

Android Mobil Uygulama Geliştirme Eğitimi | Kotlin

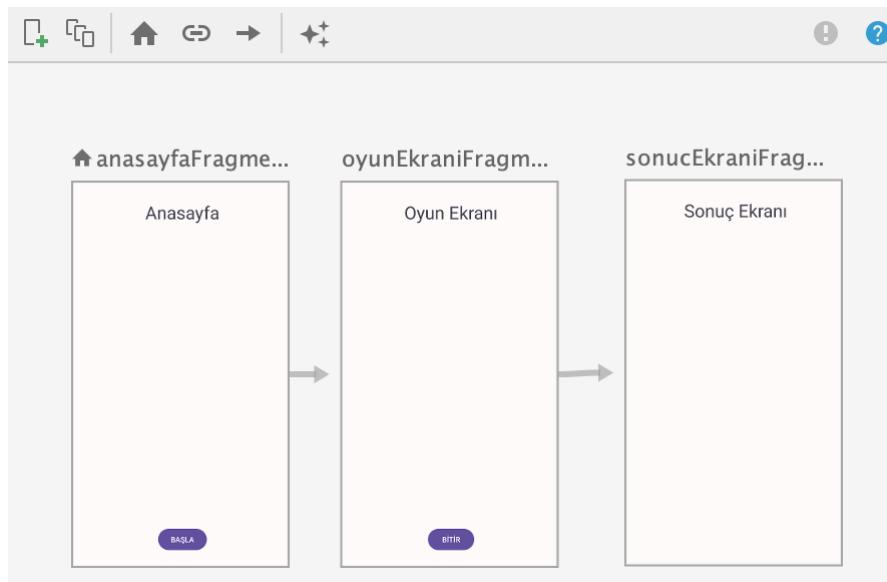
Navigation Component Kullanımı

Kasım ADALAN
Elektronik ve Haberleşme Mühendisi
Android - IOS Developer and Trainer

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.

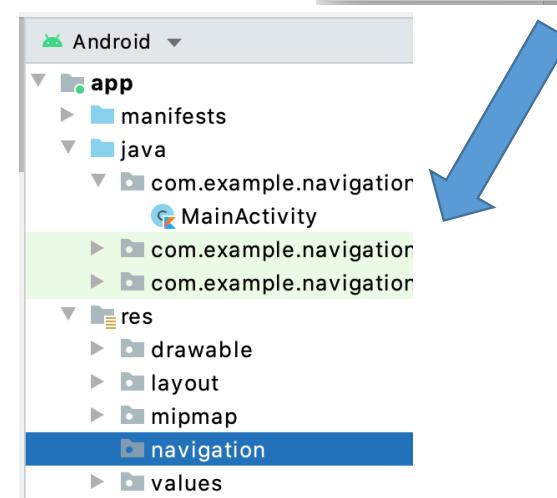
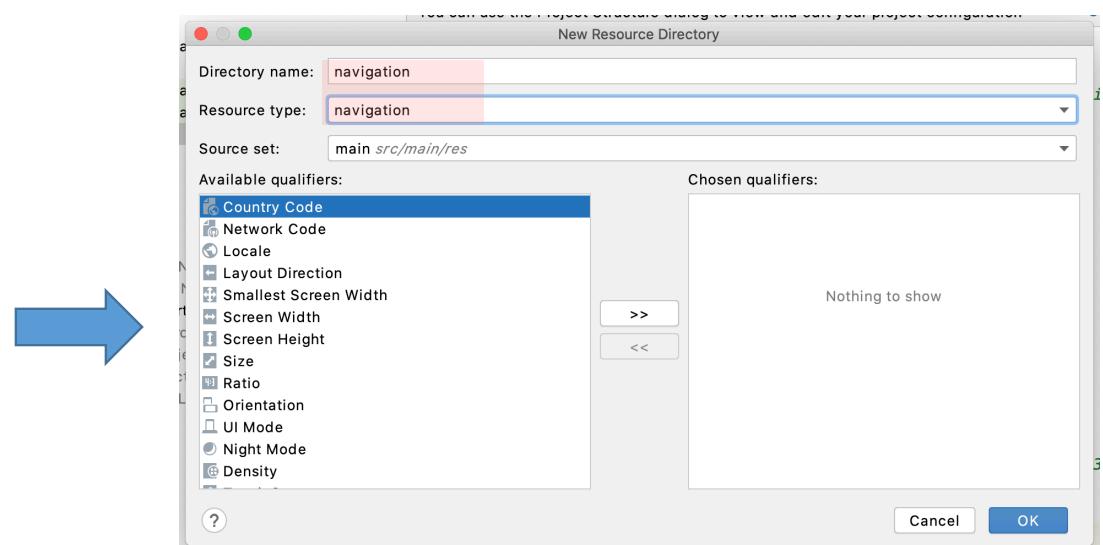
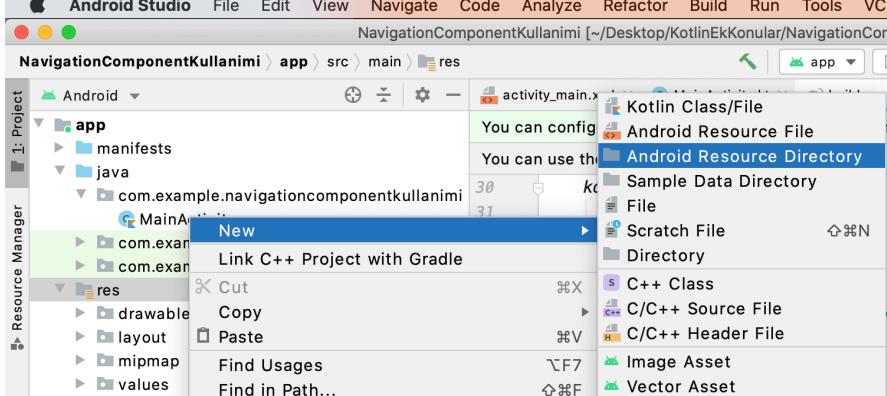


Kurulum

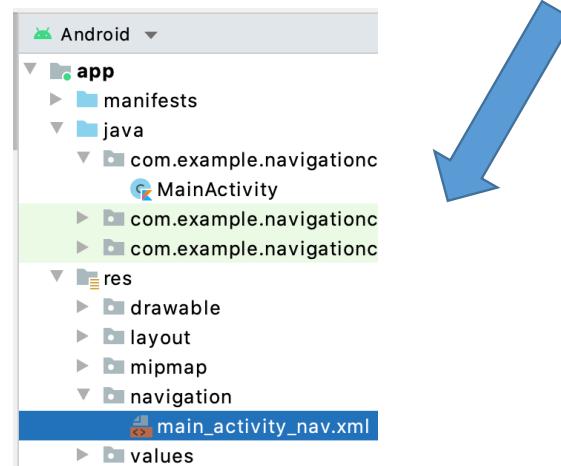
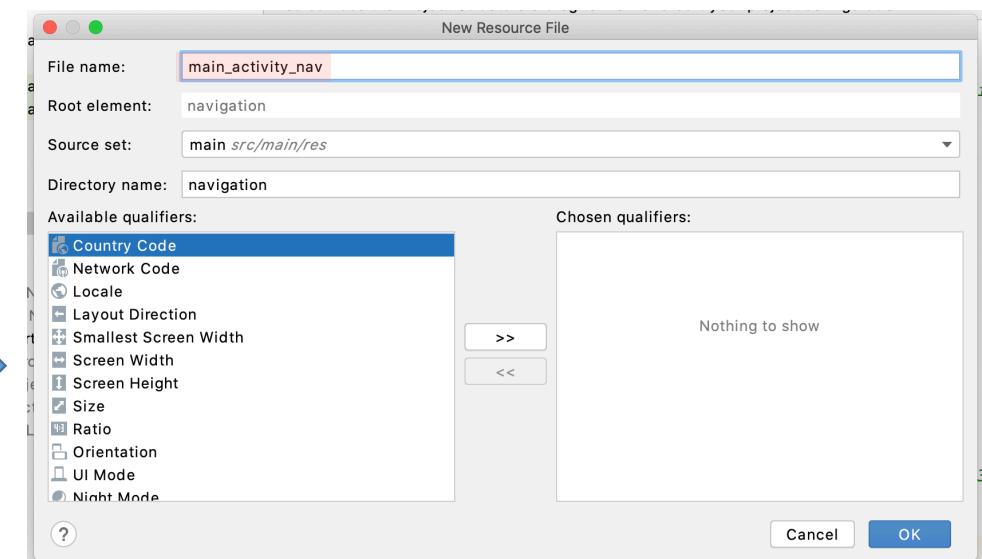
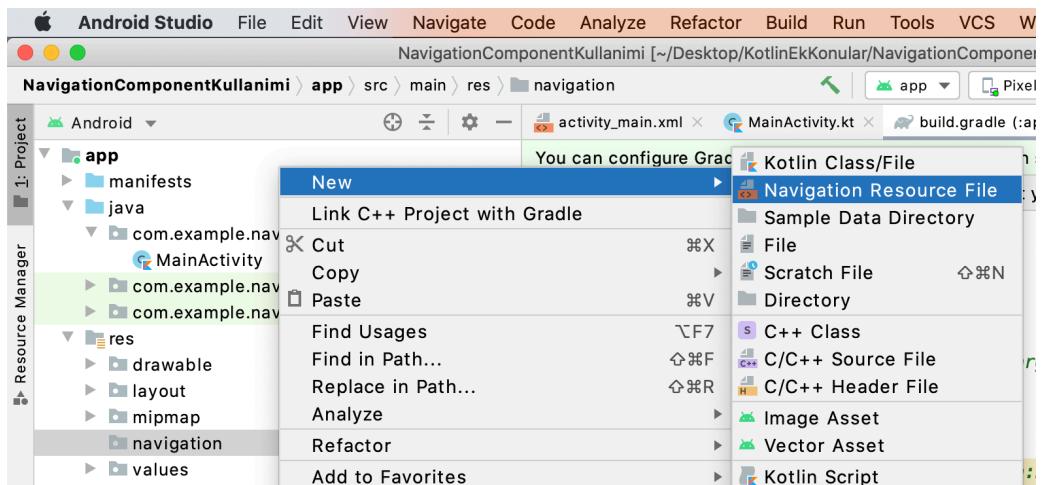
```
dependencies {  
  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.2.0'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.3.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    //Navigation Component  
    implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
    implementation "androidx.navigation:navigation-ui-ktx:2.3.3"  
}
```

```
implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
implementation "androidx.navigation:navigation-ui-ktx:2.3.3"
```

Navigation Klasörü Oluşturma

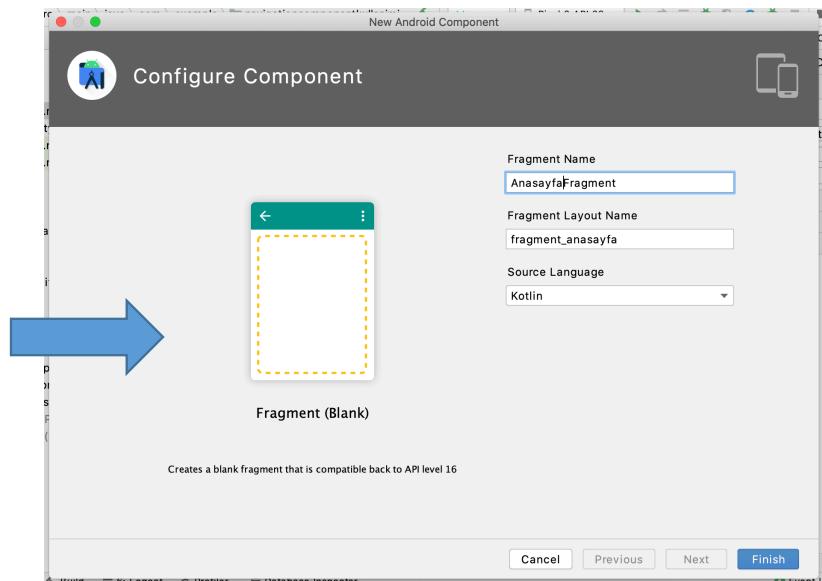
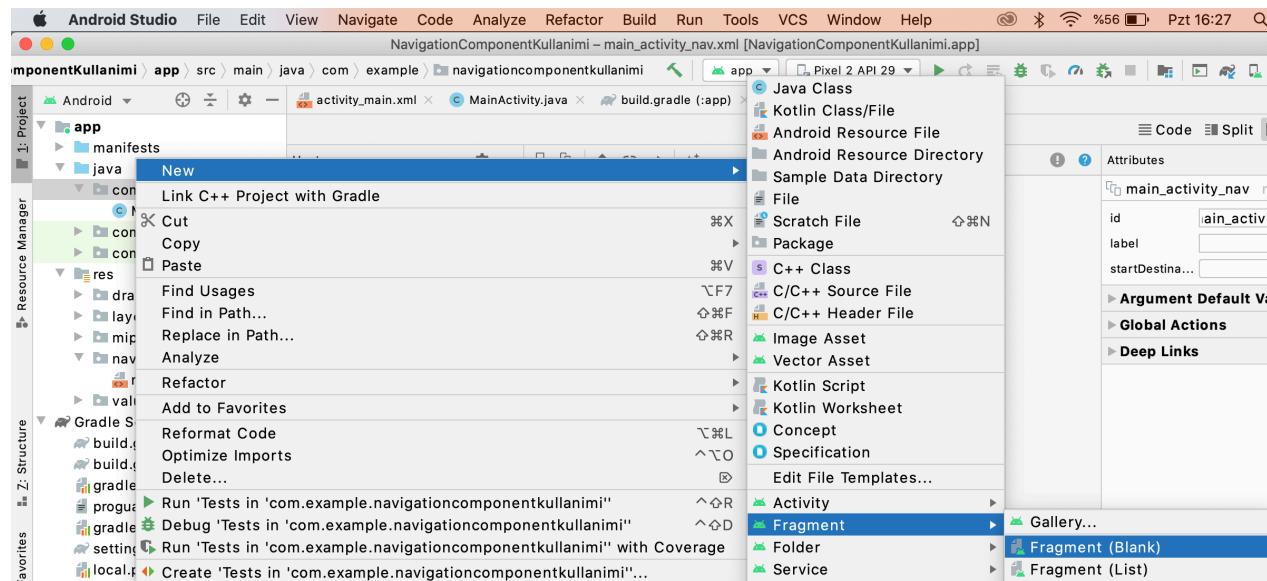


Navigation Dosyası Oluşturma



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

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.

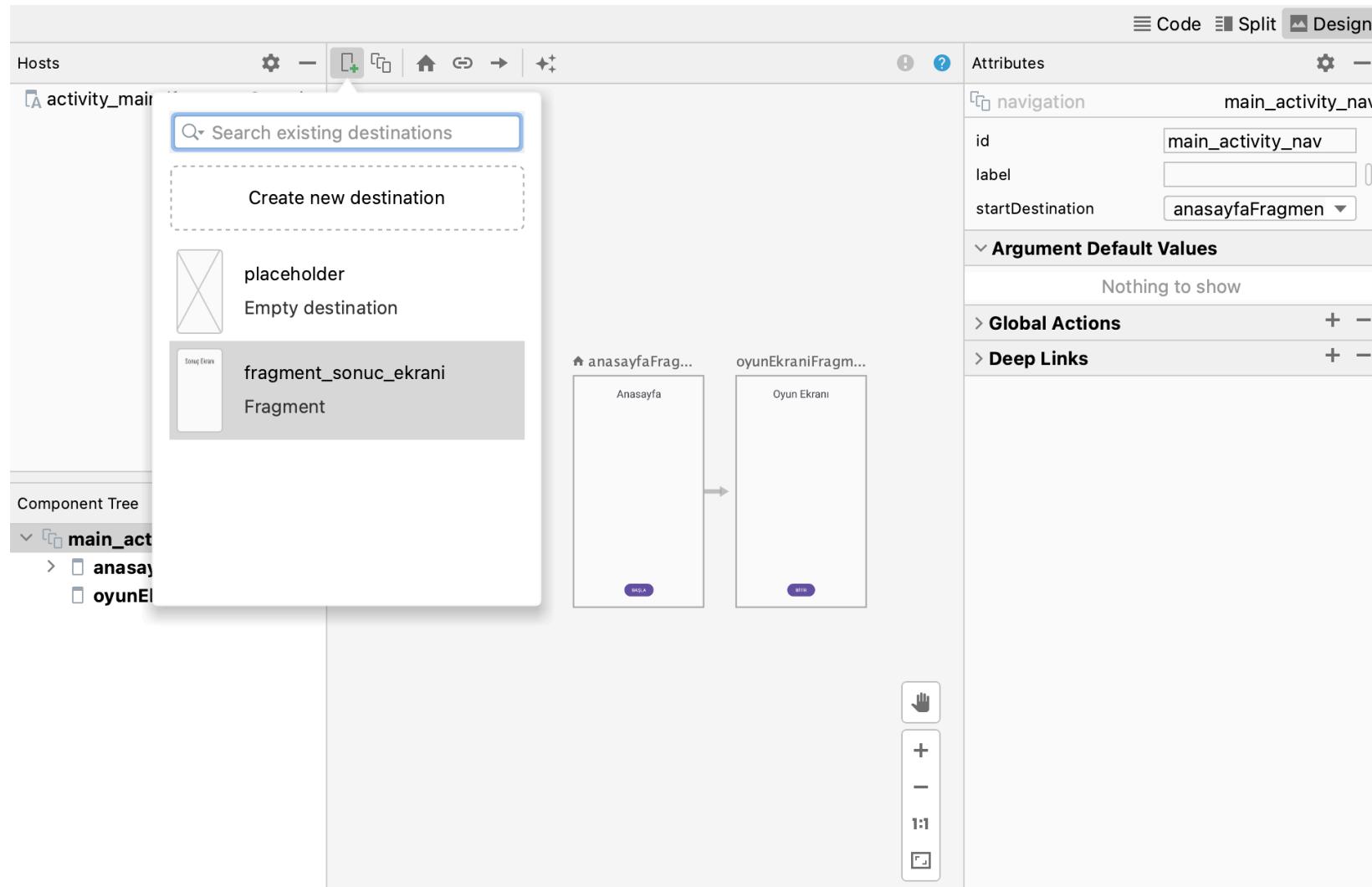
Fragment Oluşturma



Component Tree

```
└ ConstraintLayout
  └ textView "Anasayfa"
  └ buttonBasla "BAŞLA"
```

Navigation Component'e Fragment Ekleme



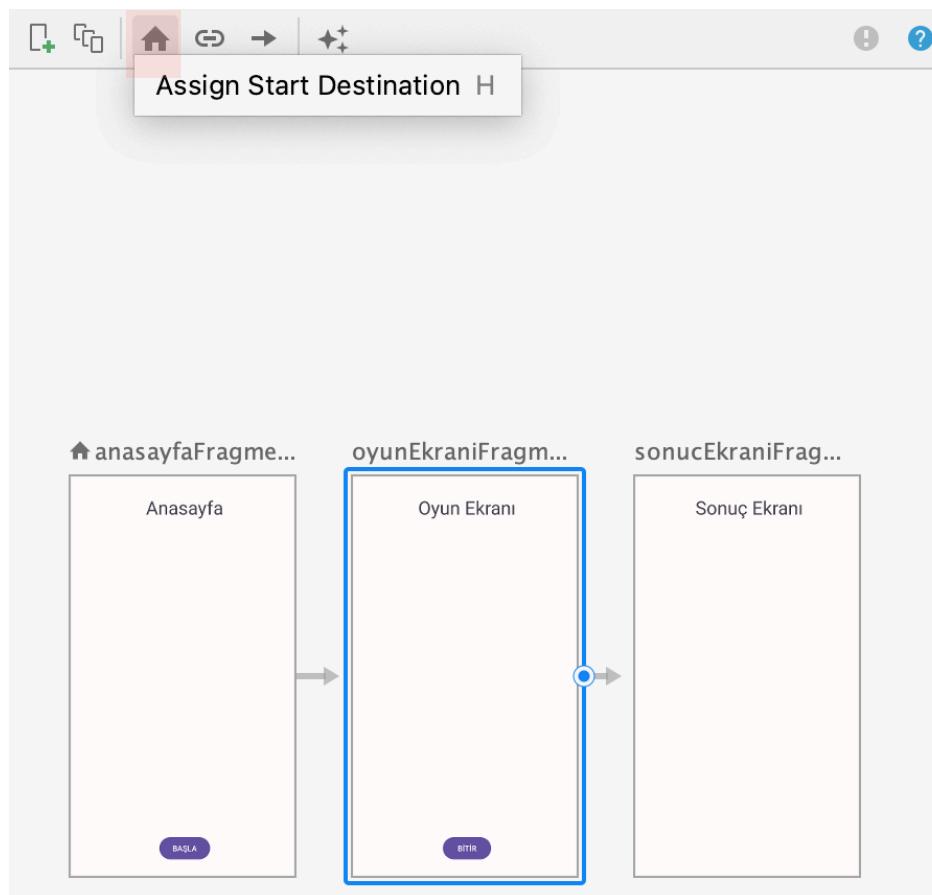
Navigation Component'e Fragment Ekleme

The screenshot shows the Android Studio interface with the Navigation Editor open. On the left, the Component Tree shows a navigation graph named 'main_activity_nav' containing three fragments: 'anasayfaFragment', 'oyunEkranıFragment', and 'sonucEkranıFragment'. The main area displays a flow diagram with three boxes labeled 'Anasayfa', 'Oyun Ekranı', and 'Sonuç Ekranı'. Arrows indicate transitions between them. A 'BASLA' button is at the bottom left, and a 'BITIR' button is at the bottom right. On the far right, the XML code for the navigation graph is shown:

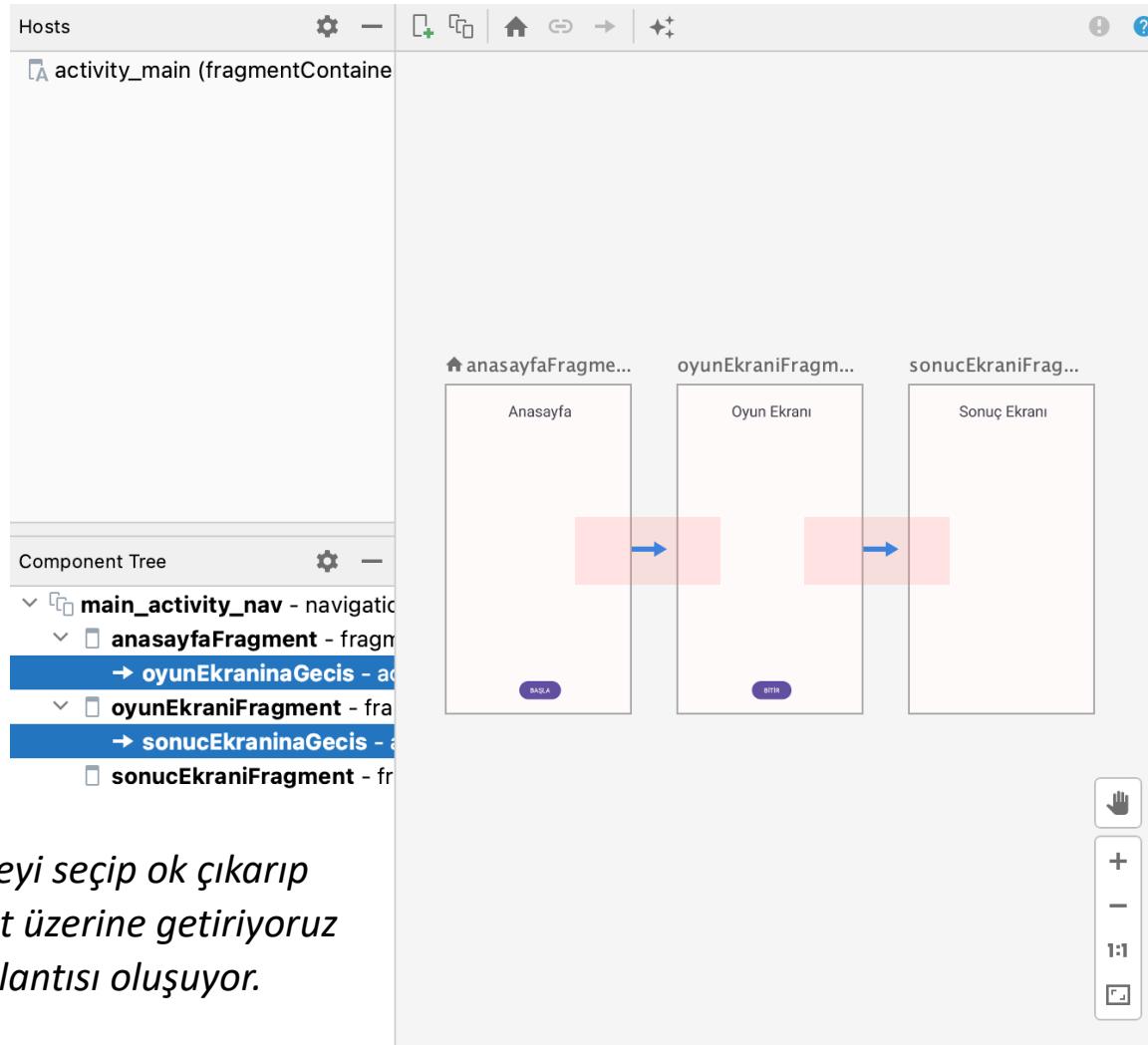
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_activity_nav"
    app:startDestination="@+id/anasayfaFragment">
    <fragment
        android:id="@+id/anasayfaFragment"
        android:name="com.example.navigationcomponentkullanimi.AnasayfaFragment"
        android:label="fragment_anasayfa"
        tools:layout="@layout/fragment_anasayfa" >
    </fragment>
    <fragment
        android:id="@+id/oyunEkranıFragment"
        android:name="com.example.navigationcomponentkullanimi.OyunEkranıFragment"
        android:label="fragment_oyun_ekrani"
        tools:layout="@layout/fragment_oyun_ekrani" >
    </fragment>
    <fragment
        android:id="@+id/sonucEkranıFragment"
        android:name="com.example.navigationcomponentkullanimi.SonucEkranıFragment"
        android:label="fragment_sonuc_ekrani"
        tools:layout="@layout/fragment_sonuc_ekrani" >
    </fragment>
</navigation>
```

Ana Fragmentı Değiştirebilme

İstediğimiz sayfayı ilk sayfa yani anasayfa yapabiliriz.



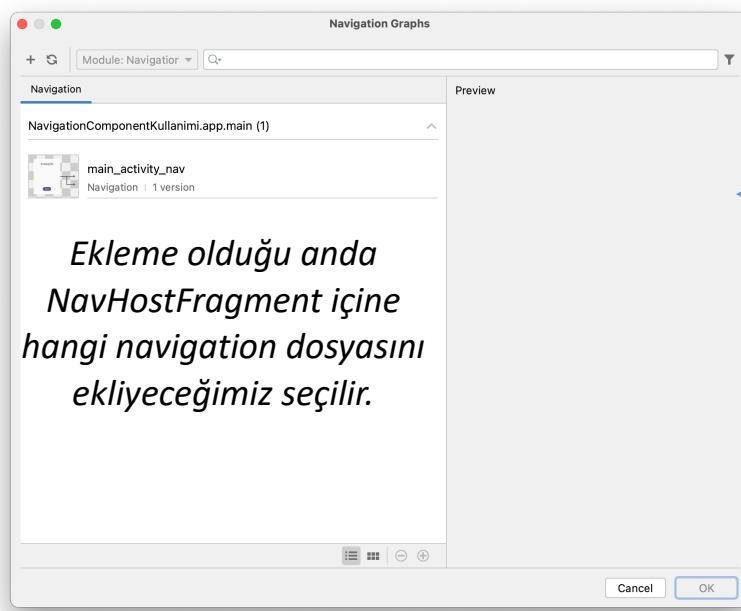
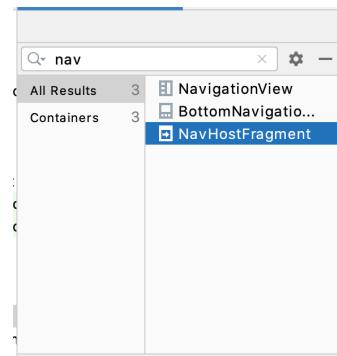
Fragmentlar Arası Geçiş Bağlantısı Oluşturma



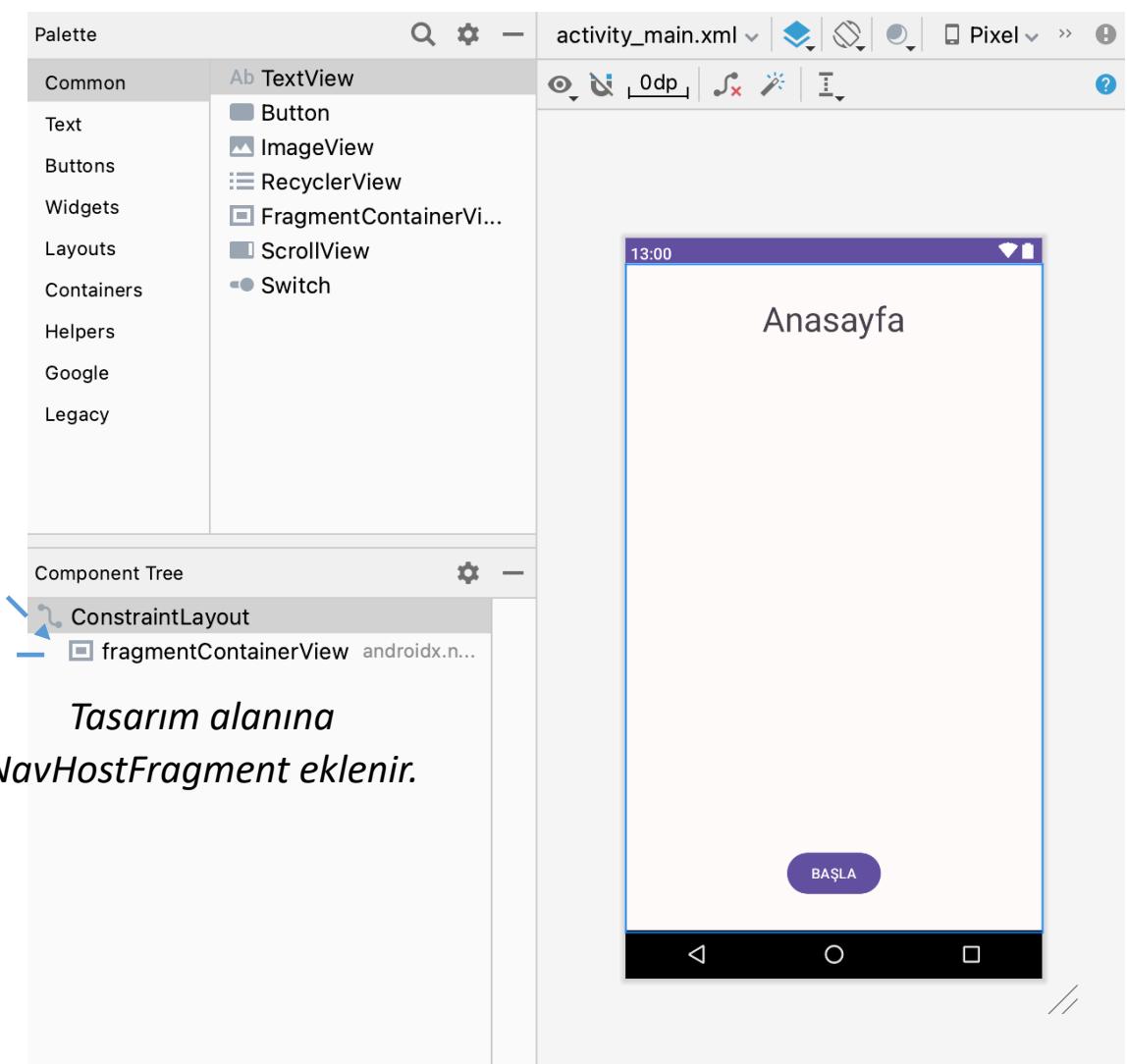
*Fragment yanındaki daireyi seçip ok çıkarıp
geçiş yapıcıımız fragment üzerine getiriyoruz
ve bırakınca geçiş bağlantısı oluşuyor.*

Activity Tasarımına Navigation Dosyasını Bağlama

Activity üzerinde geçişlerin olması ve fragmentların yer alması için bağlantı gereklidir.

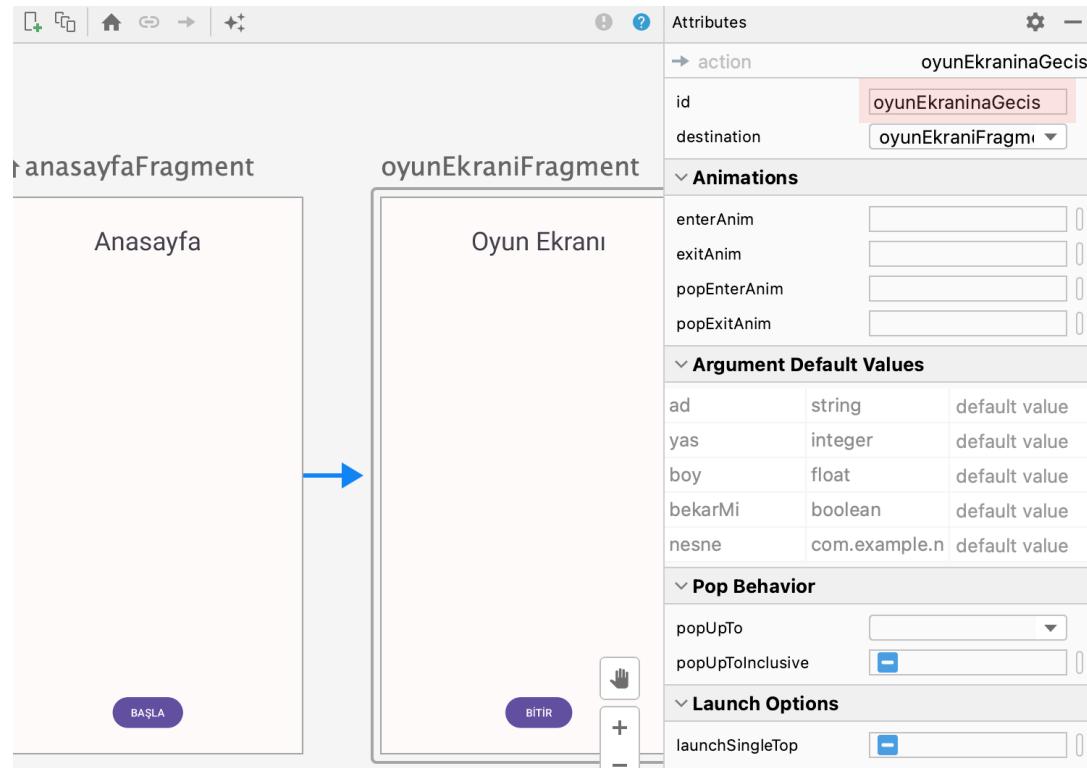


Ekleme olduğu anda
NavHostFragment içine
hangi navigation dosyasını
ekliyeceğimiz seçilir.



Tasarım alanına
NavHostFragment eklenir.

Sayfa Geçiş Kodlaması

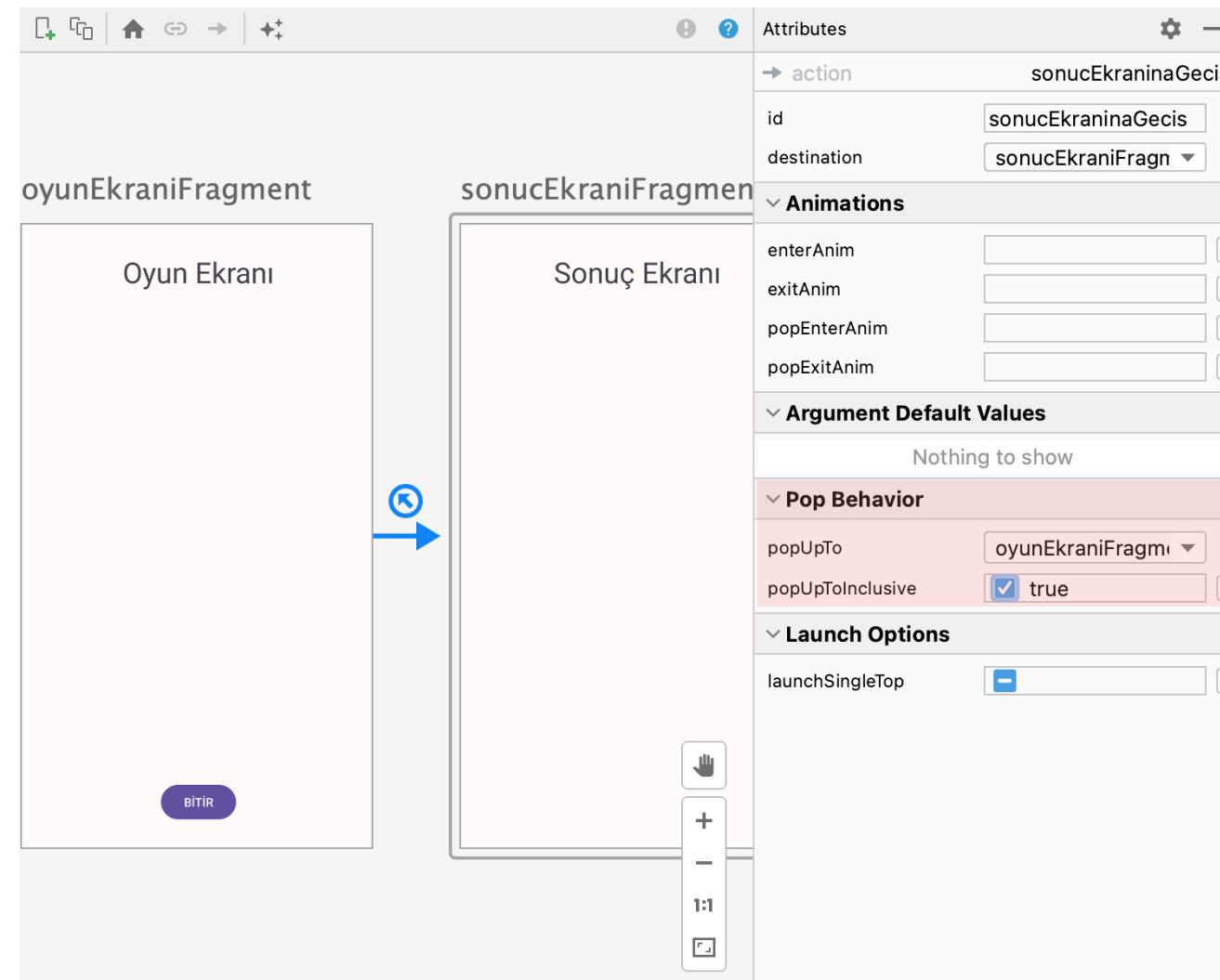


```
tasarim.buttonBasla.setOnClickListener { it: View!  
    Navigation.findNavController(it).navigate(R.id.oyunEkranaGecis)  
}
```

Geçiş id'si ile geçiş çalıştırılır.

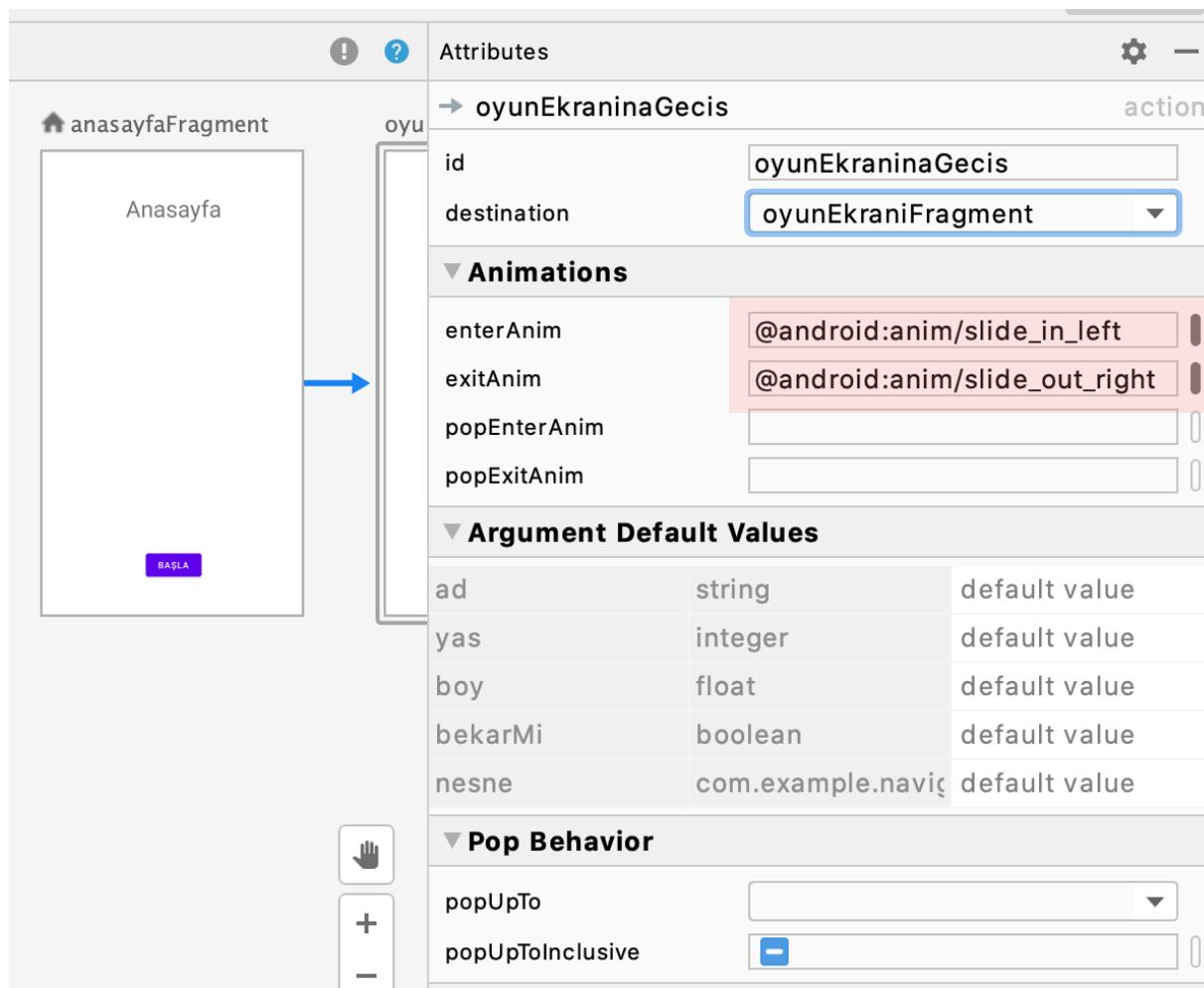
Geçiş id'si istediğimiz gibi isimlendirilebilir.

Back Stack



- Geçişe back stack silme özelliği ekleyebiliriz.
- Geçiş yaparken back stackten silmek istediğimiz sayfayı belirtmemiz yeterlidir.
- `finish()` metodu özelliğini sergiler.

Sayfa Geçişi Animasyonu



Sayfalar Arası Veri Transferi

Kurulum

- Safe args özelliği veri transferi için kurulum gereklidir.

```
//Safe args
apply plugin: "androidx.navigation.safeargs.kotlin" build.gradle/app

dependencies {

    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.2.0'
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    //Navigation Component
    implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"
    implementation "androidx.navigation:navigation-ui-ktx:2.3.3"
}
```

apply plugin: "androidx.navigation.safeargs.kotlin"

```
apply plugin: "androidx.navigation.safeargs.kotlin"
classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.5.3"
```

```
classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.4.1"

buildscript {
    repositories {
        google()
    }

    dependencies {
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.4.1"
    }
}

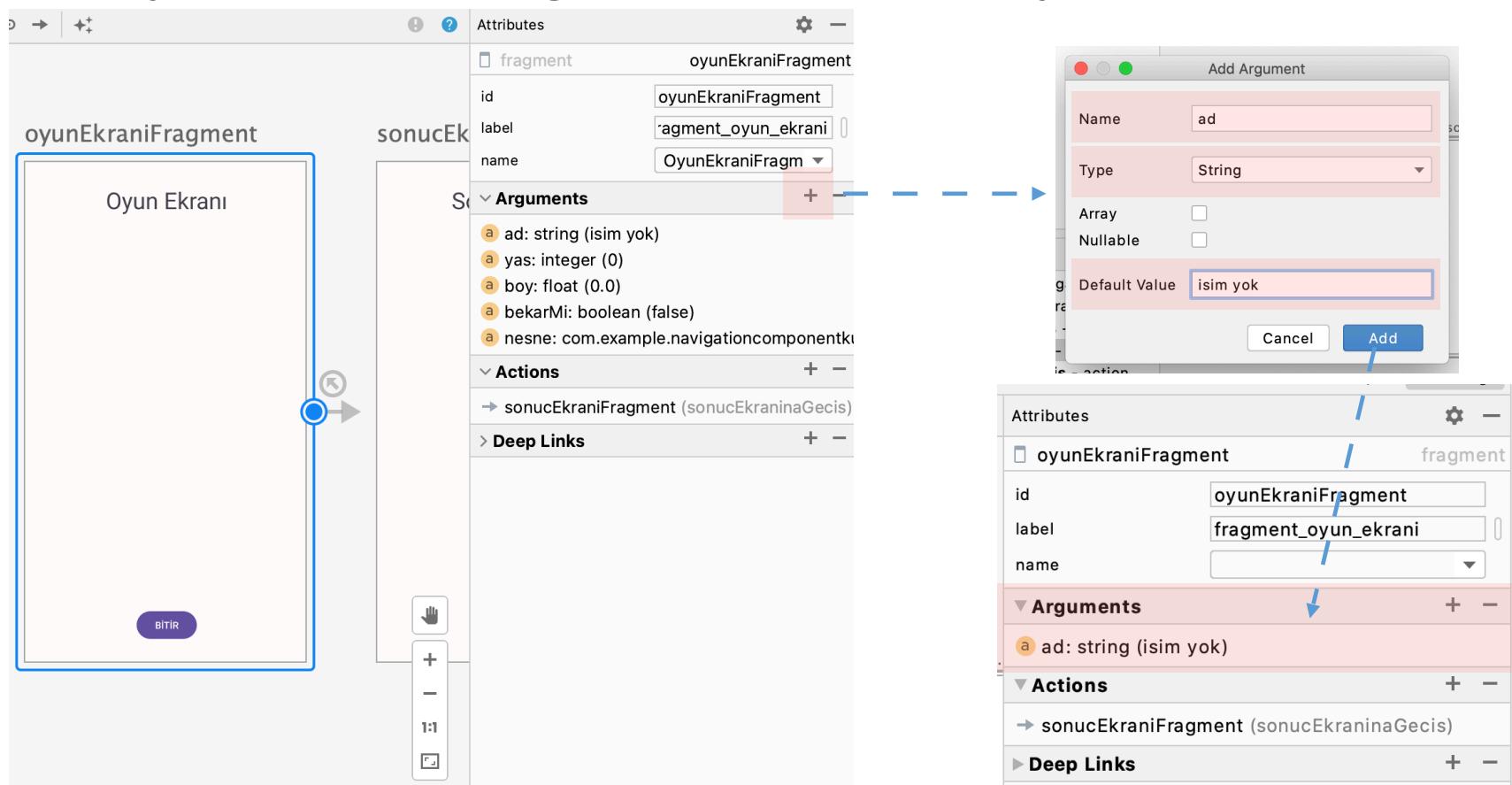
plugins {
    id 'com.android.application' version '7.1.1' apply false
    id 'com.android.library' version '7.1.1' apply false
    id 'org.jetbrains.kotlin.android' version '1.6.10' apply false
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

build.gradle/project

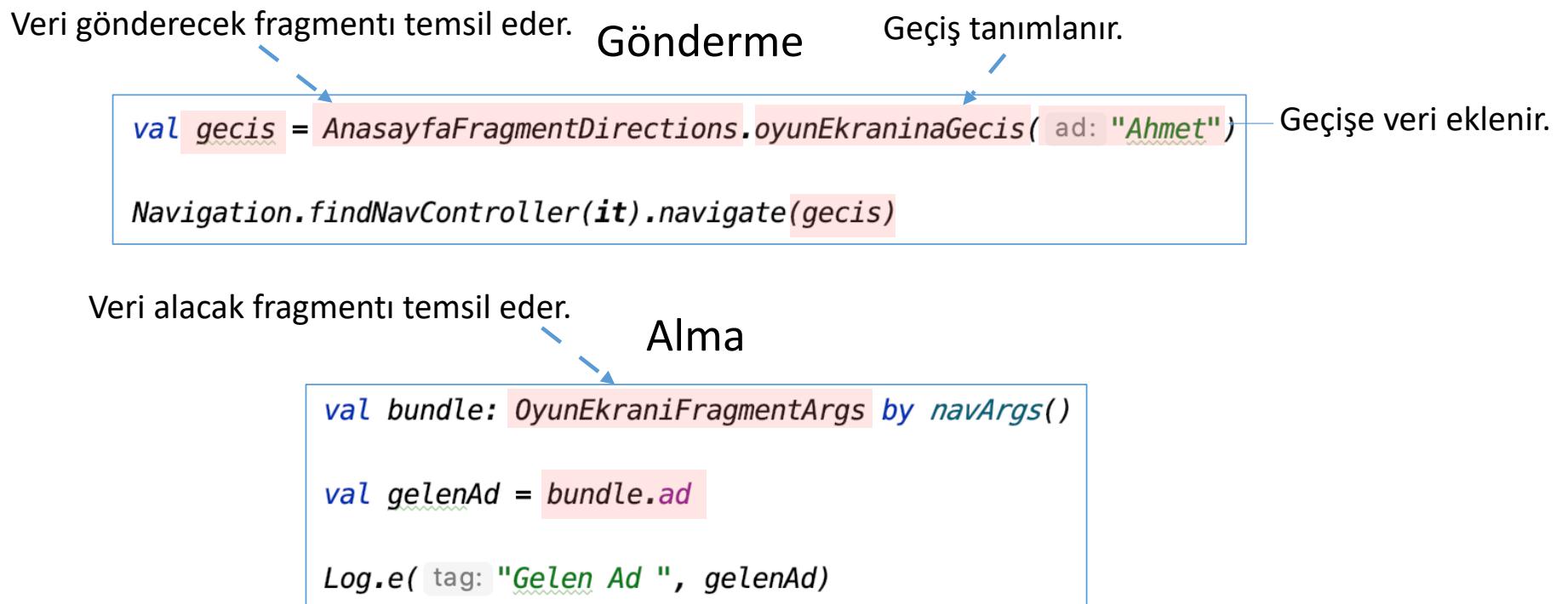
Sayfalar Arası Veri Transferi

- Veri göndermek istediğimiz fragment seçilir ve argument kısmında + işaretine basarak gönderilecek veri oluşturulur.

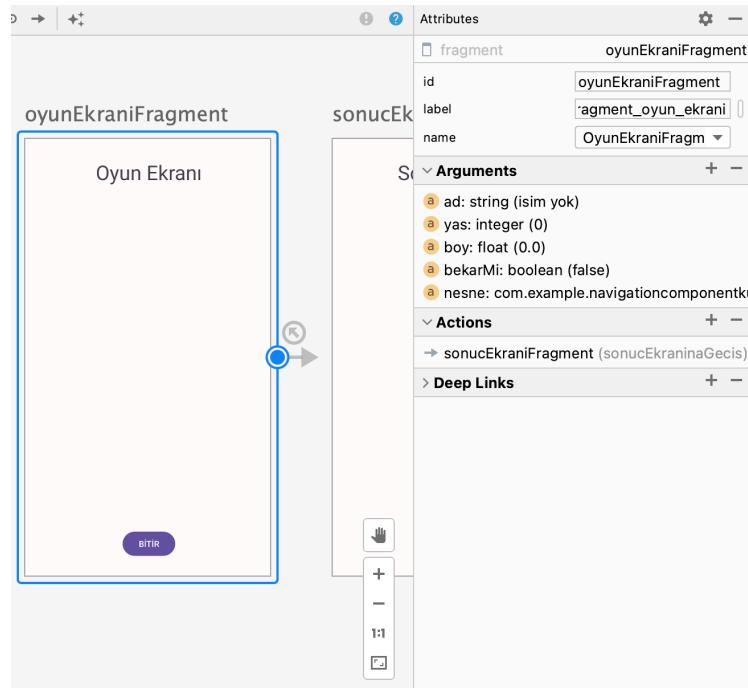


Sayfalar Arası Veri Transferi

- Veri Göndericeğimiz fragment'ın adına Directions eklenerek otomatik sınıf oluşturulur.
- Veri alacağımız Fragment'ın adına Args eklenerek otomatik sınıf oluşturulur.
- Eğer sınıflar otomatik oluşmuyorsa clean project ve rebuild project yapmalıyız.



Sayfalar Arası Veri Transferi



Gönderme

```
val gecis = AnasayfaFragmentDirections
    .oyunEkranaGecis( ad: "Ahmet", yas: 18, boy: 1.78f, bekarMi: true)

Navigation.findNavController(it).navigate(gecis)
```

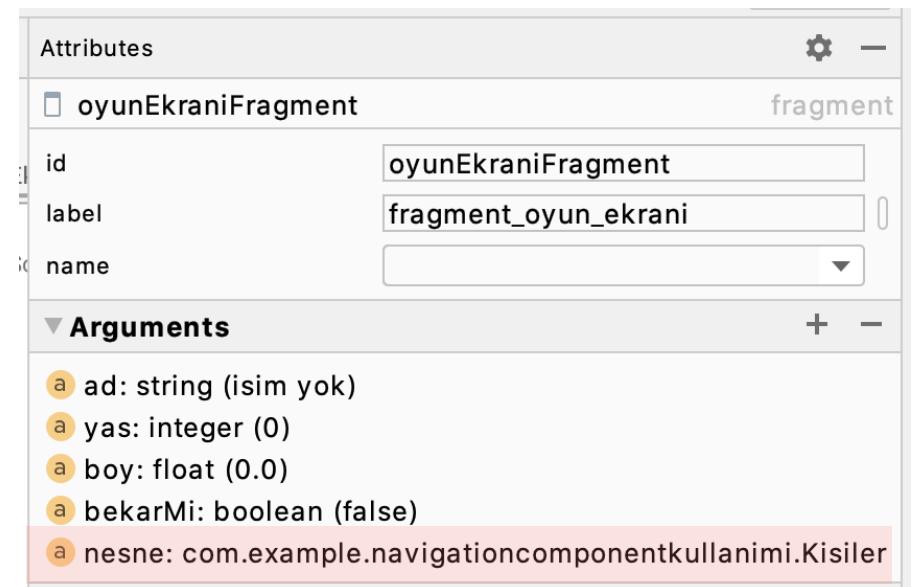
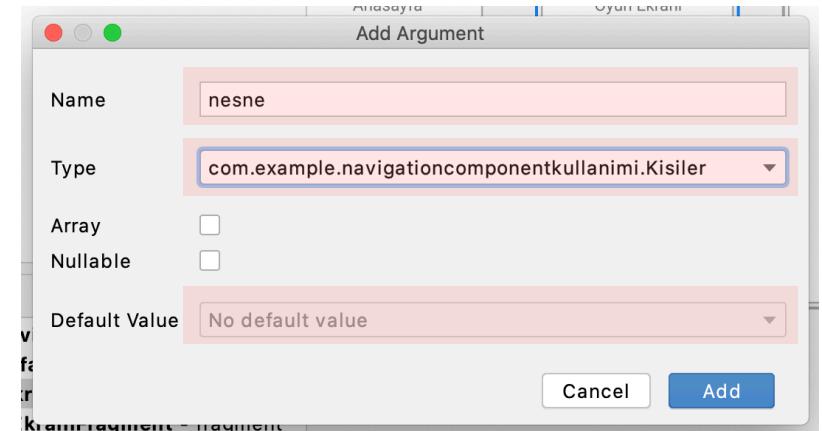
Alma

```
val bundle: OyunEkraniFragmentArgs by navArgs()

val gelenAd = bundle.ad
val gelenYas = bundle.yas
val gelenBoy = bundle.boy
val gelenBekarMi = bundle.bekarMi
```

Sayfalar Arası Veri Transferi - Nesne Tabanlı

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



Sayfalar Arası Veri Transferi - Nesne Tabanlı

Gönderme

```
val kisi = Kisiler( kisi_no: 1, kisi_ad: "Mehmet")  
  
val gecis = AnasayfaFragmentDirections  
    .oyunEkraninaGecis( ad: "Ahmet", yas: 18, boy: 1.78f, bekarMi: true, kisi)  
  
Navigation.findNavController(it).navigate(gecis)
```

Alma

```
val bundle: OyunEkranıFragmentArgs by navArgs()  
  
val gelenKisi = bundle.nesne
```

Navigation Component

ile

Bottom Navigation Kullanımı

Kurulum

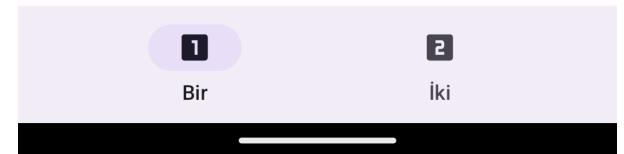
```
dependencies {  
  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.2.0'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.3.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    //Navigation Component  
    implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
    implementation "androidx.navigation:navigation-ui-ktx:2.3.3"  
}
```

```
implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
implementation "androidx.navigation:navigation-ui-ktx:2.3.3"
```

Bottom Navigation Kullanımı

- Navigation Component ile Bottom Navigation pratik şekilde kullanılabilir.
- Bottom navigation fragment tabanlı bir yapı olduğu için Navigation Component ile uyumlu çalışır.

BİR



Fragmet Tasarımı Oluşturma

- Fragment'lar için bir **sınıf** ve bir adet **Layout** oluşturuyoruz.

```
<?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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BİR"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

ADALAN



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
```

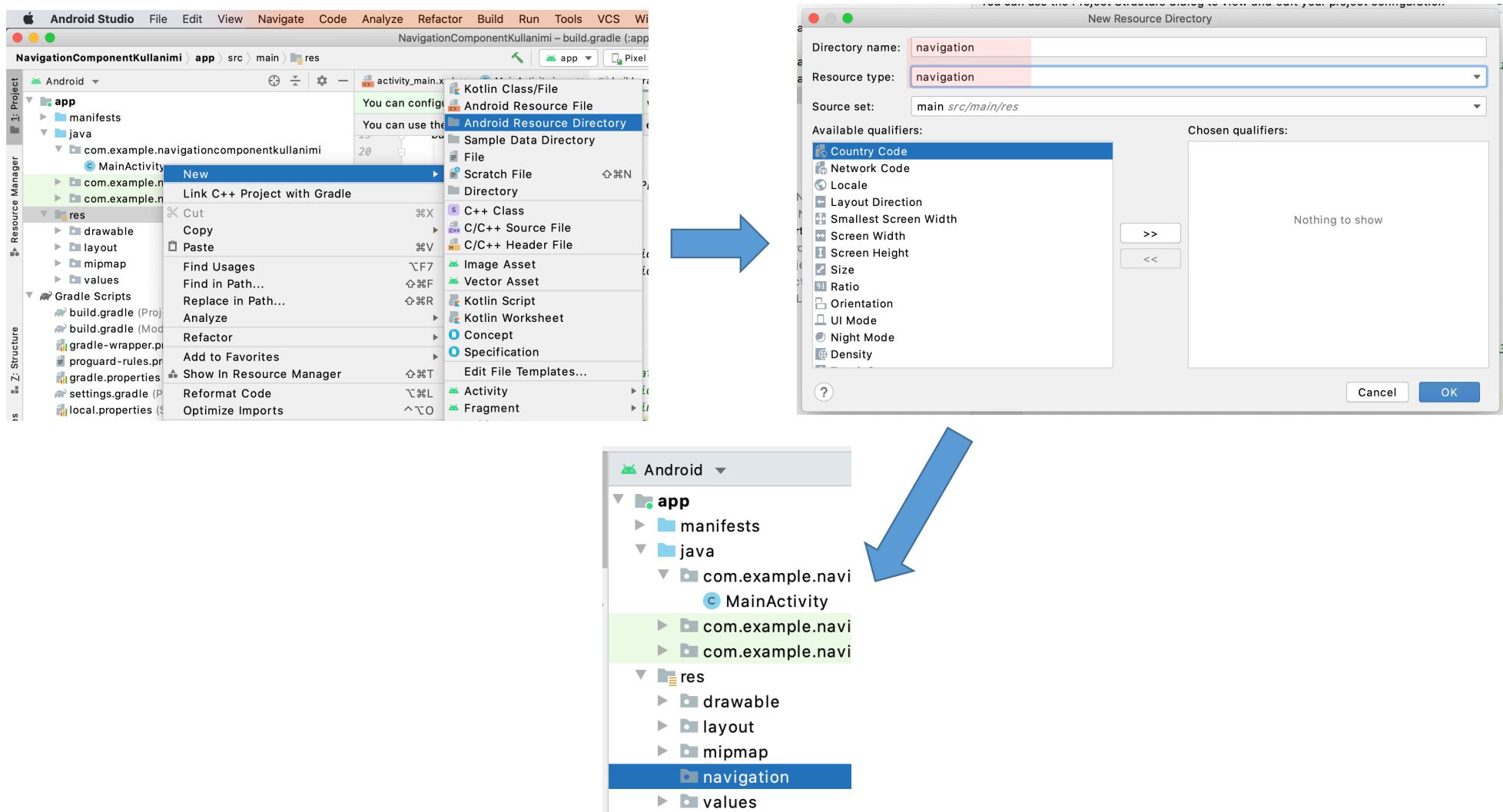
}

2

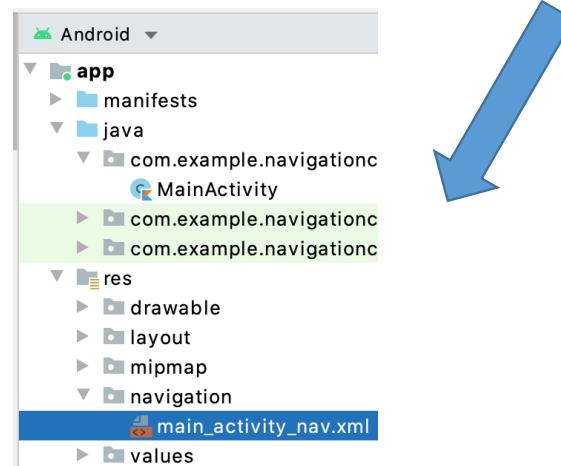
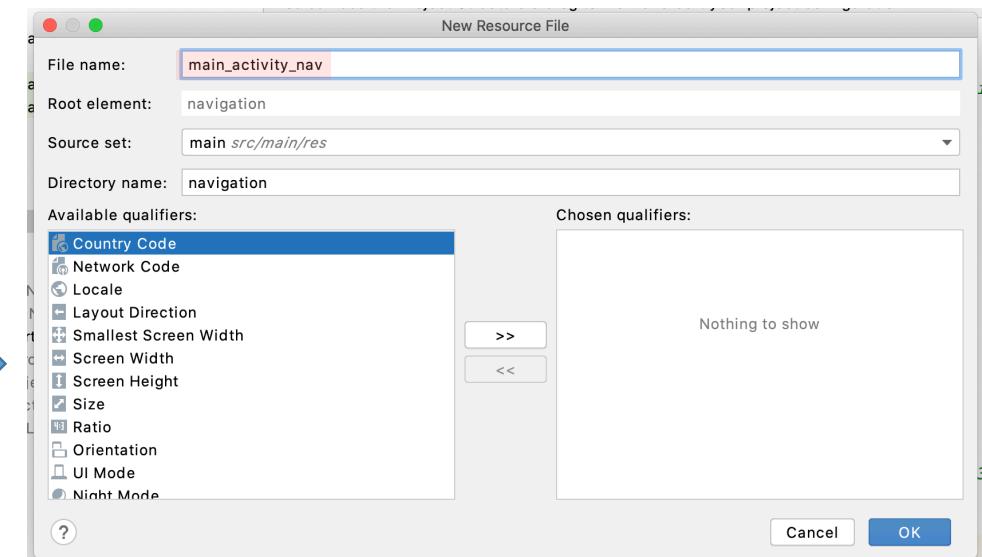
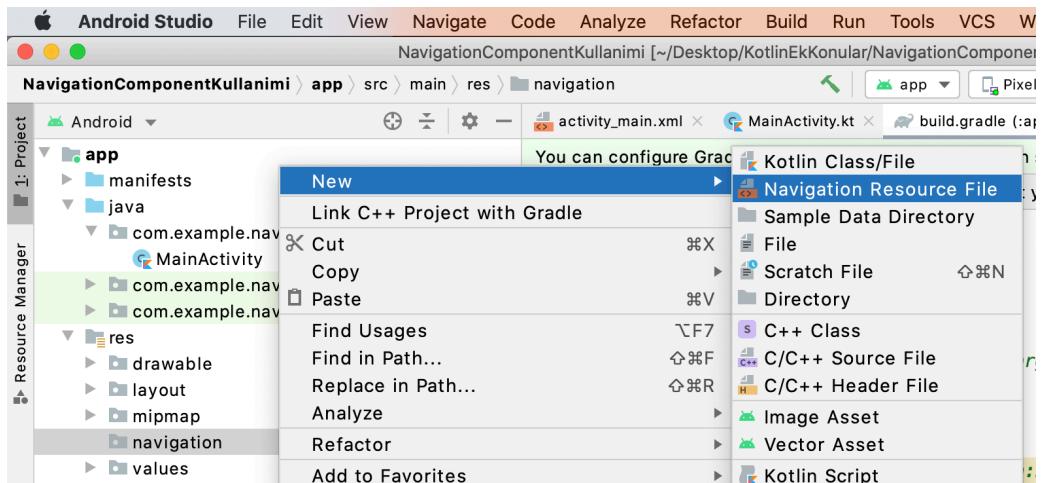
3

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

Navigation Klasörü Oluşturma

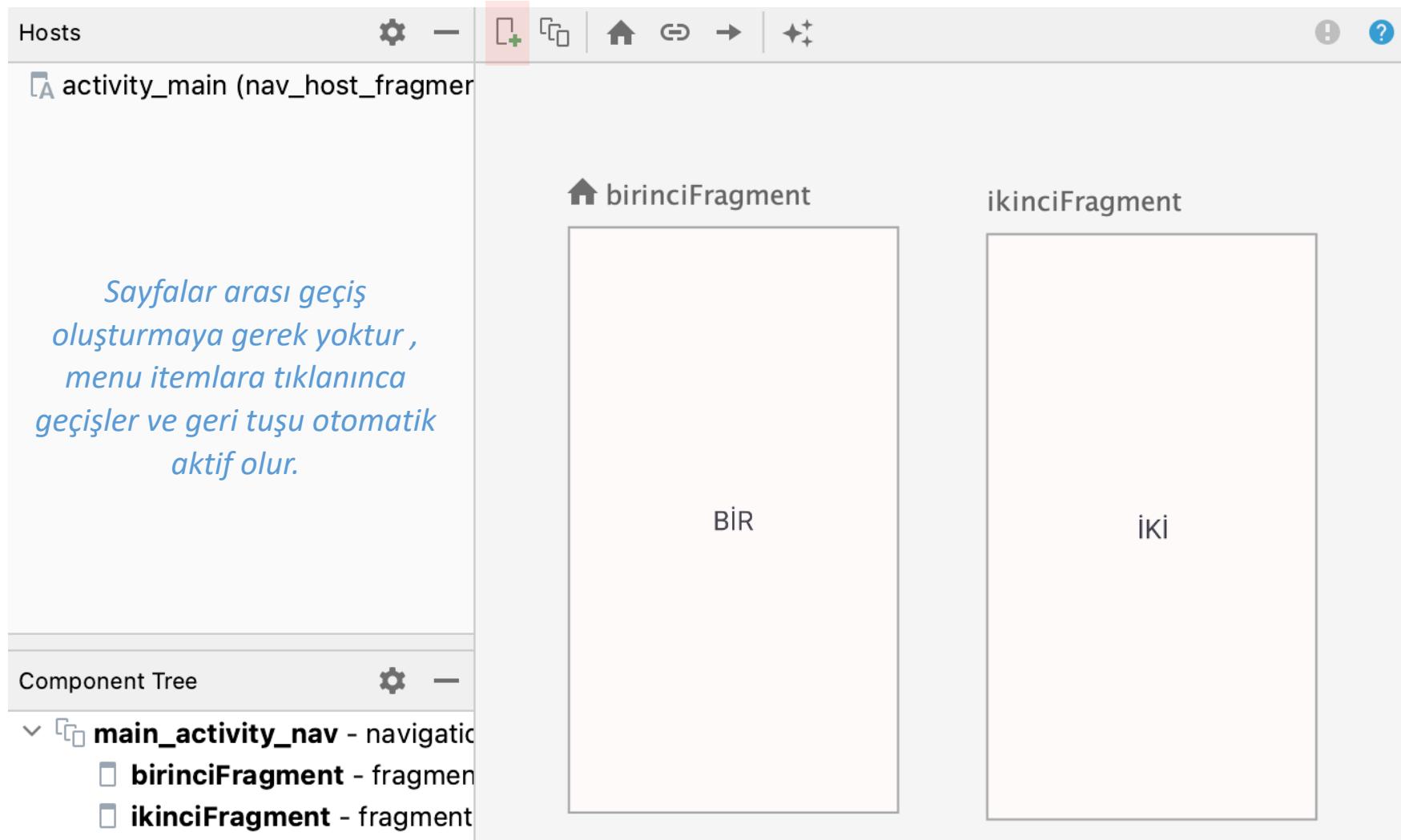


Navigation Dosyası Oluşturma



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

Navigation Component'e Fragment Ekleme



Bottom Navigation View için Menu Oluşturma

- Bottom navigation için menu oluşturulmalıdır.
- Menu item id ile navigation dosyasındaki fragment id aynı olmalıdır.
- Bottom navigation menu itemine tıklanınca ilgili id'li fragment açılacaktır.
- Örnek : Üzerinde Bir yazan item tıklanırsa birinciFragment id'li fragment açılacaktır.

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

    <item
        android:id="@+id/birinciFragment"
        android:icon="@drawable/resim_bir"
        android:title="Bir" />
    <item
        android:id="@+id/ikinciFragment"
        android:icon="@drawable/resim_iki"
        android:title="İki" />
</menu>
```

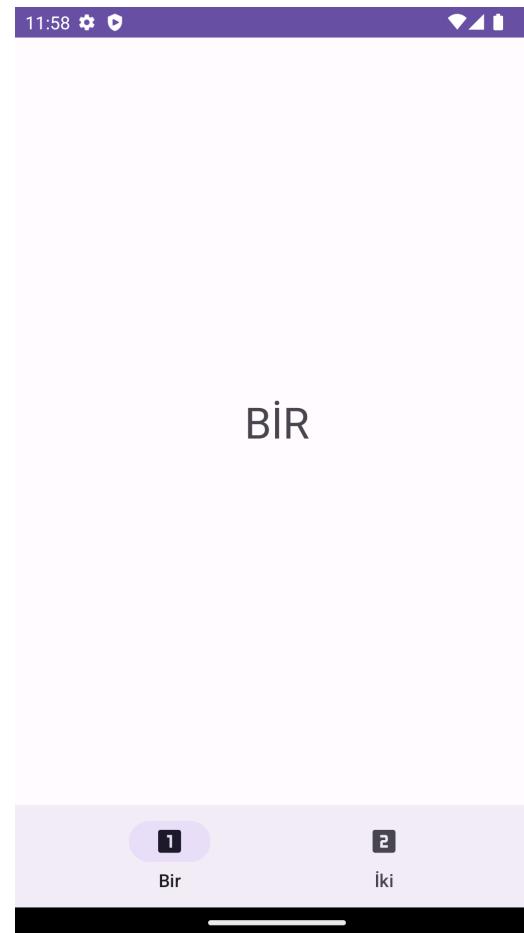
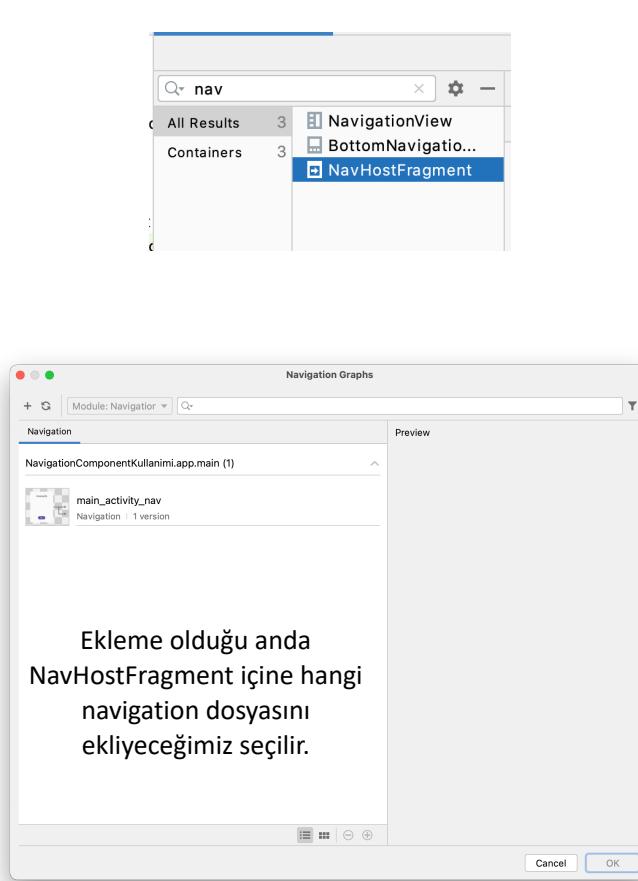
bottom_bar_menu.xml



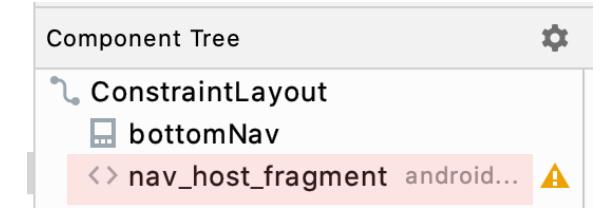
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_activity_nav"
    app:startDestination="@+id/birinciFragment">
    <fragment
        android:id="@+id/ikinciFragment"
        android:name="com.example.navigationcomponentbottomnav.IkinciFragment"
        android:label="IkinciFragment"
        tools:layout="@layout/fragment_ikinci"/>
    <fragment
        android:id="@+id/birinciFragment"
        android:name="com.example.navigationcomponentbottomnav.BirinciFragment"
        android:label="BirinciFragment"
        tools:layout="@layout/fragment_birinci" />
</navigation>
```

Ana Tasarım Oluşturma

- Bottom navigation tasarıma yerleştirilir ve menu bottom navigation'a eklenir.



Navigation yapısını çalıştırabilmek için NavHostFragment eklenir.



Oluşturduğumuz menu eklenir.

Activity Kodlaması

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)                                NavHostFragment id  
        val navHostFragment = supportFragmentManager.findFragmentById(R.id.nav_host_fragment) as NavHostFragment  
        NavigationUI.setupWithNavController(binding.bottomNav, navHostFragment.navController)  
    }                                                        NavHostFragment ile Bottom Navigation birleştiriliyor.  
}                                                        bottomNav, Bottom Navigation idsidir.
```

Navigation Component

ile

Navigation Drawer Kullanımı

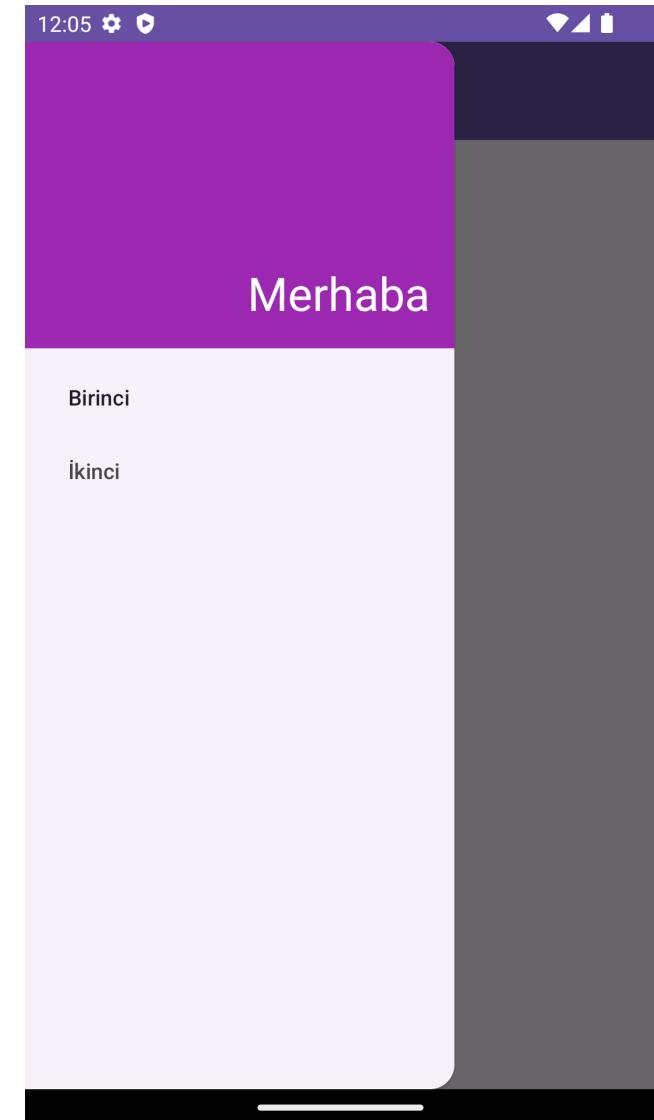
Kurulum

```
dependencies {  
  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.2.0'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.3.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    //Navigation Component  
    implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
    implementation "androidx.navigation:navigation-ui-ktx:2.3.3"  
}
```

```
implementation "androidx.navigation:navigation-fragment-ktx:2.3.3"  
implementation "androidx.navigation:navigation-ui-ktx:2.3.3"
```

Navigation Drawer

- Elimizin kaydırma hareketiyle ortaya çıkan bir menüdür.
- Menü item'larına tıklanıldığında **Fragmentlar** açılmaktadır.
- Varsayılan **ToolBar'ı** kaldırarak , kendi özelleştirilmiş ToolBar'ımızı eklememiz gereklidir.
- Toolbar üzerindeki buttona tıklayarak navigation drawer açılır.



Fragmet Tasarımı Oluşturma

- Fragment'lar için bir **sınıf** ve bir adet **Layout** oluşturuyoruz.

```
<?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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BİR"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

ADALAN



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
```

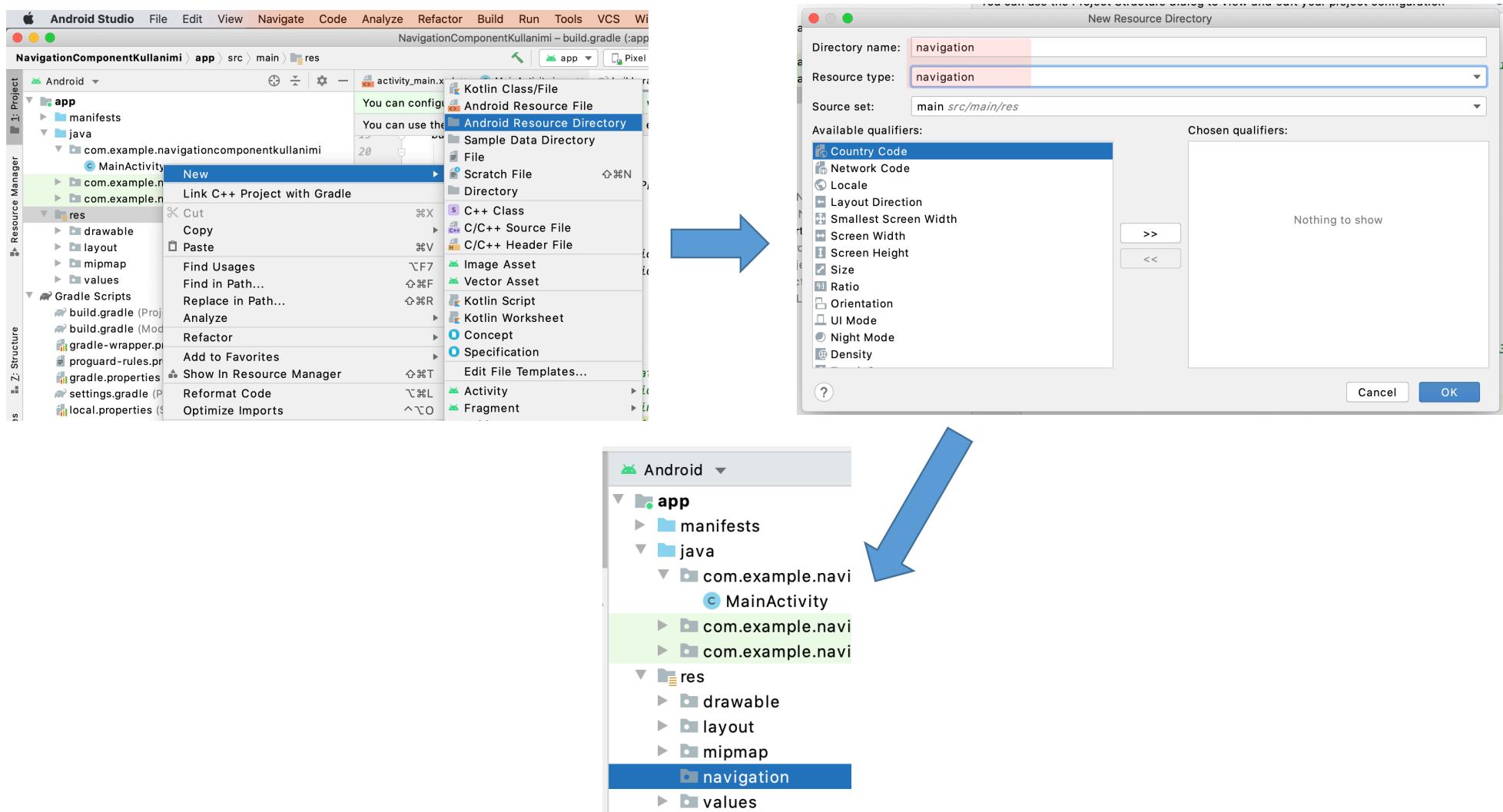
}

2

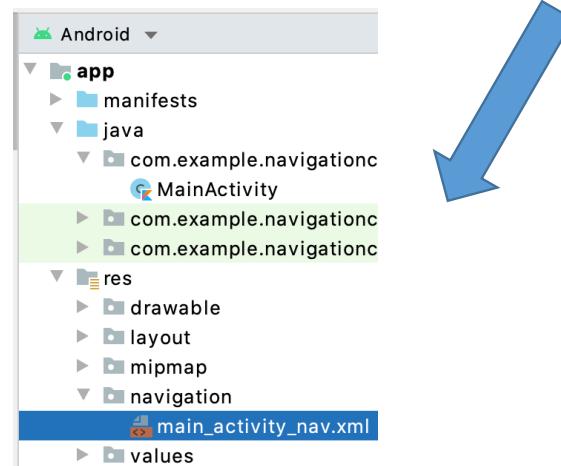
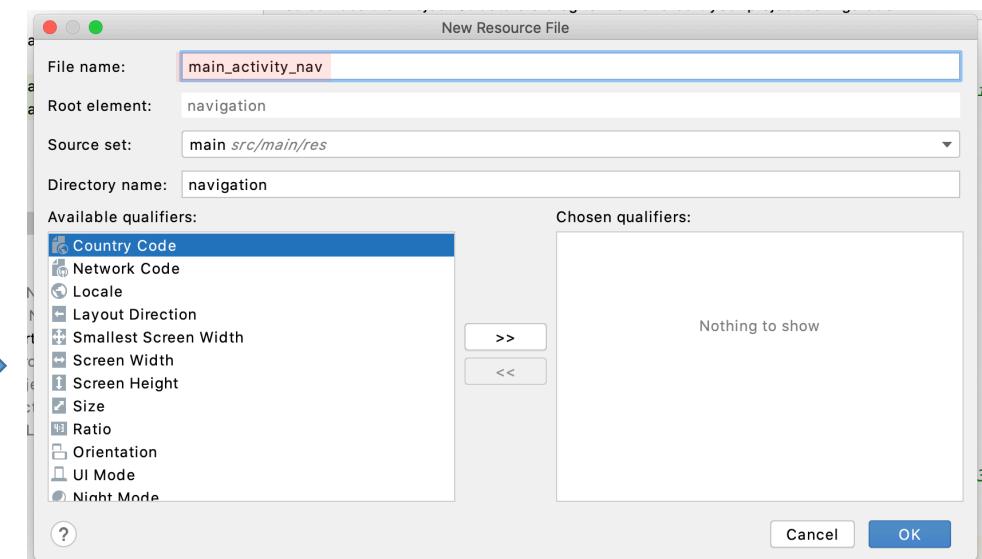
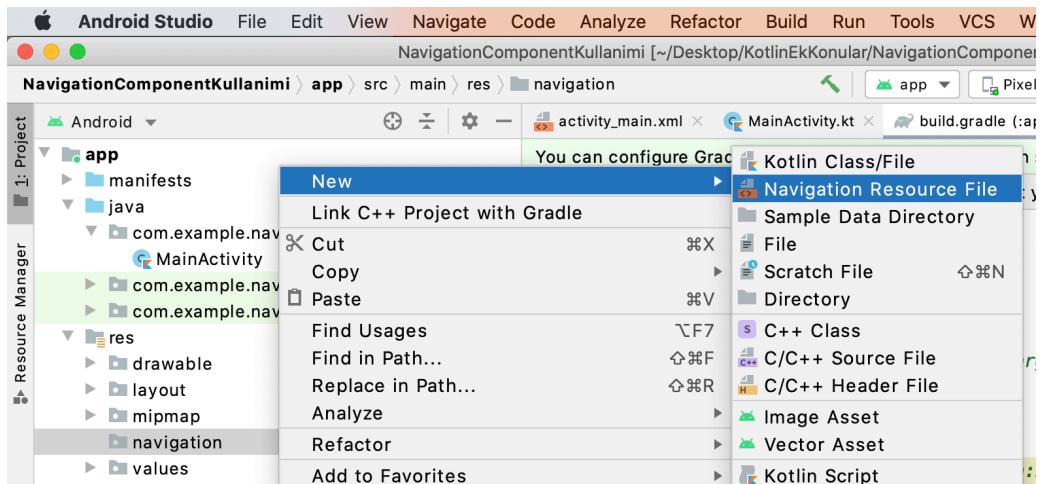
3

Tasarım isminin sonuna Binding eklenerek sınıf oluşturulur.

Navigation Klasörü Oluşturma

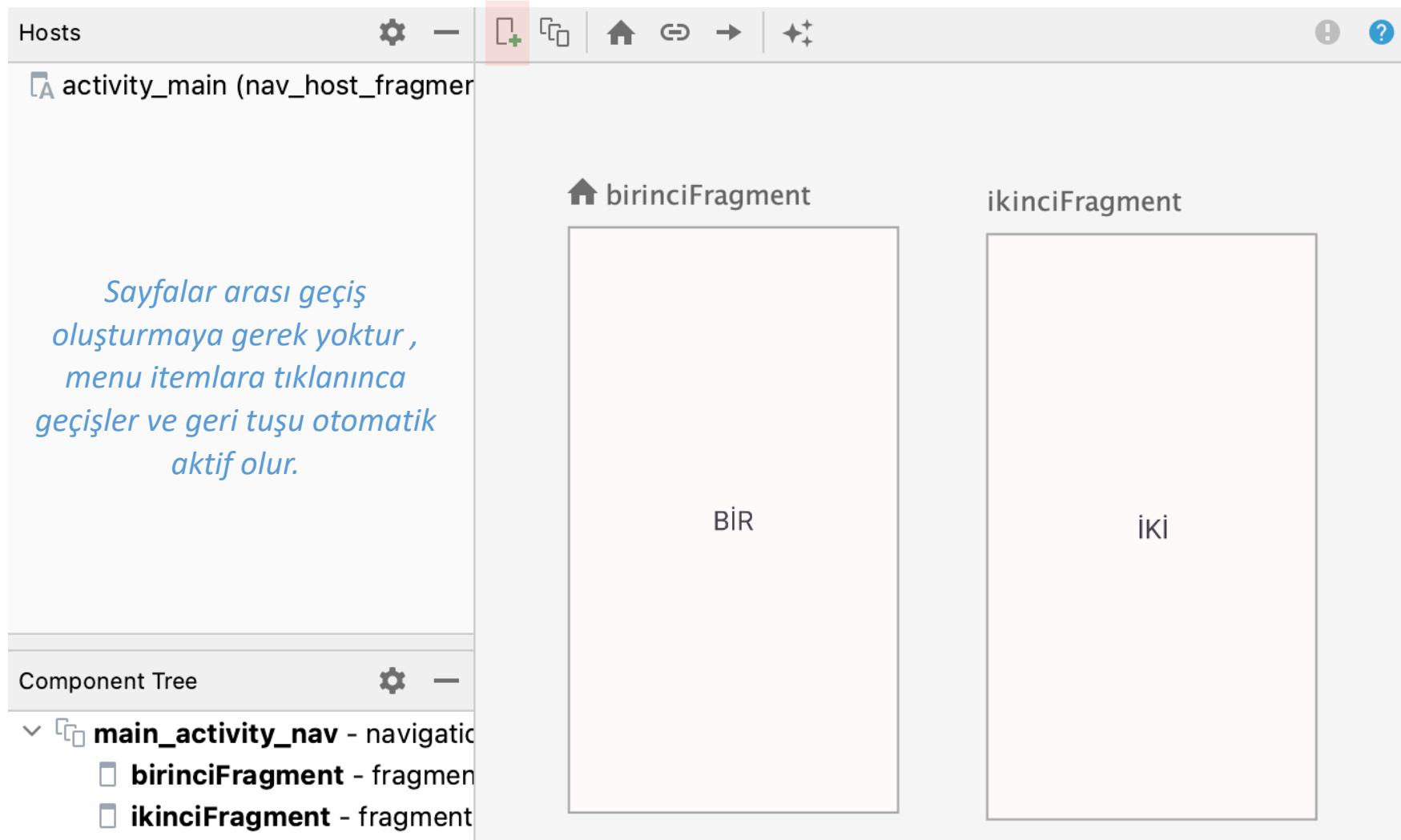


Navigation Dosyası Oluşturma



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

Navigation Component'e Fragment Ekleme



Navigation View için Menu Oluşturma

1

- Bottom navigation için menu oluşturulmalıdır.

- Menu item id ile navigation dosyasındaki fragment id aynı olmalıdır.

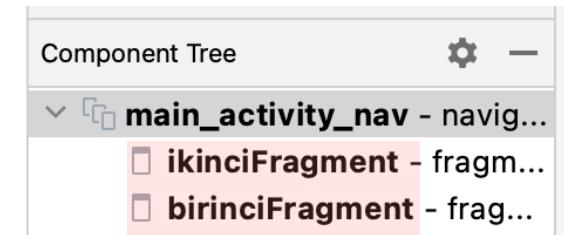
- Bottom navigation menu itemine tıklanınca ilgili id'li fragment açılacaktır.

- Örnek : Üzerinde Bir yazan item tıklanırsa birinciFragment id'li fragment açılacaktır.

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

    <item
        android:id="@+id/birinciFragment"
        android:icon="@drawable/resim_bir"
        android:title="Bir" />
    <item
        android:id="@+id/ikinciFragment"
        android:icon="@drawable/resim_iki"
        android:title="İki" />
</menu>
```

nav_menu.xml



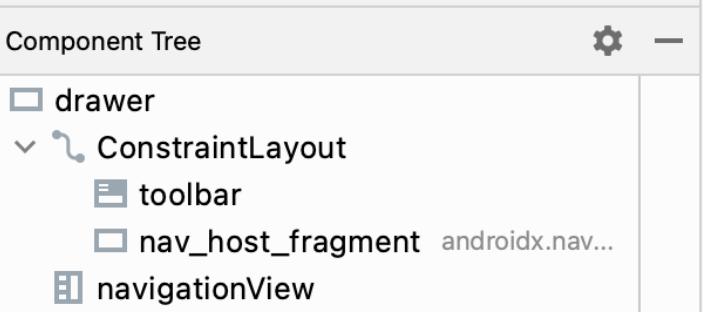
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_activity_nav"
    app:startDestination="@+id/birinciFragment">
    <fragment
        android:id="@+id/ikinciFragment"
        android:name="com.example.navigationcomponentnavigationdrawer.IkinciFragment"
        android:label="IkinciFragment"
        tools:layout="@layout/fragment_ikinci"/>
    <fragment
        android:id="@+id/birinciFragment"
        android:name="com.example.navigationcomponentnavigationdrawer.BirinciFragment"
        android:label="BirinciFragment"
        tools:layout="@layout/fragment_birinci" />
</navigation>
```

Navigation Drawer

Sayfa İçeriğinin Oluşturulması

Navigation Drawer

Tasarım içinde navigation açılmasını test için kullanılır.



Navigation Itemlarına tıklandıkça açılacak fragmentları gösterecek yapıdır.

Nav Host Fragment aynı zamanda navigation component alt yapısını buraya aktarır.

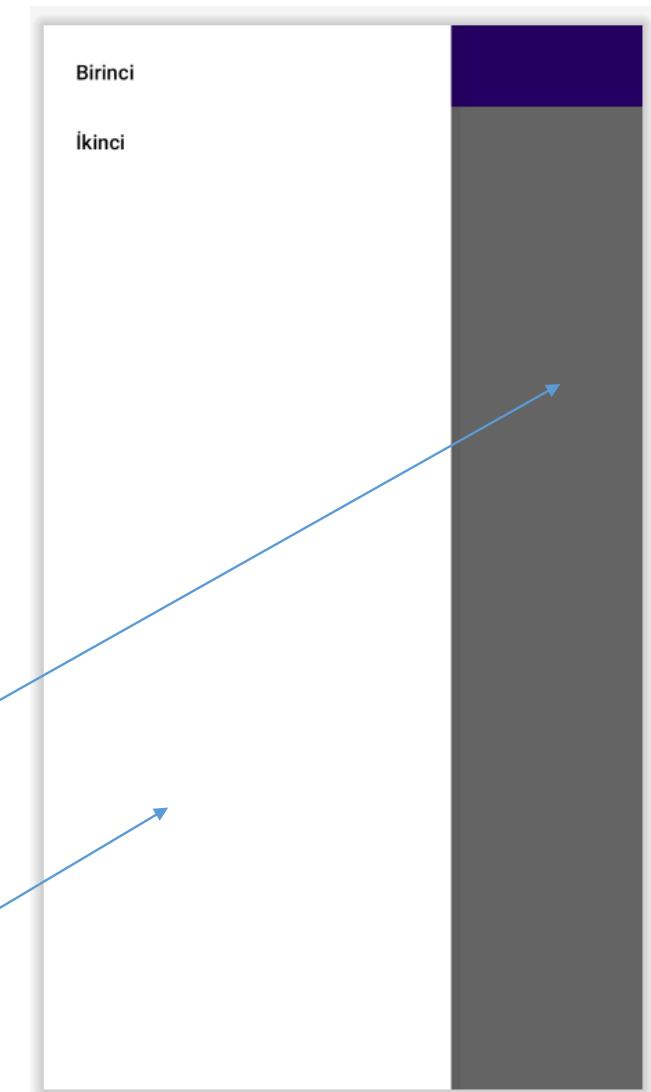
Menu ekleme

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    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:id="@+id/drawer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:openDrawer="start"
    tools:context=".MainActivity">

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

        <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" />
        <androidx.fragment.app.FragmentContainerView
            android:id="@+id/nav_host_fragment"
            android:name="androidx.navigation.fragment.NavHostFragment"
            android:layout_width="0dp"
            android:layout_height="0dp"
            app:defaultNavHost="true"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/toolbar"
            app:navGraph="@navigation/main_activity_nav" />
    </androidx.constraintlayout.widget.ConstraintLayout>

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/navigationView"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:menu="@menu/nav_menu" />
</androidx.drawerlayout.widget.DrawerLayout>
```



Navigation Drawer'ın Açılması ve Item Tıklama

3

Bu aşamada navigation drawer toolbar üzerindeki butona basılarak veya elimiz ile kaydırarak açıldığını test edebiliriz.

```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)                                     Navigation component ile Navigation Drawer bağlantısı  
        val navHostFragment = supportFragmentManager.findFragmentById(R.id.nav_host_fragment) as NavHostFragment  
        NavigationUI.setupWithNavController(binding.navigationView, navHostFragment.navController)  
  
        binding.toolbar.title = "Başlık"  
  
        val toggle =  
            ActionBarDrawerToggle(this, binding.drawer, binding.toolbar, 0, 0)      Toolbar üzerindeki menu iconu ile Drawer'ı açma  
        binding.drawer.addDrawerListener(toggle)  
        toggle.syncState()  
  
        val baslik = binding.navigationView.inflateHeaderView(R.layout.navigation_baslik)  
        val textViewBaslik = baslik.findViewById(R.id.textViewBaslik) as TextView  
        textViewBaslik.text = "Merhaba"  
    }  
}
```

Navigation Drawer'ın Geri Tuşu İle Kapanması

- Daha iyi bir kullanıcı deneyimi için
Navigation Drawer **açık** ise geri tuşuna basılınlca kapanması ,
kapalı ise uygulamadan çıkışmasını sağlamamız gereklidir.

4

```
@Deprecated("Deprecated in Java")
override fun onBackPressed() {
    if(binding.drawer.isDrawerOpen(GravityCompat.START)){
        binding.drawer.closeDrawer(GravityCompat.START)
    }else{
        onBackPressedDispatcher.onBackPressed()
    }
}
```

Not : Fragmentlar açılmışsa geri tuşu önce bir önceki fragmentta dönmek için çalışır.

Örn : 2. Fragment açıksa önce 1. fragmenta geri döner sonra kapatır.

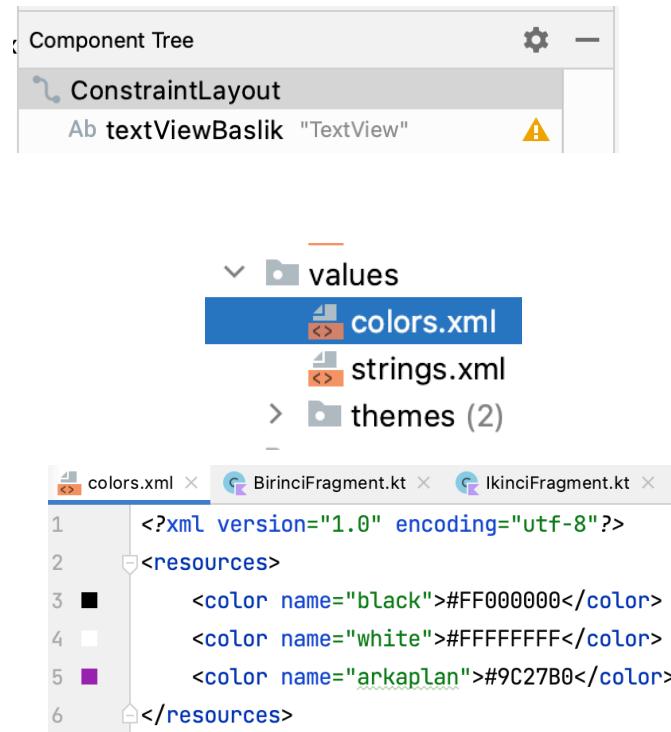
Başlık Oluşturma

- Navigation view üzerinde görünecek başlık ekleyebiliriz.
- Ayrı bir tasarım olarak hazırlanır.
- Kodlama yoluyla navigation üzerinde aktif hale getirilir.

navigation_baslik.xml

```
<?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:background="@color/arkaplan"
    android:layout_height="200dp">

    <TextView
        android:id="@+id/textViewBaslik"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:text="TextView"
        android:textColor="@color/white"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



Başlığı Kodlama ile Ekleme



Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan