 12:30 |
| 3          | 1         | Zeynep GÜR   | 07/01/2018 16:00 |
| 4          | 1         | Cemal GELİR  | 26/02/2018 9:30  |

| bolum_id | bolum_adi        |
|----------|------------------|
| 1        | Dermotoloji      |
| 2        | Kardiyoloji      |
| 3        | Fizik Tedavi     |
| 4        | Göz Hastalıkları |

| doktor_id | doktor_adi   | bolum_id |
|-----------|--------------|----------|
| 1         | Sedat AK     | 1        |
| 2         | Ceyda MERMER | 2        |
| 3         | Ahmet Ziya   | 1        |
| 4         | Ece Kasırga  | 3        |

# VERİTABANI UYGULAMASI

urunler

| urun_id | urun_adi   | urun_fiyati |
|---------|------------|-------------|
| 1       | Bilgisayar | 10000       |
| 2       | Telefon    | 7000        |
| 3       | Saat       | 3000        |

musteriler

| musteri_id | musteri_adi | musteri_adresi |
|------------|-------------|----------------|
| 10         | Ahmet       | İSTANBUL       |
| 12         | Zeynep      | BURSA          |
| 19         | Ece         | ANKARA         |

siparisler

| siparis_id | urun_id | musteri_id | siparis_adeti |
|------------|---------|------------|---------------|
| 3          | 1       | 10         | 1             |
| 5          | 1       | 19         | 2             |
| 9          | 3       | 12         | 1             |

**urunler**

▼ Gelişmiş

**Alanlar Kısıtlar**

Ekle Sil En yu...taşı Yukarı taşı Aşağı

| İsim        | Tip     | NN                       | Biri                                | Otc                                 | Ber                      |
|-------------|---------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|
| urun_id     | INTEGER | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| urun_adi    | TEXT    | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| urun_fiyati | INTEGER | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |

**musteriler**

▼ Gelişmiş

**Alanlar Kısıtlar**

Ekle Sil En yu...taşı Yukarı taşı Aşağı

| İsim           | Tip     | NN                       | Biri                                | Otc                                 | Ber                      |
|----------------|---------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|
| musteri_id     | INTEGER | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| musteri_adi    | TEXT    | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| musteri_adresi | TEXT    | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |

**siparisler**

▼ Gelişmiş

**Alanlar Kısıtlar**

Ekle Sil En yukarı taşı Yukarı taşı Aşağı taşı En aşağı taşı

| İsim          | Tip     | NN                       | Biri                                | Otc                                 | Ber                      | Və | Kç | Karşıla | Yabancı Anahtar       |
|---------------|---------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|----|----|---------|-----------------------|
| siparis_id    | INTEGER | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |    |    |         |                       |
| urun_id       | INTEGER | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |    |    |         | urunler("urun_id")    |
| musteri_id    | INTEGER | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |    |    |         | musteriler musteri_id |
| siparis_adeti | INTEGER | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |    |    |         |                       |

# SQL İFADELERİ

# TABLO OLUŞTURMA

```
CREATE TABLE "urunler" (
    "urun_id" INTEGER,
    "urun_adi" TEXT,
    "urun_fiyati" INTEGER,
    PRIMARY KEY("urun_id" AUTOINCREMENT)
);
```

```
CREATE TABLE "siparisler" (
    "siparis_id" INTEGER,
    "urun_id" INTEGER,
    "musteri_id" INTEGER,
    "siparis_adeti" INTEGER,
    PRIMARY KEY("siparis_id" AUTOINCREMENT),
    FOREIGN KEY("musteri_id") REFERENCES "musteriler"("musteri_id"),
    FOREIGN KEY("urun_id") REFERENCES "urunler"("urun_id")
);
```

# Veri Kaydı - Insert

**INSERT INTO urunler (urun\_adi,urun\_fiyati) VALUES ('Bilgisayar',10000)**

| urun_id | urun_adi     | urun_fiyati |
|---------|--------------|-------------|
| Filtre  | Filtre       | Filtre      |
| 1       | 1 Bilgisayar | 10000       |
| 2       | 2 Telefon    | 8000        |
| 3       | 3 Saat       | 3000        |
| 4       | 4 Mont       | 750         |
| 5       | 5 Kalem      | 200         |
| 6       | 6 TV         | 8000        |
| 7       | 7 Gözlük     | 1000        |

# Update - Güncelleme

```
UPDATE urunler SET urun_fiyati = 1200 WHERE urun_id = 7 ;
```

| urun_id | urun_adi     | urun_fiyati |
|---------|--------------|-------------|
| Filtre  | Filtre       | Filtre      |
| 1       | 1 Bilgisayar | 10000       |
| 2       | 2 Telefon    | 8000        |
| 3       | 3 Saat       | 3000        |
| 4       | 4 Mont       | 750         |
| 5       | 5 Kalem      | 200         |
| 6       | 6 TV         | 8000        |
| 7       | 7 Gözlük     | 1200        |

# Delete – Veri Silme

```
DELETE FROM urunler WHERE urun_id = 5 ;
```

|   | <b>urun_id</b> | <b>urun_adi</b> | <b>urun_fiyati</b> |
|---|----------------|-----------------|--------------------|
|   | Filtre         | Filtre          | Filtre             |
| 1 | 1              | Bilgisayar      | 10000              |
| 2 | 2              | Telefon         | 8000               |
| 3 | 3              | Saat            | 3000               |
| 4 | 4              | Mont            | 750                |
| 5 | 6              | TV              | 8000               |
| 6 | 7              | Gözlük          | 1200               |

# SELECT – Seçim Yapma

- Tüm verileri getir.

```
SELECT * FROM urunler
```

- İstenilen alanları getir.

```
SELECT urun_adi,urun_fiyati FROM urunler
```

- Kayıt kontrol.

```
SELECT count(*) as toplam FROM urunler WHERE urun_fiyati = 8000
```

- Foreign key li tablolardan verileri alma.

```
SELECT * FROM urunler,musteriler,siparisler WHERE  
urunler.urun_id = siparisler.urun_id and musteriler.musteri_id = siparisler.musteri_id
```

|          | <b>urun_id</b> | <b>urun_adi</b>   | <b>urun_fiyati</b> |
|----------|----------------|-------------------|--------------------|
|          | Filtre         | Filtre            | Filtre             |
| <b>1</b> | <b>1</b>       | <b>Bilgisayar</b> | <b>10000</b>       |
| <b>2</b> | <b>2</b>       | <b>Telefon</b>    | <b>8000</b>        |
| <b>3</b> | <b>3</b>       | <b>Saat</b>       | <b>3000</b>        |
| <b>4</b> | <b>4</b>       | <b>Mont</b>       | <b>750</b>         |
| <b>5</b> | <b>6</b>       | <b>TV</b>         | <b>8000</b>        |
| <b>6</b> | <b>7</b>       | <b>Gözlük</b>     | <b>1200</b>        |

# WHERE – Şart Oluşturma

```
SELECT * FROM urunler WHERE urun_adi = 'Telefon'
```

```
SELECT * FROM urunler WHERE urun_fiyati > 5000
```

```
SELECT * FROM urunler WHERE urun_fiyati > 1000 and urun_fiyati < 5000
```

Not : Şart yazarken yazı mı ? Sayısal veri mi?  
Buna dikkat edilmelidir.  
Sayısal veriler normal yazılrken sayısal ifadeler ” veya ‘ ile yazılır.

| urun_id | urun_adi     | urun_fiyati |
|---------|--------------|-------------|
| Filtre  | Filtre       | Filtre      |
| 1       | 1 Bilgisayar | 10000       |
| 2       | 2 Telefon    | 8000        |
| 3       | 3 Saat       | 3000        |
| 4       | 4 Mont       | 750         |
| 5       | 6 TV         | 8000        |
| 6       | 7 Gözlük     | 1200        |

# ORDER BY - SIRALAMA

- Ürünler tablosundaki ürünleri harf sırasına göre artan şekilde getir.

```
SELECT * FROM urunler ORDER BY urun_adi ASC
```

ASC : Artan  
DESC : Azalan

- Kisiler tablosundan yaşı 18 olanları getir.

```
SELECT * FROM kisiler ORDER BY urun_fiyati DESC
```

|   | urun_id | urun_adi   | urun_fiyati |
|---|---------|------------|-------------|
|   | Filtre  | Filtre     | Filtre      |
| 1 | 1       | Bilgisayar | 10000       |
| 2 | 2       | Telefon    | 8000        |
| 3 | 3       | Saat       | 3000        |
| 4 | 4       | Mont       | 750         |
| 5 | 6       | TV         | 8000        |
| 6 | 7       | Gözlük     | 1200        |

# LIKE – BENZERLİK ARAMA

SELECT \* FROM urunler WHERE urun\_adi like '%a%'

| urun_id | urun_adi     | urun_fiyati |
|---------|--------------|-------------|
| Filtre  | Filtre       | Filtre      |
| 1       | 1 Bilgisayar | 10000       |
| 2       | 2 Telefon    | 8000        |
| 3       | 3 Saat       | 3000        |
| 4       | 4 Mont       | 750         |
| 5       | 6 TV         | 8000        |
| 6       | 7 Gözlük     | 1200        |

# LIMIT - SINIRLI SAYIDA VERİ GETİR

```
SELECT * FROM urunler LIMIT 2
```

```
SELECT * FROM urunler WHERE urun_fiyati < 5000 LIMIT 2
```

```
SELECT * FROM urunler ORDER BY RANDOM() LIMIT 2
```

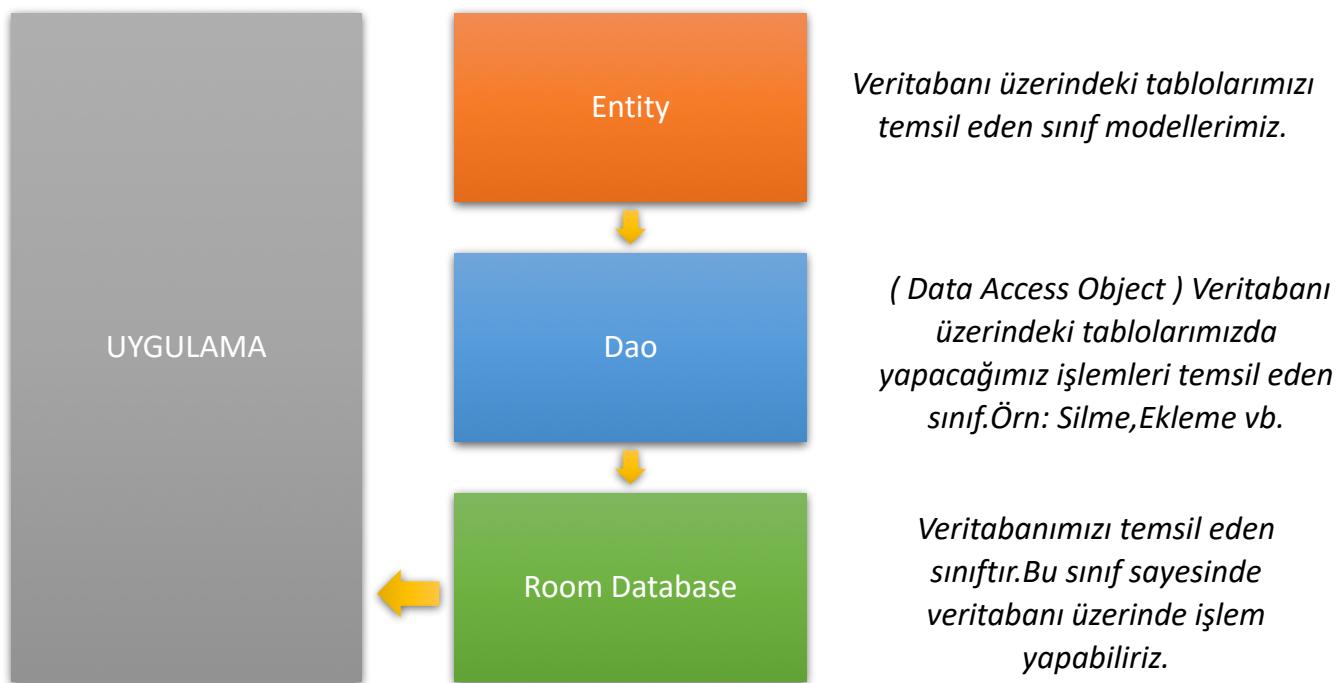
|   | urun_id | urun_adi   | urun_fiyati |
|---|---------|------------|-------------|
|   | Filtre  | Filtre     | Filtre      |
| 1 | 1       | Bilgisayar | 10000       |
| 2 | 2       | Telefon    | 8000        |
| 3 | 3       | Saat       | 3000        |
| 4 | 4       | Mont       | 750         |
| 5 | 6       | TV         | 8000        |
| 6 | 7       | Gözlük     | 1200        |

# Room - Kotlin

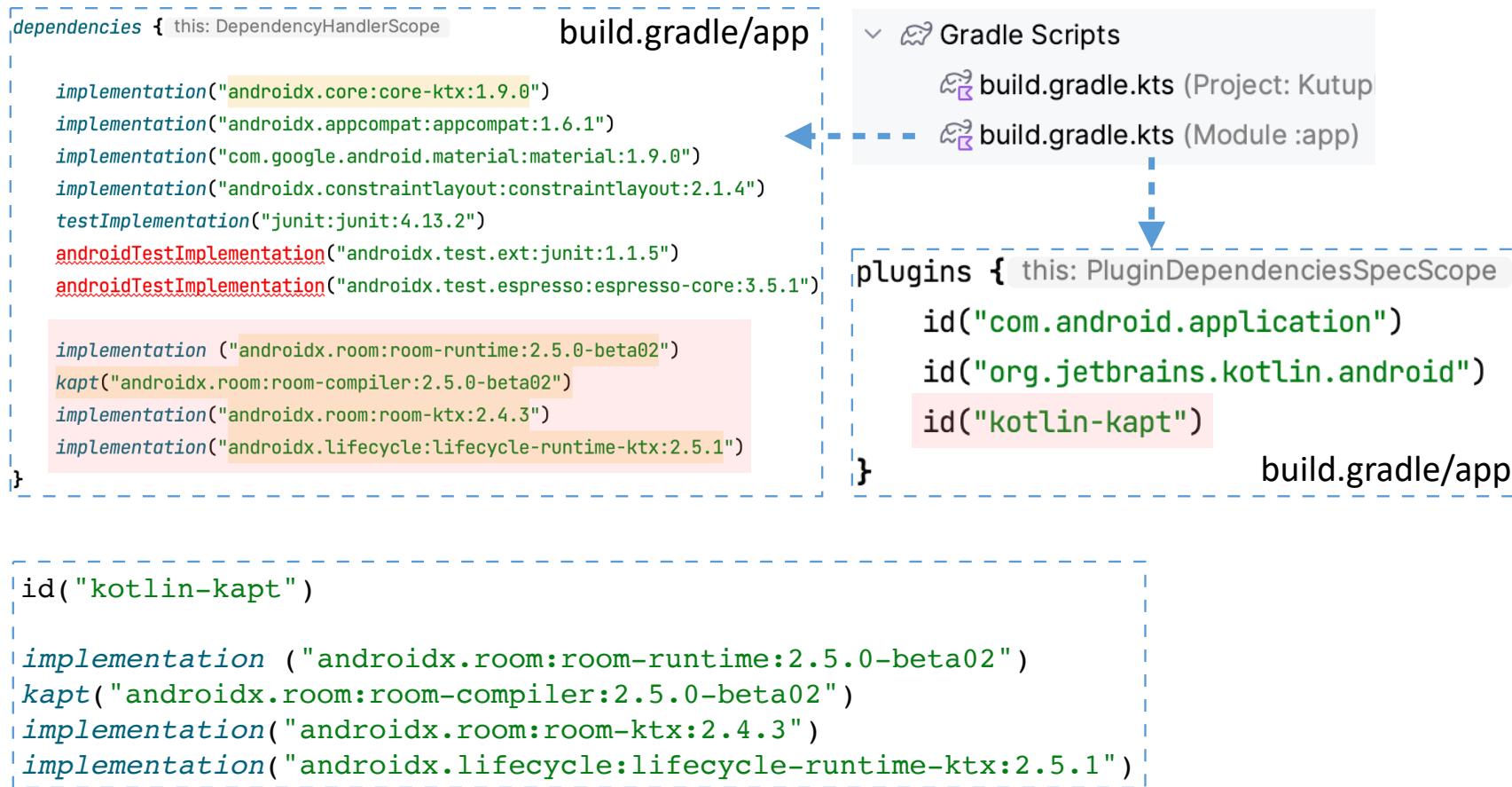
# Room Kütüphanesi

- SQLite veritabanı üzerinde çalışan bir kütüphanedir.
- Androdin kendi kütüphanesidir.
- Nesne tabanlı şekilde çalışmaktadır.
- SQLite kullanımını oldukça kolaylaştıran bir yapıdır.
- Room kütüphanesini Asenkron çalışması için Coroutine yapısıyla kullanmaktayız.

# Room Kütüphanesi Temel Yapıları



# Kurulum



# Kapt Kurulum Hatası Çözümü

```
compileOptions { this: CompileOptions
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
kotlinOptions { this: KotlinJvmOptions
    jvmTarget = "1.8"
}

dependencies {
    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation ("androidx.room:room-runtime:2.5.0-beta02")
    kapt("androidx.room:room-compiler:2.5.0-beta02")
    implementation("androidx.room:room-ktx:2.4.3")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.5.1")
}
```

build.gradle/app

*Not : Hata alırsanız bunu ekleyebilirsiniz.*

```
compileOptions {
    sourceCompatibility = JavaVersion.VERSION_1_7
    targetCompatibility = JavaVersion.VERSION_1_7
}
kotlinOptions {
    jvmTarget = "17"
}
```

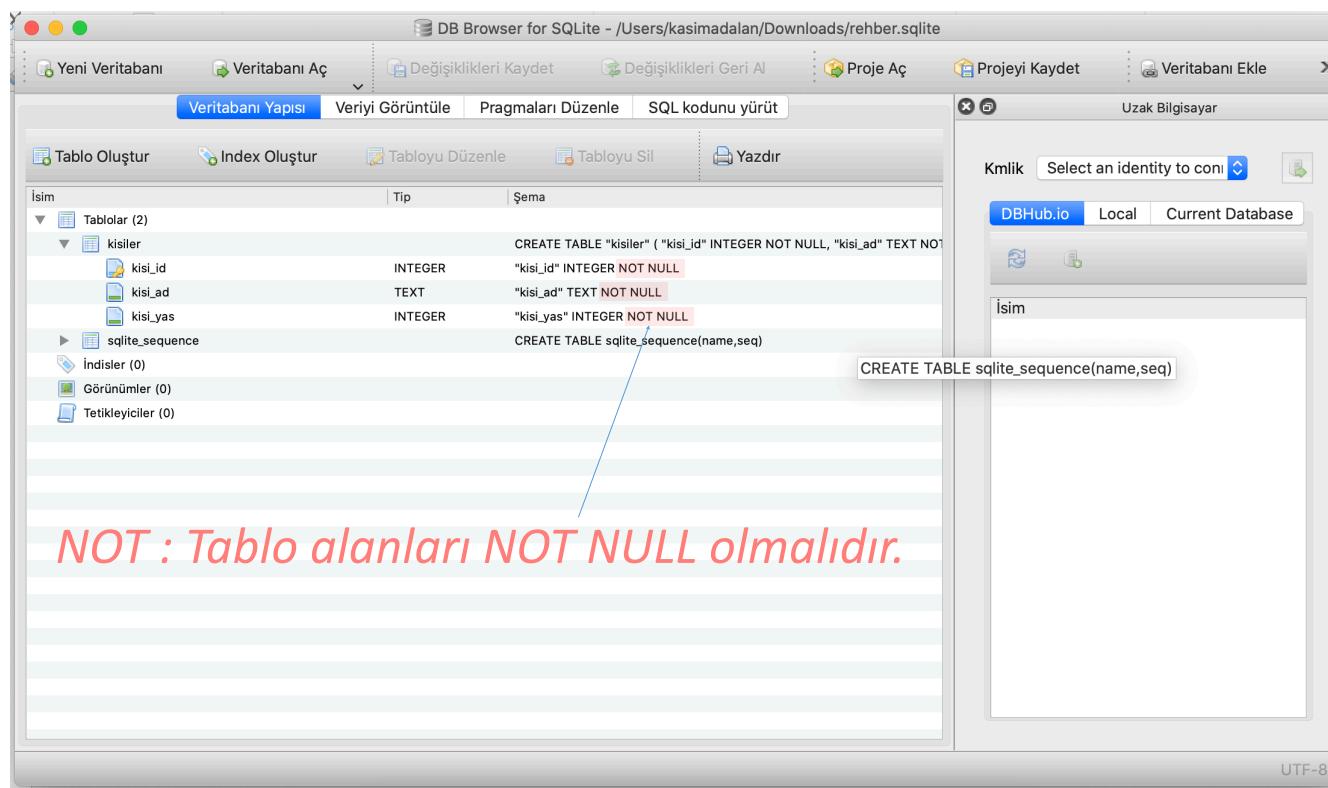
Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

▼ Tablolar (2)

|                    |  |                            |  |
|--------------------|--|----------------------------|--|
| ▼ kisiler          | CREATE TABLE "kisiler" ( "kisi_id" INTEGER NOT NULL, "kisi_ad" TEXT NOT NULL, "kisi_tel" TEXT NOT NULL ) |                            |  |
| kisi_id            | INTEGER  | "kisi_id" INTEGER NOT NULL |  |
| kisi_ad            | TEXT   | "kisi_ad" TEXT NOT NULL    |  |
| kisi_tel           | TEXT   | "kisi_tel" TEXT NOT NULL   |  |
| > sqlite_sequence  | CREATE TABLE sqlite_sequence(name,seq)   |                            |  |
| Indisler (0)       |  |                            |  |
| Görünümler (0)     |  |                            |  |
| Tetikleyiciler (0) |  |                            |  |

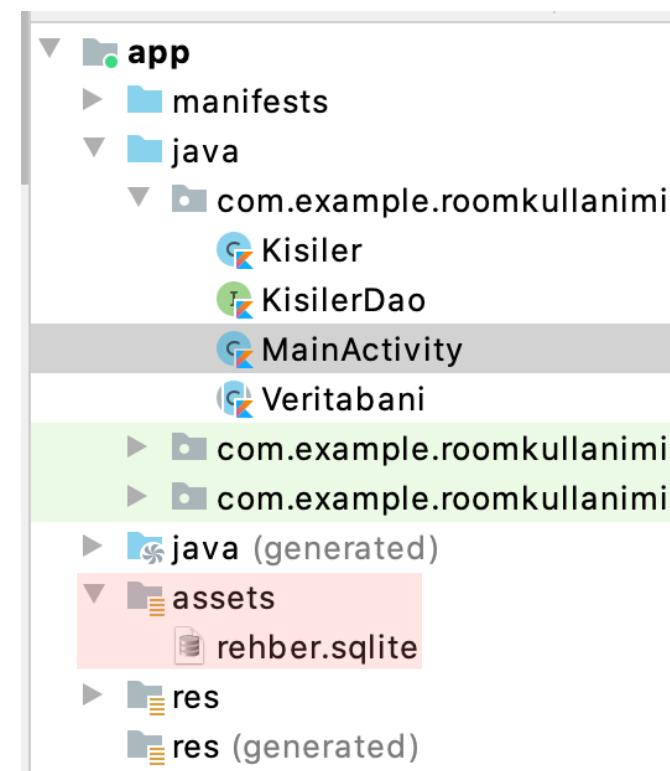
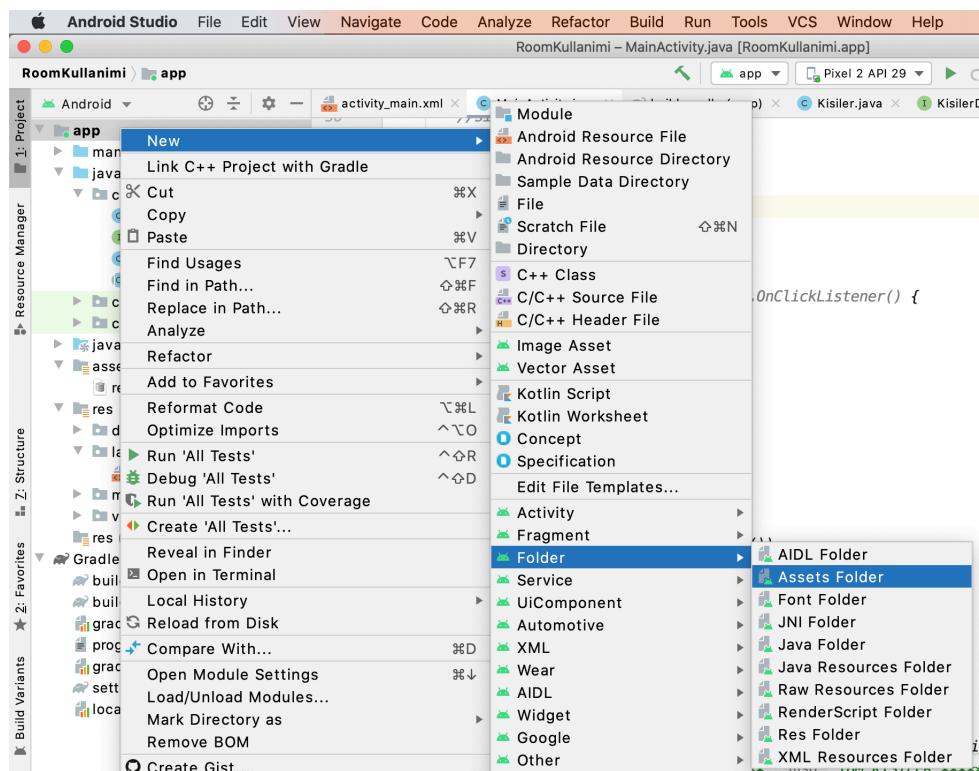
2

## DB Browser ile Veri tabanı Oluşturma



3

# Veri tabanını Assets Dosyasına Aktarma



## 4

# Veritabanı Tablosu için Entity Oluşturma

*Sınıfın veritabanında  
temsil edeceği tablo adı*

```
@Entity(tableName = "kisiler")
```

*Primary Key ve Auto Increment özellikleri*

*Veri tabanı tablo alanı adı*

```
data class Kisiler(@PrimaryKey(autoGenerate = true) ————— Primary Key ve Auto Increment özellikleri
@ColumnInfo(name = "kisi_id") @NotNull var kisi_id:Int,
@ColumnInfo(name = "kisi_ad") @NotNull var kisi_ad: String,
@ColumnInfo(name = "kisi_yas") @NotNull var kisi_yas:Int) {
```

}

|          | CREATE TABLE "kisiler" ( |         |                             |
|----------|--------------------------|---------|-----------------------------|
|          |                          |         |                             |
| kisiler  |                          | INTEGER | "kisi_id" INTEGER NOT NULL  |
| kisi_id  |                          | TEXT    | "kisi_ad" TEXT NOT NULL     |
| kisi_ad  |                          | INTEGER | "kisi_yas" INTEGER NOT NULL |
| kisi_yas |                          |         |                             |

Veritabanındaki alanlar NOT NULL özelliğine sahip olmalıdır. Aksi halde hata alırız.  
 Veritabanı alanında String ifade not null ise entity içinde @NonNull ifadesi yazmalyız  
 primitif tiplere @NonNull ifadesi gereklidir ama veritabanı modelinizin böyle olduğu  
 görebilmek için primitif tiplere @NonNull ifadesi ekleyebilirsiniz.

## 5

*Metodun veritabanında yapacağı işlem türü.*  
*Veri bir kere alınacak ve dinleme işlemi olmayacak.*  
*Verileri dinlemek için*  
*Sadece işlem yapar ve sonuç dönmez.*

## Dao Interface Sınıfı

```

@Dao ← Dao notasyonu
interface KisilerDao {
    @Query( value: "SELECT * FROM kisiler")
    suspend fun tumKisiler(): List<Kisiler>

    @Query( value: "SELECT * FROM kisiler")
    fun tumKisilerFlow(): Flow<List<Kisiler>>

    @Insert
    suspend fun kisiEkle(kisi: Kisiler)

    @Update
    suspend fun kisiGuncelle(kisi: Kisiler)

    @Delete
    suspend fun kisiSil(kisi: Kisiler)

    @Query( value: "SELECT count(*) FROM kisiler WHERE kisi_ad=:kisi_ad")
    suspend fun kayıtKontrol(kisi_ad: String):Int

    @Query( value: "SELECT * FROM kisiler WHERE kisi_id=:kisi_id")
    suspend fun kisiGetir(kisi_id: Int): Kisiler

    @Query( value: "SELECT * FROM kisiler WHERE kisi_ad like '%' || :aramaKelimesi || '%'")
    suspend fun kisiArama(aramaKelimesi: String): List<Kisiler>

    @Query( value: "SELECT * FROM kisiler ORDER BY RANDOM() LIMIT 2")
    suspend fun rasgele2KisiGetir(): List<Kisiler>
}

```

- Veritabanı üzerinde işlem yapacak metodlarını oluşturduğumuz sınıfır.
- İşlemleri asenkron olarak yapabilmek için Coroutine kullanılmıştır.

## 5

# Veritabanı sınıfı oluşturma

- Veritabanı üzerinde çalışabilmek için bu sınıfı oluşturmalıyız.
- Bu sınıf'a istediğimiz kadar entity ekleyebiliriz.

```
@Database(entities = [Kisiler::class], version = 1) ← entities kısmına başka sınıfları ekleyebiliriz.  
abstract class Veritabani : RoomDatabase() {  
    abstract fun kisilerDao(): KisilerDao ← Veritabanı üzerinde işlem yapabilmek için dao sınıfına erişmeliyiz.  
  
    companion object {  
        var INSTANCE: Veritabani? = null ← Veritabanı üzerinde çalışmak için nesnemiz.  
  
        fun veritabaniErisim(context: Context): Veritabani? {  
            if (INSTANCE == null){  
  
                synchronized(Veritabani::class){  
  
                    INSTANCE = Room.databaseBuilder(context.applicationContext,  
                        Veritabani::class.java,  
                        name: "rehber.sqlite") ← Veritabanı adı  
                        .createFromAsset(databaseFilePath: "rehber.sqlite") ← Asset dosyasındaki veritabanı adı  
                        .build()  
  
                }  
            }  
            return INSTANCE  
        }  
    }  
}
```

Asset dosyasındaki veritabanı alınır ve otomatik olarak telefona kopyalanır.

## Veritabanına Erişim

```
class MainActivity : AppCompatActivity() {  
    private lateinit var vt:Veritabani  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        vt = Veritabani.veritabaniErisim( context: this)!!
```

# Select - Veri Okuma

suspend olduğu için  
veri okuma işlemi  
gerçekleşir ve işlemi  
biter.

```
@Query( value: "SELECT * FROM kisiler")  
suspend fun tumKisiler(): List<Kisiler>
```

```
lifecycleScope.launch { this: CoroutineScope  
    Log.e( tag: "*****", msg: "TÜM KİŞİLER *****")  
  
    val gelenListe = vt.kisilerDao().tumKisiler()  
  
    for(k in gelenListe){  
        Log.e( tag: "*****", msg: "*****")  
        Log.e( tag: "Kişi id",k.kisi_id.toString())  
        Log.e( tag: "Kişi ad",k.kisi_ad)  
        Log.e( tag: "Kişi yaşı",k.kisi_yas.toString())  
    }  
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Insert – Veri Kaydı

*@Insert*

*suspend fun kisiEkle(kisi: Kisiler)*

```
lifecycleScope.launch { this: CoroutineScope  
    val kisi = Kisiler( kisi_id: 0, kisi_ad: "arda", kisi_yas: 18)  
    vt.kisilerDao().kisiEkle(kisi)  
}
```

Kişi id önemli değildir veritabanı  
kisi\_id'yi otomatik oluşturur.  
Oyüzden öylesine 0 yazıldı.

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Select - Veri Okuma - ( Dinleyerek )

Flow olduğu için veri okuma işlemi gerçekleşir ve veritabanındaki değişimler dinlenir.

```
@Query( value: "SELECT * FROM kisiler")  
fun tumKisilerFlow(): Flow<List<Kisiler>>
```

```
lifecycleScope.launch { this: CoroutineScope  
    Log.e( tag: "*****", msg: "TÜM KİŞİLER FLOW *****")  
  
    vt.kisilerDao().tumKisilerFlow().collect { gelenListe ->  
        for(k in gelenListe){  
            Log.e( tag: "*****", msg: "*****")  
            Log.e( tag: "Kişi id",k.kisi_id.toString())  
            Log.e( tag: "Kişi ad",k.kisi_ad)  
            Log.e( tag: "Kişi yaşı",k.kisi_yas.toString())  
        }  
    }  
}
```

Not : Bunu test etmek için bir button altında ekleme işlemi yapılrsa en son sonuçlar otomatik olarak konsola düşecektir.

Verileri alabiliriz ve veritabanındaki değişimleri dinleyebiliriz.  
Tabloda değişim olduğu anda bu dinleme çalışır ve en son verileri alır.

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Update – Veri Güncelleme

```
@Update  
suspend fun kisiGuncelle(kisi: Kisiler)
```

```
lifecycleScope.launch { this: CoroutineScope  
    val kisi = Kisiler( kisi_id: 3, kisi_ad: "yeni ece", kisi_yas: 99)  
    vt.kisilerDao().kisiGuncelle(kisi)  
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

Güncelleme yaparken nesnenin  
kisi\_id kısmına güncelleme  
yapılacak kisi\_id eklenir.  
Diğer alanlara güncellemek  
istediğimiz alanlar yazılır.

# Delete – Veri Silme

```
@Delete  
suspend fun kisiSil(kisi: Kisiler)
```

```
lifecycleScope.launch { this: CoroutineScope  
    val kisi = Kisiler( kisi_id: 3, kisi_ad: "ece", kisi_yas: 99)  
    vt.kisilerDao().kisiSil(kisi)  
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

Silme yaparken nesnenin `kisi_id` kısmına silme yapılacak `kisi_id` eklenir.  
Diğer alanlara silme işleminde önemli değildir.

# Select - Kayıt Kontrol

```
@Query( value: "SELECT count(*) FROM kisiler WHERE kisi_ad=:kisi_ad")  
suspend fun kayitKontrol(kisi_ad: String):Int
```

```
lifecycleScope.launch { this: CoroutineScope  
    val sonuc = vt.kisilerDao().kayitKontrol( kisi_ad: "ahmet")  
    Log.e( tag: "SONUÇ",sonuc.toString())  
}
```

Sorgu içinde parametre kullanma : ve daha sonra metod parametre adı yazılarak ile yapılır.

Girilen isimden tabloda kaç tane olduğunu gösterir.

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Select – Bir Kayıt Getirme

```
@Query( value: "SELECT * FROM kisiler WHERE kisi_id=:kisi_id")
suspend fun kisiGetir(kisi_id: Int): Kisiler
```

```
lifecycleScope.launch { this: CoroutineScope
    val kisi = vt.kisilerDao().kisiGetir( kisi_id: 2)

    Log.e( tag: "*****", msg: "KİŞİ *****")
    Log.e( tag: "Kişi id",kisi.kisi_id.toString())
    Log.e( tag: "Kişi ad",kisi.kisi_ad)
    Log.e( tag: "Kişi yaşı",kisi.kisi_yas.toString())
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Select – Arama İşlemi

```
@Query( value: "SELECT * FROM kisiler WHERE kisi_ad like '%' || :aramaKelimesi || '%'")  
suspend fun kisiArama(aramaKelimesi: String): List<Kisiler>
```

```
lifecycleScope.launch { this: CoroutineScope  
    Log.e( tag: "*****", msg: "ARAMA *****")  
  
    val gelenListe = vt.kisilerDao().kisiArama( aramaKelimesi: "e")  
  
    for(k in gelenListe){  
        Log.e( tag: "*****", msg: "*****")  
        Log.e( tag: "Kişi id",k.kisi_id.toString())  
        Log.e( tag: "Kişi ad",k.kisi_ad)  
        Log.e( tag: "Kişi yaşı",k.kisi_yas.toString())  
    }  
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

# Select – Rasgele ve Sınırlı Sayıda

```
@Query( value: "SELECT * FROM kisiler ORDER BY RANDOM() LIMIT 2")
suspend fun rasgele2KisiGetir(): List<Kisiler>
```

```
lifecycleScope.launch { this: CoroutineScope
    Log.e( tag: "*****", msg: "RASGELE *****")

    val gelenListe = vt.kisilerDao().rasgele2KisiGetir()

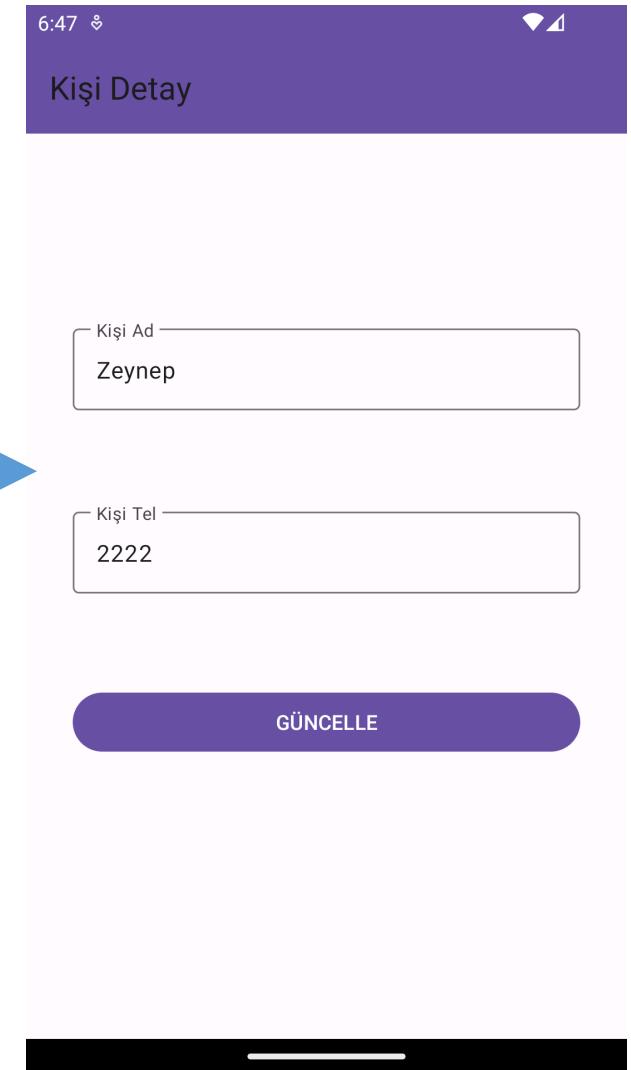
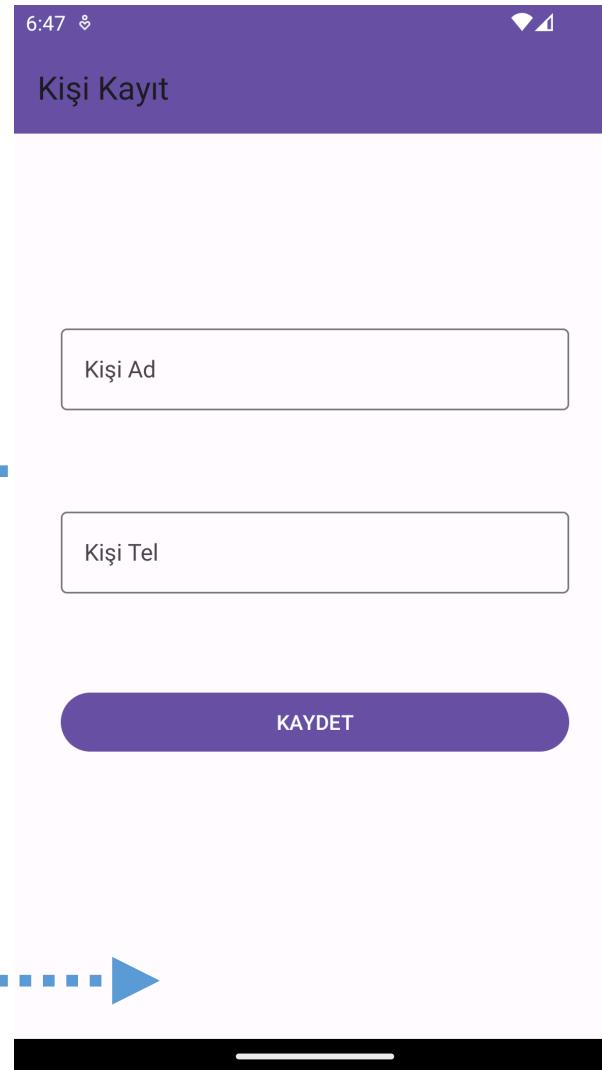
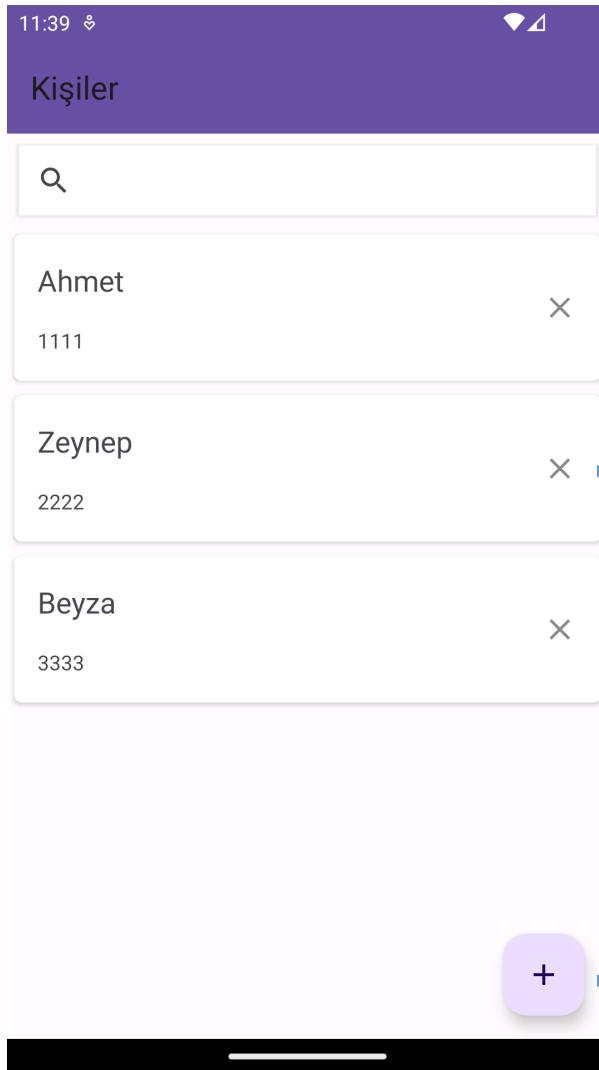
    for(k in gelenListe){
        Log.e( tag: "*****", msg: "*****")
        Log.e( tag: "Kişi id",k.kisi_id.toString())
        Log.e( tag: "Kişi ad",k.kisi_ad)
        Log.e( tag: "Kişi yaşı",k.kisi_yas.toString())
    }
}
```

| kisiler  | Türü   |
|----------|--------|
| kisi_id  | int    |
| kisi_ad  | String |
| kisi_yas | int    |

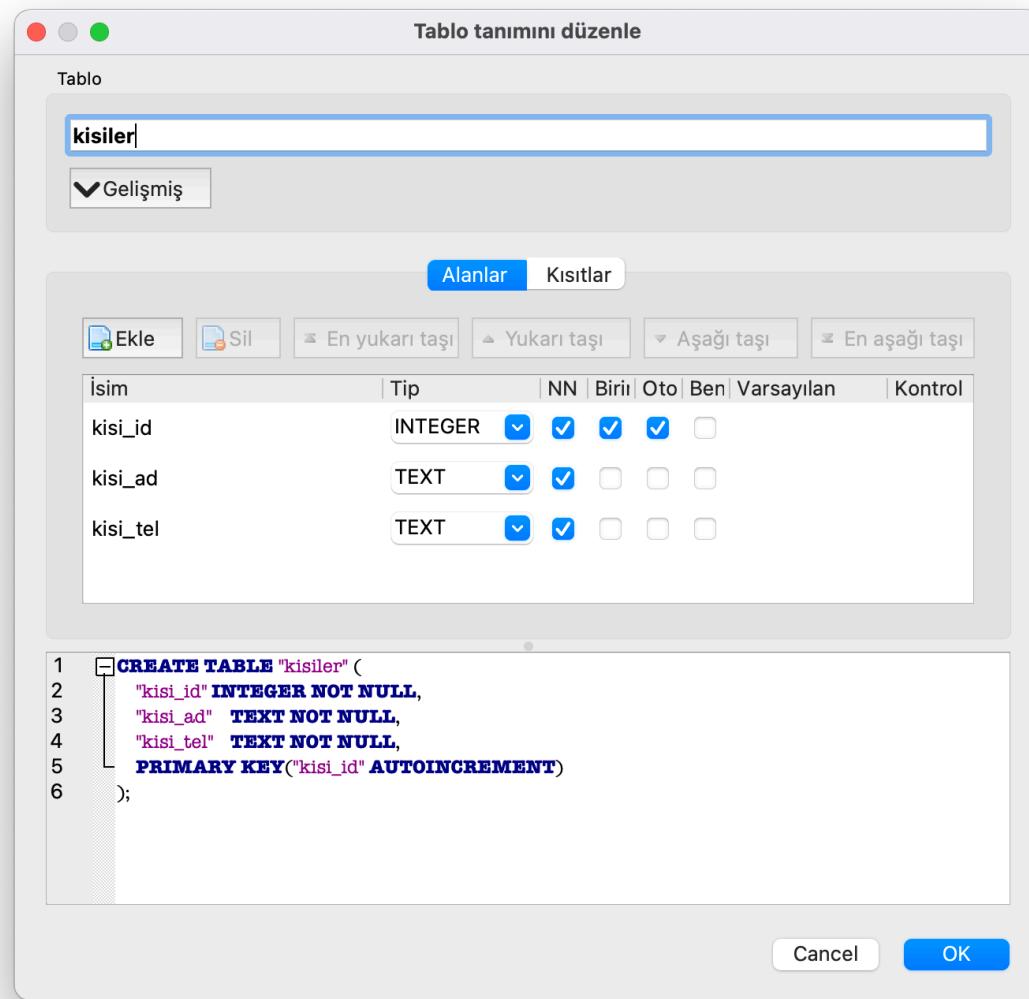
# Kişiler Uygulaması

## Room

# Kişiler Uygulaması



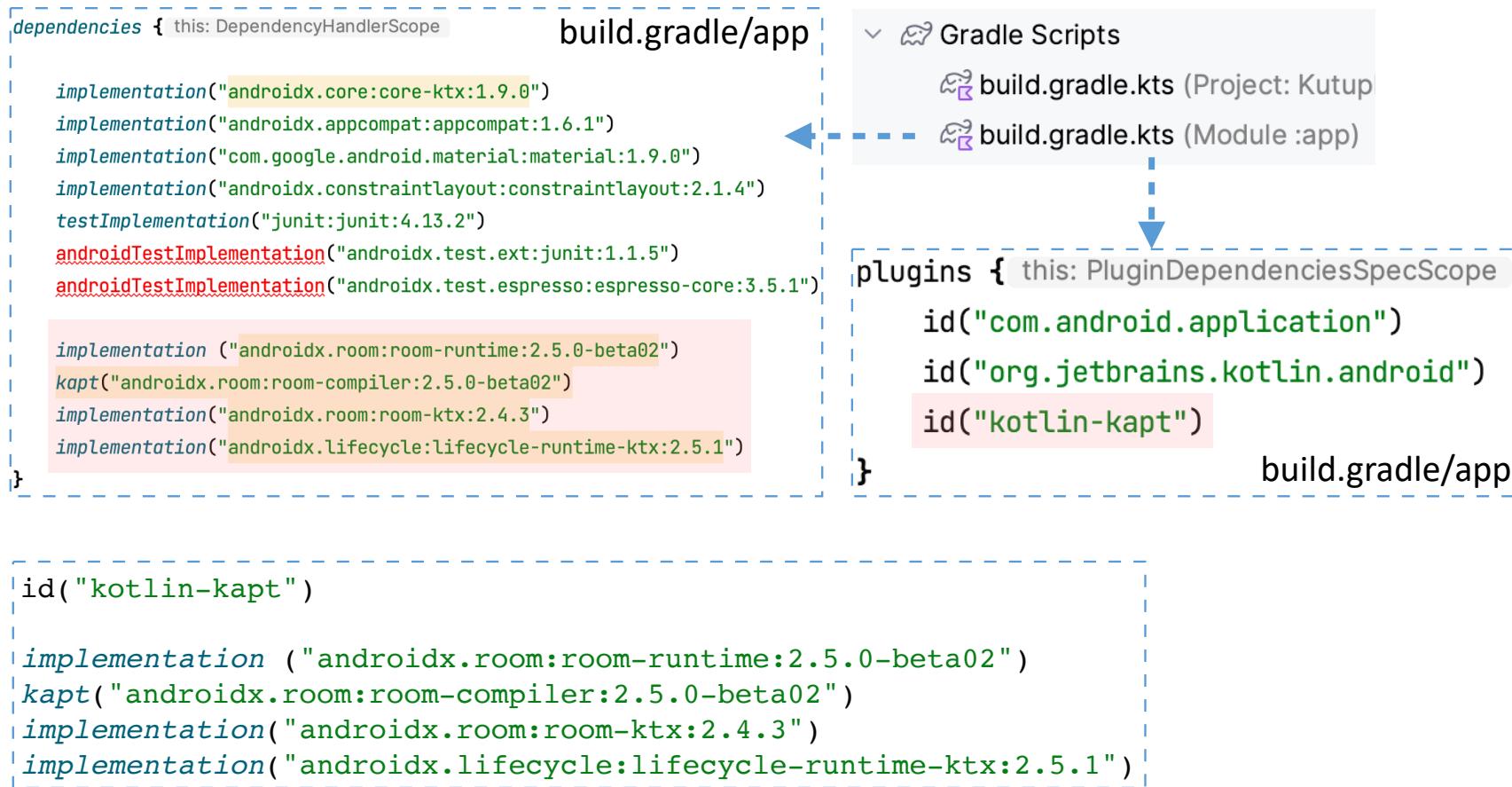
# Kişiler Uygulaması - Room



Tablo: kisiler

|        | kisi_id | kisi_ad | kisi_tel |
|--------|---------|---------|----------|
| Filtre | Filtre  | Filtre  |          |
| 1      | 1       | Ali     | 9 9 9 9  |
| 2      | 2       | Ece     | 8 8 8 8  |

# Kurulum



# Kapt Kurulum Hatası Çözümü

```
compileOptions { this: CompileOptions
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
kotlinOptions { this: KotlinJvmOptions
    jvmTarget = "1.8"
}

dependencies {
    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation ("androidx.room:room-runtime:2.5.0-beta02")
    kapt("androidx.room:room-compiler:2.5.0-beta02")
    implementation("androidx.room:room-ktx:2.4.3")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.5.1")
}
```

build.gradle/app

*Not : Hata alırsanız bunu ekleyebilirsiniz.*

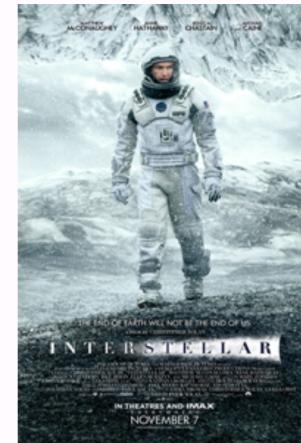
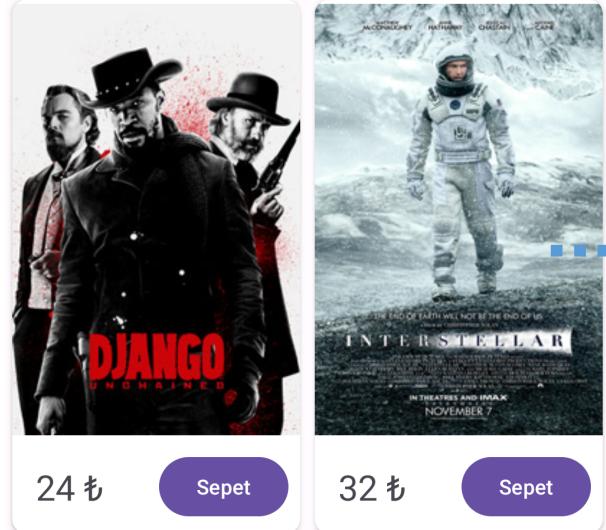
```
compileOptions {
    sourceCompatibility = JavaVersion.VERSION_1_7
    targetCompatibility = JavaVersion.VERSION_1_7
}
kotlinOptions {
    jvmTarget = "17"
}
```

- ▼ 🐍 Gradle Scripts
  - 🐍 build.gradle.kts (Project: Kutup)
  - 🐍 build.gradle.kts (Module :app)

# Filmler Uygulaması

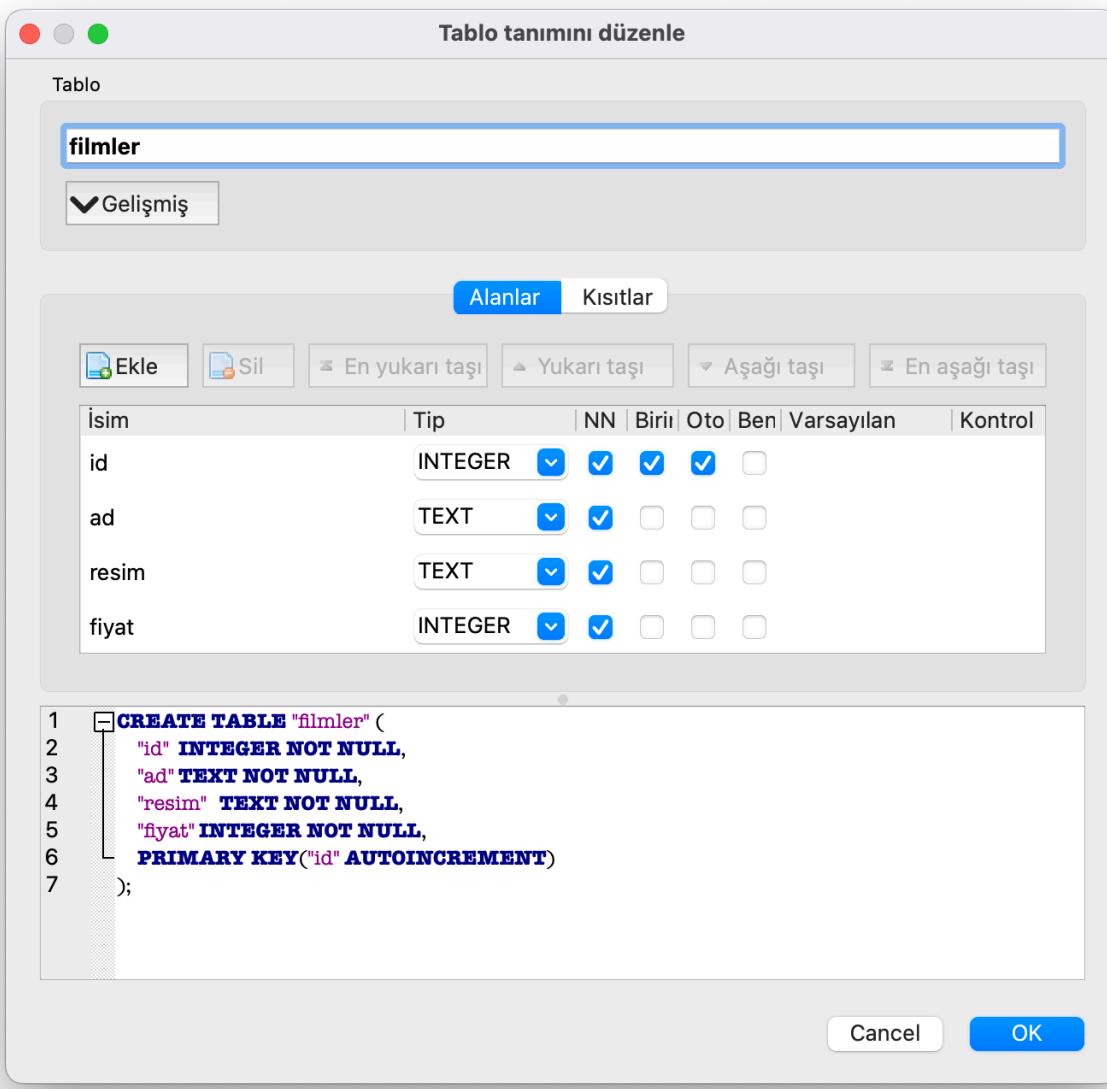
## Room

# Film Uygulaması



32 ₺

# Filmler Uygulaması - Room

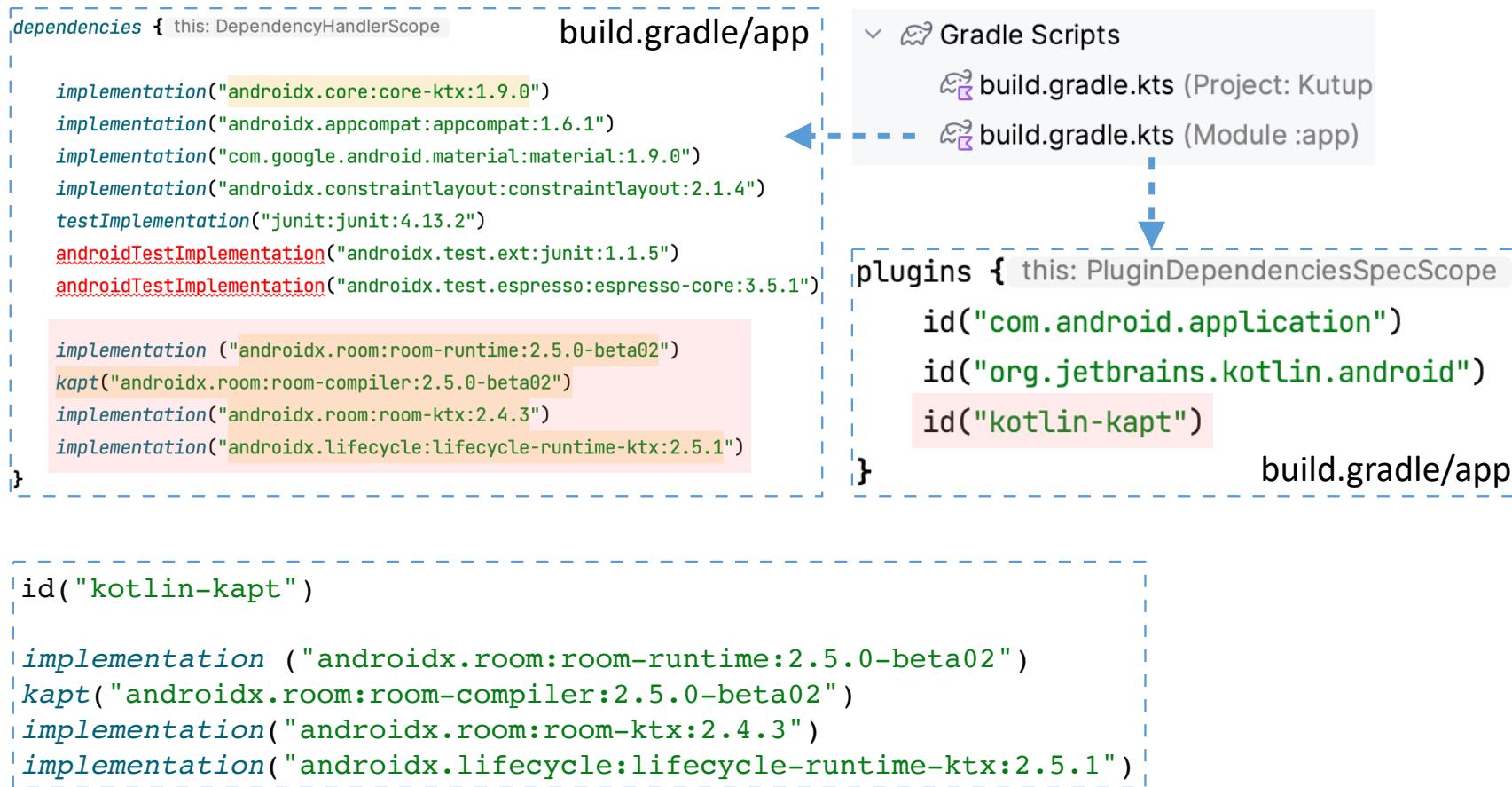


Tablo: **filmers**

En haut En bas En haut En bas

|   | <b>id</b> | <b>ad</b>         | <b>resim</b>    | <b>fiyat</b> |
|---|-----------|-------------------|-----------------|--------------|
|   | Filtre    | Filtre            | Filtre          | Filtre       |
| 1 | 1         | Django            | django          | 2 4          |
| 2 | 2         | Interstellar      | interstellar    | 3 2          |
| 3 | 3         | Inception         | inception       | 1 6          |
| 4 | 4         | The Hateful Eight | thehatefuleight | 2 8          |
| 5 | 5         | The Pianist       | thepianist      | 1 8          |
| 6 | 6         | Anadoluda         | anadoluda       | 1 0          |

# Kurulum



# Kapt Kurulum Hatası Çözümü

```
compileOptions { this: CompileOptions
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
kotlinOptions { this: KotlinJvmOptions
    jvmTarget = "1.8"
}

dependencies {
    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation ("androidx.room:room-runtime:2.5.0-beta02")
    kapt("androidx.room:room-compiler:2.5.0-beta02")
    implementation("androidx.room:room-ktx:2.4.3")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.5.1")
}
```

build.gradle/app

*Not : Hata alırsanız bunu ekleyebilirsiniz.*

```
compileOptions {
    sourceCompatibility = JavaVersion.VERSION_1_7
    targetCompatibility = JavaVersion.VERSION_1_7
}
kotlinOptions {
    jvmTarget = "17"
}
```

Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

# İnternet Tabanlı İşlemler

# RESTful Mimarisi Aşamaları

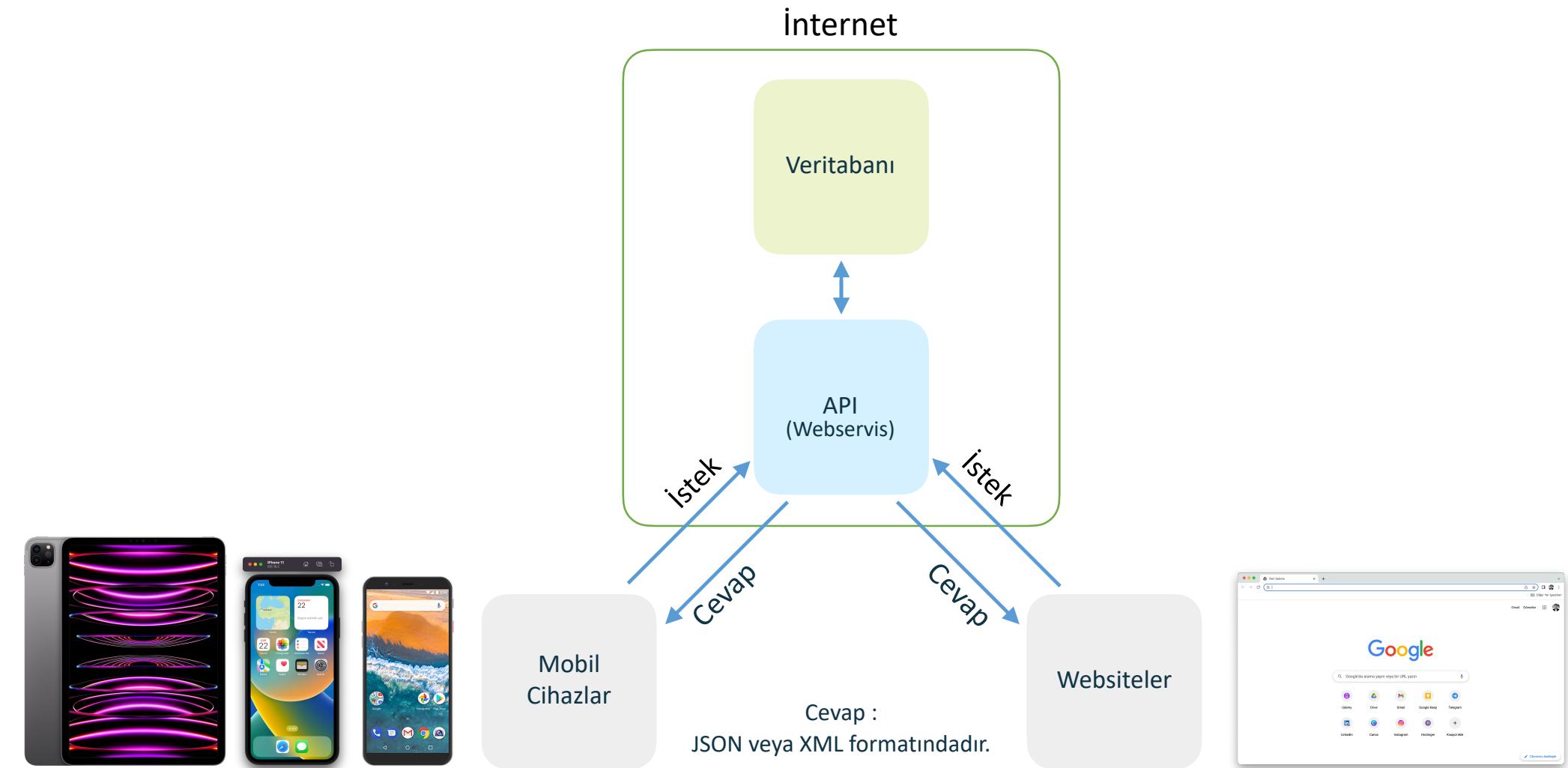
# RESTful Mimarisi Aşamaları

- RESTful Mimarisi Giriş
- Domain Adresi Alma
- Hosting Hizmeti Alma
- Veritabanı Oluşturma
- Veritabanı Üzerinde Tablo Oluşturma
- Veritabanı - Insert - Update - Delete İşlemleri
- Veritabanı - Select Sorguları
- Webservis Giriş
- Veritabanı Bağlantı Bilgileri Dosyası Oluşturma
- Webservis Oluşturma
- Webservisin Hosting İşlemi
- Webservisin Çalıştırılması

# RESTful Web Services

- REST : Representational State Transfer
- Web servise , Bilgileri basit bir formatta transfer etmek için kullanılır.
- XML ve JSON en çok kullanılan formattır.
- Web tabanlı olduğu için internet izni istenmektedir.

# RESTful Mimarisi



# Domain Adresi Alma

- İnternet üzerinde veritabanı ve dosyalama işlemleri için hosting hizmeti almamız gereklidir.
- Hosting hizmeti içinde bir tane domain adresi olması gereklidir.
- Bazı hosting hizmetlerinde ücretsiz domain verilmektedir.
- Domain adresi internet hizmetinize erişeceğiniz website ismi gibi düşünebilirsiniz.
- Örn : kasimadalan.pe.hu

Home | hPanel

hpanel.hostinger.com

HOSTINGER Home Websites Hosting Emails Domains VPS SSL Billing

Hi, KASIM 🙌

Search

**Hosting**

**Single Web Hosting**  
Expires on 2024-06-23 [kasimadalan.pe.hu](#) Manage

**Email**

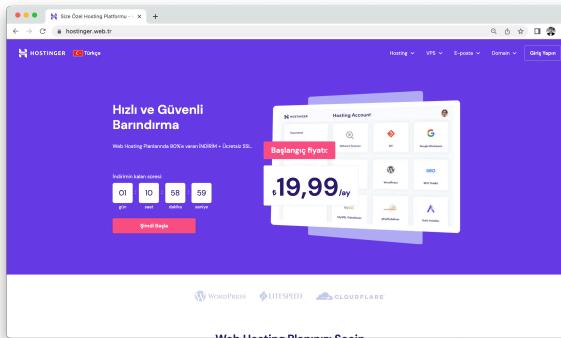
**kasimadalan.pe.hu** Manage

hpanel-main.hostinger.com bekleniyor...

350

# Hosting Hizmeti Alma

- İnternet üzerinde veritabanı ve dosyalama işlemleri için hosting hizmeti almamız gereklidir.
- Hosting hizmeti temelde 24 saat sürekli çalışan bilgisayarları kullanmanızı sağlar.
- Bu hizmet genellikle ücretlidir.
- Hizmeti alırken veritabanı kullanımı sağlandığından emin olunmalıdır.
- Ek olarak dosyalama sistemi var mı , phpMyAdmin arayüzünü destekliyor mu bunlar araştırılmalıdır.



# Veritabanı Oluşturma

The screenshot shows the Hostinger hPanel dashboard interface. At the top, there's a header bar with the Hostinger logo, navigation links for Home, Websites, Hosting, Emails, Domains, VPS, SSL, and Billing, and a user profile icon. Below the header is a search bar and a dropdown menu for the website name, currently set to "kasimadalan.pe.hu". The main dashboard area has a title "Dashboard" and a breadcrumb trail "Hosting - kasimadalan.pe.hu". It displays several service status cards: "Single Web Hosting" (Active), "Domain" (Manage), "Hostinger Free Email" (Active), "Daily backups" (Disabled), "File manager" (Manage), "Databases" (Manage), "Auto installer" (Manage), "Loading time" (Run page speed test), and a "Safe" status card indicating "No malware found" with a "See malware scanner" link. A large green checkmark icon with the text "Your website is running smoothly" and "No issues were found" is prominently displayed. On the left side, a sidebar lists various management categories: Dashboard (selected), Hosting, Performance, Security, Emails, Domains, Website, and Files.

# Veritabanı Oluşturma

Management



- Hosting - kasimadalan.pe.hu - Databases - Management

## + Create a New MySQL Database And Database User

MySQL database name

u139539474\_

db

MySQL username

u139539474\_

admin

Password \*

123456Ab

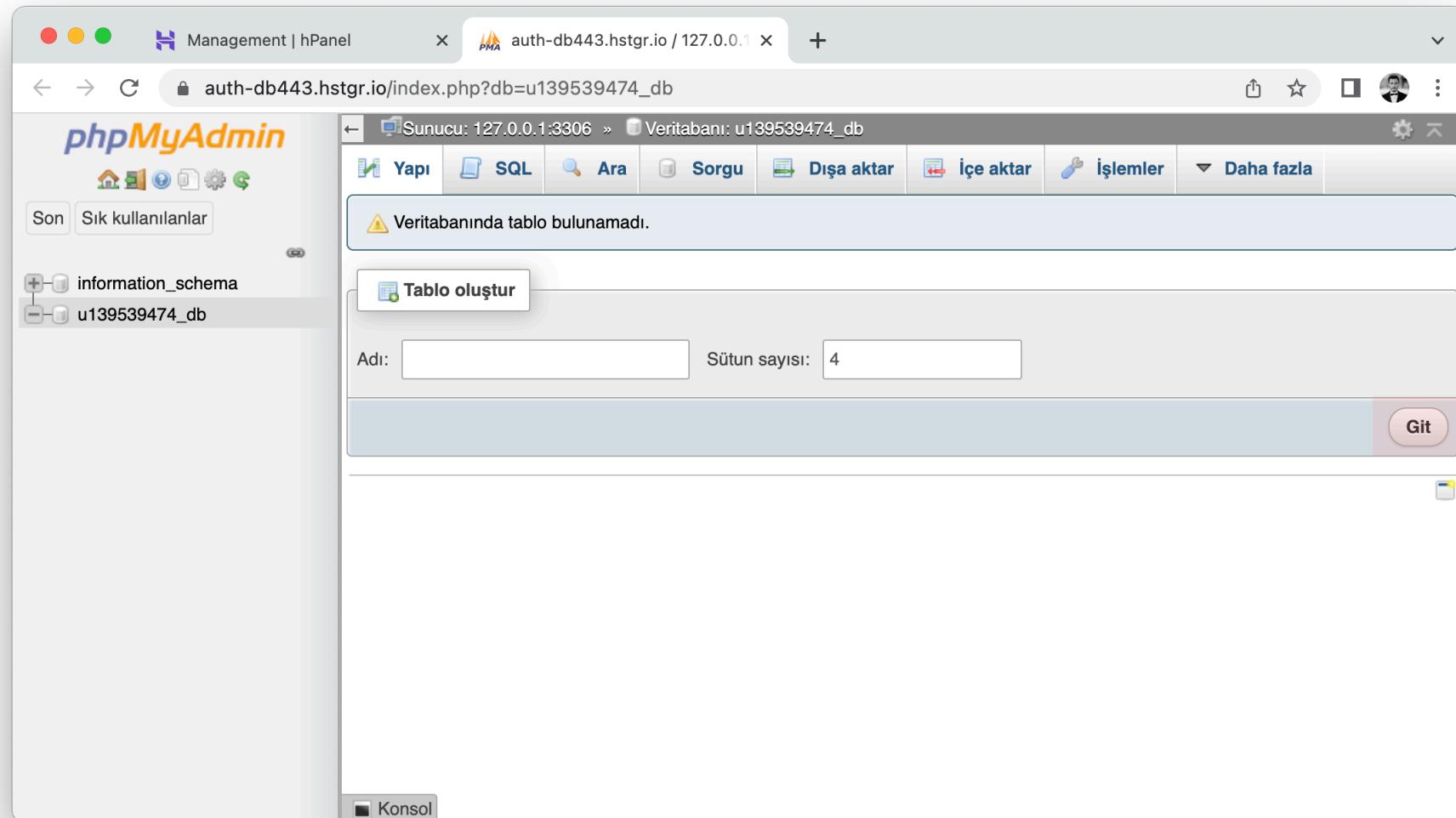


✓ Create

# Veritabanı Oluşturma

| List of Current MySQL Databases And Users |                  |            |                          |                                  |   |
|---|------------------|------------|--------------------------|----------------------------------|---|
| MySQL Database                            | MySQL User       | Created at | Website                  |                                  |   |
| u139539474_db<br>1 MB                     | u139539474_admin | 2023-02-22 | kasimadalan.pe.hu        | <a href="#">Enter phpMyAdmin</a> | ⋮ |
| u139539474_movie<br>4 MB                  | u139539474_root  | 2020-05-28 | <a href="#">+ Assign</a> | <a href="#">Enter phpMyAdmin</a> | ⋮ |

# Veritabanı Oluşturma



# Veritabanı Üzerinde Tablo Oluşturma

**Tablo oluştur**

Adı: kisiler Sütun sayısı: 3

**Git**

---

Gözat Yapı SQL Ara Ekle Dışa aktar İçé aktar İşlemler Tetikleyiciler

Tablo adı: kisiler Ekle 1 sütun(lar) Git

| Adı      | Türü    | Uzunluk/Değerler | Varsayılan | Karşılaştırma | Öznitelikler | Boş Index | A.I   |
|----------|---------|------------------|------------|---------------|--------------|-----------|---|
| kisi_id  | INT     |                  | Yok        |               |              | PRIMARY   | <input checked="" type="checkbox"/> PRIMARY |
| kisi_ad  | VARCHAR | 100              | Yok        |               |              | ---       | <input type="checkbox"/>                    |
| kisi_tel | VARCHAR | 100              | Yok        |               |              | ---       | <input type="checkbox"/>                    |

**Yapı**

Tablo açıklamaları: Karşılaştırma: Depolama Motoru: InnoDB

**PARTITION tanımı:**

Bölümleyen: ( İfade veya sütun listesi )

Bölümler:

SQL Önizle Kaydet

# Veritabanı Üzerinde Tablo Oluşturma

The screenshot shows the MySQL Workbench interface with the 'SQL' tab selected. A table named 'kisi' is being created with the following structure:

| # | Adı      | Türü         | Karşılaştırma      | Öznitelikler | Boş   | Varsayılan | Açıklamalar | Ekstra         | Eylem                        |
|---|----------|--------------|--------------------|--------------|-------|------------|-------------|----------------|------------------------------|
| 1 | kisi_id  | int(11)      |                    |              | Hayır | Yok        |             | AUTO_INCREMENT | Değiştir  Kaldır  Daha fazla |
| 2 | kisi_ad  | varchar(100) | utf8mb4_unicode_ci |              | Hayır | Yok        |             |                | Değiştir  Kaldır  Daha fazla |
| 3 | kisi_tel | varchar(100) | utf8mb4_unicode_ci |              | Hayır | Yok        |             |                | Değiştir  Kaldır  Daha fazla |

# Veritabanı Üzerinde Tablo Oluşturma

The screenshot shows the MySQL Workbench interface for creating a new table. The top navigation bar includes tabs for Gözat, Yapı, SQL, Ara, Ekle, Dışa aktar, İçe aktar, İşlemler, and Tetikleyiciler. The 'Ekle' tab is selected.

The main area displays the table structure for 'kisi'. It has three columns: 'kisi\_id' (int(11)), 'kisi\_ad' (varchar(100)), and 'kisi\_tel' (varchar(100)). The 'kisi\_ad' column contains the value 'Ahmet', and the 'kisi\_tel' column contains the value '1111'.

At the bottom, there are buttons for Yeni satır olarak ekle (New row as insert), ve ondan sonra (After this), Önceki sayfaya geri dön (Go back to previous page), SQL Önizle (SQL Preview), Sıfırla (Clear), and Git (Go).

Below the table creation interface, a message says '1 satırla eklemeye devam et' (Continue adding with 1 row).

# Veritabanı Üzerinde Tablo Oluşturma

Gözat **Yapı** SQL Ara Ekle Dışa aktar İçe aktar İşlemler

✓ Gösterilen satır 0 - 1 (toplam 2, Sorgu 0.0013 saniye sürdü.)

```
SELECT * FROM `kisiler`
```

Profil çıkart [ Satır içi düzenle ] [ Düzenle ] [ SQL'i açıkla ] [ PHP kodu oluştur ] [ Yenile ]

```
UPDATE `kisiler` SET `kisi_tel` = '2222' WHERE `kisiler`.`kisi_id` = 2;
```

[ Satır içi düzenle ] [ Düzenle ] [ PHP kodu oluştur ]

Tümünü göster | Satır sayısı: 25 Satırları süz: Bu tabloda ara Anahtara göre sıra

+ Seçenekler

|                          | Düzenle | Kopyala | Sil | kisi_id | kisi_ad | kisi_tel |
|--------------------------|---------|---------|-----|---------|---------|----------|
| <input type="checkbox"/> |         |         |     | 1       | Ahmet   | 1111     |
| <input type="checkbox"/> |         |         |     | 2       | Zeynep  | 2222     |

↑  Tümünü işaretle **Seçilileri:** Düzenle Kopyala Sil Dışa aktar

# Veritabanı Üzerinde İşlemler

Gözat **Yapı** SQL Ara Ekle Dışa aktar İçe aktar İşlemler Tetikleyiciler

u139539474\_db.kisiler tablosu üzerinde SQL sorgusunu/sorgularını çalıştır: [?](#)

```
1 SELECT * FROM `kisiler`
```

| kisi_id | kisi_ad | kisi_tel |
|---------|---------|----------|
|---------|---------|----------|

SELECT \*        [<<](#)

Parametreleri bağla [?](#)

Sınırlayıcı :   Bu sorguyu burada tekrar göster  Sorgu kutusunu tut  Tamamlandığında geri döndür  Dış anahtar denetlemelerini etkinleştir [Git](#)

# Veri Kaydı - Insert

```
INSERT INTO `kisiler`(`kisi_ad`, `kisi_tel`) VALUES ('Beyza', '9999')
```

| kisi_id | kisi_ad | kisi_tel |
|---------|---------|----------|
| 1       | Ahmet   | 1111     |
| 2       | Zeynep  | 2222     |
| 4       | Beyza   | 9999     |

# Veri Güncelleme - Update

```
UPDATE `kisiler` SET `kisi_ad`='Yeni Beyza', `kisi_tel`='0000' WHERE kisi_id = 4
```

| kisi_id | kisi_ad    | kisi_tel |
|---------|------------|----------|
| 1       | Ahmet      | 1111     |
| 2       | Zeynep     | 2222     |
| 4       | Yeni Beyza | 0000     |

# Veri Silme - Delete

```
DELETE FROM `kisiler` WHERE kisi_id = 4
```

| kisi_id | kisi_ad | kisi_tel |
|---------|---------|----------|
| 1       | Ahmet   | 1111     |
| 2       | Zeynep  | 2222     |

# Select Sorgusu

- Tüm verileri getir

```
SELECT * FROM `kisiler`
```

- İstenilen alanları getir

```
SELECT kisi_ad,kisi_tel FROM `kisiler`
```

# WHERE Şartı

- Eşitlik

```
SELECT * FROM `kisiler` WHERE kisi_ad = 'Ahmet'
```

- Kayıt kontrol

```
SELECT COUNT(*) as sonuc FROM `kisiler` WHERE kisi_ad = 'Ahmet'
```

# ORDER BY - Sıralama

- Kişiler tablosundaki kisi\_adlarına göre artan şekilde getirir.

```
SELECT * FROM `kisiler` ORDER BY kisi_ad ASC
```

- Kişiler tablosundaki kisi\_adlarına göre azalan şekilde getirir.

```
SELECT * FROM `kisiler` ORDER BY kisi_ad DESC
```

- Varsayılsan sıralama ASC dir.

# LIKE - Benzerlik Arama

- Aratılan ifade ilgili alan içeriyor mu şeklinde arama yapılır.

```
SELECT * FROM `kisiler` WHERE kisi_ad LIKE '%e%'
```

# LIMIT - Sınırlı Sayıda Veri Getirme

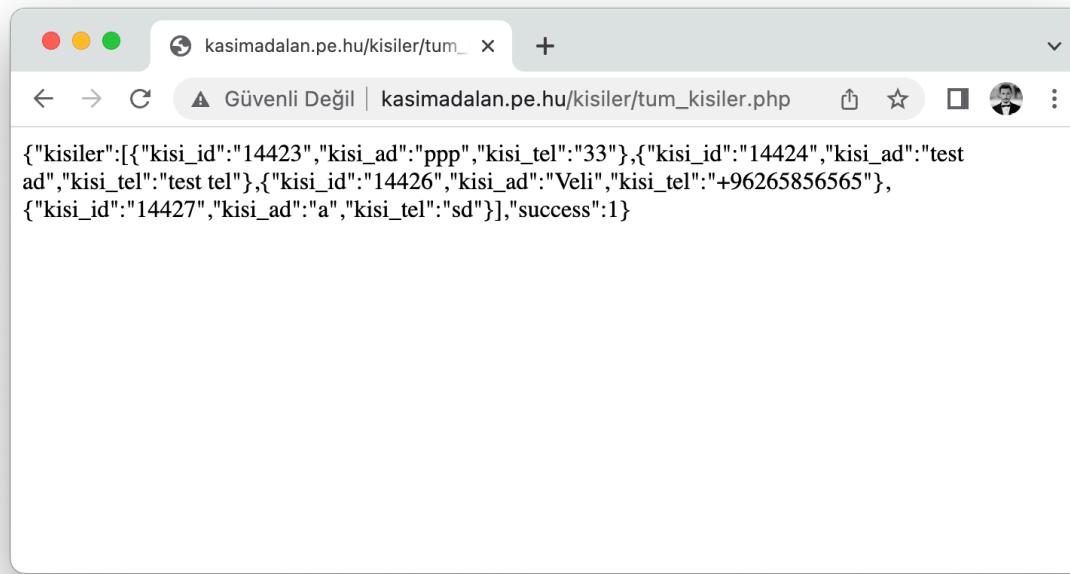
```
SELECT * FROM `kisiler` LIMIT 1
```

```
SELECT * FROM `kisiler` WHERE kisi_ad = 'Ahmet' LIMIT 1
```

```
SELECT * FROM `kisiler` ORDER BY RAND() LIMIT 1
```

# API - Webservis Oluşturma

- Webservisler internet üzerindeki veritabanından verileri almamızı sağlayan metodlardır.
- Url olarak çalışırlar.
- İstediğiniz dilde webservis oluşturabilirsiniz.



# Veritabanı Bağlantı Bilgileri Dosyası Oluşturma

- Bağlanmak istediğimiz veritabanı ile ilgi bilgileri bir dosya içinde oluşturmalıyız.
- Bu bilgiler ile veritabanı bağlantısı oluşturulabiliriz.
- 

List of Current MySQL Databases And Users

| MySQL Database           | MySQL User       | Created at | Website                  |  |
|--------------------------|------------------|------------|--------------------------|--|
| u139539474_db<br>1 MB    | u139539474_admin | 2023-02-22 | kasimadalan.pe.hu        | <a href="#">Enter phpMyAdmin</a> <a href="#">⋮</a> |
| u139539474_movie<br>4 MB | u139539474_root  | 2020-05-28 | <a href="#">+ Assign</a> | <a href="#">Enter phpMyAdmin</a> <a href="#">⋮</a> |

db\_config.php — Edited

```
<?php
define('DB_USER', "u139539474_admin");
define('DB_PASSWORD', "123456Ab");
define('DB_DATABASE', "u139539474_db");
define('DB_SERVER', "localhost");
?>
```

Line: 9 Col: 1

```
<?php
$response = array();

if (isset($_POST['kisi_id'])) {
    $kisi_id = $_POST['kisi_id'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }
    $sqlsorgu = "DELETE FROM kisiler WHERE kisiler.kisi_id = $kisi_id";

    if (mysqli_query($baglanti, $sqlsorgu)) {

        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {

        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

{ -

    "success":0,  
    "message":"Required field(s) is missing"

}

```
<?php
$response = array();

if (isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "INSERT INTO kisiler (kisi_ad,kisi_tel) VALUES ('$kisi_ad','$kisi_tel')";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

{  "success":0,  
"message":"Required field(s) is missing" }

```
<?php

$response = array();

if (isset($_POST['kisi_id']) && isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_id = $_POST['kisi_id'];
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "UPDATE kisiler SET kisiler.kisi_ad = '$kisi_ad',kisiler.kisi_tel = '$kisi_tel' WHERE
                kisiler.kisi_id = $kisi_id  ";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

{ -

    "success":0,  
    "message":"Required field(s) is missing"

}

```

<?php
// array for JSON response
$response = array();

//DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
require_once __DIR__ . '/db_config.php';

// Bağlantı oluşturuluyor.
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

// Bağlantı kontrolü yapılır.
if (!$baglanti) {
    die("Hatalı bağlantı : " . mysqli_connect_error());
}

$sqlsorgu = "SELECT * FROM kisiler";
$result = mysqli_query($baglanti, $sqlsorgu);

// result kontrolü yap
if (mysqli_num_rows($result) > 0) {

    $response["kisiler"] = array();

    while ($row = mysqli_fetch_assoc($result)) {
        // temp user array
        $kisiler = array();

        $kisiler["kisi_id"] = $row["kisi_id"];
        $kisiler["kisi_ad"] = $row["kisi_ad"];
        $kisiler["kisi_tel"] = $row["kisi_tel"];

        // push single product into final response array
        array_push($response["kisiler"], $kisiler);
    }
    // success
    $response["success"] = 1;
}

// echoing JSON response
echo json_encode($response);

```



```

{
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "kisi_id": "380",
            "kisi_ad": "Mehmet",
            "kisi_tel": "05375483215"
        },
        {
            "kisi_id": "381",
            "kisi_ad": "Ayşe",
            "kisi_tel": "05375483216"
        },
        {
            "kisi_id": "382",
            "kisi_ad": "Mustafa",
            "kisi_tel": "05375483217"
        }
    ],
    "success": 1
}
//bağlantı koparılır.
mysqli_close($baglanti);
?>

```

```
<?php
```

```
$response = array();

if (isset($_POST['kisi_ad'])) {
    $kisi_ad = $_POST['kisi_ad'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "SELECT * FROM kisiler WHERE kisiler.kisi_ad like '%$kisi_ad%'";
    $result = mysqli_query($baglanti, $sqlsorgu);

    if (mysqli_num_rows($result) > 0) {

        $response["kisiler"] = array();

        while ($row = mysqli_fetch_assoc($result)) {

            $kisiler = array();

            $kisiler["kisi_id"] = $row["kisi_id"];
            $kisiler["kisi_ad"] = $row["kisi_ad"];
            $kisiler["kisi_tel"] = $row["kisi_tel"];

            array_push($response["kisiler"], $kisiler);
        }

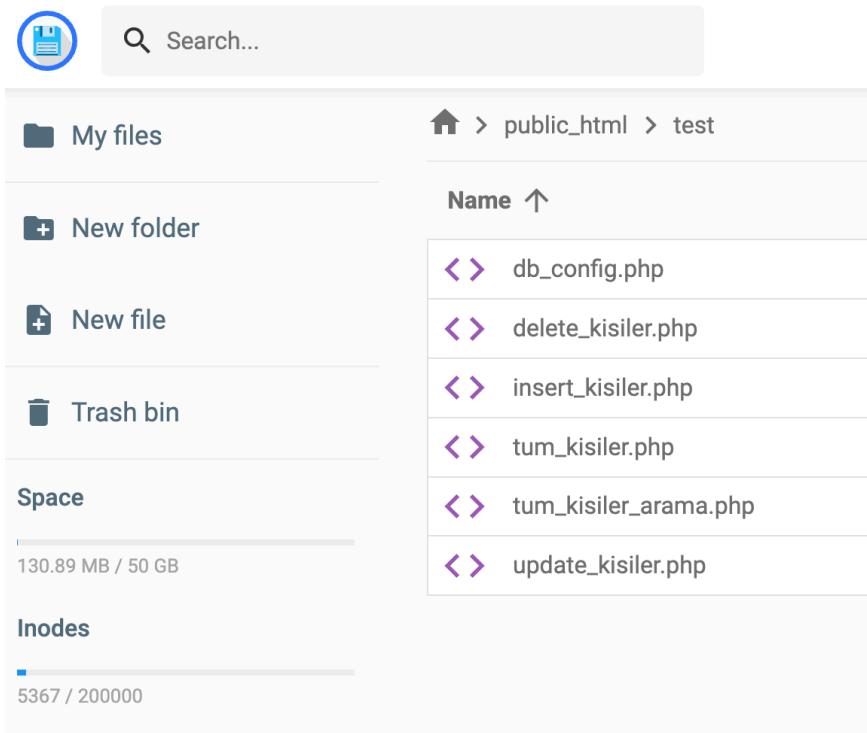
        $response["success"] = 1;
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
}
```

```
{ [
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "+"
        },
        {
            "+"
        },
        {
            "+"
        },
        {
            "+"
        },
        {
            "+"
        }
    ],
    "success": 1
}
```

```
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

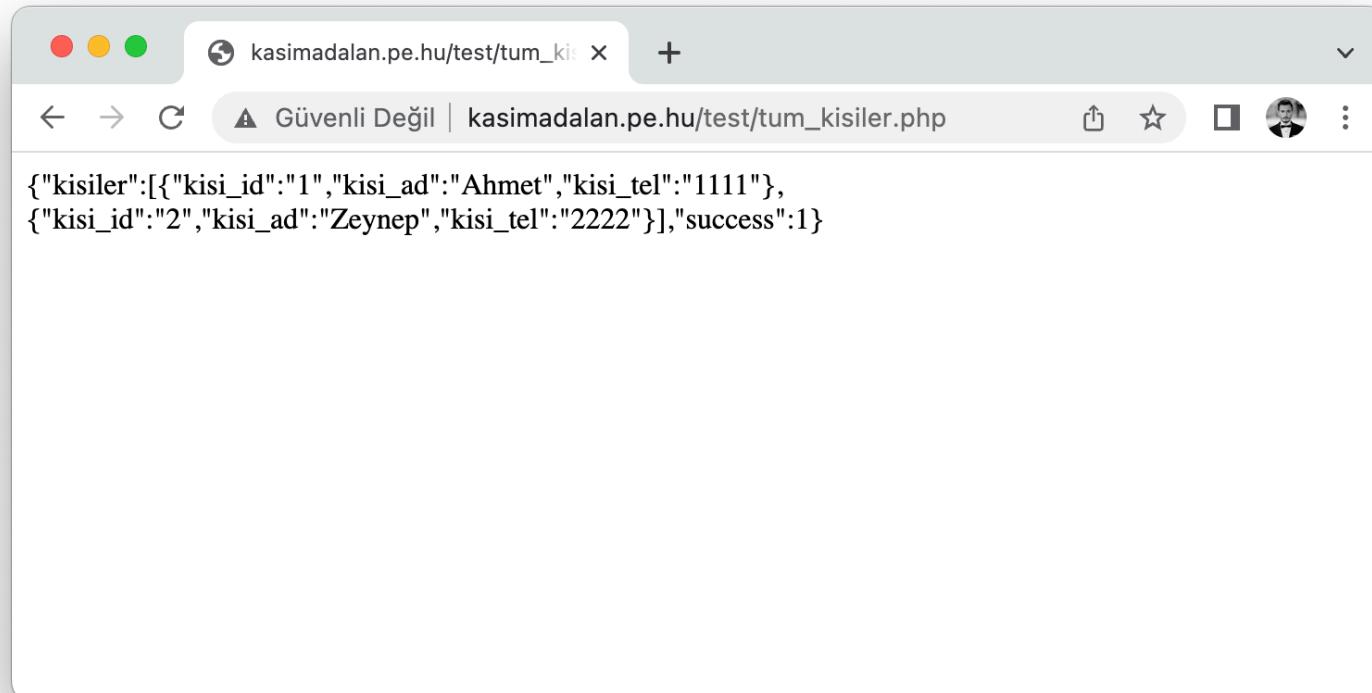
# Webservis Hosting İşlemi

- Webservisleri çalıştırılmak için hosting hizmetinin dosya sistemine yüklenmelidir.



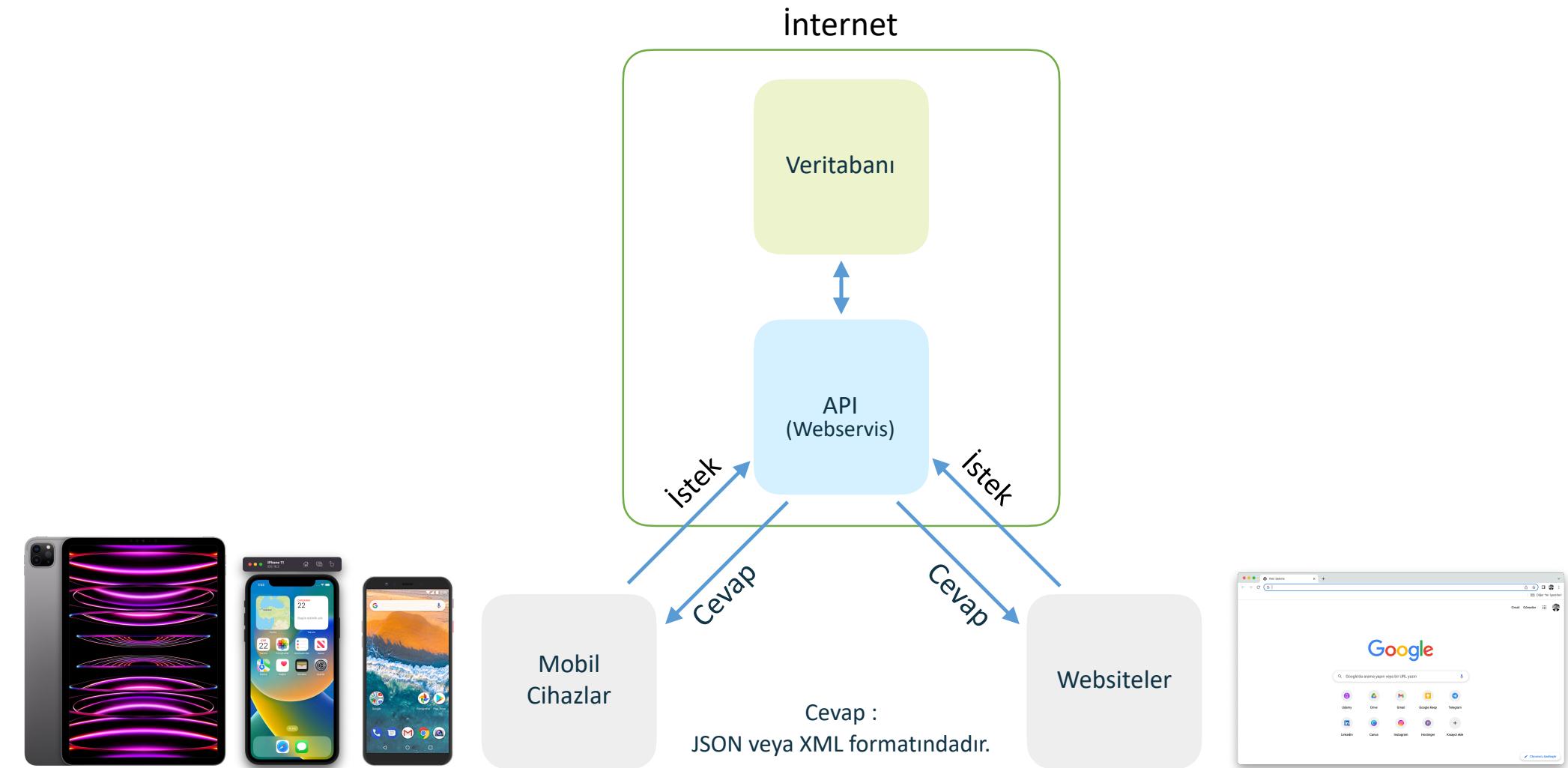
# Webservisin Çalıştırılması

- Webservisin urlsini http isteği çalıştırabilen platformada kullanabiliriz.
- [http://kasimadalan.pe.hu/test/tum\\_kisiler.php](http://kasimadalan.pe.hu/test/tum_kisiler.php)



# JSON Parse İşlemi

# RESTful Mimarisi



# JSON VERİ MODELİ

```
{ [ parantez Json array ( dizidir yani listedir. )  
    "kisiler": [ [ ]  
{ [ parantez Json object ( nesnesidir ) { [ ]  
    "kisi_id": "1",  
    "kisi_ad": "Ahmet",  
    "kisi_tel": "12312312"  
},  
{ [ ]  
    "kisi_id": "2",  
    "kisi_ad": "Mehmet",  
    "kisi_tel": "912318212"  
}  
]  
}
```

Json nesnesi içeriği key ve value şeklinde oluşturulur.

# JSON PARSE ÖRNEK 1

```
try {
    val jsonObject = JSONObject(jsonVeri)

    val bayraklarListe = jsonObject.getJSONArray( name: "bayraklar" )  

        ↑  

        "bayraklar" → JSON Array  

for (i in 0 until bayraklarListe.length()) {  

    val b = bayraklarListe.getJSONObject(i)  

    val bayrak_id = b.getInt( name: "bayrak_id" )  

    val bayrak_ad = b.getString( name: "bayrak_ad" )  

    val bayrak_resim = b.getString( name: "bayrak_resim" )  

    Log.e( tag: "*****", msg: "*****" )
    Log.e( tag: "bayrak_id", bayrak_id.toString() )
    Log.e( tag: "bayrak_ad", bayrak_ad )
    Log.e( tag: "bayrak_resim", bayrak_resim )
    Log.e( tag: "*****", msg: "*****" )

}
} catch (e: JSONException) {
    e.printStackTrace()
}
```

```
{
    "bayraklar": [
        {
            "bayrak_id": "1",
            "bayrak_ad": "T\u00fcrk\u011f",
            "bayrak_resim": "turkiye"
        },
        {
            "bayrak_id": "2",
            "bayrak_ad": "Almanya",
            "bayrak_resim": "almanya"
        },
        {
            "bayrak_id": "3",
            "bayrak_ad": "Fransa",
            "bayrak_resim": "fransa"
        },
        {
            "bayrak_id": "4",
            "bayrak_ad": "Bir\u0131n\u011fik Krall\u00f3k",
            "bayrak_resim": "bir\u0131n\u011fik_krall\u00f3k"
        },
        {
            "bayrak_id": "5",
            "bayrak_ad": "Italya",
            "bayrak_resim": "italya"
        },
        {
            "bayrak_id": "6",
            "bayrak_ad": "G\u00fcr\u011f",
            "bayrak_resim": "g\u00fcr\u011f"
        },
        {
            "bayrak_id": "7",
            "bayrak_ad": "Portekiz",
            "bayrak_resim": "portekiz"
        },
        {
            "bayrak_id": "8",
            "bayrak_ad": "S\u00fcrel",
            "bayrak_resim": "s\u00fcrel"
        },
        {
            "bayrak_id": "9",
            "bayrak_ad": "Danimarka",
            "bayrak_resim": "danimarka"
        },
        {
            "bayrak_id": "10",
            "bayrak_ad": "Norve\u011f",
            "bayrak_resim": "norve\u011f"
        }
    ],
    "success": 1
}
```

## JSON PARSE ÖRNEK 2

```
try {
    val jsonObject = JSONObject(jsonFilmVeri)

    val filmlerListe = jsonObject.getJSONArray( name: "filmler" )

    for (i in 0 until filmlerListe.length()) {

        val f = filmlerListe.getJSONObject(i)

        val film_id = f.getInt( name: "film_id" )
        val film_ad = f.getString( name: "film_ad" )
        val film_yil = f.getInt( name: "film_yil" )
        val film_resim = f.getString( name: "film_resim" )

        val kategori = f.getJSONObject( name: "kategori" )
        val kategori_id = kategori.getInt( name: "kategori_id" )
        val kategori_ad = kategori.getString( name: "kategori_ad" )

        val yonetmen = f.getJSONObject( name: "yonetmen" )
        val yonetmen_id = yonetmen.getInt( name: "yonetmen_id" )
        val yonetmen_ad = yonetmen.getString( name: "yonetmen_ad" )
    }

} catch (e: JSONException) {
    e.printStackTrace()
}
```

# Retrofit

# Retrofit Kütüphanesinin kullanımı

Retrofit

A type-safe HTTP client for Android and Java

|                  | <b>One Discussion</b> | <b>Dashboard<br/>(7 requests)</b> | <b>25 Discussions</b> |
|------------------|-----------------------|-----------------------------------|-----------------------|
| <b>AsyncTask</b> | 941 ms                | 4,539 ms                          | 13,957 ms             |
| <b>Volley</b>    | 560 ms                | 2,202 ms                          | 4,275 ms              |
| <b>Retrofit</b>  | 312 ms                | 889 ms                            | 1,059 ms              |

# Kurulum

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.glidekullanimi">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        tools:ignore="FullBackupContent"
        tools:targetApi="28">

```

# Kurulum

```
dependencies { this: DependencyHandlerScope
```

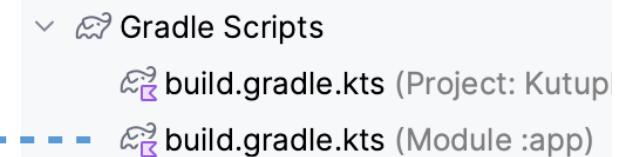
build.gradle/app

```
implementation("androidx.core:core-ktx:1.9.0")
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.9.0")
implementation("androidx.constraintlayout:constraintlayout:2.1.4")
testImplementation("junit:junit:4.13.2")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```

```
}
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```



# **JSON CEVAPLARI İÇİN**

# **SINIF DÖNÜŞÜMLERİ**

# JSON Nesnesi için Sınıf Oluşturma

```
data class Kisiler(var kisi_id:Int,  
                    var kisi_ad:String,  
                    var kisi_tel:String) {  
}
```

```
{  
    "kisiler": [  
        {  
            "kisi_id": "376",  
            "kisi_ad": "Ahmet",  
            "kisi_tel": "5348564412"  
        },  
        {  
            "kisi_id": "379",  
            "kisi_ad": "Zeynep",  
            "kisi_tel": "05375483214"  
        },  
        {  
            "kisi_id": "380",  
            "kisi_ad": "Mehmet",  
            "kisi_tel": "05375483215"  
        },  
        {  
            "kisi_id": "381",  
            "kisi_ad": "Ayşe",  
            "kisi_tel": "05375483216"  
        },  
        {  
            "kisi_id": "382",  
            "kisi_ad": "Mustafa",  
            "kisi_tel": "05375483217"  
        }  
    ],  
    "success": 1  
}
```

# JSON Cevabı için Sınıf Oluşturma

```
data class KisilerCevap(var kisiler>List<Kisiler>,  
                        var success:Int) {  
}
```

```
{  
    "kisiler": [  
        {  
            "kisi_id": "376",  
            "kisi_ad": "Ahmet",  
            "kisi_tel": "5348564412"  
        },  
        {  
            "kisi_id": "379",  
            "kisi_ad": "Zeynep",  
            "kisi_tel": "05375483214"  
        },  
        {  
        },  
        {  
        },  
        {  
        },  
        {  
        }  
    ],  
    "success": 1  
}
```

# JSON Cevabı için Sınıf Oluşturma

```
data class CRUDCevap(var success:Int,  
                      var message:String){  
    {  
        "success":0,  
        "message":"Required field(s) is missing"  
    }  
}
```

### 3. Interface oluşturma

```
interface KisilerDao {  
    @GET("kisiler/tum_kisiler.php")  
    suspend fun kisileriYukle() : KisilerCevap ← Json cevap sınıfları  
  
    @POST("kisiler/tum_kisiler_arama.php")  
    @FormUrlEncoded  
    suspend fun ara(@Field("kisi_ad") aramaKelimesi:String) : KisilerCevap  
  
    @POST("kisiler/insert_kisiler.php")  
    @FormUrlEncoded  
    suspend fun kaydet(@Field("kisi_ad") kisi_ad:String,  
                       @Field("kisi_tel") kisi_tel:String) : CRUDCevap  
  
    @POST("kisiler/update_kisiler.php")  
    @FormUrlEncoded  
    suspend fun guncelle(@Field("kisi_id") kisi_id:Int,  
                         @Field("kisi_ad") kisi_ad:String,  
                         @Field("kisi_tel") kisi_tel:String) : CRUDCevap  
  
    @POST("kisiler/delete_kisiler.php")  
    @FormUrlEncoded  
    suspend fun sil(@Field("kisi_id") kisi_id:Int) : CRUDCevap  
}
```

## 4. Retrofit Nesnesi Alınır

```
class RetrofitClient {  
    companion object {  
        fun getClient(baseUrl:String) : Retrofit {  
            return Retrofit.Builder()  
                .baseUrl(baseUrl)  
                .addConverterFactory(GsonConverterFactory.create())  
                .build()  
        }  
    }  
}
```

## 5. Retrofit nesnesiyle interface'i bağlama

```
class ApiUtils {
    companion object{
        val BASE_URL = "http://kasimadalan.pe.hu/"

        fun getKisilerDao() : KisilerDao {
            return RetrofitClient.getClient(BASE_URL).create(KisilerDao::class.java)
        }
    }
}
```

## 6. Interface metodunun çağrılmaması

```
suspend fun kisileriGetir(){
    val kdao = ApiUtils.getKisilerDao()
    val liste = kdao.kisileriYukle().kisiler

    for(k in liste){
        Log.e( tag: "Kişi id",k.kisi_id.toString())
        Log.e( tag: "Kişi ad",k.kisi_ad)
        Log.e( tag: "Kişi tel",k.kisi_tel)
    }
}
```

```

<?php
// array for JSON response
$response = array();

//DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
require_once __DIR__ . '/db_config.php';

// Bağlantı oluşturuluyor.
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

// Bağlantı kontrolü yapılır.
if (!$baglanti) {
    die("Hatalı bağlantı : " . mysqli_connect_error());
}

$sqlsorgu = "SELECT * FROM kisiler";
$result = mysqli_query($baglanti, $sqlsorgu);

// result kontrolü yap
if (mysqli_num_rows($result) > 0) {

    $response["kisiler"] = array();

    while ($row = mysqli_fetch_assoc($result)) {
        // temp user array
        $kisiler = array();

        $kisiler["kisi_id"] = $row["kisi_id"];
        $kisiler["kisi_ad"] = $row["kisi_ad"];
        $kisiler["kisi_tel"] = $row["kisi_tel"];

        // push single product into final response array
        array_push($response["kisiler"], $kisiler);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
}

```

## tum\_kisiler.php

```

{
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "kisi_id": "380",
            "kisi_ad": "Mehmet",
            "kisi_tel": "05375483215"
        },
        {
            "kisi_id": "381",
            "kisi_ad": "Ayşe",
            "kisi_tel": "05375483216"
        },
        {
            "kisi_id": "382",
            "kisi_ad": "Mustafa",
            "kisi_tel": "05375483217"
        }
    ],
    "success": 1
}

// no products found
$response["success"] = 0;
$response["message"] = "No data found";

// echo no users JSON
echo json_encode($response);
}

//bağlantı koparılır.
mysqli_close($baglanti);
?>

```



```
@GET("kisiler/tum_kisiler.php")
suspend fun kisileriYukle() : KisilerCevap

suspend fun kisileriGetir(){
    val kdao = ApiUtils.getKisilerDao()
    val liste = kdao.kisileriYukle().kisiler

    for(k in liste){
        Log.e( tag: "Kişi id",k.kisi_id.toString())
        Log.e( tag: "Kişi ad",k.kisi_ad)
        Log.e( tag: "Kişi tel",k.kisi_tel)
    }
}
```

```

<?php

$response = array();

if (isset($_POST['kisi_ad'])) {
    $kisi_ad = $_POST['kisi_ad'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "SELECT * FROM kisiler WHERE kisiler.kisi_ad like '%$kisi_ad%'";
    $result = mysqli_query($baglanti, $sqlsorgu);

    if (mysqli_num_rows($result) > 0) {

        $response["kisiler"] = array();

        while ($row = mysqli_fetch_assoc($result)) {

            $kisiler = array();

            $kisiler["kisi_id"] = $row["kisi_id"];
            $kisiler["kisi_ad"] = $row["kisi_ad"];
            $kisiler["kisi_tel"] = $row["kisi_tel"];

            array_push($response["kisiler"], $kisiler);
        }

        $response["success"] = 1;
        echo json_encode($response);
    }
    //bağlanti koparılır.
    mysqli_close($baglanti);
}

```

## tum\_kisiler\_arama.php

```

{
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "kisi_id": "380",
            "kisi_ad": "Mehmet",
            "kisi_tel": "05375483215"
        },
        {
            "kisi_id": "381",
            "kisi_ad": "Ayşe",
            "kisi_tel": "05375483216"
        },
        {
            "kisi_id": "382",
            "kisi_ad": "Mustafa",
            "kisi_tel": "05375483217"
        },
        {
            "kisi_id": "383",
            "kisi_ad": "Fatma",
            "kisi_tel": "05375483218"
        }
    ],
    "success": 1
}

```



```

} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>

```

```
@POST("kisiler/tum_kisiler_arama.php")
@FormUrlEncoded
suspend fun ara(@Field("kisi_ad") aramaKelimesi:String) : KisilerCevap

    suspend fun ara(){
        val kdao = ApiUtils.getKisilerDao()
        val liste = kdao.ara( aramaKelimesi: "s").kisiler

        for(k in liste){
            Log.e( tag: "Kişi id",k.kisi_id.toString())
            Log.e( tag: "Kişi ad",k.kisi_ad)
            Log.e( tag: "Kişi tel",k.kisi_tel)
        }
    }
}
```

```
<?php
$response = array();

if (isset($_POST['kisi_id'])) {
    $kisi_id = $_POST['kisi_id'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlanti oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlanti kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }
    $sqlsorgu = "DELETE FROM kisiler WHERE kisiler.kisi_id = $kisi_id";

    if (mysqli_query($baglanti, $sqlsorgu)) {

        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {

        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

## delete\_kisiler.php

# Çalıştırmak istediğimiz PHP kod

```
@POST("kisiler/delete_kisiler.php")
@FormUrlEncoded
suspend fun sil(@Field("kisi_id") kisi_id:Int) : CRUDCevap

suspend fun sil(){
    val kdao = ApiUtils.getKisilerDao()
    val cevap = kdao.sil( kisi_id: 11028)

    Log.e( tag: "Başarı",cevap.success.toString())
    Log.e( tag: "Mesaj",cevap.message)
}
```

```
<?php
$response = array();

if (isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "INSERT INTO kisiler (kisi_ad,kisi_tel) VALUES ('$kisi_ad','$kisi_tel')";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

## insert\_kisiler.php

# Çalıştmak istediğimiz PHP kod

```
@POST("kisiler/insert_kisiler.php")
@FormUrlEncoded
suspend fun kaydet(@Field("kisi_ad") kisi_ad:String,
                   @Field("kisi_tel") kisi_tel:String) : CRUDCevap

suspend fun kaydet(){
    val kdao = ApiUtils.getKisilerDao()
    val cevap = kdao.kaydet( kisi_ad: "test ad", kisi_tel: "test tel")

    Log.e( tag: "Başarı",cevap.success.toString())
    Log.e( tag: "Mesaj",cevap.message)
}
```

```
<?php

$response = array();

if (isset($_POST['kisi_id']) && isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_id = $_POST['kisi_id'];
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "UPDATE kisiler SET kisiler.kisi_ad = '$kisi_ad',kisiler.kisi_tel = '$kisi_tel' WHERE
                kisiler.kisi_id = $kisi_id ";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

## update\_kisiler.php

# Çalıştırmak istediğimiz PHP kod

```
    @POST("kisiler/update_kisiler.php")
    @FormUrlEncoded
    suspend fun guncelle(@Field("kisi_id") kisi_id:Int,
                         @Field("kisi_ad") kisi_ad:String,
                         @Field("kisi_tel") kisi_tel:String) : CRUDCevap

    suspend fun guncelle(){
        val kdao = ApiUtils.getKisilerDao()
        val cevap = kdao.guncelle( kisi_id: 11043, kisi_ad: "test ad x", kisi_tel: "test tel y")

        Log.e( tag: "Başarı",cevap.success.toString())
        Log.e( tag: "Mesaj",cevap.message)
    }
```

# Glide

# Kurulum

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.glidekullanimi">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        tools:ignore="FullBackupContent"
        tools:targetApi="29">

```

# Kurulum

```
dependencies { this: DependencyHandlerScope  
  
    implementation("androidx.core:core-ktx:1.9.0")  
    implementation("androidx.appcompat:appcompat:1.6.1")  
    implementation("com.google.android.material:material:1.9.0")  
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")  
    testImplementation("junit:junit:4.13.2")  
    androidTestImplementation("androidx.test.ext:junit:1.1.5")  
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
  
    implementation("com.github.bumptech.glide:glide:4.13.2")  
}  
  
implementation("com.github.bumptech.glide:glide:4.13.2")
```

build.gradle/app

Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(LayoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3



```
class MainActivity : AppCompatActivity() {
    private lateinit var tasarim:ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        tasarim = ActivityMainBinding.inflate(layoutInflater)
        setContentView(tasarim.root)

        resimGoster( resimAdi: "kofte.png")

        tasarim.buttonResim1.setOnClickListener { it: View!
            resimGoster( resimAdi: "ayran.png")
        }

        tasarim.buttonResim2.setOnClickListener { it: View!
            resimGoster( resimAdi: "su.png")
        }
    }

    fun resimGoster(resimAdi:String){
        val url = "http://kasimadalan.pe.hu/yemekler/resimler/$resimAdi"
        Glide.with( activity: this).load(url).override( width: 300, height: 300).into(tasarim.imageView)
    }
}
```

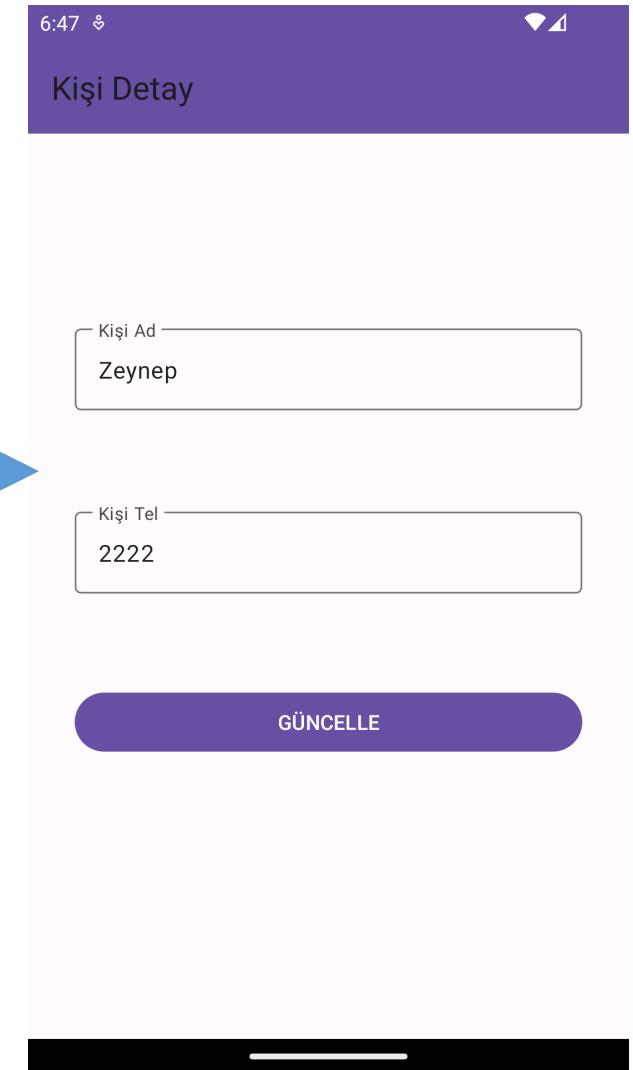
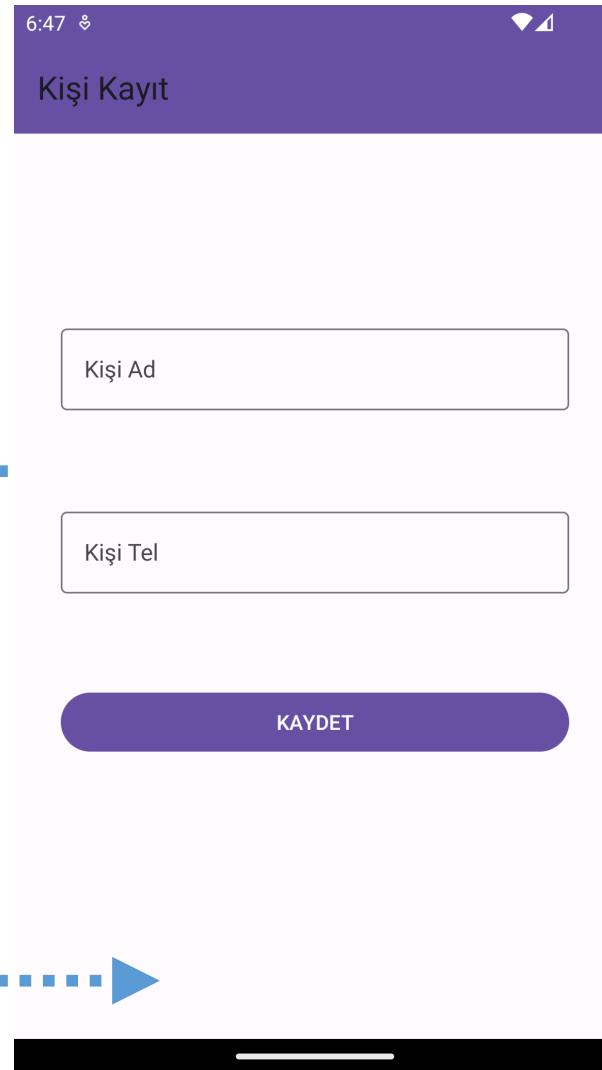
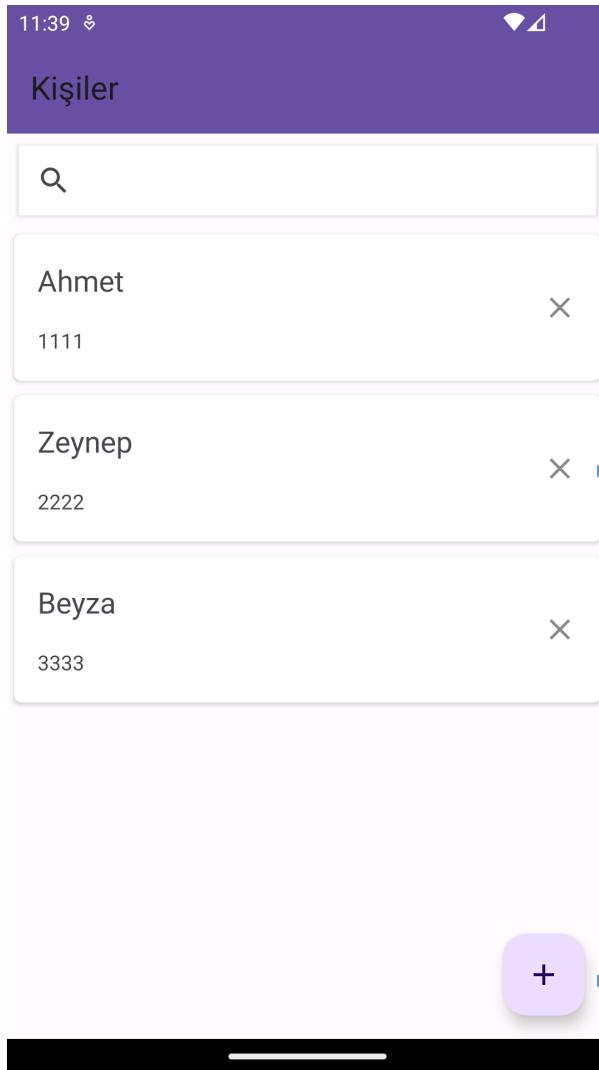
Resim 1

Resim 2

# Kişiler Uygulaması

# Retrofit

# Kişiler Uygulaması



# Kurulum

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.glidekullanimi">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        tools:ignore="FullBackupContent"
        tools:targetApi="28">

```

# Kurulum

```
dependencies { this: DependencyHandlerScope
```

build.gradle/app

```
implementation("androidx.core:core-ktx:1.9.0")
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.9.0")
implementation("androidx.constraintlayout:constraintlayout:2.1.4")
testImplementation("junit:junit:4.13.2")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```

```
}
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```



```

<?php
$response = array();

if (isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "INSERT INTO kisiler (kisi_ad,kisi_tel) VALUES ('$kisi_ad','$kisi_tel')";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>

```

[http://kasimadalan.pe.hu/kisiler/insert\\_kisiler.php](http://kasimadalan.pe.hu/kisiler/insert_kisiler.php)

```
<?php  
  
$response = array();  
  
if (isset($_POST['kisi_id']) && isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {  
    $kisi_id = $_POST['kisi_id'];  
    $kisi_ad = $_POST['kisi_ad'];  
    $kisi_tel = $_POST['kisi_tel'];  
  
    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.  
    require_once __DIR__ . '/db_config.php';  
  
    // Bağlantı oluşturuluyor.  
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);  
  
    // Bağlantı kontrolü yapılır.  
    if (!$baglanti) {  
        die("Hatalı bağlantı : " . mysqli_connect_error());  
    }  
  
    $sqlsorgu = "UPDATE kisiler SET kisiler.kisi_ad = '$kisi_ad',kisiler.kisi_tel = '$kisi_tel' WHERE  
                kisiler.kisi_id = $kisi_id ";  
  
    if (mysqli_query($baglanti, $sqlsorgu)) {  
        $response["success"] = 1;  
        $response["message"] = "successfully ";  
        echo json_encode($response);  
    } else {  
        $response["success"] = 0;  
        $response["message"] = "No product found";  
        echo json_encode($response);  
    }  
    //bağlantı koparılır.  
    mysqli_close($baglanti);  
} else {  
    $response["success"] = 0;  
    $response["message"] = "Required field(s) is missing";  
    echo json_encode($response);  
}  
?  
  
http://kasimadalan.pe.hu/kisiler/update\_kisiler.php
```

{ -

    "success":0,  
    "message":"Required field(s) is missing"

}

```

<?php
$response = array();

if (isset($_POST['kisi_id'])) {
    $kisi_id = $_POST['kisi_id'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }
    $sqlsorgu = "DELETE FROM kisiler WHERE kisiler.kisi_id = $kisi_id";

    if (mysqli_query($baglanti, $sqlsorgu)) {

        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {

        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>

```

[http://kasimadalan.pe.hu/kisiler/delete\\_kisiler.php](http://kasimadalan.pe.hu/kisiler/delete_kisiler.php)

{  }

```

    "success":0,
    "message":"Required field(s) is missing"
}
```

```

<?php
// array for JSON response
$response = array();

//DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
require_once __DIR__ . '/db_config.php';

// Bağlantı oluşturuluyor.
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

// Bağlantı kontrolü yapılır.
if (!$baglanti) {
    die("Hatalı bağlantı : " . mysqli_connect_error());
}

$sqlsorgu = "SELECT * FROM kisiler";
$result = mysqli_query($baglanti, $sqlsorgu);

// result kontrolü yap
if (mysqli_num_rows($result) > 0) {

    $response["kisiler"] = array();

    while ($row = mysqli_fetch_assoc($result)) {
        // temp user array
        $kisiler = array();

        $kisiler["kisi_id"] = $row["kisi_id"];
        $kisiler["kisi_ad"] = $row["kisi_ad"];
        $kisiler["kisi_tel"] = $row["kisi_tel"];

        // push single product into final response array
        array_push($response["kisiler"], $kisiler);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
}

```

[http://kasimadalan.pe.hu/kisiler/tum\\_kisiler.php](http://kasimadalan.pe.hu/kisiler/tum_kisiler.php)

```

{
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "kisi_id": "380",
            "kisi_ad": "Mehmet",
            "kisi_tel": "05375483215"
        },
        {
            "kisi_id": "381",
            "kisi_ad": "Ayşe",
            "kisi_tel": "05375483216"
        },
        {
            "kisi_id": "382",
            "kisi_ad": "Mustafa",
            "kisi_tel": "05375483217"
        }
    ],
    "success": 1
}

// no products found
$response["success"] = 0;
$response["message"] = "No data found";

// echo no users JSON
echo json_encode($response);
}

//bağlantı koparılır.
mysqli_close($baglanti);
?>

```

```

<?php

$response = array();

if (isset($_POST['kisi_ad'])) {
    $kisi_ad = $_POST['kisi_ad'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "SELECT * FROM kisiler WHERE kisiler.kisi_ad like '%$kisi_ad%'";
    $result = mysqli_query($baglanti, $sqlsorgu);

    if (mysqli_num_rows($result) > 0) {

        $response["kisiler"] = array();

        while ($row = mysqli_fetch_assoc($result)) {

            $kisiler = array();

            $kisiler["kisi_id"] = $row["kisi_id"];
            $kisiler["kisi_ad"] = $row["kisi_ad"];
            $kisiler["kisi_tel"] = $row["kisi_tel"];

            array_push($response["kisiler"], $kisiler);
        }

        $response["success"] = 1;
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
}

```

[http://kasimadalan.pe.hu/kisiler/tum\\_kisiler\\_arama.php](http://kasimadalan.pe.hu/kisiler/tum_kisiler_arama.php)

```

{
    "kisiler": [
        {
            "kisi_id": "376",
            "kisi_ad": "Ahmet",
            "kisi_tel": "5348564412"
        },
        {
            "kisi_id": "379",
            "kisi_ad": "Zeynep",
            "kisi_tel": "05375483214"
        },
        {
            "kisi_id": "380",
            "kisi_ad": "Mehmet",
            "kisi_tel": "05375483215"
        },
        {
            "kisi_id": "381",
            "kisi_ad": "Ayşe",
            "kisi_tel": "05375483216"
        },
        {
            "kisi_id": "382",
            "kisi_ad": "Mustafa",
            "kisi_tel": "05375483217"
        },
        {
            "kisi_id": "383",
            "kisi_ad": "Fatma",
            "kisi_tel": "05375483218"
        }
    ],
    "success": 1
}

```



```

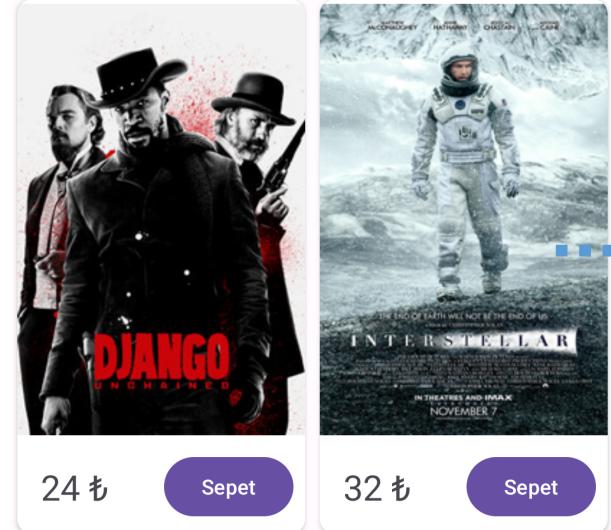
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>

```

# Filmler Uygulaması

## Retrofit

# Film Uygulaması



32 ₺

# Retrofit Kurulum

```
dependencies { this: DependencyHandlerScope
```

build.gradle/app

```
implementation("androidx.core:core-ktx:1.9.0")
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.9.0")
implementation("androidx.constraintlayout:constraintlayout:2.1.4")
testImplementation("junit:junit:4.13.2")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```

```
}
```

```
implementation("com.squareup.retrofit2:retrofit:2.6.0")
implementation("com.google.code.gson:gson:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.5.0")
```



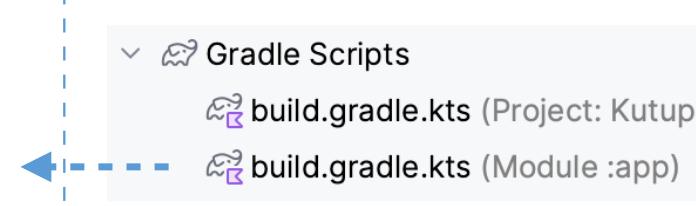
# Glide Kurulum

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    package="com.example.glidekullanimi">  
  
    <uses-permission android:name="android.permission.INTERNET"/>  
  
    <application  
        android:usesCleartextTraffic="true"  
        android:allowBackup="true"  
        android:dataExtractionRules="@xml/data_extraction_rules"
```

# Glide Kurulum

```
dependencies { this: DependencyHandlerScope  
    implementation("androidx.core:core-ktx:1.9.0")  
    implementation("androidx.appcompat:appcompat:1.6.1")  
    implementation("com.google.android.material:material:1.9.0")  
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")  
    testImplementation("junit:junit:4.13.2")  
    androidTestImplementation("androidx.test.ext:junit:1.1.5")  
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
  
    implementation("com.github.bumptech.glide:glide:4.13.2")  
}
```

build.gradle/app



Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

```

<?php
$response = array();
require_once __DIR__ . '/db_config.php';
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);
if (!$baglanti) {
    die("Hatalı bağlantı : " . mysqli_connect_error());
}

$sqlsorgu = "SELECT * FROM filmler_yeni";
$result = mysqli_query($baglanti, $sqlsorgu);

if (mysqli_num_rows($result) > 0) {
    $response["filmler"] = array();

    while ($row = mysqli_fetch_assoc($result)) {
        $filmler = array();
        $filmler["id"] = intval($row["id"]);
        $filmler["ad"] = $row["ad"];
        $filmler["resim"] = $row["resim"];
        $filmler["fiyat"] = intval($row["fiyat"]);
        array_push($response["filmler"], $filmler);
    }

    $response["success"] = 1;
    echo json_encode($response);
} else {
    $response["success"] = 0;
    $response["message"] = "No data found";
    echo json_encode($response);
}
mysqli_close($baglanti);
?>

```

[http://kasimadalan.pe.hu/filmler\\_yeni/tum\\_filmller.php](http://kasimadalan.pe.hu/filmler_yeni/tum_filmller.php)

```

"filmler": [
    {
        "id": 1,
        "ad": "Interstellar",
        "resim": "interstellar.png",
        "fiyat": 30
    },
    {
        "id": 2,
        "ad": "Inception",
        "resim": "inception.png",
        "fiyat": 20
    },
    {
        "id": 3,
        "ad": "The Dark Knight",
        "resim": "the-dark-knight.png",
        "fiyat": 25
    },
    {
        "id": 4,
        "ad": "The Shawshank Redemption",
        "resim": "the-shawshank-redemption.png",
        "fiyat": 15
    }
],
"success": 1
}

```

# Resimleri Görüntüleme

[http://kasimadalan.pe.hu/filmler\\_yeni/resimler/django.png](http://kasimadalan.pe.hu/filmler_yeni/resimler/django.png)

NOT : Glide ile resim oranını 500x750 yapabilirsin.



Base Url altında yer alan resimler

| Name ↑              |
|---------------------|
| anadoluda.png       |
| django.png          |
| inception.png       |
| interstellar.png    |
| thehatefuleight.png |
| thepianist.png      |

# Firebase Firestore

# Firestore



Gerçek zamanlı çalışabilen bir veritabanıdır.

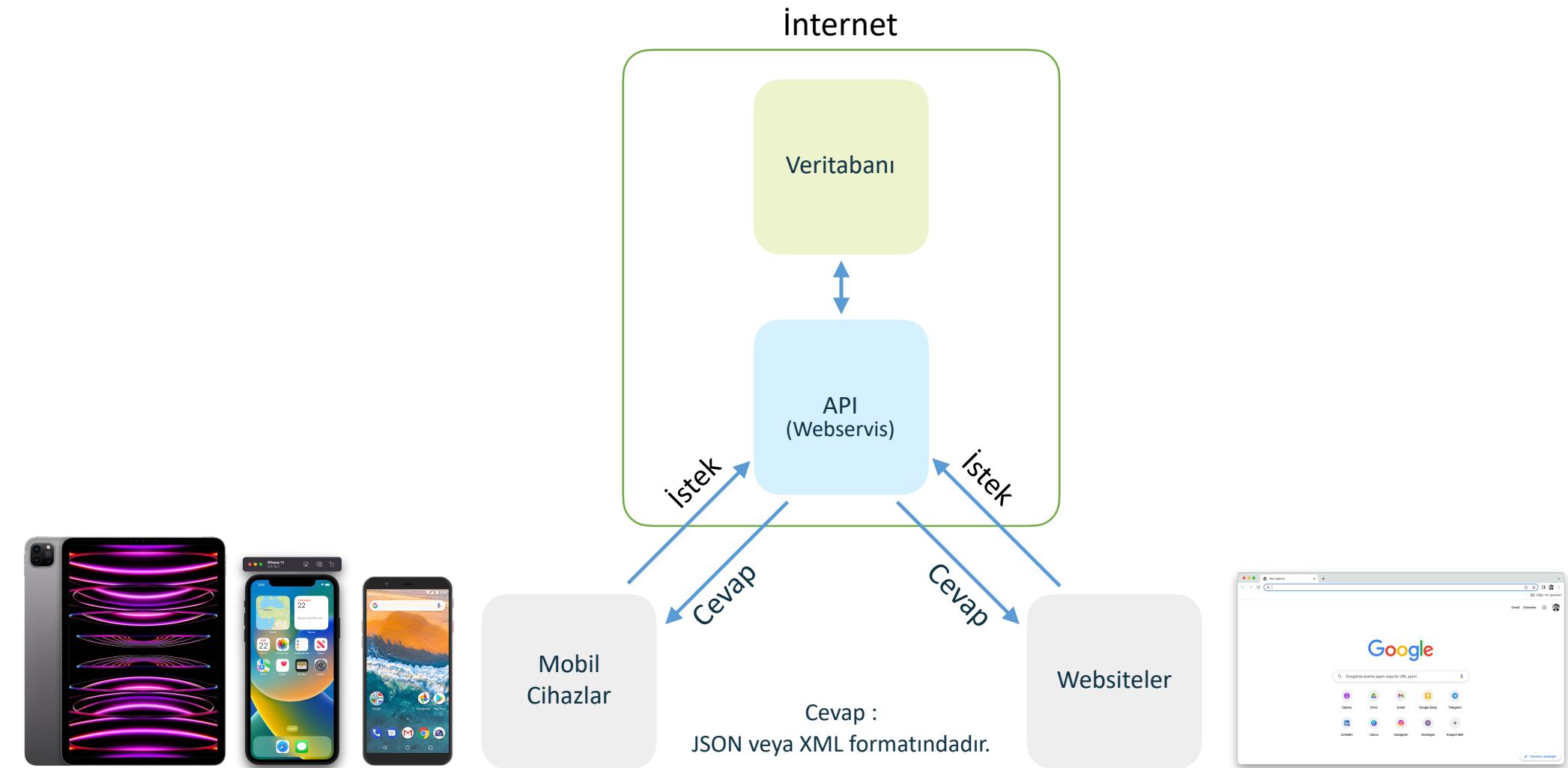
Veritabanında oluşan değişiklik anlık olarak uygulamaya aktarılabilir.

No SQL veritabanıdır , klasik sql sorguları çalışmamaktadır.

Webservis ( API ) kullanmanıza gerek yoktur.

Veriler json modeli ile kullanılmaktadır.

# RESTful Mimarisi



# Kurulum Aşamaları

1

Firebase Projesi Oluşturma

2

Firestore Veritabanı Oluşturma

3

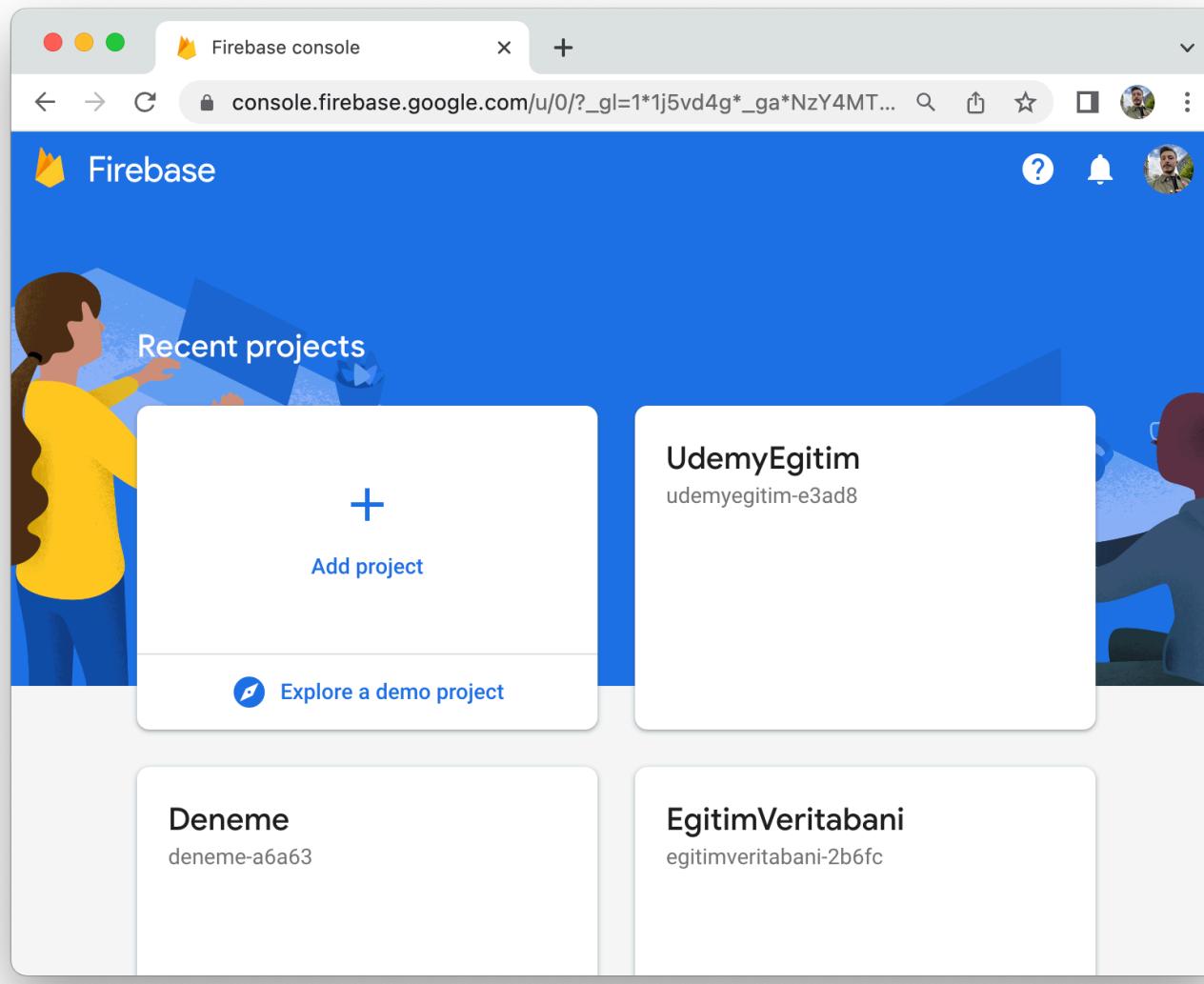
Android Studio Projesi Oluşturma

4

Firebase ve Android Studio Bağlantısı Oluşturma

1

# Firebase Projesi Oluşturma



2

# Firebase Veritabanı Oluşturma

## Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

[Create database](#)



Is Cloud Firestore right for you? [Compare Databases](#)

# Firebase Veritabanı Oluşturma

## Create database

1 Secure rules for Cloud Firestore —— 2 Set Cloud Firestore location

After you define your data structure, **you will need to write rules to secure your data.**  
[Learn more ↗](#)

Start in **production mode**  
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode**  
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';

service cloud.firestore {
    match /databases/{database}/documents {
        match /{document=**} {
            allow read, write: if false;
        }
    }
}
```

**i** All third party reads and writes will be denied

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project

Cancel **Next**

# Firebase Veritabanı Oluşturma

## Cloud Firestore

Data   Rules   Indexes   Usage    Extensions NEW



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

[Configure App Check](#)



[Panel view](#)

[Query builder](#)



More in Google Cloud ▾

 udemyegitim-e3ad8

 Start collection

# Veritabanı Kuralı Düzenleme

Cloud Firestore

Data    Rules **Rules**    Indexes    Usage    Extensions NEW

Edit rules    Monitor rules    Develop & Test

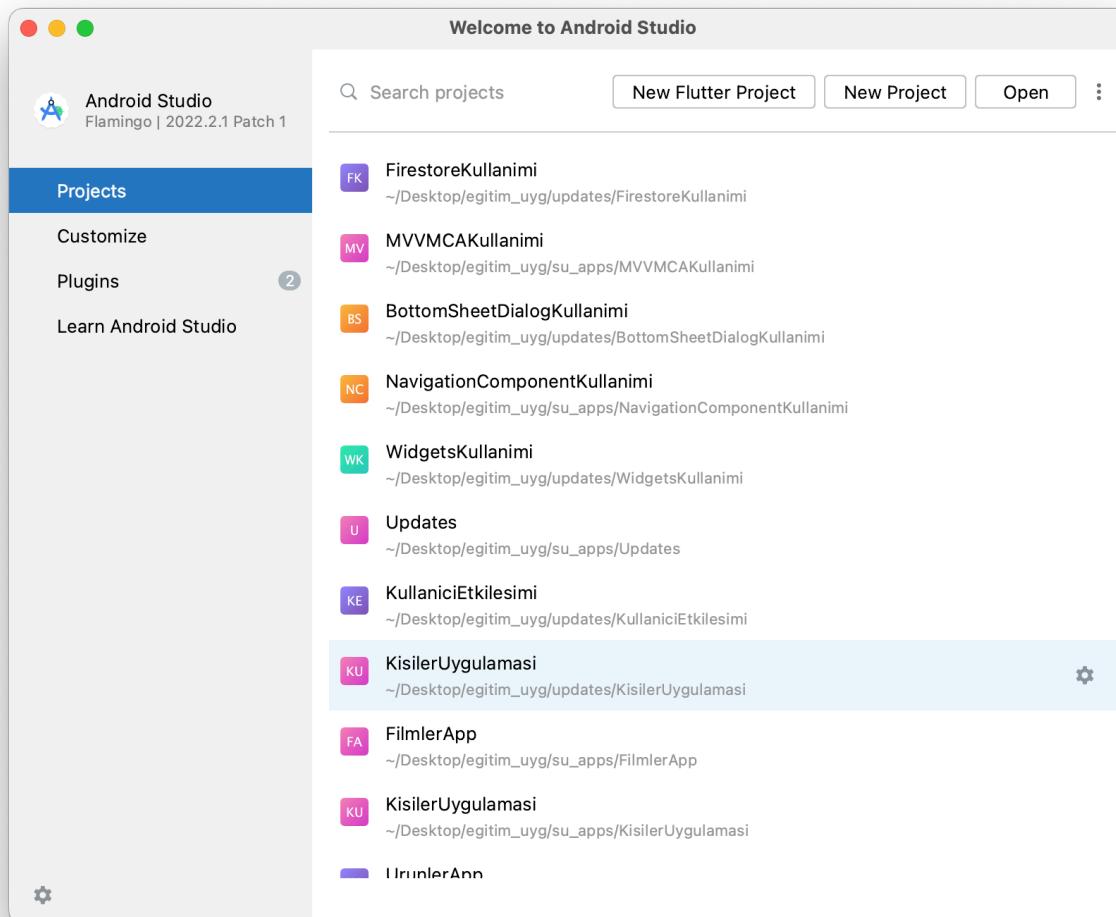
Today • 12:52 PM

Today • 12:51 PM

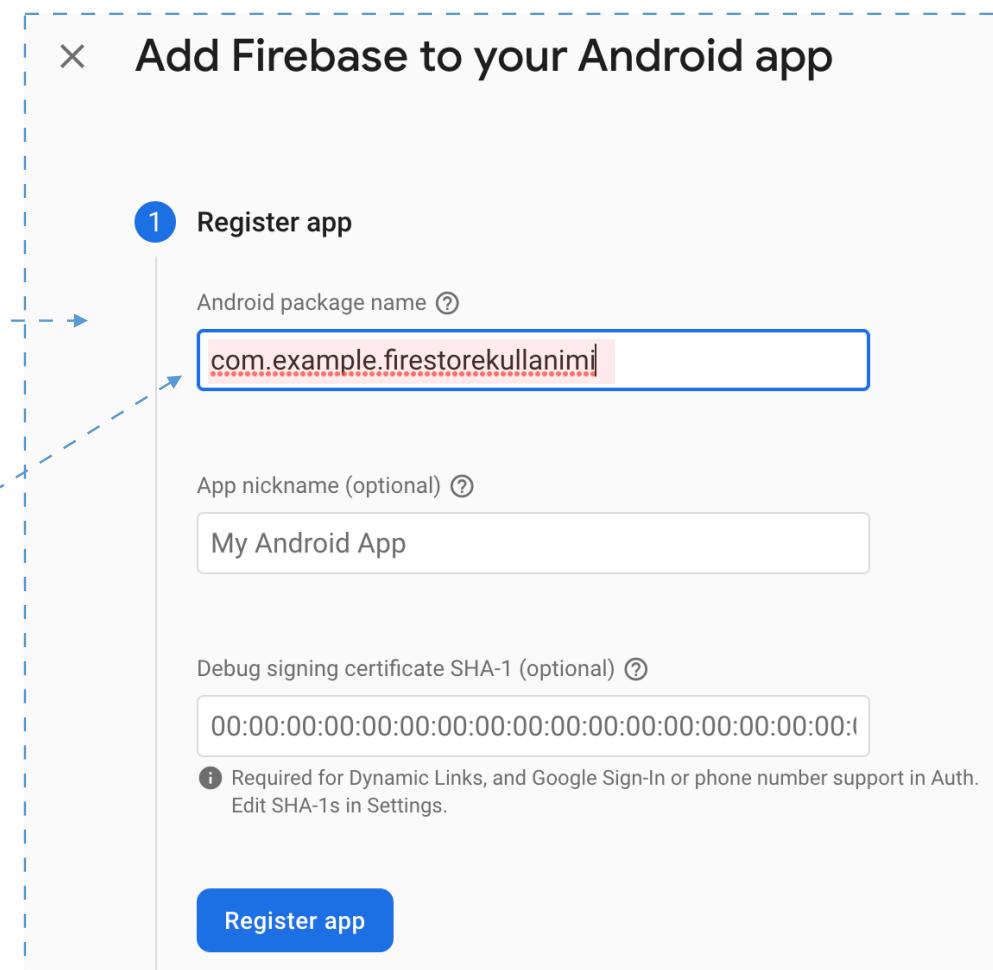
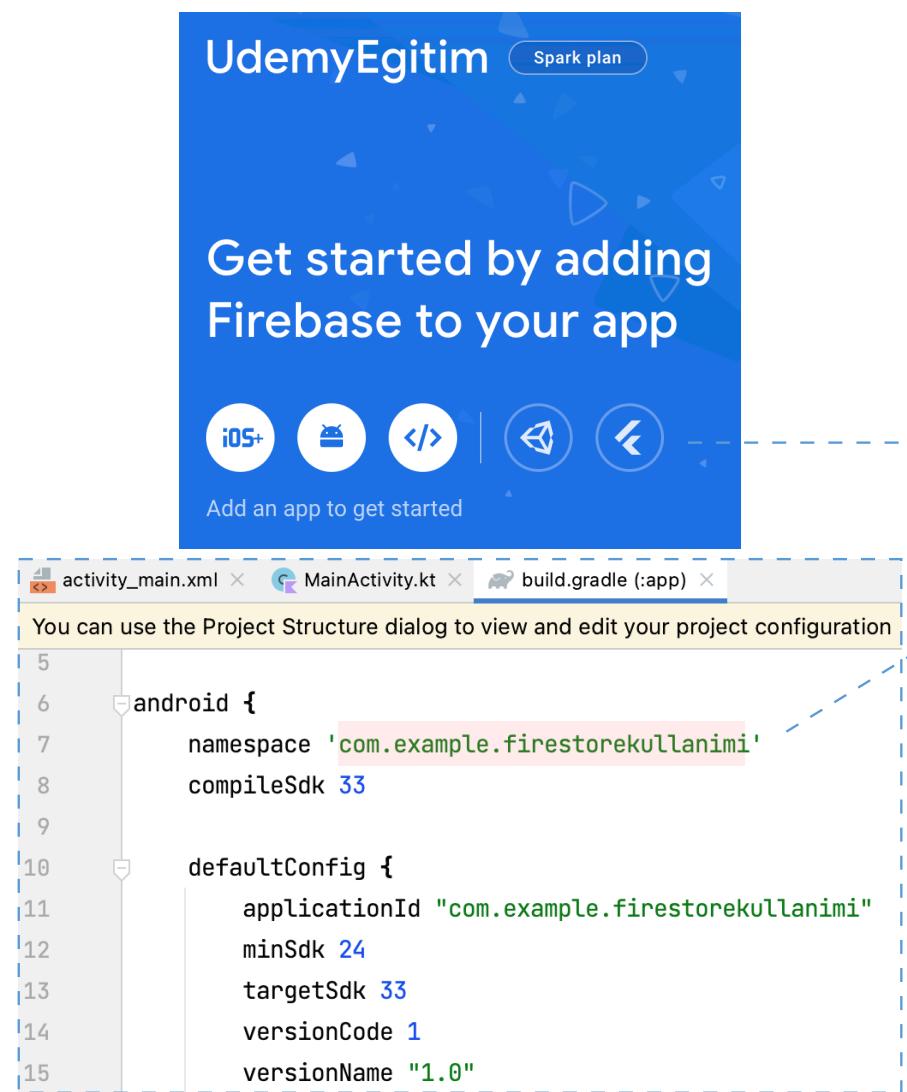
|   |   |
|---|---|
| 1 | rules_version = '2';                    |
| 2 |   |
| 3 | service cloud.firestore {               |
| 4 | match /databases/{database}/documents { |
| 5 | match /{document=**} {                  |
| 6 | allow read, write: if true;             |
| 7 | }                                       |
| 8 | }                                       |
| 9 | }                                       |

3

# Android Studio Projesi Oluşturma



# Firebase ve Android Studio Bağlantısı Oluşturma



# Config Dosyasını Projeye Ekleme

- Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

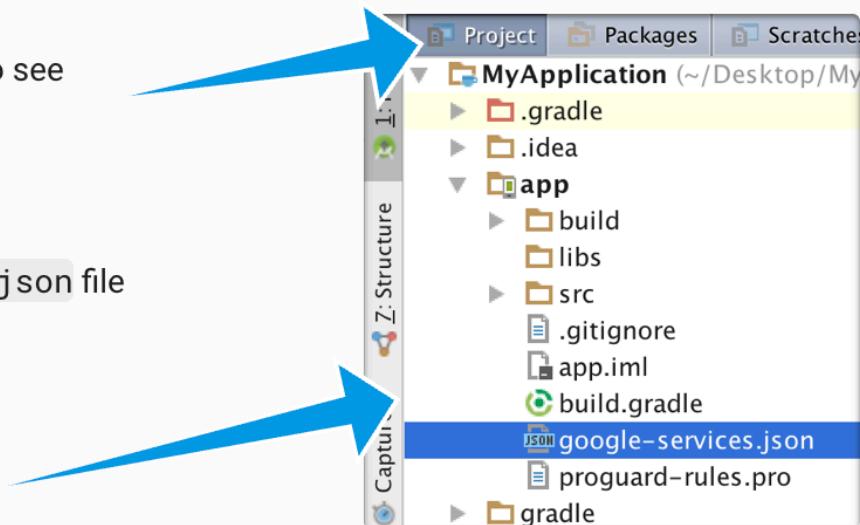
 [Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.



google-services.json



[Next](#)

# Standart Kütüphane Ekleme

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

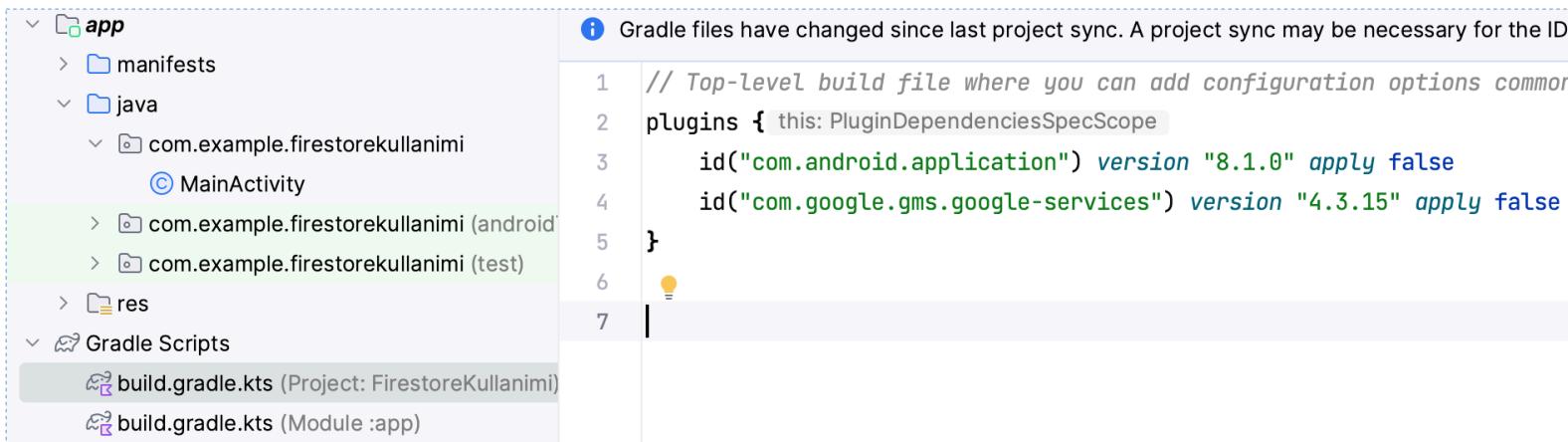
Kotlin DSL (`build.gradle.kts`)  Groovy (`build.gradle`)

Add the plugin as a dependency to your project-level `build.gradle.kts` file:

Root-level (project-level) Gradle file (`<project>/build.gradle.kts`):

```
plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.3.15" apply false
}
```



# Standart Kütüphane Ekleme

Module (app-level) Gradle file (<project>/<app-module>/build.gradle.kts):

```
plugins {
    id("com.android.application")
    // Add the Google services Gradle plugin
    id("com.google.gms.google-services")
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:32.2.2"))
    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

# Çalışmak istediğimi Kütüphaneyi Ekleme

<https://firebase.google.com/docs/android/setup#available-libraries>

|   |  |               |
|---|--|---------------|
| App Distribution                        | com.google.firebaseio:firebase-appdistribution         | 16.0.0-beta08 |
| App Distribution API                    | com.google.firebaseio:firebase-appdistribution-api-ktx | 16.0.0-beta08 |
| App Distribution plugin                 | com.google.firebaseio:firebase-appdistribution-gradle  | 4.0.0         |
| Authentication                          | com.google.firebaseio:firebase-auth-ktx                | 22.0.0        |
| Cloud Firestore                         | com.google.firebaseio:firebase-firebase-store-ktx      | 24.6.1        |
| Cloud Functions for Firebase Client SDK | com.google.firebaseio:firebase-functions-ktx           | 20.3.1        |

```
dependencies { this: DependencyHandlerScope

    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation(platform("com.google.firebaseio:firebase-bom:32.2.2"))

    implementation("com.google.firebaseio:firebase-firebase-store:24.7.0")
}
```

Gradle Scripts

- build.gradle (Project: FirestoreKullanimi)
- build.gradle (Module :app)
- proguard-rules.pro (ProGuard Rules for "com.google.firebase:firebase-firebase-store")
- gradle.properties (Project Properties)
- gradle-wrapper.properties (Gradle Version)
- local.properties (SDK Location)
- settings.gradle (Project Settings)

# Proje Kurulum Kontrol

## 4 Next steps

You're all set!

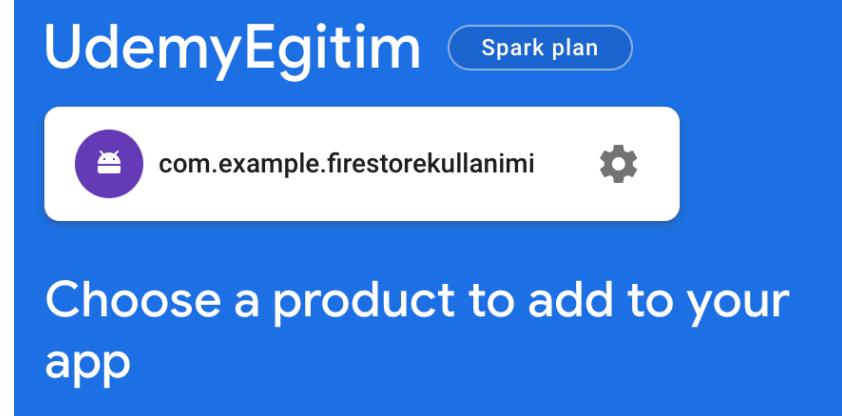
Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

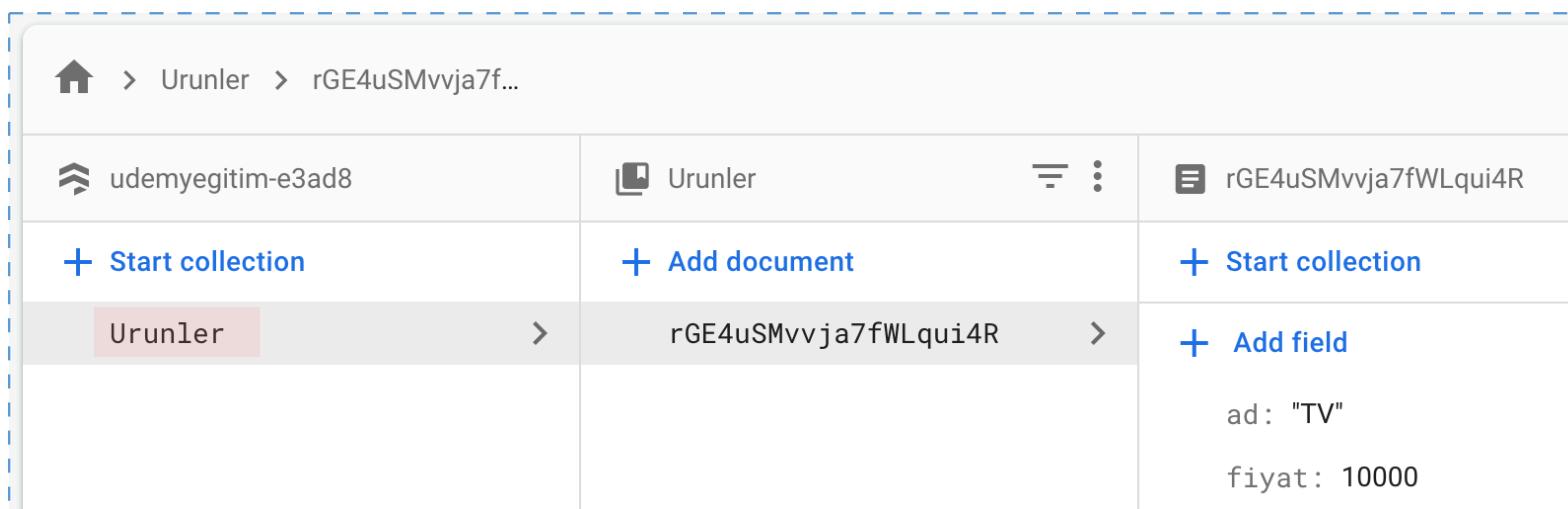
Previous

Continue to console



# İşlem Yapılacak Tablonun ( Collection ) Seçilmesi

```
class MainActivity : AppCompatActivity() {  
    private lateinit var collectionUrunler:CollectionReference  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val db = Firebase.firebaseio  
        collectionUrunler = db.collection(collectionPath: "Urunler")  
    }  
}
```

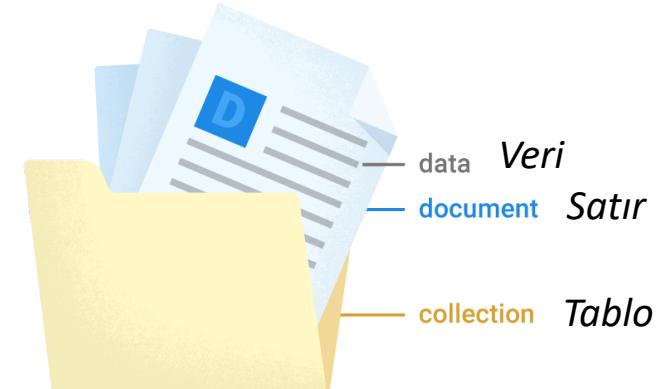


The screenshot shows the Firebase Realtime Database interface. At the top, there's a navigation bar with icons for home, collections, and a specific document. Below this, the main view displays a collection named 'Urunler'. A single document is shown under it, with its ID 'rGE4uSMvvja7fWLqui4R' visible. The document has two fields: 'ad' with the value 'TV' and 'fiyat' with the value '10000'.

| udemyegitim-e3ad8  | Urunler              | rGE4uSMvvja7fWLqui4R |
|--------------------|----------------------|----------------------|
| + Start collection | + Add document       | + Start collection   |
| Urunler            | rGE4uSMvvja7fWLqui4R | + Add field          |
|                    |                      | ad: "TV"             |
|                    |                      | fiyat: 10000         |

# Veri Yapısı

```
data class Urunler(var ad:String? = "",  
                    var fiyat:Int? = 0) {  
}
```



Panel view    Query builder

Home > Urunler > 5yHotBXmeqLc.. More in Google Cloud

| udemyegitim-e3ad8  | Urunler              | 5yHotBXmeqLcHEPBLxpe |
|--------------------|----------------------|----------------------|
| + Start collection | + Add document Satır | + Start collection   |
| Urunler            | 5yHotBXmeqLcHEPBLxpe | + Add field          |
|                    | DeSkdnMeg66zJcb4WKMz | ad: "Saat"           |
|                    | QGW5BiCj35M97Bwgq8oc | fiyat: 5000          |
|                    | nYt0ylgCYQfHuvoZ1Yge | Veri                 |
|                    | rGE4uSMvvja7fWLqui4R |                      |

Tablo    Veri

# Veri Yapısı

```
data class Urunler(var ad:String? = "",  
                    var fiyat:Int? = 0) {  
}
```

Panel view    Query builder    [View docs](#)

Run    Clear

Query scope    Path [?](#)

Collection    /Urunler    **Tablo**

[+ Add to query](#)



— data **Veri**  
— document **Satır**  
— collection **Table**

Query results

| Document ID          | ad       | fiyat |
|----------------------|----------|-------|
| 5yHotBXmeqLcHEPBLxpe | "Saat"   | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"  | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop" | 20000 |

**Satır**

# Klasik Veritabanı Görünümü

Panel view [Query builder](#) [View docs ↗](#)

[Run](#) [Clear](#)

Query scope Path ⓘ  
Collection /Urunler

[+ Add to query](#)

Query results ☰

| Document ID          | ad        | fiyat |
|----------------------|-----------|-------|
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |
| rGE4uSMvvja7fWLqui4R | "TV"      | 10000 |

Items per page: 50 ▾ 1 - 5 of 5 < >

# Insert - Veri Kaydı

```
fun kayit(){
    val yeniUrun = Urunler( ad: "TV", fiyat: 10000)
    collectionUrunler.document().set(yeniUrun)
}
```

| Query results        |           |       |
|----------------------|-----------|-------|
| Document ID          | ad        | fiyat |
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |
| rGE4uSMvvja7fWLqui4R | "TV"      | 10000 |

# Update - Veri Güncelleme

```
fun guncelleme(){
    val guncellenenUrun = HashMap<String, Any>()
    guncellenenUrun["ad"] = "Yeni TV"
    guncellenenUrun["fiyat"] = 9999
    collectionUrunler.document( documentPath: "rGE4uSMvvja7fWLqui4R" ).update(guncellenenUrun)
}
```

| Document ID          | ad        | fiyat |
|----------------------|-----------|-------|
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |
| rGE4uSMvvja7fWLqui4R | "Yeni TV" | 9999  |

# Delete - Veri Silme

```
fun silme(){
    collectionUrunler.document(documentPath: "rGE4uSMvvja7fWLqui4R").delete()
}
```

## Query results

| Document ID          | ad        | fiyat |
|----------------------|-----------|-------|
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |

# Query - Veri Okuma

## Klasik okuma işlemi

```
fun veriOkuma(){
    collectionUrunler.get().addOnCompleteListener { it: Task<QuerySnapshot!>
        for (d in it.result){
            val urun = d.toObject(Urunler::class.java)

            Log.e( tag: "*****", msg: "*****")
            Log.e( tag: "Urun key",d.id)
            Log.e( tag: "Urun ad",urun.ad!!)
            Log.e( tag: "Urun fiyat",urun.fiyat.toString())
        }
    }
}
```

| Query results        |           |       |
|----------------------|-----------|-------|
| Document ID          | ad        | fiyat |
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |

# Query - Veri Okuma

## Realtime ( Gerçek zamanlı ) okuma işlemi

```
fun veriOkumaRealtime(){
    collectionUrunler.addSnapshotListener { value, error ->
        if (value != null) {

            for (d in value.documents){
                val urun = d.toObject(Urunler::class.java)

                if(urun != null){
                    Log.e( tag: "*****", msg: "*****")
                    Log.e( tag: "Urun key",d.id)
                    Log.e( tag: "Urun ad",urun.ad!!)
                    Log.e( tag: "Urun fiyat",urun.fiyat.toString())
                }
            }
        }
    }
}
```

Query results

| Document ID          | ad        | fiyat |
|----------------------|-----------|-------|
| 5yHotBXmeqLcHEPBLxpe | "Saat"    | 5000  |
| DeSkdnMeg66zJcb4WKMz | "Kalem"   | 1000  |
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |

Sorgu

# QueryBuilder

Firestore konsolu üzerinden veriler üzerinde sorgu oluşturabilir.

The screenshot shows the Firestore QueryBuilder interface. At the top, there are 'Run' and 'Clear' buttons. Below them, 'Query scope' is set to 'Collection' with path '/Urunler'. The main area contains a 'Where' clause: 'fiyat > number 10000'. On the left, a sidebar lists query operators: Count, And, Order by, and Limit, each with a brief description. On the right, the 'Query results' section displays two documents: one for a 'Laptop' at 20000 and another for a 'Telefon' at 30000. Navigation controls like 'Items per page: 50', '1 - 2 of 2', and arrows are at the bottom right.

Run Clear

Query scope Path [?](#)

Collection /Urunler

Where fiyat > number 10000

+ Add to query

Count  
Return the number of documents that match all conditions

And  
Use specific operators to match conditions

Order by  
Sort results in ascending or descending order by property

Limit  
Return up to a specific number of results

Query results

| Document ID          | ad        | fiyat |
|----------------------|-----------|-------|
| QGW5BiCj35M97Bwgq8oc | "Laptop"  | 20000 |
| nYtOylgCYQfHuvoZ1Yge | "Telefon" | 30000 |

Items per page: 50 1 - 2 of 2

# Where

İstediğiniz alanda koşul oluşturarak veri okuyabilirsiniz.

| Koşul | Firestore Karşılığı         | Çalıştığı Tür      |
|-------|-----------------------------|--------------------|
| ==    | whereEqualTo()              | Sayısal - Metinsel |
| !=    | whereNotEqualTo()           | Sayısal - Metinsel |
| >     | whereGreaterThan()          | Sayısal            |
| >=    | whereGreaterThanOrEqualTo() | Sayısal            |
| <     | whereLessThan()             | Sayısal            |
| <=    | whereLessThanOrEqualTo()    | Sayısal            |

```
val sorgu = collectionUrunler.whereEqualTo(field: "ad", value: "Saat")
```

```
sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>
    for (d in it.result){
        val ürün = d.toObject(Urunler::class.java)
```

# Where

```
fun whereSorgu(){                                ad == "Saat"
    val sorgu = collectionUrunler.whereEqualTo( field: "ad", value: "Saat")

    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>
        for (d in it.result){
            val urun = d.toObject(Urunler::class.java)

            Log.e( tag: "*****", msg: "*****")
            Log.e( tag: "Urun key",d.id)
            Log.e( tag: "Urun ad",urun.ad!!)
            Log.e( tag: "Urun fiyat",urun.fiyat.toString())
        }
    }
}
```

# Where

```
fun whereSorgu(){  
    val sorgu = collectionUrunler.whereGreaterThanOrEqual(field: "fiyat", value: 10000)  
  
    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>  
        for (d in it.result){  
            val urun = d.toObject(Urunler::class.java)  
  
            Log.e( tag: "*****", msg: "*****")  
            Log.e( tag: "Urun key", d.id)  
            Log.e( tag: "Urun ad", urun.ad!!)  
            Log.e( tag: "Urun fiyat", urun.fiyat.toString())  
        }  
    }  
}
```

# Where - Birleşik Koşul

```
fun whereSorgu(){
    val sorgu = collectionUrunler
        .whereGreaterThanOrEqual("fiyat", 10000).whereLessThan("fiyat", 25000)

    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>
        for (d in it.result){
            val urun = d.toObject(Urunler::class.java)

            Log.e("*****", "*****")
            Log.e("Urun key", d.id)
            Log.e("Urun ad", urun.ad!!)
            Log.e("Urun fiyat", urun.fiyat.toString())
        }
    }
}
```

# OrderBy

Verileri sıralama koşuludur.

```
fun orderBySorgu(){
    val sorgu = collectionUrunler.orderBy( field: "fiyat")      ASCENDING :
                                                                küçükten büyüğe
                                                                sıralama
    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>
        for (d in it.result){
            val ürün = d.toObject(Urunler::class.java)

            Log.e( tag: "*****", msg: "*****")
            Log.e( tag: "Urun key", d.id)
            Log.e( tag: "Urun ad", ürün.ad!!)
            Log.e( tag: "Urun fiyat", ürün.fiyat.toString())
        }
    }
}
```

# OrderBy

Verileri sıralama koşuludur.

```
fun orderBySorgu(){
    val sorgu = collectionUrunler.orderBy( field: "fiyat",Query.Direction.DESCENDING) DESCENDING : büyükten küçüğe
    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!> sıralama
        for (d in it.result){
            val urun = d.toObject(Urunler::class.java)

            Log.e( tag: "*****", msg: "*****")
            Log.e( tag: "Urun key",d.id)
            Log.e( tag: "Urun ad",urun.ad!!)
            Log.e( tag: "Urun fiyat",urun.fiyat.toString())
        }
    }
}
```

# Limit

Sınırlı sayıda veri okuma koşulu

```
fun limitSorgu(){  
    val sorgu = collectionUrunler.limit(2)  
    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>  
        for (d in it.result){  
            val urun = d.toObject(Urunler::class.java)  
  
            Log.e( tag: "*****", msg: "*****")  
            Log.e( tag: "Urun key", d.id)  
            Log.e( tag: "Urun ad", urun.ad!!)  
            Log.e( tag: "Urun fiyat", urun.fiyat.toString())  
        }  
    }  
}
```

*Sıralamaya göre  
ilk 2  
veriyi getirir.*

# Limit

Sınırlı sayıda veri okuma koşulu

```
fun limitSorgu(){  
    val sorgu = collectionUrunler.limitToLast( limit: 2)  
    sorgu.get().addOnCompleteListener { it: Task<QuerySnapshot!>  
        for (d in it.result){  
            val urun = d.toObject(Urunler::class.java)  
            Log.e( tag: "*****", msg: "*****")  
            Log.e( tag: "Urun key", d.id)  
            Log.e( tag: "Urun ad", urun.ad!!)  
            Log.e( tag: "Urun fiyat", urun.fiyat.toString())  
        }  
    }  
}
```

*Sıralamaya göre  
son 2  
veriyi getirir.*

# Count

Sorgu sonucunda kaç adet kayıt olduğu bilgisini alabiliriz.

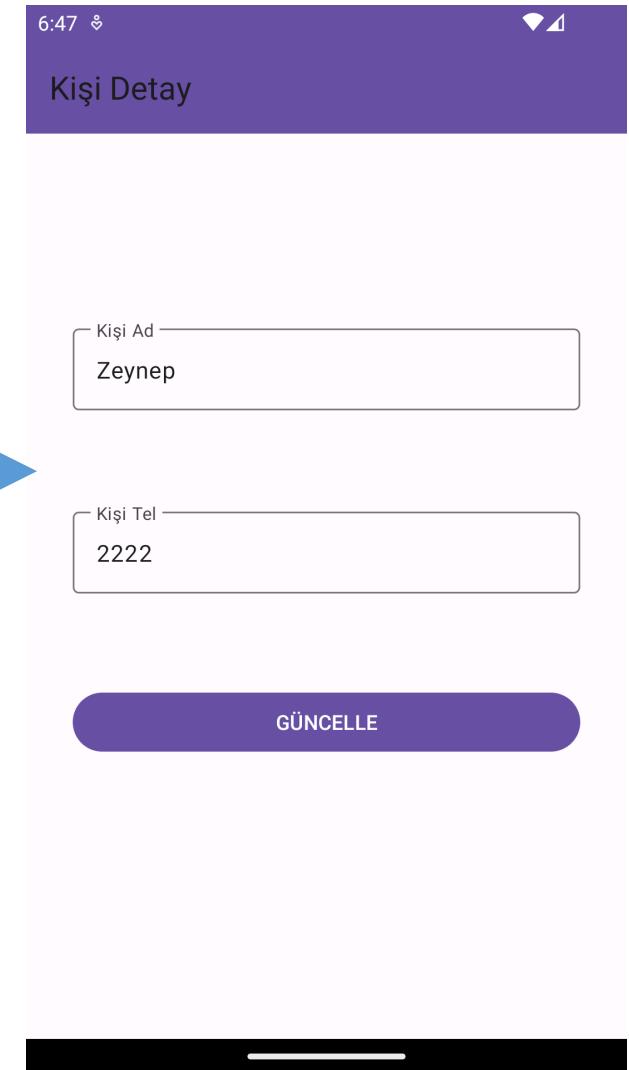
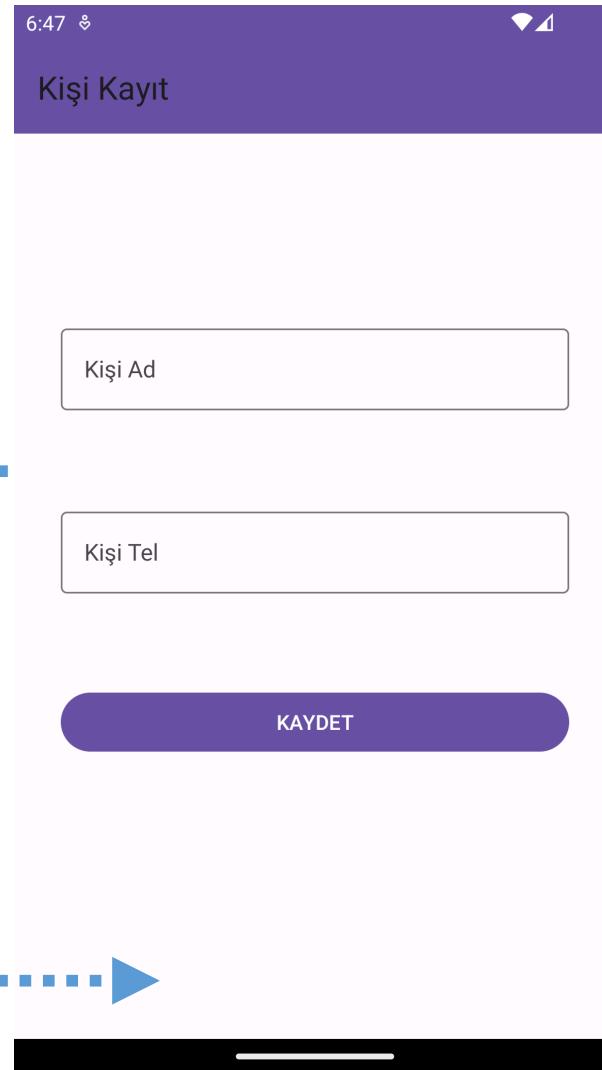
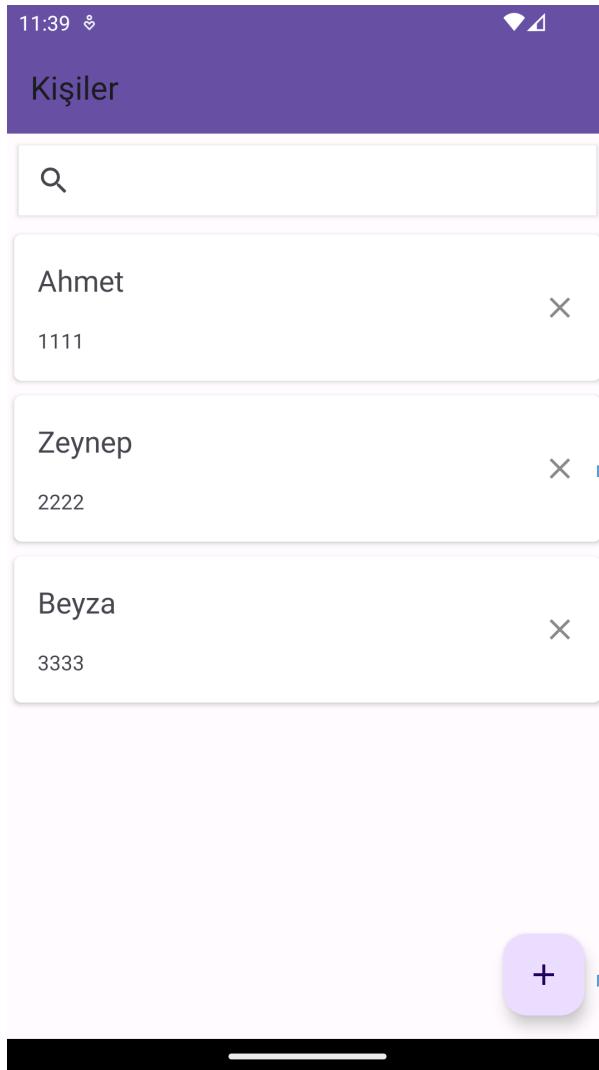
```
fun countSorgu(){
    val sorgu = collectionUrunler.count()

    sorgu.get(AggregateSource.SERVER).addOnCompleteListener { it: Task<AggregateQuerySnapshot!>
        val sonuc = it.result
        Log.e( tag: "Sonuç", sonuc.count.toString())
    }
}
```

# Kişiler Uygulaması

## Firestore

# Kişiler Uygulaması



## Query results



| Document ID          | kisi_ad | kisi_id | kisi_tel |
|----------------------|---------|---------|----------|
| DoDf7rgv25n51NuC3UQn | "Ali"   | ""      | "4444"   |
| ZQx0AkM4LqUpieZlpZi8 | "Ece"   | ""      | "6666"   |

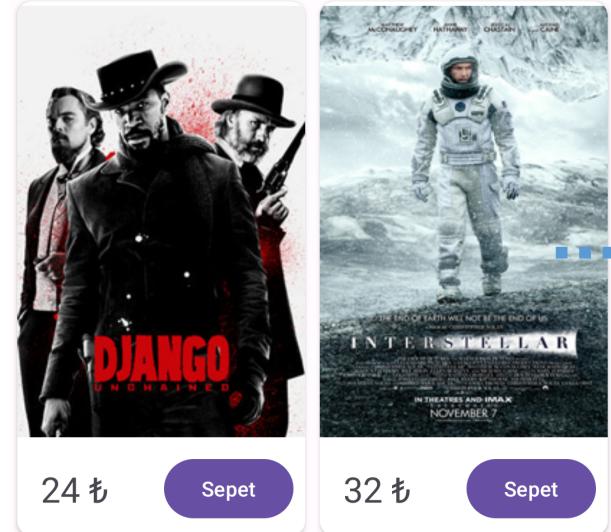
Items per page: 50 ▾ 1 - 2 of 2 < >

```
data class Kisiler(var kisi_id:String? = "",  
                   var kisi_ad:String? = "",  
                   var kisi_tel:String? = "") : Serializable
```

# Filmler Uygulaması

## Firestore

# Film Uygulaması



32 ₺

## Query results



| Document ID          | ad                  | fiyat | id | resim                 |
|----------------------|---------------------|-------|----|-----------------------|
| A4URtX9rNE0m0acbJDWe | "Inception"         | 16    | "" | "inception.png"       |
| EszaHVdHibjRmbd0mHoB | "The Hateful Eight" | 28    | "" | "thehatefuleight.png" |
| TZK76YSboenW9IQmhGMJ | "The Pianist"       | 18    | "" | "thepianist.png"      |
| YCX2m9CHgHSnEwurGnLg | "Anadoluda"         | 10    | "" | "anadoluda.png"       |
| dkzqxG0aPVEnRlfNbrJJ | "Interstellar"      | 32    | "" | "interstellar.png"    |
| mhRv7UMqLHgHEELTRNUX | "Django"            | 24    | "" | "django.png"          |

```
data class Filmler(var id:String? = "",  
                   var ad:String? = "",  
                   var resim:String? = "",  
                   var fiyat:Int? = 0) : Serializable
```

# Data Oluşturma

- Firestore import veya export ücretli olduğu için pratik şekilde verilerimizi kayıt yaparak oluşturuyoruz.

```
val db = Firebase.firestore
val collectionFilmler = db.collection("Filmler")

val f1 = Filmler("", "Django", "django.png", 24)
val f2 = Filmler("", "Interstellar", "interstellar.png", 32)
val f3 = Filmler("", "Inception", "inception.png", 16)
val f4 = Filmler("", "The Hateful Eight", "thehatefuleight.png", 28)
val f5 = Filmler("", "The Pianist", "thepianist.png", 18)
val f6 = Filmler("", "Anadoluda", "anadoluda.png", 10)

collectionFilmler.document().set(f1)
collectionFilmler.document().set(f2)
collectionFilmler.document().set(f3)
collectionFilmler.document().set(f4)
collectionFilmler.document().set(f5)
collectionFilmler.document().set(f6)
```

# Glide Kurulum

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.glidekullanimi">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        tools:ignore="FullBackupContent,AllowBackup,UsesCleartextTraffic">

```

# Glide Kurulum

```
dependencies { this: DependencyHandlerScope  
  
    implementation("androidx.core:core-ktx:1.9.0")  
    implementation("androidx.appcompat:appcompat:1.6.1")  
    implementation("com.google.android.material:material:1.9.0")  
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")  
    testImplementation("junit:junit:4.13.2")  
    androidTestImplementation("androidx.test.ext:junit:1.1.5")  
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
  
    implementation("com.github.bumptech.glide:glide:4.13.2")  
}  
  
implementation("com.github.bumptech.glide:glide:4.13.2")
```

build.gradle/app

Gradle Scripts  
build.gradle.kts (Project: Kutup)  
build.gradle.kts (Module :app)

# Resimleri Görüntüleme

[http://kasimadalan.pe.hu/filmler\\_yeni/resimler/django.png](http://kasimadalan.pe.hu/filmler_yeni/resimler/django.png)

NOT : Glide ile resim oranını 500x750 yapabilirsin.



Base Url altında yer alan resimler

| Name ↑              |
|---------------------|
| anadoluda.png       |
| django.png          |
| inception.png       |
| interstellar.png    |
| thehatefuleight.png |
| thepianist.png      |

# ileri Android

# Work Manager

# Work Manager

- Arkaplanda işlem yapmak için kullandığımız bir yapıdır.
- İstediğiniz işlemleri arkaplanda yaptırabiliriz.
  - Örneğin : Bildirim oluşturmak , resim yüklemek , sayısal işlem yaptırmak vb.
- Bu işlemlere geçikme verip aynı zamanda periyodik olarak yaptırabiliriz.
- Zamana bağlı olarak durumları tetikleyebiliriz.
- Telefon kapansada ekstra bir işlem yapmadan çalışmaya devam edecektir.
- En güncel yapıdır ve Android Jetpack yapısı ile birlikte gelmektedir.
- Alarm Manager ve Job Scheduler altyapılarının üzerine kurulmuş bir yapıdır.
- Alarm Manager sadece eski android sürümlerini desteklemektedir.
- Work Manager bütün sürümler ile uyumludur.

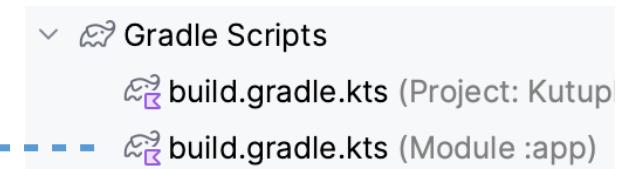
# Kurulum

```
dependencies { this: DependencyHandlerScope           build.gradle/app

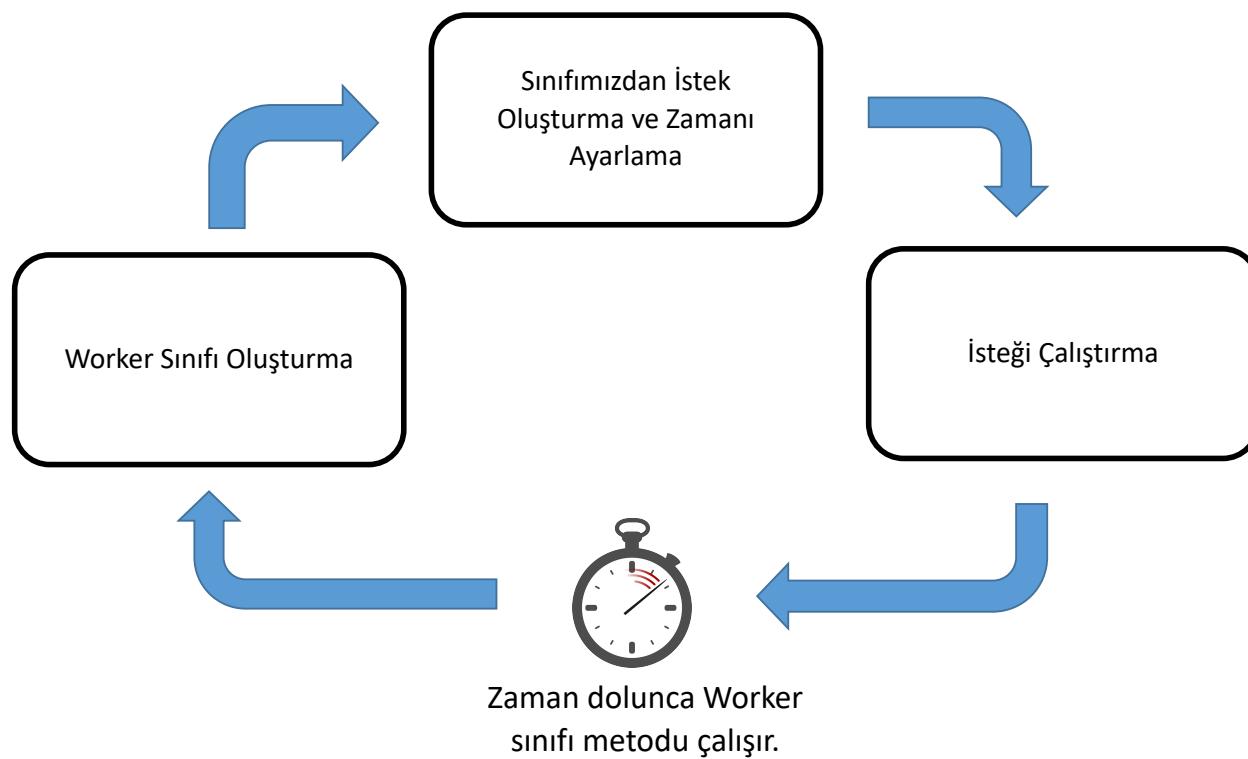
    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation("androidx.work:work-runtime-ktx:2.7.1")
}

implementation("androidx.work:work-runtime-ktx:2.7.1")
```



# Çalışma Yapısı



# KODLAMA YAPISI

# Worker Sınıfı Oluşturma

- Zamansal olarak işlem yapmak için bir sınıf oluşturulur.
- Bu sınıf *Worker* sınıfından miras yolu ile extend edilir.

```
class MyWorker(appContext: Context, workerParams: WorkerParameters): Worker(appContext, workerParams) {  
  
    override fun doWork(): Result {  
        val toplam = 10 + 20  
        Log.e( tag: "Arkaplan İşlemi Sonucu", toplam.toString())  
  
        return Result.success()  
    }  
}
```

Arkaplarda çalıştırılacak kodların yazılacağı metod.

Dönüş değeri duruma göre değişebilir.

Result.success() İşlemen başarılı şekilde bittiğini belirtir.

Result.failure() İşlem olurken hata olduğunu belirtir.

# İstek Oluşturma

- Oluşturduğumuz sınıfıktaki metodu çalıştırmak için istek oluşturulur ve zamana duyarlı bir şekilde çalışması beklenir.
- İki farklı istek oluşturabiliriz.
  - *OneTimeWorkRequest* : Bir kere çalışacak istek.

```
val istek = OneTimeWorkRequestBuilder<MyWorker>().build()
```

- *PeriodicWorkRequest* : Tekrarlı işlem yapmak için istek.

```
val istek = PeriodicWorkRequestBuilder<MyWorkerBildirim>(  
    repeatInterval: 15, TimeUnit.MILLISECONDS)  
    .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
    .build()
```

## İsteği Çalıştırma

```
WorkManager.getInstance( context: this@MainActivity).enqueue(istek)
```

# İstek Türleri

# OneTimeWorkRequest

- Bir kere çalışmak için ayarlanan istektir.
  - Anlık veya gecikmeli olarak arka planda işlem yaptırabilirsiniz.

• *Anlık Çalıştırma ;*

```
val istek = OneTimeWorkRequestBuilder<MyWorker>().build()
```

#### **Gecikmeli Çalıştırma ;**

```
val istek = OneTimeWorkRequestBuilder<MyWorker>()
    .setInitialDelay(duration: 10, TimeUnit.SECONDS)
    .build()
```

# Gecikme Süre miktarı

# Örnek : OneTimeWorkRequest

Butona basıldığı anda anlık veya gecikmeli işlem yapırma

```
class MyWorker(appContext: Context, workerParams: WorkerParameters): Worker(appContext, workerParams) {  
  
    override fun doWork(): Result {  
        val toplam = 10 + 20  
        Log.e( tag: "Arkaplan İşlemi Sonucu", toplam.toString())  
  
        return Result.success()  
    }  
}
```

# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

1

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(LayoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3

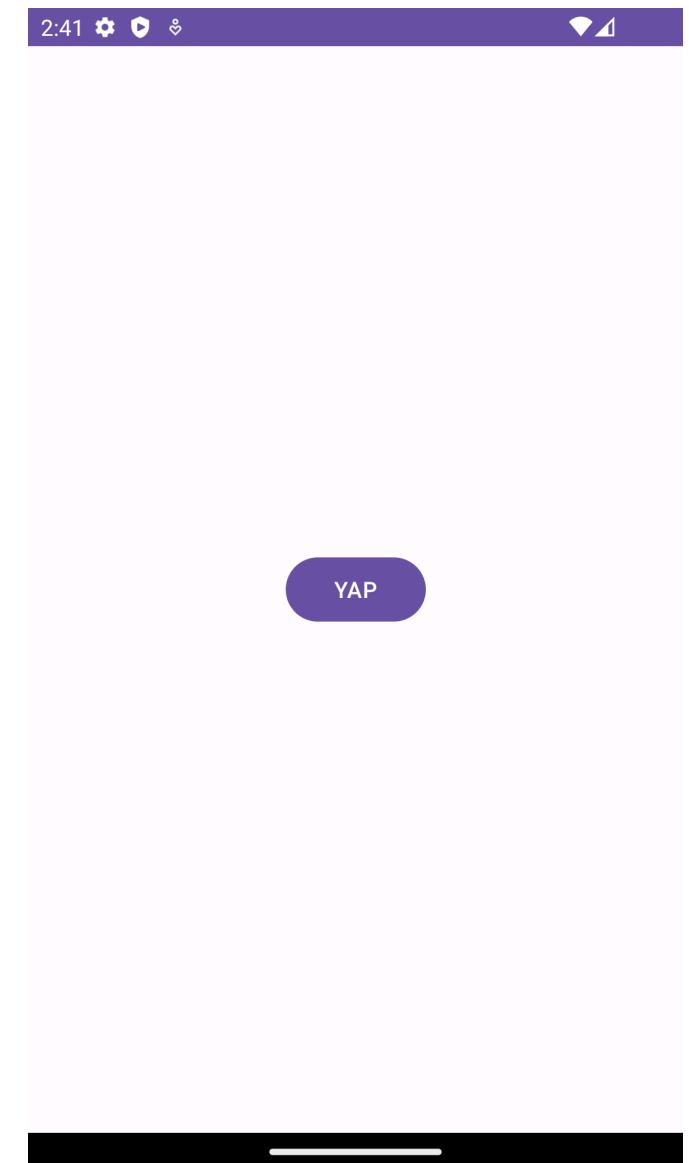
```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonYap.setOnClickListener { it: View!  
            val istek = OneTimeWorkRequestBuilder<MyWorker>().build()  
  
            WorkManager.getInstance(context: this@MainActivity).enqueue(istek)  
        }  
    }  
}
```

YAP

Butona basıldığı anda arka planda işlem yapar ve sonucu konsoldan takip edebiliriz.

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonYap.setOnClickListener { it: View!  
            val istek = OneTimeWorkRequestBuilder<MyWorker>()  
                .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
                .build()  
  
            WorkManager.getInstance( context: this@MainActivity).enqueue(istek)  
        }  
    }  
}
```

Butona basıldığı anda 10 saniye gecikmeli olarak arka planda işlem yapar ve sonucu konsoldan takip edebiliriz.



# Yapılan İşi Takip Etme

- Arkaplan işinin çalışma durumu bilgisi alınabilir.

```
WorkManager.getInstance( context: this ).getWorkInfoByIdLiveData(istek.id)
    .observe( owner: this ) { it: WorkInfo!
        val durum = it.state.name
        Log.e( tag: "Arkaplan İşlem Durumu", durum )
    }
```

E/İşlem Durumu: ENQUEUED  
E/Arkaplan İşlemi Sonucu: 30  
E/İşlem Durumu: SUCCEEDED

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonYap.setOnClickListener { it: View!  
            val istek = OneTimeWorkRequestBuilder<MyWorker>()  
                .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
                .build()  
  
            WorkManager.getInstance( context: this@MainActivity).enqueue(istek)  
  
            WorkManager.getInstance( context: this).getWorkInfoByIdLiveData(istek.id)  
                .observe( owner: this) { it: WorkInfo!  
                    val durum = it.state.name  
                    Log.e( tag: "Arkaplan İşlem Durumu", durum)  
                }  
        }  
    }  
}
```

2:41 🔍 🌐

YAP

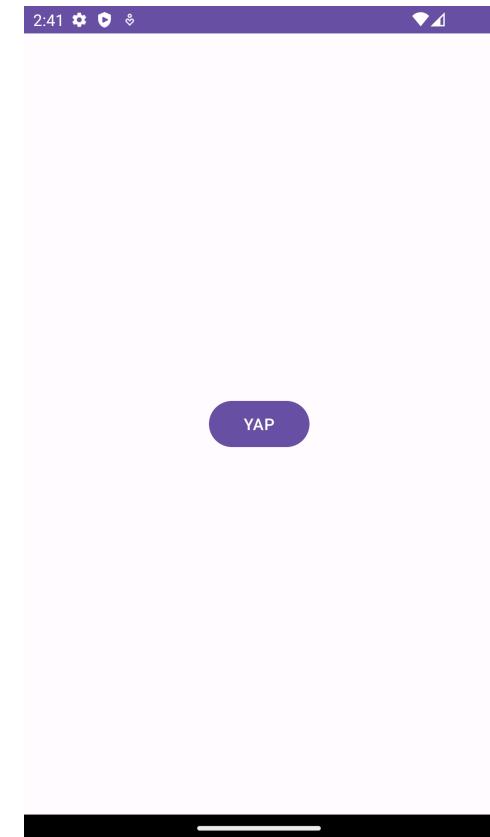
# Arkaplan işlerini koşullara göre çalıştırma

- Constraint oluşturarak çalışma şartı oluşturabiliriz
- Örneğin sadece telefon internete ( wifi ve mobil ) bağlısa çalış diyebiliriz.

```
val calismaKosulu = Constraints.Builder()  
    .setRequiredNetworkType(NetworkType.CONNECTED)  
    .build()
```

```
val istek = OneTimeWorkRequestBuilder<MyWorker>()  
    .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
    .setConstraints(calismaKosulu)  
    .build()
```

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonYap.setOnClickListener { it: View!  
            val calismaKosulu = Constraints.Builder().  
                setRequiredNetworkType(NetworkType.CONNECTED).build()  
  
            val istek = OneTimeWorkRequestBuilder<MyWorker>()  
                .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
                .setConstraints(calismaKosulu)  
                .build()  
  
            WorkManager.getInstance( context: this@MainActivity).enqueue(istek)  
            WorkManager.getInstance(applicationContext).getWorkInfoByIdLiveData(istek.id)  
                .observe( owner: this@MainActivity, Observer { workInfo: WorkInfo ->  
                    val durum = workInfo.state.name  
                    Log.e( tag: "İşlem Durumu",durum)  
                })  
        }  
    }  
}
```



YAP

Test İşlemi :

1. Uygulama çalıştırılır.
2. Internet ( Wifi ve Mobil ) bağlantısı kesilir.
3. Buttona basılarak işlem yapmaya çalışır.
4. Internet olmadığı için çalışma olmayacağıdır.
5. Internet açılır.
6. Internet açılır açılmaz çalışmasını görebiliriz.

# PeriodicWorkRequest

- İstenirse belirli bir süre içinde arka planda tekrarlı işlem yaptırılabilir.
- Tekrar süresi minimum 15 dk olmalıdır.

```
val istek = PeriodicWorkRequestBuilder<MyWorkerBildirim>(repeatInterval: 15, TimeUnit.MINUTES)  
    .setInitialDelay(duration: 10, TimeUnit.SECONDS)  
    .build()
```

İlk çalışma için geçikme miktarı.  
10 sn sonra ilk çalışma olacak  
ve  
15 dk arayla çalışacak

Tekrarlama Süre miktarı  
Sürenin türü saniye, dakika vb.

TimeUnit ile birçok türde zaman birimine göre tekrar aralığı belirleyebilirsiniz

TimeUnit.MINUTES   TimeUnit.HOURS   TimeUnit.DAYS

# Örnek : PeriodicWorkRequest

```
class MyWorkerBildirim(appContext: Context, workerParams: WorkerParameters): Worker(appContext, workerParams) {  
  
    override fun doWork(): Result {  
        bildirimOlustur()  
        return Result.success()  
    }  
  
    fun bildirimOlustur(){  
        val builder:NotificationCompat.Builder  
        val bildirimYonetici = applicationContext  
            .getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager  
        val intent = Intent(applicationContext,MainActivity::class.java)  
        val gidilecekIntent = PendingIntent.getActivity(applicationContext,  
            requestCode: 1,intent, flags: PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE)  
    }  
}
```

*Bildirim için gerekli izin*

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

*Ayarlardan uygulama bildirimine izin verilmelidir.*

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){

    val kanalId = "kanalId"
    val kanalAd = "kanalAd"
    val kanalTanitim = "kanalTanitim"
    val kanalOnceligi = NotificationManager.IMPORTANCE_HIGH

    var kanal : NotificationChannel? = bildirimYonetici.getNotificationChannel(kanalId)

    if(kanal == null){
        kanal = NotificationChannel(kanalId,kanalAd,kanalOnceligi)
        kanal.description = kanalTanitim
        bildirimYonetici.createNotificationChannel(kanal)
    }

    builder = NotificationCompat.Builder(applicationContext,kanalId)

    builder.setContentTitle("Başlık")
        .setContentText("İçerik")
        .setSmallIcon(R.drawable.resim)
        .setContentIntent(gidilecekIntent)
        .setAutoCancel(true)
    }

    else{
        builder = NotificationCompat.Builder(applicationContext)

        builder.setContentTitle("Başlık")
            .setContentText("İçerik")
            .setSmallIcon(R.drawable.resim)
            .setContentIntent(gidilecekIntent)
            .setAutoCancel(true)
            .priority = Notification.PRIORITY_HIGH
    }

    bildirimYonetici.notify( id: 1,builder.build())
}

}
```

```
class MainActivity : AppCompatActivity() {  
    private lateinit var tasarim:ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonYap.setOnClickListener { it: View!  
            val istek = PeriodicWorkRequestBuilder<MyWorkerBildirim>(  
                repeatInterval: 15, TimeUnit.MILLISECONDS)  
                .setInitialDelay( duration: 10, TimeUnit.SECONDS)  
                .build()  
  
            WorkManager.getInstance( context: this@MainActivity).enqueue(istek)  
        }  
    }  
}
```

*ilk çalışma için geçikme miktarı.  
10 sn sonra ilk çalışma olacak  
ve  
15 dk arayla çalışacak*

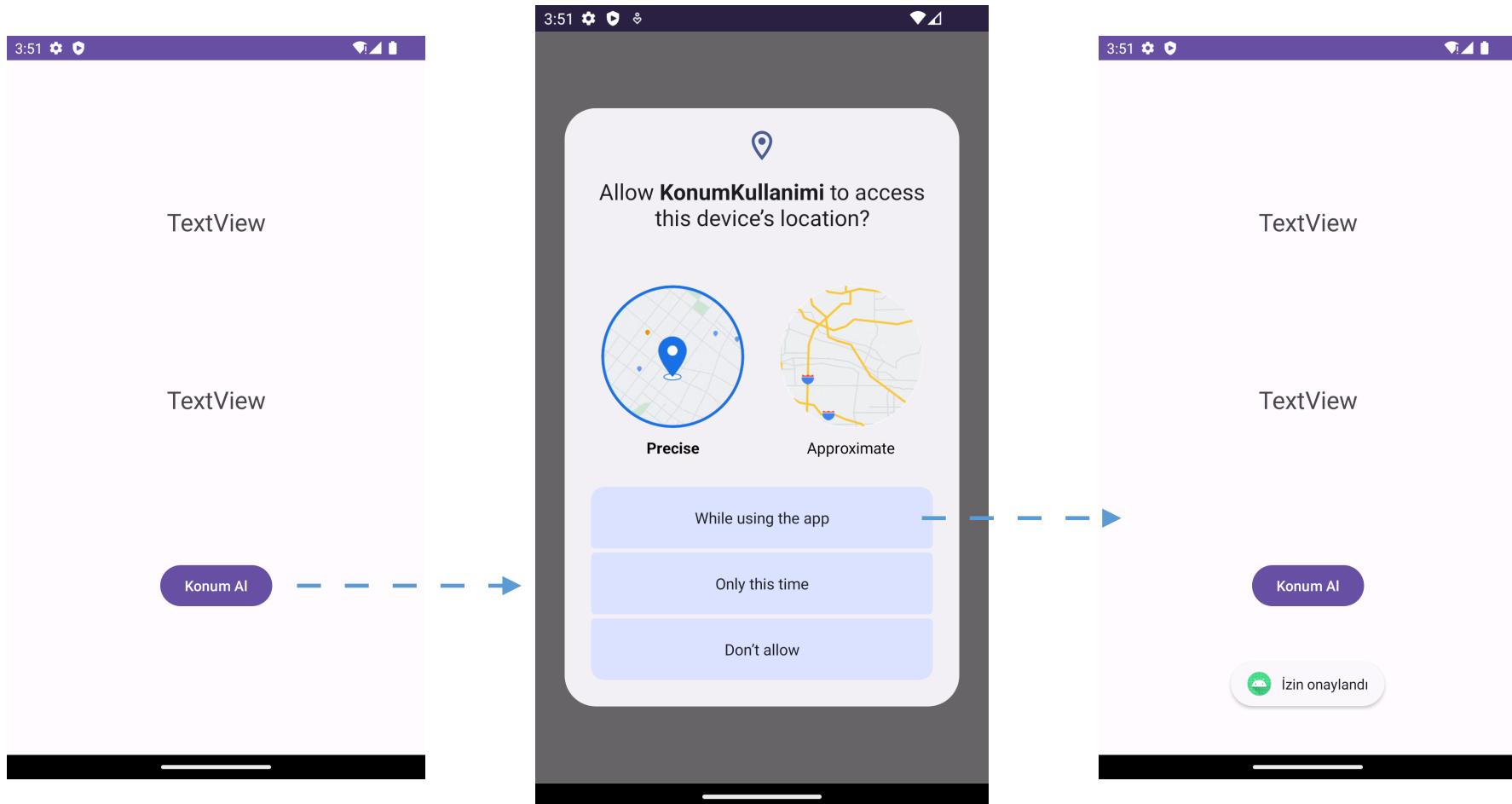
YAP

# Lokasyon İşlemleri

# Konum Kullanımı

# İzin Kontrol

- Uygulama izinlerini kullanıcıya onaylatmak için kolaylık sağlayan bir yapıdır.



# Konum için Kurulum

- İzin kontrolü için manifest dosyasında en az bir izin tanımlaması yeterlidir ve bütün izinler için aynı süreç yürütülebilir.

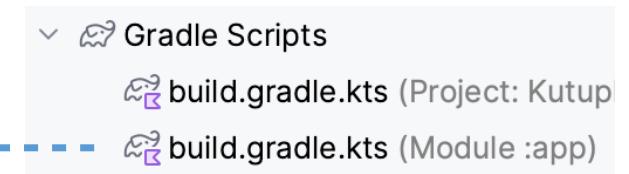
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

# Konum Kurulum

```
dependencies { this: DependencyHandlerScope }  
  
implementation("androidx.core:core-ktx:1.9.0")  
implementation("androidx.appcompat:appcompat:1.6.1")  
implementation("com.google.android.material:material:1.9.0")  
implementation("androidx.constraintlayout:constraintlayout:2.1.4")  
testImplementation("junit:junit:4.13.2")  
androidTestImplementation("androidx.test.ext:junit:1.1.5")  
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")  
  
implementation("com.google.android.gms:play-services-location:21.0.1")  
}  
  
implementation("com.google.android.gms:play-services-location:21.0.1")
```

build.gradle/app



# ViewBinding Kullanımı

*Build.gradle/module*

```
android {  
    compileSdk 31  
  
    buildFeatures{  
        viewBinding = true  
    }  
  
    defaultConfig {  
        applicationId "com.example.myapplication"  
        minSdk 21  
    }  
}
```

*activity\_main.xml*

Component Tree

- ConstraintLayout
  - buttonx "Button"

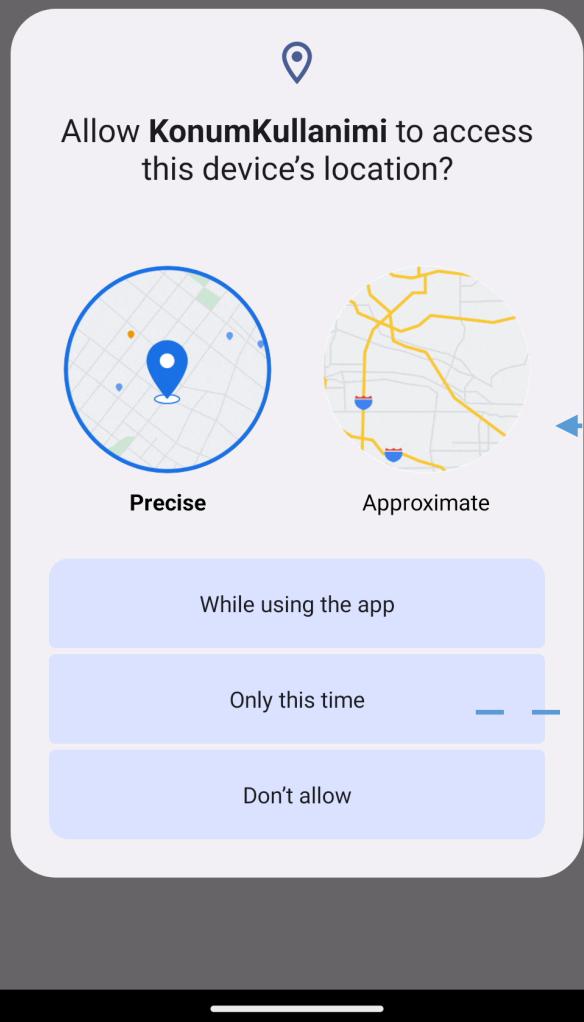
1

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tasarim: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        tasarim = ActivityMainBinding.inflate(LayoutInflater)  
        setContentView(tasarim.root)  
  
        tasarim.buttonx.setOnClickListener { it: View!  
            Snackbar.make(it, text: "Merhaba", Snackbar.LENGTH_SHORT).show()  
        }  
    }  
}
```

2

3



```

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private var izinKontrol = 0
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.buttonKonumAl.setOnClickListener { it: View! -->
            izinKontrol = ContextCompat.checkSelfPermission(context: this,Manifest.permission.ACCESS_FINE_LOCATION)

            if (izinKontrol == PackageManager.PERMISSION_GRANTED){//Onay verilmişse
                ActivityCompat.requestPermissions(activity: this,
                    arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), requestCode: 100)
            }
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if(requestCode == 100){
            if(grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED){
                Toast.makeText(applicationContext, text: "İzin onaylandı",Toast.LENGTH_SHORT).show()
            }else{
                Toast.makeText(applicationContext, text: "İzin onaylanmadı",Toast.LENGTH_SHORT).show()
            }
        }
    }
}

```

*Onaylatma arayüzü açar.  
Array içine manifest  
dosyasında yer alan  
onaylatmak istediğimiz  
izinleri yazarız.*

*Kullanıcı onayının kontrol edilmesi.*



Enlem : 37.4219983

Boylam : -122.084

Konum Al

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private var izinKontrol = 0

    private lateinit var flpc:FusedLocationProviderClient
    private lateinit var locationTask:Task<Location>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        flpc = LocationServices.getFusedLocationProviderClient(activity: this)

        binding.buttonKonumAl.setOnClickListener { it: View!
            izinKontrol = ContextCompat.checkSelfPermission(context: this, Manifest.permission.ACCESS_FINE_LOCATION)

            if (izinKontrol == PackageManager.PERMISSION_GRANTED){//Onay verilmişse
                locationTask = flpc.lastLocation
                konumBilgisiAl()
            }else{
                ActivityCompat.requestPermissions(activity: this,
                    arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), requestCode: 100)
            }
        }
    }
}
```

4:14 ☁ ⚡



...

```
fun konumBilgisiAl(){
    locationTask.addOnSuccessListener{location : Location ->
        if (location != null){
            binding.textViewEnlem.text = "Enlem : ${location.latitude}"
            binding.textViewBoylam.text = "Boylam : ${location.longitude}"
        }else{
            binding.textViewEnlem.text = "Enlem : Bulunamadi"
            binding.textViewBoylam.text = "Boylam : Bulunamadi"
        }
    }
}
```

Enlem : 37.4219983

Boylam : -122.084

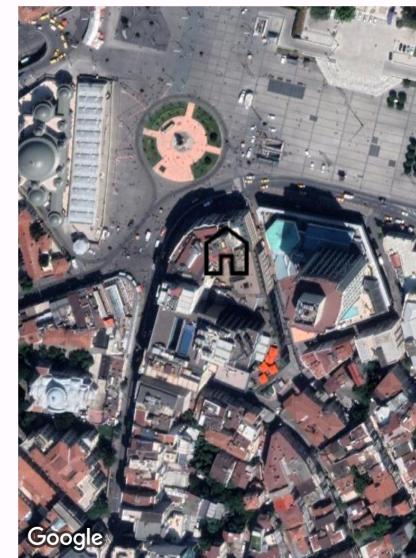
Konum Al

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if(requestCode == 100){
        izinKontrol = ContextCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION)

        if(grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED){
            Toast.makeText(applicationContext, text: "İzin onaylandı", Toast.LENGTH_SHORT).show()
            locationTask = flpc.lastLocation
            konumBilgisiAl()
        }else{
            Toast.makeText(applicationContext, text: "İzin onaylanmadı", Toast.LENGTH_SHORT).show()
        }
    }
}
```

# Harita Kullanımı

# Harita Kullanımı



KONUMA GİT

KONUMA GİT

# Kurulum

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

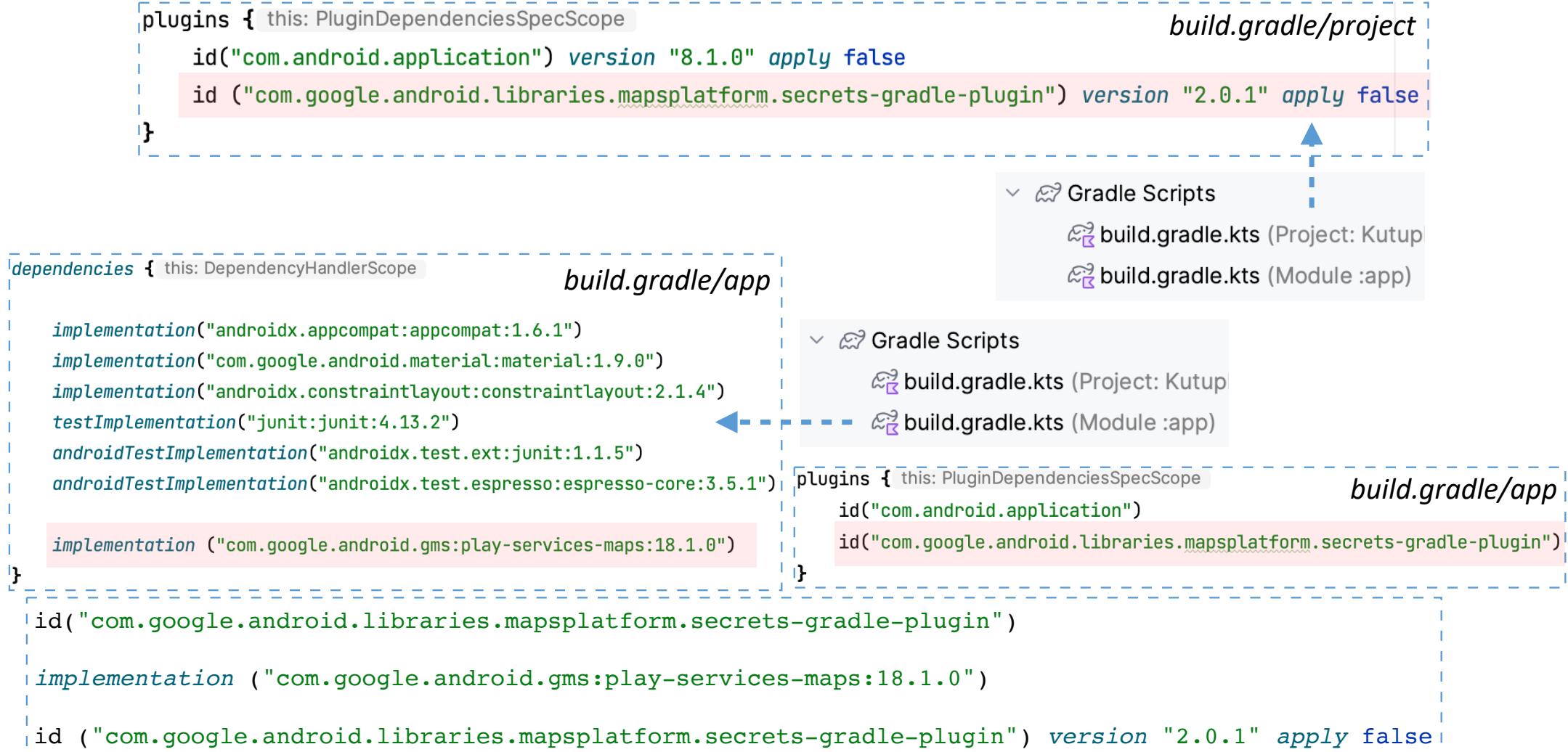
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyDyynJC04P_USAWmzvMZ-y29s0Xv6vUVR8" />

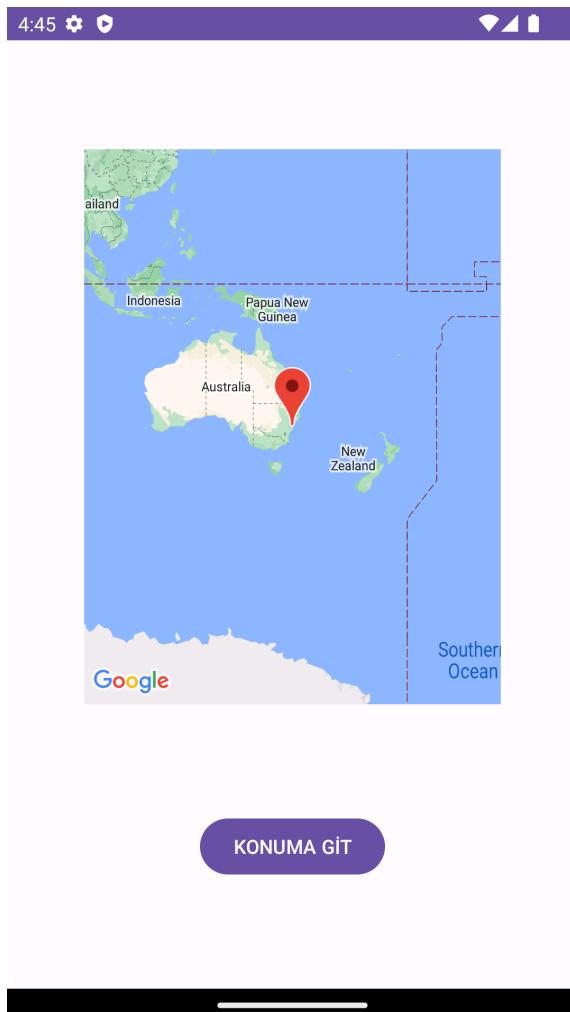
</application>
</manifest>

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyDyynJC04P_USAWmzvMZ-y29s0Xv6vUVR8" />
```

# Kurulum



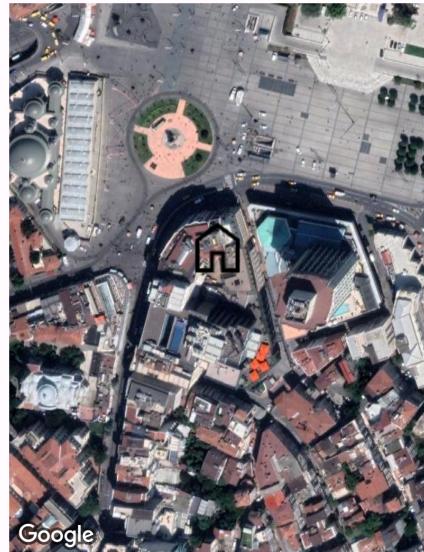
# Tasarım



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="300dp"
        android:layout_height="400dp"
        app:layout_constraintBottom_toTopOf="@+id/buttonKonumaGit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:context=".MapsActivity" />

    <Button
        android:id="@+id/buttonKonumaGit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="KONUMA GIT"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/map" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



```
class MainActivity : AppCompatActivity() , OnMapReadyCallback {  
    private lateinit var binding: ActivityMainBinding  
    private lateinit var mMap: GoogleMap  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        val mapFragment = supportFragmentManager  
            .findFragmentById(R.id.map) as SupportMapFragment  
        mapFragment.getMapAsync(this)  
  
        binding.buttonKonumaGit.setOnClickListener {  
            val konum = LatLng(41.0361566, 28.9854576)  
            mMap.addMarker(MarkerOptions()  
                .position(konum)  
                .title("Konum").icon(BitmapDescriptorFactory.fromResource(R.drawable.resim)))  
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(konum, 17f))  
            mMap.mapType = GoogleMap.MAP_TYPE_SATELLITE  
        }  
    }  
  
    override fun onMapReady(googleMap: GoogleMap) {  
        mMap = googleMap  
  
        val konum = LatLng(-34.0, 151.0)  
        mMap.addMarker(MarkerOptions()  
            .position(konum)  
            .title("Sydney"))  
        mMap.moveCamera(CameraUpdateFactory.newLatLng(konum))  
    }  
}
```

*resim türü png olmalıdır.*

val konum = LatLng(41.0361566, 28.9854576)

Teşekkürler...



kasim-adalan



kasimadalan@gmail.com



kasimadalan