

YAZILIM VE DONANIM TASARIMI DERSİ

FOTOĞRAFTAN YAZI TESPİTİ VE ÇEVİRİ UYGULAMASI

RAPOR

Uygulamamın amacı kullanıcının galeriden veya telefonun kamerasından fotoğraf veya resim alarak içindeki metinsel kısımları tespit ederek dijital ortama yazı olarak aktarmak, aktarılan yazıyı düzenleyebilmek, İngilizce-Türkçe, Türkçe-İngilizce çevirisini yapmak ve bu işlemlerden sonra Whatsapp, Mesaj, Mail vb. uygulamalara paylaşımını sağlamaktır.

Proje Kotlin diliyle Android Studio üzerinden yazılmıştır.

Uygulamam 4 Activity den oluşuyor. Main, Scan, Translate, ve Splash Activity.

--Main Activity



>Galeriden fotoğraf seçmek için tıklanır.

>Kameradan fotoğraf çekerek görüntü almak için tıklanır.

--Burada butonların tanımlamalarını yaptım.

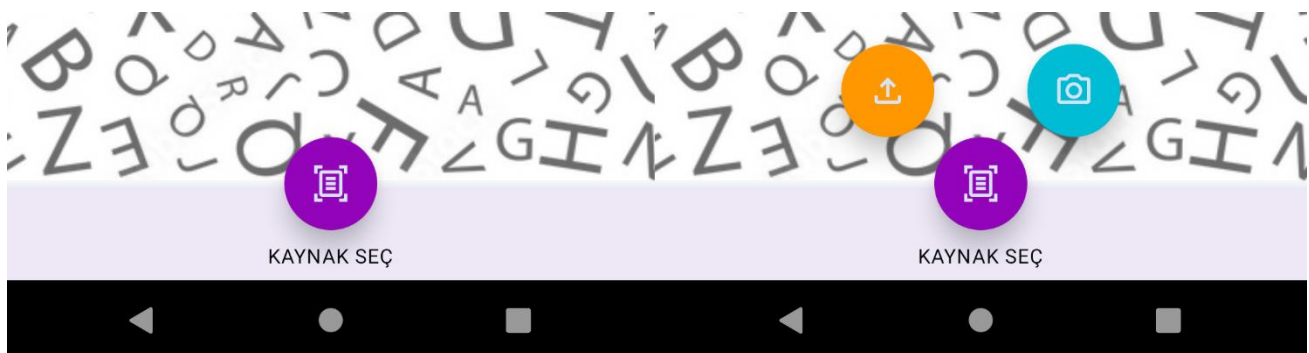
```
//buton tanımları
val altMenu: BottomNavigationView = findViewById(R.id.altMenu)
val altMenuContainer: MotionLayout = findViewById(R.id.altMenuContainer)
val hafizadanFotoSec: FloatingActionButton = findViewById(R.id.hafizadanFotoSec)
val fotoCek: FloatingActionButton = findViewById(R.id.fotoCek)
```

--Galeriden seçme butonuna tıklayınca çalışacak kod bloğu:

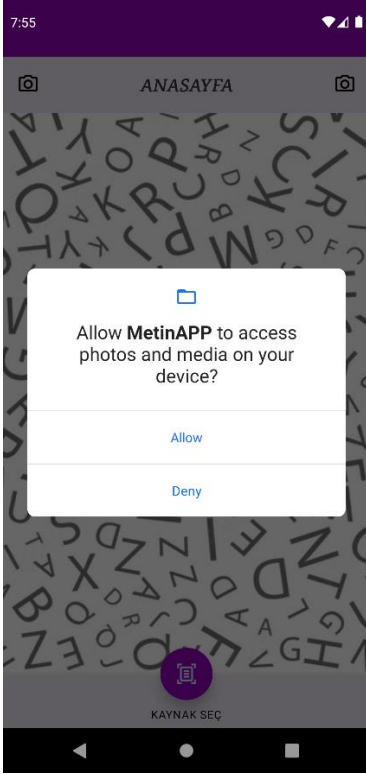
```
//galeriden foto seçme işlevi
hafizadanFotoSec.setOnClickListener { it: View!
    if (altMenuContainer.currentState == altMenuContainer.endState) {
        altMenuContainer.transitionToStart()
    }

    //hafıza okuma izin kontrolü
    if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) == PackageManager.PERMISSION_DENIED) {
        val permissions = arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE)
        requestPermissions(permissions, PERMISSION_CODE)
    } else {
        galeridenSec()
    }
}
```

--Galeriden seçme butonu açılan kapanan kaynak seçme butonun içindedir. İlk if döngüsü galeriden seçme veya kameradan seçme butonu seçimi yapıldıktan sonra açılır kapanır butonun eski haline dönmesini sağlar.



--Sonraki if bloğunda Galeriden seçme butonuna tıklayınca galeriye erişmek için daha önceden izin alınmış mı diye kontrolünü yapar alınmamışsa izin kontrolünü fonksiyonunu açar. Daha önceden alınmış ise direk galeriden seçme fonksiyonu çalışır. İzin verildikten sonra galeriden seçme fonksiyonu çalışır. İzin verildiyse uygulama diğer açılışlarda izin istemez.



İzin kontrolü kod bloğu:

```
//izin kontrolü sonucuna göre yapılacaklar
override fun onRequestPermissionsResult(
    requestCode: Int, permissions: Array<out String>, grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    when (requestCode) {
        PERMISSION_CODE -> {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                galeridenSec()
            } else {
                Toast.makeText(context: this, text: "Permission denied", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

--Galeriden seç kod bloğu

```
//galeriden seçme fonksiyonu
private fun galeridenSec() {
    val intent = Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI)
    startActivityForResult(intent, IMAGE_CHOOSE)
}
```

--Kameradan seçme butonun basılınca çalışacak kod bloğu:

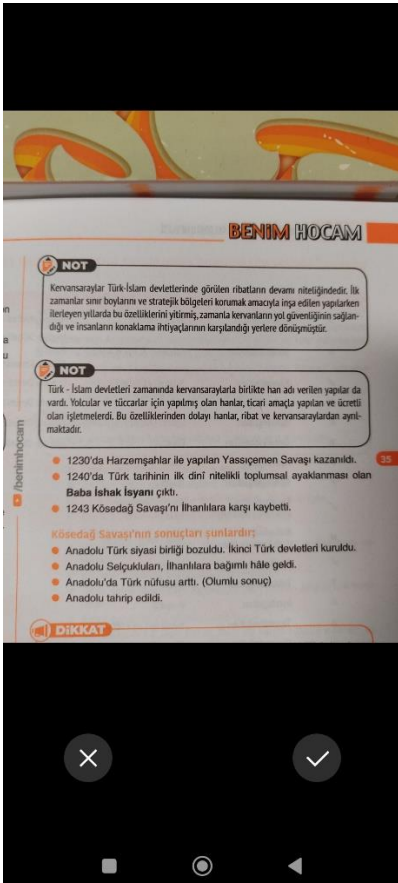
```
//fotoğraf çekme butonu işlevi
fotoCek.setOnClickListener { it: View!
    if (altMenuContainer.currentState == altMenuContainer.endState) {
        altMenuContainer.transitionToStart()
    }

    //içerik alışıveriş nesnesi (intent)
    val cekilenFotoIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
    cekilenFoto = getFotoDosyası()
    val providerFile = FileProvider.getUriForFile(
        context: this,
        authority: "com.brcn.metinapp.fileprovider",
        cekilenFoto
    )
    cekilenFotoIntent.putExtra(MediaStore.EXTRA_OUTPUT, providerFile)

    try {
        startActivityForResult(cekilenFotoIntent, TAKE_PHOTO)
    } catch (e: ActivityNotFoundException) {
        Toast.makeText(context: this, text: "Camera could not open", Toast.LENGTH_SHORT).show()
    }
}
```

--İlk olarak öncesinde anlattığım gibi floating butonun açma kapama kontrolünü yapar.

Sonra kamerayı açar.



Sonra kamera uygulamasından kendi uygulamamıza veri aktarımı yapmak için intent nesnesi oluşturur. Sonra çekilen fotoğrafı uygulamanın başında tanımlanan sonradan oluşturulacak dosya anlamına gelen değişkenin içine geçici olarak hafızaya alma fonksiyonu ile atar.

```
private lateinit var çekilenFoto: File
```

```
//fotoyu geçici olarak hafızaya alma
private fun getFotoDosyası(): File {
    val directoryStorage = getExternalFilesDir(Environment.DIRECTORY_PICTURES)
    return File.createTempFile(prefix: "photo", suffix: ".jpg", directoryStorage)
}
```

Oluşturulan file provider (dosya sağlayıcı) nesnesi ile geçici olarak hafızaya alınan nesneyi işlememize olanak tanır.

--Metin recognize işlemine başlangıç:

```
// Metin Recognize işlemi
@OptIn(DelicateCoroutinesApi::class)
private suspend fun extractedText(bitmap: Bitmap) {
    val fotoMetinRecognizer: TextRecognizer =
        TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS)
    val inputImage = InputImage.fromBitmap(bitmap, rotationDegrees: 0)

    ScanActivityUtils.bitmap = bitmap

    inputImage.let { image ->
        val value = GlobalScope.async { this: CoroutineScope
            fotoMetinRecognizer.process(image).addOnSuccessListener { it: Text!
                ScanActivityUtils.sonucText = it.text
            }.addOnFailureListener { it: Exception
                Toast.makeText(applicationContext, text: "Metin Bulunamadı", Toast.LENGTH_SHORT)
                    .show()
            }
        }
        value.await().addOnSuccessListener { it: Text!
            val intent = Intent(applicationContext, ScanActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Burada recognize işlemine başlanıyor. Öncelikle parametresi bitmap olan suspend bir fonksiyon (durup yeniden başlayabilen fonksiyon) oluşturuluyor. Burada coroutinler kullanıldığı için asenkron

işlemden belirlenen girdiye göre işlem yapıp Scanactivityutils deki sınıfın içindeki sonucText e çıkarım yapılmış text atılıyor. Ve en alttaki kod bloğu ile Scan Activity e geçiş sağlanıyor.

--Asenkron olarak girdinin kameradan mı galeriden mi alındığının belirlendiği işlem:

```
@OptIn(DelicateCoroutinesApi::class)
@Deprecated(message: "Deprecated in Java")
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    if (requestCode == TAKE_PHOTO && resultCode == Activity.RESULT_OK) {
        val takenPhoto = BitmapFactory.decodeFile(cekilenFoto.absolutePath)
        GlobalScope.launch { this: CoroutineScope
            extractedText(takenPhoto)
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data)
    }

    if (requestCode == IMAGE_CHOOSE && resultCode == Activity.RESULT_OK) {
        val chosenPhoto = MediaStore.Images.Media.getBitmap(this.contentResolver, data?.data)
        GlobalScope.launch { this: CoroutineScope
            extractedText(chosenPhoto)
        }
    }
}
```

Burada verinin kameradan mı galeriden mi geldiği belirlenip extractedTexte oradan gelen veri yansıtılıyor.

Main Activity de import edilen kütüphaneler:

```
package com.brcn.metinapp.ui.activities

import android.Manifest
import android.app.Activity
import android.content.ActivityNotFoundException
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.os.Bundle
import android.os.Environment
import android.provider.MediaStore
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.constraintlayout.motion.widget.MotionLayout
import androidx.core.content.FileProvider
import com.brcn.metinapp.R
import com.brcn.metinapp.utils.ScanActivityUtils
import com.google.android.material.bottomnavigation.BottomNavigationView
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.mlkit.vision.common.InputImage
import com.google.mlkit.vision.text.TextRecognition
import com.google.mlkit.vision.text.TextRecognizer
import com.google.mlkit.vision.text.latin.TextRecognizerOptions
import kotlinx.coroutines.DelicateCoroutinesApi
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.async
import kotlinx.coroutines.launch
import java.io.File
```

Burada özellikle Google ML kitin text recognition kütüphanesi kullanılmıştır.

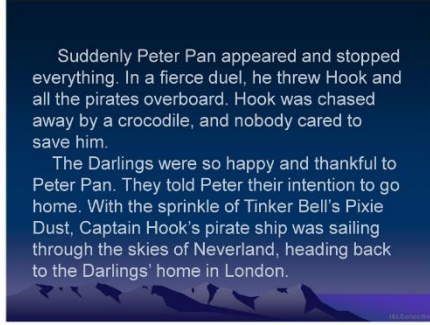
Kütüphaneyi implemente etmek için build.gradle.app kısmını kullanıyoruz

```
implementation 'com.google.android.gms:play-services-mlkit-text-recognition:18.0.2'
```

--Scan Activity



TARAMA SONUCU



Suddenly Peter Pan appeared and stopped everything. In a fierce duel, he threw Hook and all the pirates overboard. Hook was chased away by a crocodile, and nobody cared to save him.

The Darlings were so happy and thankful to Peter Pan. They told Peter their intention to go home. With the sprinkle of Tinker Bell's Pixie Dust, Captain Hook's pirate ship was sailing through the skies of Neverland, heading back to the Darlings' home in London.

1SLCollective.com



>Activityyi kapatma butonu.

>Resimden yapılmış çıkarımı uygulamalar ile paylaşma butonu.

>Türkçe-İngilizce çeviri butonu.

>İngilizce-türkçe çeviri butonu

>Çıkarılmış texti düzenleme butonu.

--Burada butonların ve diğer değişkenlerin tanımlamasını yaptım.

```
// 0 = OKUMA MODU
// 1 = DÜZENLEME MODU
var durum = 0

val eklentiler = ContextExtensions(context: this)

//butonlar
val secilenVeyaCekilenFoto: ImageView = findViewById(R.id.secilenVeyaCekilenFoto)
val sonucMetin: TextView = findViewById(R.id.sonucMetin)
val sonucuKapat: ImageView = findViewById(R.id.translateKapat)
val sonucuPaylas: ImageView = findViewById(R.id.translatePaylas)
val sonucuDuzenle: FloatingActionButton = findViewById(R.id.sonucuDuzenle)
val sonucuCevir: FloatingActionButton = findViewById(R.id.sonucuCevir)
val sonucuCevirTrEn: FloatingActionButton = findViewById(R.id.sonucuCevirTrEn)
val sonucTextDuzenleyici: TextArea = findViewById(R.id.sonucTextDuzenleyici)
val sonucTextContainer: ScrollView = findViewById(R.id.sonucTextContainer)

secilenVeyaCekilenFoto.setImageBitmap(ScanActivityUtils.bitmap)
sonucMetin.text = ScanActivityUtils.sonucText
```

--Sonucu düzenleme işlevi

```
sonucuDuzenle.setOnClickListener { it: View!
    if (!sonucTextContainer.isVisible) {
        sonucuDuzenle.setImageDrawable(ContextCompat.getDrawable(context: this, R.drawable.ic_done))

        sonucMetin.visibility = View.INVISIBLE
        sonucTextContainer.visibility = View.VISIBLE

        sonucTextDuzenleyici.setText(ScanActivityUtils.sonucText, TextView.BufferType.EDITABLE)

        durum = 1
    } else {
        sonucuDuzenle.setImageDrawable(ContextCompat.getDrawable(context: this, R.drawable.ic_edit))

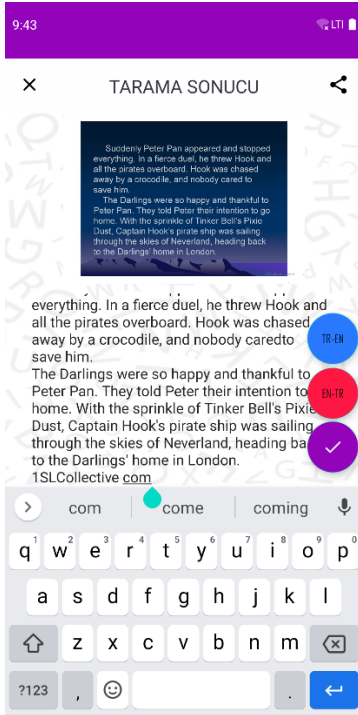
        ScanActivityUtils.sonucText = sonucTextDuzenleyici.text.toString()
        sonucMetin.text = ScanActivityUtils.sonucText

        sonucTextContainer.visibility = View.INVISIBLE
        sonucMetin.visibility = View.VISIBLE

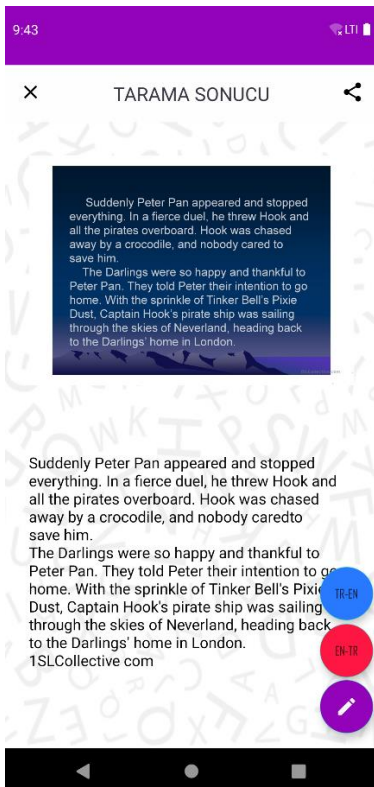
        if (durum == 1) {
            eklentiler.hideKeyboard(sonucuDuzenle)
        }

        durum = 0
    }
}
```

Öncelikle container görünümü açık değilse sonucMetin değişkeninin kapatıp düzenlenebilir container şeklinde açılmasını sağlıyor. Duruma 1 ataması yapıyor Yani düzenleme modunda olduğu belirleniyor ve buton simgesi düzenleme modunda olduğunu bildirir şekilde değiştiriliyor.



Düzenleme modu açılıp gereken değişiklik yapıldıktan sonra butona tıklandığında else kısmı devreye giriyor ve Scanactivity utilsdeki sonucText değişkenine son hali atılıyor. Container görünüm kapatılıyor. sonucMetin görünümü görünür hale getiriliyor. Durum kontrolü sağlanıp klavye gizleme eklentisi devreye giriyor ve klavye kapatılıyor. Durum tekrar okuma moduna alınıyor. Ve buton simgesi tekrar eski haline döndürülüyor.



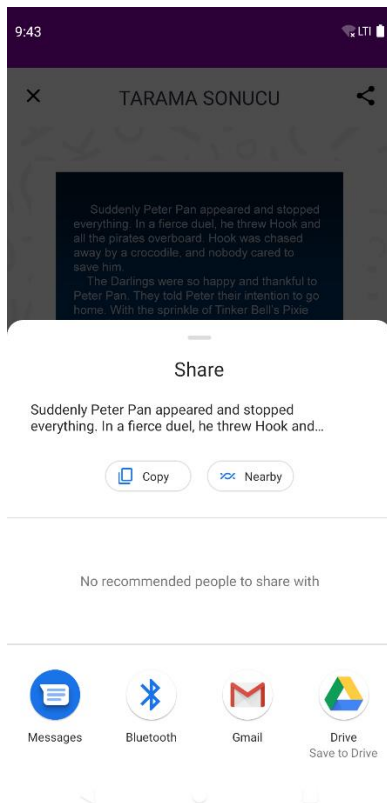
--Klavyeyi gizleme

```
class ContextExtensions(val context: Context) {  
  
    fun hideKeyboard(view: View) {  
        val inputMethodManager = context.getSystemService(Activity.INPUT_METHOD_SERVICE) as InputMethodManager  
        inputMethodManager.hideSoftInputFromWindow(view.windowToken, flags: 0)  
    }  
}
```

--Scan Activity Paylaş Butonu

Bu butona tıklanınca sonucText değişkeni share intentine düzenlenebilir text formatında gönderilir ve share intentini başlatır

```
sonucuPaylas.setOnClickListener { it: View!  
    val sendIntent: Intent = Intent().apply { this: Intent  
        action = Intent.ACTION_SEND  
        putExtra(Intent.EXTRA_TEXT, ScanActivityUtils.sonucText)  
        type = "text/plain"  
    }  
  
    val shareIntent = Intent.createChooser(sendIntent, title: null)  
    startActivity(shareIntent)  
}
```



--Scan Activity kapat butonu

```
sonucuKapat.setOnClickListener {  
    finish()  
}
```

Kod bloğu ile activity sonlandırılarak ana sayfaya dönüşü sağlanır.

Ve

```
override fun onDestroy() {  
    super.onDestroy()  
  
    ScanActivityUtils.run { this: ScanActivityUtils  
        sonucText = ""  
        bitmap = createBitmap( width: 1, height: 1)  
    }  
}
```

Kaptıldığında aynı zamanda sonucText ve bitmap görüntü nesnemiz boşaltılır.

--EN-TR Butonu

```
sonucuCevir.setOnClickListener{ it: View!  
    ScanActivityUtils.ceviriText = ""  
    cevirmeIslemi()  
    val intent = Intent(applicationContext, TranslateActivity::class.java)  
    startActivity(intent)
```

Öncelikle tekrar tekrar işlem yapılabilmesi için çevirilmiş textin içi boşaltılır. Sonra çevirme işlemi çalıştırılıp Translate Activityye geçiş sağlanır.

****build.gradle.app** dosyasına çeviri kütüphanesi implemente edilmelidir.

```
implementation 'com.google.mlkit:translate:17.0.1'
```

****android** manifest.xml dosyasına erişim izni eklenmelidir

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

--Çevirme işlemi

```

private fun cevirmeIslemi(){

    val options:TranslatorOptions=TranslatorOptions.Builder()
        .setSourceLanguage(TranslateLanguage.ENGLISH)
        .setTargetLanguage(TranslateLanguage.TURKISH)
        .build()

    englishTurkishTranslator=Translation.getClient(options)

    //dil dosyası yoksa indirme işlemi
    englishTurkishTranslator.downloadModelIfNeeded().addOnSuccessListener { it: Void!

        metniCevir()

    }.addOnFailureListener { it: Exception

        ScanActivityUtils.ceviriText = "HATA ${it.message}"
    }

}

```

Kütüphanenin dokümantasyonuna göre çeviri ayarları yapılıyor sonra sayfanın başında tanımlanan lateinit nesnenin içine atılıyor.

```
lateinit var englishTurkishTranslator:Translator
```

Sonra istenen dil dosyası indirilip metniCevir fonksiyonu devreye sokuluyor.

--metniCevir

```

private fun metniCevir() {

    englishTurkishTranslator.translate(ScanActivityUtils.sonucText).addOnSuccessListener { it: String!

        ScanActivityUtils.ceviriText =it
    }

    .addOnFailureListener { it: Exception

        ScanActivityUtils.ceviriText = "HATA ${it.message}"
    }

}

```

Burada translator nesnesindeki ayarlarla beraber sonucText işleme tabi tutulup ceviriText değişkeninin içine çevirisi yapılmış text atılıyor. Ve çeviri işlemi tamamlanmış oluyor.

*Türkçe-İngilizce çeviri butonuda aynı şekilde çalışmaktadır.

--Translate Activity



Burada düzenleme, kapatma ve paylaşma butonu yer almaktadır. Buradaki düzenleme ve paylaşma butonu çeviri yapılmış text üzerinde işlem yapar. Kapatma butonuna tıklandığında bütün activiyler kapanır ve anasayfaya dönülür.

--Tanımlamalar

```
// 0 = OKUMA MODU
// 1 = DÜZENLEME MODU
var durum = 0

val eklentiler = ContextExtensions(context: this)

//butonlar

val sonucTranslateMetin: TextView = findViewById(R.id.sonucTranslateMetin)
val sonucTranslateContainer : ScrollView = findViewById(R.id.sonucTranslateContainer)
val sonucTranslateDuzenleyici: TextArea = findViewById(R.id.sonucTranslateDuzenleyici)

val sonucMetin: TextView = findViewById(R.id.sonucMetin)
val translateKapat: ImageView = findViewById(R.id.translateKapat)
val translatePaylas: ImageView = findViewById(R.id.translatePaylas)
val sonucuDuzenle: FloatingActionButton = findViewById(R.id.sonucuDuzenle)

sonucMetin.text = ScanActivityUtils.sonucText
sonucTranslateMetin.text = ScanActivityUtils.ceviriText
```

--Kapatma butonu

```
translateKapat.setOnClickListener { it: View!
    val intent = Intent(applicationContext, MainActivity::class.java)
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
    startActivity(intent)
}
```

Farklı olarak önceki activitiyle beraber bu activityyi de sonlandırarak main activitye dönüşü sağlar.

--Paylaşma Butonu

```
translatePaylas.setOnClickListener { it: View!
    val sendIntent: Intent = Intent().apply { this: Intent
        action = Intent.ACTION_SEND
        putExtra(Intent.EXTRA_TEXT, ScanActivityUtils.ceviriText)
        type = "text/plain"
    }

    val shareIntent = Intent.createChooser(sendIntent, title: null)
    startActivity(shareIntent)
}
```

--Düzenleme Butonu

```
sonucuDuzenle.setOnClickListener { it: View!
    if (!sonucTranslateContainer.isVisible) {
        sonucuDuzenle.setImageDrawable(ContextCompat.getDrawable(context: this, R.drawable.ic_done))

        sonucTranslateMetin.visibility = View.INVISIBLE
        sonucTranslateContainer.visibility = View.VISIBLE

        sonucTranslateDuzenleyici.setText(ScanActivityUtils.ceviriText, TextView.BufferType.EDITABLE)

        durum = 1
    } else {
        sonucuDuzenle.setImageDrawable(ContextCompat.getDrawable(context: this, R.drawable.ic_edit))

        ScanActivityUtils.ceviriText = sonucTranslateDuzenleyici.text.toString()
        sonucTranslateMetin.text = ScanActivityUtils.ceviriText

        sonucTranslateContainer.visibility = View.INVISIBLE
        sonucTranslateMetin.visibility = View.VISIBLE

        if (durum == 1) {
            eklentiler.hideKeyboard(sonucuDuzenle)
        }

        durum = 0
    }
}
```

Components: Anc

TASARIM (LAYOUT) KISMI

--Giriş Animasyonu Splash Activity

```
class SplashActivity : AppCompatActivity() {  
    companion object {  
        const val animasyonZamani: Long = 5000  
    }  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_splash)  
  
        Handler(this.mainLooper).postDelayed({  
  
            startActivity(Intent(packageContext: this, MainActivity::class.java))  
  
            finish()  
  
        }, animasyonZamani)  
    }  
}
```

Animasyonun zamanını ayarlıyoruz ve tamamlandıktan sonra ana ekrana geçişi sağlıyoruz.



Yükleniyor...



Animasyonu kullanmak için build.gradle.app dosyasına lottie imlemente etmeliyiz.

```
implementation "com.airbnb.android.lottie:$lottieVersion"
```

Açılışta ilk olarak çalışmasını sağlamak için android manifest.xml dosyasına aşağıdaki satır eklenmelidir.

```
<activity
    android:name=".ui.activities.SplashActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
```

--Splash Animation Layout dosyası

```
<com.airbnb.lottie.LottieAnimationView
    android:id="@+id/acilisEkrani"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:lottie_autoPlay="true"
    app:lottie_loop="true"
    app:lottie_rawRes="@raw/splash_anim" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="70dp"
    android:text="Yükleniyor..."
    android:textSize="48sp"
    app:layout_constraintBottom_toBottomOf="@+id/acilisEkrani"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

Gerekli konumlandırma ayarları ve animasyon kaynağı seçim işlemleri yapılır.

activity_main.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:layout_height="match_parent"
    android:background="@color/background"
    tools:context=".ui.activities.MainActivity">

    <RelativeLayout
        android:id="@+id/topBar"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:background="#EDE7F6"
        android:gravity="center_vertical"
        app:layout_constraintTop_toTopOf="parent">

        <ImageView
            android:id="@+id/ivMenu"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_marginStart="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_camera_outlined" />

        <TextView
            android:id="@+id/tvTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:fontFamily="@font/crete_round_italic"
            android:text="@string/home"
            android:textAlignment="center"
            android:textColor="@color/font_color_dark"
            android:textSize="20sp" />

        <ImageView
            android:id="@+id/ivBack"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginEnd="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_camera_outlined" />
    </RelativeLayout>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentContainerView"
        android:name="com.brcn.metinapp.ui.fragments.HomeFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"

        app:layout_constraintBottom_toTopOf="@id/guidelineBottom"
        app:layout_constraintTop_toBottomOf="@id/topBar"
        tools:layout="@layout/fragment_home">

    </androidx.fragment.app.FragmentContainerView>

    <androidx.constraintlayout.motion.widget.MotionLayout
        android:id="@+id/altMenuContainer"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        app:layoutDescription="@xml/floating_action_button_group_scene"
        app:layout_constraintBottom_toBottomOf="parent">

        <com.google.android.material.bottomnavigation.BottomNavigationView
            android:id="@+id/altMenu"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            app:layout_constraintBottom_toBottomOf="parent">
        </com.google.android.material.bottomnavigation.BottomNavigationView>
    </androidx.constraintlayout.motion.widget.MotionLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

```

        android:layout_height="wrap_content"
        android:background="#EDE7F6"
        app:itemIconTint="#9303BA"
        app:itemTextColor="@color/black"
        app:labelVisibilityMode="selected"

        app:layout_constraintBottom_toBottomOf="parent"
        app:menu="@menu/navigation" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fotoCek"
    style="@style/Widget.MaterialComponents.FloatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:contentDescription="@string/take_photo"
    android:visibility="invisible"
    app:backgroundTint="#00BCD4"
    app:elevation="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:srcCompat="@drawable/ic_camera_outlined"
    app:tint="@color/gray" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/hafizadanFotoSec"
    style="@style/Widget.MaterialComponents.FloatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:layout_marginBottom="30dp"
    android:contentDescription="@string/pick_image"
    android:visibility="invisible"
    app:backgroundTint="#FF9800"
    app:elevation="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_insetEdge="bottom"
    app:srcCompat="@drawable/ic_upload_outlined"
    app:tint="@color/gray" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/floatingButton"
    style="@style/Widget.MaterialComponents.FloatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:contentDescription="@string/scan"
    app:backgroundTint="#9303BA"
    app:elevation="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:srcCompat="@drawable/ic_scan_outlined"
    app:tint="@color/gray" />
</androidx.constraintlayout.motion.widget.MotionLayout>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guidelineBottom"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_end="60dp" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Fragment_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    android:backgroundTint="#6BFFFFFF"
    android:backgroundTintMode="src_over"
    tools:context=".ui.fragments.HomeFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textColor="#9303BA"
        android:gravity="center"
        android:text="@string/hello_blank_fragment"
        android:fontFamily="@font/alfa_slab_one"
        android:textSize="40sp"
    />

</FrameLayout>
```

--Fragment container ile anasayfayı görsellik kazandırılmaya çalışılmıştır.

```
class HomeFragment : Fragment() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

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



activity_scan.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:layout_height="match_parent"
    android:background="@drawable/background"
    android:backgroundTint="#FFFFFF"
    android:backgroundTintMode="src_over"
    tools:context=".ui.activities.ScanActivity">

    <RelativeLayout
        android:id="@+id/topBar"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:background="@color/bottom_navigation_background"
        android:gravity="center_vertical"
        app:layout_constraintTop_toTopOf="parent">

        <ImageView
            android:id="@+id/translateKapat"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_marginStart="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_close" />

        <TextView
            android:id="@+id/tvTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:text="@string/result"
            android:textAlignment="center"
            android:textColor="@color/font_color_dark"
            android:textSize="20sp" />

        <ImageView
            android:id="@+id/translatePaylas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginEnd="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_share" />
    </RelativeLayout>

    <ImageView
        android:id="@+id/secilenVeyaCekilenFoto"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginStart="25dp"
        android:layout_marginEnd="25dp"
        android:layout_marginBottom="10dp"
        android:contentDescription="@string/image_chosen_by_user"
        app:layout_constraintBottom_toTopOf="@id/sonucMetin"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/topBar"
        tools:ignore="ImageContrastCheck" />

    <TextView
        android:id="@+id/sonucMetin"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginStart="25dp"
        android:layout_marginTop="25dp"
        android:layout_marginEnd="25dp"
        android:textAlignment="viewStart"
        android:textColor="@color/black"
        android:textSize="16sp">
```

```

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@id/secilenVeyaCekilenFoto" />

<ScrollView
    android:id="@+id/sonucTextContainer"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginTop="10dp"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="@id/secilenVeyaCekilenFoto"
    app:layout_constraintVertical_bias="0.0"
    tools:layout_editor_absoluteX="-143dp">

    <com.brcn.metinapp.views.TextArea
        android:id="@+id/sonucTextDuzenleyici"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="25dp"
        android:gravity="start"
        android:importantForAutofill="no"
        android:inputType="text|textMultiLine|textNoSuggestions"
        android:textAlignment="viewStart"
        android:textSize="16sp"
        tools:ignore="LabelFor" />

</ScrollView>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/sonucuDuzenle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="5dp"
    android:layout_marginBottom="16dp"
    android:backgroundTint="#9303BA"
    android:contentDescription="@string/edit_extracted_text"
    android:src="@drawable/ic_edit"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:tint="@color/background" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/sonucuCevir"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="5dp"
    android:layout_marginBottom="75dp"
    android:backgroundTint="#FF1744"
    android:contentDescription="@string/translate"
    android:src="@drawable/ic_translateen"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:tint="@color/black" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/sonucuCevirTrEn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="5dp"
    android:layout_marginBottom="135dp"
    android:backgroundTint="#2979FF"
    android:contentDescription="@string/translateTr"
    android:src="@drawable/ic_translateetr"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:tint="@color/black" />

</androidx.constraintlayout.widget.ConstraintLayout>

```


activity_translate.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:layout_height="match_parent"
    android:background="@drawable/background"
    android:backgroundTint="#FFFFFF"
    android:backgroundTintMode="src_over"
    tools:context=".ui.activities.TranslateActivity">

    <RelativeLayout
        android:id="@+id/topBar"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:background="@color/bottom_navigation_background"
        android:gravity="center_vertical"
        app:layout_constraintTop_toTopOf="parent">

        <ImageView
            android:id="@+id/translateKapat"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_marginStart="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_close" />

        <TextView
            android:id="@+id/tvTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentEnd="true"
            android:text="@string/translate_result"
            android:textAlignment="center"
            android:textColor="@color/font_color_dark"
            android:textSize="20sp" />

        <ImageView
            android:id="@+id/translatePaylas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginEnd="15dp"
            android:contentDescription="@string/navigation_icon"
            android:src="@drawable/ic_share" />
    </RelativeLayout>

    <TextView
        android:id="@+id/sonucMetin"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginStart="25dp"
        android:layout_marginTop="25dp"
        android:layout_marginEnd="25dp"
        android:layout_marginBottom="10dp"
        android:textAlignment="viewStart"
        android:textColor="@color/black"
        android:textSize="14sp"
        app:layout_constraintBottom_toTopOf="@id/sonucTranslateContainer"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/topBar" />

    <ScrollView
        android:id="@+id/sonucTextContainer"
        android:layout_width="match_parent"
        android:layout_height="207dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toTopOf="@+id/sonucTranslateContainer"
        app:layout_constraintEnd_toEndOf="parent">
```

```

app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/topBar">

<com.brcn.metinapp.views.TextArea
    android:id="@+id/sonucTextDuzenleyici"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="25dp"
    android:gravity="start"
    android:importantForAutofill="no"
    android:inputType="text|textMultiLine|textNoSuggestions"
    android:textAlignment="viewStart"
    android:textSize="14sp"
    tools:ignore="LabelFor" />
</ScrollView>

<TextView
    android:id="@+id/sonucTranslateMetin"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginStart="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginEnd="25dp"
    android:textAlignment="viewStart"
    android:textColor="@color/black"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="@id/sonucTextContainer" />

<ScrollView
    android:id="@+id/sonucTranslateContainer"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toBottomOf="@id/sonucTextContainer"
    app:layout_constraintVertical_bias="0.0"
    tools:layout_editor_absoluteX="-143dp">

    <com.brcn.metinapp.views.TextArea
        android:id="@+id/sonucTranslateDuzenleyici"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="25dp"
        android:gravity="start"
        android:importantForAutofill="no"
        android:inputType="text|textMultiLine|textNoSuggestions"
        android:textAlignment="viewStart"
        android:textSize="16sp"
        tools:ignore="LabelFor" />
    </ScrollView>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/sonucuDuzenle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="5dp"
        android:layout_marginBottom="16dp"
        android:backgroundTint="#9303BA"
        android:contentDescription="@string/edit_extracted_text"
        android:src="@drawable/ic_edit"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:tint="@color/background" />

</androidx.constraintlayout.widget.ConstraintLayout>

```