

Oyunun Çalışma Mantığı

1-Kelime ekleme sayfası üzerinden DBHelper sınıfındaki “KelimeEkle” metodu sayesinde Veritabanıma kelimeleri ekliyorum.

DbHelper

```
fun KelimeEkle(kelime: Kelimeler){
    val sqLiteDB = this.writableDatabase
    val contentValues = ContentValues()
    contentValues.put(COL_KELIME , kelime.kelime)
    val result = sqLiteDB.insert(TABLE_NAME,null,contentValues)
    Toast.makeText(context,if(result != -1L) "Kayıt Başarılı" else "Kayıt yapılamadı.", Toast.LENGTH_SHORT).show()
}
```

2-Oyun ekranında DBHelper sınıfından “KelimeleriGetir()” metodu sayesinde veritabanımda kayıtlı olan kelimeler üzerinden işlem yapabilmek için “kelimeList” isimli değişkene aktarıyorum

OyunEkranı

```
var kelimeList = db.KelimeleriGetir()// veritabanımdaki tüm kelimeleri kelimeList Değişkenine aktardım
```

DbHelper

```
fun KelimeleriGetir():MutableList<Kelimeler>{
    val kelimeList = mutableListOf<Kelimeler>()
    val sqLiteDB = this.readableDatabase
    val query = "SELECT * FROM $TABLE_NAME"
    val result = sqLiteDB.rawQuery(query,null)
    if(result.moveToFirst()){
        do {
            val kelime = Kelimeler()
            kelime.id = result.getString(result.getColumnIndex(COL_ID)).toInt()
            kelime.kelime = result.getString(result.getColumnIndex(COL_KELIME))
            kelimeList.add(kelime)
        }while (result.moveToNext())
    }
    result.close()
    sqLiteDB.close()
    return kelimeList
}
```

3-“kelimeList” değişkeninin içerisine aktardığım kelimeler üzerinden random komutu ile rastgele bir kelimeyi seçip “kelime” değişkeninin içerisine aktarıyorum

```
var rnd = (0..kelimeList.count()).random()
var kelime = kelimeList[rnd].kelime.toString()
```

4-Oyunu oynayan kullanıcının, random komutuyla seçilen kelime uzunluğunu görebilmesi için for döngüsüne sokarak kelime uzunluğu kadar ekrana “_” işareti koyuyorum.

(örneğin kelime değişkenimde elma varsa, oyun ekranında “_ _ _ _” olarak gözükmesi için)

```
for (k in kelime) { //gelen kelimenin uzunluğu kadar _ oluşturur ve gizliKelime
değişkenine aktardım
    gizliKelime += "_ "
}
txtViewKelime.text = gizliKelime
```

5-oluşturduğum bu gizlenmiş kelimeyi, “gizliKelime” isimli değişkende tutuyorum

6- Kullanıcı oyuna girdiği zaman, bu gizliKelimeyi ekrana yazdırıyorum. Kullanıcının harf girmesi için bir adet textbox ve gönder butonu bulunmakta.



7-Kullanıcı gönder butonuna bastığı zaman. Textboxa girilen harf uzunluğunu kontrol ediyorum. Eğer 1 karakterden fazla yazdıysa Toast komutu ile hata mesajı gönderiyorum

```
if (txtHarf.text.length > 1) //TEXTBOXA KAÇ KARAKTER GİRDİĞİNİ KONTROL EDİYORUM.  
1 karakterden fazla girdiyse hata veriyorum  
{Toast.makeText(applicationContext, "LÜTFEN TEK HARF GİRİN", Toast.LENGTH_SHORT)  
.show()}}
```

8-Kullanıcının girmiş olduğu tüm harfleri girilenKarakterList değişkeninde tutuyorum.

```
girilenKarakterList.add(txtHarf.text.single()) //girilen karakteri  
girilenKarakterList'ime aktardım
```

9-Kullanıcının girmiş olduğu karakterleri girilenKarakterListem üzerinden for döngüsüyle kontrol ediyorum. Eğer girmiş olduğu karakter daha önceden girilmişse toast mesajıyla hata gönderiyorum

```
for (i in girilenKarakterList) //Aynı harfin girilip girilmediğini kontrol  
ediyorum  
{  
    if (i == txtHarf.text.single())  
    { aynıHarfMi = true  
      Toast.makeText(  
          applicationContext,  
          "AYNI HARFI TEKRAR GİREMEZSİNİZ",  
          Toast.LENGTH_SHORT  
      ).show())  
    }  
}  
if (aynıHarfMi == false) {
```

10-eğer kullanıcının girdiği harf, tek harfse ve daha önce girilmemişse, bu yeni harfi girilenKarakterList'ime ekliyorum. Ve girdiği harfin. Gizli kelime içerisinde olup olmadığını kontrol ediyorum.

11-Kullanıcının girmiş olduğu tüm harfleri tuttuğum girilenKarakterList'imdeki harfleri, gizli kelimem üzerindeki harflerle uyuşup uyuşmadığını iç içe for döngüsü ile tek tek kontrol ediyorum.

12-Eğer girilen harfler, gizli kelimemdeki harflerde bulunuyorsa. Bulunduğu konumları gizliKelime değişkenim üzerinden ekliyorum

(örneğin kelime elma ise, girilen harfler de 'a' ve 'e' ise : gizliKelime değişkenim: e _ _ a olarak düzenlenir.)

13- Eğer gizlikelime değişkenimdeki veri, içerisindeki boşlukları kaldırdıktan sonra eşleşiyorsa. Oyunu kazanmış olduğunu anlıyorum

(örneğin kelime elma ise, girilen harfler 'e' 'a' 'm' 'l' ise: gizlikelime değişkenim: e l m a olacak. Kelime= elma, gizlikelime= e l m a olduğu için gizlikelimedeki boşlukları kaldırınca kelimem ile eşleşiyor. Yani oyunu kazanmış oluyor.)

```

for (k in kelime) { //kelime değişkenimdeki her harfi tek tek kontrol ediyorum
    kontrol = false //girilen karakter kelime içerisinde olup olmadığını kontrol ederken kullanıyorum
    for (l in girilenKarakterList) { //kullanıcının girdiği tüm harfleri, kelimedeki harflerle uyup uymadığını kontrol ediyorum

        if (k == l && kontrol == false) { //eğer kelimedeki harf, girilen karakterlerden biriyle uyuyorsa. O harfin görünümünü açıyorum.
            gizliKelime += k + " " //açılan karakteri gizli kelimeye ekliyorum
            kontrol = true
            dogruHarfler.add(k) //girilen karakter doğru olduğu için dogruharfler listeye ekliyorum.
            if (gizliKelime.replace(" ", "") == kelime) { //gizli kelimenin arasındaki boşlukları kaldırınca. Hafızadaki kelime ile aynı oluyorsa. Oyunu kazanmış oluyorum
                Toast.makeText(
                    applicationContext,
                    "TEBRİKLER OYUNU KAZANDINIZ",
                    Toast.LENGTH_SHORT
                ).show()
            }
            break
        }
    }
}
if (kontrol == false) { //eğer girilen harf, kelime içerisinde yoksa gizliKelime değişkenimin o kısmına "_" ekliyorum
    gizliKelime += "_ "
}
}
}

```

DatabaseHelper

```
class DBHelper(mainActivity: MainActivity) {
    class DBHelper(val context: Context) :
        SQLiteOpenHelper(context, DBHelper.DATABASE_NAME, null, DBHelper.DATABASE_VERSION) {
            private val TABLE_NAME = "Kelimeler" // KELİMELER TABLOSU OLUŞTURDUM
            private val COL_ID = "id" //KELİMELER TABLOSUNA ID SÜTUNU
            private val COL_KELIME = "kelime" //KELİMELER TABLOSUNDA KELİMELERİ
            private val COL_KELIME = "kelime" //KELİMELER TABLOSUNDA KELİMELERİ

            companion object {
                private val DATABASE_NAME = "SQLITE_DATABASE"//database adı
                private val DATABASE_VERSION = 1
            }

            override fun onCreate(db: SQLiteDatabase?) { //EĞER VERİTABANINDA
                'Kelimeler' ADINDA TABLO YOKSA AŞAĞIDAKİ TABLO OLUŞTURMA KODU ÇALIŞACAK
                val createTable = "CREATE TABLE $TABLE_NAME ($COL_ID INTEGER PRIMARY
                KEY AUTOINCREMENT, $COL_KELIME VARCHAR(256))"
                db?.execSQL(createTable)
            }

            override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion:
            Int) {
            }

            fun KelimeEkle(kelime: Kelimeler){ //Veritabanına Ekleme Komutu
                val sqliteDB = this.writableDatabase
                val contentValues = ContentValues()
                contentValues.put(COL_KELIME , kelime.kelime) //Metoduma kelen kelime
                sınıfının içindeki kelimeyi sql ime kaydettim (id yi almadım çünkü otomatik
                artıyor)

                val result = sqliteDB.insert(TABLE_NAME,null,contentValues)

                Toast.makeText(context,if(result != -1L) "Kayıt Başarılı" else "Kayıt
                yapılamadı.", Toast.LENGTH_SHORT).show()
            }

            fun KelimeleriGetir():MutableList<Kelimeler>{ // veritabanındaki tüm
            kelimeleri çekme komutu
                val kelimeList = mutableListOf<Kelimeler>() //kelimeleri bu listede
                tutuyorum

                val sqliteDB = this.readableDatabase
                val query = "SELECT * FROM $TABLE_NAME"
                val result = sqliteDB.rawQuery(query,null)
                if(result.moveToFirst()){
                    do {
                        val kelime = Kelimeler()
                        kelime.id =
                        result.getString(result.getColumnIndex(COL_ID)).toInt()
                        kelime.kelime =
                        result.getString(result.getColumnIndex(COL_KELIME))
                        kelimeList.add(kelime)
                    } while (result.moveToNext())
                }
            }
        }
}
```

```

        }while (result.moveToNext())
    }
    result.close()
    sqliteDB.close()
    return kelimeList
}

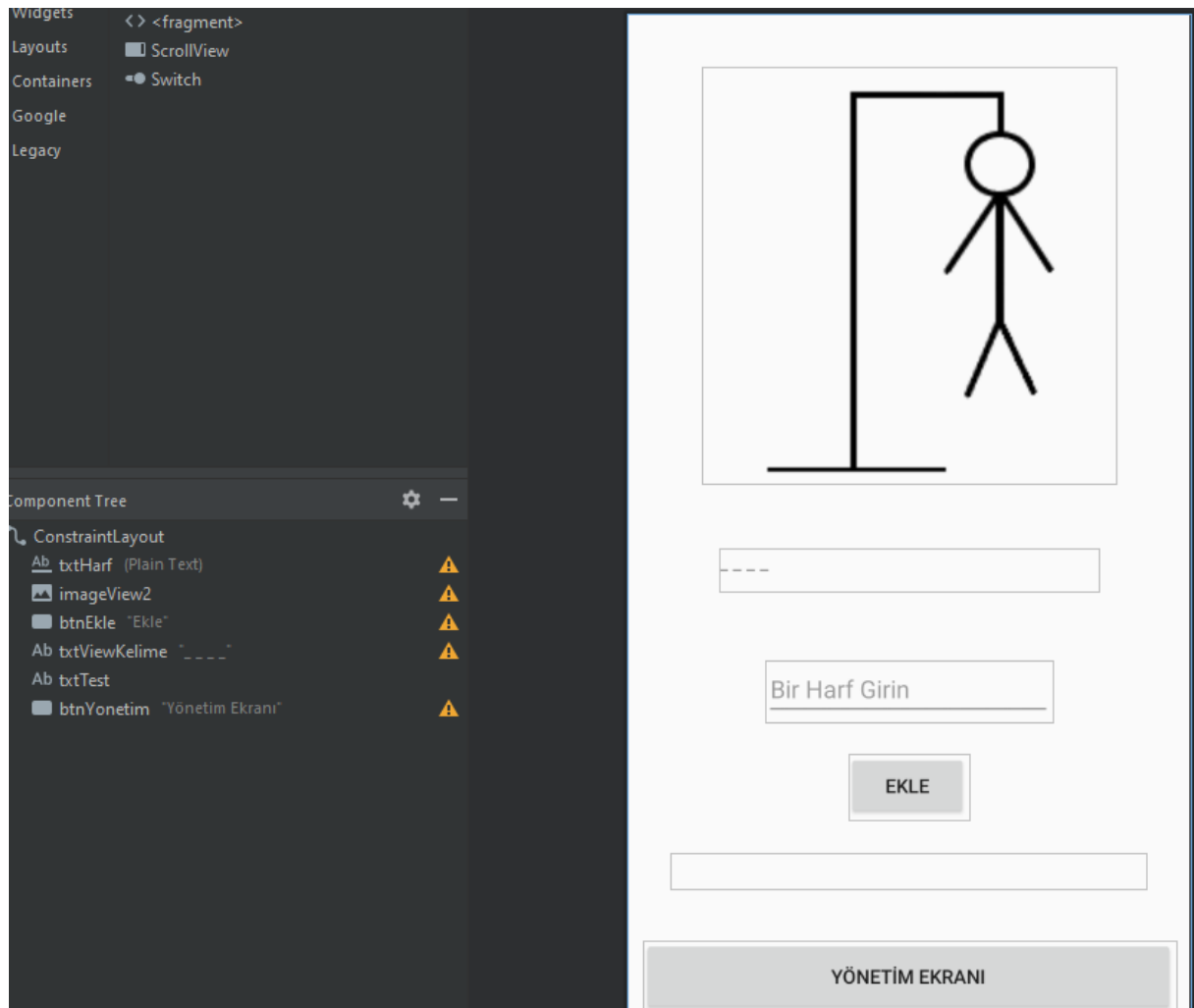
fun KelimeleriSil(){ //Tüm verileri silme komutu
    val sqliteDB = this.writableDatabase
    sqliteDB.delete(TABLE_NAME,null,null)
    sqliteDB.close()
}

}

data class Kelimeler (var id:Int = 0,var kelime:String = "") // veri düzenini
ayarladığımız bir model sınıfı

```

activity_oyun_ekrani



Kodları

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

    <EditText
        android:id="@+id/txtHarf"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp"
        android:ems="10"
        android:hint="Bir Harf Girin"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toTopOf="@+id/btnEkle"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="302dp"
        android:layout_height="0dp"
        android:layout_marginTop="39dp"
        android:layout_marginBottom="129dp"
        app:layout_constraintBottom_toTopOf="@+id/txtHarf"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/resim10" />

    <Button
        android:id="@+id/btnEkle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="143dp"
        android:text="Ekle"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtHarf" />

    <TextView
        android:id="@+id/txtViewKelime"
        android:layout_width="277dp"
        android:layout_height="32dp"
        android:text=" _ _ _ _ "
        app:layout_constraintBottom_toTopOf="@+id/txtHarf"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />

    <TextView
```

```

        android:id="@+id/txtTest"
        android:layout_width="347dp"
        android:layout_height="26dp"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnEkle" />

<Button
    android:id="@+id/btnYonetim"
    android:layout_width="389dp"
    android:layout_height="55dp"
    android:text="Yönetim Ekranı"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

OyunEkranı Class

```

class OyunEkranı : AppCompatActivity() {
    val db by lazy { DBHelper.DBHelper(this) }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_oyun_ekrani)

        var gizliKelime = ""

        imageView2.setImageResource(R.drawable.resim0)

        var kelimelist = db.KelimeleriGetir()// veritabanımdaki tüm kelimeleri
        kelimelist Değişkenine aktardım
        var rnd = (0..kelimelist.count()).random() //listeden rastgele bir kelime
        aldım
        var kelime = kelimelist[rnd].kelime.toString() //kelime listesinden
        randomla gelen indeksteki kelimeyi çektim
        var kontrol = false //girilen karakter kelime içerisinde olup olmadığını
        kontrol ederken kullanıyorum
        var girilenKarakterList = mutableListOf<Char>() // Girilen karakter list.
        var dogruHarfler= mutableListOf<Char>() //kelimede bulunan doğru harfleri
        burada tutuyorum
        var yanlisTahminSayisi = 0 // yanlış tahmin girildiğinde burası artar

        for (k in kelime) { //gelen kelimenin uzunluğu kadar _ oluşturur ve
        gizliKelime değişkenine aktardım
            gizliKelime += "_"
        }
        txtViewKelime.text = gizliKelime // _ leri textviewime yazdırdım

        var tahminYanlisMi = false; //textboxa yazılan kelimenin doğruluğunu
        kontrol ederken kullanıyorum
        var ayniHarfMi = false //textboxa yazılan kelimenin daha önceden girilip
        girilmediğini kontrol ederken kullanıyorum

        btnEkle.setOnClickListener { //Ekle butonuna tıklandığı zaman

```



```

        aynıHarfMi = false //aşağıda girilen harfi kontrol ederken. eğer aynı
        harfse bu değişkeni true olarak değiştireceğim.

        if (txtHarf.text.length > 1) //TEXTBOXA KAÇ KARAKTER GİRDİĞİNİ
        KONTROL EDİYORUM. 1 karakterden fazla girdiyse hata veriyorum
        {Toast.makeText(applicationContext, "LÜTFEN TEK HARF GİRİN",
        Toast.LENGTH_SHORT)
        .show()}
        else {

            for (i in girilenKarakterList) //Aynı harfin girilip
            girilmediğini for döngüsüyle kontrol ediyorum
            {
                if (i == txtHarf.text.single())
                { aynıHarfMi = true
                Toast.makeText(
                    applicationContext,
                    "AYNI HARFİ TEKRAR GİREMEZSİNİZ",
                    Toast.LENGTH_SHORT
                ).show()}
            }
            if (aynıHarfMi == false) { // aynı harf girilmediyse çalışır

                girilenKarakterList.add(txtHarf.text.single()) //girilen
                karakteri girilenKarakterList'ime aktardım
                // txtTest.text=girilenKarakterList.toString()

                gizliKelime = "" //gizli kelimemi aşağıda tekrar oluşturacağım
                için içini boşaltıyorum

                tahminYanlisMi = true; // varsayılan olarak tahmini yanlış
                olarak belirtiyorum. Aşağıda for döngüsüyle kontrol ettikten sonra tahminin doğru
                olduğunu anlarsam bu değer false olacak;
                for (k in kelime) { //kelime değişkenimdeki her harfi tek tek
                kontrol ediyorum

                    kontrol = false //girilen karakter kelime içerisinde olup
                    olmadığını kontrol ederken kullanıyorum
                    for (l in girilenKarakterList) { //kullanıcının girdiği
                    tüm harfleri, kelimemdeki harflerle uyuşup uyuşmadığını kontrol ediyorum

                        if (k == l && kontrol == false) { //eğer kelimemdeki
                        harf, girilenkarakterlerden biriyle uyuyor. O harfin görünümünü açıyorum.
                        gizliKelime += k + " " //açılan karakteri
                        gizlikelimeme ekliyorum

                            kontrol = true
                            dogruHarfler.add(k) //girilen karakter doğru
                            olduğu için dogruharfler listeme ekliyorum.
                            if (gizliKelime.replace(" ", "") == kelime) {
                                //gizli kelimemin arasındaki boşlukları kaldırınca. Hafızadaki kelime ile aynı
                                oluyorsa. Oyunu kazanmış oluyorum

                                    Toast.makeText(
                                        applicationContext,
                                        "TEBRİKLER OYUNU KAZANDINIZ",
                                        Toast.LENGTH_SHORT
                                    ).show()
                                }
                            }
                        break
                    }
                }
            }
        }
    }
}

```

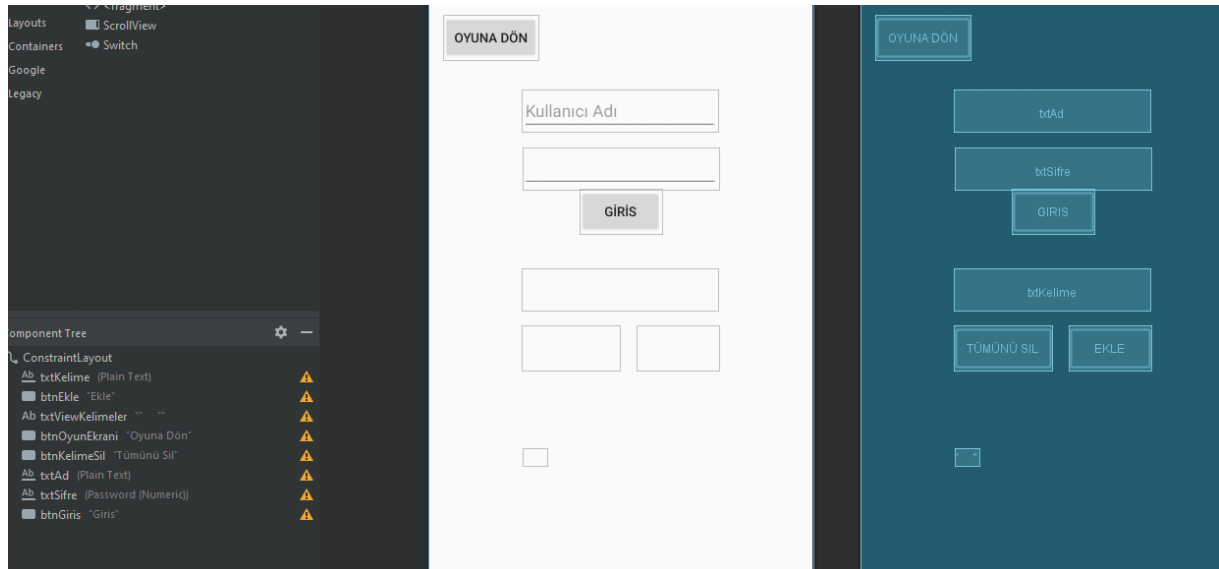
```

    }
    }
    if (kontrol == false) { //eğer girilen harf, kelimem
içerisinde yoksa gizliKelime değişkenimin o kısmına "_" ekliyorum
        gizliKelime += "_"
    }
}
for (l in dogruHarfler) //girilen harf, doğru harfler
listemde var mı diye kontrol ediyorum. Eğer yoksa, adasmaca karakterinin 1 canı
azalmış olacak.
{
    if(l==txtHarf.text.single())
        tahminYanlisMi=false
}
if (tahminYanlisMi == true) { //eğer kullanıcının tahmini
yanlışsa, her yanlış hak için ayrı fotoğraf tanımladım. 10 kere yanlış yaparsa
kaybediyor.
    yanlisTahminSayisi++
    when (yanlisTahminSayisi) {
        1 -> imageView2.setImageResource(R.drawable.resim1)
        2 -> imageView2.setImageResource(R.drawable.resim2)
        3 -> imageView2.setImageResource(R.drawable.resim3)
        4 -> imageView2.setImageResource(R.drawable.resim4)
        5 -> imageView2.setImageResource(R.drawable.resim5)
        6 -> imageView2.setImageResource(R.drawable.resim6)
        7 -> imageView2.setImageResource(R.drawable.resim7)
        8 -> imageView2.setImageResource(R.drawable.resim8)
        9 -> imageView2.setImageResource(R.drawable.resim9)
        10 -> {
            imageView2.setImageResource(R.drawable.resim10)
            Toast.makeText(
                applicationContext,
                "KAYBETTİNİZ",
                Toast.LENGTH_SHORT
            ).show()
            txtHarf.visibility = View.INVISIBLE
        }
    }
}
}
txtViewKelime.text = gizliKelime
}
}
}
}

btnYonetim.setOnClickListener {
    val intent = Intent(this,MainActivity::class.java) //diğer sayfaya
geçiş kodu
    startActivity(intent)
}
}
}

```

activity_main.xml



Kodları

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/txtKelime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="84dp"
        android:ems="10"
        android:hint="Kelime Ekle"
        android:inputType="textPersonName"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtSifre"
        app:layout_constraintVertical_chainStyle="packed" />
    <Button
        android:id="@+id/btnEkle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Ekle"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="@+id/txtKelime"
        app:layout_constraintTop_toBottomOf="@+id/txtKelime" />
    <TextView
        android:id="@+id/txtViewKelimeler"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="101dp"
        android:layout_marginLeft="101dp"
```

```

        android:layout_marginTop="83dp"
        android:text=""
        android:visibility="invisible"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnEkle" />
<Button
    android:id="@+id/btnOyunEkrani"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:text="Oyuna Dön"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/btnKelimeSil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Tümünü Sil"
    android:visibility="invisible"
    app:layout_constraintStart_toStartOf="@+id/txtKelime"
    app:layout_constraintTop_toBottomOf="@+id/txtKelime" />
<EditText
    android:id="@+id/txtAd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="32dp"
    android:ems="10"
    android:hint="Kullanıcı Adı"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toTopOf="@+id/txtSifre"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnOyunEkrani" />

<EditText
    android:id="@+id/txtSifre"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:inputType="numberPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtAd" />
<Button
    android:id="@+id/btnGiris"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Giris"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtSifre" />

```

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

Main Activity

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

        val db by lazy { DBHelper.DBHelper(this) } //veritabanına erişim yetkisi
        veriyorum
        btnEkle.setOnClickListener {

            if (txtKelime.text!=null){
                db.KelimeEkle(kelime = Kelimeler(kelime = txtKelime.text.toString()))
                kelimeleriYazdir(db.KelimeleriGetir())}
        }
        btnGiris.setOnClickListener {
            if (txtAd.text.toString()=="beyza" && txtSifre.text.toString()=="123")
            //kullanıcı adı şifre doğru mu diye kontrol ediyorum
            {
                txtAd.visibility=View.INVISIBLE;
                txtSifre.visibility=View.INVISIBLE;
                btnGiris.visibility=View.INVISIBLE;

                txtKelime.visibility=View.VISIBLE
                btnKelimeSil.visibility=View.VISIBLE
                btnEkle.visibility=View.VISIBLE
                txtViewKelimeler.visibility=View.VISIBLE
            }
            else
                Toast.makeText(applicationContext, "Kullanıcı Adı Veya Parola Hatalı",
                Toast.LENGTH_SHORT).show()
        }

        btnKelimeSil.setOnClickListener {
            db.KelimeleriSil()
        }
        btnOyunEkranı.setOnClickListener {
            val intent = Intent(this,OyunEkranı::class.java)

            startActivity(intent)
        }

    }

    fun kelimeleriYazdir(list:MutableList<Kelimeler>){ //veritabanında kayıtlı
    olan kelimeleri çekiyorum
        txtViewKelimeler.text = ""
        list.forEach {
            txtViewKelimeler.text = txtViewKelimeler.text.toString() + "\n" +
it.kelime
        }

    }

}
```