****

**İsim Soyisim: Salih Karakaya**

**Öğrenci No: 191816775**

1. **Ödeve ait isterler ve bu isterlerin açıklanması**

Bu mobil uygulama ödevi kapsamında, kullanıcıların kişisel bilgilerini girmesine ve bu bilgilerin cihazda saklanarak görsel şekilde listelenmesine olanak sağlayan bir Android uygulaması geliştirilmiştir. Uygulamanın gerçekleştirmesi gereken temel işlevsel isterler aşağıda açıklanmıştır:

#### **1.1. Kullanıcı Bilgisi Girişi**

Kullanıcının ad, soyad, doğum tarihi, cinsiyet, hobiler ve iletişim bilgilerini girebileceği bir arayüz tasarlanmalıdır.  
Bu alanlarda kullanıcıdan doğru ve okunabilir veri alınabilmesi için TextInputLayout ve TextInputEditText bileşenleri kullanılmıştır. Bu sayede kullanıcı arayüzü modern ve kullanıcı dostu hale getirilmiştir.

#### **1.2. Cinsiyet Seçimi**

Kullanıcının cinsiyet bilgisi seçimli (opsiyonel) olarak girilmelidir. Bu işlem RadioGroup ve RadioButton bileşenleri ile gerçekleştirilmiştir. Kullanıcı, "Erkek" ya da "Kadın" seçeneklerinden birini seçerek cinsiyetini belirleyebilir.

#### **1.3. Doğum Tarihi Seçimi**

Doğum tarihi manuel girilmek yerine daha güvenli ve kullanıcı dostu olması için DatePickerDialog ile seçilmiştir. Böylece kullanıcı, takvim üzerinden bir tarih seçerek hatalı giriş yapma riskinden korunmuştur.

#### **1.4. Verilerin Kaydedilmesi**

Kullanıcının girdiği tüm bilgiler, SharedPreferences yapısı ile cihazda kalıcı olarak saklanmaktadır. Uygulama kapatılsa bile bilgiler saklı kalır ve yeniden açıldığında gösterilir.

#### **1.5. Bilgilerin Listelenmesi**

Girilen tüm kullanıcı bilgileri şık bir şekilde ekranda görüntülenmelidir. Bu amaçla özel tasarlanmış bir ListView kullanılmıştır. Her kullanıcı bilgisi, kart görünümlü bir yapıda listelenmektedir.

#### **1.6. Liste Elemanına Tıklama**

Kullanıcı, listelenmiş bir bilgiye tıkladığında detayları görmek isterse, AlertDialog ile tüm bilgiler ayrı bir pencerede sunulur.

#### **1.7. Liste Elemanını Uzun Basarak Silme**

Kullanıcı, bir bilgiyi silmek istediğinde o elemana uzun basarak silme işlemini gerçekleştirebilir. Silme işlemi öncesinde kullanıcıya onay sorulur ve ardından hem listeden hem de SharedPreferences’tan veri silinir.

1. **Var ise XML kodlarının metin olarak yapıştırılması ve açıklanması**

**activity\_main**

**<?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"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16sp">  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/gazi\_logo"  
 android:alpha="0.25"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/inputLayoutAd"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Ad"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/editTextAd"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/inputLayoutSoyad"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Soyad"  
 app:layout\_constraintTop\_toBottomOf="@id/inputLayoutAd"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="8sp">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/editTextSoyad"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/inputLayoutDogum"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
  
  
 app:layout\_constraintTop\_toBottomOf="@id/inputLayoutSoyad"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="8sp">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/editTextDogum"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:focusable="false"  
 android:clickable="true"  
 android:hint="Doğum Tarihi"  
 android:inputType="none"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <RadioGroup  
 android:id="@+id/radioGroupCinsiyet"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintTop\_toBottomOf="@id/inputLayoutDogum"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:layout\_marginTop="12sp">  
  
 <RadioButton  
 android:id="@+id/radioErkek"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Erkek" />  
  
 <RadioButton  
 android:id="@+id/radioKadin"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Kadın" />  
 </RadioGroup>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/inputLayoutHobi"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Hobiler"  
 app:layout\_constraintTop\_toBottomOf="@id/radioGroupCinsiyet"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="12sp">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/editTextHobi"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/inputLayoutIletisim"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="İletişim"  
 app:layout\_constraintTop\_toBottomOf="@id/inputLayoutHobi"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="8sp">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/editTextIletisim"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/buttonKaydet"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text="Kaydet"  
 android:backgroundTint="@android:color/holo\_blue\_light"  
 app:layout\_constraintTop\_toBottomOf="@id/inputLayoutIletisim"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="12sp" />  
  
 <ListView  
 android:id="@+id/listViewBilgiler"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:divider="@android:color/darker\_gray"  
 android:dividerHeight="1sp"  
 app:layout\_constraintTop\_toBottomOf="@id/buttonKaydet"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="16sp"/>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

Bu XML dosyası, uygulamanın ana ekranına (Main Activity) ait kullanıcı arayüzünü tanımlamaktadır. Tasarım, ConstraintLayout kullanılarak oluşturulmuş olup, aşağıdaki temel bileşenlerden oluşmaktadır:

1. **ImageView:**  
   Arka planda, uygulamaya ait logo (gazi\_logo) düşük opaklık (alpha="0.25") değeriyle gösterilerek tasarıma sade bir arka plan efekti verilmiştir.
2. **TextInputLayout ve TextInputEditText:**  
   Kullanıcının **ad**, **soyad**, **doğum tarihi**, **hobi** ve **iletişim** bilgilerini girmesi için modern ve kullanışlı giriş alanları tasarlanmıştır. Bu bileşenler, Google Material Design yapısına uygun bir görünüm sunar.
3. **editTextDogum:**

Doğum tarihi giriş alanı elle yazılmak yerine, kullanıcıya takvim seçimi sunmak amacıyla focusable="false" ve clickable="true" olarak ayarlanmıştır. Bu alan, tıklanabilir hale getirilmiş ve Java tarafında DatePickerDialog ile ilişkilendirilmiştir.

1. **RadioGroup ve RadioButton:**

Kullanıcının cinsiyet bilgisini seçebilmesi için iki seçenek sunulmuştur: "Erkek" ve "Kadın". RadioGroup, bu iki seçeneğin aynı anda seçilmesini engeller.

1. **Button:**

Kullanıcı formu doldurduktan sonra bilgileri kaydetmek için "Kaydet" butonu eklenmiştir. Arka plan rengi açık mavi (holo\_blue\_light) olarak ayarlanarak görsel vurgu sağlanmıştır.

1. **ListView:**

Kayıt edilen kullanıcı bilgilerinin listelenmesi için ListView bileşeni kullanılmıştır. Divider rengi darker\_gray olarak belirlenmiş ve satır ayrımı için görsel netlik sağlanmıştır.

**Tasarımdaki Ekstra Detaylar:**

* Tüm öğeler ConstraintLayout içerisinde hizalanarak hem dikey hem yatay konumlamalar yapılmıştır.
* Bileşenler arası boşluklar için layout\_marginTop="8sp" gibi değerlerle arayüz estetiği sağlanmıştır.
* Arka plan görselinin üzerine diğer bileşenler şeffaf şekilde yerleştirilmiştir.

**list\_item**

**<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="8dp"  
 card\_view:cardCornerRadius="12dp"  
 card\_view:cardElevation="6dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="12dp"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/textViewBilgi"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Kullanıcı Bilgileri"  
 android:textSize="16sp"  
 android:textColor="#000000" />  
  
 </LinearLayout>  
</androidx.cardview.widget.CardView>**

Uygulama içerisinde kayıtlı kullanıcı bilgilerini daha şık göstermek amacıyla **ListView** için özel bir öğe tasarlanmıştır. Bu öğe, **CardView** yapısı kullanılarak hazırlanmış ve kullanıcı bilgileri kart şeklinde gösterilmiştir.

1. **CardView:**

Kart görünümü sağlayan, yuvarlatılmış köşeler ve gölge (elevation) efektleri ile kullanıcı bilgilerini vurgulayan yapı.

* cardCornerRadius="12dp" → köşeleri yumuşatır.
* cardElevation="6dp" → kartın arka plandan yukarıda görünmesini sağlar.

1. **LinearLayout:**

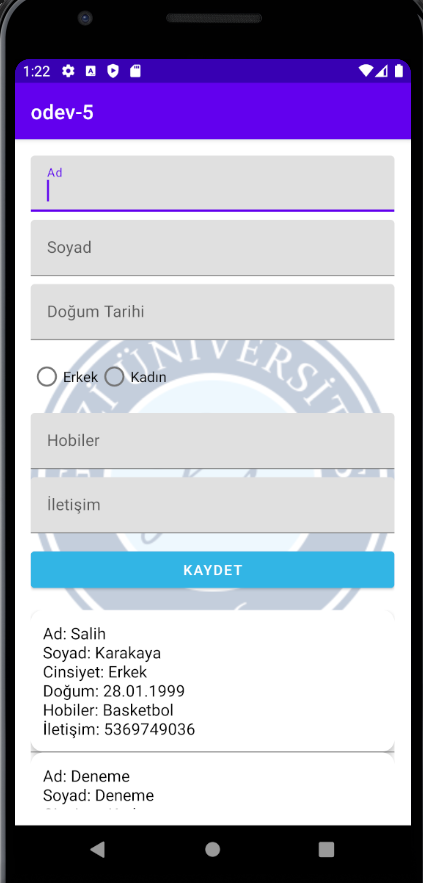
Kartın içinde elemanların dikey olarak sıralanmasını sağlar. Bu durumda her kartta sadece bir TextView olduğundan oldukça sadedir ama geliştirilmeye açıktır.

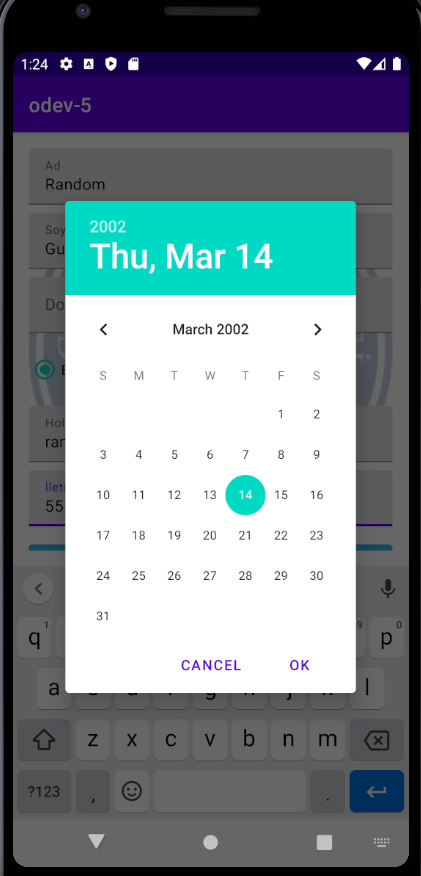
1. **TextView:**

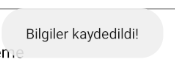
Kullanıcının tüm bilgilerini metin olarak tek bir blokta gösterir.  
Bu TextView, Java kodunda her kullanıcı için oluşturulan metni göstermek üzere textViewBilgi ID’si ile tanımlanmıştır.

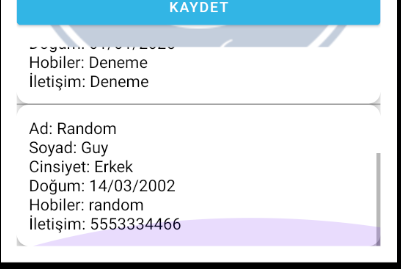
Bu özel tasarım sayesinde uygulamadaki kullanıcı listesi, sade bir metin dizisinden çok daha modern ve okunabilir bir görünüme kavuşmuştur.

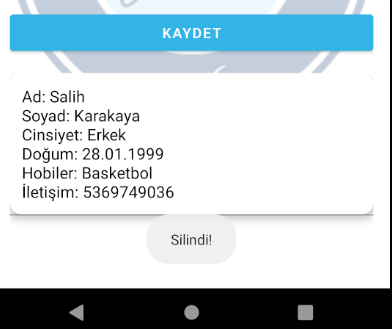
1. **XML kodları sonucunda ortaya çıkan ekran tasarımının ekran görüntüsü**

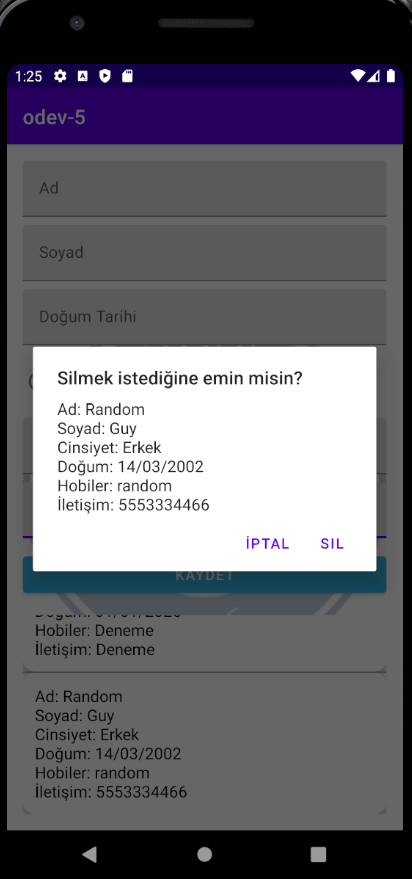
****

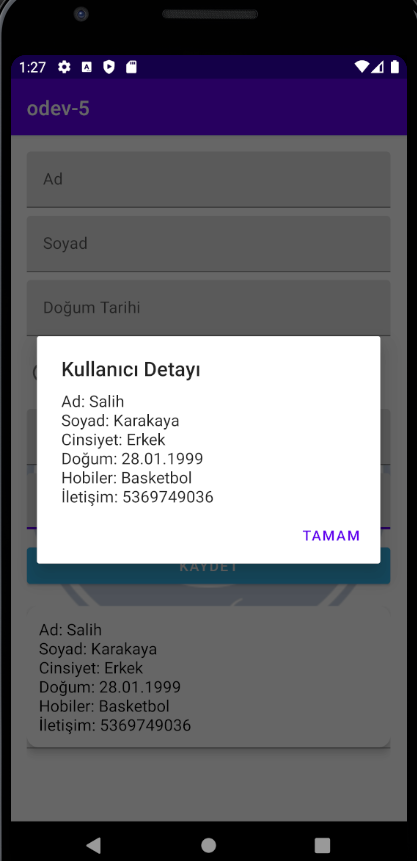
****

****

****

****

****

****

1. **Var ise Java kodlarının (yorum satırına ek olarak) metin halinde yapıştırılması ve bloklar halinde açıklanması**

**Main\_Activity**

**package com.mobilprogramlama.odev\_5;  
  
import android.app.DatePickerDialog;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.\*;  
import androidx.appcompat.app.AppCompatActivity;  
import com.google.android.material.textfield.TextInputEditText;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.HashSet;  
import java.util.Set;  
  
public class Main\_Activity extends AppCompatActivity {  
  
 // Kullanıcıdan alınacak bilgiler için alanlar  
 TextInputEditText editTextAd, editTextSoyad, editTextDogum, editTextHobi, editTextIletisim;  
 RadioGroup radioGroupCinsiyet;  
 Button buttonKaydet;  
 ListView listViewBilgiler;  
  
 // SharedPreferences: kalıcı veri saklama  
 SharedPreferences sharedPreferences;  
  
 // ListView için adapter ve liste  
 ArrayAdapter<String> adapter;  
 ArrayList<String> kullaniciListesi;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*); // activity\_main.xml ile arayüz bağlanır  
  
 // XML ile Java arasında bağlantı kuruluyor  
 editTextAd = findViewById(R.id.*editTextAd*);  
 editTextSoyad = findViewById(R.id.*editTextSoyad*);  
 editTextDogum = findViewById(R.id.*editTextDogum*);  
 editTextHobi = findViewById(R.id.*editTextHobi*);  
 editTextIletisim = findViewById(R.id.*editTextIletisim*);  
 radioGroupCinsiyet = findViewById(R.id.*radioGroupCinsiyet*);  
 buttonKaydet = findViewById(R.id.*buttonKaydet*);  
 listViewBilgiler = findViewById(R.id.*listViewBilgiler*);  
  
 // SharedPreferences başlatılıyor  
 sharedPreferences = getSharedPreferences("KullaniciBilgileri", *MODE\_PRIVATE*);  
 kullaniciListesi = new ArrayList<>();  
  
 // Uygulama açıldığında daha önce kaydedilmiş verileri SharedPreferences'tan yükler  
 Set<String> set = sharedPreferences.getStringSet("veriler", new HashSet<>());  
 kullaniciListesi.addAll(set);  
  
 // Özel tasarımlı custom adapter kullanılıyor  
 adapter = new CustomAdapter(this, kullaniciListesi);  
 listViewBilgiler.setAdapter(adapter);  
  
 // ListView'e tıklanınca detay gösterme (AlertDialog ile)  
 listViewBilgiler.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 String secilenBilgi = kullaniciListesi.get(position);  
  
 new android.app.AlertDialog.Builder(Main\_Activity.this)  
 .setTitle("Kullanıcı Detayı")  
 .setMessage(secilenBilgi)  
 .setPositiveButton("Tamam", null)  
 .show();  
 }  
 });  
  
 // Uzun basıldığında kullanıcıyı silme işlemi  
 listViewBilgiler.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
 String silinecek = kullaniciListesi.get(position);  
  
 new android.app.AlertDialog.Builder(Main\_Activity.this)  
 .setTitle("Silmek istediğine emin misin?")  
 .setMessage(silinecek)  
 .setPositiveButton("Sil", (dialog, which) -> {  
 // Listeden sil  
 kullaniciListesi.remove(position);  
 adapter.notifyDataSetChanged();  
  
 // SharedPreferences'tan güncelleme  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 Set<String> yeniSet = new HashSet<>(kullaniciListesi);  
 editor.putStringSet("veriler", yeniSet);  
 editor.apply();  
  
 Toast.*makeText*(Main\_Activity.this, "Silindi!", Toast.*LENGTH\_SHORT*).show();  
 })  
 .setNegativeButton("İptal", null)  
 .show();  
  
 return true; // Uzun tıklama işlemi tamamlandı  
 }  
 });  
  
 // Kaydet butonuna tıklanınca yapılacak işlemler  
 buttonKaydet.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Kullanıcıdan alınan veriler okunur  
 String ad = editTextAd.getText().toString().trim();  
 String soyad = editTextSoyad.getText().toString().trim();  
 String dogum = editTextDogum.getText().toString().trim();  
 String hobi = editTextHobi.getText().toString().trim();  
 String iletisim = editTextIletisim.getText().toString().trim();  
  
 // Cinsiyet seçim kontrolü yapılır  
 int selectedId = radioGroupCinsiyet.getCheckedRadioButtonId();  
 RadioButton selectedRadio = findViewById(selectedId);  
 String cinsiyet = (selectedRadio != null) ? selectedRadio.getText().toString() : "Belirtilmedi";  
  
 // Tek bir metin haline getirilir  
 String bilgi = "Ad: " + ad + "\nSoyad: " + soyad + "\nCinsiyet: " + cinsiyet +  
 "\nDoğum: " + dogum + "\nHobiler: " + hobi + "\nİletişim: " + iletisim;  
  
 // Listeye eklenir ve güncellenir  
 kullaniciListesi.add(bilgi);  
 adapter.notifyDataSetChanged();  
  
 // SharedPreferences'a kalıcı olarak kaydedilir  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 Set<String> set = new HashSet<>(kullaniciListesi);  
 editor.putStringSet("veriler", set);  
 editor.apply();  
  
 // Giriş alanları temizlenir  
 editTextAd.setText("");  
 editTextSoyad.setText("");  
 editTextDogum.setText("");  
 editTextHobi.setText("");  
 editTextIletisim.setText("");  
 radioGroupCinsiyet.clearCheck();  
  
 Toast.*makeText*(Main\_Activity.this, "Bilgiler kaydedildi!", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 editTextDogum.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Mevcut tarih alınır  
 final Calendar calendar = Calendar.*getInstance*();  
 int year = calendar.get(Calendar.*YEAR*);  
 int month = calendar.get(Calendar.*MONTH*);  
 int day = calendar.get(Calendar.*DAY\_OF\_MONTH*);  
  
 // DatePicker gösterilir  
 DatePickerDialog datePickerDialog = new DatePickerDialog(Main\_Activity.this,  
 (view, selectedYear, selectedMonth, selectedDay) -> {  
 String tarih = String.*format*("%02d/%02d/%04d", selectedDay, selectedMonth + 1, selectedYear);  
 editTextDogum.setText(tarih);  
 }, year, month, day);  
  
 // Maksimum tarih bugünün tarihi olsun (gelecek tarih seçilemesin)  
 datePickerDialog.getDatePicker().setMaxDate(System.*currentTimeMillis*());  
 datePickerDialog.show();  
 }  
 });  
  
 }  
}**

***4.1. Değişken Tanımlamaları ve XML ile Bağlantı***

TextInputEditText editTextAd, editTextSoyad, editTextDogum, editTextHobi, editTextIletisim;

RadioGroup radioGroupCinsiyet;

Button buttonKaydet;

ListView listViewBilgiler;

SharedPreferences sharedPreferences;

ArrayAdapter<String> adapter;

ArrayList<String> kullaniciListesi;

**Açıklama:**Bu blokta, kullanıcıdan alınacak tüm bilgileri tutacak giriş alanları, kaydet butonu ve liste görüntüleme için ListView tanımlanmıştır.  
Ayrıca verilerin kalıcı saklanması için SharedPreferences, liste görüntüleme için ArrayAdapter ve veri listesi olarak ArrayList kullanılmıştır.

***4.2. onCreate() Metodu – İlk Yükleme ve Bileşen Bağlantıları***

sharedPreferences = getSharedPreferences("KullaniciBilgileri", MODE\_PRIVATE);

kullaniciListesi = new ArrayList<>();

Set<String> set = sharedPreferences.getStringSet("veriler", new HashSet<>());

kullaniciListesi.addAll(set);

adapter = new CustomAdapter(this, kullaniciListesi);

listViewBilgiler.setAdapter(adapter);

**Açıklama:**Uygulama başlatıldığında (onCreate içinde) önce XML bileşenleri Java tarafına bağlanır.  
SharedPreferences ile daha önce kayıtlı olan bilgiler alınır ve ListView'e eklenmek üzere listeye aktarılır.  
Özel CustomAdapter ile görsel olarak şık bir listeleme sağlanır.

***4.3. Liste İşlemleri – Tıklama ve Uzun Basma***

listViewBilgiler.setOnItemClickListener(...);

listViewBilgiler.setOnItemLongClickListener(...);

**Açıklama:**

* Tıklama ile detay gösterimi: Kullanıcı liste elemanına tıkladığında AlertDialog ile kayıtlı tüm bilgiler ayrıntılı şekilde gösterilir.
* Uzun basınca silme: Kullanıcı bir kayda uzun süre bastığında, silinmek istenip istenmediği sorulur. Onay verilirse hem listeden hem de SharedPreferences'tan silinir.

***4.4. Kaydet Butonu ile Bilgi Toplama ve Kayıt Etme***

buttonKaydet.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Giriş alanlarından verileri al

// Cinsiyet RadioGroup'undan seçim al

// Bilgileri birleştirip listeye ekle

// SharedPreferences'a kaydet

// Alanları temizle ve kullanıcıya Toast göster

}

});

**Açıklama:**Kullanıcı “Kaydet” butonuna tıkladığında, tüm giriş alanlarından veriler alınır.  
Cinsiyet seçimi kontrol edilerek, kullanıcı bilgileri bir bütün metin olarak ListView’e eklenir. Aynı anda bu bilgiler SharedPreferences ile kalıcı olarak cihazda saklanır.  
Form temizlenir ve kullanıcıya başarı mesajı gösterilir.

**Ek Notlar:**

* DatePickerDialog, editTextDogum alanına uygulanarak manuel tarih girme hataları önlenmiştir.
* CustomAdapter kullanımı, ListView'i daha modern ve görsel olarak zengin hale getirmiştir.
* SharedPreferences ile offline çalışabilirlik sağlanmıştır.

**CustomAdapter**

**package com.mobilprogramlama.odev\_5;**

**import android.content.Context;**

**import android.view.LayoutInflater;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.ArrayAdapter;**

**import android.widget.TextView;**

**import java.util.List;**

**// CustomAdapter, ListView'e özel görünümlü öğeler sağlamak için ArrayAdapter sınıfından türetilmiştir**

**public class CustomAdapter extends ArrayAdapter<String> {**

**// Constructor - Verileri ve bağlamı (Context) alır**

**public CustomAdapter(Context context, List<String> bilgiler) {**

**// Varsayılan layout değeri yerine 0 veriyoruz çünkü kendi layout'umuzu kullanacağız**

**super(context, 0, bilgiler);**

**}**

**// Her bir öğe (satır) ListView'de nasıl görünecek, burada tanımlanır**

**@Override**

**public View getView(int position, View convertView, ViewGroup parent) {**

**// Belirtilen konumdaki veriyi al**

**String bilgi = getItem(position);**

**// Eğer daha önce oluşturulmuş bir görünüm yoksa, yeni bir tane şişir (inflate et)**

**if (convertView == null) {**

**convertView = LayoutInflater.from(getContext()).inflate(R.layout.list\_item, parent, false);**

**}**

**// list\_item.xml'de yer alan TextView'i bul**

**TextView textViewBilgi = convertView.findViewById(R.id.textViewBilgi);**

**// TextView'e alınan veriyi yazdır**

**textViewBilgi.setText(bilgi);**

**// Oluşturulan veya yeniden kullanılan görünümü döndür**

**return convertView;**

**}**

**}**

**Açıklamalar:**

* Bu adapter, ListView'in her satırını bizim özel olarak tasarladığımız list\_item.xml dosyasına göre doldurur.
* convertView'un yeniden kullanılması performans için önemlidir (View Recycling).
* LayoutInflater, XML’deki list\_item.xml dosyasını Java nesnesine dönüştürür.

Bu yapı sayesinde kullanıcı bilgileri sade bir metin listesi yerine, kart şeklinde modern bir görünümle gösterilmiş olur.

1. **Kullanılan ek teknolojiler ve yapıların belirtilmesi ve açıklanması**

Bu mobil uygulama geliştirme sürecinde, Android'in temel bileşenlerine ek olarak aşağıdaki modern yapı ve teknolojiler kullanılmıştır:

***5.1. ConstraintLayout***

* Uygulamanın arayüzü ConstraintLayout kullanılarak geliştirilmiştir.
* Bu yapı, kullanıcı arayüzünün farklı ekran boyutlarında uyumlu ve esnek olmasını sağlar.
* Görsel öğeler birbirine ve üst kapsayıcıya göre hizalanarak daha profesyonel bir tasarım elde edilmiştir.

***5.2. Material Design Bileşenleri***

* TextInputLayout ve TextInputEditText bileşenleri, Google’ın Material Design yapısına uygun modern veri giriş alanları sunar.
* Kullanıcı deneyimini artırmak ve sade bir görünüm sağlamak amacıyla tercih edilmiştir.

***5.3. DatePickerDialog***

* Kullanıcıların doğum tarihi bilgilerini manuel girmek yerine takvim üzerinden seçebilmesini sağlamak için DatePickerDialog kullanılmıştır.
* Bu yapı sayesinde kullanıcıların yanlış tarih girmesi engellenmiş, daha hatasız ve kullanıcı dostu bir deneyim sağlanmıştır.

***5.4. SharedPreferences***

* Kullanıcıdan alınan verilerin kalıcı olarak saklanabilmesi için SharedPreferences yapısı kullanılmıştır.
* Uygulama kapatılsa dahi, daha önce kaydedilen kullanıcı bilgileri tekrar gösterilebilmektedir.
* Basit veri saklama işlemleri için hafif ve hızlı bir çözümdür.

***5.5. Custom Adapter (Özel Liste Görünümü)***

* ListView bileşeni ile veri gösteriminde daha estetik ve işlevsel bir yapı oluşturmak amacıyla CustomAdapter sınıfı yazılmıştır.
* Bu adapter, list\_item.xml dosyasındaki özel kart tasarımını her bir kullanıcı bilgisi için uygular.
* Bu sayede veriler daha okunabilir, görsel olarak zengin ve modern bir biçimde sunulmuştur.

***5.6. CardView***

* Liste elemanları CardView kullanılarak kutu (kart) şeklinde gösterilmiştir.
* CardView; köşeleri yumuşatılmış, gölgeli (elevation) bir yapı sağlayarak görsel olarak sade ve profesyonel bir liste tasarımı sunar.

Bu teknolojiler sayesinde uygulama hem kullanıcı deneyimi açısından başarılı hem de teknik açıdan güncel ve sürdürülebilir bir yapıya kavuşmuştur.