



— İSTANBUL —
OKAN ÜNİVERSİTESİ
OKAN ÜNİVERSİTESİ

ANDROID UYGULAMA GELİŞTİRME
FİNAL ÖDEVİ
İNGİLİZCE KELİME ÖĞRETİCİ KARTLAR

19MY93019
ABDULBAKİ ZIRIH

Android Studio, Kotlin ile geliştirilmiş ingilizce kelime ezberlenmesi için kelime kartlarının yönetilebildiği bir uygulama.

İçindekiler

1 Giriş.....	3
1.1 Projede Kullanılan Teknolojiler.....	3
1.2 Projede Kullanılan Bağımlılıklar.....	3
2 Uygulamanın Geliştirilmesi.....	4
1.1 Tema Ayarları.....	4
1.2 İzinler.....	5
1.3 Oturum Ekranı.....	6
1.4 Ana Ekran.....	7
1.5 Veritabanı SQLite3.....	8
1.6 Kelime Kartları.....	9
3 Sonuç.....	10

1 Giriş

1.1 Projede Kullanılan Teknolojiler

- Android Studio
- Kotlin
- XML
- SQLite3, Room

1.2 Projede Kullanılan Bağımlılıklar

Proje kapsamında bir çok bağımlılık kullanılmıştır. Bunlar animasyon, api şeklinde devam etmektedir.

```
dependencies {  
    def room_version = "2.2.6"  
    def dynamicanimation_version = "1.0.0"  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.3.2'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'com.google.android.material:material:1.2.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    implementation "androidx.room:room-runtime:$room_version"  
    kapt "androidx.room:room-compiler:$room_version"  
    implementation "androidx.dynamicanimation:dynamicanimation:$dynamicanimation_version"  
    implementation 'org.jsoup:jsoup:1.13.1'  
    //noinspection GradleCompatible  
    implementation 'com.android.support:support-compat:28.0.0'  
}
```

2 Uygulamanın Geliştirilmesi

1.1 Tema Ayarları

Tema da yapılan değişiklikler "ActionBar"ın kaldırılması ve uygulamanın tam ekran olmasıdır.

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.İngilizceKelimeÖğreticiKartlar" parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/white</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_700</item>
    <item name="colorOnSecondary">@color/black</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
    <item name="android:windowFullscreen">true</item>
  </style>
</resources>
```

1.2 İzinler

Android işletim sistemi üzerinde bazı önemli özelliklerin kullanılabilmesi için izin istenmesi gerekmektedir. Bu uygulamada internet üzerinden kelimeler çekildiğinden internet izni istenmiştir.

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.İngilizceKelimeÖğreticiKartlar">
        <activity android:name=".MainActivity" />
        <activity android:name=".SessionActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

1.3 Oturum Ekranı

Oturum ekranı için bir adet activity oluşturulup hem giriş hemde kayıt işlemleri ilgili activity üzerinden yapılmaktadır. Kayıt oluşturmak için kullanıcı adı ve şifreye ek olarak e-posta alanı eklenmektedir. E-posta alanı animasyonlu bir şekilde "SpringAnimation" kullanılarak eklenmektedir.

```
private fun signIn() {
    val iUsername = username.text.toString()
    val iPassword = password.text.toString()

    if (iUsername.isEmpty() || iPassword.isEmpty()) {
        Toast.makeText(applicationContext, "Lütfen tüm alanları doldurun.", Toast.LENGTH_LONG).show()
    } else {
        val db = Room.databaseBuilder(
            this,
            AppDatabase::class.java, "app"
        ).fallbackToDestructiveMigration().allowMainThreadQueries().build()
        val user = db.userDao().getUser(iUsername)

        if (user == null || user.password != iPassword) {
            Toast.makeText(applicationContext, "Hatalı kullanıcı adı veya şifre.", Toast.LENGTH_LONG).show()
        } else {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }
    }
}
```

1.4 Ana Ekran

Ana Ekran da bir çok işlem yapılmaktadır. Bu işlemler şu şekildedir.

- Kelime Seçme (İnternette Çekilen Veriler)
- Kartı sağa sürükleme(Güncelleme)
- Kartı sola sürükleme(Silme)

```
override fun onTouchEvent(event: MotionEvent): Boolean {
    if (words.isEmpty()) return true
    when (event.action) {
        MotionEvent.ACTION_DOWN → {
            touchX = event.x
            cardTouchX = wordCard.x
        }
        MotionEvent.ACTION_UP → {
            val db = Room.databaseBuilder(
                this,
                AppDatabase::class.java, "app"
            ).allowMainThreadQueries().build()
            val x = wordCard.x + (wordCard.width / 2)
            val word = words[activeIndex]
            if (x < 0) {
                SpringAnimation(wordCard, DynamicAnimation.TRANSLATION_X, -abs(screenWidth).toFloat()).start()
                db.wordDao().delete(word.id!!)
                words.remove(word)

                when {
                    words.isEmpty() → {
                        activeIndex = 0
                    }
                    activeIndex < words.size → {
                        setWord(words[activeIndex])
                    }
                    else → {
                        activeIndex = 0
                        setWord(words[activeIndex])
                    }
                }
            } else if (x > screenWidth) {
                SpringAnimation(wordCard, DynamicAnimation.TRANSLATION_X, screenWidth.toFloat()).start()
                db.wordDao().changeStatus(word.id!!, 1)
                word.status = 1

                if (activeIndex + 1 < words.size) {
                    setWord(words[++activeIndex])
                } else {
                    activeIndex = 0
                    setWord(words[activeIndex])
                }
            }
        }
        MotionEvent.ACTION_MOVE → {
            wordCard.x = event.x - touchX!! + cardTouchX
        }
    }
    return super.onTouchEvent(event)
}
```

1.5 Veritabanı SQLite3

Veritabanı için roomlar kullanılmıştır.

```
package english.words.database

import androidx.room.*

@Dao
interface UserDao {
    @Query("SELECT * FROM User WHERE username=:username LIMIT 1")
    fun getUser(username: String): User?

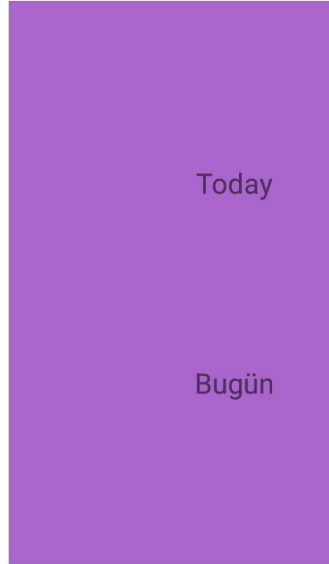
    @Insert
    fun add(vararg user: User)

    @Query("SELECT * FROM User")
    fun getUsers(): List<User>
}

@Entity
data class User (
    @PrimaryKey(autoGenerate=true) var id: Int? = null,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "username") val username: String?,
    @ColumnInfo(name = "password") val password: String?
)
```


1.6 Kelime Kartları

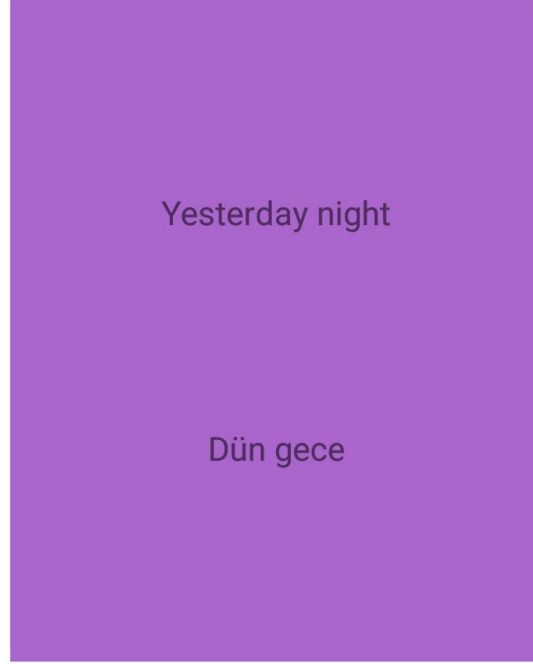
Kelime kartlarını sürüklemenin işlemi için onTouchEvent metodu kullanılmıştır.



ÖRNEK KELİMELER

3 Sonuç

Sonuç olarak uygulama kelime kartlarının eklenip silinebildiği kelime ezberlemek için geliştirildi.



Bu rapor dosyasında temel anlamda uygulamanın geliştirilme süreci ve kullanılan teknolojilere değinilmiştir.