

T.C. İSTANBUL Okan Üniversitesi

Meslek Yüksek Okulu

Elektronik ve Otomasyon Bölümü



— İSTANBUL —
OKAN ÜNİVERSİTESİ

BİTİRME PROJESİ

ITEM CARD

GÜLER FATMA AÇIKGÖZ

DANIŞMAN

NİLGÜN İNCEREİS

PROGRAM

MOBİL TEKNOLOJİLERİ

MAYIS 2020

BİTİRME PROJESİ

ITEM CARD

GÜLER FATMA AÇIKGÖZ

tarafından

OKAN ÜNİVERSİTESİ

Mobil Teknolojileri Bölümüne

Önlisans

derecesi şartını sağlamak için sunulmuştur.

Onaylayan:

Danışman:
Nilgün İncereis

ÜYE

ÜYE

ÜYE

MOBİL TEKNOLOJİLERİ

MAYIS 2020

ÖNSÖZ

“ITEM CARD” isimli Android cihazlar için hazırlanmış bu çalışma, Okan Üniversitesi, Mobil Teknolojileri Bölümü’nde “Bitirme Projesi” olarak hazırlanmıştır.

Bu projeyi hazırlamam esnasında her türlü konuda bana yardımcı olan, beni proje yapmaya heveslendiren ve Mobil Teknolojileri alanında bilgi edinmemi sağlayan sayın danışman eğitim görevlisi Nilgün İncereis’e, önlisans eğitimim boyunca her türlü bilgilendirmeyi sağlayan sayın Yrd. Doç. Dr. Ali Kılınç ve sayın Arhan Demirali’ ye ve eğitim hayatım boyunca desteklerini esirgemeyen herkese teşekkür ediyorum.

GÜLER FATMA AÇIKGÖZ

İçindekiler

1. GENEL BİLGİLER	6
1.1. ANDROID STUDIO	6
1.1.1. PROJE OLUŞTURMA	6
1.2. JAVA PROGRAMLAMA DİLİ	3
1.3. FIREBASE VERİTABANI SİSTEMİ	3
1.4. SQLite VERİTABANI	4
2. ITEM CARD PROJESİ	4
2.1. PROJE GENEL BİLGİLERİ	4
2.1.1. PROJENİN AMACI NEDİR?	4
2.1.2. PROJENİN KONUSU NEDİR?	5
2.1.3. KULLANICI KAYITLARI	5
2.1.3.1. KULLANICI NASIL KAYIT OLUR?	5
2.1.3.2. KULLANICI NASIL GİRİŞ YAPAR?	2
2.1.3.3. KULLANICI ÜRÜNÜ SEPETE NASIL EKLER?	1
2.1.3.4. KULLANICI ÜRÜNLERİ NASIL SATIN ALIR?	1
2.1.4. KULLANILAN ARAYÜZ ELEMANLARI	2
2.1.4.1. ConstraintLayout	2
2.1.4.2. LinearLayout	2
2.1.4.3. ScrollView	3
2.1.4.4. ImageView	4
2.1.4.5. TextView	4
2.1.4.6. PlainText	5
2.1.4.7. Button	6
2.1.5. VERİTABANI İŞLEMLERİ	7
2.1.5.1. SQLite KULLANIMI	7
2.1.5.1.1. SQLite FONKSİYONLARININ KULLANILMASI	10
2.1.6. İNTERNET SERVİSİ	11
3. GELECEKTE YAPILACAKLAR	12
4. SONUÇ	13
5. KAYNAKÇA	14

ÖZET

MOBİL TEKNOLOJİLERİ ITEM CARD

Proje Danışmanı: Nilgün İncereis

Bu proje oyun oynayan insanların hayatlarını kolaylaştırmak adına yapılmıştır. Oyun oynayan insanlar oynadıkları oyunlara eklentiler satın almak isteyebilirler. Bu eklentileri her oyunun kendi içerisinde satın almak yerine bu mobil uygulama içerisinde toplu bir şekilde satın alma işlemlerini gerçekleştirebileceklerdir. Ayrıca olası indirim kampanyalarını da tek bir oyun içerisinde değil indirim geçerli olduğu bütün oyunlar ve eklentiler için de kullanabileceklerdir.

Bu uygulama içerisinde kullanıcılar kendi ellerinde bulunan eklentileri, oyunları ve kullanıcı hesaplarını satabileceklerdir. Uygulamanın mobil uygulama olması da bu yüzden avantajlıdır. Oyuncular genellikle masaüstü bilgisayarlar veya oyuncu bilgisayarları kullanmakta olduğu için gittikleri her yere bu cihazları taşıyamazlar. Ama bu uygulamayı kullanıyorlarsa akıllı cihaz olan her yerden hesaplarına erişerek yapacakları işleri kolaylıkla halledebilirler.

1. GENEL BİLGİLER

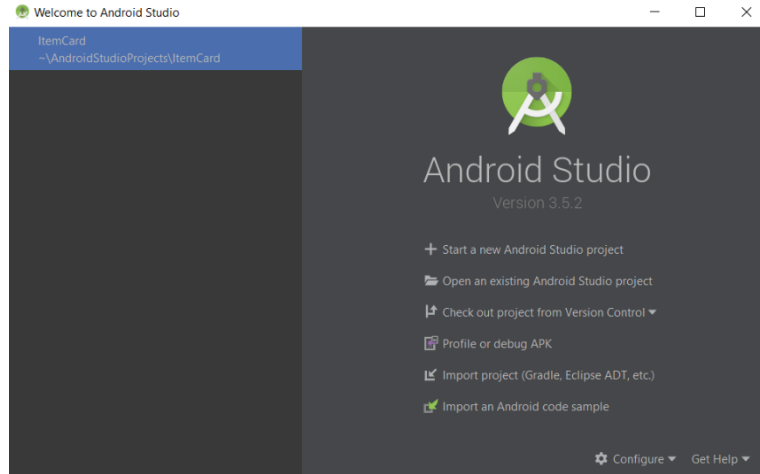
1.1. ANDROID STUDIO



Resim 1: ANDROID STUDIO LOGO

Google tarafından sunulan bir programdır. Java dilinde kodları yazarak uygulama yazılabilen ve uygulama geliştirilebilen bir platformdur. Bu program MacOS, MacBook ve Windows fark etmeksizin kurulup çalıştırılabilmektedir. Bu platforma resmi sayfaları üzerinden ulaşabilir ve kurulum dosyaları indirilerek kurulum yapılabilir.

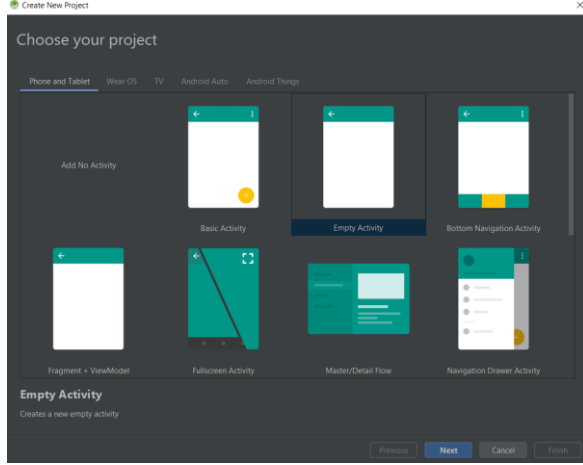
1.1.1. PROJE OLUŞTURMA



Resim 2: PROJE OLUŞTURMA ADIM 1

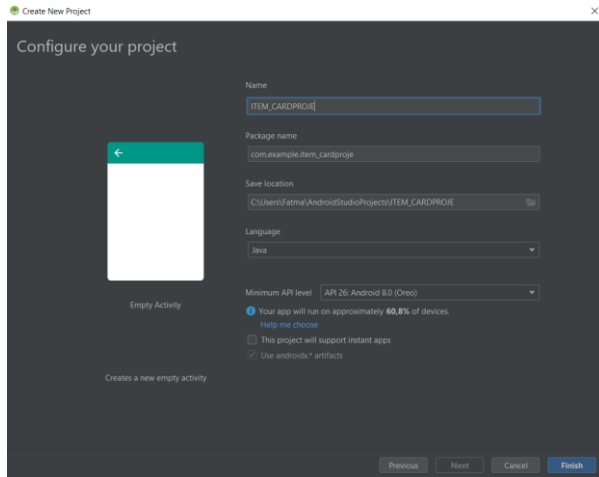
Android Studio açıldıktan sonra Resim 2’ de görülen şekilde uygulama açılmaktadır. Bu bölüm de “Start a new Android Studio Project” ile yeni proje oluşturabilirsiniz. “Open an existing Android Studio project” ile daha önceden Android Studio ile hazırlanmış olan proje açılabilir. Bu

kısımda bunların haricinde Gradle, SDK ve NDK gibi eklentilerle ilgili deęişikler ve ayarlar da yapılabilir.



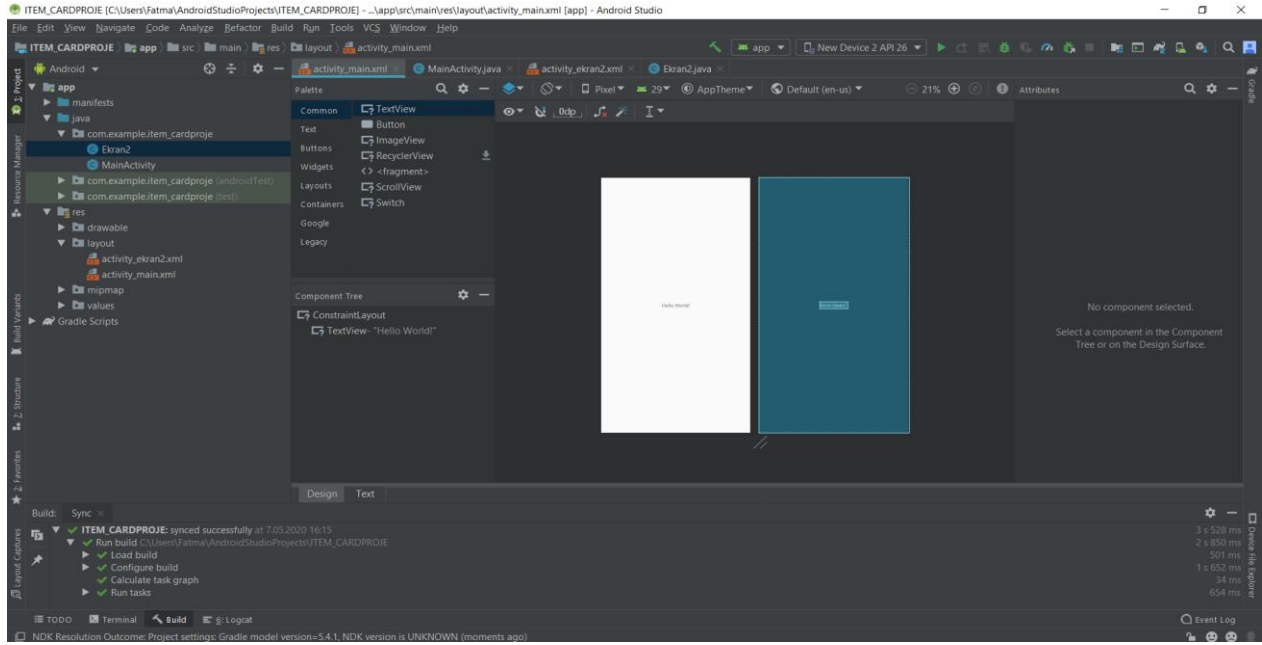
Resim 3: PROJE OLUŐTURMA ADIM 2

Çıkan ilk pencerede “Start a new Android Studio Project” seçeneęi seçilirse Resim 3’ de görünen pencere açılacaktır. Bu pencere de hangi cihaz için yazılım yapılacağı seçilebilmektedir. Telefon, tablet, akılla saat, android yazılımlı araçlar ve dięer android cihazlar için yazılım yapılabilir. Cihaz seçimi yapıldıktan sonra bu pencere içerisinde projede oluşacak ilk ekranın nasıl olması isteniyor ise o seçenek seçilmelidir. Örneęin ekran tasarımı baştan yapılacaksa “Empty Activity” seçeneęi seçilebilir.



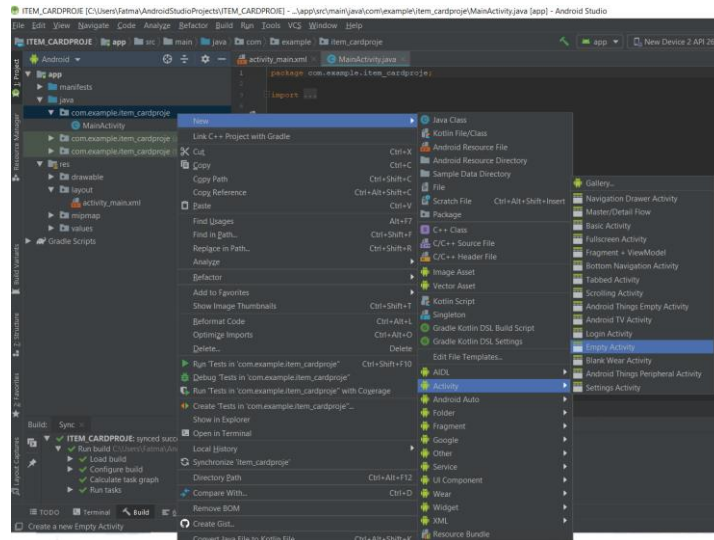
Resim 4: PROJE OLUŐTURMA ADIM 3

Resim 3’ de görünen penceredeki adımları tamamlayıp “NEXT” butonuna tıkladıktan sonra Resim 4’ de görünen pencere açılacaktır. Bu pencere içerisinde proje ismi, projenin paket ismi, proje dosyasının nereye kaydedileceęi, proje içerisinde kodları yazmak için kullanılacak dil ve android versiyonu gibi seçimler yapılmaktadır.



Resim 5: PROJE OLUŞTURMA ADIM 4

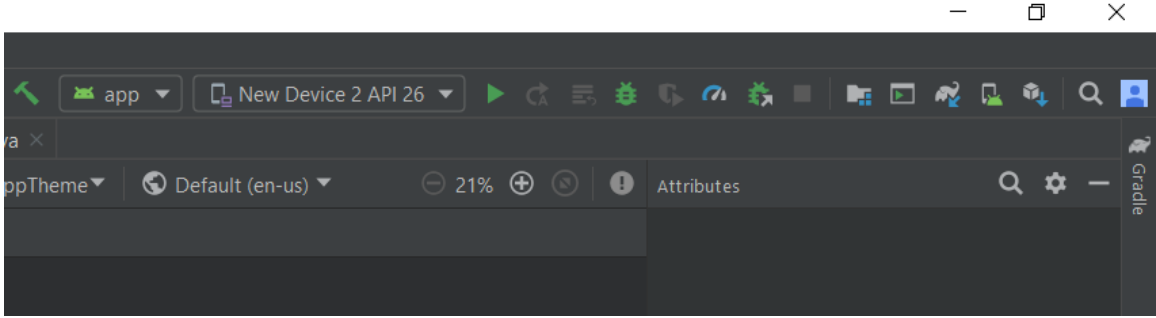
Resim 4’ de görülen pencere de işlemleri gerçekleştirdikten sonra “FINISH” butonuna tıklayarak proje oluşturulabilir. Bu işlem sonrasında Resim 5’ de görünen şekilde proje açılacaktır. Bu aşamadan sonra ekran tasarımları yapılabilir ve gerekli olan kodlar yazılarak proje tamamlanabilir.



Resim 6: PROJE OLUŞTURMA ADIM 5

Uygulama içerisine yeni ekran eklenmek istiyorsa Resim 6 içerisinde gösterilen aşamalar takip edilerek yeni ekran eklenebilir. Project bölümü içerisinde yapılan uygulamanın ekranları

java sayfaları gradle ve manifest dosyalarına ulaşılabilir. Bu kısım platform içerisinde uygulamanın dosya görünümüdür.



Resim 7: PROJE OLUŞTURMA ADIM 6

Proje oluşturulduktan sonra test etmek isteniyorsa test edilecek sanal cihaz veya gerçek cihaz seçildikten sonra yeşil üçgen şeklinde görünen “Run” butonuna basılarak veya Shift+F10 kısa yolu kullanılarak uygulamayı cihaza yükleyebilir ve uygulama test edilebilir.

1.2. JAVA PROGRAMLAMA DİLİ

Android uygulamaları Java dili kullanılarak geliştirilir. Java’nın temel özellikleri öğrenmesinin ve anlamasının kolay olması, program bağımsız ve güvenli olması, sanal makinelerle uyumlu olmasıdır. Android de bu temellere büyük ölçüde uyum sağlar. Java’ yı öğrenen programcılar, Android geliştirme işlemleri de yapabilirler. Kaynak kodu açısından bakıldığında, yazılan kod aynıdır.

1.3. FIREBASE VERİTABANI SİSTEMİ



Resim 8: Firebase logo

Günümüzde uygulamalar platform fark etmeksizin aynı veriye her cihazdan erişmek istiyor. Uygulaması, birçok kullanıcı tarafından yüklenen geliştiricilerin uygulama içerisinde veri analizi, kayıt tutma, verileri ve dosyaları depolama gibi işlemleri rahatlıkla yönetebileceği bir yönetim paneli gerekmektedir. Google tarafından çalışmalar yapılarak sürekli geliştiren Firebase,

uygulama geliştiricilerinin ihtiyaçlarını karşılayabilmek için uygulama geliştiricilerine ilk aşamada ücretsiz kullanım da sunabilen bir bulut veri tabanı platformudur.

1.4. SQLite VERİTABANI

SQLite bir veritabanı kütüphanesidir. Bu kütüphane dışarıda bağlı olacağı bir veritabanı platformu aramaz. Uygulama içerisinde yazılan kodlar ve sql sorgularıyla yine uygulama içerisinde bir veritabanı oluşturur

2. ITEM CARD PROJESİ

2.1. PROJE GENEL BİLGİLERİ

Proje oluşturulurken Android Studio platformu ve Java programlama dili kullanılmıştır. Proje içerisinde iki adet veritabanı sistemi kullanılmıştır. Bunlar; Firebase veritabanı ve SQLite veritabanı sistemidir. Firebase içerisinde şu an için sadece kullanıcı kayıtları tutulmaktadır. SQLite ile kullanıcıların sipariş kayıtları tutulmaktadır.

Kullanıcı kayıt işlemini gerçekleştirdiği esnada uygulama internet bağlantısı aracılığıyla Firebase veritabanı sistemine ulaşır ve kayıt işlemini gerçekleştirir.

Kullanıcı ürünleri sepete eklemek için tıkladığı zaman seçtiği ürün sepete eklenir. Bu işlemi gerçekleştirmek için SQLite kütüphanesi kullanılır. Bu kütüphane sayesinde uygulama içerisinde her kullanıcının kendisine ait olan veritabanı ve tabloları oluşturulur. Daha sonra bu tablolara ürün eklemesi yapılarak satın alınan ürünlerin kaydedilmesi sağlanır.

2.1.1. PROJENİN AMACI NEDİR?

Proje içerisinde bilgisayar oyunu ve oyun ürünlerinin satışı yapılacaktır. Bilgisayar oyunları içerisinde bulunan ürünlere ‘Item’ denmektedir. Ürün satın alma sayfalarında ürünlerin bilgileri bir tanıtım kartı gibi olduğu için ‘Card’ kısmı eklenmiştir. Sonuç olarak Ürün Kartı anlamına gelen “Item Card” ismi doğmuştur.

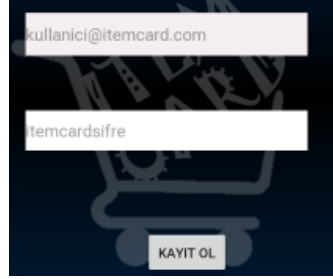
Bu çalışmanın amacı, mobil ve bilgisayar oyuncularını aynı platform üzerinde birleştirmeyi sağlayacak bir uygulama (Android/IOS) yapmaktır.

2.1.2. PROJENİN KONUSU NEDİR?

Oyun sektörü iki gruba ayrılmış bulunmaktadır: Mobil oyuncuları ve bilgisayar oyuncuları. Oyun içerisinde bulunabilecek her türlü ihtiyaca yönelik satış bu uygulama içerisinde bulunmaktadır. Oyuncular bu platformda aradıkları oyunları, kullanacakları oyun eşyalarını, oyun paralarını ve oyun hesaplarını bulabilecek ve satın alabileceklerdir.

2.1.3. KULLANICI KAYITLARI

2.1.3.1. KULLANICI NASIL KAYIT OLUR?



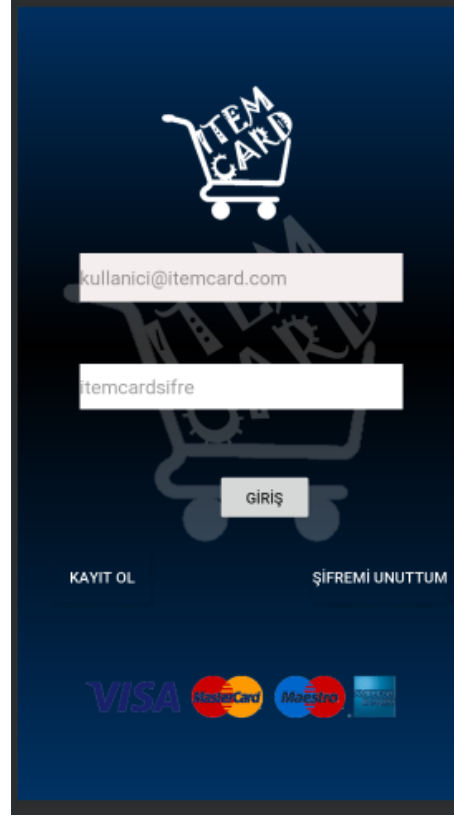
Resim 9: KULLANICI KAYDI

Kullanıcı uygulamayı cihazına indirdikten sonra eğer daha önce kullanıcı kaydı oluşturduysa bu bilgilerle ilk ekrandan girişini gerçekleştirebilir. Eğer kullanıcı kaydı daha önce oluşturmadıysa giriş ekranından “Kayıt Ol” butonuna basarak Resim 9’ da görülen ekrana geçiş yapacaktır.

Bu ekran içerisinde iki adet veri girişi kısmı bulunmaktadır. Kullanıcı bunlardan birine var olan bir mail adresi diğerine ise uygulamaya giriş için kullanacağı şifreyi girecektir.

Kayıt esnasında bir problem olmadıysa ekranda kullanıcıya “Kayıt Olundu!” uyarısı verilmektedir. Eğer kayıt esnasında bir sorun oluştuysa hata mesajı çıkmaktadır. Bu mesajlar boş alan olmasıyla ya da sistem arızalarıyla ilgilidir. Kullanıcı eğer alanlardan birini boş bırakırsa kullanıcı kayıt işlemi gerçekleştirilemeyecektir. Bu sebeple kullanıcıya hata mesajı verilerek boş alanları doldurması gerektiği bilgisi verilmektedir. Bu uyarı haricinde bazı sistem hataları olmaktadır. Bunlar da aynı şekilde oluşan problem ile birlikte kullanıcıya hata mesajı şeklinde iletilmektedir.

2.1.3.2. KULLANICI NASIL GİRİŞ YAPAR?

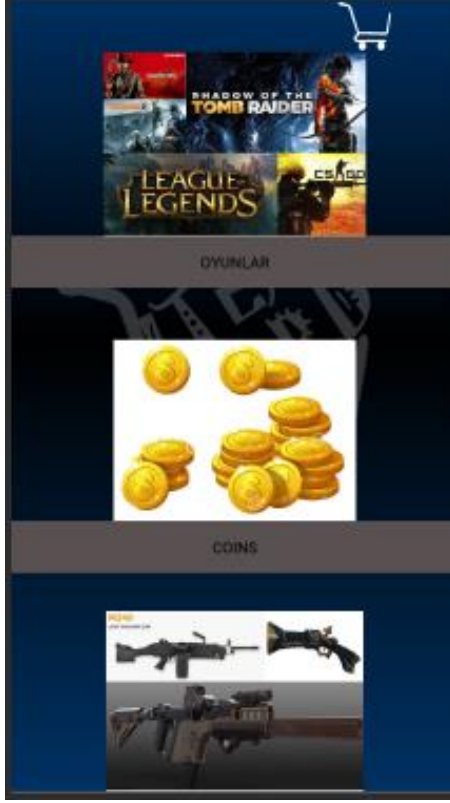


Resim 10: KULLANICI GİRİŞİ

Uygulama ilk açıldığında kullanıcı girişi ekranı açılır. Eğer Kullanıcı bir Item Card hesabına sahip değilse “Kayıt Ol” butonuna basarak kullanıcı kayıt ekranına geçerek kayıt işlemini gerçekleştirebilir. Kullanıcı kayıt aşamasını tamamladıktan sonra giriş ekranına yönlendirilir. Bu ekran içerisinde kayıt esnasında kullanılan mail adresi ve şifre kullanılarak giriş işlemi gerçekleştirilir.

Giriş esnasında bir problem olmadıysa ekranda kullanıcıya “Giriş Başarılı!” uyarısı verilmektedir. Eğer giriş esnasında bir sorun oluştuysa hata mesajı çıkmaktadır. Bu mesajlar boş alan olmasıyla ya da sistem arızalarıyla ilgilidir. Kullanıcı eğer alanlardan birini boş bırakırsa kullanıcı girişi işlemi gerçekleştirilemeyecektir. Bu sebeple kullanıcıya hata mesajı verilerek boş alanları doldurması gerektiği bilgisi verilmektedir. Bu uyarı haricinde bazı sistem hataları olmaktadır. Bunlar da aynı şekilde oluşan problem ile birlikte kullanıcıya hata mesajı şeklinde iletilmektedir.

2.1.3.3. KULLANICI ÜRÜNÜ SEPETE NASIL EKLER?



Resim 11: ANASAYFA



Resim 12: OYUNLAR EKRANI

Kullanıcı uygulamaya giriş yaptıktan sonra karşısına Resim 11’de görülen anasayfa çıkmaktadır. Bu kısımda ürün gruplarına yönlendirme yapılan görseller ve butonlar bulunmaktadır. Kullanıcı bu kısımda hangi ürünü almak istiyorsa o ürünün grubunu seçerek ürüne ulaşabilir ve sepetine ekleyerek bu ürünü satın alabilir.

Kullanıcı ilgili ürün grubunu seçip aradığı ürünü bulduktan sonra satın al butonuna tıklayarak ürünü sepete ekleyebilir. Örneğin; Kullanıcı Fifa 2020 oyununu satın almak istiyorsa anasayfa üzerinden oyunlar grubunu seçip daha sonrasında Fifa 2020 (Resim 12) ürününü bulup fiyatı üzerine tıklayarak sepete ekleyebilir.

2.1.3.4. KULLANICI ÜRÜNLERİ NASIL SATIN ALIR?

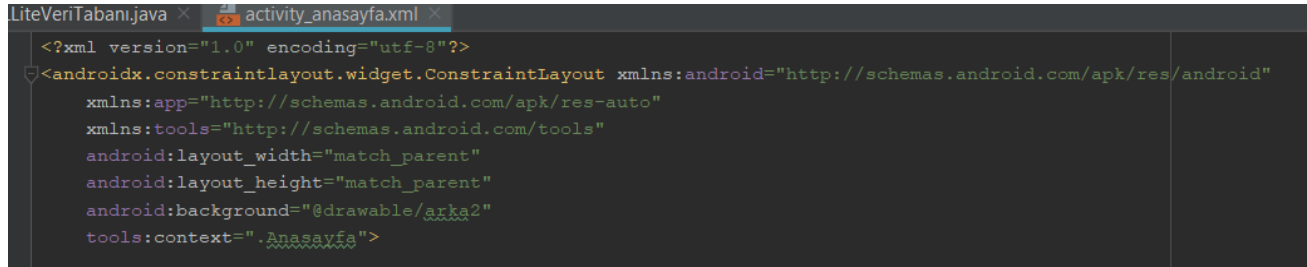
Kullanıcı satın almak istediği ürünleri sepetine ekleyip bu işlemi tamamladıktan sonra Resim 11 ve Resim 12’ de de görülen sağ üst kısımda bulunan sepet resmine tıklayarak sepet

ekranına geiş yapabilir. Bu ekran ierisinde kullanıcının sepete eklediėi rnler bulunmaktadır. Almak istediėi rnleri kontrol ettikten sonra kullanıcı deme ekranına geiş yaparak rnleri satın alıp rn kodlarına ulaşabilir.

2.1.4. KULLANILAN ARAYZ ELEMANLARI

2.1.4.1. ConstraintLayout

API 9 ve tnde bulunan tm API' ler ile kullanılabilen bir yapıdır. Karmaşıık tasarımı ekranlarda layout' lar birbirine baėımlı olmayacak şekilde kullanılabilme imkanı sunar. Bu layout ierisinde bulunan viewlerden birinin yeri deėiştirildiėinde baėlı olan view de yer deėiştirmektedir. Bu yzden daha kolay ve daha performanslı alıřmaktadır.



```
<?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"
    android:background="@drawable/arka2"
    tools:context=".Anasayfa">
```

Resim 13: RNEK CONSTRAINTLAYOUT KODU

Resim 13' de ConstraintLayout' un android projesi ierisinde bulunması gereken odları verilmiřtir. Bu kod ierisinde layout' un konumu, tasarımı ve hangi sayfada olduėu gibi zellikler belirtilmektedir. Eėer bu kodu projelerin xml sayfalarına eklenirse; ConstraintLayout ekrana eklenmiř olur.

2.1.4.2. LinearLayout

Android ierisinde nesneleri konumlandırmak iin kullanılan bir arayz elemanıdır. Nesneleri yatay olarak veya dikey olarak konumlandırmayı saėlar. Bu eleman sayesinde nesneler dzgn bir şekilde hizalanabilir ve konumlandırılabilir.

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="horizontal">

```

Resim 14: ÖRNEK LINEARLAYOUT KODU

Resim 14 içerisinde örnek bir LinearLayout kodu verilmiştir. Bu nesne iki şekilde kullanılabilir; yatay ve dikey olarak. Yazılan kodlar içerisinde bulunan android:orientation içerisine horizontal yazılırsa yatay, vertical yazılırsa dikey olarak bir nesne oluşturulur. Görünen diğer kodlarda ise konumu ve boyutu gibi özellikler bulunmaktadır.

2.1.4.3. ScrollView

Bu nesne, uygulama ekranına koyulması gereken fakat ekran içerisine sığmayan nesneleri ekrana koymayı sağlar. Yani ekranı aşağı yukarı hareket ettirebilmemizi sağlar. Bir sınırı yoktur. Arayüz elemanlarını bu nesne içerisine ekleyerek kullanıcıya göstermek istenen nesneler aşağıya doğru eklenmiş olur. Uygulamayı kullanan kişi ekranı aşağı-yukarı hareket ettirerek eklenen nesneleri görmesi sağlanır.

```

<ScrollView
    android:id="@+id/scrollView2"
    android:layout_width="100%"
    android:layout_height="100%"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="horizontal">

            <Button
                android:id="@+id/button10"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:background="@color/white" />

            <Button
                android:id="@+id/button11"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:background="@color/white" />

```

Resim 15: ÖRNEK SCROLLVIEW KODU

Resim 15 içerisinde görüldüğü gibi nesneler ScrollView içerisine eklenebilir. Nesne eklemek için <ScrollView> etiketinin açılması gerekmektedir. Eklenecek nesneler bittikten sonra </ScrollView> etiketi ile kapatılır. Bu iki etiket arasına eklenen her nesne ekran hareket ettirildiğinde görüntülenebilir.

2.1.4.4. ImageView

Oluşturulan ekran içerisinde gösterilecek görselleri eklemek için kullanılan etikettir. Bu etiket içerisinde hangi görselin uzantısı verilirse o görseli gösterir. Uygulamaya bir görsel ekleyebilmek için uygulamanın kök dosyası içerisinde bulunan Drawable klasörü içerisine görselin eklenmesi gerekmektedir. Daha sonra bu görseli istenilen yerlerde imageview etiketi ile gösterilebilir.

```
<ImageView
    android:id="@+id/imageView6"
    android:layout_width="match_parent"
    android:layout_height="167dp"
    app:srcCompat="@drawable/coin2" />
```

Resim 16: ÖRNEK IMAGEVIEW KODU

Resim 16 içerisinde gösterilen kodlar uygulama ekranına eklendiğinde resim dosyası çağırılmış olur. Android:id kodlarından resim nesnesine isim verilebilir. App:srcCompat ile uygulama kök dosyası içerisinde bulunan Drawable dosyası içerisinde istenilen görsel eklenebilir.

2.1.4.5. TextView

Uygulama ekranı içerisinde gösterilecek metinler için kullanılır.

```
<TextView
    android:id="@+id/forzacvazi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="FORZA COINS"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textSize="24sp" />
```

Resim 16: ÖRNEK TEXTVIEW KODU

2.1.4.6. PlainText

Uygula ekranında kullanıcıdan bir veri alınacaksa; yani kullanıcının giriş yapması gereken bir durum varsa bu etiket kullanılır. E-mail, password ve phone gibi seçenekleri bulunmaktadır. Bu seçenekler sayesinde herhangi bir ekstra kod yazmaya gerek kalmadan kullanıcın girişi kısıtlanarak arayüz kontrolü sağlanabilir.

```
<EditText
    android:id="@+id/email_giris"
    android:layout_width="0dp"
    android:layout_height="45dp"
    android:layout_marginStart="57dp"
    android:layout_marginEnd="57dp"
    android:layout_marginBottom="6dp"
    android:background="#FDF7F0F0"
    android:ems="10"
    android:hint="kullanici@itemcard.com"
    android:inputType="textEmailAddress"
    android:maxLines="1"
    android:textColor="#2C2A2A"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

Resim 17: ÖRNEK PLAINTEXT KODU

Arayüz içerisinde hızlı nesne ekleme menüsünde bu nesne birçok farklı isimde bulunmaktadır. PlainText, E_mail, Phone vs. gibi birçok farklı şekilde görebiliriz. Fakat bu nesne kod kısmında EditText olarak geçmektedir. Bu nesneyi ilgili ekranın java sınıfında çağırırken de EditText olarak çağırılmaktadır. Android:inputType ile nesnenin türü belirtilir. Örneğin android:inputType="password" gibi. Bu şekilde arayüz elemanının hangi türde kullanılacağı belirtilebilir.

Bunların dışında bu nesnenin boyutu, tasarımı ve üzerinde otomatik olarak yazılacak yazı gibi özellikleri de bu kısım da düzenlenebilir.

2.1.4.7. Button

Bu nesne ile ekran içerisinde gerçekleşecek olaylar eklenebilir. Sayfalar arası geçiş ve sayfa içi olayların gerçekleşebilmesi için button gibi nesneler kullanılmaktadır.

```
<Button
    android:id="@+id/giris"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="25dp"
    android:text="giris"
    app:layout_constraintBottom_toTopOf="@+id/yardimsayfa"
    app:layout_constraintEnd_toStartOf="@+id/yardimsayfa" />
```

Resim 18: ÖRNEK BUTTON KODU

Resim 18 içerisinde görünen kodlar xml sayfasına eklenirse bir adet button ekrana eklenmiş olur.

```
giris.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String email=email_giris.getText().toString();
        String sifre=sifre_giris.getText().toString();

        if (TextUtils.isEmpty(email) || TextUtils.isEmpty(sifre))
        {
            loginProgress.setTitle("Hata!");
            loginProgress.setMessage("Boş alanlar var!");
            loginProgress.setCanceledOnTouchOutside(false);
            loginProgress.show();
        }

        else if (!TextUtils.isEmpty(email) || !TextUtils.isEmpty(sifre))
        {
            loginProgress.setTitle("Oturum Açılıyor!");
            loginProgress.setMessage("Lütfen Bekleyiniz!");
            loginProgress.setCanceledOnTouchOutside(false);
            loginProgress.show();
            kullanıcıGirisi(email,sifre);
        }
    }
});
```

Resim 19:ÖRNEK BUTTON BAĞLANTISI

Resim 19 içerisinde arayüz elemanı olan button ilgili ekranın java classına eklenmiştir. Burada buttona tıklanma olayı eklenmiştir. Bu buttona tıklanması durumunda olacak olay işlenmiştir.

2.1.5. VERİTABANI İŞLEMLERİ

2.1.5.1. SQLite KULLANIMI

Proje içerisinde kullanılan veritabanı sistemlerinden bir tanesidir. Bu veritabanı sistemi bir kütüphanedir. Uygulama içerisinde dışarıda bir bağlantı olmadan bir veritabanı oluşturmayı ve kendi içerisinde bir kayıt tutmayı sağlar.

```
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;
```

Yukarı da bulunan kodlar ile SQLite kütüphanesi projeye dahil edilir.

```
private static final String DATABASE_NAME = "ItemCard";  
private static final String TABLO_Urunler = "Urunler";  
private static final String ROW_ID = "id";  
private static final String ROW_UrunAd = "urun";  
private static final String ROW_Fiyat = "fiyat";
```

Bu kodlar ile veritabanı için daha sonra kullanılmak üzere fonksiyonlar oluşturulur. Bu kısımda veritabanının ismi, kullanacak tablonun ismi ve bu tablonun alanları ne olacağı yazılır. Yazım kurallarına dikkat edilmesi gerekmektedir. Boşluk ve karakter gibi kullanımların bu kısımda hatalı olması veritabanının oluşmasında sorunlar yaratarak programın çalışmasını engelleyebilir. Daha sonra bu fonksiyonlar sql komutları içerisinde çağırılarak veritabanı işlemleri gerçekleştirilecektir.

@Override

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL("CREATE TABLE " + TABLO_Urunler + "("  
        + ROW_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "  
        + ROW_UrunAd + " TEXT NOT NULL, "  
        + ROW_Fiyat + " TEXT NOT NULL)");}
```

Bu kodlarla yukarıda oluşturulan fonksiyonlar sql komutlarıyla birleştirilir. Veritabanının oluşturulması, tablonun oluşturulması ve sütunların veri tiplerinin belirtilmesi gibi işlemler bu kodlar içerisinde gerçekleştirilir.

SQLite sınıfı oluşturulurken otomatik olarak iki fonksiyon oluşur. Bunlar; onCreate ve onUpgrade fonksiyonlarıdır. onCreate fonksiyonu içerisinde gerçekleştirilecek olaylar oluşturulur. onUpgrade içerisinde ise kullanıcı işlemlerini tamamlayıp programı kullanmayı bıraktığı zaman olacaklar bulunmaktadır.

@Override

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
    db.execSQL("DROP TABLE IF EXISTS " + TABLO_Urunler);  
    onCreate(db);  
  
}
```

Bu kodlar ile kullanıcı uygulamaya son verdiği zaman olacaklar yazmaktadır. Kodlar da bu durumda oluşturulan tablonun silinmesi ve yeni bir boş tablo oluşturulması yazmaktadır. Silme işlemini “DROP TABLE IF EXIST” ile yeni tablo oluşturma işlemini ise onCreate(db) kodları ile fonksiyonu çağırarak yapmaktadır.

```
public void VeriEkle(String urun, String fiyat) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    try {  
        ContentValues cv = new ContentValues();  
        cv.put(ROW_UrunAd, urun);  
        cv.put(ROW_Fiyat, fiyat);  
        db.insert(TABLO_Urunler, null, cv);  
    } catch (Exception e) {  
    }  
    db.close();  
}
```

VeriEkle fonksiyonu içerisinde tabloya veri eklenmektedir. Kullanıcı bir ürünü satın alma seçeneğini seçerse bu ürünü tabloya eklemek için bu kodlar çalışır. Öncelikle daha sonra arayüzden alınacak veriler için iki adet değişken alınır. Daha sonra bu değişkenler tablo içerisinde bulunan ilgili alanlara eklenir. Daha sonra işlemler bittikten sonra veritabanı db.close(); kodu ile sonlandırılır.

```
public List<String> VeriListele() {  
    List<String> veriler = new ArrayList<String>();  
    SQLiteDatabase db = this.getReadableDatabase();
```

```

try {
    String[] stunlar = {ROW_ID, ROW_UrunAd, ROW_Fiyat,};
    Cursor cursor = db.query(TABLO_Urunler, stunlar, null, null, null, null, null);
    while (cursor.moveToNext()) {
        veriler.add(cursor.getInt(0)
            + " - "
            + cursor.getString(1)
            + " - "
            + cursor.getString(2));
    }
} catch (Exception e) {

}
db.close();
return veriler;
}

```

VeriListele fonksiyonu içerisinde tabloya eklenmiş olan verileri kullanıcıya gösterebilmek için gerekli kodlar bulunmaktadır. Bu kodlarla veriler id'lerine göre listelenir. Daha sonrasında işlemler tamamlandığında db.close(); kodu ile veritabanı bağlantısı sonlandırılır.

```

public void VeriSil(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    try {
        String where = ROW_ID + " = " + id;
        db.delete(TABLO_Urunler, where, null);
    } catch (Exception e) {
    }
    db.close();
}

```

VeriSil fonksiyonu içerisinde tablodan satır silme işlemi yapılmaktadır. Kullanıcı bir ürünü yanlışlıkla eklediye bu kod sayesinde ürünü seçerek silme işlemi gerçekleştirebilir. Bu işlemi yapabilmek için delete sql komtu kullanılmıştır. Seçilen ürün id' sine göre tablodan silinmektedir.

2.1.5.1.1. SQLite FONKSİYONLARININ KULLANILMASI

Oluşturulan SQLite veritabanı sistemi ekranların java classları içerisinde çağırılarak kullanılmaktadır. Veri ekleme, veri silme ve veri listeleme işlemleri bu sınıflar içerisinde gerçekleştirilir.

```
csbuy.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String urunad=csyazi.getText().toString();
        String urunfiyat=csfiyat.getText().toString();

        SQLiteVeriTabanı vt=new SQLiteVeriTabanı( context: Oyunlar.this);
        vt.VeriEkle(urunad,urunfiyat);
        Toast.makeText(getApplicationContext(), text: "Eklendi!",Toast.LENGTH_SHORT).show();
    }
});
```

RESİM 20: SQLite ÖRNEK KOD

Resim 20 içerisinde görünen kod ile oluşturulmuş olan SQLiteVeriTabanı sınıfından fonksiyon çağırma komutları bulunmaktadır. Burada csbuy isimli butona tıklanma olayı eklenmiştir. Bu butona tıklanması durumunda arayüzde bulunan ve java classına csyazi ve csfiyat olarak dahil edilen textleri urunad ve urunfiyat isimli iki değişkene göndererek veritabanına eklenmesi sağlanmıştır. Oluşturulan bu değişkenler veritabanı sınıfı dahil edildikten sonra bu veritabanında bulunan VeriEkle fonksiyonuna gönderilerek Urunler tablosuna eklenmiştir. Daha sonra kullanıcının eklenip eklenmediğini anlayabilmesi için “Eklendi” mesajı yönlendirilmiştir.

```
public void Listele(){
    SQLiteVeriTabanı vt = new SQLiteVeriTabanı( context: Sepet.this);
    List<String> list = vt.Verilistele();
    ArrayAdapter<String> adapter = new ArrayAdapter<>( context: Sepet.this, android.R.layout.simple_list_item_1,android.R.id.text1,list);
    Verilistele.setAdapter(adapter);
}
```

Resim 21: SQLite ÖRNEK KOD

Resim 21 içerisinde veritabanında bulunan urunler tablosunun arayüzde görünebilmesi için gerekli olan kodlar bulunmaktadır. Listele fonksiyonu içerisinde önce oluşturulan veritabanı sınıfı çağırılmıştır. Daha sonra arayüzde bulun ListView içerisinde bu verileri gösterebilmek için

ArrayAdapter kullanılmıştır. Ana fonksiyon içerisinde Listele(); kodu yazıldıktan sonra bu kod devreye girecektir.

2.1.6. İNTERNET SERVİSİ

Yapılan uygulama bulut veritabanı sistemine bağlı olduğu için internet bağlantısı olması gerekmektedir. Kullanıcı bu bağlantının olup olmadığını fark edemeyebilir. Böyle bir durumda uygulamaya giriş yapmaya çalışırsa bunu başaramaz. Olası bir sorunla karşılaşmaması için uygulama içerisinde internet servisi kullanılmıştır.

Android projeleri içerisinde birçok servis kullanılabilir. Konum, internet ve uçuş modu bunlardan bazılarıdır. Bu uygulama içerisinde internet servisine ihtiyaç duyulduğu için internet servisi kullanılmıştır. Bu servis wifi ağı üzerinden yada mobil ağ üzerinden internet bağlantısı olup olmadığını kontrol etmektedir.

Bu servis sayesinde eğer internet bağlantısı hatası yüzünden kullanıcının uygulama kullanımında internet bağlantısı kaynaklı bir sorun olup olmadığını kontrol edilmektedir.

```
private boolean isNetworkAvailable(Context context) {
    ConnectivityManager connectivity = (ConnectivityManager)
        context.getSystemService(Context.CONNECTIVITY_SERVICE); //Sistem ağını dinliyor internet var mı yok mu

    if (connectivity != null) {
        NetworkInfo[] info = connectivity.getAllNetworkInfo();
        if (info != null) {
            for (int i = 0; i < info.length; i++) {
                if (info[i].getState() == NetworkInfo.State.CONNECTED) {

                    if (!isConnected) { //internet varsa
                        isConnected = true;
                        Log.v(LOG_TAG, "İnternete Bağlantısı var!");
                        Toast.makeText(context, "İnternet Bağlantısı var!", Toast.LENGTH_LONG).show();
                    }
                    return true;
                }
            }
        }
        isConnected = false;
        Toast.makeText(context, "İnternet Yok", Toast.LENGTH_LONG).show();
        Log.v(LOG_TAG, "İnternet Yok!");
        return false;
    }
}
```

Resim 22: ÖRNEK İNTERNET SERVİSİ KODU

İnternet servisini kullanabilmek için öncelikle bir sınıf yaratılmalıdır. Bu sınıf içerisine Resim 22 içerisinde bulunan kodlar eklenmelidir. Bu kodlar sayesinde uygulama internet ağını

dinler ve bir cevap bekler dönen cevaba göre kullanıcıya bağlantı var ya da yok şeklinde uyarı mesajı görüntüler.

```
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);  
receiver = new NetworkChangeReceiver();  
registerReceiver(receiver, filter);
```

Resim 23: ÖRNEK İNTERNET SERVİSİ KODU

Servis sınıfı oluşturulduktan sonra bu servisi hangi ekran içerisinde kullanmak istiyorsak onun ilgili java sınıfı içerisine Resim 23 içerisinde bulunan kodlar eklenmelidir. Bu sayede bu ekran her açıldığında internet bağlantısı kontrol edilir ve kullanıcıya bilgi verir.

3. GELECEKTE YAPILACAKLAR

- Kullanıcıların uygulama üzerinde kendilerine ait bir profil sayfaları olacaktır. Bu sayfa içerisinde kullanıcı bilgileri profil fotoğrafları ve dilerlerse yaptıkları alış-verişler yer alacaktır.
- Kullanıcıların kendi aralarında satış yapabilecekleri bir bölüm oluşturulacaktır. Bu bölüm içerisinde kullanıcılar sahip oldukları fakat kullanmadıkları ürünlerin satışını gerçekleştirebilecekler.
- Kullanıcıları birbirlerini arkadaş olarak ekleyebileceklerdir.
- Kullanıcılar satış işlemleri gerçekleştirebilmek için bir mesajlaşma bölümü eklenecektir. Kullanıcılar buradan satış yapacakları ürünler hakkında daha detaylı bilgi verebilir veya arkadaşlarıyla ürünler hakkında sohbet edebilirler.
- Yapım aşamasında olan ve tasarımları hazırlanan ekranlar tamamlanarak kullanıcıya sunulacaktır.
- SQLite üzerinden yapılan sipariş kayıtları Firebase sistemi üzerine geçecektir. Burada bulunan hazır tablolar içerisinde kayıtlı ürün kodları bulunacaktır. Kullanıcılar sipariş verdiklerinde bu kodlardan random bir ürün anahtarı seçilerek kullanıcıya sunulacaktır.
- Daha fazla ürün grubu eklenecektir. Ürün yelpazesi genişlerse uygulama kullanıcıların bir o kadar ilgisini çeker. Bu sayede kullanıcı sayısı artış gösterir.

- Kullanıcılar isterlerse bu uygulama üzerinden oyun hesaplarının satışlarını da gerçekleştirebileceklerdir.

4. SONUÇ

Kullanıcılar hem bilgisayar hem mobil oyunları için ortak bir platform üzerinden iletişime geçebilecek veya var olan ürünler ile ihtiyaçlarını gidereceklerdir. Bu sayede kullanıcılar her bir oyun için farklı farklı platformlarla uğraşmak yerine tek bir platform üzerinde bütün işlemlerini gerçekleştirebileceklerdir.

Dünyada eşi bulunmayan Item Card kullanıcıların rahatları ve kazançları için mükemmel şekilde tasarlanmış bir uygulamadır. Bu uygulama sayesinde kullanıcılar hem fiyattan hem zamandan kazanarak her alanda kar edecekler.

Rapor içerisinde uygulama içerisinde kullanılan konu başlıkları ile ilgili bilgiler verilmiştir. Görsellerde bulunan kodlar tamamen Item Card projesi içerisinde alınarak rapora eklenmiştir.

Bu proje ve rapor bu konuda çalışmak isteyen kişilere ihtiyacı olan bilgileri verecektir. Gerekli olan kod blokları proje içerisinde detaylıca fakat anlaması kolay şekilde işlenmiştir.

Android projesi hazırlamak isteyenlere bu projenin bir destek olacağına inanmaktayım.

5. KAYNAKÇA

- https://www.youtube.com/watch?v=ha3kvTXWSQY&list=PLBXWnq3alGf8SkHpkSsCCadlk_TOphnEH&index=1
- <https://gelecegiyazanlar.turkcell.com.tr/blog/firebase-nedir-avantajlari-nelerdir>
- <https://mucahitkambur.wordpress.com/2015/09/06/androidte-splash-screen-yapimi-4/>
- <http://www.kodumundunyasi.net/android-2/android-aktivitelerin-yasam-dongusu-activities-lifecycle.html>
- <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-401/firebase-proje-olusturma>