

# Android “Uygulamayı Puanla” Butonu Ayarlama

Android uygulamaları içinde en çok kullanılan kod bloklarından biri de kullanıcının uygulamayı puanlamak için kullanıcıyı Google Play Store sayfasına yönlendiren kod bloktur. Aşağıdaki kodu, kopyalayıp yapıştırarak kullanıcılarınızı uygulamanızın Google Play sayfasına yönlendirebilirsiniz.

```
Uri uri = Uri.parse("market://details?id=" +
getPackageName());
        Intent goToMarket = new Intent(Intent.ACTION_VIEW,
uri);
        // To count with Play market backstack, After
pressing back button,
        // to taken back to our application, we need to
add following flags to intent.
goToMarket.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY |
        Intent.FLAG_ACTIVITY_NEW_DOCUMENT |
        Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
        try {
            startActivity(goToMarket);
        } catch (ActivityNotFoundException e) {
            startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("http://play.google.com/store/apps/details?id=" +
getPackageName())));
        }
```

---

# SQLite Insert Hızlandırma

Üzerinde çalıştığım Android uygulamasında yaşadığım bir sıkıntıyı ve çözümünü burada sizinle paylaşmak istedim.

[Zararlı Bağlantılar](#) isimli uygulamamızda siber dünyada tespit edilen zararlı linkleri ve kategorilerini paylaşıyoruz. Veriler sürekli güncellendiğinden internetten verileri alıp local veri tabanına kaydedilmesi gerekiyor. Aldığımız kayıt sayısı yirmi binleri aştığından standart SQLite insert komutu ve foreach döngüsü ile 10 dakikayı aşan bir süre alıyordu. İnternetten yaptığım araştırma sonucunda aşağıdaki kod ile insert işlemini yapınca yaklaşık 10 saniye kadar bir zaman aldı.

```
String sql = "INSERT INTO table (number, nick) VALUES (?, ?)";
db.beginTransaction();

SQLiteStatement stmt = db.compileStatement(sql);
for (int i = 0; i < values.size(); i++) {
    stmt.bindString(1, values.get(i).number);
    stmt.bindString(2, values.get(i).nick);
    stmt.execute();
    stmt.clearBindings();
}

db.setTransactionSuccessful();
db.endTransaction();
```

# Android Ekranlar (Activity) Arası Geçiř



İkinci Activity Geçiř

Yapılan yazılımların hemen hemen hepsi birden fazla ekran içermektedir. Android yazılımları da bu kuraldan müstesna değildir. Android yazılımında ekranlar activity olarak adlandırılır ve aşağıdaki kodda ikinci activity açmayı ve açılan yeni activiye veri gönderip almayı öğreneceğiz:

Activity başlatmak için faydalanacağımız sınıf Intent sınıfıdır. Intent nesnesinin kullanmak için Intent sınıfını projemize import etmemiz gerekmektedir.

```
import android.content.Intent;
```

Activity açmak için Intent nesnesini tanımlayıp `startActivity(intent)` metodu ile activity başlatıyoruz.

```
Intent intent = new Intent(this, MesajGosterActivity.class);
startActivity(intent);
```

Bu işlem sonucunda ikinci activity başlamış olur. İkinci ekran açılırken en sık yapılan işlemlerden biri de ikinci ekrana veri gönderilmesidir. Bu işlem için Intent nesnesinin `putExtra` ve `getXXXX` metotlarından faydalanıyoruz. Veri göndermek için:

```
Intent intent = new Intent(this, AcilacakEkran.class);
intent.putExtra("anahtar", veri);
startActivity(intent);
```

Kodunu kullanıyoruz. Açılan yeni activity nesnesine gelen veriyi almak için de `onCreate` metodu içerisinde:

```
Intent intent = getIntent();
string data = intent.getStringExtra("anahtar");
```


Kodu ile de activityye gelen string değeri alırız.

# Android Yazılımı İçerisinde Reklam Vermek – 3

## Reklam birimi oluşturun


1 Reklam biçimini seçin ————— 2 Reklam birimi ayarlarını yapılandır

Remember Words  
Android




**Banner**  
Uygulama düzeninin bir kısmını kaplayan dikdörtgen reklamlardır. Bu reklamlar, belirli bir süre sonra otomatik olarak yenilenebilir.

SEÇ



**Geçiş**  
Seviye atlama gibi doğal bekleme ve geçiş anlarında gösterilen tam sayfa reklam biçimi.

SEÇ



**Ödüllü**  
Kullanıcıları reklam izledikleri için ödüllendiren reklam biçimidir. Ücretsiz içerik kullanıcıları üzerinden para kazanmak için idealdir.

SEÇ

## Geçiş Reklamları

Merhaba arkadaşlar bu yazımızda Android yazılımımıza verebileceğimiz bir diğer reklam çeşidi olan geçiş reklamlarını göreceğiz. Geçiş reklamları özellikle sayfa veya bölüm geçişlerinde kullanabileceğiniz ve tam sayfa kaplayan reklamlardır.

Geçiş reklamı eklemek için ilk iş olarak [daha önceki yazımızda](#) anlattığımız AdMob hizmeti üzerinde hesap ve uygulama açmamız

gerekmektedir. İkinci adım olarak da “Geçiş” türünde bir reklam birimi oluşturmamız gerekmektedir. Reklam birimi oluşturmayı da [serimizin ikinci yazısında](#) işlemiştik. Reklam birimi oluşturma ve kütüphanelerin projeye eklenmesinde sonra ki işlemler aşağıdaki gibidir.

```
private InterstitialAd mInterstitialAd;
```

Şeklinde sınıf genelinde geçerli bir geçiş reklam nesnesi tanımladıktan sonra:

```
mInterstitialAd = new InterstitialAd(this);  
mInterstitialAd.setAdUnitId("REKLAM BİRİMİ KİMLİĞİNİZ");
```

kodu ile reklam birimi nesnemizi tanımlıyoruz. Banner reklam biriminde olduğu gibi burada da dikkat etmemiz gereken şey, testlerimizi AdMob geçiş reklamları için belirlediği test birim kimliği kullanmamız. Test ID değeri:

**ca-app-pub-3940256099942544/1033173712**

Tanımlamaları yaptıktan sonra da:

```
mInterstitialAd.loadAd(new AdRequest.Builder().build());
```

kod satırını ile nesnemize reklam talep ederek yükleme yapıyoruz. Yükleme işlemi yaptıktan sonra da:

```
mMyButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (mInterstitialAd.isLoaded()) {  
            mInterstitialAd.show();  
        } else {  
            Log.d("TAG", "Reklam henüz yüklenmedi.");  
        }  
    }  
});
```

Örnek kodu ile reklamı ekrana gösteriyoruz. Her gösterimden sonra bir sonraki gösterim için yeni reklam talebinde bulunmamız gerekmektedir. Bunun için reklam biriminin kapanış olayından sonra yeni reklam isteminde bulunuyoruz.

```
mInterstitialAd.setAdListener(new AdListener() {  
    @Override  
    public void onAdClosed() {  
        // Reklam kapandığında yeni reklam istemi.  
        mInterstitialAd.loadAd(new  
AdRequest.Builder().build());  
    }  
});
```

```
});
```

---

## Android Yazılımı İçerisinde Reklam Vermek – 2

Tekrardan merhaba arkadaşlar, [dünkü yazımızda](#) AdMob servisi üzerinde uygulama oluşturmayı işlemiş ve bu yazımızda da uygulamamıza ilk reklamı ekleyeceğimiz yazmıştık.

Reklamları uygulamamızda gösterebilmek için öncelikle AdMob üzerinde reklam birimleri oluşturmamız gerekmektedir. Reklamları bu birimler üzerinden göstereceğiz. Aşağıda bir banner reklam birimi oluşturmayı ekran görüntüleri ile adım adım gördükten sonra uygulamaya eklemeyi göreceğiz.



Google AdMob

Reklam birimleri

Reklam göstermek için reklam birimleri oluşturun

Banner'lerden, geçiş reklamları, ödüllü reklamlar ve yerel reklamlara kadar seçebileceğiniz birçok reklam biçimi var. Bir reklam birimi oluşturun, biçim atayın, reklam biriminin kimliğini uygulamanıza ekleyin ve işlem tamam! Para kazanmaya hazırsınız. Ayrıntılı bilgi için SDK'yi takip edin.

BAŞLAT

## Reklam birimi oluşturma

“BAŞLAT” butonuna tıklayarak oluşturabileceğimiz reklam birimleri görebileceğimiz ekrana geçiş yapıyoruz.

Reklam birimi oluşturun

1 Reklam biçimini seçin — 2 Reklam birimi ayarlarını yapılandır

Remember Words  
Android

**Banner**  
Uygulama düzeninin bir kısmını kaplayan dikdörtgen reklamlardır. Bu reklamlar, belirli bir süre sonra otomatik olarak yenilenebilir.

SEÇ

**Geçiş**  
Seviye atlama gibi doğal bekleme ve geçiş anlarında gösterilen tam sayfa reklam biçimi.

SEÇ

**Ödüllü**  
Kullanıcıları reklam izledikleri için ödüllendiren reklam biçimidir. Ücretsiz içerik kullanıcıları üzerinden para kazanmak için idealdir.


SEÇ


## Oluşturabilecek Reklam Birimleri

Biz konumuz olan “Banner” birimine “SEÇ” deyip birim ayarlarını gireceğimiz ekrana giriş yapıyoruz.

## × Reklam birimi oluřturun

✓ Reklam biçimini seçin ————— 2 Reklam birimi ayarlarını yapılandır

 Remember Words  
Android

Reklam biçimi ⓘ  Banner

Reklam birimi adı ⓘ   
8 / 80

▼ Geliřmiş ayarlar

**REKLAM BİRİMİ OLUŐTUR** GERİ

### Banner Reklam Birimi Ayarları

Reklam birimine vereceğimiz adı girdikten sonra “REKLAM BİRİMİ OLUŐTUR” butonuna tıklayarak reklam birimine ait kimlik değerlerini alacağımız ekrana geçiř yaparak reklam birimi oluřturmayı tamamlıyoruz.

✓ Reklam birimi bařarıyla oluřturuldu

Yeni reklam birimlerinin reklam gstermeye bařlamasının bir saat kadar srebileceđini unutmayın. Beklerken reklam birimlerini [test etmeye](#) ne dersiniz?

Ardından, reklam birimini uygulamanıza yerleřtirin

řu talimatları uygulayın:

1. řu uygulama kimliđini kullanarak [Google Mobil Reklamlar SDK'sı kılavuzundaki](#) talimatları uygulayın:

 ca-app-pub-

2. SDK'yı entegre etmek iin [banner uygulama kılavuzuna](#) bařvurun. řu reklam birimi kimliđini kullanarak kodu entegre ettiđinizde reklam tr, boyutu ve yerleřim belirtirsiniz:

 ca-app-pub-

3. Gerekleřtirdiđiniz uygulamanın politikalara uygunluđunu denetlemek iin [AdMob politikalarını](#) inceleyin.

[TALİMATLARI E-POSTAYLA GNDER](#)

TAMAMLANDI

[BAřKA BİR REKLAM BİRİMİ OLUřTUR](#)

## Reklam Birimine Ait Deđerler

## Android Uygulamamıza Reklamın Eklenmesi

Bundan sonraki adımlarımızda da uygulamamıza reklamı eklemeyi iřleyeceđiz:

Proje seviyesinde build.gradle dosyamızı aıp maven ktphanesini ařađıdaki gibi ekliyoruz. Ekleme yapacađımız yer **maven** kısmıdır.

```
allprojects {
    repositories {
        jcenter()
        maven {
            url "https://maven.google.com"
        }
    }
}
```

```
}  
}
```

Maven kütüphanemizi ekledikten sonra app seviyesinde build.gradle dosyamızda da dependencies altına aşağıdaki gibi kütüphaneyi ekliyoruz.

```
implementation 'com.google.android.gms:play-services-  
ads:15.0.1'
```

Kütüphanemizi de ekledikten sonra manifest dosyamıza reklam için meta datamızı ekliyoruz. Ekleyeceğimiz alan meta-data alanıdır. [ADMOB\_APP\_ID] yerine AdMob'tan aldığımız uygulama kimliğimizi yazıyoruz.

```
<manifest>  
  <application>  
    <!-- Sample AdMob App ID: ca-app-  
pub-3940256099942544~3347511713 -->  
    <meta-data  
android:name="com.google.android.gms.ads.APPLICATION_ID"  
      android:value="[ADMOB_APP_ID]"/>  
  </application>  
</manifest>
```

Bütün işlemlerden sonra uygulamamızın açılış ekranında bir defaya

mahsus reklam servisini oluřturuyoruz. Bu iřlemi onCreate metoduna yazabiliriz.

```
// Sample AdMob app ID: ca-app-pub-3940256099942544~3347511713
MobileAds.initialize(this, "YOUR_ADMOB_APP_ID");
```

řimdi sıra geldi layout dosyamıza reklamı eklemeye;

```
<com.google.android.gms.ads.AdView
xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_alignParentBottom="true"
    ads:adSize="BANNER"
    ads:adUnitId="ca-app-
pub-3940256099942544/6300978111">
    </com.google.android.gms.ads.AdView>
```

Kodu ile banner reklamımızı uygulamamıza eklemiř oluyoruz. ads:adUnitId alanına AdMob'tan aldıđımız reklam birimi kimliđini yazıyoruz. Burada dikkat etmemiz gereken nokta testlerimizi yaparken AdMob'un banner iin belirlemiř olduđu test reklam birimi kimliđini kullanmamız gerektiđidir. AdMob test kimliđi:

**ca-app-pub-3940256099942544/6300978111**

Ve geldik reklamı yüklemeye:

```
AdView mAdView = findViewById(R.id.adView);  
    AdRequest adRequest = new AdRequest.Builder().build();  
    mAdView.loadAd(adRequest);
```

---

## Android Yazılımı İçerisinde Reklam Vermek – 1

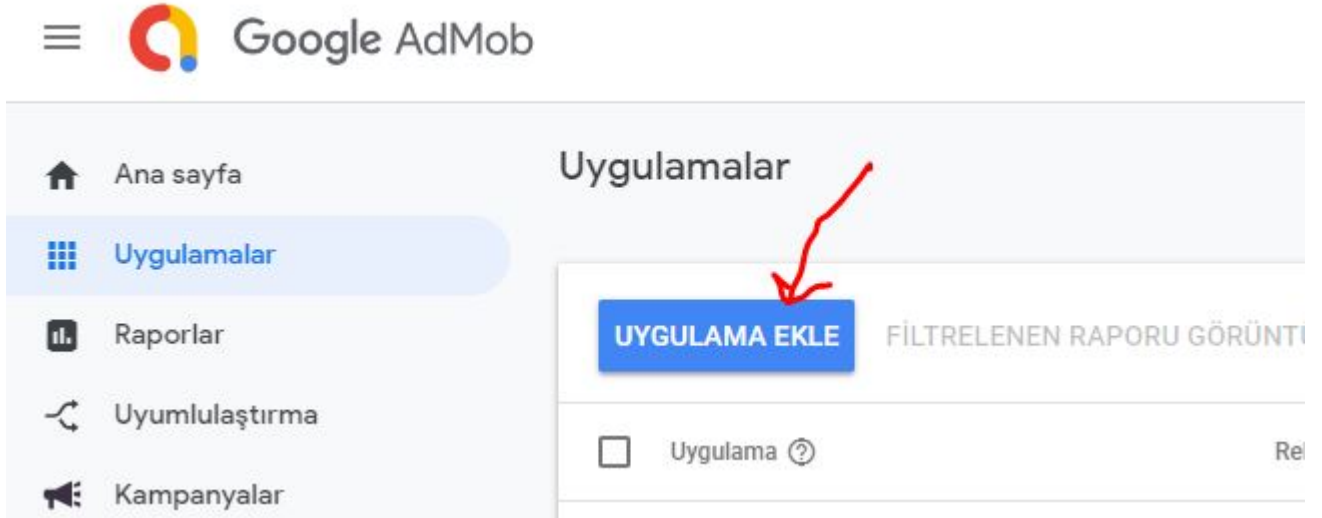
Her ne kadar klasikleşmiş bir söz olarak “öncelikle para kazanacağınız değil mutlu olacağınız işi yapın” deseler de ortada bir gerçek var ki iş de para kazanmak için yapılıyor. Bu yazı serimizde geliştirdiğimiz android yazılımlarımıza reklam vermeyi göreceğiz.

Android yazılımını yazıp bitirdikten sonra iki şekilde para kazanılabilir. İlk seçenek yazılımı ücretli olarak Google Play üzerinde satmak ancak ne yazık ki bu seçenek ülkemizde çok işlemiyor zira yazılıma para vermek çöpe atılan para gibi görülmektedir ya da lisanslı yazılım kullanımı “kreği vardı ya ne diye para veriyorsun” sözü ile genellikle ayıplanmaktadır. İkinci seçenek de yazılım içerisinde reklam göstermek; bu yazı serimizde

reklam verme işlemlerini baştan sona işlemeye çalışacağız.

Reklam verme işlemi için öncelikle [AdMob](#) servisine üye olmak gerekiyor. AdMob Google tarafından satın alınmış olan bir reklam hizmetidir.

Üye olduktan sonra ilk işlem olarak uygulama oluşturmamız gerekmektedir.



AdMob uygulama oluşturma

İkinci adım olarak uygulamanın yayınlanıp yayınlanmadığını soruyor:



Uygulamanızı Google Play veya App Store'da yayınladınız mı?

[EVET](#) [HAYIR](#)

Uygulama yayın durumu

Biz burada "HAYIR" seçeneğini tıklayarak işlemimize devam ediyoruz:

Uygulamamızın yayında olmadığını seçtikten sonra karşımıza uygulamamızın platformu ve adını soran ekran çıkacaktır. Gerekli bilgileri girdikten sonra "EKLE" diyerek devam ediyorum

Uygulama bilgilerini girin

Remember Words

14 / 80

Platform



Android



iOS

[EKLE](#)


[GERİ DÖN](#)

Uygulama adı ve platform seçimi



“EKLE” butonuna tıkladıktan sonra reklam birimi oluřturmamızı isteyen ekran gelecektir karřımıza, “DAHA SONRA YAPACAĐIM” diyerek iřlemi sonlandırıyoruz.

**Tebrikler! Uygulamanız AdMob'a eklendi**



**Remember Words**  
Android  
Uygulama Kimliđi [REDACTED]

**Sonraki adımlar**

1. Yeni uygulama kimliđinizi not edin. Bu kimliđi, AdMob'da reklam yayınlamak iin uygulamanızın kaynak koduna eklemeniz gerekecektir.
2. Uygulamanızda reklam gstermek iin reklam birimi oluřturun.
3. Uygulamanız Google Play veya App Store'da yayınlanıyorsa, uygulamanıza bađlantı oluřturmak iin geri gelmeyi unutmayın.

**SONRAKİ ADIM: REKLAM BİRİMİ OLUŐTURUN** DAHA SONRA YAPACAĐIM

Uygulama oluřturma sonu

Yarın ki yazımızda, reklamların uygulamaya eklenmesini sađlayan reklam birimlerinden banner reklam oluřturma ve eklemeyi iřleyeceđiz.

## İř Saatleri Takip Uygulaması

Merhabalar,

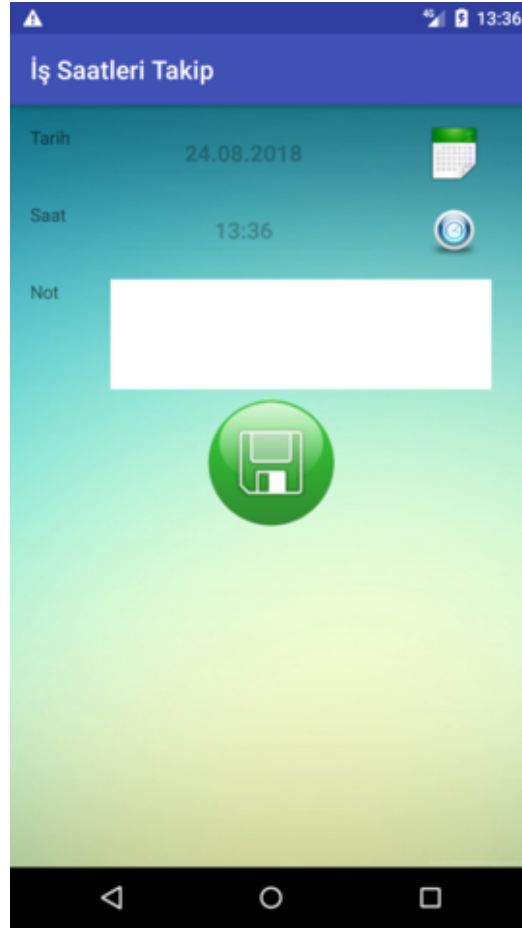
Sitemizin genelinde de dikkat çektiđi üzere öğrenciliđimden bu yana sürekli .Net uygulamaları ile haşır neşir oldum ancak bu sene farklı bir alanda Android alanında çalışmalara başlamaya karar verdim. Ve Android sisteminde ilk uygulamalarımın birini daha yayına almış bulunmaktayım. Bu yazılım ile günlük işe başlama ve çıkış saatlerinizi takip altında tutabilirsiniz.

Aşağıdaki ekranlarda uygulama hakkında temel bilgiler yer almaktadır.



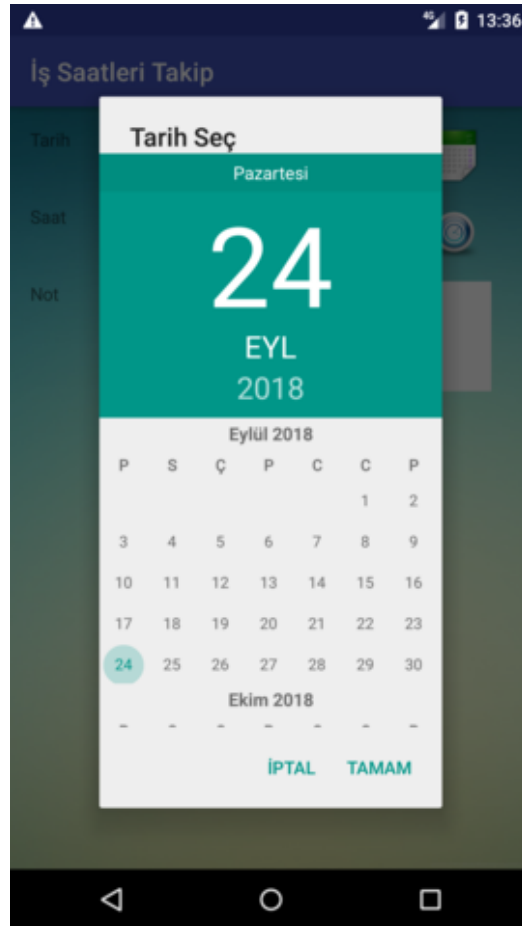
1.

Uygulama ana ekranı



2.

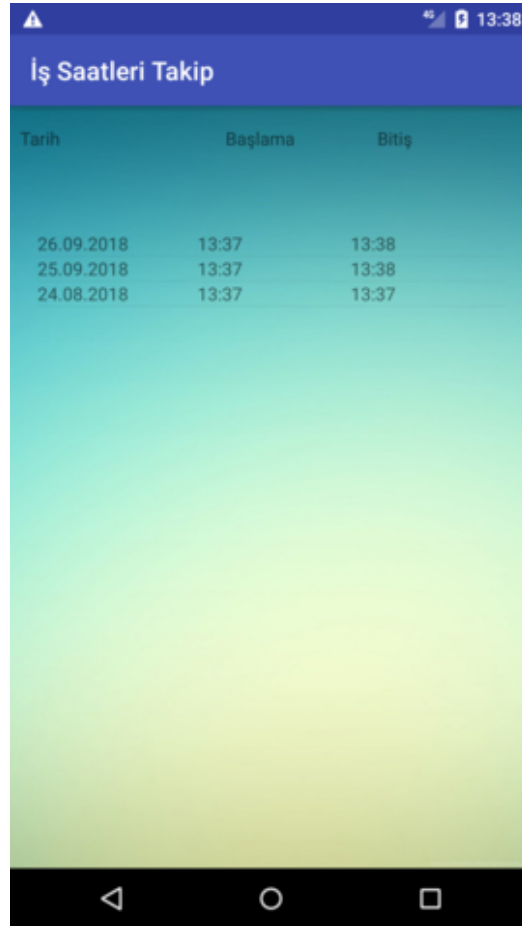
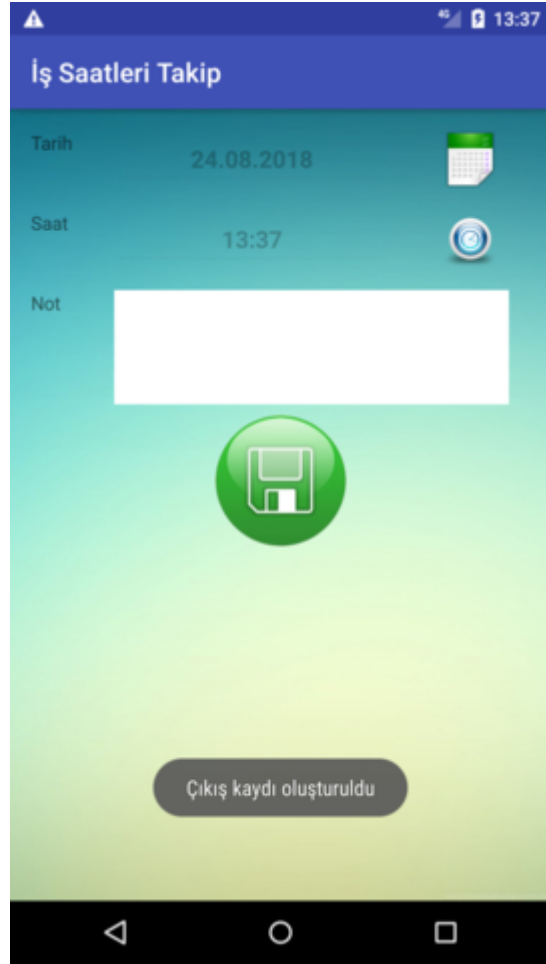
Giriş çıkış kayıt ekranı



Kayıt için tarih seçme işlemi

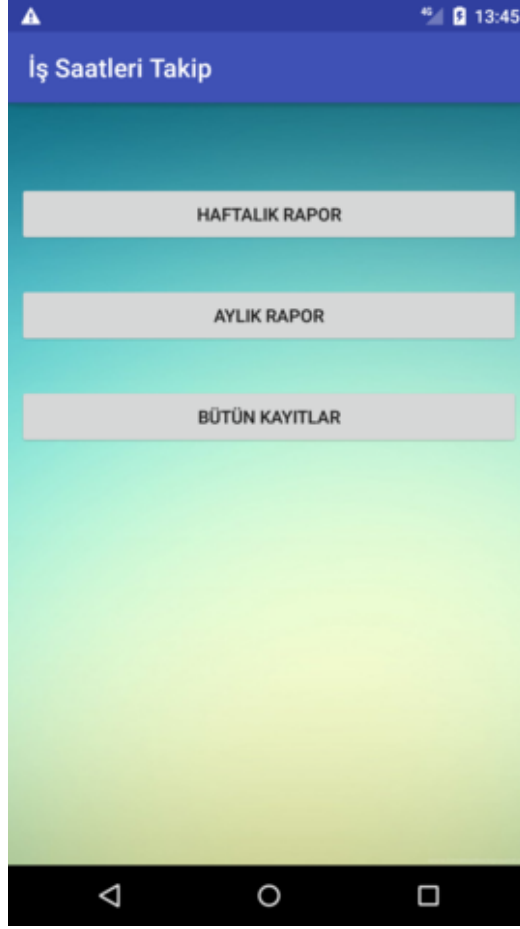


Kayıt için saat seçme işlemi



3.

## Rapor Ekranı



Sunulan Raporlar

Yazılımı bu linkten indirebilirsiniz:

<https://play.google.com/store/apps/details?id=com.saitorhan.workinghours>