Mobil Programlama

Navigation Ekleme

- Sayfalar arasında daha etkin dolanabilmek için projemize res klasöründe navigation klasörü ile navigation dosyasını oluşturarak kurulumu yapmamız gerekmektedir.
- Öncelikle build.grandle(Module) dosyasında dependencies kısmına aşağıdaki kod ile kurulum yapılmalıdır.

```
implementation ("androidx.navigation:navigation-fragment:2.5.0")
implementation ("androidx.navigation:navigation-ui:2.5.0")
```

- Eğer güncelleme ve versiyon kaynaklı hata çıkarsa grandle (module) dosyasında navigation versiyonları 2.5.0'a düşürülebilir.

Fragment Yapısı

- Fragment , Activity'e bağlı olan kullanıcı arayüzünün bir parçasıdır.
- Fragment'ların kendilerine özgü bir yaşam döngüleri (lifecycle) ve UI bileşenleri vardır.
- Fragment'lar kullanıcı arayüzünü (UI) tasarımını zenginleştirmeye, ekranlar arasında veri aktarmaya ve farklı cihaz boyutlarına uyum sağlamaya yardımcı olur.
- TabLayout, BottomNavigation veya NavigationView gibi özellikler kullanılacağında yüksek verimi ve uyum sağlaması sebebiyle tercih edilir.
- Geri tuşu işlemini çok akıllı bir şekilde gerçekleştirir.
- Sürekli yeni bir Activity başlatmak yerine Fragment ile tek bir Activity içerisinde birden fazla sayfa kullanabilirsiniz.
- Bir Fragment birden fazla Activity içerisinde kullanılabilir.

- New ile Fragmentler oluşturulur. Oluşturulan sayfada onCreateView hariç diğer kodlar kaldırılabilir. ViewBinding işlemi uygulanarak gerekli sayfalar hazırlanabilir. ViewBinding işleminde;
  - **bndg**=FragmentIlkBinding.*inflate*(inflater,container,**false**);

    return bndg.getRoot();

- Oluşturulan fragment'ların xml kısmında FrameLayout kısmı ConstraitLayout olarak değiştirilir.

- Navigasyon sayfasına eklenen Fragment'ler istenilen şekilde birbirine bağlanarak, bağlantılara istenen id'ler verilir. Bu id'ler butona tıklanınca çağırmak için kullanılır;
  - Navigation.*findNavController*(view).navigate(R.id.*ikinciFragment*);
- Navigation dosyasına Fragment'ler eklendikten sonra layout klasöründeki activity_main_xml sayfasına NavHostFragment komponent'i eklenir.
- Projede sayfalar arası gezinmede geri tuşu da etkin bir şekilde çalışır.
- Geri tuşuna basıldığında bir önceki sayfaya değil de ilk sayfaya gitmesini istiyorsak bağlantının pop behavior özelliği aktif edilir. Bu finish() ile aynı görevi görür.

Uygulama

- Projeniz için içinde bir adet metin ve butonun bulunduğu 3 adet fragment sayfası oluşturarak bunları navigation klasörü içine oluşturduğunuz dosyaya ekleyin. Butona tıklandıkça sayfalar arasında gezinmek için gerekli kodlamayı yapın.

Navigation İşlemi İle Veri Transferi

- Öncelikle build.grandle(Project) dosyasına en üste aşağıdaki kod ile kurulum yapılmalıdır.

```
buildscript{

    repositories{
        google()
    }
    dependencies{
        classpath ("org.jetbrains.kotlin:kotlin-gradle-plugin:1.5.20")
        classpath ("androidx.navigation:navigation-safe-args-gradle-plugin:2.5.0")
    }
}
```

- Daha sonra build.grandle(Module) dosyasına plugins içine aşağıdaki kod eklenmelidir.
  - id ("androidx.navigation.safeargs")

- Hangi sayfaya veri gönderilmek isteniyor ise o fragment seçilerek Arguments kısmından değişkenler belirlenir.
- Daha sonra hangi sayfadan veri göndermek istiyorsak o sayfanın Activity sayfasında gerekli kodlama yapılır.

Uygulama

- Projeniz için oluşturulmuş 2 adet fragment sayfasında veri aktarımı yapmanız istenmektedir. 1. Fragment'den ikinci fragment'a string, int, double verileri butona tıklanınca aktaran uygulamayı yazınız.

- Not: Yeni bir proje açtıysanız Navigation kurulumunu unutmayınız.

Uygulama

- Projeniz için Müşteri isimli bir sınıf oluşturun. Bu sınıfın değişkenleri, yapılandırıcıları ve getter-setter metotları olsun. Bu sınıftan nesne türeterek değerleri 2. Fragment'a yazdıran uygulamayı yazınız.

- Not: Bunun için navigation için oluşturduğunuz xml sayfasında verinin aktarılacağı fragment'ı seçip argument kısmına aynı değişken ekler gibi nesne eklemeniz gerekmektedir.

Bottom Navigation Ekleme

- Fragment'lar arasında gezinmek için uygulamamıza bottom navigation ekleyebiliriz.
- Bu işlem için res klasörü içine menü klasörü oluşturmamız ve içerisine menu resource file eklenir.
- Oluşan sayfaya eklenen menu item'lara istenen icon seçilir.
- Fragment'lar arasında dolaşabilmek için menu item'ların id si ile dolaşılmak istenen fragment'ın id si aynı olmak zorundadır.
- Menu oluşturulduktan sonra activity main sayfasına bottomNavigationView olarak eklememiz gerekmektedir. Eklenen bottomNavigationView'a common attributes'dan oluşturulan menu eklenir.
- Daha sonra MainActivity dosyasında gerekli kodlamalar yapılır.

Uygulama

- Projenizde iki item menu olacak sekilde bir bottom menü oluşturunuz. Daha sonra bu menülere tıklanıldığında 2. Ve 3. Fragment'ı açan uygulamayı yazınız.

Navigation Drawer

- Kaydırma hareketiyle ortaya çıkan menüdür.
- Bu menüyü oluşturmak için activity_main.xml arayüzünü constraint değil drawer yapmak gerekmektedir.
  - <androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
- Daha sonra component kısmından bu drawerLayot'a constraintLayout ve bunun altına da toolbar ve navHostFragment eklenir.
- drawerLayout kısmına NavigationView eklenerek id verilip özelliklerden oluşturulan menu eklenir, layout_gravity start seçilir
- Yapının açılıp kapanabilmesi için özelliklerden drawerLayout'un openDrawer özelliği start seçilir ve kodlamalar yapılır.

Uygulama

- Projenizde 3 çizgiye basınca açılan bir drawer menu olacak sekilde bir menü oluşturunuz. Daha sonra bu menülere tıklanıldığında 2. Ve 3. Fragment'ı açan uygulamayı yazınız.

KAYNAKÇA

- https://medium.com/@sezerozaltunn/