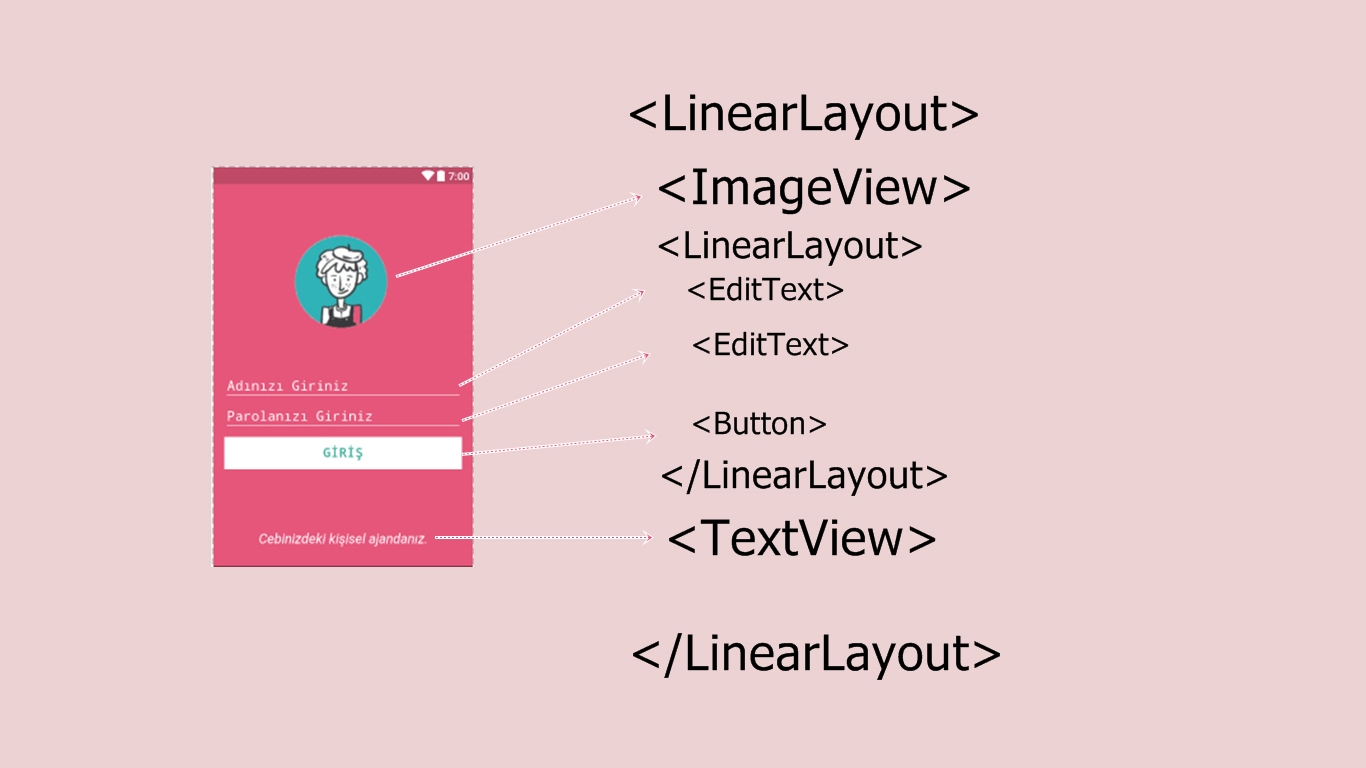
LoginApp  
Android Studio’da uygulamamızı oluşturduğumuzda bizim için varsayılan olarak MainActivity.java ve activity\_main.xml dosyası oluşturulur.  
  
İlk olarak tasarım kısmı için activity\_main.xml’de neler yapmamız gerekiyor onları inceleyelim.  
  


activity\_main.xml’de component hiyerarşisi bu şekilde. Şimdi kodlar üzerinden gerekli açıklamamızı yapalım.  
  
*1<?***xml version="1.0" encoding="utf-8"***?>*2<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
3 xmlns:app="http://schemas.android.com/apk/res-auto"  
4 xmlns:tools="http://schemas.android.com/tools"  
5 android:layout\_width="match\_parent"  
6 android:layout\_height="match\_parent"  
7 android:orientation="vertical"  
8 android:background="#E6567A"  
9 android:padding="16dp"  
10 tools:context="com.hasibezafer.loginapp.MainActivity"**>  
  
  
11 <**ImageView  
12 android:layout\_marginTop="3dp"  
13 android:id="@+id/imageView"  
14 android:backgroundTint="#ffffff"  
15 android:layout\_width="match\_parent"  
16 android:layout\_height="wrap\_content"  
17 app:srcCompat="@drawable/user"** />  
  
18 <**LinearLayout  
19 android:layout\_marginTop="60dp"  
20 android:orientation="vertical"  
21 android:layout\_width="match\_parent"  
22 android:layout\_height="wrap\_content"**>  
  
23 <**EditText  
24 android:id="@+id/etName"  
25 android:layout\_width="match\_parent"  
26 android:layout\_height="wrap\_content"  
27 android:ems="10"  
28 android:inputType="textPersonName"  
29 android:text=""  
30 android:textSize="20dp"  
31 android:fontFamily="monospace"  
32 android:textColor="#ffffff"  
33 android:backgroundTint="#ffffff"  
34 android:textColorHint="#ffffff"  
35 android:hint="Adınızı Giriniz"  
36** />  
  
37 <**EditText  
38 android:id="@+id/etPassword"  
39 android:layout\_width="match\_parent"  
40 android:layout\_height="wrap\_content"  
41 android:ems="10"  
42 android:textSize="20dp"  
43 android:fontFamily="monospace"  
44 android:textColor="#ffffff"  
45 android:backgroundTint="#ffffff"  
46 android:textColorHint="#ffffff"  
47 android:hint="Parolanızı Giriniz"  
48 android:inputType="textPassword"** />  
  
49 <**Button  
50 android:id="@+id/btnSign"  
51 android:layout\_width="match\_parent"  
52 android:layout\_height="wrap\_content"  
53 android:text="Giriş"  
54 android:background="#ffffff"  
55 android:textColor="#44B39D"  
56 android:textSize="18sp"  
57 android:textStyle="bold"  
58 android:fontFamily="monospace"  
59 android:layout\_marginTop="8dp"  
60 tools:layout\_editor\_absoluteX="147dp"  
61 tools:layout\_editor\_absoluteY="133dp"** />  
  
62 </**LinearLayout**>  
  
63 <**TextView  
64 android:layout\_weight="1"  
65 android:gravity="bottom"  
66 android:id="@+id/textView"  
67 android:textColor="#ffffff"  
68 android:textSize="20dp"  
69 android:layout\_marginBottom="12dp"  
70 android:fontFamily="cursive"  
71 android:textAlignment="center"  
72 android:textStyle="italic"  
73 android:layout\_width="match\_parent"  
74 android:layout\_height="wrap\_content"  
75 android:text="Cebinizdeki kişisel ajandanız."** />  
  
76 </**LinearLayout**>

1-10. satır kısımlarında LinearLayout için gerekli tanımlamalarımızı yaptık. Burada en önemli kısım bizim için 7. Satırda yer alan android:orientation="vertical" kısmı. Bu kısımda bileşenlerimizin nasıl yerleşeceğini belirtiyoruz. “vertical” dediğimiz için bileşenler dikeyde hizalanacaktır. Ekranımızın tamamına pembe rengini vermek için 8. Satırda background niteliğini kullandık.  
  
11-17. Satır aralarında ImageView tanımladık. Burada bizim için en önemli kısım id tanımlamasıdır. Tanımladığımız id sayesinde MainActivity.java kısmında bu bileşene erişebileceğiz. Bu yüzden id isimlerine kullandığımızın bileşenin amacına göre isim vermek büyük projelerde çalışırken karmaşıklığı azaltmanıza yardımcı olacaktır. 17. Satırda ImageView bileşeninin nereden dolacağını belirten yolu yazdık. Yani drawable klasörünün içindeki user isimli görseli kullandık.

18-22.satır aralarında bir LinearLayout daha tanımlayarak bir iç hizalama yaptık. LinearLayout kapandığı yer Button bileşeninden sonrası. Yani EditText bileşenleri ve Button bileşenimizi dikeyde hizalarken bütün olarak hizalayabilmek için bir LinearLayout tanımladık.Kapsayıcı olarak düşünebiliriz.  
  
23-36. Satırlarda EditText bileşenimizi tanımladık. 33. Satırda android:backgroundTint="#ffffff" özelliği ile EditText alt çizgi rengini beyaz yaptık. 35. satırda android:hint="Adınızı Giriniz" özelliği ile EditText’imizin amacını belirten bir ipucu yazısı yazdık. EditText’e tıkladığımızda bu yazı kaybolacaktır.  
  
Bileşen tanımlamalarımızda bilmemiz gerekenler bu şekilde. Şimdi Java tarafına geçip burada id verdiğimiz bileşenlerimizi Java tarafında da tanımlayalım.  
  
MainActivity.java kısmına ilgili kodlarımızı açıklayalım.  
 **1 package** com.hasibezafer.loginapp;  
  
**2 public class** MainActivity **extends** AppCompatActivity **implements**

**3** View.OnClickListener{  
4 EditText **etName**;  
5 EditText **etPassword**;  
6 Button **btnSign**;  
  
7 @Override  
8 **protected void** onCreate(Bundle savedInstanceState) {  
9 **super**.onCreate(savedInstanceState);  
10 setContentView(R.layout.***activity\_main***);  
  
11 **etName** = (EditText) findViewById(R.id.***etName***);  
12 **etPassword** = (EditText) findViewById(R.id.***etPassword***);  
13 **btnSign** = (Button) findViewById(R.id.***btnSign***);  
14 **btnSign**.setOnClickListener(**this**);  
  
15 }  
  
16 @Override  
17 **public void** onClick(View view) {  
  
18 **if**(view == **btnSign**){  
19 String userName = **etName**.getText().toString();  
20 Intent intent = **new** Intent(getApplicationContext(),

21 WelcomeActivity.**class** );  
22 intent.putExtra(**"USERNAME"**, userName);  
 23 startActivity(intent);  
 24 }  
  
 }  
}

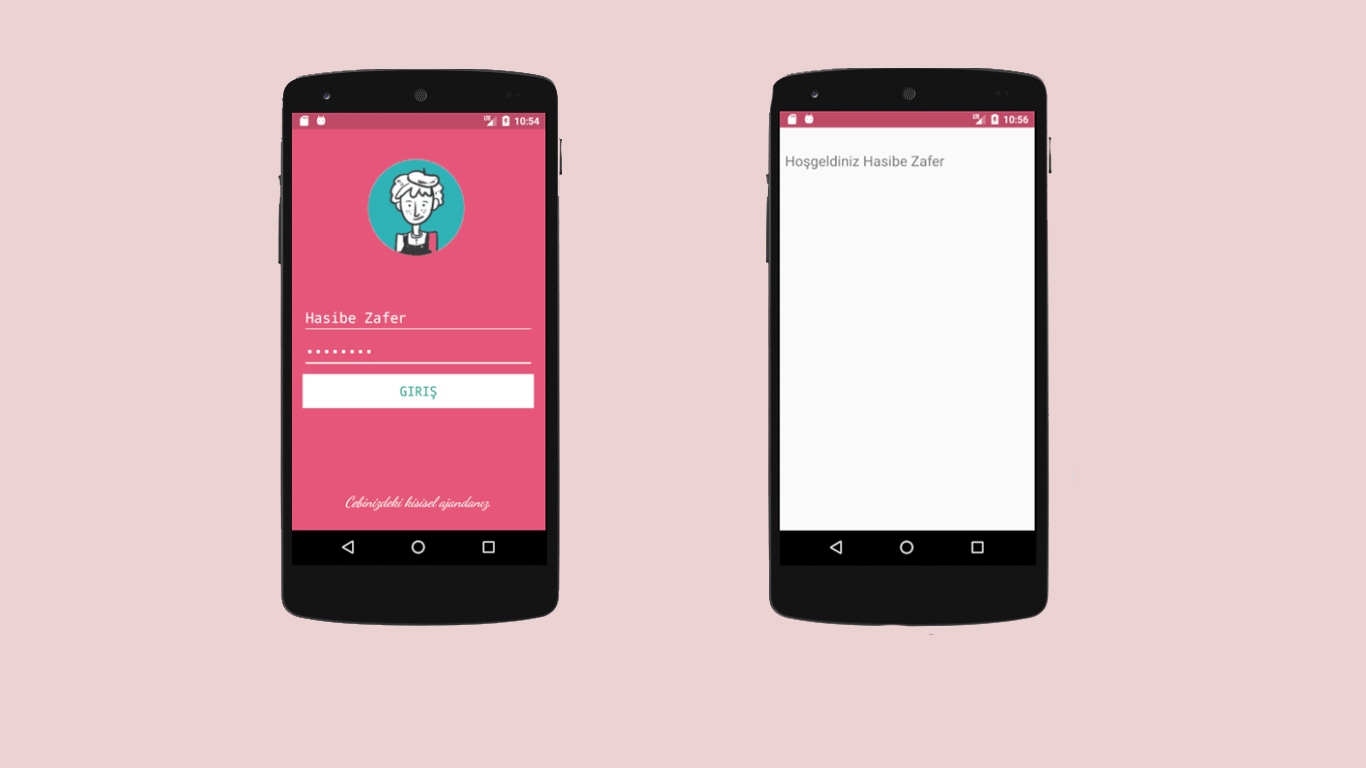
1. Satırda uygulamamızın paket adı yer almaktadır.
2. Satırda MainActivity sınıfımız AppCompatActivity sınıfından kalıtılmaktadır.

4.satırda etName adında bir EditText, 5. Satırda etPassword adında EditText ve 6. Satırda btnSign adında bir Button tanımladık.  
  
11-13. Satırlar arasında bu değişkenlere bileşenlerimizi atadık. Bu atamayı yaparken bileşenleri çağırmak için id’lerini kullandık. Yani activity\_main.xml’de ilk EditText’imizin id değeri “etName” idi. Biz bu bilgiyle o EditText’i, etName adında tanımladığımız EditText tipindeki değişkene atadık.  
  
14. Satırda butonumuza tıklanabilme(Click) özelliğini set ettik. Bunun için setOnClickListener() metodunu kullandık. Click olaylarını 1. Satırda **implements** View.OnClickListener diyerek OnClickListener interfacesini kullandık. Burada Alt+Enter diyerek bu interface’in sahip olduğu metodları set ettik. Alt+Enter dediğimizde OnClick metodumuz 17. Satırda bizi karşıladı. Şimdi butonumuza tıklandığında hangi işlemlerin gerçekleşeceğini burada tanımlayacağız. Activity’mizde birden çok buton olabilirdi. Bunun için btnSign tıkladığımızda diyebilmek için 18. Satırda gerekli if koşulumuzu yazdık.  
  
Butona tıkladığımızda gerçekleştireceğimiz işlem şu şekilde. Kullanıcının editText’e girdiği yazıyı alarak oluşturacağımız WelcomeActivity’de “Hoşgeldiniz xxxxxxx” şeklinde yazıyı ekranda görüntülemek.   
Bunun için ilk olarak 19. Satırda userName adında String bir değişken oluşturup , etName adındaki editTextimizin getText() metodu aracılığıyla içindeki ifadeyi alıp bu ifadeyi String’e çevirmek için toString() metodunu kullandık. Artık editTexte girilen ifade userName değişkenine atanmış durumda.  
  
20. Satırda bir Intent tanımladık. Intent diğer Activity geçişimizi sağlayacak.  
21. Satırda intent aracılığıyla putExtra() metodunu kullanarak diğer Activity’mize bilgi taşıyabiliyoruz. Burada etiket oluşturma mantığı ile “USERNAME” etiketi sayesinde WelcomeActivity’den bilgiyi alabileceğiz.

Evet bilgiyi gönderme işlemimizi tanımladık. Şimdiyse gönderdiğimiz bu bilgiyi yeni oluşturduğumuz WelcomeActivity.java içerisinde alacağız. İlk olarak activity\_welcome.xml içerisinde bir TextView ekliyoruz.  
  
  
Bunun için gerekli kodlarımız şu şekilde:  
  
*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout 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:orientation="vertical"  
 tools:context="com.hasibezafer.loginapp.WelcomeActivity"**>  
  
 <**TextView  
 android:id="@+id/tvUserName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:layout\_marginTop="25dp"  
 tools:layout\_editor\_absoluteX="48dp"  
 tools:layout\_editor\_absoluteY="16dp"** />  
</**LinearLayout**>

Şimdi WelcomeActivity.java kısmında ilgili kodlarımızı inceleyelim.

**1 public class** WelcomeActivity **extends** AppCompatActivity {  
  
2 TextView **tvUserName**;  
  
3 @Override  
4 **protected void** onCreate(Bundle savedInstanceState) {  
5 **super**.onCreate(savedInstanceState);  
6 setContentView(R.layout.***activity\_welcome***);  
  
7 **tvUserName** = (TextView) findViewById(R.id.***tvUserName***);  
8 String accountName = getIntent().getStringExtra(**"USERNAME"**);  
9 **tvUserName**.setText(**"Hoşgeldiniz "** + accountName);  
 }  
}

1. Satırda tvUserName adında bir EditText tanımladık. 7. Satırda activity\_welcome.xml’de tvUserName id’li TextView’imizi tvUserName değişkenimize atadık.
2. Satırda MainActivity’den gelen USERNAME etiketli bilgiyi almak için getIntent() metodunu kullandık. String tipinde bir ifade alacağımız için etiketimizi getStringExtra içerisine yazdık ve String tipinde oluşturduğumuz accountName içerisine atadık. 9. Satırda ise bu bilgiyi TextView’e set ettik. Böylelikle ilk ekranımızda EditText’e girilen ismi aldık ve kullanıcı giriş yaptığında ikinci ekranda Onu Hoşgeldiniz İsim şeklinde karşıladık. ****