

**Bölüm Adı: Elektrik ve Otomasyon**

**Program Adı: Mobil Teknolojileri**

**Ders Adı: Android Programlamaya Giriş**

**Proje Adı: Sözlük Uygulaması**

**19MY93004**

**Bilal Mercan**

İçindekiler

[Android Studio Kavramları 4](#_Toc38013613)

[Android 4](#_Toc38013614)

[Kotlin: 4](#_Toc38013615)

[Drawable 4](#_Toc38013616)

[Layout 4](#_Toc38013617)

[Menu 4](#_Toc38013618)

[Mipmap 4](#_Toc38013619)

[Values 4](#_Toc38013620)

[Gradle Scripts 4](#_Toc38013621)

[Sdk 4](#_Toc38013622)

[Sanal 4](#_Toc38013623)

[Gradle 5](#_Toc38013624)

[Android 5](#_Toc38013625)

[Main activity 5](#_Toc38013626)

[Manifest dosyası 5](#_Toc38013627)

[String 5](#_Toc38013628)

[Action bar 5](#_Toc38013629)

[Uygulama Geliştirirken Kullandığım İçerikler 6](#_Toc38013630)

[TextView 6](#_Toc38013631)

[Plain text 6](#_Toc38013632)

[Button 6](#_Toc38013633)

[İmage View 6](#_Toc38013634)

[Check Box 6](#_Toc38013635)

[Ratebutton 6](#_Toc38013636)

[Kotlin 7](#_Toc38013637)

[Kotlin Neden Geliştirildi 7](#_Toc38013638)

[Kotlin’in Desteklediği Platformlar 7](#_Toc38013639)

[Kotlin ve Java Farkları 8](#_Toc38013640)

[Sınıf (Class) 9](#_Toc38013641)

[Nesne (Object) 9](#_Toc38013642)

[Kalıtım (Inheritance) 9](#_Toc38013643)

[Constructors (Kurucu) 10](#_Toc38013644)

[Uygulama Çalışma Mantığı 10](#_Toc38013645)

[Android Studio Açılış Ekranı 11](#_Toc38013646)

[Sayfa Oluşturma 12](#_Toc38013647)

[Buton Oluşturma 12](#_Toc38013648)

[Buton yerleşimini ayarlama 13](#_Toc38013649)

[Layout 14](#_Toc38013650)

[**İmage View** 15](#_Toc38013651)

[**Edit text** 17](#_Toc38013652)

[EditText’ e girilen isimin diğer sayfaya mesaj ile çağırılması 18](#_Toc38013653)

[Rating Bar 19](#_Toc38013654)

[**Check Box** 20](#_Toc38013655)

[Check Box İşaretlendiğinde 21](#_Toc38013656)

[Sayfa Geçişleri 22](#_Toc38013657)

[Bildirim Mesajı 22](#_Toc38013658)

[Tanımlar 23](#_Toc38013659)

[OnBackPressed () Metodu 23](#_Toc38013660)

[Not Kaydetme 24](#_Toc38013661)

[Kaydet butonuna basıldığına Kaydetme 24](#_Toc38013662)

[Kaydedilen Not Dosyasını Getirme 25](#_Toc38013663)

[Kelime Listesini Okuma İşlemi 26](#_Toc38013664)

[Aranılan Kelimelerin Karşılığını bulmak 27](#_Toc38013665)

[String 28](#_Toc38013666)

[String Tanımlamaları 28](#_Toc38013667)

[**String dosyası** 29](#_Toc38013668)

[Buton ve Sayfa Tasarımı 30](#_Toc38013669)

[İkon 32](#_Toc38013670)

[Manifest Dosyası 33](#_Toc38013671)

[Sonuç 34](#_Toc38013672)

[Uygulama Konusu 34](#_Toc38013673)

[Kaynakça 35](#_Toc38013674)

# **Android Studio Kavramları**

Android; Google, Open Handset Alliance ve özgür yazılım topluluğu tarafından geliştirilen, Linux tabanlı, mobil cihaz ve cep telefonları için geliştirilmekte olan, açık kaynak kodlu bir mobil işletim sistemidir. Desteklenen uygulama uzantısı “.apk” ’dır.

Kotlin: Kotlin programlama dili, Rusya merkezli yazılım şirketi, Jetbrains tarafından desteklenen Java sanal makinesi (JVM) üzerinde çalışan ve ayrıca Javascript kaynak koduna derlenebilir statik tipli bir programlama dilidir.

Drawable: Resim ve Tasarım dosyaları burada bulundurulur. Dosyaları PNG ya da JPEG formatında olabilir.

Layout: Bu kısım tasarım dosyalarının olduğu kısımdır. XML dili ile yazılmıştır. Buradaki tasarım dosyalarını javada Oncreate metodun'da çağırılır.

Menu: Menu kısmını burada aktif edilebilir.

Mipmap: Uygulamanın açılış logolarını içerir.

Values: Uygulamada kullanılan sabit değişkenler burada saklanabilir. strings.xml dosyası uygulamada kullanılan ve ekranlarda kullanıcıya gösterilen her türlü metni saklar.

Gradle Scripts: Gradle, projeleri inşa etmeye yarayan yeni nesil bir sistemdir. Gradle en güzel özelliği test, debug, release esnasında almanız gereken proje paketlerini otomatikleştirmektedir.

Sdk: **Android** işletim sisteminde yazılım geliştirme kitine verilen kısaltmadır. Akıllı cihazlarda uygulama tasarlamak için zorunluluğu bulunan bir yazılım kitidir.

Sanal Makine: Android uygulamalar geliştirirken uygulamaları test etmek için cihazları kullanmak pratik bir çözüm olmayabilir. Her yapılan kod değişikliğinden sonra uygulamayı yeniden derleyip cihaza yüklemek hem zaman kaybına hem de cihazın yazılım hatalarından dolayı zarar görmesine sebep olabilir. Bu yüzden test işlemlerini daha hızlı gerçekleştirmek için Android SDK bize sanal cihaz (AVD) oluşturarak gerçek cihazlarda alınacak tepkiyi, cihaza yüklemeden bilgisayarda görüntülemeye yarar.

Gradle:Gradle Android uygulaması geliştirme aşamalarını otomatik olarak uygulayan bir build sistemi olup test, debug ve release işlemlerini daha kolay hale getirmektedir. Android için tercih edilmesinin sebebi de içinde xml, resim ve font dosyaları gibi Android projesinde gereken diğer bileşenleri barındıran paketlerin oluşturulabilmesidir.

Android Plugin:Android derleme sistemi olarak Gradle’ ı temel alır. Android Plugin ise Gradle’ a Android uygulamalarını oluşturmaya özel çeşitli özellikler ekler. Android Studio’ yu güncellediğinizde, Android Plugin’ i en son kullanılabilir sürüme otomatik olarak güncellemek için bir uyarı alabilirsiniz. Güncellemeyi kabul etmeyi seçebilir veya projenizin yapı gereksinimlerini temel alarak başka bir sürümü manuel olarak belirtilebilir.

## Main activity

**MainActivity,** android uygulama başlatıldığında ilk çalışacak olan sınıftır. Her Activity de **onCreate** methodu içinde **setContentView** olduğu görülür. Bu kısımda Activity’ nin hangi görsel yapıya sahip olduğunu belirlenir. Oluşturulan her bir Activity sınıfının **AndroidManifest.xml** de referansı olmak zorundadır. Yani bir tanımlama yapılması gerekiyor eğer **Android Studio** da **new->Activity** kısmından bir Activity oluşturulduysa otomatik olarak Activity’ nin referansının AndroidManifest.xml de oluştuğunu görebiliriz.

## Manifest dosyası

Manifest dosyası bir uygulamanın en temel bilgilerini içinde bulundurur. Örneğin main activity olarak başlatılacak activity, manifest dosyası sayesinde sağlanır. Uygulamadaki bulunan diğer activityler’ de burada tanımlanmalıdır. Aksi halde uygulama çökecektir.

Manifest dosyası içerisinde uygulamanın paket dosyaları, kütüphaneler, SDK sürümü, activityler, uygulama izinleri gibi uygulama ile ilgili önemli özellikler bulunur

## String

String’ ler birden fazla kullanılacak içerikler için tanım dosyasıdır. Tanımlanan string’ leri daha basit yani kısayol olarak düşünülebilir.

## Action bar

Uygulamalardaki Başlık yazısı, ikon, ikonlar, menü, arkaplan rengi değiştirilebilir, geri butonu, navigator, icon gibi özellikleri barındıran, Toolbar olarak da adlandırılan, uygulamanın en üst kısmında kaldırılmadığı sürece bulunan bir bar. Aynı zamanda ilgili Activity’ nin en üst kısmını belirtir, üst sınırı ifade eder.

# **Uygulama Geliştirirken Kullandığım İçerikler**

TextView: Kullanıcıya kullanılan yazıyı veya metini göstermek için kullanılır

Plain text: Kullanıcının veri girmesini sağlar.

Button: Kullanıcının butona basıldığında hangi aksiyonun gösterilmesi isteniyorsa o amaçla kullanılır.

İmage View: Kullanıcıya gösterilmek istenen resim, fotoğraf vb. gibi görsel öğelerin gösterilmesi için kullanılır.

Check Box: Kullanıcının onaylandı veya işaretlendi simgesini belirtmek için kullanılır.

Ratebutton: Kullanıcının uygulamayı veya bir durumu değerlendirmesi için kullanılır.

# **Kotlin**

Kotlin, 2010 yılında Rusya merkezli JetBrains firması tarafından ilk kez gündeme getirildi. Kotlin, 19 Temmuz 2011 yılında JVM Language Summit etkinliğinde duyuruldu. Kotlin, statik bir programlama dilidir. Kotlin, Apache 2.0 lisansı altında geliştirilen açık kaynak kodlu bir projedir. Projenin kaynak kodları herkese açıktır. Projeye destek olmak adına geliştirmeler yapabilirsiniz. Kotlin’ in ismi ise Rusya’da bulunan Kotlin adasından gelmektedir. Kotlin dilini öne çıkaran en önemli etken ise: **Google Şirketinin Android Developer bölümünün Kotlin diline güvenmeleri ve Android uygulamaları geliştirmek için bu dili desteklemeleridir**. 2017 yılında Google, Android Studio platformunda destek vereceğini açıklamıştır. Kotlin, mevcut durumdaki Java kütüphanelerini kullanarak uygulama geliştirmeye olanak sağlar ve Java ile birlikte çalışmaktadır. Kotlin, nesneye yönelimli (object oriented) fonksiyonel bir dildir. Java, C# ve C++ gibi nesne yönelimli bir programlama dilidir.

## Kotlin Neden Geliştirildi

Cevap olarak JetBrains firması; **“Performans ve güvenliği feda etmeden Java’dan daha özlü kodlar yazmayı sağlamak için bu dili geliştirdik”** cevabını vermektedir. Kotlin’in temel hedefi kod satırlarını azaltmak, daha güvenli kod yazmak olarak söyleyebiliriz.

# **Kotlin’in Desteklediği Platformlar**

Kotlin programlama dili ile 4 ana platformda ya da alanda çeşitli uygulamalar geliştirebilirsiniz. Aşağıda maddeler halinde geliştirme alanları listelenmiştir.

**JVM:** Server-side Uygulamalar

**Android**: Android Uygulamalar

**Browser**: JavaScript tabanlı Web Uygulamalar

**Native**: Macos, İos ve Gömülü sistemler Uygulamaları. (Geliştirilme aşamasındadır**.**)

# **Kotlin ve Java Farkları**



Görüldüğü üzere Kotlin’ de tanımlar java’ ya kıyasla daha kısa ve pratik bir şekilde yapılmasını sağlıyor.

-Null Safety;

Kotlin’ de NullPointerExceptions gibi bir exception dan kurtulmuş olduk. Bunun nedeni, Kotlin dili ile programlama yaparken varsayılan olarak hiçbir değişkene null değerini atayamazsınız ve bir metodu null değer döndüremediğiniz içindir.

-Data Classes;

Data classes özelliği harika bir zamandan tasarruf sağlar. Uygulamalarımızın çoğunun veri odaklı olduğunu göz önüne aldığımızda, sıklıkla kendimizi yalnızca veri saklamak için özellikler ve alanlar içeren sınıflar oluştururken kendimizi buluruz. Java’da, bu çok sıkıcıdır. Çünkü her kullanacağımız özellik için bir get / set yöntemi belirlememiz gerekmektedir. Fakat Kotlin kullanırsak, yazacağımız sınıf ve özelliklerini tek bir satırda bildirme imkânımız bulunmaktadır.

-Üst Düzey Fonksiyonlar ve Lambda’ lar;

Önemli olan şey, Kotlin’in işlevsel bir programlama dili olmasıdır. Temel olarak Kotlin, daha yüksek seviyeli fonksiyonlar, lambda ifadeleri, operatör aşırı yüklenmesi, tembel değerlendirme, operatör aşırı yüklenmesi ve çok daha fazlasını içeren birçok kullanışlı yöntemden oluşur.

## **Sınıf (Class)**

Sınıflar nesne özelliklerine sahip değişken ve metodların bulunduğu yapılardır. Sınıflar nesne tabanlı programlama kullanan her programlama dilinde kullanılabilir. Birçok programlama dili, sınıflara ev sahipliği yapar. Java, C++, Swift, PHP, farklı tabanlarda farklı işlemlerde ve farklı yazılım lar da yazılsalar da sınıflar çoğu programlama dilinde kullanılır. Sınıf ile nesne aynı yapılar değildir. Sınıf nesnenin özelliklerini tutar ve soyut bir yapıdır. Sınıf ve nesne birbirine bağlı kavramlardır. Çünkü her nesnenin bir sınıfı olmak zorundadır.

## **Nesne (Object)**

Sınıf ile nesne arasında bir ilişki bulunduğunu yukarıda belirtmiştik. Sınıflar soyut ifadelerdir. Sabun fabrikasında yeni bir sabun tasarımı üzerinde çalıştığımızı hayal edelim; ortaya yeni bir kalıp çıkarttık… Artık son aşama olan üretime geçmek istiyoruz. Bu kalıp içerisinde sabun nesnelerinin hangi boyutlarda, hangi renkte olacağı, nasıl kokacağı vs. gibi bilgilerin hepsi bizim tarafımızdan belirlenmiş durumda olacaktır. Üretim aşamasına geçildiğinde hep aynı sabun kalıbını kullanılarak yeni sabun nesneleri üretmemiz mümkün olacaktır. Buradaki önemli nokta, sabun kalıbı tasarımı bir kez yapılmış olmasıdır; ancak, bu kalıp ile N tane sabun nesnesi üretilebilmektedir. Buradan yola çıkılarak sabun kalıbını sınıfa, sabunlarsa nesnelere benzetilebilir.

## **Kalıtım (Inheritance)**

Kalıtım gündelik hayattaki kullanımdan esinlenerek oluşturulmuş bir yapıdır. Nesneler kalıtım özelliğini kullanarak özelliklerini başka nesnelere aktarabilirler. Kalıtım kullanılarak oluşturulan bir sınıftan yeni sınıflar türetilebilir. Kotlin dilinde tüm sınıflar Any (Herhangi bir tür olabilir anlamına gelen bir sözcük öbeğidir) sınıfından türetilerek elde edilmiştir. Kalıtım kullanılarak bir class’ ı başka bir class’ ın alt class’ ı yapılabilir. Ve bu sınıftan üretilen nesnenin üst class’ ında ki özellikleri metodları, değişkenleri kullanabilir. Ve ekstra olarak kendi içerisinde de class’ ına özel değişkenler ve metotlar tanımlayabilir.

## Constructors (Kurucu)

Bir constructor, sınıftan üretilmiş nesnelerin değişkelerinin değerlerinin belirlemek için kullanılan özel bir metottur. Sınıftan bir nesne ürettiğimizde çağrılır.

Her sınıfın bir kurucu metoda(constructor) sahip olması gerekir. Sınıfın içinde herhangi bir constructor belirtmezsek, derleyici bunu bizim için otomatik yapar ve sınıfın içine default constructor ekler.

Kotlin’ de iki tür constructor vardır;

-Primary Constructor

-Secondary Constructor

Bir Kotlin sınıfı bir primary constructor’ a ya da bir veya daha fazla ikincil kurucuya sahip olabilir.

# Uygulama Çalışma Mantığı

-Kullanıcı ana ekranda kullanıcı adını girer. (İsteğe bağlı)

-Hakkında sayfasında uygulama ile ilgili bilgileri görebilir, uygulamaya puan verebilir. (İsteğe bağlı)

-Başla butonuna basıldığında sözleşme metnini okudum onaylıyorum butonunu işaretler, aksi halde ilerleyemez.

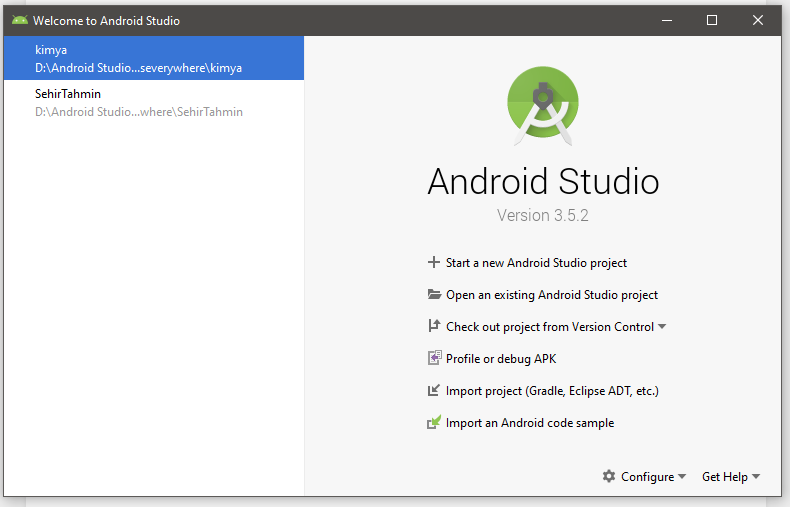
-Uygulama ile ilgili bilgi verilir.

-Kullanıcının kaydetmek istediği notlar varsa dosya adı ile birlikte kaydeder.

-Türkçe-İngilizce sözcük editText’ e girilir ve arama butonuna tıklanır. Aranılan sözcük Textview’ de gösterilir.

-İngilizce-Türkçe sözcük editText’ e girilir ve arama butonuna tıklanır. Aranılan sözcük Textview’ de gösterilir.

# **Android Studio Açılış Ekranı**



**Start a new Android Studio Project:** Yeni bir Android Studio Projeye başlamak için. Biz bu seçenekle devam edeceğiz.

**Open an existing Android Studio Project:**Önceden oluşturulmuş bir Android Studio projesini açmak için.

**Check out project from Version Control:**Git, svn gibi herhangi bir version control sisteminden çekeceksek bu seçenekle devam ediyoruz.

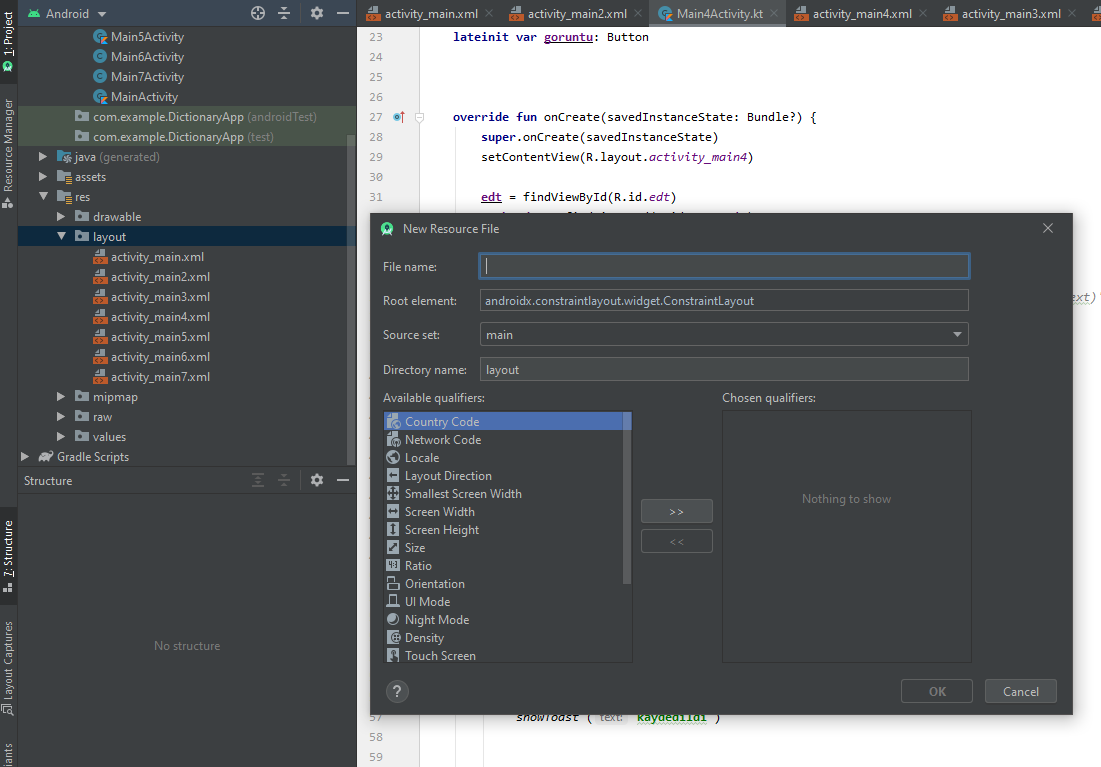
**Profile or debug APK: Profil ya da uygulama içerisinde hata alınınca hatayı düzeltmek için kullanılır.**

**Import project (Eclipse ADT, Gradle, etc...):**Farklı bir ide ya da farklı bir build sistemi ile yazılmış bir projeyi Android Studio projesine çevirerek açmak için. Özellikle önceden yazılmış bir **eclipse projesini açmak** için kullanırız.

**Import an Android code sample:**Android’in bize sunmuş olduğu hazır kodları açmak için.

# **Sayfa Oluşturma**

Android Studio uygulamasında öncelikle sayfa oluşturmak için; App/java/res/layout klasörüne sağ tık ile New seçeneğine giderek Layout resource file seçeneğine tıklayarak oluşturulabilir. Sonrasında ise oluşturulan içeriğin adını ve sayfa yapısını belirtilmesi gerekiyor. Bu işlem tamamlanınca sayfa oluşturulmuş bulunuyor.

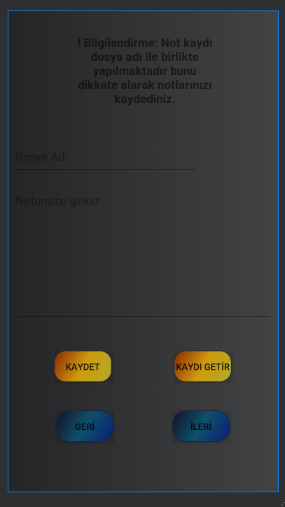


# **Buton Oluşturma**

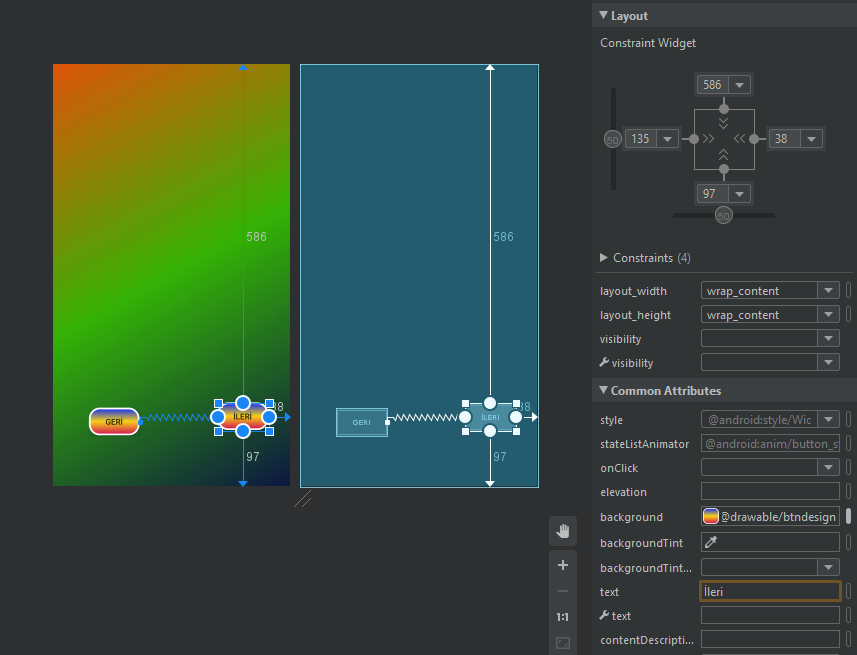
Button oluşturmak için öncelikle Palette sekmesinden buttons seçeneğine tıklayıp button simgesini ekrana doğru sürüklemek yeterlidir. Kullanılan activity main xml sayfasında text bölümünden

Aşağıda görülen buton kodlarının altında bulunan android:text”Başla” satırında çift tırnak arasına istenilen metin buton üzerine yazılabilir.





## Buton yerleşimini ayarlama



# **Layout**

**Linear** **layout**

 LinearLayout ile birlikte tüm android bileşenlerini tek bir konumda kullanabiliriz. orientation özelliğini kullanarak yatay **LinearLayout (horizontal)** ve dikey **LinearLayout (vertical)**olarak da konumlandırabiliriz. Alt alta ya da yan yana aynı çizgi üzerinde bulunacak bileşenleri LinearLayout ile kullanırız.

**Frame** **Layout**

FrameLayout ile birlikte Android bileşenleri üst üste biner. Örnek verecek olursak; iki butonunuz var ve aynı yerde konumlansın istiyorsunuz, biri gözüktüğünde diğeri kaybolsun ve tam tersi koşulda gerçekleşsin istiyorsanız FrameLayout kullanabilirsiniz. Başka bir örnek ise elinizde bulunan bir Listview üzerinde bir butonun her zaman sabit kalmasını istiyorsunuz. Bu durumda FramelLayout kullanabilirsiniz.

**Constraint Layout**

Android Studio 2.2 ile gelen bu layout ile Android bileşenlerini sürükle bırak yaparak responsive bir görüntü elde edebiliriz ayrıca alt alta ve yan yana android bileşenlerini elde etmek için kullandığımız LinearLayout yerine ConstraintLayout kullanarak her cihaz için uyumlu tasarımlar oluşturabiliriz.

**Relative** **Layout**

RelativeLayout ile birlikte Android görsel bileşenlerini istediğimiz yere sürükleyip kullanabiliriz. Bu layout içerisinde yer alan bileşenleri de kendi içerisinde konumlandırabiliriz sağına göre, soluna göre, alt kısma göre, üst kısma göre. Relative Layout ile daha esnek tasarımlar yapabiliriz.

**Table Layout**

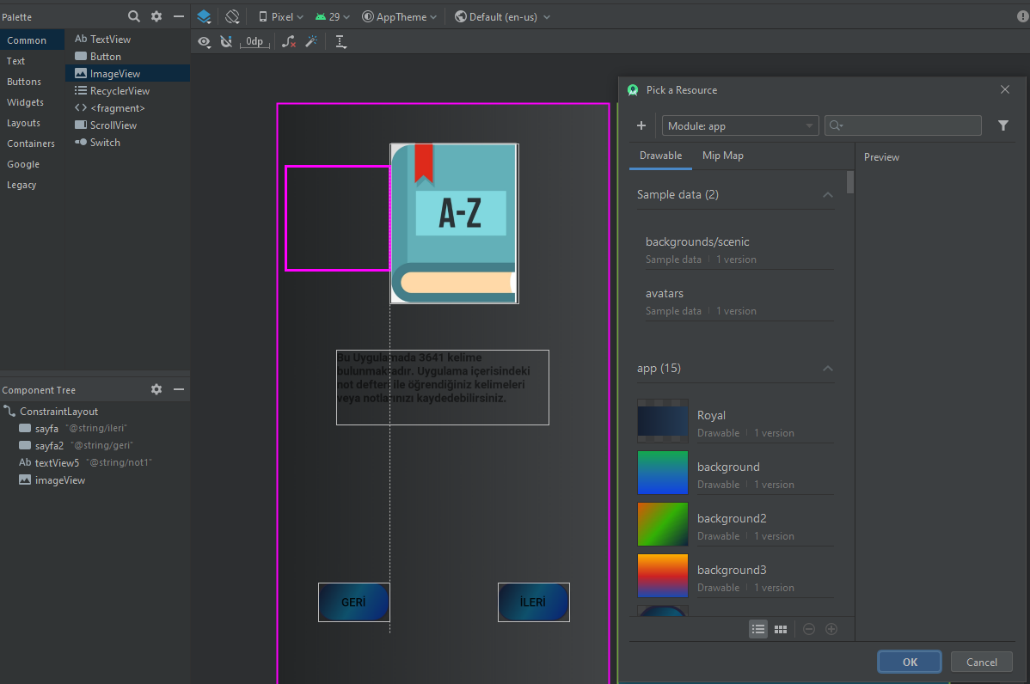
TableLayout ile birlikte Android bileşenlerini bir tablo yapısına yerleştirirsiniz. Html’deki gibi tablo yapısı şeklinde göstermek için kullanılır.

**Frame Layout**

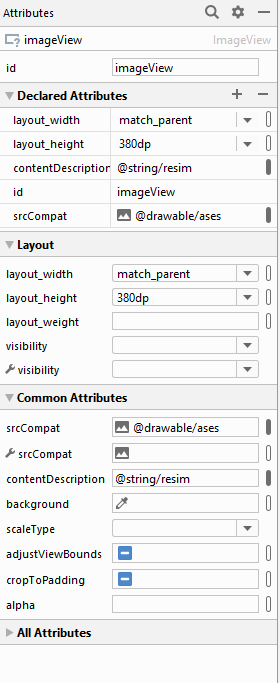
FrameLayout ile birlikte Android bileşenleri üst üste biner. Örnek verecek olursak; iki butonunuz var ve aynı yerde konumlansın istiyorsunuz, biri gözüktüğünde diğeri kaybolsun ve tam tersi koşulda gerçekleşsin istiyorsanız FrameLayout kullanabilirsiniz. Başka bir örnek ise elinizde bulunan bir Listview üzerinde bir butonun her zaman sabit kalmasını istiyorsunuz. Bu durumda FramelLayout kullanabilirsiniz.

## **İmage View**

İmage view uygulama içerisinde görüntülenmekte olan fotoğraf, resim vb. gibi dosyaları eklemek için kullanılan bir içeriktir. İmage view eklemek için palette kısmında bulunan image view seçeneğine tıklayıp sayfaya sürüklememiz gerekiyor. Sonrasında açılan pencereden ekranda görülmesi istenen fotoğraf, resim, tasarım vb. dosyasını seçtikten sonra yüklenmiş oluyor. Ekran yerleşimini de yaptıktan sonra image view düzgün bir biçimde görüntülenebilir.



Android Studio içerisinde bulunan Attributes kısmında ise kısayollar ile eklediğimiz içerikleri daha fazla özelleştirme imkânı sunuyor.



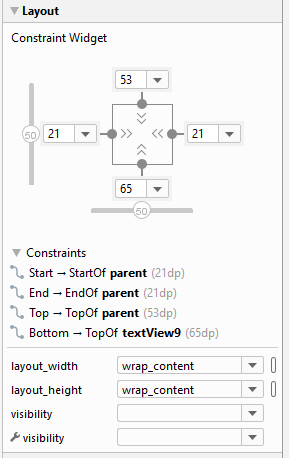
Bu sekme içerisinde layout boyutlarını, image view dosyasını, id dosyaları ve birçok seçenek düzenlenebilir ve değiştirilebilir.

İmage view kod satırında ise yine image view yüksekliğini ve genişliğini, id’sini, yerleşimini, açıklama metnini ve hangi içeriğin eklendiğini görebiliriz ve değiştirebiliriz.



Yine buton örneğinde olduğu gibi constraint layout’lar tanımlanır.

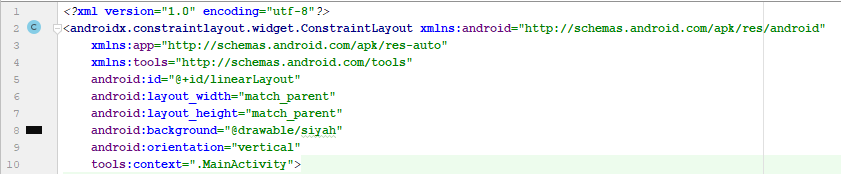
Bu uygulamada layout olarak çoğunluğu constraint layout o yüzden constraint layout’larda yerleşimine dikkat edilmelidir.



BackGround

Background arka plan anlamına gelir. Uygulama içerisinde sayfa için kullanılır genellikle.

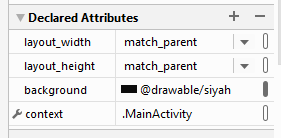
Değiştirmek için main activity dosyasının text kısmından veya design kısmından attributes penceresinden değiştirilebilir ve düzenlenebilir.



Arka plan dosyalarını, tasarımları, drawable klasörünün içerisine yüklenmesi gerekmekte

Başka bir hususta dosya adlarının türkçe olmaması gerekiyor. Aksi halde tanımlamayacaktır ve hata verecektir.

İşlem yapılan Main activity dosyasında Design kısmından Attributes penceresinden de değiştirilebilir.



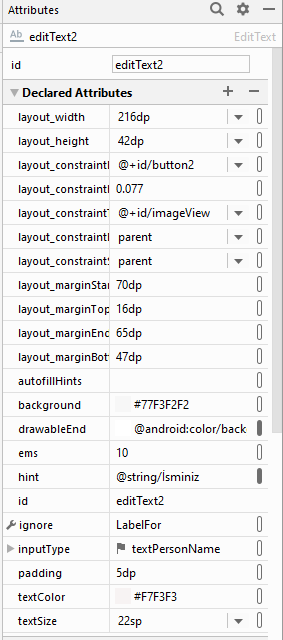
## **Edit text**

**EditText**, **TextView**’ den türeyen kullanıcıdan yazı girmesi beklenen bir android bileşenidir

Bu uygulamada kullanıcı adı girilmesi için kullanılmıştır.

Buton ve image view örneğindeki gibi constraint layout değerleri girilmesi gerekiyor.

String tanımlaması da yapılması gerekiyor.

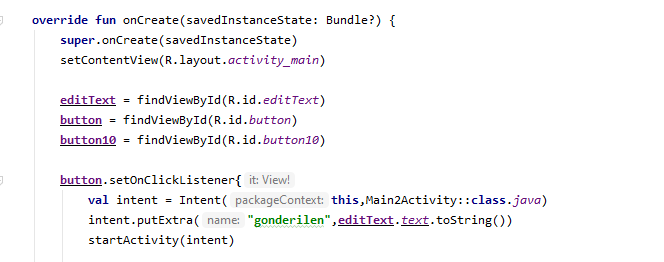
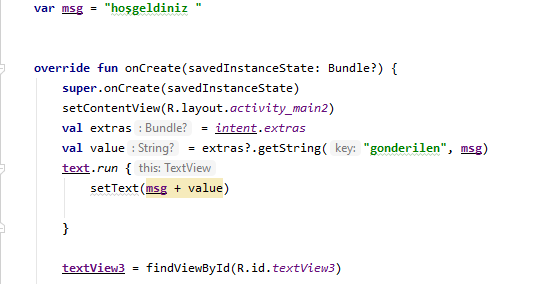


Edit text bileşenini düzenlemek için text kısmından yazı boyutunu, yazı rengini, ekran yerleşimini, id’ sini, boyutunu ayarlayabiliriz. Design sekmesi Attributes penceresinden yukarıda görülen işlemler ve daha fazlası kısayollar ile yapılabilir.

## EditText’ e girilen isimin diğer sayfaya mesaj ile çağırılması

İlgili butona intent nesnesi oluşturup putExtra ile editText’ e girilen değerler gönderiliyor.

Butona xml sayfasında text kısmında onClick özelliği veriliyor. Verilen onClick ismi aşağıdaki örnekteki “gonderilen”



Gönderilen sayfada intent extras tanımlanıyor.

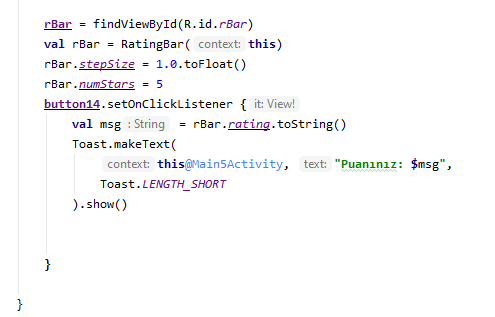
Val value ile getString içerisinde onClick ismi ile çağırılıyor.

Textview içerisinde mesaj olması için var msg adında hoş geldiniz mesajı tanımlanıyor. Textview içerisine settext içerisine msg ve value değerleri geliyor.

# **Rating Bar**

Hakkında sayfasında kullanılan Rating bar

findViewById ile rating bar id’ si eşleştiriliyor. rBar değişkeni oluşturuluyor ve ratingbar olarak tanımlanıyor. Stepsize puanın nasıl verileceğini belirtiyor. toFloat ise puanı yarım yani 3,5 puan verilmesini sağlıyor. numStars puan yıldız sayısını belirtiyor. Butona setOnClickListener içerisinde puan verdiğimizde kaç puan verdiğimizi belirmek için msg değişkeni ile ratingbar puanını toString ile yazıya dönüştürüp mesaj olarak verilmesi sağlanıyor.

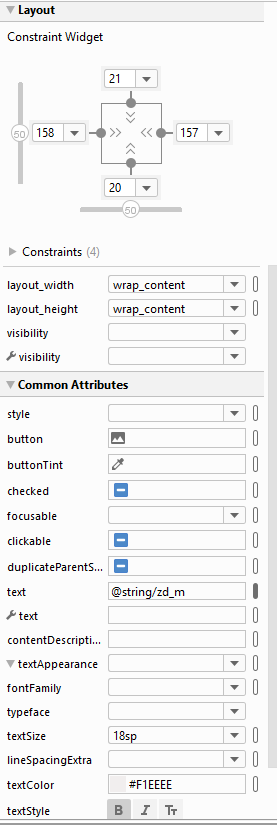
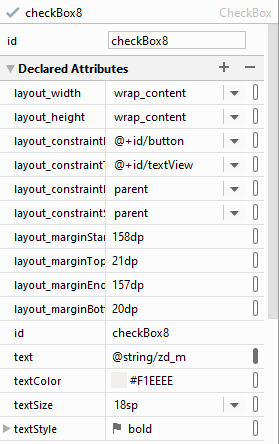


## **Check Box**

**CheckBox** Türkçeye işaret kutucuğu olarak çevrilen bu bileşen, genellikle aşağıda görüldüğü gibi kullanılır. Kullanıcının seçenekleri onayladığını veya kabul ettiğini belirtir.

Check Box bileşenini düzenlemek için text kısmından text metnini, yazı boyutunu, yazı rengini, ekran yerleşimini, id’ sini, boyutunu düzenleyebiliriz.

Design sekmesi Attributes penceresinden yukarıda görülen işlemler ve daha fazlası kısayollar ile yapılabilir. Constraint layout yerleşiminin yapılması gereklidir aksi halde hata verir.

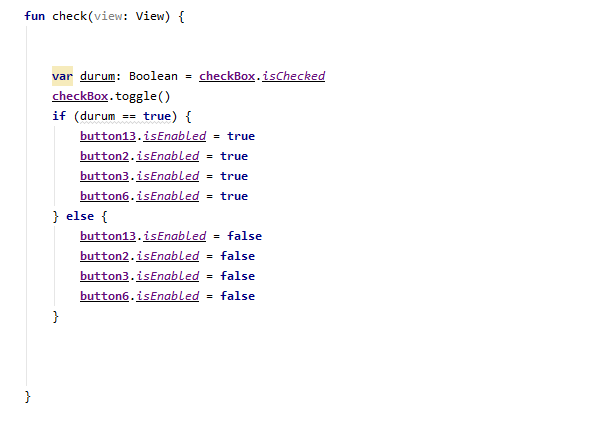


## Check Box İşaretlendiğinde

Check box işaretlendiğinde butonların aktif hale gelmesi, kaldırıldığında pasif hale gelmesi için ve diğer sayfalara geçişini sağlamak için fonksiyon oluşturulur.

Durum değişkeni boolean döngüsüne girer ve checkbox işaretlendiğinde

İf – else yapısı içerisinde eğer true ise yani tıklanmamış ise butonlar pasif hale gelir ve çalışmaz. Değilse yani checkbox işaretlenmiş ise butonlar aktif hale gelir ve kullanılabilir hale gelir.

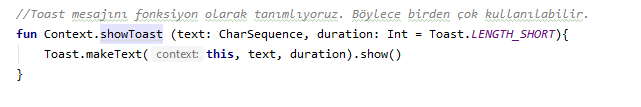


# **Sayfa Geçişleri**



Kotlin’ de sayfa geçişi için override fonksiyonu içerisinde buton.setOnClickListener seçiliyor ve val anahtar kelimesi ile intent nesnesi oluşturuluyor. this, bu butonda olduğunu belirtmek için kullanılıyor sonrasında ise sayfaya yönlendiriliyor.

## Bildirim Mesajı



Butona basıldığında mesaj verilmesi isteniliyorsa bunun için öncelikle ilgili kotlin dosyasında

OnCreate metodu içerisinde yazılmalıdır. Normalde basit olarak tanımlamak için yazılması gereken kod satırı;

Toast.makeText (getApplicationContext, "2.sayfayaya geçtiniz",Toast.LENGTH\_SHORT).show ();

(--burada yazılan Toast.makeText mesajın verilmesi için, getApplicationContext () bağlam için, “Tırnak içerisinde yazılan metin mesaj” , Toast.LENGTH\_SHORT ise mesajın kısa süre gösterileceğini belirtir istenilirse mesaj süresi uzatılabilir. , show ise mesajın gösterilmesini sağlar.)

# **Tanımlar**

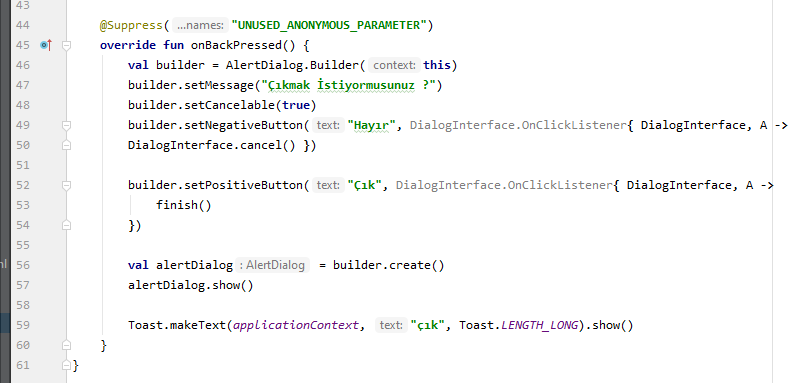
Kotlin’ de eklenen içeriklerin tanımı var-val-lateinit var anahtar kelimeleri ile yapılır. Var anahtar kelimesi; değişken adıdır, daha sonradan değiştirilebilir anahtardır. Val anahtar kelimesi var anahtar kelimesi gibi değişken adıdır, var’ dan farkı val ile oluşturulan bir değişkenin daha sonra değiştirilememesidir. Lateinit var anahtar kelimesi ise geç başlatılan değişkenler için kullanılır. Eğer kullanılacak değişkeni daha sonradan kullanılacak veya eklenmesi halinde lateinit var değişkeni kullanılmalıdır.



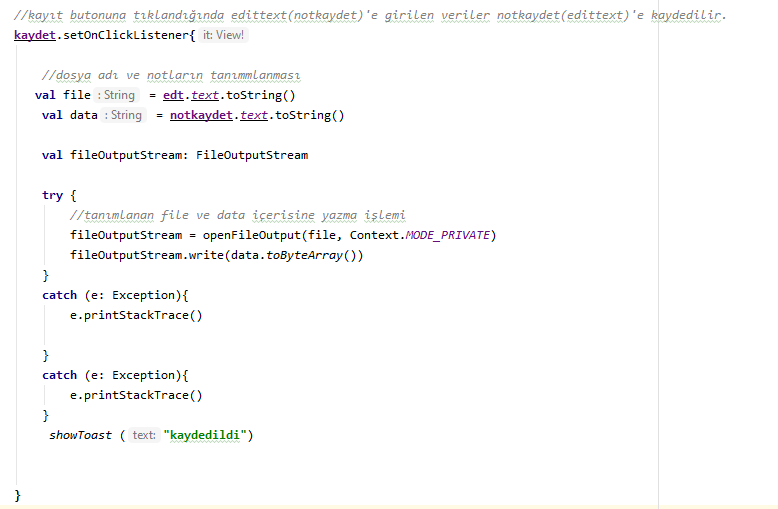
# **OnBackPressed () Metodu**

Projede uygulamadan çıkılması istenildiğinde uyarı mesajı olarak evet veya hayır olarak seçildiğinde çıkış işlemini sağlayan metot kullanılmıştır.

Override fonksiyonu olarak tanımlanan metot builder değişkeni ile AlertDialog olarak tanımlanıyor. Set mesaj uyarı mesaj içeriği, setCancelable ise hayır yani çıkılması istenilmediğinde true değerini alarak uygulamada kalması sağlanıyor. setNegativeButton (hayır seçeneği) içerisinde DialogInterface yani mesaj arayüzü hayır seçeneğini döndürüyor. Evet seçeneği için ise aynı işlemler setPositiveButton içerisinde finish() ile çıkış işlemi sağlanıyor. Fonksiyon içerisinde alertdialog tanımlanıyor ve alertdialog sekmesi show() ile görünmesi sağlanıyor.



# **Not Kaydetme**



## Kaydet butonuna basıldığına Kaydetme

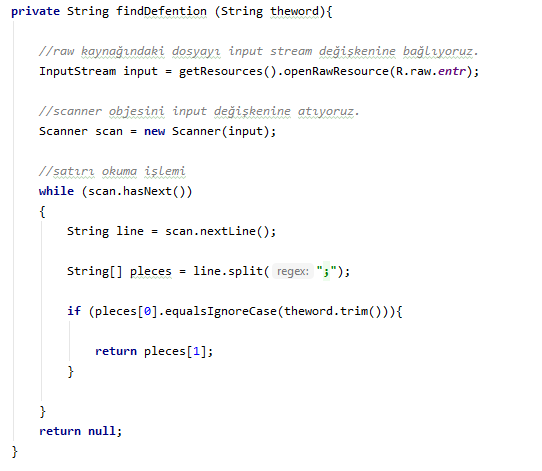
# **Kaydedilen Not Dosyasını Getirme**



# **Kelime Listesini Okuma İşlemi**

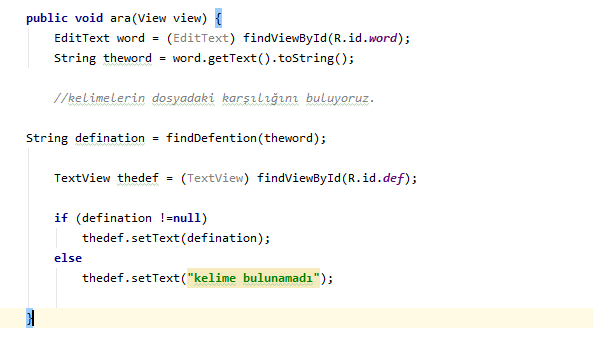
Kelimelerin karşılığını bulmak için Java Class kullanılıyor.

Proje içerisinde raw klasörü oluşturuyoruz ve içerisine txt uzantılı utf-8 formatındaki metin dosyasını içerisine aktarıyoruz. Scan.HasNext ile satır satır kelimeleri okuyoruz. (;) ile kelimelerin birbirinin karşılığı olarak belirliyoruz. Eğer “;” öncesindeki kelime aranır ise aynı satırdaki “;” sonrasındaki kelimeleri gösterir index olarak “;” öncesindeki kelimeleri 0

“;” sonrasındaki kelimelerin index’ i 1 olarak tanımlıyoruz.

# **Aranılan Kelimelerin Karşılığını bulmak**

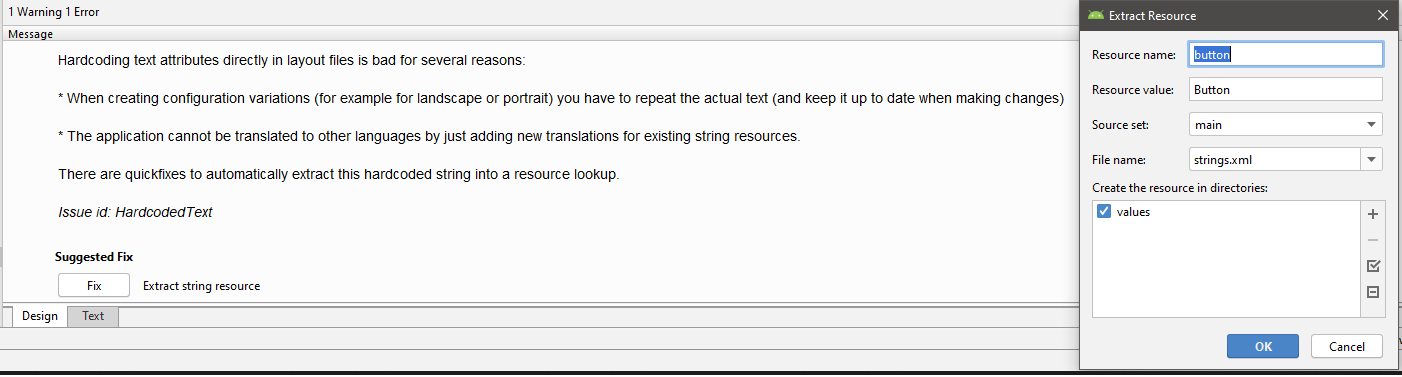
Aranılan kelimeleri editText’ ten alıp TextView’ da gösteriyoruz. Önceden okunan kelimeleri “theword” ile TextView içerisinde yazıya çevirerek gösterilmesi sağlanıyor.



# **String**

## String Tanımlamaları

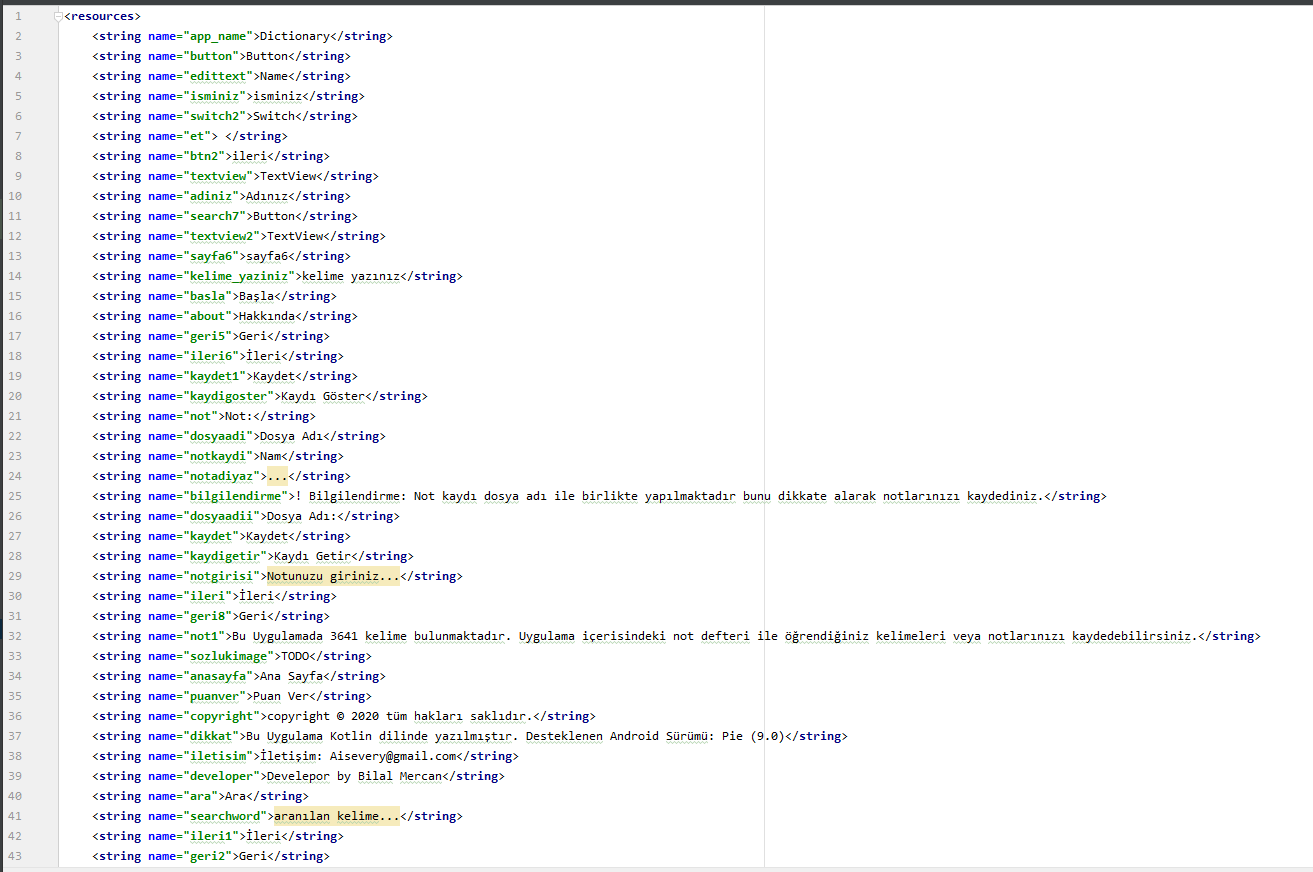
Örnek bir button string tanımlaması

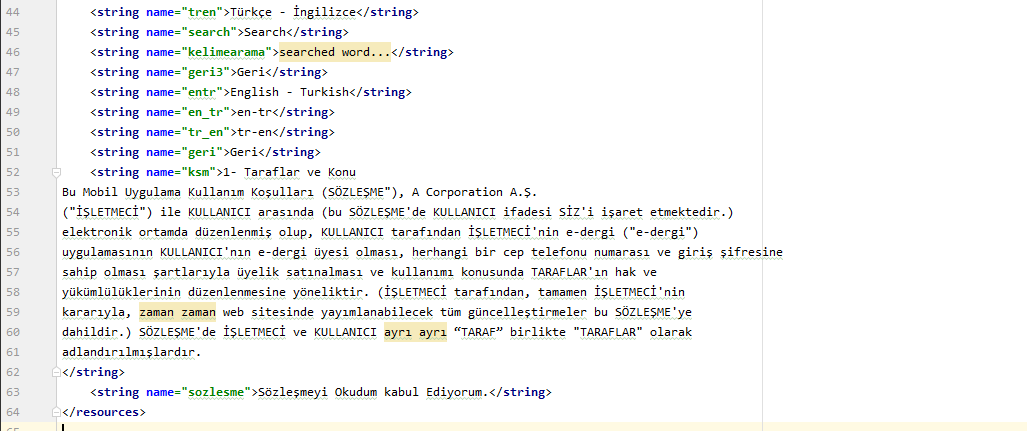


Ekranda görülen button string uyarısı string dosyasının olmadığını belirtiyor ve düzeltilmesi için seçenek sunuyor. Fix seçeneğine tıklandığında açılan pencerede button string adı ve string değer giriş ekranı geliyor. Bu pencerede kaynak adı ve kaynak değerlerini girdikten sonra string tanımlaması yapılabilir.

Bu işlemler yapılmadan da strings.xml dosyasına gidilerek oluşturulabilir. Dikkat edilmesi gerekilen husus string adı ve değerini düzenli ve doğru bir biçimde yapılması gerekmektedir.

## **String dosyası**





# **Buton ve Sayfa Tasarımı**

Eklenen butonlar için butonların daha estetik ve biçimli olması isteniliyorsa buton tasarımına ihtiyaç duyulabilir. Buton Tasarımı yapmak için öncelikle buton şeklinin nasıl olması gerektiği ayarlanmalıdır. Sonrasında ise buton üzerinde renk geçişleri ve buton kenarlarında renk ayarlamaları istenilirse yapılabilir.

2.satırda bulunan Shape ifadesi şekil anlamına gelen bir terimdir. Satırın sonunda yazan kod ise “rectangle” ifadesi ise dikdörtgen şeklini vermek için kullanılan bir terimdir.

Gradient ifadesi renk geçişi oluşturmak isteniliyorsa kullanılır.

Altında bulunan start color ve end color ise hangi renk ile başlayıp bitirileceğini belirtir.

Renklerin tanımlanması için seçilen renklerin renk kodu girilmesi gerekir.

Angle ifadesi açı anlamına geliyor 0 ile 360 arasında bir değer alınması gerekmektedir.

Buton çevresince kırpılmak istenen açı için kullanılır.

Type: Linear ifadesi ise doğrusal anlamına gelir. Butonun yerleşimini ifade eder.

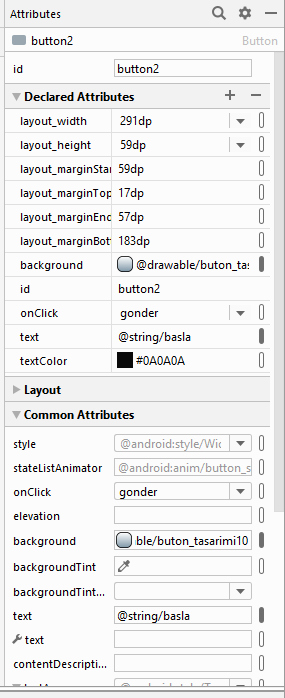
Corners ise Köşeler anlamına geliyor. Buton çevresince kırpılmak istenilen alan için kullanılıyor.

Radius ifadesi Kırpılmak istenilen alanın butonun her köşesinden kırpmak için kullanılır. İstenilirse sadece belli köşelerde kırpılabilir. Kırpılmak istenen köşeler için boyutları dp, sp, in, mm boyut seçenekleri ile de ayarlanılabilir.

Stroke ifadesi butona çerçeve eklememizi sağlıyor. Eklenen çerçevenin rengini android:color=”…” çift tırnak içerisine istenilen renk kodu yazılabilir. Çerçevenin genişliğini ayarlamak için android:width=”…” çift tırnak içerisine boyutu yazılması gerekiyor.

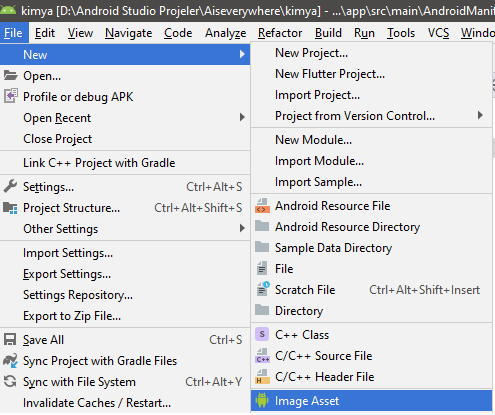


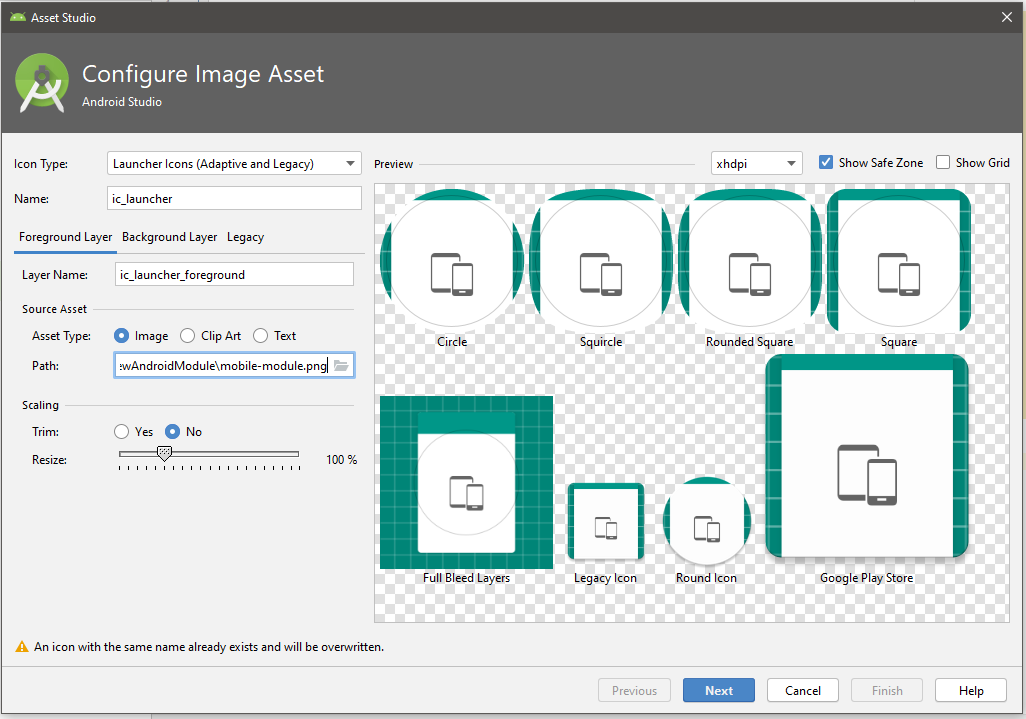
Butona renk yerine fotoğraf veya simge yüklenilmesi isteniyorsa start color ve end color satırına background=”@drawable/…” yazılarak drawable klasörünün içerisindeki fotoğraf veya simge yüklenilebilir.

Yandaki pencere içerisinde buton ile ilgili yazıtipi, yazı boyutu, buton mesajı, buton tasarımı ve button id ile ilgili bilgiler bulunmakta. Bu sekme içeresinde daha kolay bir şekilde bu bilgiler güncellenebilir.

# **İkon**

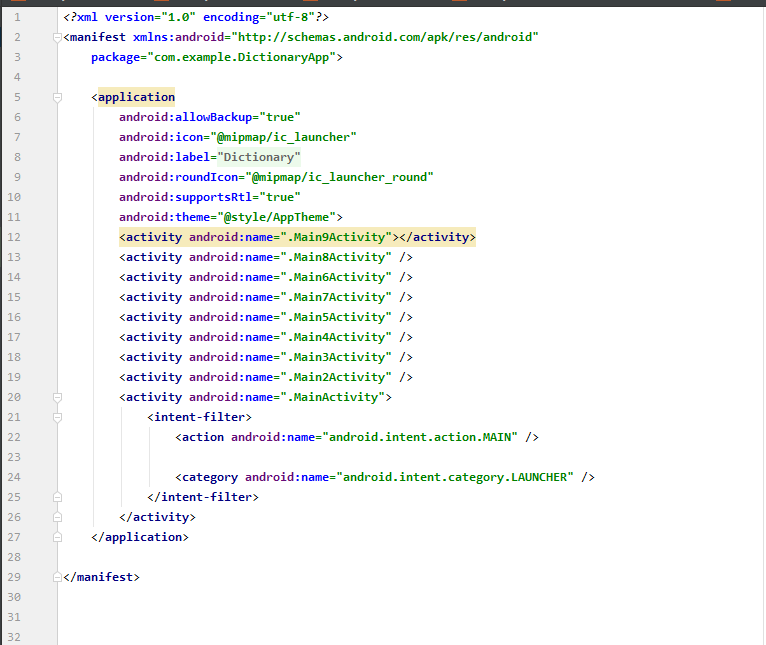
İkon Oluşturulması

İkon oluşturmak için öncelikle

 File –new—İmage Asset seçeneğine tıklamamız gerekiyor. Sonrasında ise Açılan Pencereden İkonun ayarlarını yapıp oluşturmuş oluyoruz.

Bu pencereden ikon ismi, ikon şekli, ikon dosyası ve ikon boyutu düzenlenebiliyor.

# **Manifest Dosyası**

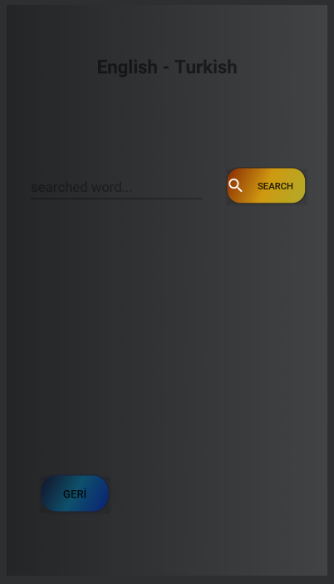
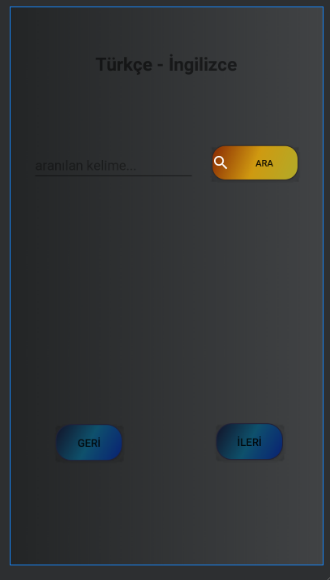
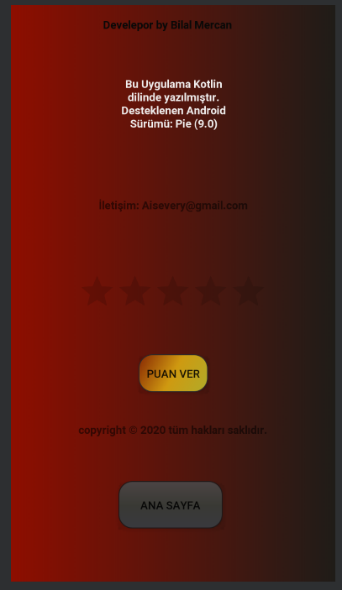
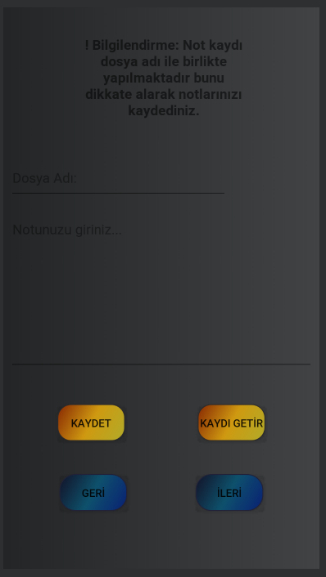
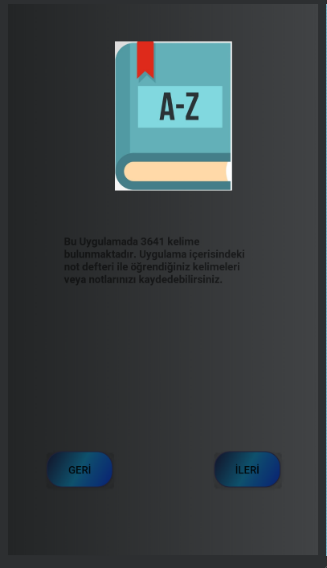
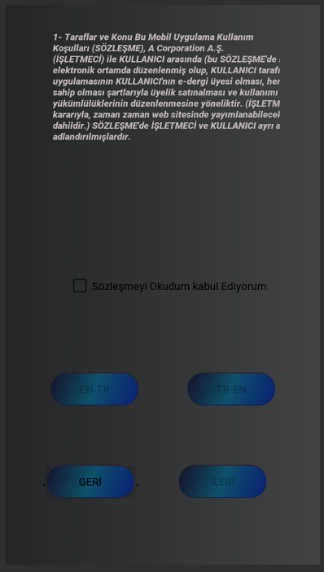
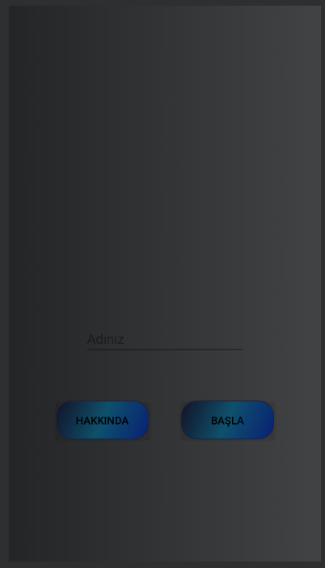


# **Sonuç**

## Uygulama Konusu

Uygulama; İngilizce-Türkçe sözlük uygulaması uygulamada 3640 kelime bulunmaktadır. Not kaydedebilir. Kaydedilen notları görüntüleyebilir, düzenleyebilirsiniz.

Uygulama Ekran Görüntüleri



# Proje, Sunum Dosyası, Afiş Github Adresi

https://github.com/BilalMercan/Kotlin\_Sozluk\_Proje.git

# **Kaynakça**

<https://kotlinlang.org/>

<https://developer.android.com/>

<https://www.mobilhanem.com/>

<https://www.youtube.com/>

<https://stackoverflow.com/>

<http://ilkinulas.github.io/>

<https://smality.com/>

<https://codinginflow.com/>

<https://gelecegiyazanlar.turkcell.com.tr/>

<https://medium.com/>

<https://www.youtube.com/watch?v=3iE_ZxBB-tc&list=PL9qDMO9EzLX25NTHm0q7svKLx__OZY8-e>

<https://www.yazilimbilisim.net/>

<https://www.mehmetkirazli.com/>