T.C.

BURSA ULUDAĞ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

EKRANLI CİHAZLAR ARASI GÖRÜNTÜ AKTARIMI

031611110 Mehmet Fatih Karabaş

031611079 Enes Gülmez

TASARIM I

BURSA 2019

T.C
BURSA ULUDAĞ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

Ekranlı Cihazlar Arası Görüntü Aktarımı

031611110 Mehmet Fatih Karabaş

031611079 Enes Gülmez

Projenin Danışmanı: Prof. Dr. Tuncay Ertaş

Bursa Uludağ Üniversitesi Mühendislik Fakültesi tez yazım kurallarına uygun olarak hazırladığım bu Mühendislik Tasarımı I çalışmasında;

- Tez içindeki bütün bilgi ve belgeleri, akademik kurallar çerçevesinde elde ettiğimizi
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları, bilimsel ahlak kurallarına uygun olarak sunduğumuzu
- Başkalarının eserlerinden yararlanılması durumunda, ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumuzu
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimizi
- Kullanılan verilerde herhangi bir tahrifat yapmadığımızı
- Bu tezin herhangi bir bölümünü üniversitemde veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımızı

beyan ederiz.

19.12.2019

Enes Gülmez

Mehmet Fatih Karabaş

Danışmanlığımda hazırlanan Mühendislik Tasarımı I çalışması, tarafımdan kontrol edilmiştir.

Prof. Dr. Tuncay Ertaş

ÖZET

Bu Mühendislik Tasarımı çalışmasında, ekranlı cihazlar arası görüntü aktarımı ihtiyacına uygun bir Android uygulaması yapılmıştır. Uygulamanın amacı uygulama ekranınındaki görüntünün bir başka ekranlı cihaza kablosuz olarak aktarılmasıdır. Uygulamanın geliştirilmesinde Android geliştirme platformu olan Android Studio IDE'si, Java, XML ve HTML programlama dilleri kullanılmıştır. Kullanılan yöntem cep telefonu ekranındaki görüntünün yakalanıp yerel ağ içerisinde yayınlanmasıdır. Uyguluma ekranındaki "Yayını Başlat" butonuna tıklanmasıyla uygulama yerel ağda bir sunucu açmaktadır. Sunucunun IP adresi buton üzerinde görüntülenecektir. Görüntülenen IP adresi görüntünün aktarılacağı cihazın tarayıcısında URL olarak kullanıldığında cep telefonu ekranındaki görüntü tarayıcı sekmesine kablosuz olarak aktarılmış olur.

ABSTRACT

In this Engineering Design study, an Android application has been made in accordance with the need for image transfer between screened devices. The purpose of the application is to wirelessly transfer the image on the application screen to another display device. Android development platform, Android Studio IDE, Java and HTML programming languages were used in the development of the application. The method used is to capture the image on the mobile phone screen and broadcast it on the local network. By clicking “Yayını Başlat ” button on the application screen, the application opens a server on the local network and start to broadcast the image on server. The IP address of the server will display on the button. When the displayed IP address is used as the URL in the browser of the device to which the image will be transferred, the image on the mobile phone screen is transferred wirelessly to the browser tab.

İÇİNDEKİLER

Sayfa No

ŞEKİLLER DİZİNİ

Sayfa No

ÇİZELGELER DİZİNİ

Sayfa No

1. HABERLEŞME

İletişim terimi bilgiyi elektriksel yollarla göndermeye, almaya, işlemeye karşılık gelir. İletişimin amacı, herhangi bir biçimdeki bilginin zaman ve uzay içinde kaynak olarak adlandırılan bir noktadan, kullanıcı denilen başka bir noktaya aktarılmasıdır. Günümüzde telefon, radyo, televizyon gibi elektriksel iletişimin çeşitleri, günlük yaşantımızın vazgeçilmez birer parçası olmuşlardır. Elektriksel iletişimin diğer bazı önemli örnekleri şu şekilde sıralanabilir; radar, telemetre dizgeleri, bilgisayarlar arası bilgi aktarımı, askeri amaçlar için kullanılan telsiz. Bu liste istenildiği kadar genişletilebilir. Elektronik devre öğeleri teknolojisindeki yeni ilerlemelere bağlı olarak önümüzdeki yıllarda iletişim dizgelerinde de önemli gelişmelerin olması kaçınılmazdır.

İletişim tarhini 1840′ larda telgrafın bulunmasına dayanmaktadır; birkaç “10 yıl” sonra telefonla ve bu yüzyılın başında da radyo ile gelişmiştir. Elektronik tüplerin bulunuşu ile doğan radyo iletişimi, büyük ölçüde 2. Dünya savaşı sırasındaki çalışmaların yoğunluğu ile hızla gelişmiştir.

Transistör, entegre devre ve diğer yarı iletken araçların bulunup, kullanılmasıyla radyo ve TV geliştirildi ve yaygın bir şekilde kullanılmaya başlandı. Son zamanlarda uydular ve fiber optik, bilgisayarlara ve diğer veri iletişimlerine artan bir önem yükledi ve iletişim daha yaygın bir duruma gelmiştir

1.1. Haberleşme İşlemi

Modern bir iletişim sistemi, bilgi göndermeden önce onun sıraya koyulmasıyla, işlenmesiyle ve korunmasıyla ilgilenir. Gerçek anlamda gönderme daha fazla işleme ve gürültünün süzülmesiyle gerçekleşir. Son olarak, kod çözme, mesajı koruma ve bilgi algılama basamaklarından oluşan alma işlemi gelir. Bu konuda iletişim şekilleri radyo-telefon, telgraf, noktadan noktaya yayın ve hareketli iletişim, bilgisayar iletişimi, radar, radyo-telemetre ve radyo ile yardım isteme metotları olarak karşımıza çıkar.

Bir haberleşme sistemi beş temel bloktan oluşur. Bloklar Şekil 1.1 içerisinde blok diyagramı halinde gösterilmiştir. Bu bloklar;

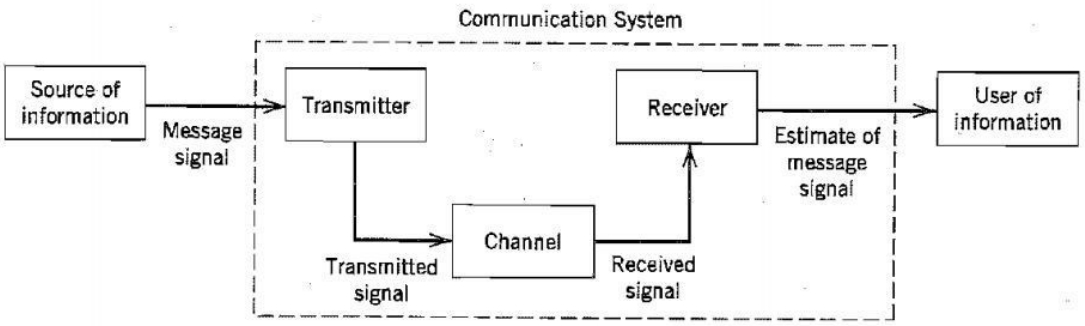
I. Bilginin Kaynağı

II. Verici

III. Kanal

IV. Alıcı

V. Bilginin Alıcısıdır.



Şekil 1. 1 Haberleşme Sistemi Elementleri (1)

Vericinin amacı mesaj kaynağı tarafından üretilmiş mesaj sinyalini kanal üzerinden iletişime uygun forma dönüştürmektir. Kanal üzerinde sinyal bozulmaya uğrayabilir. Bu bozulma gürültü ve diğer sinyal kaynaklarının kanal üzerinde girişimlerinden kaynaklanır. Kanal çıkışındaki sinyal iletilecek sinyalin bozulmuş bir versiyonudur.

Alıcının amacı kanal çıkışındaki bozulmuş mesajı yeniden yapılandırarak bilginin alıcısında tanınabilir bir mesaj haline getirmektir.

Haberleşmenin 2 basit şekli vardır. Bunlar;

I. Yayın: Tek güçlü vericiden birden fazla alıcıya tek yönlü olarak yapılır. Televizyon ve radyo yayın tipi haberleşme için yaygın kullanılan 2 örnektir.

II. Noktadan - Noktaya: Tek verici ve tek alıcı arasındaki kanal ile iletişim sağlanır. Çoğunlukla çift yönlü iletişimde kullanılır. Noktadan - noktaya haberleşme öreneği içinse telefon rahatlıkla verilebilir.

1.2. Bilginin Kaynağı

Haberleşme açısından 4 önemli bilgi kaynağı tipi iletişim sektörünü domine etmektedir. Bunlar;

I. Konuşma

II. Müzik

III. Resim

IV. Bilgisayar verileridir.

Bilgiyi taşıyan sinyaldir. Sinyali her bağımsız değişken için farklı rolü olan zamanın tek değerli bir fonksiyonu olarak tanımlanır. Sinyal konuşma, müzik ve bilgisayar verileri için tek boyutlu olabileceği gibi resim için iki boyutlu, video için üç boyutlu veya farklı veri tipleri için üçten fazla boyuta sahip olması gerekebilir.

Resimler, insanların görsel algı sistemine dayanır. Bir resim televizyondaki gibi dinamik yada fotokopi kağıdındaki gibi statik olabilir. İlk olarak dinamik resimler ele alınırsa; resimlerdeki hareket elektriksel sinyallere dönüştürülerek resimin vericiden alıcıya taşınması kolaylaştırılır. Böylelikle her resim sırayla taranmış olur. Bu tarama işlemi TV kamerasında yürütülür. Kamera foto katot içeren çok sayıda ışığa duyarlı elementin resime

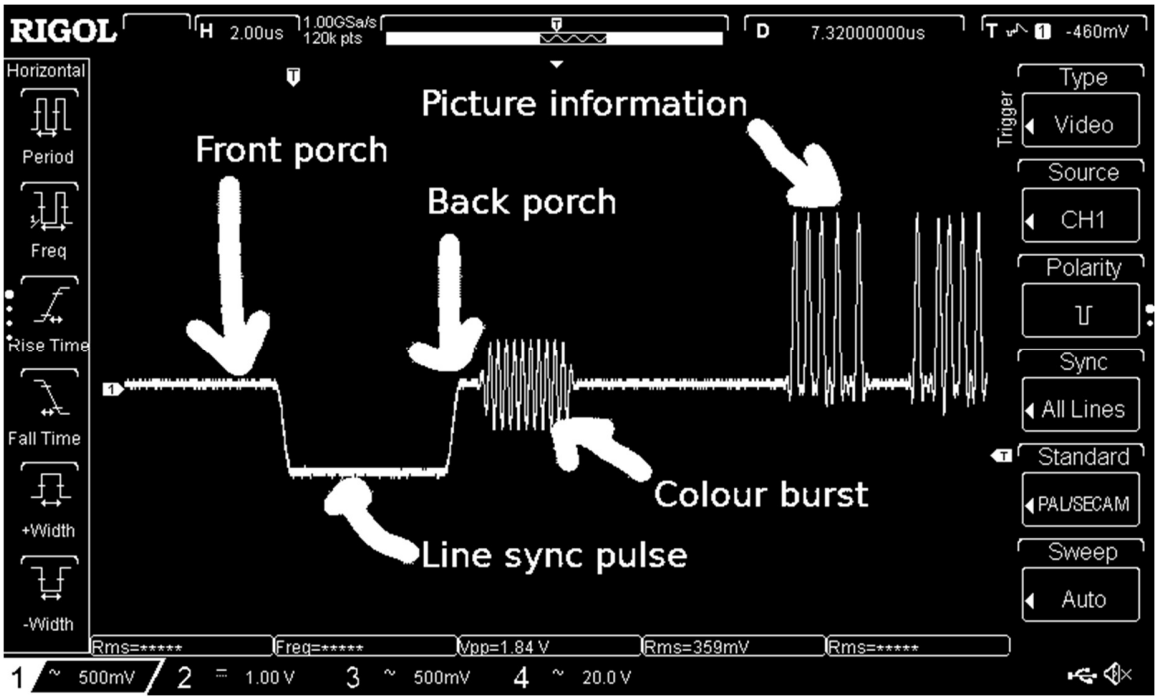
odaklanması için optik olarak tasarlanmıştır. Işığa duyarlı yüzeyde oluşan şarj deseni elektron ışını ile taranır, böylelikle orjinal resimin parlaklığı mekansal olarak değiştikçe sağlanan çıkış akımı geçici olarak değişir. Ortaya çıkan çıkış akımı video sinyalidir.

Televizyonda kullanılan tarama şekli uzaysal örneklemenin "Raster" taraması adı verilen bir formudur. Raster taraması iki boyutlu resim yoğunluğunu tek boyutlu dalga formuna dönüştürür.

Renkli ekranlardaki renk algısı insan gözündeki üç farklı renk reseptörü ile gerçekleşir. Bunlar kırmızı, yeşil ve mavidir. Dalga boyları sırası ile 570 nm, 535 nm ve 445 nm dir. Bu üç renk birincil renkler olarak adlandırılır çünkü doğada bulunan diğer tüm renkler bu üç rengin karışımı ile elde edilebilir. Bu üç renk video sinyalinde $m_R(t)$, $m_G(t)$, $m_B(t)$ sembolleri ile ifade edilir. Tüm renkler bu değişkenin lineer kombinasyonları ile elde edilir.

- Parlaklık sinyali $m_L(t)$, görüntü sinyali monokrom bir televizyonun alıcısına ulaştığında görüntünün siyah-beyaz bir versiyonun oluşturulmasını sağlar.
- $m_Q(t)$, $m_I(t)$ ile sembolize edilen iki sinyal türü ise renklilik sinyalleri olarak adlandırılır. Renklilik sinyalleri resimlerin renginin gri tonlarından ayrılma şeklini ifade eder.

Parlaklık sinyali $m_L(t)$, 4.2 MHz bant genişliğinin tümüne atanmıştır. Görüntü aktarımı testleri sırasında İnsan görüşünün belirli özellikleri nedeniyle $m_Q(t)$, $m_I(t)$ renklilik sinyallerin nominal bant genişliklerinin renklerin tatmin edici biçimde tekrar üretimi için sırasıyla 1.6 MHz ve 0.6 MHz olması gerektiği belirlenmiştir. Şekil 1.2'de görüntü sinyalinin osiloskop görüntüsün bir örneği verilmiştir.



Şekil 1. 2 Video sinyalinin osiloskopta örnek görüntüsü.

https://hackaday.com/2018/01/18/know-your-video-waveform/

1.3. Görüntü Verisinin Sıkıştırılması

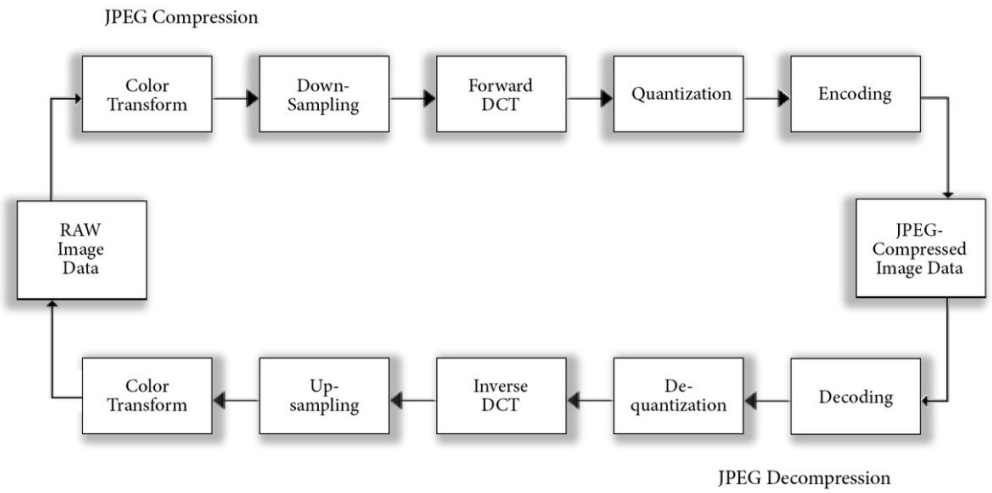
Video uygulamaları için gereken saklama alanı önemlidir. Standart sıkıştırma algoritmaları farklı üretici firmalar için birlikte çalışılabilirliğe olanak sağlamaktadır.

- JPEG resim kodlama standardı siyah-beyaz yada renkli resimleri sıkıştırmak için tasarlanmıştır. Sıkıştırma biçimi insan görsel sisteminin bilinen sınırlamalarına dayanmaktadır. JPEG yada Joint Photographic Experts Group kodlama şekli kodlayıcının girişinde fotoğrafın elementlerini yada diğer adı ile piksellerini 8 x 8 bloklar haline getirir. Bu bloklar ayrık cosinüs transformu olarak da bilinen (DCT) Fourier dönüşümüne uygulanır. (Şekil 1.3) Dönüşüm her bloktaki pikselleri 64 katsayıya ayrıştırır. Bu katsayılar iki özelliğe sahip olmalıdır.

1. Katsayılar mümkün olduğunca ilintisiz olmalıdır.

2.Giriş sinyalinin enerjisi mümkün olan en az sayıda katsayı paketini barındırmalıdır.

Kodlayıcı girişindeki bir sonraki işlem kuantalamadır. Kuantalama işlemi 64 DCT katsayısını kapsar. JPEG kuantalaması kullanıcı tarafından kodlayıcıya sağlanan kuantalama tablosu kullanılarak yapılır. Kuantalama tablosunun her elementi DCT katsayılarının basamak genişliğini ifade eden 1 ile 255 arasındaki tam sayılardan oluşur.



Şekil 1. 3 JPEG kodlaması blok diyagramı.

(https://beyondresolution.info/jpeg-the-guy-behınd-the-guy-behınd-the-guy)

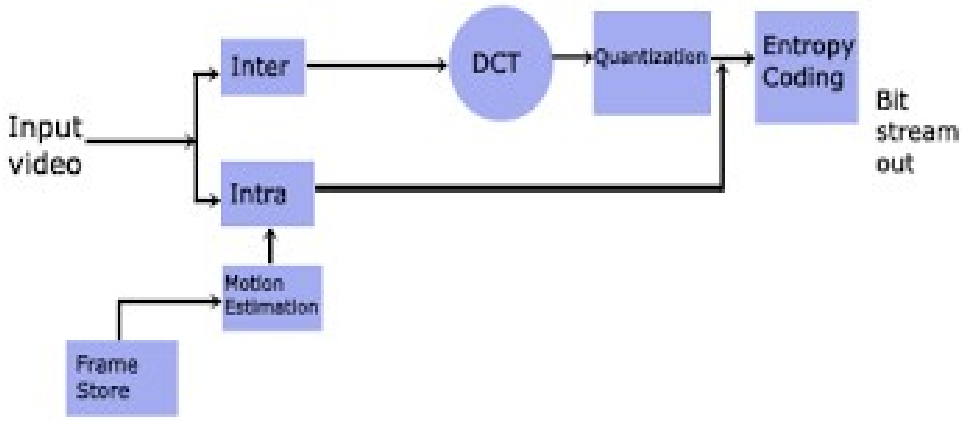
Böylelikle her DCT katsayısı 8-bit kod uzunluğu ile tesil edilmiş olur. Sonuç olarak kuantalama işleminin amacı algısal olarak farkedilemez olan bilgiyi atmaktır. Kuantalama birebir haritalamadır.

Kodlayıcıdaki son işlem Huffman kodlamasıdır. Huffman kodlaması bir entropik kaynak kodlamasıdır. Huffman kodlaması kodlanmış DCT katsayılarının istatistiksel karakteristiklerininden hareketle ilave bilginin sıkıştırılmasını sağlar. Kod çözücüde bilgi tekrar inşa edilirken kodlayıcıda sırayla yapılmış Huffman kodlaması tersten işlenip

kuantalama tablosu ile dekuantizasyonu yapılır ve son olarak ters Fourier işlemi ile resmin doğal hali alıcıda elde edilir.

- MPEG-1 video kodlama standardı, birincil olarak video sinyalini 30 fps (saniyeki kare sayısı) ve 1.5 Mbps (bit aktarım oranı) amaçlı tasarlanmıştır. MPEG yada Motion Photographic Experts Group video kodlaması bu tasarım ile dört farklı mevcut fazlalık video verisini ayıklar.
    1. Kareler arası geçici fazlalık
    2. Bir kare içindeki pikseller arası fazlalık
    3. Psychovisiual fazlalık
    4. Entropik kodlama fazlalığı

Prensipte standart bir video görüntüsünde komşu kareler fazlaca bağıntılıdır. Bu ortalama hassasiyette video bir kareden diğerine geçerken sinyalin hızla değişmemesi demektir. Sonuç olarak komşu karelerin varyansı video sinyalinin varyansından oldukça küçüktür.Bu verimli sıkıştırlmış bir video sinyali üretirken kareler arası geçici fazlalığı önemli ölçüde azaltır. JPEG kodlamasında olduğu gibi pikseller arası fazlalık DCT, kuantalama ve kayıpsız entropik kodlama ile azaltılır. (Şekil 1.4) Net sonuç tam hareketli bir video görüntüsü 1.5 Mb/s bir bilgisayar verisine dönüştürülmesidir. Bu veri kompakt disklerde, yazı yada resimlere entegre edilmiş halde saklanabilir. Sıkıştırmış veri mevcut bilgisayar ve telekominükasyon ağları üzerinden iletilecektir. Bu şekilde talep edilen video görüntüsünün iletişimi sağlanabilcektir. (1)



Şekil 1. 4 MPEG Kodlaması blok diyagramı.

(https://www.eetimes.com/introduction-to-mpeg-4-video-compression/)

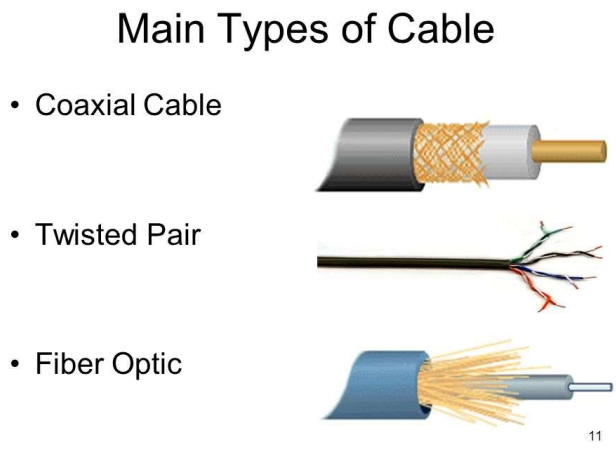
2. HABERLEŞME AĞLARI

Haberleşme ağları birbiri ile bağlantılı mikroişlemciler kullanarak üretilmiş yönlendiriciler barındırır. Bu işlemcilerin temel amacı verinin ağa rotalandırılmasını sağlamaktır. Her yönlendirici bir yada daha fazla host barındırır. Host cihazları bir başka host cihazıyla iletişim kurabilir. Ağ paylaşılan hareketli verinin hostlar arasında iletişimi ile yeni servisler yada uygulamaların desteklenmesine hizmet edecek şekilde tasarlanmıştır.

Telefon ağı bir haberleşme ağına örnektir. Telefon ağında iletişim ağı devre anahtarlaması ile 2 host arasında belirlenir. (1)

2.1. Kablolu Ağlar

Kablo kullanılarak bir noktadaki cihazdan başka bir noktadaki cihaza ses ve görüntü iletilmesini sağlar. Gönderilecek veri elektrik sinyalleri ile iletilir. Twisted-Pair, fiber ve koaksiyel kablolar günümüzde en yaygın kullanılan kablo çeşitleridir. Bant genişliği ise kısaca kapasitedir. Bant genişliği arttıkça aktarılabilecek maksimum veri miktarı da bant genişliği ile doğru orantılı olarak artar. Kablolu haberleşmede yaygın kullanılan kablo tiplerinin şekilleri Şekil 2.1'de verilmiştir.



Şekil 2. 1 Yaygın kullanılan haberleşme kablo tipleri.
(https://x9security.com/networking-4-local-networks/)

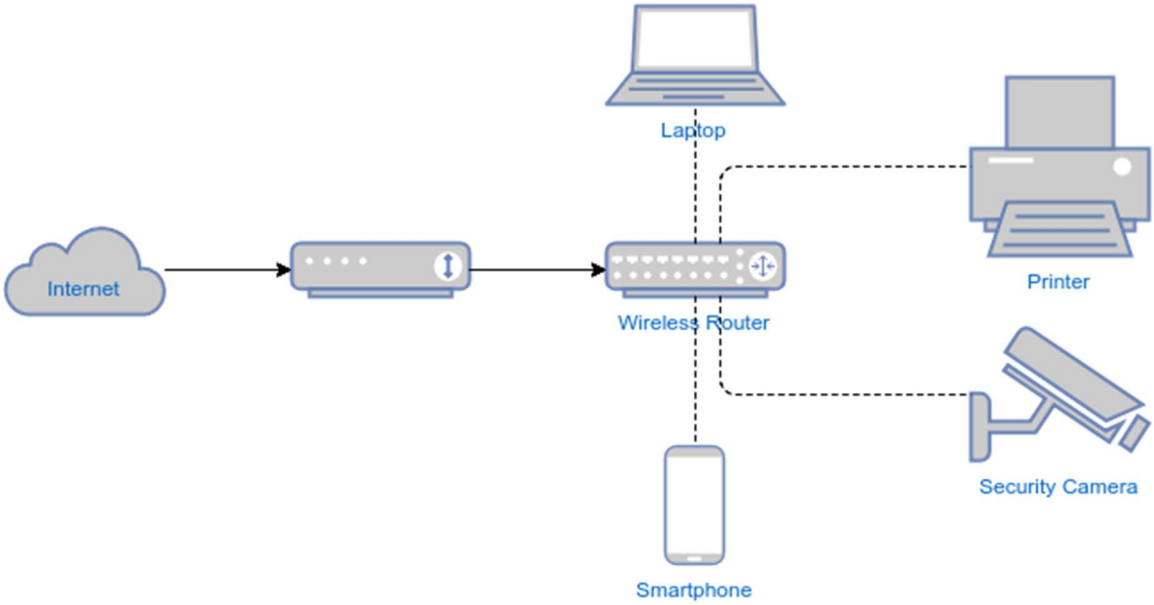
- Twisted-Pair kablolar günümüzde en yaygın kullanılan ağ kablosudur. UTP ve STP olmak üzere iki tiptir. Maksimum 100 metreye kadar haberleşme sağlanabilir.
- Koaksiyel kablolar iç ve dış iletkenler arası elektromanyetik alan kullanılarak iletişim yapılmasını sağlar.
- Fiber optik kablolar saf cam ip üzerinden ışığın geçmesi ile iletim yapmaktadır. Maksimum iletişim mesafesi kayıpsız olmak üzere uzak mesafeler için idealdir. Çevresel şartlardan etkilenmez ve iletimi ışık hızında gerçekleştirirler.

2.2. Kablosuz Ağlar

Kablolu ağlardan farklı olarak, radyo dalgaları kullanılarak kurulan kablosuz ağ teknolojisi de yaygınlık kazanmıştır. Kablosuz haberleşmenin hızlı gelişimi kablo masraflarının eliminasyonu, hareket halindeki dizüstü bilgisayarlara ağ erişimi sağlaması gibi önemli avantajlarıyla doğru orantılıdır. Kablolu ağlara kıyasla hızın yavaş olması, sinyal gürültüsünden, duvarlardan ve yağmurdan etkilenmesi ise kablosuz ağların eksik yönlerindendir. Kablosuz ağlar için iki temel standart vardır. Bunlar;

- 802.11(Wi-Fi): Kısa mesafede(birkaç yüz metre) bilgisayarları yada 802.11 destekli cihazlar arası iletişimi sağlamak için kullanılan ve git gide yaygınlaşan kablosuz ağ teknolojisidir. 1997 yılında ilk standart yayınlanmış, 1999 yılında ise 802.11a (54 Mbps hızında) ve 802.11b(11Mbps hızında) olarak iki sürümü ortaya çıkmıştır. 2003 yılında standartlaşan 802.11g sürümü, yine 54 Mbps hızında çalışan ancak daha kaliteli sinyal sağlayan bir teknolojidir.
- 802.16: Uzak mesafelerde (50 kilometreye kadar) etkili bir kablosuz ağ teknolojisidir, 2001 yılında onaylanıp standart haline getirilmiştir. Daha sonraki sürümleri WiMAX adı ile yaygınlaşmaya başlamıştır. Bu teknolojiyle sabit noktalar arasında iletişim kurulabileceği gibi, taşınabilir cihazlar ile (Cep telefonu, dizüstü bilgisayar, el bilgisayarı) da bağlantı sağlanabilir. Bu sistemle binalar arasında bakır

ya da fiber kablo kullanmak yerine radyo dalgaları ile hızlı Internet erişimi sağlanabilecektir. Şekil 2.2'de evlerde kullanılan kablosuz haberleşme ağına örnek bir diyagram verilmiştir.



Şekil 2. 2 Kablosuz haberleşme ağı örneği.

Kablosuz Ağların Avantajları

- **Kolaylık**: Günümüzde tüm dizüstü bilgisayarlarda ve birçok cep telefonunda, bir kablosuz ağ'a doğrudan bağlanmak için gereken WiFi teknolojisi bulunmaktadır. Şirketler için çalışanlar, kapsama alanındaki herhangi bir yerden ağ kaynaklarına güvenli bir şekilde erişebilir. Kapsama alanı sınırlıdır, ancak birden fazla binayı içerecek şekilde genişletilebilir.
- **Mobilite**: Cihazların hareketliliğine yada ağ değişikliklerine adaptasyonu kolaydır.
- **Verimlilik**: Ağ kablolarının zamana bağlı deformasyonu ve veri hızının bu deformasyon nedeniyle düşmesi sorunundan bağımsızdır.
- **Kurulum kolaylığı**: Kurulum için fiziksel kablolar ile kullanılmadığından kurulum hem hızlı hem de düşük maliyetli olur. Kablosuz ağlar ağ bağlantısını, fiziksel olarak kabloların erişmesinin zor olduğu yerlerde iletişimine izin verir.

- **Maliyet**: Ofis taşınmaları, yeniden yapılandırmalar veya genişletmeler sırasında kablolama maliyetlerini ortadan kaldırdığı veya azalttığı için bir kablosuz ağın işletim ve kullanım maliyeti daha düşüktür.

Kablosuz Ağların Kullanım Alanları

- Telekomünikasyon
- Tıp
- Radyo, Televizyon
- Savunma Sanayi
- Endüstriyel Üretim
- Ulaşım

Kablosuz Ağ Teknolojileri

- Kızılötesi Teknolojisi
- Bluetooth Teknolojisi
- Wi-Fi Teknolojisi
- GPRS Teknolojisi
- 3G Teknolojisi
- 4G Teknoloji

3. WI-FI TEKNOLOJİSİ

Wi-Fi (Wireless Fidelity / Kablosuz Bağlantı Alanı) kişisel bilgisayarların dijital ses oynatıcılar yada akıllı telefonlar gibi cihazların kablosuz olarak birbirlerine bağlanmasını sağlayan teknolojidir. Wi-Fi ürünlerin kablosuz bağlantı sağlayabildiğini gösteren bir uyumluluk göstergesidir ve IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n ve IEEE 802.11ac standartlarına göre belirlenir. Wi-Fi dizüstü bilgisayarlar, PDAlar ve diğer taşınabilir cihazların yakınlarındaki kablosuz erişim noktaları aracılığıyla yerel alan ağına bağlanabilmesini sağlar. Bağlantı, kablosuz erişim noktaları ve cihazın ortak desteklediği, IEEE 802.11 protokolüne bağlı olarak 2.4 GHz veya 5 GHz radyo frekansında gerçekleştirilir. Veri, CSMA/CA (Carrier sense multiple access with collision avoidance) protokolüne uygun gönderilip alınır ve böylece paketlerin iletimi sırasında hata oluşması sorunu çözülür.

Wi-Fi; kişisel bilgisayar, akıllı telefon, oyun konsolu ve dijital işitsel cihazların kablosuz ağ sahası içerisinde internete bağlanmasını sağlayan bir teknolojidir. Wi-Fi'ın erişim noktaları, bir oda gibi küçük alanlardan, birkaç milkarelik geniş alanlara kadar etki edebilir, buralarda internet erişimini sağlayabilirler. Ev ve ofislerdeki özel kullanımının yanında kamuya ait alanlarda da ücretsiz olarak veya diğer ticari şekillerde kullanılmaktadır.

Havaalanları, otel ve restoran gibi yerlerde genel olarak bu hizmet ücretsiz olarak sağlanmaktadır. 2008'den bu yana 300 metropolitte Wi-Fi (Muni-Fi) projesi başlamıştır.

3.1. Wi-Fi İsmi

Wi-Fi ismi bilinenin aksine "Wireless Fidelity"'den türetilmemiştir, Wi-Fi ismi,ilk defa Ağustos 1999'da reklam amaçlı kullanıldı.Interbrand Corporation denilen bir marka danışmanlık şirketi tarafından bulunduğu söylenildi. Belanger ayrıca Interbrand'ın Hi-Fi'dan ilham alarak bir kelime oyunuyla Wi-Fi'ı icat ettiğini ve yin-yang tarzı bir Wi-Fi logosu hazırlandığını da belirtti. Wi-Fi şirketi önceleri Wi-Fi için "Kablosuz Bağlantı için Standart" sloganını kullandı, fakat daha sonra ifadeyi piyasadan kaldırdı. (2)

3.2. Wi-Fi Özellikleri

- Lisans gerektirmeyen frekanslarda çalışır
- Ağ için kablolama gereksinimi yoktur, böylece kablo çekilemeyecek binalarda veya binalar arası bağlantılarda kolaylık sağlar.
- Diğer kablosuz çözümlere göre çok daha ucuz ve kolay alınıp kurulabilir.
- Birden çok kablosuz erişim noktası kullanılan ağlarda kablosuz dolaşım ile kablosuz iletişim kesilmeden bir erişim noktasından diğerine geçiş yapılabilir.
- WEP, WPA ve benzeri kablosuz şifreleme yöntemleri veya IEEE 802.1x gibi yetkilendirme yöntemleriyle çeşitli güvenlik seçenekleri sunar.
- Wi-Fi Global bir standart kümesidir, dünyanın her yerinde aynı şekilde çalışır.

3.2. Wi-Fi Menzili

Wi-Fi ağlarının menzili kullanılan ortama göre oldukça değişik mesafelere ulaşmaktadır. Bu sinyal kalitesinin çevredeki yapılara bağlı olarak düşmesiyle alakalı bir durumdur. Sinyal seviyesindeki düşüş ile hız kaybı da oluşmaktadır. Teorik olarak 300 mbit olarak ifade edilen hız çevresel faktörlere göre 0 mbit'e kadar düşmekte bu durum da

bağlantıda kopmalara neden olmaktadır. Teorik olarak 1.5 km ifade edilen hız kapalı alanlarda ve mevcut engellere bağlı olarak 300 metreye kadar düşmektedir.

3.3 Wi-Fi Dezavantajları

- Lisans gerektirmeyen frekans aralıklarında çalıştığı için, Wi-Fi cihazlar diğer kablosuz cihazlarla çakışabilir veya birbirlerinin iletişimini engelleyebilirler.
- 2.4 GHz frekans aralığında çalışan 802.11b ve 802.11g uyumlu cihazların iletişim kalitesi ve hızı, diğer Wi-Fi cihazlar dışında, Bluetooth, mikrodalga fırın, telsiz telefon, bazı telsizler ve benzeri radyo sinyalleriyle çalışan cihazlar tarafından düşürülebilir veya tamamen engellenebilir.
- Wi-Fi için yapılan uluslararası düzenlemelerin tümü aynı olmadığı için değişik ülkeler için üretilen cihazların bazı kanallarda uyumsuzluk çıkarması olasıdır.
- Diğer standartlara göre güç tüketimi oldukça yüksektir.
- En yaygın kablosuz şifreleme standardı, WEP'in, düzgün yapılandırıldığında bile kolaylıkla kırılabildiği görülmüştür. 2003 te cihazlarda kullanılmaya başlanan WPA ve WPA2 şifreleme standartları ise bu sorunu çözmeyi amaçlamıştır.
- Wi-Fi kullanıcılarının küçük bir kısmı, Wi-Fi kullandıktan sonra bazı sağlık sorunlarıyla karşılaştıklarını bildirmişlerdir. Wi-Fi hastalığı veya Wi-Fi duyarlılığının, olağandışı baş ağrısıyla birlikte şu belirtileri gösterdiği bildirilmiştir: mide bulantısı, kalpte ritim düzensizliği, denge kaybı ve baş dönmesi, göğüs ağrısı, aşırı stres, panik atak ve idrak ile ilgili sorunlar. Bazı sağlık uzmanları, bu sağlık sorunlarını nörolojik unsurlarla açıklamıştır

3.4. Wi-Fi Güvenliği

Kablosuz iletişimde radyo sinyalleri kullanıldığı için, kullanıcı ile kablosuz erişim noktası arasındaki iletişim dinlenebilir. Bunu engellemek için WEP, WPA ve WPA2 gibi şifreleme yöntemleri kullanılsa da, bu şifreleme yöntemleri halen yeterince güvenli

görülmemektedir. Yeterli sayıda şifrelenmiş paket toplandıktan sonra birçok şifreli kablosuz iletişim çözülebilir. Bu yüzden, çeşitli şirketlerde, okullarda vb yerlerde kablosuz bağlantı yapan kullanıcıların ağa erişmeleri için VPN ve benzeri, üst katmanlarda çalışan şifreli iletişim yöntemleri kullanılmaktadır. Protokolü 802.1x dir. Her standart kendine özel frekanslarda ve veri aktarım hızlarında çalışır. Bu frekanslar çizelge 3.1'de verilmiştir.

Çizelge 1. Wi-Fi çalışma frekansları.

Standart	Frekanslar	Kanal Sayısı
IEEE 802.11a	5,15 GHz - 5,725 GHz	19, Avrupa'da TPC ve DFS 802.11h
IEEE 802.11b	2,4 GHz - 2,4835 GHz	11 Adet ABD, 13 Adet Avrupa,14 Adet Japonya
IEEE 802.11g	2,4 GHz - 2,4835 GHz	11 Adet ABD, 13 Adet Avrupa,14 Japonya

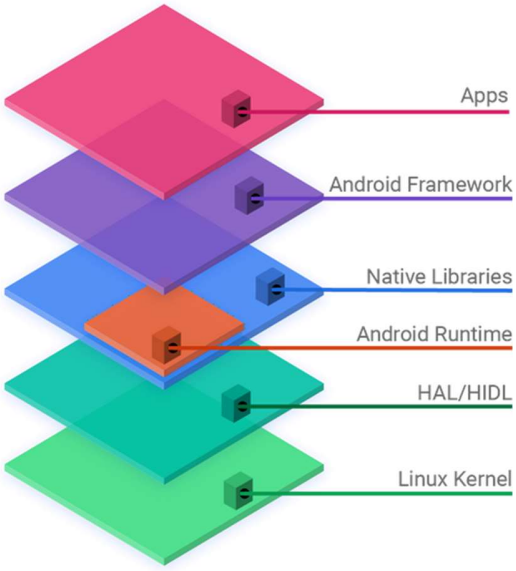
Çizelge 2. Wi-Fi veri aktararım hızları.

Standart	Hız (Mpbs)
IEEE 802.11	2 Mbps max.
IEEE 802.11a	54 Mbps max. (40 MHz'ta 108 Mbps)
IEEE 802.11b	11 Mbps max. (40 MHz'ta 22 Mbps, 60 MHz'ta 44 Mbps)
IEEE 802.11g	54 Mbps max. (g+ =108 Mbps, 125 Mbps'a kadar mümkün; 2 Mbps Karma (g+b) IEEE 802.11b)
IEEE 802.11h	54 Mbps max. (40 MHz'ta 108 Mbps)
IEEE 802.11n	300 Mbps max. (kullanılan Teknik MIMO; 20. Ocak 2006'da tasarlandı; Versiyon 2.0 19. Mart 2007'de geliştirildi)
IEEE 802.11ac	6,7 Gbps max. (80Mhz'te min 433mbps max 1,3Gbps) (160 MHz'te min 1,69Gbps)

5. ANDROID

Android; Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilmekte olan, Linux tabanlı özgür ve ücretsiz bir işletim sistemidir. Sistem açık kaynak kodlu olsa da kodlarının ufak ama çok önemli bir kısmı Google tarafından kapalı

tutulmaktadır. Google tarafından ücretsiz olmasının sebebi, sistemin daha hızlı ve çabuk gelişmesi, birçok popüler marka tarafından kullanılması ve bu sayede reklamlarını daha fazla kişiye ulaşmasını sağlamaktır. Google, Android sistemi üzerinde çalışan Google Play marketteki oyun ve uygulamalar üzerinde aldığı reklamları yayınlayarak para kazanmaktadır. Android'in desteklenen uygulama uzantısı ".apk"dır. Android, Linux çekirdeği üzerine inşa edilmiş bir mobil işletim sistemidir. Bu sistem ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu Java-uyumlu kütüphaneleri içine alan uygulama iskeleti üzerinden çalışmaktadır. Android, derlenmiş Java kodunu çalıştırmak için dinamik çevirmeli Android Runtime (ART) kullanır ve cihazların fonksiyonelliğini artıran uygulamaların geliştirilmesi için çalışan geniş bir programcı-geliştirici çevresine sahiptir. (3)



Şekil 4. 1 Android Mimarisi

4.1 Android Mimarisi

4.1.1 Temel yapı (Linux çekirdeği)

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. Şekil 4.2'de Androide ait Linux çekirdek yapısı verilmiştir.

Şekil 4. 2 Linux Çekirdeği

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.

4.1.2 Kütüphaneler (Libraries)

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. Şekil 4.3

Şekil 4. 3 Android sistem kütüphaneleri

4.1.3 Android Runtime

Bu bölüm Linux çekirdeğindeki kütüphanelerin Java ile birleştiği bölümdür. İki önemli bileşeni vardır. Bunlar temel Java kütüphaneleri ve Dalvik Sanal (virtual) Makinesi'dır (Şekil 4.4). Uygulamalar Dalvik Sanal Makinesi tarafından çalıştırılır. Temel

çalışma mekanizmasını anlamak Android projelerinin yaşam döngüsünü anlamak açısından önemlidir. Java ile yazılan uygulamalar alınır, Java kodları derlenerek bytecode dosyalarına çevrilir. Bu dosyalar dex dosyasına çevrilerek Dalvik Sanal Makinesi'nin çalıştıracağı şekle sokar. Dalvik ortamı düşük işlemci gücü, az RAM ve sınırlı batarya koşullarına göre tasarlanmıştır.

Application framework (Uygulama çatıları) belirli bir ortam için uygulamaların geliştirilmesini desteklemek için temel bir yapı sağlayan bir yazılım kütüphanesidir. Bir

Şekil 4. 4 Android runtime

uygulama çatısı, bir uygulama oluşturmak için iskelet desteği görevi görür. Uygulama çatısı tasarlamanın amacı, uygulamaların geliştirilmesi sırasında karşılaşılan genel sorunları azaltmaktır. Bu, bir uygulamanın farklı modülleri arasında paylaşılabilen kod kullanılarak başarılır. Uygulama çatıları sadece grafiksel kullanıcı arayüzü (GUI) geliştirmede değil web tabanlı uygulamalar gibi diğer alanlarda da kullanılmaktadır. (4)

4.1.4 Android Yazılım Geliştirmesi

Android işletim sistemi için yeni uygulamaların oluşturulduğu süreçtir. Uygulamalar genellikle Android yazılım geliştirme kiti (SDK) kullanılarak Java programlama dilinde geliştirilir, ancak diğer geliştirme ortamları da mevcuttur. Android yazılım geliştirme kiti (SDK) kapsamlı bir geliştirme araçları seti içerir. Bunlar arasında hata ayıklayıcı, kitaplık, QEMU'ya dayalı ahize düzenleyici, belgeler, örnek kod ve öğreticiler bulunur. (5)

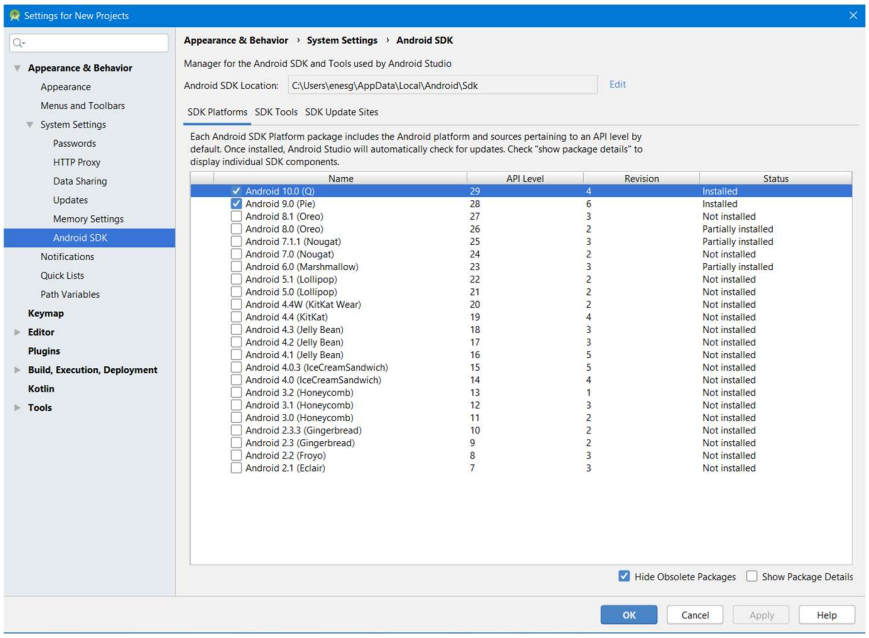
IntelliJ IDEA Java'da bilgisayar yazılımı geliştirmek için kullanılan bir tümleşik geliştirme ortamıdır (IDE). Google tarafından hazırlanan ve IntelliJ tarafından desteklenen Android Studio resmi ortamdır. Geliştiriciler başkalarını kullanmakta özgürdür, ancak Google Android Geliştirme Araçlarının (ADT) resmi geliştirme ortamı olarak Android Studio'ya odaklanması için 2015'ten bu yana resmi olarak önerilmeyeceğini açıklamıştır. Ayrıca, geliştiriciler Java ve XML dosyalarını düzenlemek için herhangi bir metin düzenleyiciyi kullanabilir, daha sonra Android uygulamalarını oluşturmak, hata ayıklamak,

ekli Android aygıtlarını kontrol etmek için komut satırı araçlarını (Java Geliştirme Kiti ve Apache Ant gereklidir) kullanabilirler. (6)

Android uygulamaları APK formatında paketlenir ve Android işletim sistemindeki / data / app klasörü altında saklanır (klasöre güvenlik nedeniyle yalnızca kök kullanıcı tarafından erişilebilir). APK paketi, .dex dosyaları (Dalvik yürütülebilir dosyaları, derlenmiş bayt kod dosyaları), kaynak dosyaları vb. içerir. (7)

4.2 Android Studio'nun Kurulumu

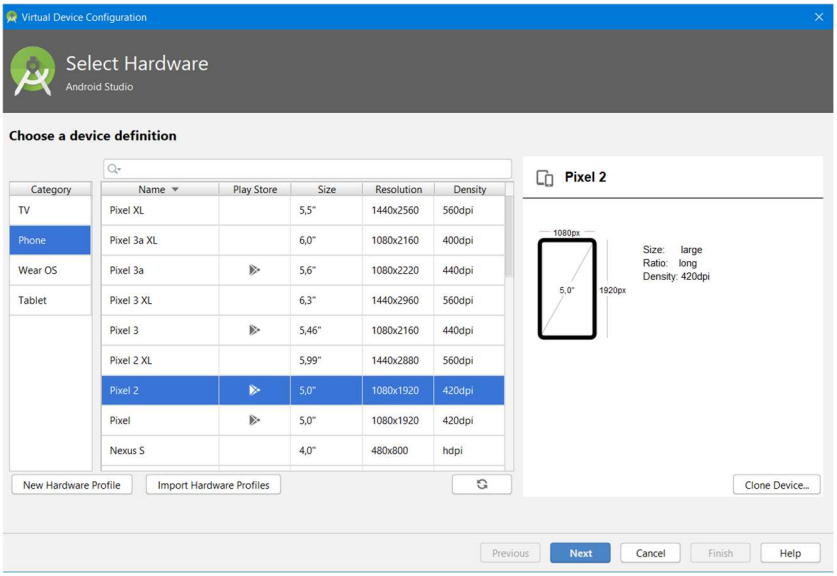
Bu çalışmada Android Studio 3.5.3 versiyonu kullanılmıştır. Android Studio'nun resmi sitesinden kurulum dosyası indirilip kurulduktan sonra SDK kurulumları yapılmalıdır. Android Studio 3.5.3 versiyonu kurulurken SDK 29'da kurulmaktadır. Fakat daha düşük sürümler de uygulama yazabilmek istenirse gerekli olan SDK'ları da indirmek gerekir. Örneğin yazılan uygulamanın Android 4.0(SDK 14) üzeri desteklemesini isteniyorsa ve o aradaki (SDK 14 – SDK 25) cihazlarda da test edilecekse, o aradaki SDK'ların da indirilmesi gerekmektedir. (Şekil 4.5) SDK Manager'dan istenen SDK'ler indirilebilmektedir.



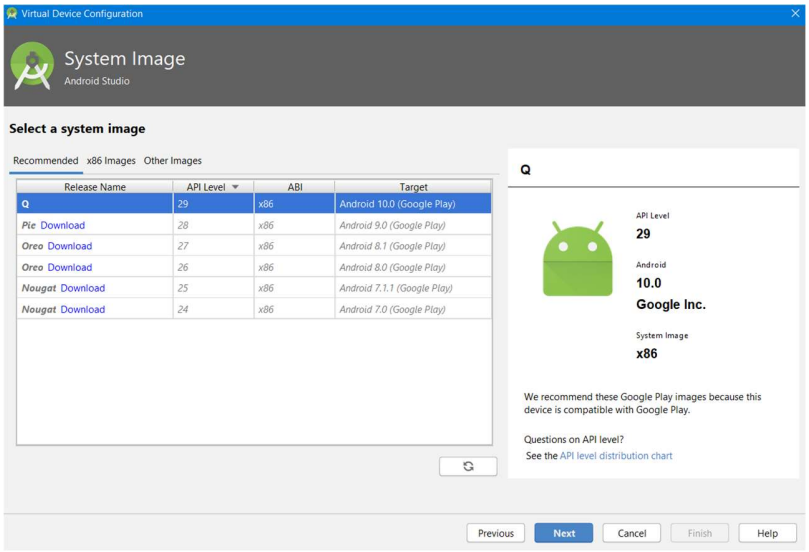
Şekil 4. 5 SDK Kurulum Ekranı

4.3 AVD Manager Kurulumu

AVD (Android Virtual Device) Manager, bilgisayara sanal olarak telefon (emülatör) eklemek ve çalıştırmak için kullanılır. Eklenecek emülatörün Android sürümü (API) ayarlanabileceği gibi ram ve hafıza boyutu kamera ayarlaması vb. çeşitli ayarlamalar yapılabilmektedir. AVD Manager'in ekran görünümü Şekil 4.6 ve Şekil 4.7 verilmiştir. (8)



Şekil 4. 6 AVD Kurulum Ekranı

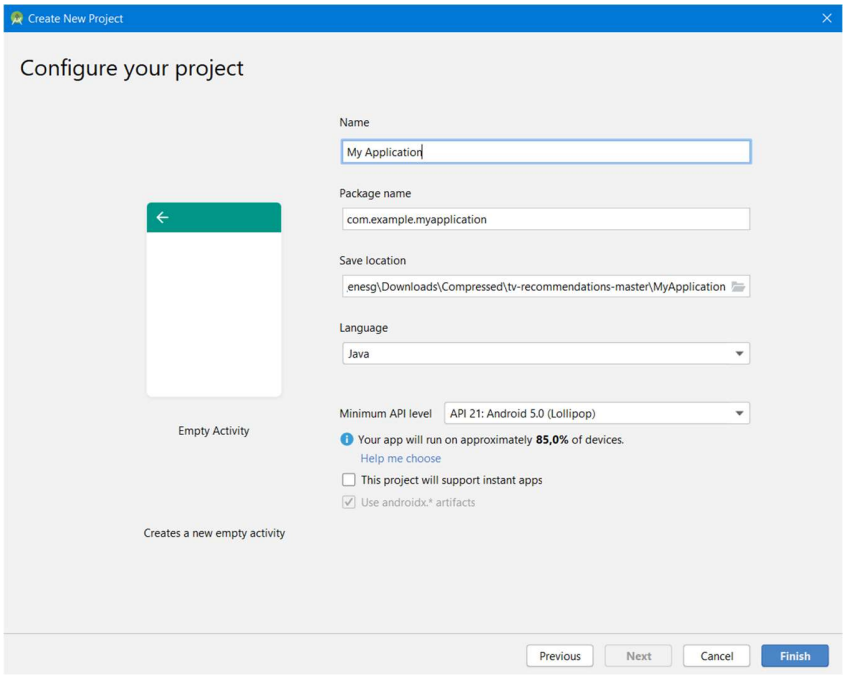


Şekil 4. 7 AVD Kurulum Ekranı

4.4 Proje Dosyasının Oluşturulması

Yeni proje oluşturma ekranı Şekil .. gibidir. Buradaki alanlar aşağıda açıklanmıştır.

- Application Name: Uygulama Adımız (Türkçe karakter kullanılmaması önerilir). Cihazlarda ikonların altında gözükecektir.
- Package Name: Company domain ve package name'e göre otomatik oluşur.Değiştirilmek istenirse sağ tarafındaki edit ile değiştirilebilir. Package name Google Play'deki unique id'niz gibi düşünebilirsiniz. Aynı package name ile başka bir uygulama yüklenemez.
- Project Location: Projemizin oluşturulacağı klasör.



Şekil 4. 8 Android Studio proje konfigurasyonları ekranı

5. EKRAN YANSIT UYGULAMASININ TASARIMI

5.1 Giriş Sayfası

Giriş sayfası için Main2Activity.java sınıfı (class) açılmıştır. Tasarım activity_main2.xml sayfasında yapılmıştır. Giriş ekranında genel proje tanıtımı ve ekran yansıtma ekranına geçme butonu bulunmaktadır. Şekil 5.1'de giriş sayfası ekran görüntüsü verilmiştir.

Şekil 5. 1 Uygulamanın giriş ekranı

'YAYIN EKRANI' adlı butonun onClick metoduna 'degistir' değişkeni atanmıştır. Main2Activity.java sınıfında buton basıldığında sayfanın değişmesi gereken kod yazılmıştır. (EK1)

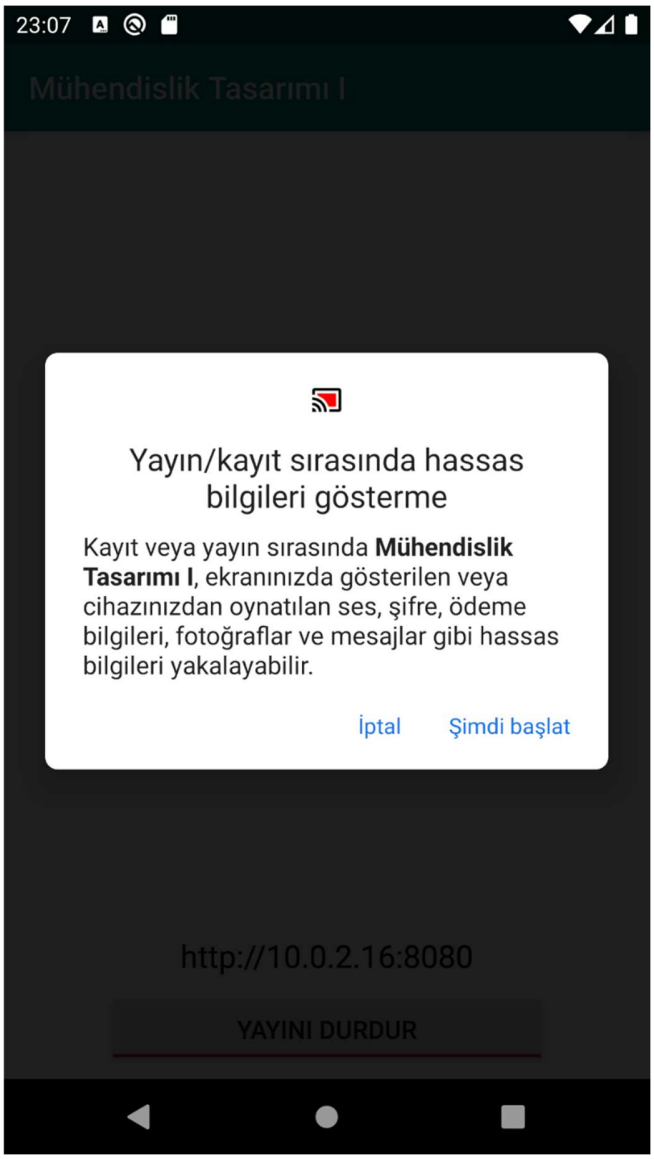
```
public void degistir (View view){

   Intent intent = new Intent(Main2Activity.this,MainActivity.class);
   startActivity(intent);
}
```

5.2 Yayınlama Ekranı

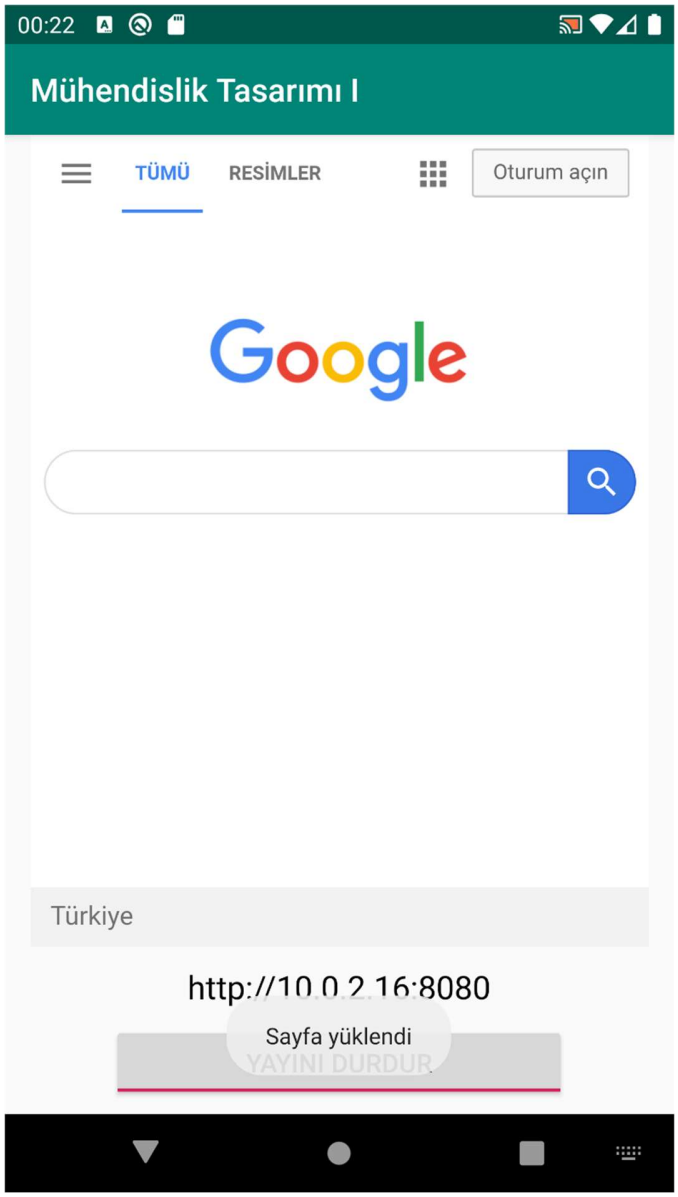
Şekil 5. 2 Uygulamanın yayın ekranı

Yayınlama Ekranında ‘Yayını Başlat’ butonu ve metin kutusu (TextView) bulunmaktadır. Yayını Başlat butonuna basıldığında metin kutusunda aktarılacak görüntüyü izleyebilmek için gerekli IP adresi metin kutusuna yazmaktadır. Yayını Başlat butonuna basıldıktan sonra uygulama ekranın yansıtılması için kullanıcıdan izin istemektedir.



Şekil 5. 3 Ekran kaydı izin penceresi

İzin verildikten ekran yansıtılmaya başlamaktadır. Uygulamanın bu sayfasındaki görüntü kablosuz olarak aktarılmaya başlamıştır.



Şekil 5. 4 Aktarılan görüntü

EKLER

Ek-1 Uygulamanınn giriş ekranının Java kodları

```
package com.example.ekranyansit;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class Main2Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
    public void degistir (View view){

        Intent intent = new Intent(Main2Activity.this,MainActivity.class);
        startActivity(intent);
    }
}
```

Ek-2 Yayın ekranının Java kodları

```
package com.example.ekranyansit;

import android.app.ProgressDialog;
import android.content.Intent;
import android.media.projection.MediaProjection;
import android.media.projection.MediaProjectionManager;
import android.os.HandlerThread;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.CompoundButton;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import ekranYansit.ForegroundServiceHandler;
import ekranYansit.NotifyImageGenerator;
import ekranYansit.screenHelper;
import android.os.Process;

import static com.example.ekranyansit.EkranYansit.getAppData;
import static com.example.ekranyansit.EkranYansit.getProjectionManager;
import static com.example.ekranyansit.EkranYansit.setMediaProjection;
```

```
public class MainActivity extends AppCompatActivity {

private screenHelper mscreenHelper;
private static MainActivity sAppInstance;
   private static final int REQUEST_CODE_SETTINGS = 2;
   private static final int REQUEST_CODE_SCREEN_CAPTURE = 1;
   private ForegroundServiceHandler mForegroundServiceTaskHandler;
      private HandlerThread mHandlerThread;
   private boolean buttonClicked = false;
   private NotifyImageGenerator mNotifyImageGenerator;
   // private WebView webView;
   @Override
   protected void onCreate(Bundle savedInstanceState) {


      sAppInstance = this;
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      mscreenHelper = new screenHelper(this);
     TextView myAwesomeTextView = (TextView)findViewById(R.id.editText);
      myAwesomeTextView.setText(getAppData().getServerAddress());
      ToggleButton toggle = (ToggleButton) findViewById(R.id.btn1);
            mHandlerThread = new HandlerThread(
            EkranYansit.class.getSimpleName(),
            Process.THREAD_PRIORITY_MORE_FAVORABLE);
      mHandlerThread.start();
```

```
        mForegroundServiceTaskHandler = new
ForegroundServiceHandler(mHandlerThread.getLooper());
        toggle.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
        {
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
                if (isChecked) {
                    buttonClicked=true;
                    final MediaProjectionManager projectionManager = getProjectionManager();
                    if (projectionManager != null) {
                        startActivityForResult(projectionManager.createScreenCaptureIntent(),
REQUEST_CODE_SCREEN_CAPTURE);
                    }
                } else {

mForegroundServiceTaskHandler.obtainMessage(ForegroundServiceHandler.HANDLER_
STOP_STREAMING).sendToTarget();
                }
            }
        });
    }
    @Override
    public void onPause()
    {
        super.onPause();
        if(!buttonClicked)
        {
```

```
mForegroundServiceTaskHandler.obtainMessage(ForegroundServiceHandler.HANDLER_
STOP_STREAMING).sendToTarget();
            mNotifyImageGenerator = new NotifyImageGenerator(getApplicationContext());
            mNotifyImageGenerator.addDefaultScreen();
            ToggleButton toggle = (ToggleButton) findViewById(R.id.btn1);
            toggle.setChecked(false);
        }


    }
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        switch (requestCode) {
            case REQUEST_CODE_SCREEN_CAPTURE:
                if (resultCode != RESULT_OK) {
                    Toast.makeText(this,
getString(R.string.main_activity_toast_cast_permission_deny),
Toast.LENGTH_SHORT).show();
                    ToggleButton toggle = (ToggleButton) findViewById(R.id.btn1);
                    toggle.setChecked(false);
                    return;
                }
                buttonClicked=false;
                final MediaProjectionManager projectionManager = getProjectionManager();
                if (projectionManager == null) return;
                final MediaProjection mediaProjection =
projectionManager.getMediaProjection(resultCode, data);
                if (mediaProjection == null) return;
                setMediaProjection(mediaProjection);
```

```
mForegroundServiceTaskHandler.obtainMessage(ForegroundServiceHandler.HANDLER_
START_STREAMING).sendToTarget();
            break;
        default:
    }
    WebView webview = (WebView) findViewById(R.id.webview);
    webview.getSettings().setJavaScriptEnabled(true);
    webview.loadUrl("https://google.com");



    final ProgressDialog progress = ProgressDialog.show(this, "Yayın Başlatılıyor",
"Yükleniyor....", true);
    progress.show();
    webview.setWebViewClient(new WebViewClient() {
        @Override
        public void onPageFinished(WebView view, String url) {
            super.onPageFinished(view, url);
            Toast.makeText(getApplicationContext(), "Sayfa yüklendi",
Toast.LENGTH_SHORT).show();
            progress.dismiss();
        }
        public void onReceivedError(WebView view, int errorCode, String description,
String failingUrl) {
            Toast.makeText(getApplicationContext(), "Bir hata oluştu",
Toast.LENGTH_SHORT).show();
            progress.dismiss();
        }
    });
```

Ek-3 Sunucu ve servis Java kodları

```
package com.example.ekranyansit;
import android.app.Application;
import android.content.Context;
import android.media.projection.MediaProjection;
import android.media.projection.MediaProjectionManager;
import android.support.annotation.Nullable;
import ekranYansit.AppData;
import ekranYansit.HttpServer;
import ekranYansit.ImageGenerator;

public class EkranYansit extends Application {
    private static EkranYansit sAppInstance;
    private MediaProjection.Callback mProjectionCallback;
    private AppData mAppData;
    private ImageGenerator mImageGenerator;
    private MediaProjection mMediaProjection;
    private MediaProjectionManager mMediaProjectionManager;
    private HttpServer mHttpServer;
    @Override
    public void onCreate() {
        super.onCreate();
        sAppInstance = this;
        mAppData = new AppData(this);
        getAppData().initIndexHtmlPage(this);
        mHttpServer = new HttpServer();
```

```
        mImageGenerator = new ImageGenerator();
        mMediaProjectionManager = (MediaProjectionManager)
getSystemService(Context.MEDIA_PROJECTION_SERVICE);
        mHttpServer.start();
    }

    public void onDestroy() {
        mHttpServer.stop(null);
    }

    public static AppData getAppData() {
        return sAppInstance.mAppData;
    }

    public static void setMediaProjection(final MediaProjection mediaProjection) {
        sAppInstance.mMediaProjection = mediaProjection;
    }
    @Nullable
    public static MediaProjection getMediaProjection() {
        return sAppInstance == null ? null : sAppInstance.mMediaProjection;
    }

    @Nullable
    public static ImageGenerator getImageGenerator() {
        return sAppInstance == null ? null : sAppInstance.mImageGenerator;
    }
```

```
    @Nullable
    public static MediaProjectionManager getProjectionManager() {
        return sAppInstance == null ? null : sAppInstance.mMediaProjectionManager;
    }
}
```

Ek-4 Giriş ekranı tasarımı XML kodları

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    tools:context=".Main2Activity">

    <TextView
        android:id="@+id/textView6"
        android:layout_width="126dp"
        android:layout_height="31dp"
        android:layout_marginStart="12dp"
        android:layout_marginBottom="12dp"
        android:gravity="center"
        android:text="2019, Güz"
        android:textColor="#000"
        app:autoSizeTextType="none"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="120dp"
```

```
    android:layout_height="120dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/uulogo" />

<ImageView
    android:id="@+id/imageView4"
    android:layout_width="120dp"
    android:layout_height="120dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/muhendisliklogo" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="305dp"
    android:layout_height="120dp"
    android:layout_marginTop="16dp"
    android:gravity="center"
    android:text="Ekranlı Cihazlar Arası Görüntü Aktarımı"
    android:textColor="#000"
    android:textSize="24sp"
    app:autoSizeTextType="uniform"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView3" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="320dp"
    android:layout_height="40dp"
    android:layout_marginTop="24dp"
    android:gravity="center"
    android:text="Proje Danışmanı: Prof. Dr. Tuncay Ertaş"
    android:textColor="#000"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="32dp"
    android:ellipsize="start"
    android:gravity="center"
    android:onClick="degistir"
    android:text="Yayın Ekranı"
    android:textSize="16sp"
```

```
        app:layout_constraintBottom_toTopOf="@+id/textView6"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Ek-5 Yayın ekranı tasarımı XML kodları

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    tools:context=".MainActivity">

    <ToggleButton
        android:id="@+id/btn1"
        android:layout_width="280dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="8dp"
        android:paddingTop="8dp"
        android:paddingBottom="8dp"
        android:textOff="Yayını  Başlat"
        android:textOn="Yayını Durdur"
        android:textSize="16sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/editText"
    android:layout_width="275dp"
    android:layout_height="45dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:autoText="false"
    android:gravity="center"
    android:text="TextView"
    android:textColor="#000"
    android:textSize="20sp"
    app:autoSizeMaxTextSize="32dp"
    app:autoSizeMinTextSize="16dp"
    app:autoSizeTextType="none"
    app:layout_constraintBottom_toTopOf="@+id/btn1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<WebView
    android:id="@+id/webview"
    android:layout_width="380dp"
    android:layout_height="500dp"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.516"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
</android.support.constraint.ConstraintLayout>

    }
}
```

KAYNAKÇA

1. **Haykin, Simon.** Communication Systems. *Communication Systems.* Ancaster, Ontario : John Wiley & Sons Inc., 2000, s. 1-33.

2. *Makale 295855.* **Journey, Digital.**

3. **Google.** The Android Source Code. *Android.* [Çevrimiçi] Google, 17 12 2014. [Alıntı Tarihi: 2019 12 14.] http://source.android.com/source/index.html.

4. **Application Framework.** ***Technopedia.*** **[Çevrimiçi] [Alıntı Tarihi: 16 12 2019.] https://www.techopedia.com/definition/6005/application-framework.**

5. How to install the Android SDK on Windows, Mac and Linux. ***Android Central.*** **[Çevrimiçi] [Alıntı Tarihi: 18 12 2019.] https://www.androidcentral.com/installing-android-sdk-windows-mac-and-linux-tutorial.**

6. Get the Android SDK. ***MIT students' portal.*** **[Çevrimiçi] MIT. [Alıntı Tarihi: 18 12 2018.] https://stuff.mit.edu/afs/sipb/project/android/docs/sdk/index.html.**

7. Backup & Restore Android Apps Using ADB. ***JonWestfall.*** **[Çevrimiçi] 25 8 2009. [Alıntı Tarihi: 18 12 2019.] http://jonwestfall.com/2009/08/backup-restore-android-apps-using-adb/.**

8. Telci, Serkan. Android Platformu İçin Gerekli Ayarlamalar. ***Mobil Uygulama (Android Ve Ios) Geliştirme.*** **Ankara : Seçkin Yayıncılık, 2019.**

9. ***Wi-Fi Definition is Not Wireless Fidelity.*** **Beal, Vangie. 14 Haziran 2010.**

10. ***What Wi-Fi Stands For.*** **Pogue, David. 1 Mayıs 2012.**

11. Android Intents. ***Vogella.*** **[Çevrimiçi] 18 06 2016. [Alıntı Tarihi: 19 12 2019.] https://www.vogella.com/tutorials/AndroidIntent/article.html.**