

T.C.
HARRAN ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÖDEVİ

PETSHOP MOBİL UYGULAMASI

Efnan AKKUŞ
Doç. Dr. İ. Berkan AYDİLEK

ŞANLIURFA

2020

T.C.
HARRAN ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÖDEVİ

PETSHOP MOBİL UYGULAMASI

Efnan AKKUŞ
Doç. Dr. İ. Berkan AYDİLEK

ŞANLIURFA

2020

Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
Bitirme Ödevi Yönergesi uyarınca hazırlanmış ve anılan bölüme
sunulmuştur.

Şanlıurfa 2020

Efnan AKKUŞ

ONAY

Doç. Dr. İ. Berkan AYDİLEK

Danışman

Dr. Öğr. Üyesi Serdar ÇİFTÇİ

Bölüm Başkanı

Dr. Öğr. Üyesi Kemal GÜNER

Komisyon Başkanı

Arş. Gör. Mehmet Umut SALUR

Üye

Arş. Gör. Harun ÇİĞ

Üye

İÇİNDEKİLER

	SayfaNo
TEŞEKKÜR.....	i
ŞEKİLLER DİZİNİ.....	ii
TABLOLAR DİZİNİ.....	iii
SİMGELER DİZİNİ.....	iv
1.GİRİŞ.....	1
1.1.Android Gelişimi ve Desteklenen Özellikler.....	5
1.2.Android Kullanım Alanları.....	6
1.3.Google Play.....	7
1.4.Android Mimarisi ve Sistem Özellikleri.....	8
1.5.Android Studio Ortamı.....	9
1.6.Layout.....	10
1.7.Ux.....	11
2.BENZER ÇALIŞMALAR/LİTERATÜR TARAMASI.....	12
3.ÖNERİLEN YÖNTEM.....	13
4.KULLANILAN TEKNOLOJİLER.....	18
4.1.Java.....	18
4.2.Firebase.....	21
4.3.Xml.....	28
4.4.Json.....	32
5.BULGULAR.....	33
6.SONUÇLAR ve ÖNERİLER.....	37
KAYNAKLAR.....	38
ÖZGEÇMİŞ.....	39

TEŞEKKÜR

Öncelikle bitirme projesi konusu seçerken isteklerimi göz önünde bulundurup, projenin yürütülmesi sırasında bana yardımcı olan proje danışmanım Doç. Dr. İ. Berkan AYDİLEK'e çalışmalarım boyunca projenin tasarımı ve fonksiyonelliği ile ilgili kafamdaki tüm sorulara yanıt vererek bu anlamda benden yardımımı esirgemediği için staj amirim Merve AKMAN'a, üniversitenin üçüncü yılında re:coded'ın katılmış olduğum android bootcamp'inde bize tecrübelerini aktararak, bir projenin planlanması, araştırılması, yürütülmesi, tasarımının hazırlanması, kodlanması ve oluşturulması gibi tüm aşamaları bize öğreterek android temellerini atmamda bana yardımcı olan ilgi ve desteğini esirgemeyen,engin bilgi ve tecrübelerinden yararlandığım sayın hocam Emin AYAR' a, hazırlık sınıfında ingilizce altyapımızın oluşmasında bugün okuduğum ingilizce eserleri anlayabiliyor olmamda başta Feyyaz DEMİRER'e ve diğer emeği geçen ingilizce hocalarıma, eğitim hayatım boyunca geldiğim bu noktaya kadar üzerimde emeği geçen tüm hocalarıma ve bölüm başkanımız Dr. Öğr. Üyesi Serdar ÇİFTÇİ' ye, tüm okul hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan sevgili aileme teşekkürlerimi sunarım.

Efnan AKKUŞ

2020

ŞEKİLLER DİZİNİ

- Şekil - 1- Google Play Store Giriş Kısmı
- Şekil - 2- Google Play Store Uygulama İndirme Sayfası
- Şekil -3- Android Studio Vektörel Varlıklar
- Şekil – 4– Yeni Proje Oluşturma ve Emulator
- Şekil -5- RecyclerView
- Şekil -6- Ux Tasarım
- Şekil -7- Admin ve User Girişi
- Şekil -8- Seller Girişi
- Şekil -9- Home, Navigation ve Profil Activity
- Şekil -10- Kullanıcı Satın Alma İşlemleri
- Şekil -11- Admin Arayüz
- Şekil -12- Seller Arayüz
- Şekil -13 - Firebase kullanılan Platformlar
- Şekil -14- Firebase Projeler
- Şekil -15- Firebase Json Veri
- Şekil -16- Firebase Json Admin
- Şekil -17- Firebase Json User
- Şekil -18- Firebase Json Seller
- Şekil -19- Firebase Json Products
- Şekil -20- Firebase Json Comments
- Şekil -21- Firebase Json CartList
- Şekil -22- Kullanılan Kütüphaneler
- Şekil -23- Login Xml
- Şekil -24- Admin Arayüz Xml
- Şekil -25- Admin Ürün Düzenleme Xml
- Şekil -26- Seller Yeni Ürün Ekle Xml
- Şekil -27- Seller Login Xml
- Şekil -28- Seller Ürün Kategori Xml
- Şekil -29- Müşteri Profil Xml
- Şekil -30- Müşteri Home Xml
- Şekil -31- Adminin Seller'ın Koyduğu Ürünü Onaylaması
- Şekil -32- AdmininMusterinin alısverişini onayladığı kısım
- Şekil -33- Adminin Ürünleri Düzenlediği Kısımdır
- Şekil -34- Home Activity
- Şekil -35- Müşteri Login Sayfası
- Şekil -36- Seller'ın Yeni Ürün Eklediği Kısım
- Şekil -37- Kullanıcının Şifresini Restlediği Activity
- Şekil -38- Settings Activity

TABLolar DİZİNİ

Tablo - 1 – Android Sürümlerinin Dağılımları

Tablo - 2– Android Mimarisi ve Sistem Yapısı

Tablo- 3- Java Yazılım Geliştirme Aşaması

Tablo- 4- Firebase Analitik

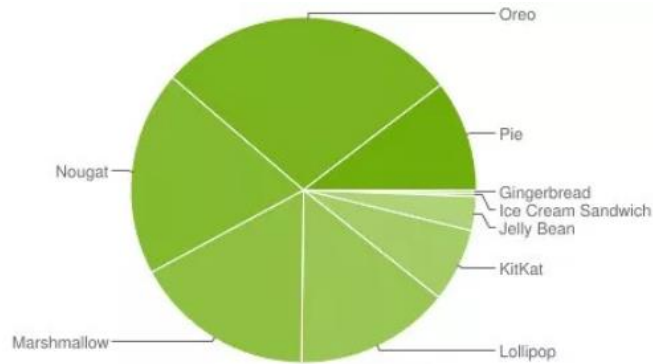
SİMGELER DİZİNİ

Java JSE	:Java Standart Edition
Java EE	:Java Enterprise Edition
Java ME	:Java Micro Edition
JVM	:Java Virtual Machine
RE	:Java Runtime Environment
JDK	:Java Development Kit
JSON	:JavaScript Object Notation
API	: Application Programming Interface
XML	: Extensible Markup Language
FCM	:Firebase Cloud Messaging
GCM	:Google Cloud Messaging
UX	:User Experience

1. Giriş

Mobil cihazlara işletim sistemlerinin (symbian, android, ios, bada, os) kurulmaya başlamasıyla Akıllı telefonlar ortaya çıkmıştır. Mobil cihazlara yüklenen ve günümüzde en yaygın olarak kullanılan işletim sistemlerinden birisi de androiddir. Android linux tabanlı, mobil cihaz ve cep telefonları için geliştirilmekte olan açık kaynak kodlu tamamen ücretsiz bir mobil işletim sistemidir. 2005 yılında Google tarafından satın alınmış olup yüzde 80 pazar payına sahiptir. Android işletim sistemi, 2012 yılı itibariyle iPhone ve Nokia hariç hemen hemen tüm cihazlarda kullanılmaya başlanmıştır. Google tarafından satın alınmasından sonra pazarda hızla büyüyen ve gelişen Android mobil işletim sistemi mobil telefonlar içinde en çok tercih edilen ve kullanılan sistem olma özelliğini sürdürmektedir. Mobil telefonlar dışında tabletler, akıllı fotoğraf makinaları, akıllı kameralar ve google gözlüğünde de android işletim sistemi kullanılmaktadır. Google tarafından ufak ama önemli kısımları kapatılmış olsada sistem açık kaynak kodludur. Android studio ortamında geliştirilen android projelerinin uzantısı apk'dır. İşletim sisteminin ücretsiz açık kaynak kodlu olması birçok kişi tarafından geliştirilmesine olanak sağladığından android işletim sistemi hızla büyüyüp gelişmiştir. Uygulamaları sadece google play store'dan indirme zorunluluğu yoktur mail yoluyla apk paylaşılabilir. Geliştirme ortamında java veya kodlin kullanılmaktadır.

Android'in google tarafından satın alınmasından kısa bir süre sonra telefon üreticileriyle Open Handset Alliance birliğini kurarak android'in geleceğini bu sayede şekillendirmiş oldular.



Tablo - 1 – Android Sürümlerinin Dağılımları

Android işletim sisteminin son üç sürümü olan Pie, Oreo ve Nougat, toplam pastanın yüzde 58'ini kapsıyor. Bu oran bir an için büyük gibi gözükse de, daha eski sürümlerin hala çok yoğun kullanım oranlarına sahip olduğunu söyleyebiliriz. Toplam Android cihaz sayısının 2,5 milyar adet olduğunu düşünürsek, her bir yüzdenin 25 milyon cihaza eşit olduğunu söyleyebiliriz. Bu da bizi bir milyarın üstünde eski sürüme sahip Android cihaz olduğu bilgisine ulaştırıyor.

1.1 Android Gelişimi ve Desteklenen Özellikler

Android 1.0 veya Android Apple Pie HTC Dream'in işletim sistemi olarak telefonlara ilk adımını atmıştır. Bu sürüm; kamera desteği Wi-Fi, bluetooth, klasörler, bir web tarayıcısı, uyarılar, sesli arama, YouTube, alarmlı saat, galeri gibi özellikler sunmaktadır. 9 Şubat 2009 da Android 1.1 veya Android Banana Bread yayınlanmıştır. Bu sürüm bir önceki sürümdeki sistem hatalarını düzeltmeyi ve API'yi geliştirmeyi amaçlamıştır. Cupcake sürümünde bluetooth desteği, kamera kaydı, video gösterimi, yazı tahmin edebilen klavye ve animasyonlu ekran, Donut sürümünde doğru çeviri yapma ve okutma özelliği, Eclair sürümünde Html 5 ve Bluetooth 2.1 desteği, Gingerbread sürümünde çoklu kamera ve çoklu dokunmatik desteği, Honeycomb sürümünde tablet desteği, Ice Cream Sandwich sürümünde yüz tanıma fonksiyonları NFC ile dosya paylaşımı özellikleri ve son olarak Jell Bean sürümünde aynı anda iki uygulama açma özelliği ve çoklu kullanıcı, 360 derece panoramik fotoğraf ve Swype klavye kazandırılmıştır. eklendi. Android market olarak faaliyet gösteren uygulama indirme sitesinin ismi de Mart 2012 yılında Google Play olarak değiştirildi. Jelly Bean Plus ise 11 Şubat 2013'te yayınlanmıştır. Bu güncelleme ile birlikte performansı ve kararlılığı artarak telefonlardaki ısınma ve donma problemleri giderilmiştir. Aynı yılın Temmuz ayında bir sonraki sürümüyle tekrar yayınlanan Jelly Bean çoklu kullanıcı için ebeveyn kontrolü, oyunlar için OpenGL ES 3.0 desteği, diğer teknolojik aletlere bağlanmak için ilk akıllı Bluetooth teknolojisi, Wi-Fi'ı açıp kapatmak yerine otomatik açılıp kapanabilen akıllı Wi-Fi teknolojisi eklenmiştir. Ayrıca uygulama izinleri bu sürümle birlikte kontrol edilebilir hale gelmiştir. Son sürümü takip eden birkaç ay sonra ise çıkması beklenen Key Lime Pie yerine Android KitKat yayınlanmıştır. Bu sürüm ile kullanıcılara görsel arayüzde birçok değişiklik yapılmıştır. RAM optimizasyonu sayesinde 512 MB RAM'e sahip cihazlarda da çalışabilmektedir. ART (Android Run Time) sanal makinesi Android 4.4 Kitkat Nexus ile deneysel olarak sunulmuştur. 24 Şubat 2014'te MWC fuarında Sony Xperia Z2 ve Samsung Galaxy S5, Android 4.4.2 KitKat kullanmıştır. Cihazlar bu işletim sistemiyle satışa sunulacaktır. Aynı yılın Kasım ayında Google Android Lollipop sürümünü çıkarttı. Google Mayıs 2015'de yaptığı I/O 2015 konferansında Android M 6.0'ı duyurdu ve önizleme sürümünü yayınladı. Aynı yıl Google Android 6.0'ın kod adını Marshmallow olarak duyurdu ve 3. önizleme sürümünü yayınladı. 29 Eylül 2015'de Android Marshmallow bulunan Nexus 5X ve 6P tanıtılmıştır. 2016'da Google, Android N'in Geliştirici Beta'sını yayınladı. 2017 Yılında ise Android 7.1.1 Sürümüne ait fabrika image dosyalarının yanı sıra Ota güncelleme dosyaları geliştiriciler sayfasından indirilmeye sunuldu.

Sürekli güncellenerek her yeni sürümüyle yeni özellikler eklenen android işletim sisteminin, gelişimiyle birlikte hayatımızda daha fazla yer edinmeye devam edecek gibi gözükmektedir.

1.2 Android Kullanım Alanları

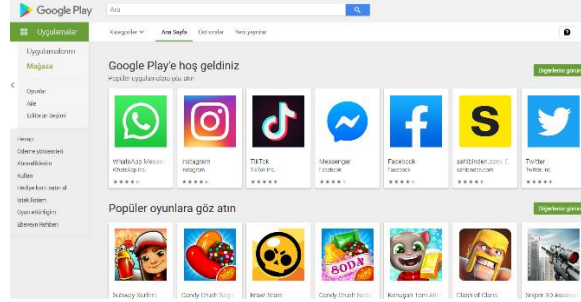
Android işletim sistemi ilk günden bu yana çok büyük gelişimler gösterdi. İlk olarak akıllı telefonlar ile hayatımıza giren Android işletim sistemleri, artık bir çok alanda kullanılmakta ve hayatımızda daha sık yer edinmektedir. Bu işletim sisteminin bazı yerlerdeki kullanım alanları ise insanı şaşırtacak cinsten, bunların birkaçını ele alacak olursak. Android artık yavaş yavaş evlerimizde sıklıkla kullandığımız cihazlar içerisinde yerini almaya başlıyor. Google 2011 yılında, Android Home adını verdiği ev otomasyon sistemini tanıttı. Bu teknoloji sayesinde ev ortamında yer alan her türlü aydınlatma, priz ve termostat tarzı cihazlar Android işletim sistemini kullanan telefon ve tabletlerden yönetilebilir hale gelmektedir. Android bununla yetinmeyip evimizin mutfağına da giriyor. Panasonic firması yakın bir zamanda Android işletim sistemine sahip bir mikrodalga fırın ve prinç pişirme makinesi geliştirdiğini açıkladı. Bunun dışında dışında kendisini yakından tanıdığımız Samsung firması da kendi geliştirdiği Android buzdolabını tanıttı.

Android işletim sisteminin bir diğer kullanım alanı ise, günlük hayatta ev ve iş yerlerinde sıklıkla kullandığımız yazıcılardır. Android işletim sistemli yazıcı, ilk olarak HP tarafından çıkarıldı. Fakat Android kitkat sürümüyle birlikte gelen gelişmiş yazdırma seçenekleri bu tür cihazların kullanımını gereksiz kıldı. Android işletim sistemlerinin kullanıldığı bir diğer ilginç kullanım alanı ise otomotiv sektörü. Zaten Google firmasının otomobillerde kullanılmak üzere standart bir Android sürümü geliştirme çalışmalarının olduğunu biliyoruz. Bir zamanların yaygın olarak kullanılan teknolojik aleti olan, müzik çalarlar'da Android işletim sisteminin kullanım alanlarından biri haline geldi. Creative, Samsung ve Sony gibi firmalar Android işletim sistemli bazı müzik çalar modellerini piyasaya sürdü. Fakat bu ürünler beklenen ilgiyi göremedi.

Android işletim sistemlerinin arkasında Google gibi büyük bir teknoloji devinin yer aldığını biliyoruz. Fakat Google'ın dışında Android işletim sistemini destekleyen ve arkasında duran bir çok teknoloji firması bulunmaktadır. Open Handset Alliance HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia gibi firmalar, bu firmalardan sadece birkaç tanesidir. Android işletim sisteminin büyük teknoloji devlerini arkasına alması, şu andaki piyasada bulunan pazar payının çoğunluğuna sahip olması, kullanıcı kitlesinin ve kullanım alanlarının her geçen gün artması gibi bir çok gelişme, bu işletim sisteminin geleceğinin parlak olduğunu göstermektedir.

1.3 Google Play

Google tarafından yönetilen Google Play, Android cihazların uygulama ve oyunlara ulaşabileceği çevrim içi bir markettir. Buradan, çoğu zaman geliştiricilerin tercihleri doğrultusunda ülkelere göre yayınlanan oyun ve uygulamalara erişilebilir. Android cihazlar üzerinden Google Play uygulamasıyla ya da Google Play sitesinden erişilebilir.



Şekil - 1 – Google Play Store Giriş Kısmı

Google Play, üzerinde yer alan uygulamaları kategorilerine göre ayırarak kullanıcılara sunar. Ana sayfasında en çok satılan ve talep gören uygulamalara yer verir. Ayrıca "Editörün Seçimi" gibi yeni bir kategorilendirmeye daha da gitmiştir. Yukarıdaki resimde yer alan arama kutusuna ilgilendiğimiz bir konuyu ya da geliştirici firma adını girerek kolayca arama yapabiliriz. Bir uygulamayı seçtiğimizde aşağıdaki uygulama detay sayfasına yönlendiriliriz. Bir uygulamanın ayrıntı sayfasına girdiğimizde, uygulamanın adını, hemen altında geliştiricisini görürüz. Yükle düğmesine tıkladığımızda uygulama Google hesabımıza bağlı Android cihazlarımıza gönderilir. Uygulama detay sayfasında aşağı doğru ilerlediğimizde, uygulama içi görselleri bulabiliriz. Görsellerin hemen altında geliştirici tarafından yapılan açıklamalara ve uygulamayı indiren kullanıcıların yorumlarını görme şansımız var. Kullanıcılar çok çeşitli amaçlarla da yorumlar yapabilirler.

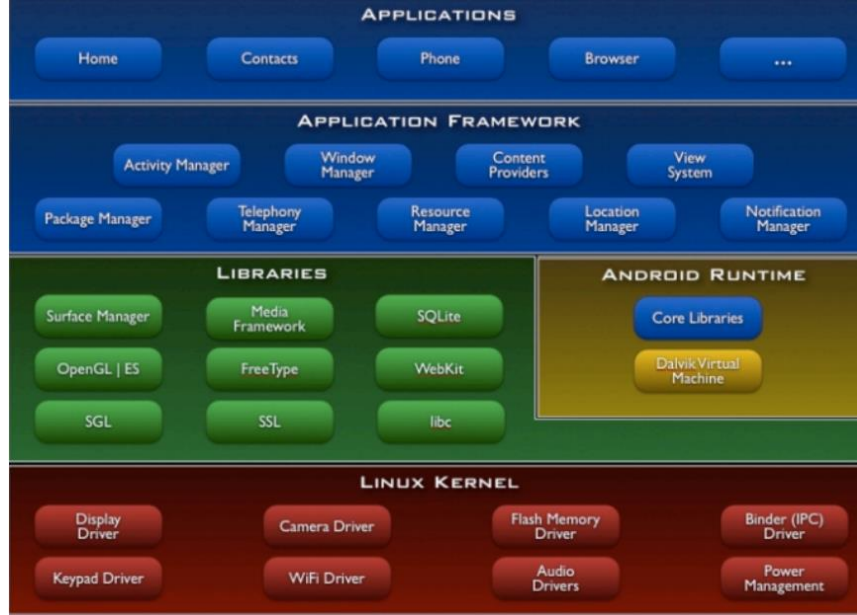


Şekil - 2 – Google Play Store Uygulama İndirme Sayfası

Detay sayfasında daha aşağıya indiğimizde uygulamaya dair ek bilgiler karşımıza çıkar. Burada uygulamanın hangi sürümde olduğu, boyutu ve hedef Android sürümleri görülebilir. Burada yer alan geliştirici bağlantı sayfaları ise, geliştiriciye ulaşabilmemiz açısından, geliştirici tarafından paylaşılan sayfalardır. Sayfanın sonunda ise benzer uygulamalar ve bu geliştiricinin diğer uygulamalarına da bakabilirsiniz.

1.4 Android Mimarisi ve Sistem Özellikleri

Android, Open Handset Alliance, Google ve özgür yazılım topluluğu tarafından geliştirilen bir mobil işletim sistemidir. Bu işletim sisteminin parçaları çekirdek (kernel), sistem kütüphaneleri, uygulama geliştirme çatıları (frameworks) ve yerleşik temel uygulamalardan oluşmaktadır. Android mimarisi aşağıdaki gibidir.



Tablo - 2– Android Mimarisi ve Sistem Yapısı

Android, Linux çekirdeğini (kernel) kullanır. Linux çekirdeğine Android için eklenen kod parçacıkları ve kütüphaneler Genel Kamu Lisansı'na sahipken, diğer bileşenler üretici firmalarına kendi kapalı ROM'larını oluşturmalarına izin verecek ama yine özgür bir şekilde Apache Lisansı ile dağıtılmaktadır.

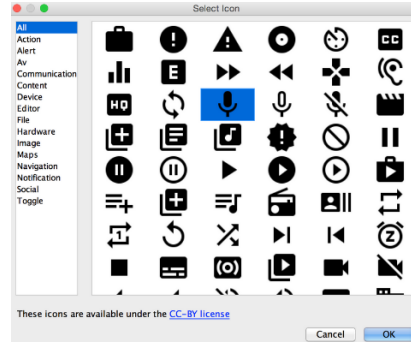
Linux çekirdeğinin doğrudan kaynak sağladığı yapılar security (güvenlik), memory ve process (hafıza ve süreç) kontrolü, dosyalama ve bağlantı için I/O işlemleri ve cihaz sürücüleridir. Çekirdekte Android için özelleştirilmiş başlıca alanlar ise güç kontrolü, paylaşılan hafıza, low memory killer ve süreçler arası iletişim içindir.

Mimarinin diğer önemli yapısı olan kütüphaneler bölümünde C ile yazılmış sistem kütüphaneleri, internet tarayıcısı (browser) motorlarının çalışması için Webkit, görüntüleme kontrolünü yapan Surface Manager, grafik işlemleri için OpenGL, ses ve video işlemleri için gereken Media Framework, veri yapıları kontrolü ve düzenlenmesi için SQLite gibi yapılar bulunur.

1.5 Android Studio Ortamı

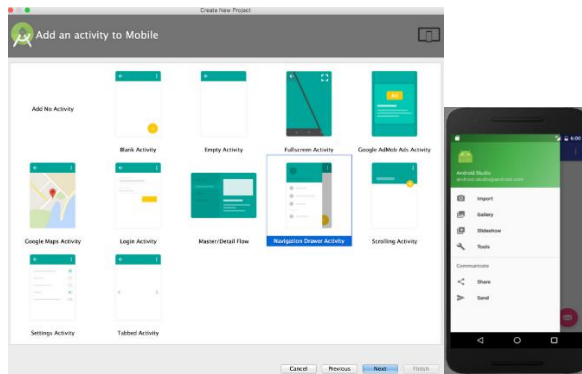
Android Studio, Android uygulamalarının geliştirildiği, üst seviye özelliklere sahip ve Google tarafından da önerilen resmi programlama aracıdır. Android Studio'nun kod geliştiricilere sunduğu temel özellikler şunlardır:

- Gradle tabanlı, esnek proje inşa sistemi.
- Hızlı ve zengin özellikli bir emülatör
- Farklı özellik ve sürümlere göre çoklu APK çıktısı.
- Genel uygulama özelliklerini oluşturmanıza ve örnek kodu içe aktarmanıza yardımcı olacak kod şablonları ve GitHub entegrasyonu
- C ++ ve NDK desteği
- Ekran tasarımlarını kolaylaştıran sürükle-bırak özellikli zengin editör.
- Uygulamanın performansı, kullanılabilirliği, farklı sürümlerde çalışabilirliğinin kontrol edilebileceği test araçları ve frameworkler
- Kolay ve güvenli APK imzalanması.
- Ek uğraşa gerek kalmadan Google hizmetlerini uygulamaya ekleyebilme.



Şekil -3– Android Studio Vektörel Varlıklar

Android Studio geliştirme ortamı, tasarım aracı olarak Vektörel Varlıklar'ı içeriyor. Uygulamalarınızda kullanabileceğiniz, özgür yazılım lisanslı pek çok vektörel grafik bu pakette mevcut. Projelerinizde res/drawable sağ tıklayıp New/Vector yolunu izleyerek dilediğiniz vektörel çizimi uygulamalarınıza dâhil edebilirsiniz.



Şekil - 4– Yeni Proje Oluşturma ve Emulator

1.6 Layout

Android uygulamalarında ekran tasarımları res klasörü altında bulunan layout dosyaları ile belirlenir. Bu dosyalar xml formatında hazırlanan dosyalardır ve Android uygulamalarına özel etiketler kullanarak görsel öğelerin yerleşimlerini ve özelliklerini bildirirler.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="@color/colorProfile"
    tools:context=".FriendsActivity">

    <TextView
        android:id="@+id/all_friends"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="Friends"
        android:textAlignment="center"
        android:textColor="@android:color/background_light"
        android:textSize="20sp"
        android:textStyle="bold" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/friend_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/all_friends"
        android:layout_alignParentStart="true"
        android:layout_margin="15dp" />

</RelativeLayout>
```

Şekil -5- RecyclerView

Bir ekranın görünümü genellikle iki farklı yerleşim tipi kullanılarak belirlenir. Bunlar RelativeLayout ve LinearLayout olarak ikiye ayrılır.

RelativeLayout görsel öğeleri diğer öğelere göre referans olarak dizer. RelativeLayout tasarımında ilk eklenen öğe ekranın en tepesinde yer alır. Daha sonra eklenen öğeler android:layout_below (verilen ID'nin altında), android:layout_above (verilen ID'nin üzerinde), android:layout_toLeftOf (verilen ID'nin solunda) ve android:layout_toRightOf (verilen ID'nin sağında) şeklinde sıralanır. Bu şekilde ekranda oluşturulan bütün öğeleri bir öğeyi referans olarak dizmemiz mümkündür. Öğelerin ebatları ise android:layout_height ve android:layout_width metotlarıyla belirlenir.

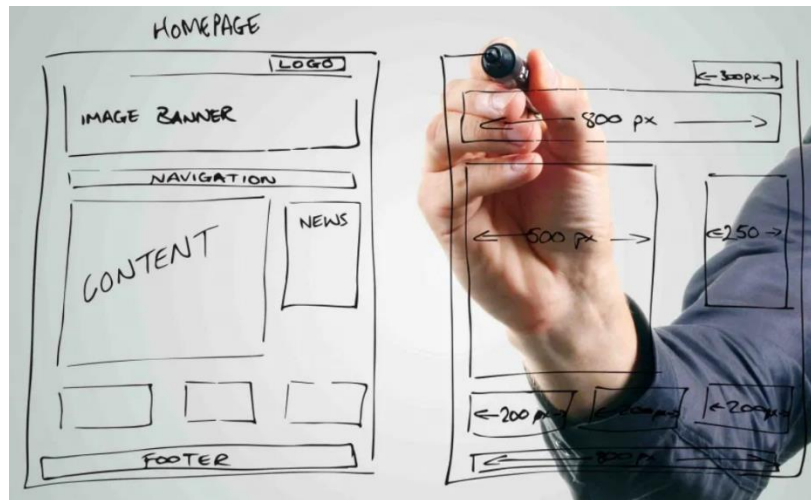
Yukarıdaki örnekte, RecyclerView öğesinin altına başka bir öğe eklendiğinde ilk olarak liste ve listenin altında bu yeni öğe görüntülenir. Öğelerin uzunluk ve genişlikleri android:layout_width ve android:layout_height özellikleriyle belirlenir. Burada fill parent (ana öğe ne kadar genişse hepsini doldurur) ve wrap_content (öğede içerisinde yer alan metin veya resim kadar yer kaplar) değerleri kullanılabileceği gibi, piksel cinsinden uzunluk da verilebilir. Eğer piksel vererek ebat belirleyeceksek, px birimi yerine dp birimini kullanmamız faydalı olacaktır. Android cihazlarda farklı ebatlarda ekranlar olduğundan, px cinsinden verilen piksellerin tasarımı her cihazda farklı olacaktır. dp birimi ise ekran boyutuna göre ölçeklendirme yapmaktadır ve farklı cihazlarda benzer görünümler elde etmenizi sağlar.

1.7 Ux

UX İngilizce User Experience Türkçesi ile Kullanıcı Deneyimi kelimelerinin kısaltması olarak kullanılıyor. Seksenli yılların başından bu yana hayatımıza yavaşça ve fark ettirmeden sokulan bir kavram olarak karşımıza çıkan Kullanıcı Deneyiminin ilk örneklerini 21. Yüzyılın dâhisi Steve Jobs hayatımıza adapte ediyor. Jobs Atari firmasında iken başladığı Kullanıcı Odaklı Tasarımlarına Mac ile devam ettiğini görüyoruz.

Kullanıcı deneyimi adı üstünde hazırladığımız bir tasarım, ürün vb çalışmayı asıl kullanacak kişilerin kullanarak deneyimlemelerinden ibaret. Burada ibaret kelimesini hiçte küçümsememek lazım. Zira UX olmazsa hiç bir ürün tam anlamı ile başarı sağlamaz. Konuya, Kullanıcı Deneyimleri doğrultusunda tasarlanmış bir e-ticaret sitesi ile kullanıcı deneyimi gözardı edilmiş bir e-ticaret sitesini karşılaştırarak daha fazla ışık tutabiliriz. Her iki sitedeki ürün gurupları fiyatlar, kampanyalar aynı olsun. Her iki siteye de aynı günün aynı saatinde aynı tepkileri veren aynı sayıda ziyaretçi girsin. UX gözardı edilen e-ticaret sitesine girenlerin yüzde sekseninden fazlası siteyi daha ana sayfadayken terk ettiklerini görebiliriz. Buna karşın UX yani Kullanıcı Deneyimi göz önünde bulundurularak tasarlanan sitede ise girenlerin yüzde sekseninden fazlasının sitede daha uzun vakit geçirdiğini ve diğer sayfalarda gezinmeye devam ettiğini görebiliriz.

UX' un kapsamı sadece UI' den yani Kullanıcı Arayüz tasarımından ibaret değildir. Sitenin açılma hızı, karşınıza çıkan mesajlar, alışverişte ne kadar adımdan geçtiğiniz, aradığınız ürünü bulma hızı ve kolaylığı, renkler, hiyerarşik tasarım, ürün açıklamaları, ürün görselleri, ürün yorumları ve daha birçok unsur UX tasarımını etkileyen faktörlerdir. UX tasarımınız kullanıcılara en iyi deneyimi yaşatmak üzerine odaklanmış olması gerekir. Bu nedenle siteye ilk girişten ve hatta daha öncesinden (reklam, sosyal medya vb) sipariş teslim ve satış sonrası hizmetlere kadar uzanan geniş bir alandır. Bunlardan birinin eksikliği sizin satış kaybınız anlamına gelir.



Şekil -6- Ux Tasarım

2. Benzer Çalışmalar/Literatür Taraması

Projeye başlamadan önce evcil hayvan ürünlerinin alımının ve satımının yapıldığı e-ticaret mobil uygulamaların araştırması yapılmıştır. Yaptığım araştırmalar sonucunda aynı şekilde evcil hayvan eşyaları satan iki mobil uygulama buldum.

Petlove(Pazar yeri Brezilya): Brezilya'daki en büyük çevrimiçi petshop ve evcil hayvanınızın ihtiyacı olan her şeyi satıyor. Artık mağazalara gidip gelmeyecek: Petlove'un evcil hayvan uygulamasıyla, ülke genelinde evde verilen 10.000'den fazla ürüne uygulama üzerinden erişebileceksiniz. Uygulamada köpek evi, kedi tırmığı, köpek yatağı, evcil hayvanınız için yatak, köpek ve kedi oyuncaklar, akvaryumlar ve daha fazlasını arayabilirsiniz. Uygulama ülkedeki en iyi marka ürünlerin satışını sağlıyor: Altın feed, Premier feed veya Royal Canin feed! Ayrıca, Bravecto, Frontline ve Nexgard gibi hat üstü ürünler ve pire karşı ürünler sunuluyor.

Zoobio (Pazar yeri Almanya): Zoobio yeni nesil bir evcil hayvan mağazasıdır. Zoobio, hayvanlar için 40.000'den fazla ürün bulabileceğiniz uygun bir evcil hayvan süpermarketidir. Sadece birkaç tıklamayla siparişiniz kapınıza geliyor. Evinizden veya iş yerinizden ayrılmadan hayvanlarınızın Zoobio ile ihtiyaç duydukları şeyleri sipariş edin. Zoobio ile alışveriş yapmanın faydaları:

250'den fazla ürün kategorisi. Evcil hayvanlarınızın çok çeşitli ihtiyaçları, zevkleri ve tercihleri olduğunu biliyoruz. Bu nedenle geniş bir evcil hayvan ürünleri kataloğumuz var. Köpek kulübeleri, kuş kafesleri, tavşan ve hamster, akvaryum ve teraryumlar - tüm bu barınaklar yeni sakinlerini bekliyor. Konfor ve eğlence için köpek yatakları, yastıklar ve şilteler, kediler ve köpekler için oyuncaklar ile aksesuarlar, tüneller ve diğer renkli ve orijinal aksesuarlar sunuyoruz.. Peki ya yürüyüşler ve hayvanları taşıma? İhtiyacınız olan her şey burada geniş taşıyıcılar, rahat kedi ve köpek koşum kayışları, tasmalar, ağızlıklar ve daha fazlası dahil. Evcil hayvanlarınızı mutlu edecek her şeye sahibiz! 10 000 yiyecek maddesi. Kataloğumuzda kuru ve ıslak köpek maması, gevrek kroket ve kedi konservesi, dengeli beslenme sunan balık yemi ve atlar, kuşlar, kemirgenler, sürüngenler ve diğer evcil hayvanlar için sağlıklı gıdalar bulacaksınız. Çevrimiçi evcil hayvan mağazamızda, özel beslenme ihtiyaçları olan bir evcil hayvan için bir diyet seçebilirsiniz: örneğin, büyüyen, bakıcı evcil hayvanlar için, kilo veren veya ameliyat sonrası iyileşenler için. Zoobio herkes için doğru yiyecek bulacaktır.

3. Önerilen Yöntem

Adminin müşteri ve satıcı ile koordineli çalıştığı e-ticarete farklı bir bakış açısı kazandıran müşterilerine çok daha kolay bir şekilde ulaşabilecek mobil destekli bir uygulamadır. E-ticaretin günümüzde yaygınlaşmasıyla birlikte bazı sıkıntılarda ortaya çıkmıştır örneğin uzun süren teslimatlar, yetersiz ürün bilgisi, alınan ürünün kalite seviyesinin düşük olması, iade süreci, müşteri hizmetleri, yanlış ürün gönderimi bu gibi durumlarda güvenilir bir e-ticaret ortamına olan ihtiyaç artmıştır.

Projenin amacı, evcil hayvan sahipleri için farklı türde binlerce ürün sunarak onların ihtiyaçlarını en iyi şekilde karşılayacak ürünü bulmalarını kolaylaştırmak. En kaliteli ürünleri uygun fiyata satın almayı amaçlayan bu uygulamayı online bir evcil hayvan süpermarketi olarak düşünebiliriz. Sadece birkaç tıklamayla siparişin kapıya kadar geliyor olması müşteri için evden veya iş yerinden ayrılmadan hayvanlarının tüm ihtiyaç duyduğu ürünleri sipariş ederek zamandan kazanç sağlayabileceklerdir.

Projenin kapsamında müşterinin sepete ürün eklemesi, aradığı ürünü filtreleyerek bulabilmesi, satış işlemi tamamlayıp ürünü alabilmesi, ürünün kullanıcı yorumlarını ve puanlarını değerlendirebilmesi gibi detaylar yer alacaktır.

Bir e-ticaret uygulaması olan PetShop'a üç farklı kullanıcı türü giriş yapabilecek. Birinci kullanıcı modeli admin olarak uygulamaya giriş yapabilecek bu kişi uygulamanın arka planında çalışacak (petshop sahibi ya da satış ve pazarlamadan anlayan herhangi biri olabilir) admin aynı zamanda müşterilerine uygulama üzerinden de ulaşarak ürünlerini daha kolay ve reklam ücreti ödemedi satabilecek. Satışından söz edilen bu ürünler evcil hayvan sahiplerinin tüm ihtiyaçlarını karşılayabilecek nitelikteki ürünlerdir. Örneğin, kuş yemi, kuş kafesi, kedi maması, tasma ...

İkinci kullanıcı modeli ise, evinde bu ürünlerden ihtiyacından fazla bulunan ya da gereksiz olduğunu düşündüğü ürünlerini satarak para kazanmak isteyen üyelerdir. Burada satıcılar belli bir miktar ürün satışından sonra her sattığı üründe admine bir miktar komisyon verecektir. Buradaki amaç satıcıların uygulamaya giriş yapar yapmaz hemen komisyonlardan bahsetmeyerek uygulamayı kullanmaya daha çok teşvik etmektir.

Satıcıların ürün bilgileri ile ilgili yaptığı fotoğraflı içerikler admin tarafından uygun bulunup onaylandıktan sonra satışa sunulacaktır. Bunun sebebi ise kötü niyetli insanların satıcı olarak uygulamaya giriş yapıp uygun olmayan içerik paylaşımlarının önüne geçmek.

Üçüncü ve son kullanıcı modeli ise bu ürünleri satın almak isteyen üyelerdir. Müşteriler uygulamaya giriş yaptıklarında hem satıcıların hemde adminin satışını yaptığı ürünleri görebilecek ve aralarında kıyaslama yaparak ürüne sahip olabilecekler.



Şekil -7- Admin ve User Girişi

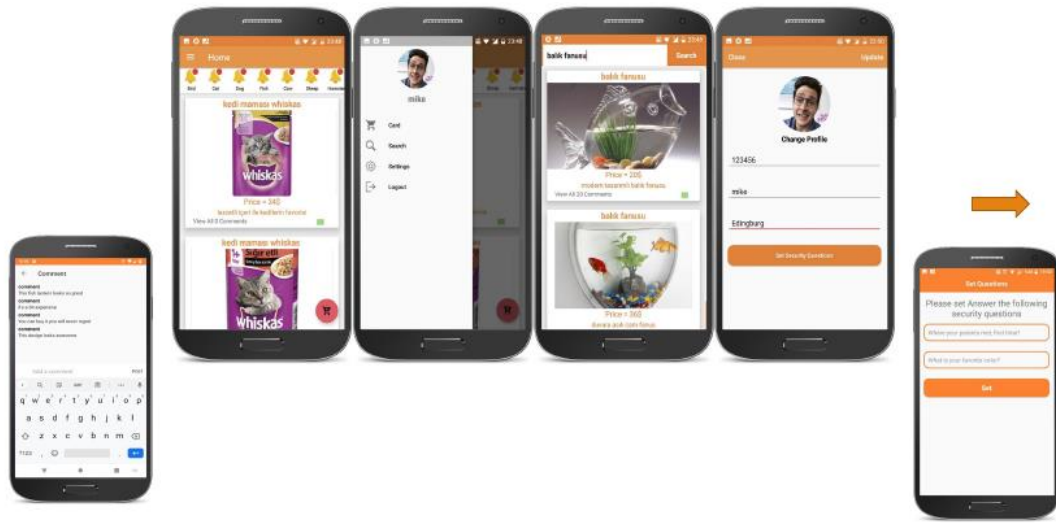


Şekil -8- Seller Girişi

Son ürün bir mobil uygulama olacağı için hedef kullanıcıların Play Store'dan uygulamayı indirdikten sonra bazı bilgilerini girerek üye olmaları gerekmektedir. Üyeliklerinden sonra uygulamanın ana sayfasında bulunan ürünleri inceleyerek diledikleri ürünü sepete ekleyerek satın alabileceklerdir. Kullanıcılar uygulamada filtreleme ve kategori seçimi yaparak seçtikleri ürünlerin siparişini verdikten sonra uygulamadaki siparişler kategorisinden verdikleri siparişleri takip edebilirler. Uygulamaya satıcı olarak giriş yapan kullanıcılar ise satış gerçekleşene kadar tüm aşamaları takip edebilecekler ve satıcının satış işlemi admin onayından sonra gerçekleştirilecektir.

Uygulama giriş kısmı 3 aşamalı olacaktır müşteriler, satıcılar ve admin şeklinde müşterilerin, satıcı gibi ürün satma yetkileri olmayacak aynı şekilde satıcıların ise sadece ürün satma yetkileri olacak ürün satın alamayacaklar. Adminin ise satıcının satacağı ürünü onaylama yetkisi olacak ve tabi kendi tarafından satılacak ürünleri uygulamaya koyabilecek ve müşterinin spariş verdiği ürünlerin paketlenip teslimat aşamasının yapılmasını ve onaylanması işlerinin takibini yapacak.

Başarı hedefi tek tip ürün satışı yapan diğer e-ticaret uygulamaları gibi sektörde uzun soluklu kalabilmek. Örneğin sadece kitap, kahve, şifalı bitkiler ya da teknolojik aletler satan diğer uygulamalar gibi.



Şekil -9- Home, Navigation ve Profil Activity

Müşterilerin uygulamaya giriş yaptıktan sonra; uygulama içinde aradığı ürünü bulma, ürün hakkında yapılan yorumları okuyarak diğer ürünler ile kıyas yapabilme, satın alma ve kendi profilini düzenleme işlemlerini yaptığı kısımdır.

Anasayfada bulunan navigation(sürüklenebilir bar) sayesinde kullanıcı alışveriş sepeti, arama, ayarlar ve çıkış sayfalarına kolay bir şekilde gidebilir.

Kullanıcı satın almak istediği ürünü eklediği sepetinde ürünleri satın al butonunu tıkladığında ürün kontrol sayfasından kredi kartı bilgilerini girdiği sayfaya aktarılmaktadır ve satın alma işlemi gerçekleşmektedir.

Kullanıcı ürün satın almadan önce almak istediği ürünü arama motoru sayesinde dilediği gibi aratarak ürüne erişebilir.

Eğer kullanıcı şifresini unutmuş ise uygulamaya daha önceden kaydettiği güvenlik şifresini girerek giriş yapabilir. Güvenlik şifresinde kullanıcılardan cevaplanması beklenen sorular arasında en sevdiği renk, annesi ile babasının tanıştıkları şehir ismi gibi sorular yer almaktadır.



Şekil -10- Kullanıcı Satın Alma İşlemleri

Kullanıcı bu bölümde; anasyfada gördüğü ürünleri yakından tüm ayrıntılarına kadar inceleyebilmek için ürünün üzerine tıklar ve ürün detay sayfasına aktarılır. Ürün detay sayfasında ise kullanıcı satın alacağı ürün miktarını seçebilir, ürünü yarumlayıp, puan verebilir ve son olarak ürünü sepetine ekler.

Kullanıcı ikinci sayfada aldığı tüm ürünlerin listesini görebilir ve en son kontrol etme fırsatına sahip olur. Burada kullanıcı istemediği ürünleri sepetinden çıkarabilir.

Kullanıcı satın almak istediği ürünleri seçtikten sonra ödeme sayfasına yönlendirilir ve bu sayfada kullanıcının kredi kartı bilgilerinin girilmesi istenir böylece ödeme işlemlerini gerçekleştirir.

Son olarak kullanıcı satın alma işlemini tamamladıktan sonra ürünü eline geçene kadar her adımını kargo takip numarası ile takip edebilir. Böylece kargosunun nerede olduğunu kolayca görebilir.

Kısacası kullanıcı bu bölümde; satın alacağı ürünlerisepete ekleme, puanlama ve ödeme işlemlerini gerçekleştirir.



Şekil -11- Admin Arayüz

Admin, seller tarafından eklenen ürünleri onaylar bu ürünlerden uygun olmayanları listeden kaldırabilir ya da satışını onaylayabilir. ve eklenen bu ürünlerde ürün adı, fiyat ve açıklaması yönünden bazı düzenlemeler yapar. Admin aynı zamanda bu bölümde müşteriler tarafından sipariş verilen ürünleri inceler ve satışını onaylar.



Şekil -12- Seller Arayüz

Seller olarak üye olup giriş yaptıktan sonra, eklenen yeni ürünlerin admin tarafından onaylanmasının ardından satışa sunulduğu bölümdür. Seller burada önceden satışa sunduğu ürünlerin listesini görebilir. Yeni ürün ekleyebilir eklediği yeni ürün admin tarafından onaylanır ise, müşterinin anasayfasında satışa sunulur.

4. Kullanılan Teknolojiler

4.1 Java

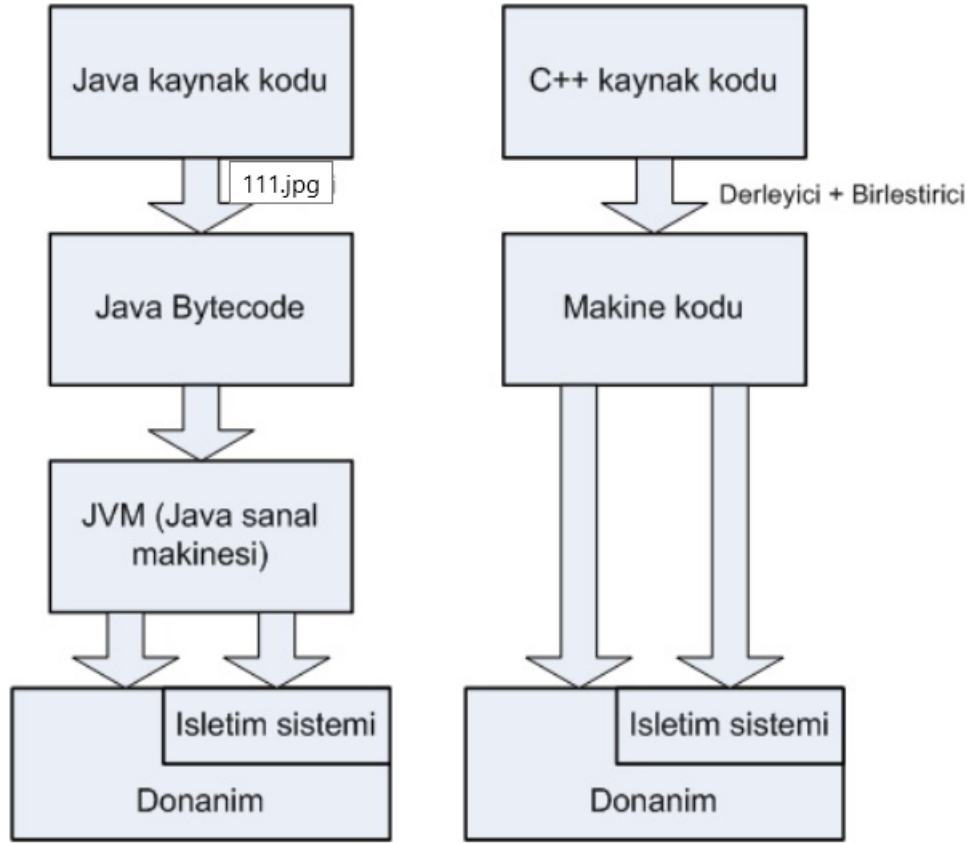
Android uygulamaların yapı taşlarını java dilinin temel kavram ve prensipleri oluşturur. Dolayısıyla bu yola baş koyan bir android severin, java dilinin temel kavram ve prensiplerini çok iyi talim etmesi gerekir.

Java nesne yönelimli programlama tekniğini benimsemiş, platformlardan bağımsız olarak çalışan çok işlevli bir programlama dilidir. Java diliyle geliştirilen uygulamaların platformdan bağımsız olarak çalışması, bu uygulamaların her işletim sistemi üzerinde çalıştırılabilir olduğu manasına gelmektedir. Java dilinin sahip olduğu bu özellik, kendi kategorisinde bulunan diğer dillerden kendisini ayıran en önemli özelliklerinden bir tanesidir. Java diliyle masaüstü uygulamaları dahil olmak üzere, web uygulamaları, mobil cihazlara yönelik uygulamalar gibi bir çok alanda ve cihazda kullanılabilecek uygulamalar geliştirmek mümkündür. Java diliyle uygulama geliştirmek için kullanılan 3 farklı java sürümü mevcuttur. Bu sürümler geliştiricilere ücretsiz olarak sunulmaktadır. Bunları sırasıyla tanıyalım: Java JSE(Java Standart Edition): Sunucu uygulamaları, masaüstü uygulamaları geliştirmek için gerekli olan araç ve kütüphaneleri içerisinde barındıran en temel sürümdür. Java EE (Java Enterprise Edition): Geniş ölçekli kurumsal ve web uygulamaları geliştirmek için kullanılan sürümdür. Java ME (Java Micro Edition): Taşınabilir mobil telefonlar, tabletler vb. cihazlar ve PDA ve TV setleri gibi aygıtlar için uygulama geliştirmek için kullanılan sürümdür. Java diliyle ilgili bahsedilmesi gereken diğer kavramlar ise, bir önceki bölüm-den anımsadığımız JVM, JRE ve JDK kavramlarıdır. JVM (Java Virtual Machine): Türkçede java sanal makinesi olarak adlandırılan bu bileşen, java diliyle yazılmış olan kodların tüm platformlarda çalıştırılmak üzere geliştirilmiş bir bileşendir. Bununla birlikte her platforma uygun ayrı sürümleri de mevcuttur. Dolayısıyla Java, platformlardan bağımsız olarak çalışması ve bir kere yaz her yerde çalışsın felsefesini JVM ile gerçekleştirebilmektedir. JRE (Java Runtime Environment): Java programlarını çalıştırmak için gerekli olan yazılımdır. İçerisinde JVM ve Java kütüphanelerini de barındırır. Dolayısıyla bir bilgisayar üzerinde, java diliyle geliştirilmiş bir uygulamayı çalıştırabilmek için bu yazılımın bilgisayara kurulu olması gerekmektedir. JDK (Java Development Kit): JDK 'nın bir önceki bölümde, java uygulamalarını geliştirmek için gerekli olan tüm araçları barındıran bir yazılım geliştirme ortamı olduğundan bahsetmiştik. Burada JDK ile ilgili değinmek istediğim bir diğer husus ise, JDK'nın bünyesinde JVM ve JRE'yi de barındırdığı gerçeğidir.

Java bir programlama dilinden daha fazlasıdır. Java kendi başına bir platformdur. Java “platform bağımsızdır” ifadesi buradan gelir. Kendi platformunun yetenekleri ile çok çeşitli cihazlarda kullanılabilir özelliktedir. Java nesne yönelimli bir dildir. Bu sebeple Soyutlama (Abstraction), Paketleme (Encapsulation), Kalıtım (Inheritance) ve Çok Biçimlilik (Polymorphism) özelliklerine sahiptir. Java yüksek performans vadeder. Java ile program geliştirmek için gereken tüm bileşenler ücretsizdir. Ücretsiz bir çok Java geliştirme ortamı (Netbeans, Eclipse) mevcuttur. Java öğrenmesi basit, multi threading yeteneklerine sahip, dinamik, yüksek seviyeli, adım adım çalıştırılan ve taşınabilir yapıdadır.

Java çok fazla olanak sunan, yapabildiklerinin insanın hayalleriyle sınırlı olduğu web platformu için vazgeçilmez bir açık kaynak kodlu yazılım platformudur. Java Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış gerçek nesneye yönelik, platform bağımsız, yüksek performanslı, çok işlevli, yüksek seviye bir program dilidir. Java ilk çıktığında daha çok küçük cihazlarda kullanılmak için tasarlanmış ortak bir platform dili olarak düşünülmüş. Ancak bakmışlar ki platform bağımsızlığı özelliği C ve C++’tan çok daha üstün ve güvenli bir yazılım geliştirme ve işletme ortamı sunuyor, hemen her yerde kullanılmaya başlanmış. Şu anda özellikle kurumsal alanda ve mobil cihazlarda son derece popüler olan java, özellikle J2SE 1.4 ve 1.5 sürümü ile masaüstünde de gücünü arttırmayı hedefliyor. Java’nın ilk sürümü olan Java 1.0(1995) Java Platform 1 olarak adlandırıldı ve tasarlama amacına uygun olarak küçük boyutlu ve kısıtlı özelliklere sahipti. Daha sonra platformun gücü gözlemlendi ve tasarımında büyük değişiklikler ve eklemeler yapıldı. Bu büyük değişikliklerden dolayı geliştirilen yeni platforma Java Platform 2 adı verildi ama versiyon numarası 2 yapılmadı, 1.2 olarak devam etti. Bu bahar çıkacak Java 1.5, geçen 1.2, 1.3 ve 1.4 sürümlerinin ardından en çok gelişme ve değişikliği barındıran sürüm olacak.

Java nesneye yönelik bir dil olduğundan tüm yazılım sınıflar ve nesnelere üzerinden yürütülür. Sınıflar uygulamadaki nesnelere tanımlandığı kod parçalarıdır. Java’da her bir sınıf bir dosya içerisinde yer alır. Dosyaların uzantıları .java şeklindedir. Dosya adı ise içinde tanımlanan sınıf ile aynıdır. Örneğin, BenimSinif.java gibi.



Tablo- 3- Java Yazılım Geliştirme Aşaması

Bir java yazılımı şu şekilde geliştirilir:

- Programcı java kodunu yazar.
- Bu kod bir java derleyicisi ile derlenir. Sonuçta bytekod adı verilen bir tür makine kodu ortaya çıkar. Platform bağımsızlığını sağlayan şey bytekod'dur. Çünkü bir kere bytekod oluştuktan sonra yazılım tüm işletim sistemlerinde çalışabilir.
- Bu bytekod Java Virtual Machine(Java Sanal Makinesi) tarafından adım adım işletilir. Aşağıda java ve C++ kodunun geçirdiği aşamalar gösterilmiştir.

Derleyici kısaca herhangi bir editor ile yazılan java kaynak kodlarını(yani .java uzantili sınıfların yer aldığı dosyaları) java sanal makinesinin çalıştırabileceği bir tür makine dili(assembly) olan Bytecode'a dönüştürür. Bu dönüştürülen bytekod ise .class dosyaları içerisinde saklanır. Java kodunu derlemek için bir java derleyicisine ve java kütüphanelerine ihtiyacınız var. Şu anda iki derleyici yaygın olarak kullanılmaktadır. Bir tanesi Sun'ın SDK'si ile birlikte gelen javac. diğeri ise IBM'in açık kodlu derleyicisi jikes. Benim yaygın olarak kullandığım Jikes, çok hızlı derlemesi ile ünlü olsa da en son java yazılımlarını derlemek için javac'e ihtiyaç duyulabilir. Ayrıca linux altında derleme yapmak için gnu lisansı ile geliştirilen açık kodlu Gnuj kullanılabilir. Derleyici ve kütüphanelerin(Java API) bulunduğu uygulamaya "J2SE SDK" adı verilmiştir.

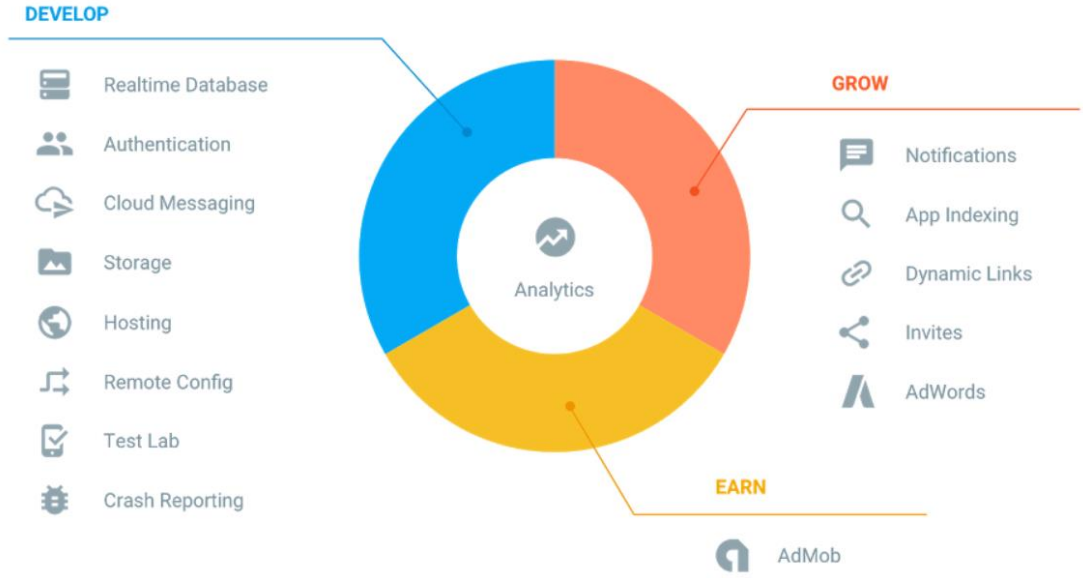
4.2 Firebase

JSON verilerimizi uzak bir sunucuda gerçek zamanlı olarak depolayabildiğimiz, bulut (Cloud) tabanlı bir platformdur.

Firebase servisinin geçmişi alsında James Tamplin ve Andrew Lee tarafından 2011 yılında kurulan Envolv şirketine dayanmakta. Geliştiricilerin web sayfalarına API aracılığıyla online chat özelliği kazandırabilmeleri amacıyla kurulan Envolv geliştiriciler tarafından planlanandan farklı bir şekilde (uygulama dataalarını gerçek zamanlı olarak kullanıcılarla senkronize bir şekilde paylaşılması gibi) kullanılır. Kurucular bu sebeble 2012 yılında bildiğimiz Firebase'in de temellerinin atıldığı servis ayrımı kararını verirler. 2012 yılından itibaren pek çok gelişmeyi barındıran, mobil ve web uygulama geliştirme platformu olarak konumlandırılan Firebase 2014 yılında Google tarafından satın alınır.

Google tarafından yeni özelliklerin eklenmesiyle gelişen Firebase, bütün bu ihtiyaçları karşılayabilme iddiasında ücretsiz kullanım da sunan bir platform. Herhangi bir platformda uygulama geliştirmeye bir sebeple başlayıp daha sonra bir kontrol paneli ve her durumda ulaşılabilir kullanıcı veri deposuna ihtiyacınız olduğunu hemen hissetmişsinizdir. Günümüzde uygulamalar platform farketmeksizin aynı veriye her cihazdan erişmek istiyor. Uygulaması, birçok kullanıcı tarafından yüklenen geliştiricilerin de kayıt - oturum bilgilerini tutma, uygulamaların kullanım verilerini analiz etme, yeni duyurular yapmak için aynı zamanda kullanıcıya bildirim gönderme, uygulamayı test etme gibi işlemleri rahatlıkla yönetebileceği bir yönetim paneli gerekiyor. İşte Google tarafından yeni özelliklerin eklenmesiyle sürekli kendini geliştiren Firebase, bütün bu ihtiyaçları karşılayabilmek için uygulama geliştiricilerine ücretsiz kullanım da sunan bir platformdur.

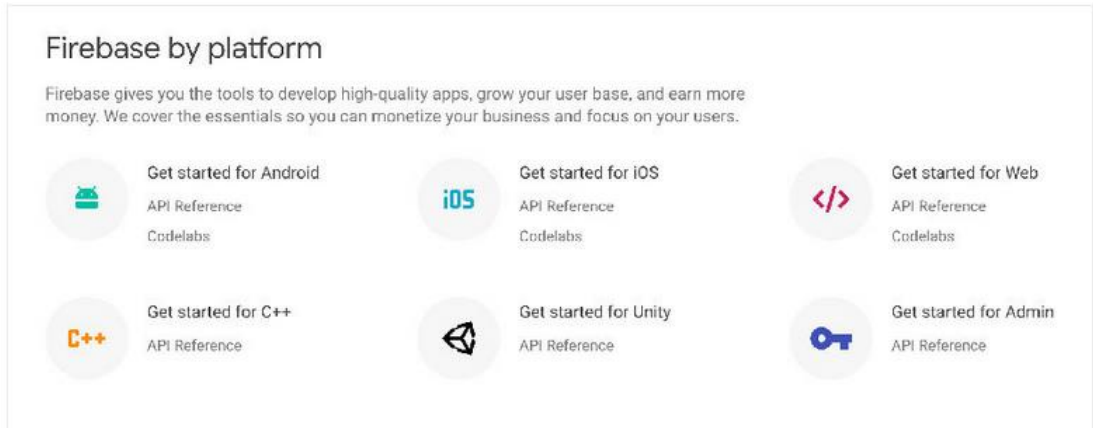
Satın alma sonrasında Google tarafından pek çok ek özellik eklenen servis küçük ve orta ölçekte uygulama geliştirme ihtiyaçlarının tamamını karşılama iddiasında. Bu amaçla ücretsiz kullanım (Spark Plan) imkanı da sunan (ek olarak) servis başlangıç aşaması itibariyle kontrol paneli üzerinden kullanıcı verisi deposuna, kayıt, oturum ve ekran görüntüleme gibi uygulama içi etkinliklerden bildirim iletimine ve testlere platform farkı olmaksızın erişim ve yönetim imkanı sunmakta. Ücretli planlar (Flame Plan ve Blaze Plan) elbette daha kapsamlı yönetim ve ek özellikler sunmakta. Ücretlendirme ve planlar dahilinde sunulan özelliklere Pricing sayfasından ulaşabilirsiniz.



Tablo- 4- Firebase Analitik

Uygulama yönetim, kullanım takip, veri depolama, bildirim gönderme gibi işlemleri, extra bir sunucuya ve sunucu tarafı kod yazmaya gerek kalmadan halleden Firebase, yeni geliştirici dostu arayüzünde Realtime Database, Notification, Remote Config gibi özelliklerle donatılmış her uygulama için ayrı ayrı ulaşım imkanı veriyor.

FCM - Firebase Cloud Messaging servisiyle de şimdiye kadar geliştiricilerin anlık bildirim - push notification göndermek için kullandığı GCM - Google Cloud Messaging' in yerini alacak olan Firebase, bildiğimiz tablolar ve sql yerine verileri root-child şeklinde organize edilen json veri parçacıklarında saklıyor.

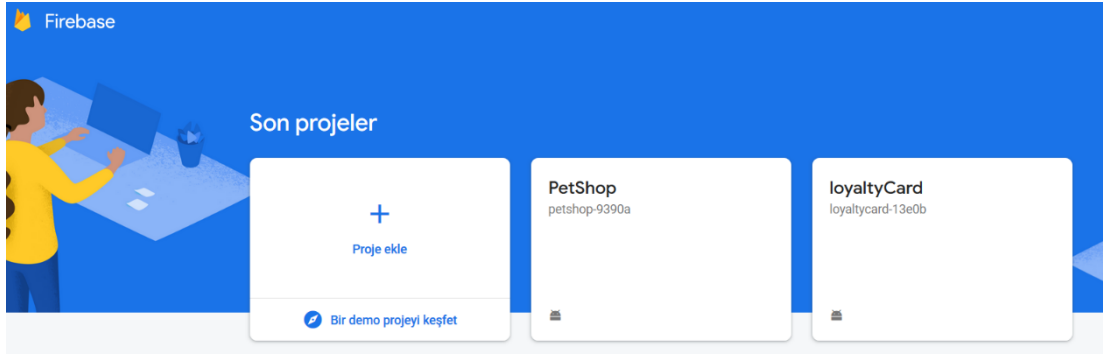


Şekil-13 - Firebase kullanılan Platformlar

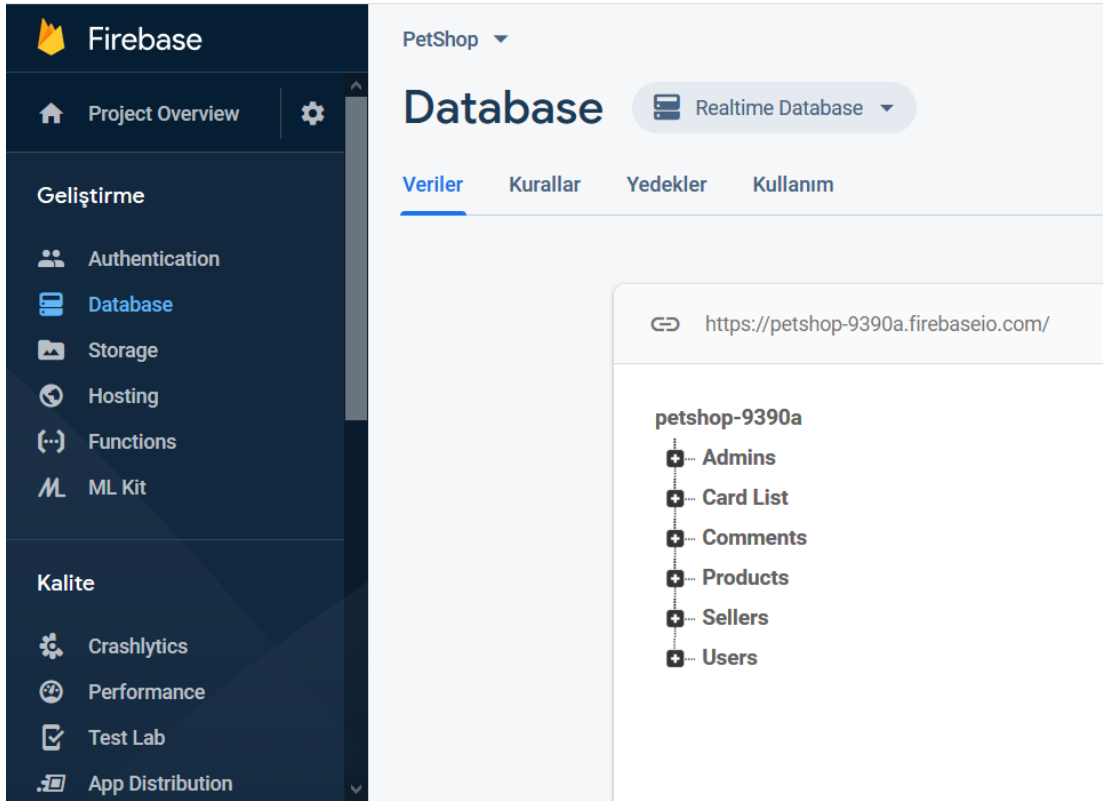
Öne çıkan birkaç özelliğe değinmek gerekirse;

- Uygulama yönetimi,
- Kullanıcı etkinlikleri,
- Veri depolama,
- Bildirim gönderimi

Firestore kullanımı için pek çok Google servisinde olduğu gibi bir Google kimliğine sahip olmanız gerekmektedir. Bu aşamada varsa sahip olduğunuz Google Mail (GMail) hesabınızdan faydalanabilir, Firestore web sayfası üzerinden giriş yaparak kullanıcı paneline ulaşabilirsiniz. Get Started butonunu tıklayarak ya da say üst bölümde yer alan Console bağlantısını tıklayarak ilerlenebilir. Artık yeni bir proje oluşturabiliriz. Bir proje uygulama için platformlar arası bir kapsayıcıdır. Bu sayede Android, iOS ve web uygulamaları arasında Veritabanı, Kullanıcı Yönetimi ve Uzak Yapılandırma gibi özellikler paylaşılabilir hale gelmektedir. Proje Adı, Proje Kimliği, Konumlar ve ilgili diğer alanları da uygun şekilde doldurduktan sonra Proje Oluştur butonu ile proje oluşturma sürecini de tamamlamış olmaktadır.

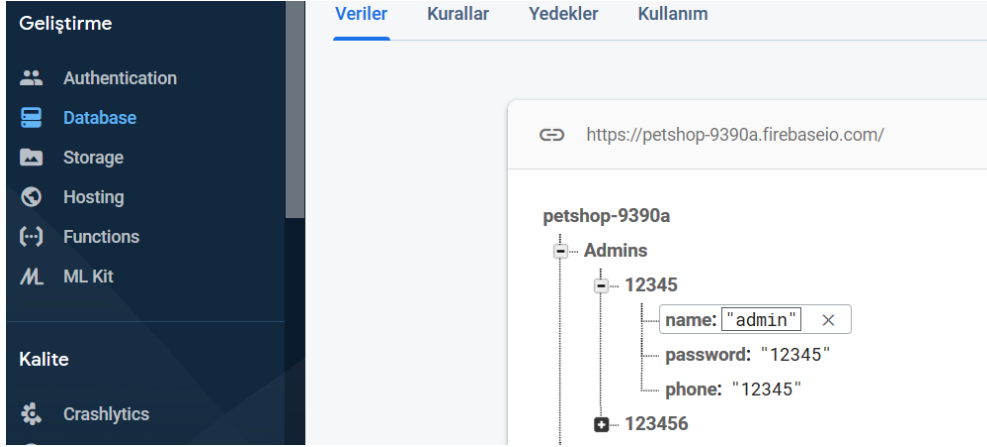


Şekil -14- Firestore Projeler

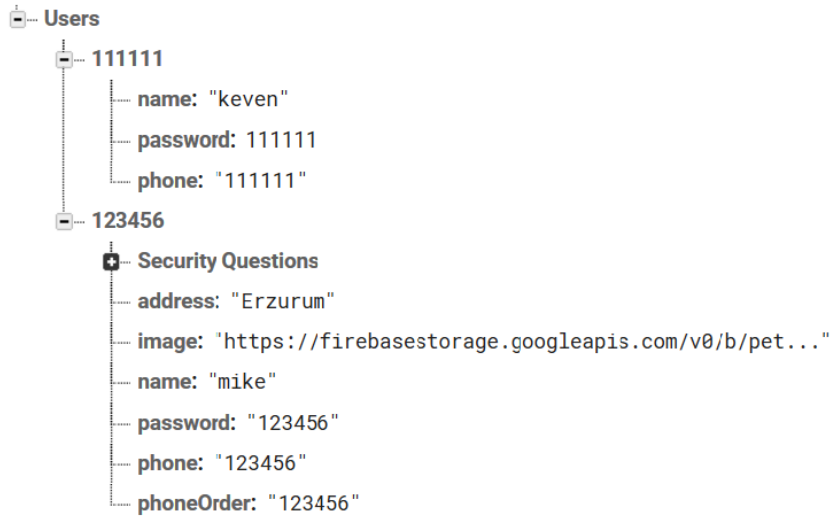


Şekil -15- Firestore Json Veri

Bu aşamadan sonra projenin iOS, Andorid ve/ya web uygulamasına entegre edilmesi gerekmektedir. İlgili yönergelere proje sayfasından ulaşılabilir. iOS entegrasyonu için iOS paket kimliği zorunlu olarak gerekmektedir. Ayrıca işteğe bağlı olarak uygulama takma adı ve App Store kimliğini de tanımlayabilirsiniz. Android uygulama entegrasyonu için ise Android paket adı zorunlu alan olarak gereklidir. İsteğe başlı olarak uygulama takma adı ve Dynamic Links,Invites ve Auth'ta Google ile oturum açma veya telefon desteği için SHA-1 karması gerekmektedir. Web uygulaması entegrasyonu için ise ilgili snippet'in web uygulamasına eklenmesi gerekmektedir. Konuyla ilgili olarak Google Analytics mobil uygulama mülkü yazımı inceleyebilirsiniz. İlgili alanların doldurulması sonrasında yapılandırma dosyasının indirilmesi, Firebase SDK (Software Development Kit)'sının eklenmesi, yükleme işleminin doğrulanması süreçleri gelmekte. Android için Firebase SDK ve diğer detayları Add Firebase to Your Android Project, iOS için ise Add Firebase to your iOS Project sayfaları incelenebilir. Ayrıca Flutter, Web ve Games kullanılacak diğer seçenekler.



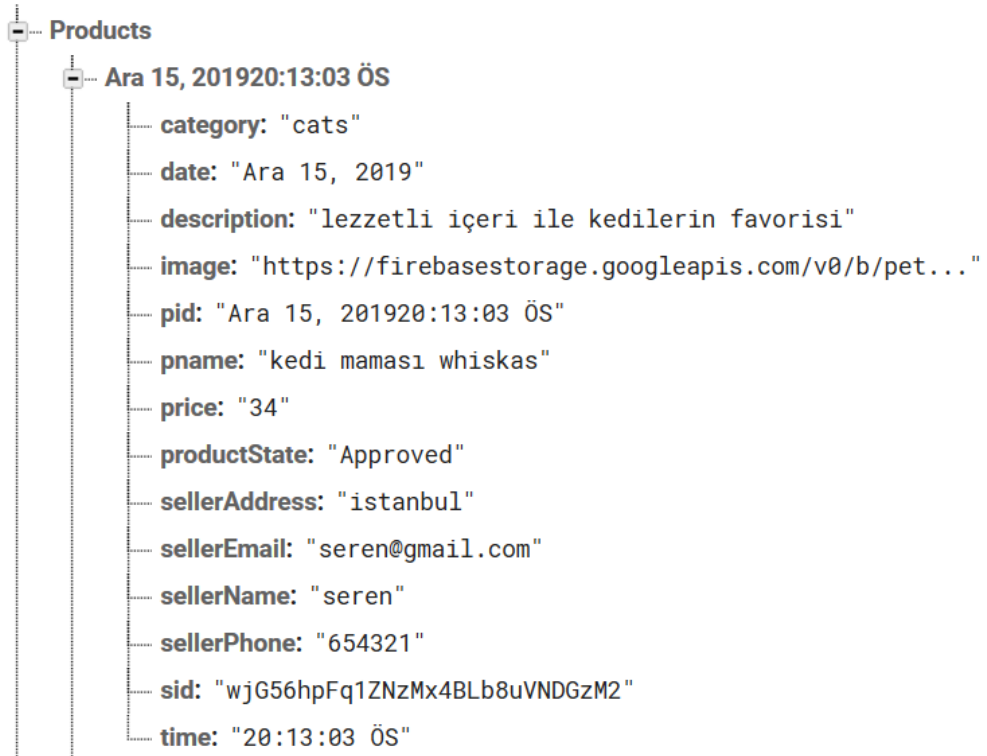
Şekil -16- Firebase Json Admin



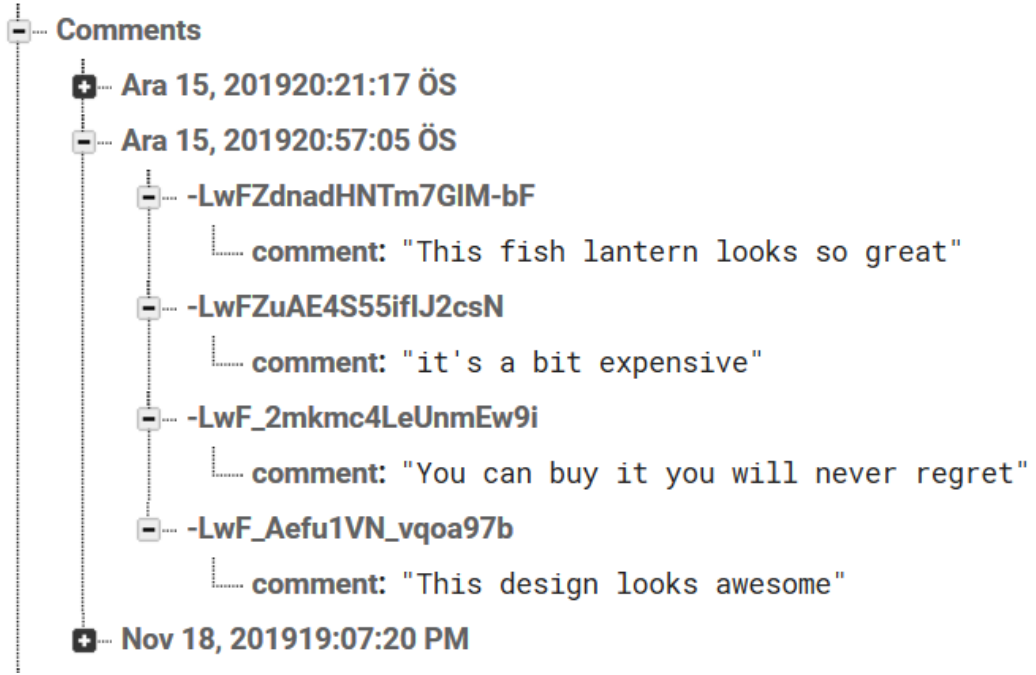
Şekil -17- Firebase Json User



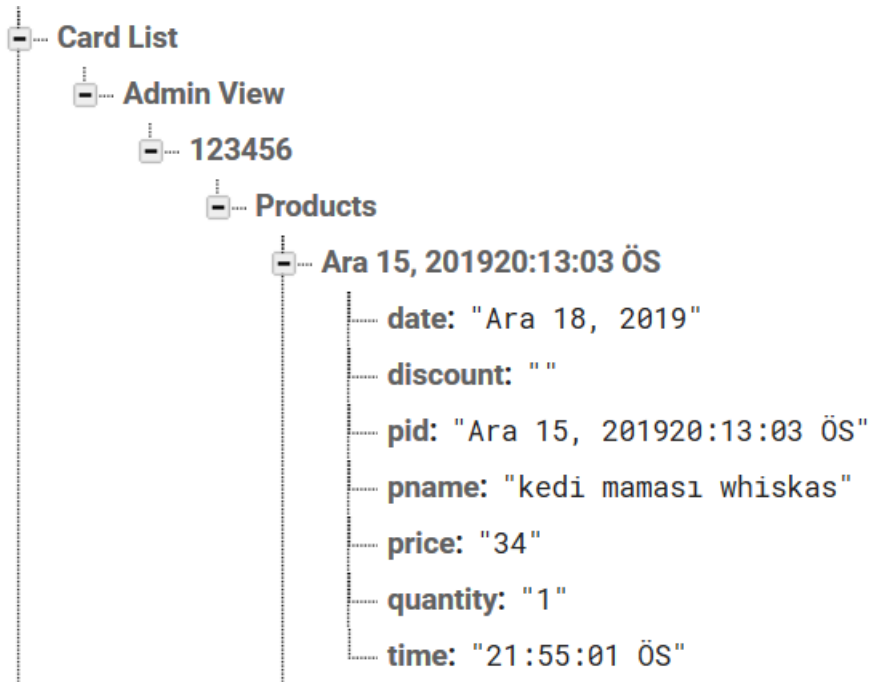
Şekil -18- Firebase Json Seller



Şekil -19- Firebase Json Products



Şekil -20- Firebase Json Comments



Şekil -21- Firebase Json CartList

Authentication ile e-posta/şifre, telefon, Google, Play Oyunlar, Facebook, Twitter, GitHub, Anonim oturum açma sağlayıcılarıyla yetkilendirilen alanlar için kullanıcılar sunucu tarafı kodu olmadan doğrulanabilmekte ve yönetilebilmekte.

Database, güçlü sorgular ve otomatik ölçeklendirme ile eşzamanlı veritabanı kullanımını mümkün kılmakta. Yeni veritabanı oluşturmak istediğinizde kilitli mod ve test modu olarak 2 seçenekle karşılaşabilirsiniz. Kilitli modda başladığınızda üçüncü taraf tarafından gerçekleştirilecek tüm okuma ve yazma işlemleri reddedilir ve veritabanı gizli tutulur. Test modunda ise veritabanı referansına sahip herkes tarafından okuma ve yazma işlemleri gerçekleştirilebilir.

Storage ile kullanıcıların oluşturduğu resim, ses ve video gibi içerikler sunucu tarafı kodu olmadan depolanır ve erişilir. Temel kurulumda varsayılan olarak tüm okuma ve yazma işlemlerine izin verilir. Güvenlik amacıyla veri yapısı tamamlandıktan sonra özel verilerin güvenliğini sağlamak için yeni kurallar yazabilirsiniz.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'me.zhanghai.android.materialratingbar:library:1.2.0'
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.rengwuxian.materialedittext:library:2.1.4'
    implementation 'com.github.d-max:spots-dialog:0.7@aar'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.google.firebase:firebase-analytics:16.0.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.google.firebase:firebase-storage:16.0.1'
    implementation 'com.firebaseui:firebase-ui-database:4.1.0'
    implementation 'com.theartofdev.edmodo:android-image-cropper:2.7.+'
```

```
implementation 'com.squareup.picasso:picasso:2.71828'
```

```
implementation 'io.paperdb:paperdb:2.6'
```

```
implementation 'de.hdodenhof:circleimageview:3.0.1'
```

```
implementation 'com.github.rey5137:material:1.2.5'
```

```
implementation 'com.cepheuen.elegant-number-button:lib:1.0.2'
```

```
implementation 'com.google.firebase:firebase-auth:16.0.1'
```

```
implementation 'com.braintreepayments:card-form:3.1.1'
```

```
implementation 'com.android.support:support-v4:28.0.0'
```

```
testImplementation 'junit:junit:4.12'
```

```
androidTestImplementation 'com.android.support.test:runner:1.0.2'
```

```
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

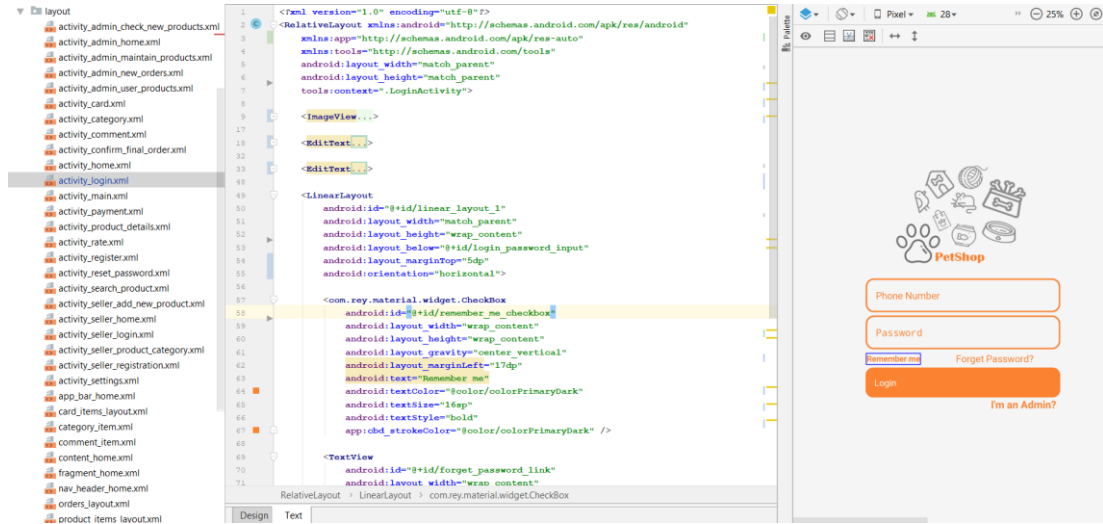
```
implementation 'com.android.support:recyclerview-v7:28.0.0'
```

```
apply plugin: 'com.google.gms.google-services'
```

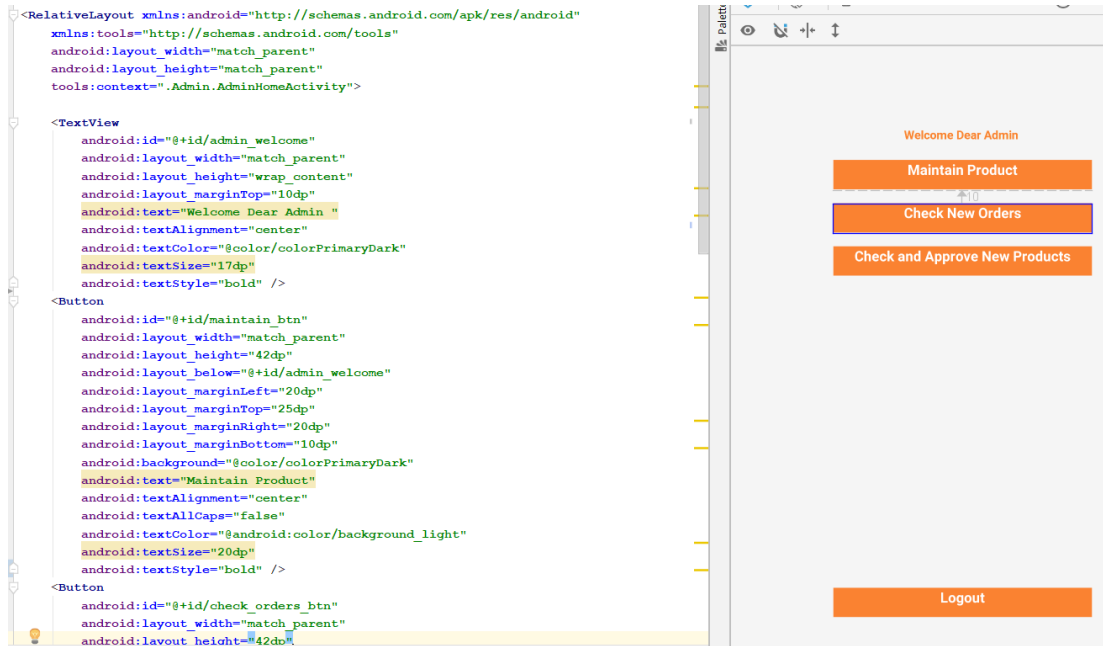
Şekil -22- Kullanılan Kütüphaneler

4.3 Xml

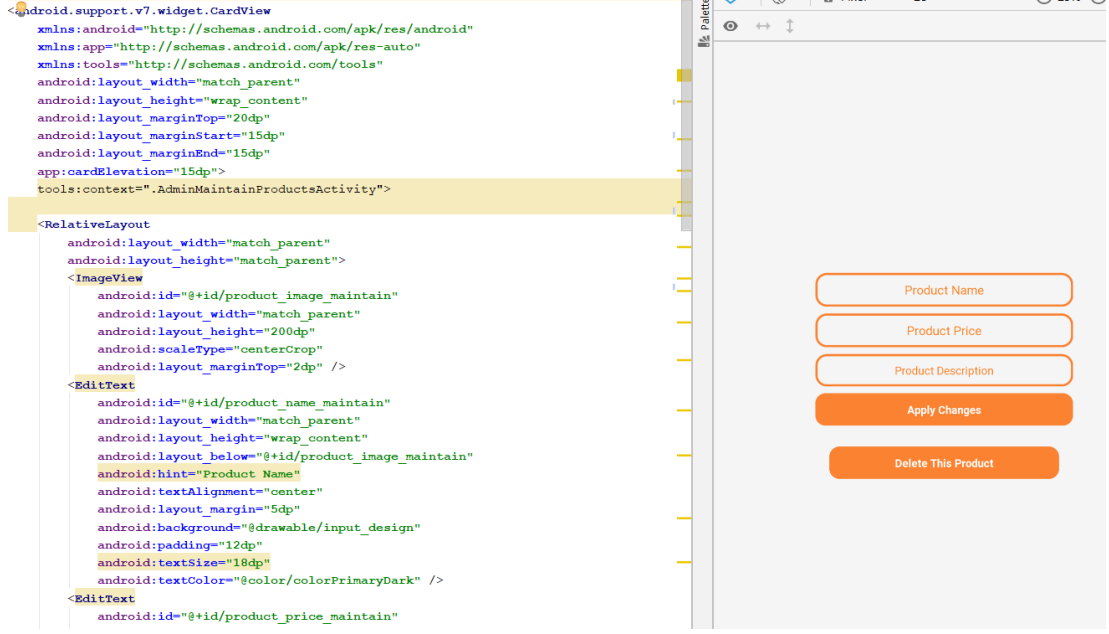
XML (Extensible Markup Language - Geniştirilebilir İşaretleme Dili), hem insanlar hem de makineler tarafından kolayca okunabilen ve İnternet üzerinde bilgi paylaşmak için çok sık kullanılan bir işaretleme dili formatıdır. Haber siteleri ya da bloglar gibi sıkça güncellenen içeriğe sahip web siteleri genellikle bir XML kaynağı sağlar ve böylece harici yazılımlar buralardaki içerik değişikliklerini bir arada görebilir. XML gönderme ve XML ayrıştırma (parsing) işlemleri, ağ bağlantılı uygulamalarda sıkça yapılan bir işittir.



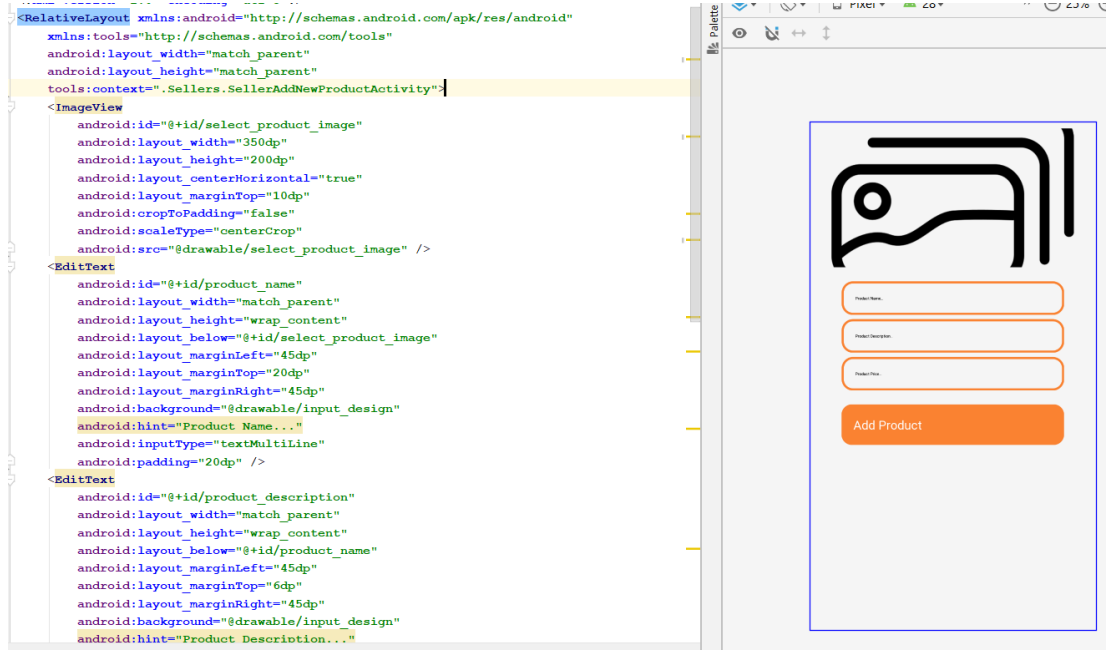
Şekil -23- Login Xml



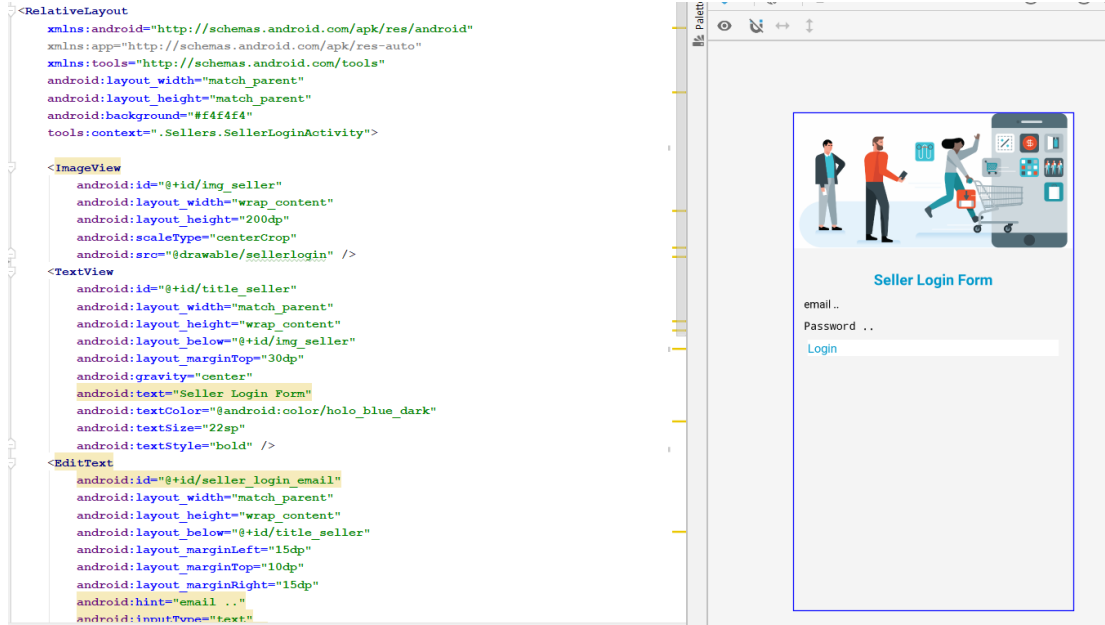
Şekil -24- Admin Arayüz Xml



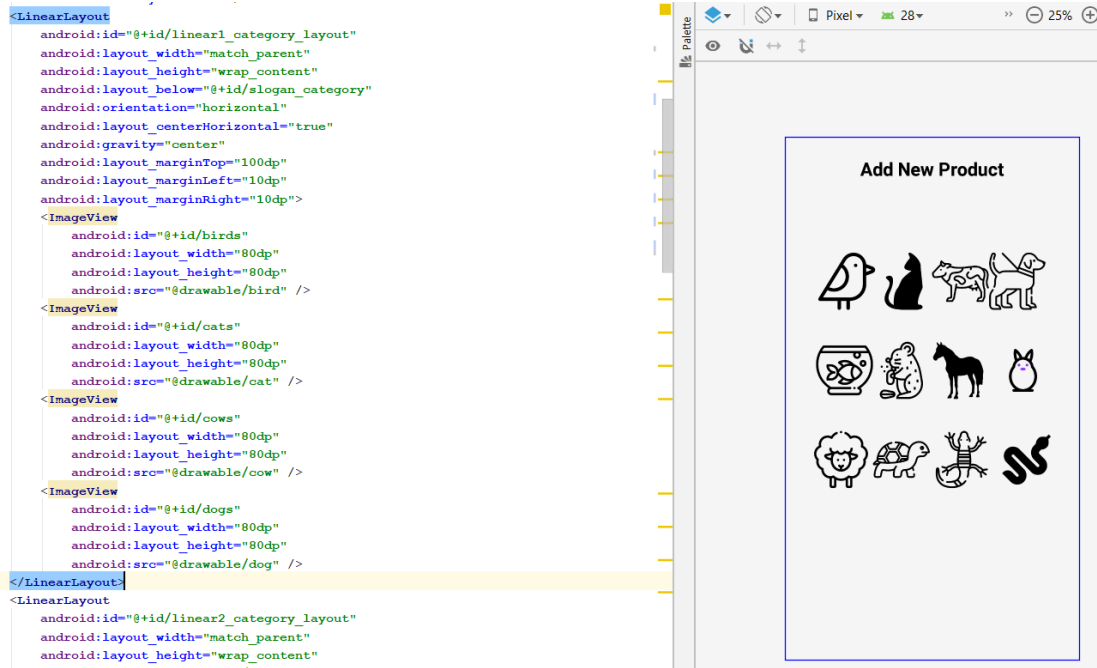
Şekil -25- Admin Ürün Düzünleme Xml



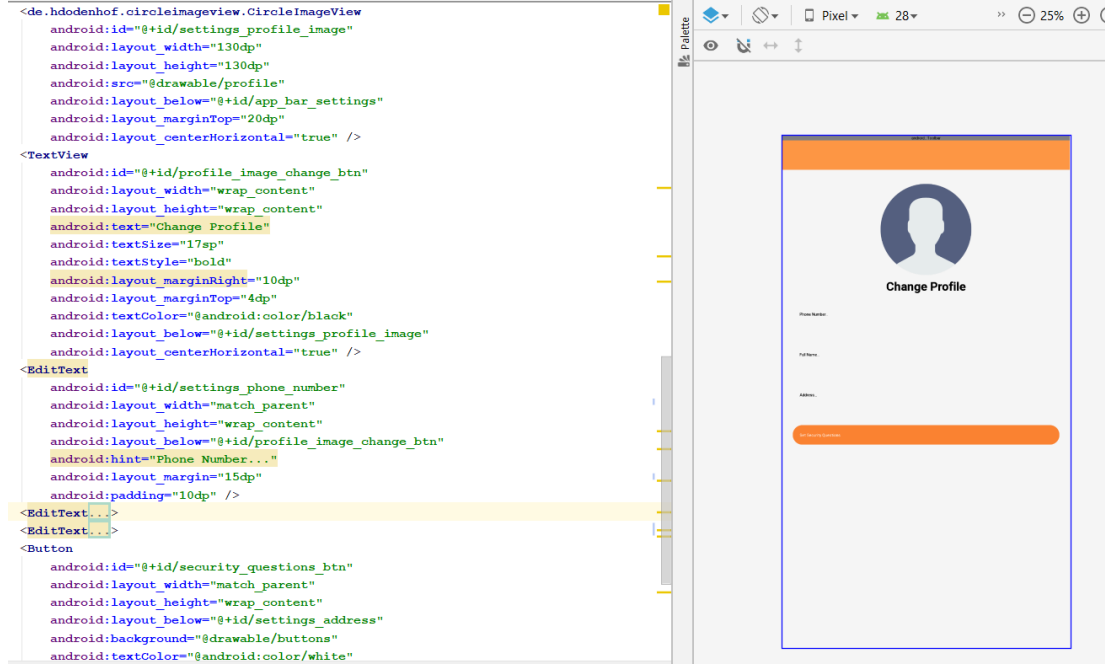
Şekil -26- Seller Yeni Ürün Ekle Xml



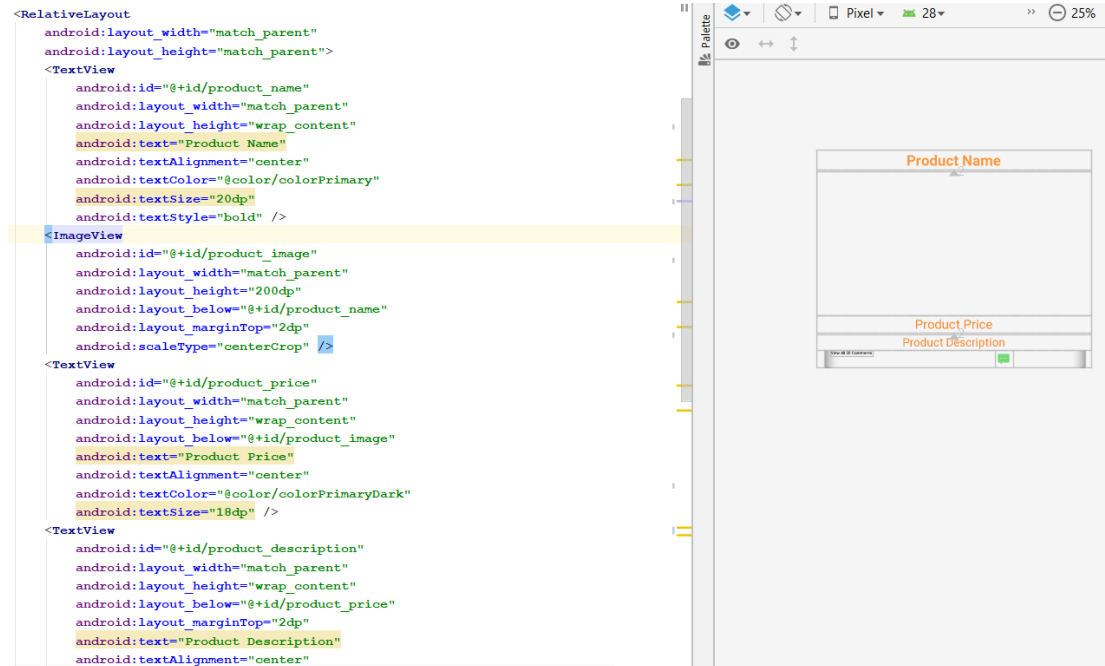
Şekil -27- Seller Login Xml



Şekil -28- Seller Ürün Kategori Xml



Şekil -29- Müşteri Profil Xml



Şekil -30- Müşteri Home Xml

4.4 JSON

JSON (JavaScript Object Notation – JavaScript Nesne Notasyonu) insanlar için okunabilir olan bilgi saklama ve alışveriř formatıdır. Bir JSON dosyası sadece metin kapsar ve .json uzantısını kullanır.

JSON yapılı veri depolayan ve genellikle bir sunucu ve istemci arasında veri alışveriři için kullanılan bir formattır. Bir JSON dosyası benzer fonksiyonlara sahip olan XML'e (ing) daha basit ve hafif bir alternatiftir. Bu formatlar depolanan verinin eşzamansız yüklenebilmesi için uyumlu bir şekilde çalışırlar, yani bir web site sayfayı yenilemeden bilgilerini güncelleyebilir. Bu işlemi XML/RSS'e kıyasla JSON ile yapmak daha kolaydır. Bugünlerde birçok web sitesi AJAX'ı benimsediğinden .json dosyası oldukça popüler hale gelmiştir.

Bunlara ek olarak, JSON kullanıcıların JSONP adlı bir yöntemle <script> etiketlerini uygulayarak farklı bir alan adından veri istemelerine izin verir. Aksi takdirde alan adları arasında aynı kök politikası nedeniyle veri transfer edemezsiniz.

Bir .json dosyası oluşturmak için doğru sözdizimini takip etmeniz gerekir. Bir JSON nesnesinin iki adet çekirdek elementi mevcuttur: Anahtarlar ve Değerler. Anahtarlar diziler olmalıdır. Tırnak işaretleriyle çevirili karakterler dizisinden oluşurlar.

Değerler geçerli bir JSON veri tipidir. Dizin, nesne, dizi, boole değeri, rakam veya boşluk şeklinde olabilirler. Bir JSON nesnesi {} küme parantezleriyle başlar ve biter. İçinde iki veya daha fazla anahtar/değer çiftlerine sahip olabilir ve bu ikisini bir virgül ile ayırır. Bu sırada, her bir anahtardan sonra değerden ayırtırmak için iki nokta üst üste gelir. İşte bir örnek: {"sehir":"Istanbul", "ulke":"Turkiye"} Burada iki adet anahtar/değer çiftimiz mevcuttur: şehir ve ülke anahtarlar; Istanbul ve Turkiye ise değerlerdir.

Dizin sıralı veri koleksiyonlarından oluşur. Köşeli parantezlerle başlar ve biterler ve içerideki her bir değer bir virgülle ayrılır. Bir dizin değeri JSON nesnelere içerebilir, yani aynı anahtar/değer çifti konseptini kullanır.

```
"ogrenciler":[ {"ad":"Ali", "soyad":"Demir"},  
 {"ad":"Ayse", "soyad":"Turan"},  
 {"ad":"Hasan", "soyad":"Yilmaz"}]
```

Köşeli parantezlerin aralarındaki bilgiler dizilerdir ve üç adet nesneye sahiptirler. Bir nesne bir anahtar ve değer içerir. Her anahtardan sonra bir iki nokta üst üste bulunmaktadır ve ayrıca her nesneyi de ayıran her değerden sonra ise bir virgül gelmektedir. İkisi de tırnak işaretleri içindedir. Bir değer olan nesne ise nesne ile aynı kuralı kullanmalıdır.

5. Bulgular

```

@Override
protected void onStart() {
    super.onStart();

    FirebaseRecyclerOptions<Products> options =
        new FirebaseRecyclerOptions.Builder<Products>()
            .setQuery(unverifiedProductsRef.orderByChild("productState").equalTo("Not Approved"), Products.class)
            .build();
    FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter =
        new FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
            @Override
            protected void onBindViewHolder(@NonNull ProductViewHolder holder, int position, @NonNull final Products model) {
                holder.txtProductName.setText(model.getName());
                holder.txtProductDescription.setText(model.getDescription());
                holder.txtProductPrice.setText("Price = " + model.getPrice() + "$");
                Picasso.get().load(model.getImage()).into(holder.imageView);

                holder.itemView.setOnClickListener((v) -> {
                    final String productID = model.getPid();

                    CharSequence options[] = new CharSequence[]{
                        "Yes",
                        "No"
                    };

                    AlertDialog.Builder builder = new AlertDialog.Builder(context: AdminCheckNewProductsActivity.this);
                    builder.setTitle("Do you want to approved this product? Are you sure?");
                    builder.setItems(options, (dialog, position) -> {
                        if (position == 0) {
                            ChangeProductState(productID);
                        }
                        if (position == 1) {
                        }
                    });
                    builder.show();
                });
            }
        };
}

```

Şekil -31- Adminin Seller'ın Koyduğu Ürünü Onaylaması

```

@Override
protected void onStart() {
    super.onStart();

    FirebaseRecyclerOptions<AdminOrders> options =
        new FirebaseRecyclerOptions.Builder<AdminOrders>()
            .setQuery(ordersRef, AdminOrders.class)
            .build();
    FirebaseRecyclerAdapter<AdminOrders, AdminOrdersViewHolder> adapter =
        new FirebaseRecyclerAdapter<>(options) {
            @Override
            protected void onBindViewHolder(@NonNull AdminOrdersViewHolder holder, final int position, @NonNull AdminOrders model) {
                holder.userName.setText("Name: " + model.getName());
                holder.userPhoneNumber.setText("Phone: " + model.getPhone());
                holder.userTotalPrice.setText("Total Amount: " + model.getTotalAmount());
                holder.userDateTime.setText("Order at: " + model.getDate() + " " + model.getTime());
                holder.userShippingAddress.setText("Shipping Address: " + model.getAddress() + ", " + model.getCity());

                holder.ShowOrdersBtn.setOnClickListener((view) -> {

                    String uid = getRef(position).getKey();//veritabanındaki orders kısmına erişebilmek için bunu kullandık

                    Intent intent = new Intent(context: AdminNewOrdersActivity.this, AdminUserProductsActivity.class);
                    intent.putExtra("name", uid);
                    startActivity(intent);
                });

                holder.itemView.setOnClickListener((v) -> {
                    CharSequence options[] = new CharSequence[]{
                        "Yes",
                        "No"
                    };

                    AlertDialog.Builder builder = new AlertDialog.Builder(context: AdminNewOrdersActivity.this);
                    builder.setTitle("Have you shipped this order products ?");

                    builder.setItems(options, (dialog, i) -> {

```

Şekil -32- AdmininMusterinin alıverişini onayladığı kısım

```

private void applyChanges() {
    String pName = name.getText().toString();
    String pPrice = price.getText().toString();
    String pDescription = description.getText().toString();

    if (pName.equals("")) {
        Toast.makeText(this, "Write down Product Name", Toast.LENGTH_SHORT).show();
    } else if (pPrice.equals("")) {
        Toast.makeText(this, "Write down Product Price", Toast.LENGTH_SHORT).show();
    } else if (pDescription.equals("")) {
        Toast.makeText(this, "Write down Product Description", Toast.LENGTH_SHORT).show();
    } else {
        HashMap<String, Object> productMap = new HashMap<>();
        productMap.put("pid", productID);
        productMap.put("description", pDescription);
        productMap.put("price", pPrice);
        productMap.put("pname", pName);

        productsRef.updateChildren(productMap).addOnCompleteListener((task) -> {
            if (task.isSuccessful()) {
                Toast.makeText(AdminMaintainProductsActivity.this, "Changes applied successful", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent( packageContext: AdminMaintainProductsActivity.this, SellerProductCategoryActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
}
}

```

Şekil -33- Adminin Ürünleri Düzenlediği Kısmıdır

```

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    activity: this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

frameLayout = findViewById(R.id.main_frame);
setFragment(new HomeFragment());

View headerView = navigationView.getHeaderView( index: 0);
TextView userNameTextView = headerView.findViewById(R.id.user_profile_name);
CircleImageView profileImageView = headerView.findViewById(R.id.user_profile_image);

if (!type.equals("Admin")) {
    userNameTextView.setText(Prevalent.currentOnlineUser.getName());
    Picasso.get().load(Prevalent.currentOnlineUser.getImage()).placeholder(R.drawable.profile).into(profileImageView);
}

recyclerView = findViewById(R.id.recycler_menu);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);

Bundle intentBundle=getIntent().getExtras();
if(intentBundle !=null ){
    String username=intentBundle.getString( key: "name");
    SharedPreferences.Editor editor=getSharedPreferences( name: "PREPS", MODE_PRIVATE).edit();
    editor.putString("name", username);
    editor.apply();
}

```

Şekil -34- Home Activity


```

private void LoginUser() {
    String phone = InputPhoneNumber.getText().toString();
    String password = InputPassword.getText().toString();
    if (TextUtils.isEmpty(phone)) {
        Toast.makeText(this, "Please write your phone number...", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(password)) {
        Toast.makeText(this, "Please write your password...", Toast.LENGTH_SHORT).show();
    } else {
        loadingBar.setTitle("Login Account");
        loadingBar.setMessage("Please wait, while we are checking the credentials.");
        loadingBar.setCanceledOnTouchOutside(false);
        loadingBar.show();
        AllowAccessToAccount(phone, password);
    }
}

private void AllowAccessToAccount(final String phone, final String password) {
    if (checkBoxRememberMe.isChecked()) // we use paper library for keep sign in the user
    { Paper.book().write(Prevalent.UserPhoneKey, phone);
      Paper.book().write(Prevalent.UserPasswordKey, password); }
    final DatabaseReference RootRef;
    RootRef = FirebaseDatabase.getInstance().getReference();

    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.child(parentDbName).child(phone).exists()) {
                Users usersData = dataSnapshot.child(parentDbName).child(phone).getValue(Users.class);

                if (usersData.getPhone().equals(phone)) {
                    if (usersData.getPassword().equals(password)) //userın girdiği password ile veritabanındaki aynı mı diye bakıyor
                    {
                        if (parentDbName.equals("Admins")) {
                            Toast.makeText(LoginActivity.this, "Welcome Admin, you are logged in Successfully...", Toast.LENGTH_SHORT).show();
                            loadingBar.dismiss();

                            Intent intent = new Intent( packageContext: LoginActivity.this, AdminHomeActivity.class);
                            startActivity(intent);
                        } else if (parentDbName.equals("Users")) {

```

Şekil -35- Müşteri Login Sayfası

```

private void StoreProductInformation() {

    loadingBar.setTitle("Add New Product");
    loadingBar.setMessage("Dear Seller, please wait while we are adding the new product.");
    loadingBar.setCanceledOnTouchOutside(false);
    loadingBar.show();
    Calendar calendar = Calendar.getInstance();
    SimpleDateFormat currentDate = new SimpleDateFormat( pattern: "MMM dd, yyyy");
    saveCurrentDate = currentDate.format(calendar.getTime());
    SimpleDateFormat currentTime = new SimpleDateFormat( pattern: "HH:mm:ss a");
    saveCurrentTime = currentTime.format(calendar.getTime());
    productRandomKey = saveCurrentDate + saveCurrentTime;
    final StorageReference filePath = ProductImagesRef.child(ImageUri.getLastPathSegment() + productRandomKey + ".jpg");
    final UploadTask uploadTask = filePath.putFile(ImageUri);
    uploadTask.addOnFailureListener((e) -> {
        String message = e.toString();
        Toast.makeText(SellerAddNewProductActivity.this, "Error: " + message, Toast.LENGTH_SHORT).show();
        loadingBar.dismiss();
    }).addOnSuccessListener((OnSuccessListener) (taskSnapshot) -> {
        Toast.makeText(SellerAddNewProductActivity.this, "Product image uploaded Successfully...", Toast.LENGTH_SHORT).show();
        Task<Uri> urlTask = uploadTask.continueWithTask((task) -> {
            if (!task.isSuccessful()) {
                throw task.getException();
            }
            downloadImageUrl = filePath.getDownloadUrl().toString(); //image url ini burada saklıyoruz
            return filePath.getDownloadUrl();
        }).addOnCompleteListener((task) -> {
            downloadImageUrl = task.getResult().toString();
            Toast.makeText(SellerAddNewProductActivity.this, "got the Product image Url Successfully...", Toast.LENGTH_SHORT).show();
            SaveProductInfoToDatabase();
        });
    });
}

private void SaveProductInfoToDatabase() {
    HashMap<String, Object> productMap = new HashMap<>();
    productMap.put("pid", productRandomKey);
    productMap.put("date", saveCurrentDate);
    productMap.put("time", saveCurrentTime);
}

```

Şekil -36- Seller'ın Yeni Ürün Eklediği Kısım

```

private void setAnswers() {
    String answer1 = question1.getText().toString().toLowerCase();
    String answer2 = question2.getText().toString().toLowerCase();
    if (question1.equals("") && question2.equals("")) {
        Toast.makeText(ResetPasswordActivity.this, "Please answer both questions", Toast.LENGTH_SHORT).show();
    } else {
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference().child("Users").child(Prevalent.currentOnlineUser.getPhone());
        HashMap<String, Object> userdataMap = new HashMap<>();
        userdataMap.put("answer1", answer1);
        userdataMap.put("answer2", answer2);
        ref.child("Security Questions").updateChildren(userdataMap).addOnCompleteListener((task) -> {
            if (task.isSuccessful()) {
                Toast.makeText(ResetPasswordActivity.this, "You have answer security questions successfully", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(ResetPasswordActivity.this, HomeActivity.class);
                startActivity(intent);
            }
        });
    }
}

private void displayPreviousAnswers() {
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference().child("Users").child(Prevalent.currentOnlineUser.getPhone());
    ref.child("Security Questions").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {
                String ans1 = dataSnapshot.child("answer1").getValue().toString();
                String ans2 = dataSnapshot.child("answer2").getValue().toString();
                question1.setText(ans1);
                question2.setText(ans2);
            }
        }
    });
}

```

Şekil -37- Kullanıcının Şifresini Restlediği Activity

```

private void updateOnlyUserInfo() {
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference().child("Users");
    HashMap<String, Object> userMap = new HashMap<>();
    userMap.put("name", fullNameEditText.getText().toString());
    userMap.put("address", addressEditText.getText().toString());
    userMap.put("phoneOrder", userPhoneEditText.getText().toString());
    ref.child(Prevalent.currentOnlineUser.getPhone()).updateChildren(userMap);
    startActivity(new Intent(ResetPasswordActivity.this, HomeActivity.class));
    Toast.makeText(ResetPasswordActivity.this, "Profile Info update successfully.", Toast.LENGTH_SHORT).show();
    finish();
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE && resultCode == RESULT_OK && data!=null) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        imageUri = result.getImageUri();
        profileImageView.setImageURI(imageUri);
    } else {
        Toast.makeText(this, "Error, Try Again.", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(ResetPasswordActivity.this, SettingsActivity.class));
        finish();
    }
}

private void userInfoSaved() {
    if (TextUtils.isEmpty(fullNameEditText.getText().toString())) {
        Toast.makeText(this, "Name is mandatory.", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(addressEditText.getText().toString())) {
        Toast.makeText(this, "Name is address.", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(userPhoneEditText.getText().toString())) {
        Toast.makeText(this, "Name is mandatory.", Toast.LENGTH_SHORT).show();
    } else if (checker.equals("clicked"))
}

```

Şekil -38- Settings Activity

6. Sonular ve neriler

Sonu olarak proje iin tasarladığım admin, user ve seller giriřlerini hatasız başarılı bir şekilde gerekleřtirdim. Bu giriřler tamamlandıktan sonra her bir kullanıcı türü iin uygulama sayfalarında istenen ve beklenen tasarımlara fonksiyonellik kazandırıldı.

Yazılım yapısal olarak veritabanına admin tarafından eklenen ürünlerin müşteri tarafında uygulamanın anasayfasında gözükmesidir. Buna ek olarak satıcının da aynı şekilde arayüz ile veritabanına ürün eklemesi ve eklediği ürün admin tarafından kabul edildikten sonra satıřa ıkabilmesi iin yazılımda veritabanına ürün ekleme ve eklenen ürünlerin uygulamada kullanıcılara gösterilmesi şeklinde olacağı iin bunu işlevsel olarak yapabilecek java kodları olacaktır.

Uygulamanın sınırlamaları řu şekilde yapılmıştır: Uygulama giriş kısmı 3 aşamalıdır müşteriler, satıcılar ve admin şeklinde müşterilerin, satıcı gibi ürün satma yetkileri olmayacak aynı şekilde satıcıların ise sadece ürün satma yetkileri olacak ürün satın alamayacaklar. Adminin ise satıcının satacağı ürünü onaylama yetkisi olacak ve tabi kendi tarafından satılacak ürünleri uygulamaya koyabilecek ve müşterinin spariş verdiği ürünlerin paketlenip teslimat aşamasının yapılmasını ve onaylanması işlerinin takibini yapacaktır.

Akıllı android işletim sistemine sahip her telefona play store den kolaylıkla indirilebilecek olan bu mobil uygulamayı evcil hayvan eşyaları satın alabilme potansiyeline sahip herkes kullanabilir. Uygulamanın kullanıcılardan bazı hususlarda izin alması gerekmektedir. Örneğin; uygulama kullanıcının galerisine girmek iin, konumunu görebilmek iin ve satıř işlemi iin kredi kartı bilgilerinin girilmesi gibi durumlarda kullanıcı izni gerekmektedir.

Müşterilerin satın alma işlemi iin sepette seçtikleri ürünler admin tarafından veritabanı bölümünde kargo ile teslimat aşamasına kadar kontrol edilmektedir. Aynı şekilde satıcılarında satmak istedikleri ürünler admin tarafından onaylandıktan sonra satıřa sunulmaktadır. Müşteri memnuniyeti yorumlar ve beğeniler de aynı şekilde admin tarafından veritabanında izlenmektedir.

Sisteme giriş; admin, satıcılar ve müşteriler sisteme giriş yapıp sistemi kullanabilirler. Bilgi güncelleme; bilgi güncellemelerini satıcılar ve admin yapabilmektedir. Müşteriler ise beğeni, yorum ve sepete ürün ekleme kısmında bilgi güncellemesi yapabilirler. Kontrol Sağlama; kontrol sağlama işlemini sadece admin yapabilmektedir.

Kaynaklar

- [1] Android Gelişimi Erişim Adresi: <https://blog.mustafaergec.com.tr/android-tarihi-ve-gelisimi.html> Erişim Tarihi: 06.02.2020
- [2] Akıllı Telefonların Hayatımızdaki Yeri Erişim Adresi: <http://ayhanturkel.blogspot.com/2015/01/akll-telefonlarn-hayatmzdaki-yeri-ve.html> Erişim Tarihi: 06.02.2020
- [3] Android'in Doğuşu Erişim Adresi: <http://www.dolubilgi.com/androidin-dogusu-ve-gelisimi/> Erişim Tarihi: 06.02.2020
- [4] Android Layout Kullanımı ve Firebase için Yapılan Konfigurasyonlar Erişim Adresi: <https://gelecegiyazanlar.turkcell.com.tr/blog/android-uygulamalarda-dogru-dialog-kullanimleri> Erişim Tarihi: 07.02.2020
- [5] Android Güncellemeleri Erişim Adresi: <https://shiftdelete.net/android-verileri-sonunda-guncellendi> Erişim Tarihi: 07.02.2020
- [6] Android Sürümleri Erişim Adresi: <https://www.yaparimben.com/android-surumleri/> Erişim Tarihi: 07.02.2020
- [7] Java Nedir Java'nın Tarihi Erişim Adresi: <https://www.bilgiustam.com/java-nedir-nasil-calisir/> Erişim Tarihi: 07.02.2020
- [8] Akıllı Telefonların Hayatımızdaki Yeri Erişim Adresi: <http://ayhanturkel.blogspot.com/2015/01/akll-telefonlarn-hayatmzdaki-yeri-ve.html> Erişim Tarihi: 07.02.2020
- [9] Json Nedir? Erişim Adresi <https://umiitkose.com/2015/03/json-nedir-ne-amacla-kullanilir/> Erişim Tarihi: 08.02.2020
- [10] Firebase Nedir ve Nasıl Kullanılır? Erişim Adresi <http://umutonur.com/firebase-nedir-nasil-kullanilir-ve-ozellikleri-nelerdir/> Erişim Tarihi: 08.02.2020
- [11] Ux Nedir? Erişim Adresi: <https://www.globya.com.tr/ux-nedir-kullanimi-ve-nekadar-onemli/> Erişim Tarihi: 09.02.2020

Özgeçmiş

Efnan AKKUŞ:

21 Ocak 1996 tarihinde Erzincan'ın Çayırılı ilçesinde dünyaya geldim. Ben henüz bir yaşındayken İstanbul'a taşınmışız. İlkokulu ve ortaokulu mahallemizde bulunan Fatma Gözen Eralp İlköğretim Okulu'nda tamamladım. O dönemde hayalimde Anadolu Lisesi kazanmak vardı ve bende sürekli hayallerim doğrultusunda çalıştım ve yaşadığım Pendik ilçesinde bulunan Gülizar Zeki Obdan Anadolu Lisesi'ni kazandım. Liseye geçtikten sonra ortaokul ve ilkokuldaki başarıyı sürdürmedim. Yeni bir okula alışma arkadaş edinme ve derslere alışma süreci derken notlarım çok istediğim gibi olmadı. Lise son sınıfa geldiğimde temelimin çok sağlam ve donanımlı olmadığını farkındaydım ve bir yandanda ara vermeden üniversiteye başlamak istiyordum. 2014 yılında Harran Üniversitesi Bilgisayar Mühendisliği Bölümü'ne yerleştim. Bir yıl hazırlık okuduktan sonra bölüme geçtim. Hazırlık benim için oldukça verimli bir dönemdi İngilizce alt yapımı sağlam bir şekilde oluşturdum. Benimle birlikte hazırlık okuyan diğer arkadaşlarımla Erasmus programına katılmak ve avrupayı gezmek gibi bir hayalimiz oldu. İkinci sınıfın ikinci dönemi Polonya'ya erasmus öğrencisi olarak gittik. Erasmus'tan döndükten sonra Meteoroloji Bölge Müdürlüğü'nde ilk stajımı gerçekleştirdim. Stajda hava durum rapounu gösteren bir uygulama yaptım. Üçüncü sınıfa başlarken okulumuzda re:coded isimli bir girişimin android bootcamp programına katıldım. Bootcamp 6 ay sürdü bu süreç boyunca 10'dan fazla proje geliştirdik ve son olarak gerçek bir müşteri projesi yaptık. Re:coded'ın İstanbul'da düzenlediği Social Innovation Hackathon'a katıldım bu benim ilk Hackaton deneyimimdi bana ve arkadaşlarıma çok şey kattığını düşünüyorum. Üçüncü sınıfın yaz tatilinde Mentornity isimli bir şirkette android stajyeri olarak 45 gün yaz stajı yaptım. Dördüncü sınıfa başladığımda re:ceded'in Judenhack Alman Türk Gençliği Programı kapsamında Berlin'e gittik ve orada bir hafta boyunca yazılım şirketlerini gezdik ve hackathona katıldık bizim için oldukça verimli bir hafta olduğunu söyleyebilirim. Dördüncü sınıf bittikten sonra yaz tatilinde hem yaz okulu hemde staj yapmam gerekiyordu. Sakarya'da yaz okulu yaptıktan sonra TOP Hair Clinic isimli şirkette geri kalan staj günlerimi tamamladım. Yüz yüze dersim kalmamıştı sadece bitirme projemi okula teslim etmem gerekiyordu. Bu dönem boyunca en son staj yaptığım şirkette part-time olarak çalışmaya devam ettim aynı zamanda da kendi bitirme projemi tamamdım. TOP Hair Clinic'te çalıştığım süre boyunca şirketin mobil uygulamasını yaptım. Re:coded'ın katılmış olduğum bootcamp'i ve kendi çabalarımla öğrenmiş olduğum android programlama şuan için benim çalışmak istediğim bir alan olmuştur. Bunun yanı sıra Machine Learning ve Deep Learning alanlarında kendimi geliştirmek istiyorum.