



**MOBİL CİHAZLARA UYGULAMA
GELİŞTİRMEK İÇİN KULLANILAN
PLATFORMLARIN VE
DİLLERİN KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Hilmi BİLİCİ

Danışman

Dr. Öğr. Üyesi Said Mahmut ÇINAR

**İNTERNET VE BİLİŞİM TEKNOLOJİLERİ
YÖNETİMİ ANABİLİM DALI**

Haziran 2019

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MOBİL CİHAZLARA UYGULAMA GELİŞTİRMEK İÇİN
KULLANILAN PLATFORMLARIN VE DİLLERİN
KARŞILAŞTIRILMASI

Hilmi BİLİCİ

Danışman

Dr. Öğr. Üyesi Said Mahmut ÇINAR

İNTERNET VE BİLİŞİM TEKNOLOJİLERİ YÖNETİMİ
ANABİLİM DALI

Haziran 2019

TEZ ONAY SAYFASI

Hilmi BİLİCİ tarafından hazırlanan “Mobil Cihazlara Uygulama Geliştirmek İçin Kullanılan Platformların ve Dillerin Karşılaştırılması” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 21/06/2019 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı’nda** **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

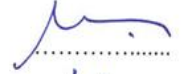
Danışman : Dr. Öğr. Üyesi Said Mahmut ÇINAR

İmza

Başkan : Doç. Dr. Devrim AKGÜN
Sakarya Üniv., Bilgisayar ve Bilişim Fak.



Üye : Dr. Öğr. Üyesi Said Mahmut ÇINAR
Afyon Kocatepe Üniv., Mühendislik Fak.



Üye : Dr. Öğr. Üyesi Emre AKARSLAN
Afyon Kocatepe Üniv., Mühendislik Fak.



Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

Prof. Dr. İbrahim EROL
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

**Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım
bu tez çalışmada;**

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

18/07/2019


Hilmi BİLİCİ

ÖZET
Yüksek Lisans Tezi

**MOBİL CİHAZLARA UYGULAMA GELİŞTİRMEK İÇİN KULLANILAN
PLATFORMLARIN VE DİLLERİN KARŞILAŞTIRILMASI**

Hilmi BİLİCİ

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

İnternet ve Bilişim Teknolojileri Yönetimi Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Said Mahmut ÇINAR

Günümüzde mobil cihazlar çok yaygın bir kullanım alanına sahiptir. Bu cihazlarda farklı işlemler için geliştirilmiş mobil uygulama olarak adlandırılan programlar kullanılmaktadır. Bugün yazılım sektöründe amatör ya da profesyonel mobil uygulamalar geliştirmek amacıyla kullanılabilecek çok sayıda platform mevcuttur. Söz konusu uygulama geliştirme platformlarının seçiminde çeşitli ölçütlere dikkat edilmesi gerekmektedir. Bunlar; mobil cihazların işletim sistemi (Android, IOS, Microsoft vb.), platformda kullanılan yazılım geliştirme dili (C, Swift, Java vb.), platformun çalışma şekli (çevrimiçi veya çevrimdışı), platformun ücretli ya da ücretsiz olması gibi sıralanabilmektedir. Bu tezde mobil uygulama geliştirmekte kullanılan elliden fazla çerçeve, ondan fazla geliştirme platformu ve editör incelenmiş, öğrenme olanakları, süreleri, avantajları ve dezavantajları karşılaştırılmıştır. Ayrıca karşılaştırılması yapılan platformlarla başlangıç seviyesinde uygulama tasarımları yapılarak mobil programlamaya başlayacak kişilere yol gösterici bir kaynak oluşturulmaya çalışılmıştır.

2019, xi + 150 sayfa

Anahtar Kelimeler: Mobil uygulama, Mobil platformlar, Native uygulamalar, Mobil responsive web sitesi, IOS, Android, Mobil uygulama, Programlama

ABSTRACT
M.Sc. Thesis

**COMPARISON THE PLATFORMS AND LANGUAGES USED TO DEVELOP
MOBILE APPLICATIONS FOR IOS OR ANDROID DEVICES.**

Hilmi BİLİCİ

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Internet and Information Technology Management

Supervisor: Asst. Prof. Said Mahmut ÇINAR

Nowadays, mobile devices are widely used. These devices use programs called mobile applications developed for different operations. Today, there are many platforms in the software industry that can be used to develop amateur or professional mobile applications. Various criteria need to be considered in the selection of these application development platforms. These; operating systems of mobile devices (Android, IOS, Microsoft, etc.), the software development language used on the platform (C, Swift, Java, etc.), the way the platform works (online or offline), whether the platform is paid or free. In this thesis, more than fifty frameworks used in developing mobile applications, more than ten development platforms and editors are examined and learning opportunities, durations, advantages and disadvantages are compared. In addition, by making application designs at the beginning level using compared platforms it was tried to create a resource to guide people who starts mobile programming.

2019, xi + 150 pages

Keywords: Mobile Application, Mobile Platform, Native Application, Mobile Responsive Web Sites, IOS, Android, Mobil application, Programming

TEŞEKKÜR

Bu araştırmanın konusu, çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Dr. Öğr. Üyesi Said Mahmut ÇINAR'a, araştırma ve yazım süresince yardımlarını esirgemeyen, her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma teşekkür ederim. Bu araştırma boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Hilmi BİLİCİ
AFYONKARAHİSAR, 2019



İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ.....	xi
1. GİRİŞ	1
1.1 Tezin Amacı.....	2
2. LİTERATÜR BİLGİLERİ	3
2.1 Mobil Cihazlar	3
2.1.1 Mobil Teknoloji Dünyası ve Tarihçe	4
2.1.2 Mobil İşletim Sistemleri	5
2.2 Mobil Cihazlar İçin Yazılım	9
2.2.1 Mobil Cihazlara Uygulama Geliştirmek Kullanılan İçin Yazılım Dilleri ...	10
2.2.2 Nesne Yönelimli Programlama (Object Oriented Programming/OOP)	15
2.3 Literatür	16
3. MATERYAL ve METOT	18
3.1 Mobil Uygulama Geliştirme Platformları.....	18
3.1.1 Mobil Web Uygulaması Geliştirme:	19
3.1.2 Native (Yerel) Uygulama Geliştirme:	19
3.1.3 Hybrid (Melez) Uygulama Geliştirme:.....	19
3.2 Platform ve Programlama Dili Seçimi.....	21
3.2.1 Android Programlama	21
3.3. Çerçeveler	35
3.3.1 NativeScript.....	36
3.3.2 Flutter	37
3.3.3 Cordova	38
3.3.4 Appcelerator Titanium.....	40
3.3.5 Ionic	41
3.3.6 Alpha Anywhere.....	43
3.3.7 Xamarin	43

3.3.8 PhoneGap	45
3.3.9 Sencha Touch	48
3.3.10 jQueryMobile	49
3.3.11 IBM Worklight	51
3.3.12 Corona	53
3.3.13 ENYO	54
3.3.14 LUNGO	55
3.3.15 React Native	55
3.3.16 Kivy	57
3.3.17 JQTouch	58
3.3.18 Alpha Anywhere.....	58
3.3.19 Xojo	59
3.3.20 RhoMobile	60
3.3.21 Mobil Oyun Geliştirme Çerçeveleri	62
3.4 Çevrimiçi (Online) Mobil Uygulama Oluşturma Platformları	63
3.4.1 Appery.io	63
3.4.2 Mobileroadie.com.....	64
3.4.3 The AppBuilder	65
3.4.4 Good Barber	66
3.4.5 ShoutEm	67
3.4.6 Appy Pie	68
3.4.7 AppMachine	69
3.4.8 GameSalad.....	70
3.4.9 BiznessApps	71
3.4.10 AppMakr.....	72
3.4.11 App Press.....	73
3.4.12 EachScape.....	73
3.4.13 iBuildApp	74
3.4.14 QuickBase.....	74
3.4.15 Salesforce1 ve Lightning.....	75
3.5 Geliştirme Ortamları (IDE) ve Editörler.....	76
3.5.1 Eclipse	77
3.5.2 Android Studio	78
3.5.3 XCODE	78
3.5.4 Adobe Air	79
3.5.5 Visual Studio Code.....	79

3.5.6 Visual Studio .NET	80
3.5.7 MIT APP INVENTOR	82
3.6 Kütüphaneler ve Api Kavramı	89
3.6.1 Kütüphane Kavramı.....	89
3.6.2 Apiler	94
3.7 Mobil Uygulamalarda Veri Tabanı Kullanımı.....	95
3.7.1 SQLite.....	95
3.7.2 Realm DB	95
3.7.3 ORMLite.....	96
3.7.4 Berkeley DB	96
3.7.5 Mongo DB	97
3.8 Mobil Uygulamalarda Resim Kullanımı.....	97
4. BULGULAR	101
4.1 Mobil Uygulama Geliştiricilerinin Dikkat Etmesi Gereken Noktalar	105
4.1.1 Mobil Uygulama Geliştiricilerinin Dikkat Etmesi Gereken Noktalar.....	105
4.1.2 Mobil Geliştirme Platformlarının Temel Faydaları.....	107
5. TARTIŞMA, SONUÇ VE ÖNERİLER.....	108
6. KAYNAKLAR.....	109
ÖZGEÇMİŞ.....	112
EKLER	113

KISALTMALAR DİZİNİ

Kısaltmalar

API	Application Programming Interface- Uygulama programlama ara yüzü
ARC	Automatic Reference Counting – Otomatik Referans Sayacı
CLI	Common Line Interface - Komut Tabanlı Ara yüz
CSS	Cascading Style Sheets - Basamaklı Stil Şablonları
HTML	HyperText Markup Language - Zengin Metin İşaret Dili
HTTPS	Secure Hypertext Transfer Protocol - Güvenli zengin metin aktarım iletişim kuralı
IDE	Integrated Development Environment – Tümüleşik Geliştirme Ortamı
IoT	Internet of things- Nesnelerin interneti
JDK	Java Development Kit - Java Geliştirme Paketi
MBaaS	Mobile Backend As A Service – Üçüncü parti mobil arka uç servisi sağlayan bulut hizmeti
OOP	Object Oriented Programming - Nesne yönelimli programlama
ORM	Object Relational Mapping (Nesne-İlişkisel Eşleme)
SDK	Software Development Kit- Yazılım geliştirme kiti
SRGS	Speech Recognition Grammar Specification- Konuşma Tanıma Dilbilgisi Belirtimi
WWW	World Wide Web - Dünya Çapında Ağ
XML	Extensible Markup Language -Genişletilebilir İşaretleme Dili

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1 Geçmişten günümüze mobil cihazlar	4
Şekil 2.2 Mobil işletim sistemleri	5
Şekil 2.3 2012-2018 yılları arası mobil işletim sistemi kullanım grafiği	6
Şekil 2.4 Türkiye'de çok kullanılan işletim sistemleri	7
Şekil 2.5 Nesne yönelimli programlama	15
Şekil 3.1 WORA prensibi.....	23
Şekil 3.2 Kotlin ile Native uygulamalar yapma	25
Şekil 3.3 iOS işletim sistemleri kullanım oranları	28
Şekil 3.4 Swift dili ile yazılmış bazı popüler uygulamalar	30
Şekil 3.5 BlackBerry anahtar edinme sayfası.....	31
Şekil 3.6 Eclipse ile BlackBerry sistemine kod yazma	32
Şekil 3.7 BlackBerry için ilk uygulama	33
Şekil 3.8 Eclipse ve BlackBerry JDE eklentisi ile ilk programlama.....	34
Şekil 3.9 Tarayıcı üzerinde çalışan hali.....	41
Şekil 3.10 Ionic kişisel uygulamalarım ekranı	42
Şekil 3.11 Tasarım ara yüzü ile çalışma.....	42
Şekil 3.12 Xamarin çalışma prensibi.....	44
Şekil 3.13 Xamarin çalışma sistemi	44
Şekil 3.14 PhoneGap çalışma sistemi.....	46
Şekil 3.15 PhoneGap Yeni proje oluşturma ekranı	47
Şekil 3.16 PhoneGap uygulama çalıştırma	47
Şekil 3.17 JQuery nesneleri.....	50
Şekil 3.18 JQuery örnek uygulama çıktı ekranı	51
Şekil 3.19 IBM Worklight çalışma prensibi.....	52
Şekil 3.20 Platformlar arası mobil uygulama geliştirme çerçevesi React Native	55
Şekil 3.21 React native'in zamanla gösterdiği kullanım artışı	56
Şekil 3.22 React Native ve Flutter karşılaştırması	57
Şekil 3.23 Xojo proje oluşturma ekranı	60
Şekil 3.24 Xojo çalışma ekranı	60
Şekil 3.25 Appery.io ana sayfası	63
Şekil 3.26 Mobil Roadie ana sayfası	64

Şekil 3.27 The AppBuilder ana sayfası	65
Şekil 3.28 Good Barber ana sayfası	66
Şekil 3.29 Mobile App Maker anasayfası	67
Şekil 3.30 Appy Pie ana sayfası	68
Şekil 3.31 AppMachine ana sayfa ekranı	69
Şekil 3.32 GameSalad ana sayfası.....	70
Şekil 3.33 BiznessApps ana sayfası	71
Şekil 3.34 AppMakr ana sayfası	72
Şekil 3.35 EachScape ana sayfası.....	73
Şekil 3.36 iBuildApp ana sayfası	74
Şekil 3.37 QuickBase çalışma ekrnı	74
Şekil 3.38 Tümüleşik geliştirme ortamı(IDE) kavramı.....	76
Şekil 3.39 Visual Studio Code	79
Şekil 3.40 Evrensel Windows platfromu UWP.....	81
Şekil 3.41 Microsoft Visual Studio ekranı	81
Şekil 3.42 App Invertor ekranı	84
Şekil 3.43 Blocks ekranı.....	85
Şekil 3.44 App Inventor örnek tasarımlar	87
Şekil 3.45 App Inventor komut blokları.....	87
Şekil 3.46 Apk uzantılı dosya olarak kayıt etme.....	88
Şekil 3.47 Build.gradle modülü.....	90
Şekil 3.48 Rest API kavramı	94
Şekil 3.49 Mobil uygulamalar için kullanıla bilecek veri tabanları	95
Şekil 4.1 Mobile programlama süreci	105
Şekil 6.1 Chocolatey komut satırında kurulumu	113
Şekil 6.2 Node.Js kurulumu	114
Şekil 6.3 JDK kurulumu.....	114
Şekil 6.4 Android Studio Kurulumu	115
Şekil 6.5 Tns doctor	116
Şekil 6.6 QR kodu oluşumu	116
Şekil 6.7 Telefonda QR code tarama.....	117
Şekil 6.8 Emülatörden gelen myApp çıktısı.....	118
Şekil 6.9 Güncellenmiş myApp çıktısı.....	118
Şekil 6.10 Node Js yükleme ekranı	119

Şekil 6.11	Platfom kontrolleri	120
Şekil 6.12	Gereksinim kontrolleri	121
Şekil 6.13	Appcelerator indirme	121
Şekil 6.14	Appcelerator kurulum	122
Şekil 6.15	Axway studio proje başlatma ekranı	123
Şekil 6.16	İlk uygulamayı hazırlama.....	124
Şekil 6.17	Ionic komut tabanlı kurulumu başlatma.....	124
Şekil 6.18	Ionic çalışma ekranı	125
Şekil 6.19	.NET ile mobil uygulama geliştirme.....	126
Şekil 6.20	Android yapılandırma ekranı	126
Şekil 6.21	Uygulama telefona yüklenen live player ile canlı görünüm sağlanabilir....	127
Şekil 6.22	Android Xamarin uygulama başlatma ekranı	127
Şekil 6.23	Emulator ekranı.....	128
Şekil 6.24	Worklight Hybrid proje oluşturma ekranı.....	131
Şekil 6.25	Worklight Environment seçim penceresi	132
Şekil 6.26	Corona program penceresi	133
Şekil 6.27	Kurulumu başlatma	134
Şekil 6.28	Uygulama oluşturma	135
Şekil 6.29	Dizin içindeki dosyalama.....	135
Şekil 6.30	Expo kurulumu.....	137
Şekil 6.31	Karekod oluşturma	137
Şekil 6.32	React Native çalıştırmak için kurulacak telefon uygulaması.....	138
Şekil 6.33	Kod değişikliği ve güncelleme ekranı.....	139
Şekil 6.34	Python kurulum ekranı.....	140
Şekil 6.35	Eclipse kurulumu	141
Şekil 6.36	Eclipse kurulumunda IDE seçimi	141
Şekil 6.37	Yeni proje oluşturma.....	142
Şekil 6.38	Proje oluşturma sihirbazı	142
Şekil 6.39	İlk uygulama oluşturma	143
Şekil 6.40	Eclipse çalışma ekranı.....	143
Şekil 6.41	Yeni sınıf oluşturma.....	144
Şekil 6.42	Çalışma ekranı.....	144
Şekil 6.43	İlk projenin çalıştırılması	146

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1 Cihaz türüne göre dünya çapında cihaz gönderileri, 2016-2019(*).....	3
Çizelge 2.2 Web aramalarında çıkma frekansına göre programlama dilleri	14
Çizelge 3.1 Mobil platforma göre programlama dilleri	18
Çizelge 3.2 Native ve Hybrid mobil uygulamaların karşılaştırılması	20
Çizelge 3.3 Swift, Objective C karşılaştırması.....	30
Çizelge 3.4 24.12.2018 tarihinde Android Studio indirilebilir paketleri.....	78
Çizelge 3.5 Glide Picasso kütüphanelerinin karşılaştırılması	91
Çizelge 4.1 Çevrim dışı ve içi çerçevelerin karşılaştırılması	101
Çizelge 4.2 Ionic, Cordova ve PhoneGap ortamlarının karşılaştırılması*	102
Çizelge 4.3 Mobil işletim sistemleri kıyaslaması.....	103
Çizelge 4.4 Flutter ve React Native karşılaştırması	104

1. GİRİŞ

Bilişim Teknolojileri ve İletişim Kurumunun (BTK) 2017 yılı üçüncü çeyrek için yayınladığı “Türkiye Elektronik Haberleşme Sektörü 3 Aylık Pazar Verileri Raporu” na göre (İnt. Kyn.1) Türkiye’de yaklaşık %97,6 erişim oranına karşılık 62.992.758 4.5G abonesi ve 11.586.255 3G abonesi ile toplam 77.882.845 toplam mobil abone sayısı bulunmaktadır. 3G ve 4.5G hizmeti kullanan abonelerin 2017 Aralık ayı itibarıyla %97’sinin akıllı telefonlara sahip oldukları tahmin edilmektedir. İnsanlar iletişimi, web sayfalarını ziyaret etmeyi, alışveriş işlemlerini, eğlenceyi kısaca bilgiye erişimi bilgisayarlardan çok mobil aygıtlar üzerinden kullanmaktadır. Mobil aygıtların taşınabilirliğinin kolaylaşması ve her an erişim imkânının bulunması gün geçtikçe kullanımlarını artırmıştır.

Mobil uygulamaların yaygın olarak kullanılmadığı dönemlerde her şey internet tabanlı olarak kullanılırken şimdilerde çevrim dışı uygulamalar da kullanılabilmektedir. En çok kullanılan mobil uygulama türlerine bakıldığında; Sosyal medya uygulamaları, Mobil iletişim uygulamaları, Arama motoru uygulamaları, Oyun uygulamaları, Diğer uygulamalar (Eğlence, Yaşam, Haber, Sağlık ve Spor, Fotoğraf, Seyahat) olduğu görülmektedir.

Yazılımcısını zengin eden hem dünya çapında hem de yerel anlamda birçok mobil yazılım olduğu görülmektedir. Örneğin mesajlaşma sistemine alternatif olarak hazırlanan Whatsapp uygulamasını yazan Amerika’ya göç eden Ukraynalı Jan Kourn’a milyar dolarlar kazandırmıştır. Yine Instagram uygulaması programcılarını uygulamayı sattıklarında programcılarını iki yüz ellışer milyon dolar para kazandırmıştır. Şuan da ki değeri on milyonları bulan biTaksi uygulaması kurucusu, sahibi Nazım Salur ve 40 metrekaire bir odada kurulan, uygulamayı yapan Melih Ödemiş, Nevzat Aydın, Gökhan Akan ve Cem Nufusi adlı programcılarını Mayıs 2015 yılında 589 milyon dolara satışı yaparak zengin eden yemeksepeti uygulamaları da yerel örnekler arasında göstermek mümkündür.

Ülkemizde 2018’de bir yılda indirilen mobil uygulama sayısının 400 milyona çıkması

beklendiđi düşünöldüğönde ve milyarlarca cihazın kullandığı bu işletim sistemleri için üretilen yazılımlardan hangisini kullanacağız ve nereden başlayacağız sorularını programcı olmak isteyenler için karşılaştırmalı olarak sunulacaktır.

Mobil yazılım tüketicilerinin uygulamalara harcadığı toplam tutar göz alındığında yüzde 32 Çinli, yüzde 22 Amerikan, yüzde 21 Japon şirketlere giderken, Türkiye merkezli şirketlerin tüketicilerin mobil uygulamalar için yaptığı toplam 101 milyar dolarlık harcama ile pay yüzde 1'i (1 milyar doları) bulduğu görölmektedir.

1.1 Tezin Amacı

Bu tezin amacı hızla gelişen ve hayatımıza giren mobil cihazlar için gerekli programlama platformları konusunda interaktif ortamlarda Türkçe kaynak oluşturmak ve Türk programcılarına hem çevrim içi hem de taşınabilir formatta sunmaktır. Geleceğın programlama dünyasında ölkemiz yazılımcılarının yerini almasını sağlamak ve yazılım konusunda eğiterek, mobil uygulamalar geliştirmelerine destek olmak bu alanlardaki istihdam kapasitesinin artışına yön vermek ve girişimciliğı artırmaktır.

IOS ve Android işletim sistemi kullanan cihazlar için uygulama geliştirme eğitimleri ile mobil teknoloji endüstrisinde çalışmak veya mobil teknolojileri kullanarak iş fikrini hayata geçirmek isteyenler için gerekli platformları ve yazılım dillerini tanıtmak tezin asıl amaçları arasındadır.

Mobil uygulamalar son yıllarda hızla gelişerek milyonlarca kişinin hayatının merkezine yerleşti. Günlük hayatı kolaylaştıran, iş sektörlerinde kullanımı ile çeşitli kolaylıklar sağlayan, sosyal paylaşım sitelerine her an bağılı olabilme imkânı sunarak çevrim içi haberleri anında almamızı sağlayan ve yeri geldiğinde boş zamanımızı daha eğlenceli bir şekilde geçirmeyi sağlayan mobil uygulamalar sayesinde teknolojiye daha fazla ihtiyaç duyacağımız düşünölerek mobil uygulama geliştirmek isteyenleri doğru adımı atmalarını sağlamak hedeflenen amaç arasındadır.

2. LİTERATÜR BİLGİLERİ

Mobil cihazlara uygulama geliştirme kavramında öncelikle Mobil cihazları, kullanılan işletim sistemlerini ve hangi yazılım dillerinin kullanıldığını bilmek gereklidir.

2.1 Mobil Cihazlar

Mobil cihazlar - akıllı telefonlar, tabletler, e-kitaplar, telefonlar, PDA'lar ve netbook'ları içeren, temel özelliği büyüklükleri ve çeşitli işlevleri gerçekleştiren bir dizi cihazlardır. Akıllı telefonlar - önemli bir özelliği büyüklük ve taşıma kabiliyetinin yanı sıra çok sayıda işlevselliğe sahip olan cihazlardır. Tabletler daha geniş bir ekrana sahiptir ve interneti, kitapları, ofis programlarını ve oyunları kullanmaya olanak sağlar.

Çizelge 2.1 Cihaz türüne göre dünya çapında cihaz gönderileri, 2016-2019(*)

Aygıt Tipi	2016	2017	2018	2019
PC (Dizüstü ve Masa üstü bilgisayar)	220(**)	204	193	187
Ultramobile (Kaliteli)	50	59	70	80
Toplam PC	270	262	264	267
Ultramobile (Basit)	169	160	159	156
Bilgi İşlem Cihazı Piyasası	439	423	423	423
Telefonlar	1,893	1,855	1,903	1,924
Toplam Aygıt	2,332	2,278	2,326	2,347

(*) Kaynak: Worldwide Device Shipments by Device Type - Gartner (January 2018)

(**)Milyon birim

Çizelge 2.1 'de görüleceği gibi Gartner (2018 yılında yaptığı değerlendirmede) 2019 yılına kadar mobil uygulama pazarında tüm şirketlerin %50'sinden fazlasının mobil uygulamalarına sahip olacaklarını öngörmektedir.

2.1.1 Mobil Teknoloji Dünyası ve Tarihçe

Mobil cihazların geçmişi oldukça eskilere dayanıyor. Eğer ilk tek yönlü iletişim kurabilen cihazı da katarsak tarih 1921'leri bulur. IBM firması tarafından cep telefonu ve bilgisayarı birleştiren ilk cihaz ise 1993 yılında Simon telefonudur. Günümüze kadar birçok farklı firma tarafından üretilmeye başlayan ve akıllı cihaz sektörü adı verilen PDA (Kişisel cep bilgisayarı) 'lar hayatımızın her alanı içerisine girmiştir.



Şekil 2.1 Geçmişten günümüze mobil cihazlar

Gelişen teknolojiye en büyük payı mobil cihazlar almıştır. Bebeklerin ellerine kadar düşen bu mobil cihazların kullanımı hızla ilerlemektedir. Her alanda bilişim teknolojileri ile her geçen gün artan internet kullanımı mobil yazılım alanında kendini göstermeye başlamıştır (Şekil 2.1).

2018'in 2.çeyreğinde İnternet, sosyal medya ve mobil kullanıcı istatistiklerine baktığımızda 4.08 milyar internet kullanıcısı, dünya nüfusunun %54'ünü, 3.29 milyar sosyal medya kullanıcısı, dünya nüfusunun %43' ü, 5.06 milyar mobil kullanıcısı, dünya nüfusunun %66' sını, 3.08 milyar mobil sosyal medya kullanıcısı, dünya nüfusunun %41'ini ifade etmekte olduğunu görürüz. (İnt. Kyn.2)

Apple iPhone cihazını 2007 yılında duyurması ile mobil cihazlar pazarında yeni gelişmeler gözlenmiştir. Bu cihaz ile Apple firması ürünün cep telefonu olması dışında, hem iPod hem de internet özelliğini birleştiriyordu. Öncesinde daha yavaş çalışan mobil cihazlar, bu yenilikten sonra yüksek çözünürlüklü dokunmatik ekranlara, güçlü işlemci ve ram yapısına, hızlı internet ağlarına (2G, GPRS, EDGE, 3G, HSPA) kavuştular. Web sayfalarını ve uygulamaları akıllı telefonlar üzerinden tam ekran olacak şekilde ve hızlı görüntülemek, iki parmakla büyütme ve kaydırma yapma işlemi mümkün hale geldi. İlk defa App Store, Haziran 2008’de açıldığında sadece 500 Adet uygulama vardı. Ekim 2011’de bu sayı yarım milyonun üzerinde bir sayıya ulaşırken 2017 yılının başında 2,2 milyon uygulama vardı. Ve halen günlük bini üzerinde yeni uygulama ekleniyor.

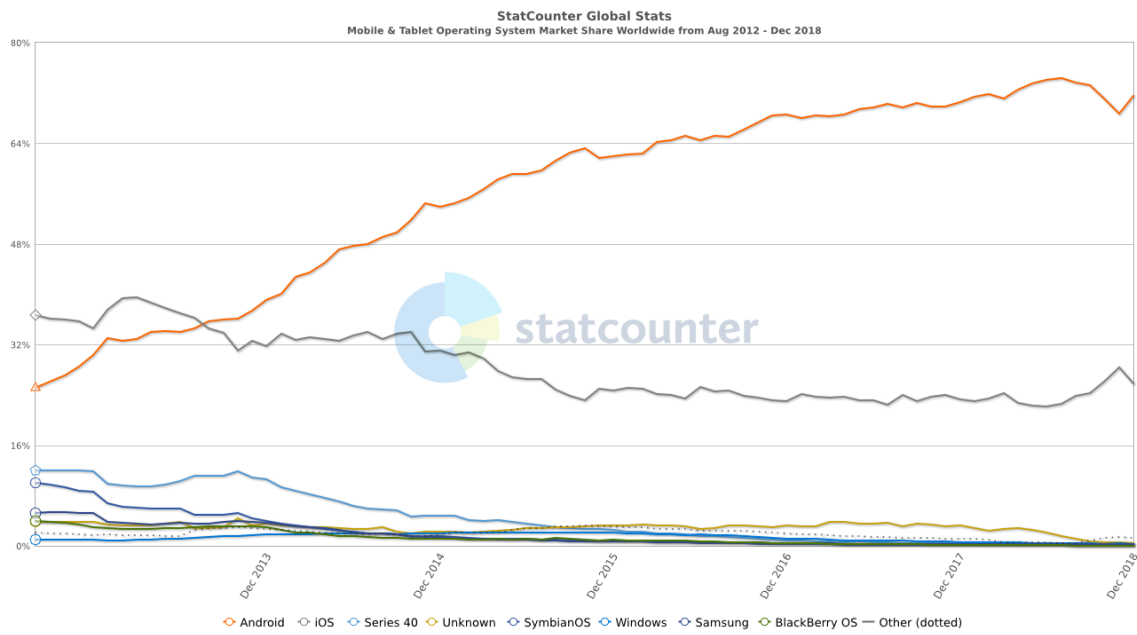
2.1.2 Mobil İşletim Sistemleri



Şekil 2.2 Mobil işletim sistemleri

İşletim sistemi, donanımın doğrudan kontrolü ve yönetiminden, tüm sistem işlemlerinden ve uygulama programlarını çalıştırmaktan sorumlu olan en temel programdır. Donanımlar hızla gelişip karmaşık bir hal alırken bu donanımların ne şekilde çalışacaklarını, kaynak yönetimini düzenleyen ve arada köprü olan yazılımlar işletim sistemleridir. Tablet bilgisayarlar, akıllı telefonlar, PDA’lar gibi taşınabilir aygıtlar için özel olarak hazırlanan işletim sistemlerine mobil işletim sistemleri adı

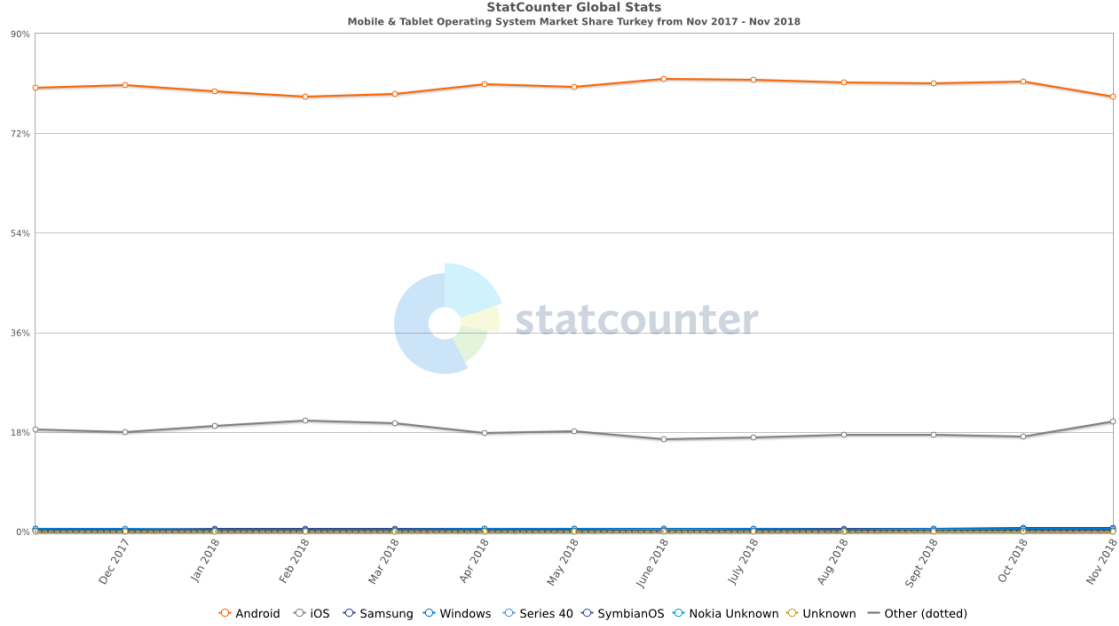
verilir. Şu an en çok kullanılan ve popüler olan Şekil 2.2’ de logoları görülen-mobil işletim sistemleri Android, iOS ve Blackberry gibi işletim sistemleridir. Nokia akıllı cihaz piyasasında tutunamasa da Windows Phone 8 işletim sistemi ve diğer mobil cihaz üreticileri de yeni ürünler çıkarmaya devam ediyor. Kısa zaman önce Linux dünyasından tanınan Ubuntu da telefonlara ve televizyonlara kullanılmaya başlandı. Bunlardan başka ismini çokça duyduğumuz, kullandığımız işletim sistemleri de oldu ancak mobil cihaz sektöründe ki hızlı değişime ayak uyduramadıklarından yok olmaya başladı. Bunlardan en bilineni Nokia telefonlarda kullanılan Symbian işletim sistemidir.



Şekil 2.3 2012-2018 yılları arası Mobil işletim sistemi kullanım grafiği

Genel olarak Şekil 2.3’e bakıldığında Symbian, Windows Mobile – Microsoft, MeeGo, Bada – Samsung, webOS, Maemo, Palm OS, Verdict gibi mobil işletim sistemleri unutulmuş ya da az tercih edilen mobil işletim sistemlerindendir. Fakat iOS, Android ve Blackberry piyasa da en çok tercih edilen mobil işletim sistemi arasındadır. Türkiye’de mobil işletim sistemi kullanım oranına (Şekil 2.4) bakıldığında ise Android: %78.65 iOS: %20.12 Windows Mobile: %0.34 Samsung: %0.45 Series40: %0.18 Symbian: %0.1 diğer: %0.04 olduğu görülmektedir. (İnt. Kyn.3). Mobil işletim sistemleri bilgisayarlar için tasarlanan işletim sistemleri gibi farklı özelliklere ve yapılar sahip olabilirler. Farklı geliştiriciler tarafından üretilen bu yazılımlar bazen açık kaynak kodluyken bazen de kapalı kaynak koduyla kullanıcıya sunulmaktadır. Türkiye

piyajasına henüz girmemiş olsa da Firefox OS işletim sistemi de yurt dışında 100 dolar fiyatı altında ki cihazlar için tercih edilmektedir.



Şekil 2.4 Türkiye'de çok kullanılan işletim sistemleri

Mobil işletim sistemleri kısaca:

- **Android İşletim Sistemi:**

Android işletim sistemi Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilerek 23 Eylül 2008'te piyasaya sürülen yazılımdır. Akıllı telefonlar ve tabletler gibi dokunmatik ekranlı mobil cihazlar için Linux tabanlı olarak özgür ve ücretsiz dağıtılan bir işletim sistemidir. Sistem açık kaynak kodlu olsa da, kodlarının ufak ama çok önemli bir kısmı Google şirketi tarafında saklıdır. C++, C ve Java dili ile kodlanmıştır.

- **Apple IOS İşletim Sistemi:**

iOS Apple'ın başlangıçta iPhone için geliştirdiği ancak daha sonra iPod touch ve iPad'de de kullandığı ve 29 Haziran 2007'de piyasaya sunduğu mobil işletim sistemidir. C++,

Objective-C ve Swift dili ile yazılmıştır.

- **Windows Phone İşletim Sistemi:**

Microsoft firması tarafından geliştirilmiş ve 21 Ekim 2010'da piyasaya sürülmüş olan işletim sistemi Windows Mobile serilerinin devamı niteliğinde bir mobil işletim sistemidir. C++ programlama dili ile yazılmıştır.

- **RIM BlackBerry İşletim Sistemi:**

BlackBerry marka akıllı telefonlar için geliştirilerek 1999 yılında piyasaya sürülen mobil işletim sistemidir. C++ programlama dili ile yazılmıştır.

- **MeeGo İşletim Sistemi:**

Intel ve Nokia tarafından geliştirilmiş, Mayıs 2010 tarihinde piyasaya sürülmüş bir açık kaynaklı mobil işletim sistemi projesidir. MeeGo işletim sistemi öncelikli olarak akıllı telefonlar, navigasyon cihazları, akıllı televizyonlar, giriş seviyesi masaüstü bilgisayarlar, Netbooklar, Laptoplar, tabletler ve PDA (Kişisel cep bilgisayarları)' ları hedeflemiştir. C++ programlama dili ile yazılmıştır.

- **WebOS İşletim Sistemi:**

LG webOS, Linux çekirdeği üzerinde çalışan ve ilk olarak 2009 yılında piyasaya sürülen bir mobil işletim sistemidir. Palm tarafından geliştirildi ve 2013 yılında LG firması tarafından satın alındı. C++, Qt programlama dilleri ile yazılmıştır.

- **Maemo İşletim Sistemi:**

Maemo, Nokia'nın akıllı telefonlar ve tablet bilgisayarlar için geliştirerek 2009 yılında piyasaya sürdüğü açık kaynaklı ve Debian tabanlı bir mobil işletim sistemidir. C++, Mono, C# programlama dilleri ile yazılmıştır.

- **Firefox OS İşletim Sistemi:**

Mozilla tarafından telefonlar için geliştirilerek 2019 yılında piyasaya sürülen Linux tabanlı açık kaynak bir işletim sistemidir. HTML5, CSS, JavaScript, C++ programlama dilleri ile yazılmıştır.

- **kaiOS İşletim Sistemi:**

Google tarafından desteklenen ve piyasaya sunum tarihi 2017 olan web tabanlı mobil işletim sistemidir. CSS (Cascading Style Sheets), JavaScript programlama dilleri ile yazılmıştır.

2.2 Mobil Cihazlar İçin Yazılım

İnsanlar günümüzde ortalama olarak 2-5 saatlerini mobil cihazlar ile internette vakit geçiriyor ve bir insan günde 150-500 defa arasında telefonunu kontrol etme ihtiyacı hissediyor. 2009 yılında web site trafiklerinin mobil cihaz payı %1 bile değilken, 2018 yılı itibarıyla web site trafiklerinin yarıya akını mobil cihazlar üzerinden sağlanmaktadır. Mobil cihazlar ile internet kullanım oranı artarak devam ediyor. Cep telefonlarında harcanan zamanın %80'i ise mobil uygulamalarda geçiyor.(İnt. Kyn. 4) Yine istatistiklere göre mobil cihaz kullanıcılarının %79'u ise ürün siparişi için cihazlarını kullanıyor. Bu veriler göz önünde bulundurulduğunda günümüz mobil uygulamaları her ölçekte girişimci için önem kazanıyor. Ticari açıdan bakıldığında web sitelerine oranla mobil uygulamalar, müşterilerine daha yakın duruyor. Bunun farkına varan ticari firmalar, kamu mobil dünyaya daha fazla yatırım yapmaya başlamışlardır. Çünkü müşterilerine daha iyi bir hizmet sunmak ve sektörde farklılık yaratarak yerini sağlama almak isteyen herkes mobil uygulamalar sayesinde daha geniş bir kitleye hitap etmesini ve mobil yazılım bildirimleri ile siteden daha fazla kişinin haberdar olmasını hedeflemektedir. Bu şekilde gerçek kullanıcılara erişim imkânı da sağlanmaktadır.

2.2.1 Mobil Cihazlara Uygulama Geliştirmek Kullanılan İçin Yazılım Dilleri

Günümüzde mobil platformlara hemen her dil ile uygulama yazılabiliyor. Neredeyse yazılamayan dil yok gibi. Mesela Python ile kivy uygulama çatısı kullanarak yazıldığı gibi Delphi ile ve Delphi'nin açık kaynak sürümü olan Lazarus ile bile yazılabiliyor. Tabi bu saydıklarımız en uç ve sıra dışı örneklerdir. Normalde Android için Java ve Kotlin en çok tercih edilenler arasında iken; IOS için ise Objective C ile Swift tercih edilen dillerin en üstündedir. C/C++, C# ile de ya da ismini daha hiç duymadığınız bir dil ile de yazmak mümkündür. Yeni çıkan Ruby, Crystal gibi diller buna örnek verilebilir.

İstatistiklere göre (Çizelge 2.2) mobil yazılım geliştirme dillerinde en çok tutulanları ise şunlardır:

- C#
- C/C++
- Java
- Objective C
- Swift
- HTML
- CSS
- Javascript
- Kotlin
- Python
- Ruby
- Dart
- Delphi

C# Dilinde Mobil Programlama

Microsoft firmasının geliřtirmiş olduđu **C# (C Sharp)**, yeni nesil bir programlama dilidir. .NET Teknolojisi için geliřtirilmiş ve nesne tabanlı dillerin arasından en önemli olanlardandır. Java ile birçok ortak noktası vardır. Hata oranını en aza indirmesinin bir sebebi de gelişmiş bir derleyiciye sahip olmasıdır. Hazırlanan uygulama çalıştırıldıktan sonra derleyici tarafından tespit edilen hatalar geliřtiricinin önüne farklı bir ekranda sunulur ve geliřtirici bu hataları kolaylıkla düzeltme imkânı bulur.

C++ Dilinde Mobil Programlama

1979 yılından itibaren düşük seviyeli bir dil olarak geliřtirilen C++, mobil uygulama dillerinden en popülerlerinden birisi olmuřtur. Dil C dilinin nesne tabanlı türü olarak tanımlanabilir. C++, pek çok mobil uygulamada ve işletim sisteminin kodlanmasında kullanılmaktadır. Nesne tabanlı programlamanın kolaylıklarını sunan **C++** sayesinde geliřtirilen mobil uygulamalar, uygun iyileřtirmeler sonucunda oldukça hızlı bir şekilde derlenir ve çalışır.

Java Dilinde Mobil Programlama

Sun Microsystems tarafından geliřtirilen, güçlü bir şekilde yazılmış nesne yönelimli programlama dili olan Java; uygulama geliřtiricileri tarafından “taşınabilir programlama dili” olarak adlandırılır. **Java** ile kodlanmış programlar; iOS, Linux, Mac OS, Windows, Android gibi farklı işletim sistemlerinde pek fazla bir deęişikliğe tabi olmadan çalıştırılır.

Objective-C Dilinde Mobil Programlama

Apple tarafından kullanılan, C dili ve Smalltalk paradigmalarına (değerler dizisine) dayanan derlenmiş nesne yönelimli bir programlama dili olan **Objective-C** iPhone ve iPad’lerde kullanılabilecek mobil uygulamaların geliřtirilmesini sağlar. Nextstep firması tarafından üretilen ve iOS işletim sistemleri için uygulama geliřtirilebilen nesne tabanlı

bir yapıya sahiptir. Ücretsizdir fakat güncel sürümlerini çalıştırabilmek için Apple marka bir bilgisayara ihtiyaç vardır. Apple marka cihazı olmayanlar sanallaştırıcı makineleri kullanarak Mac işletim sistemi kurabilir ve Apple yazılım dünyasına adım atabilir.

Swift Dilinde Mobil Programlama

iOS işletim sistemiyle uyumlu mobil uygulamalar geliştirebilmek için, Apple tarafından oluşturulan nesne tabanlı bir programlama dilidir. Kullanımı oldukça kolay olan **Swift**'in Açık kaynaklı bir dil olması da kullanımını artırmıştır.

HTML Dilinde Mobil Programlama

Web sayfa hazırlama dili olarak bilinen **HTML**, günümüzde kullanılan en yaygın metin tabanlı dildir. En son sürümü HTML5 ile responsive web sayfaları tasarlamamanın yanında mobil uygulamalarda yazıla biliniyor.

CSS Dilinde Mobil Programlama

Gerçekte bir programlama dili olmayan **CSS**, mobil uygulamadaki HTML elementlerinin nasıl bir görsellikle sunulacağını düzenleyen bir işaretleme dilidir. Mobil uygulamalara ait olan nesneleri, renkleri, boyutları, konumları vb. özellikleri düzenlemek için kullanılıyor.

JavaScript Dilinde Mobil Programlama

Netscape firması tarafından geliştirilen **JavaScript**, etkileşimli ve dinamik bir betik dilidir. Java'nın farklı bir sürümü sanılsa da aslında birbirinden oldukça farklı teknolojilerdir. Genellikle oyun geliştirmede kullanılır.

Kotlin Dilinde Mobil Programlama

JVM (Java Virtual Machine) üzerinde çalışan ve JetBrains firması tarafından ortaya

ıkarılan Kotlin dili Google I/O 2017 etkinlięinde (Int. Kaynaęı 5) Java ve C++ ‘dan sonra Android’in 3. resmi geliřtirici dili olarak duyurulan ve aık kaynak koda sahip bir programlama dilidir.

Python Dilinde Mobil Programlama

Geliřtirilmesine 1990 yılında bařlayan Python, nesne ynelimli, yorumlamalı, birimsel ve etkileřimli yksek seviyeli bir programlama dilidir. Kivy modl ile Python dilinde mobil programlama yapma imknı saęlanmıřtır.

Ruby Dilinde Mobil Programlama

Ruby, dinamik, nesne tabanlı, reflektif yntemli st seviye bir programlama dilidir. Sz dizimi aısından Perl ve Eiffel’e benzeyen Ruby dili, Yukihiro Matsumoto tarafından Japonya’da tasarlanmaya ve geliřtirilmeye bařlanmış olup dilin bazı zellikleri Python, Lisp, Dylan ve Clu’dan alınmıřtır.

Dart Dilinde Mobil Programlama

Dart ilk kez Google firması tarafından geliřtirilerek JavaScript’in yerini almasını dřndę, 2011 yılında tanıtılan ok maksatlı ve nesne ynelimli, smalltalk tarzını benimsemiř bir programlama dilidir.

Delphi Dilinde Mobil Programlama

Pascal programlama dili zerine inřa edilen nesne ynelimli bir programlama dilidir. Delphi programlama alanında ok nemli bir yer sahipken kullanıcı kitlesini kaybetmiřtir. Fakat Delphi XE2 srm ile birlikte mobil desteęi sayesinde Android ve IOS yazılım pazarında olduęunu gstermiřtir.

Peki, hangi programlama dili seçilmelidir?

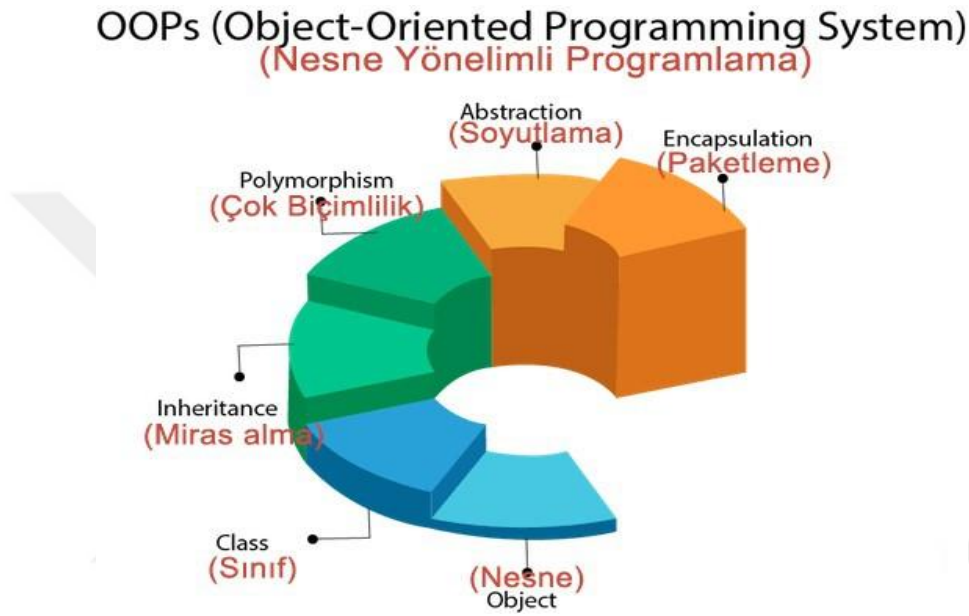
Hangi programlama dilinin seçilmesi ile ilgili kafalarda bir sürü oluşabilir. Burada tabii ki asıl önemli olan “hangi tür ve amaç için bir yazılım oluşturacak” gibi soruların cevabında yatıyor. TIOBE adlı kuruluşun programlama dillerinin Google, msn, yahoo, wikipedia'nın web aramalarında çıkma frekansına göre belirlendiği indekse göre Çizelge 2.2 de görülen oranlar karşımıza çıkmaktadır (İnt.Kyn. 9).

Çizelge 2.2 Web aramalarında çıkma frekansına göre programlama dilleri

Aralık 2018	Aralık 2017	Değişim	Programlama Dili	Reyting	Değişim
1	1		Java	15.932%	+2.66%
2	2		C	14.282%	+4.12%
3	4	↑	Python	8.376%	+4.60%
4	3	↓	C++	7.562%	+2.84%
5	7	↑	V. Basic .NET	7.127%	+4.66%
6	5	↓	C#	3.455%	+0.63%
7	6	↓	JavaScript	3.063%	+0.59%
8	9	↑	PHP	2.442%	+0.85%
9	-	↑↑	SQL	2.184%	+2.18%
10	12	↑	Objective-C	1.477%	-0.02%
11	16	↑↑	Delphi/Object Pascal	1.396%	+0.00%
12	13	↑	Assembly language	1.371%	-0.10%
13	10	↓	MATLAB	1.283%	-0.29%
14	11	↓	Swift	1.220%	-0.35%
15	17	↑	Go	1.189%	-0.20%
16	8	↓↓	R	1.111%	-0.80%
17	15	↓	Ruby	1.109%	-0.32%
18	14	↓↓	Perl	1.013%	-0.42%
19	20	↑	Visual Basic	0.979%	-0.37%
20	19	↓	PL/SQL	0.844%	-0.52%

2.2.2 Nesne Yönelimli Programlama (Object Oriented Programming/OOP)

1960'lı yılların sonunda ortaya çıkan Soyutlama (Abstraction), Sarmalama / Paketleme (Encapsulation), Miras Alma (Inheritance), Çok Biçimlilik (Polymorphism) yeteneklerini birlikte barındırarak kodlamada ki karmaşıklığı gideren yaklaşıma Nesne yönelimli Programlama dilleri denir.



Şekil 2.5 Nesne Yönelimli Programlama

Nesne tabanlı programlama olarak ta bilinen yaklaşım uygulamaları nesneler kullanarak oluşturmak demektir. Nesne tabanlı programlama temelinde sınıf (class) adlı kavram bulunur. Sınıflar, nesne üretmemizi sağlayan veri tipleridir (Şekil 2.6).

Avantajları:

- Karmaşık ve anlaşılmaz programlama yapılarını ortadan kaldırır.
- Normal bir işlemi çok daha hızlı ve kolay bir şekilde yapılır.
- Kod tekrar kullanabileceği için kod yazım maliyeti düşer.
- Genişletilebilirdir.
- Takım çalışmasına uygundur.
- Parçalara bölüldüğü için hata tespiti ve bakımı kolaydır.

2.3 Literatür

Her ne kadar internet ve arama motorlarından IOS, Android ve diğer mobil cihazlara yazılım hazırlama ile ilgili bilgiler bulunulacağı düşünülse de yeni başlayanlar için platform ve programlama seçiminde doğru karar veremedikleri ve ülkemizde mobil yazılımlar konusunda bir boşluk olduğu görülmektedir. Konunun bu derce önemine karşılık, şimdiye kadar kapsamlı bir karşılaştırma bibliyografyası hazırlanmamıştır. Bu tez ile mobil programlamaya başlayacak kişileri doğru yönlendirme amacı güdülmektedir.

Charland, A., ve LeRoux tarafından yazılan “Web yada Native Mobil uygulama geliştirme platformları” adlı makalede 9 ayrı mobil işletim sistemi hakkında yazılım hazırlamak için gerekli platformlar sunulmuş ve sadece JavaScript betik dilinin değerlendirilmesi yapılmıştır.

Klein tarafından hazırlanan “Web sitesi mi yoksa mobil uygulama mı yapmak” adlı çalışmada sadece taşınabilir aygıtlar için web sitesi mi yoksa mobil uygulama geliştirmenin artılarından, eksilerinden bahsedilmiştir.

Kadir Demir, Ercan Akpınar tarafından hazırlanan “Mobil öğrenmeye yönelik tutum ölçeği geliştirme çalışması” makalede lisans öğrencilerinin mobil öğrenmeye yönelik tutumlarını ölçebilecek bir ölçek geliştirilmiştir. Bu da Mobil cihazların eğitime olan katkısı hakkında bizi yönlendirmektedir.

Bu konuda en kapsamlı çalışmanın ise Keskin ve Kılınç tarafından mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırılmasının yapıldığı ve örnek uygulamaların verildiği çalışma olduğu görülmektedir. Fakat adı geçen makalede sadece mobil öğrenme uygulamalarını geliştirmeye yönelik bir çalışma yapılmıştır. Gezgin tarafından hazırlanan başka bir çalışmada mobil öğrenme uygulamalarının geliştirilmesinde kullanılan yöntemler ve platformların karşılaştırılmaları yapılmıştır.

Bu çalışma ise günümüze kadar mobil uygulama geliştirme için kullanılan programlama dillerini, hangi sistem için hangi platform ve dilin öğrenilmesi gerektiğini, uygulamaların hazırlanması için gerekli çevrim içi ve çevrim dışı olarak kullanılan platformların (framework) karşılaştırmasını, gerekli geliştirme ortamlarını, ek olarak mobil uygulamalarda kullanılan veri tabanlarını, resim kullanma özelliklerini ve mobil uygulama geliştiricilerinin dikkat etmesi gereken noktalar ve yayınlama konularını incelemektedir. Mobil cihazlar için kullanılan 9 işletim sistemi ve sistemlere programlama yapabilen 13 programlama dili, 7 IDE ve editör, 18 çerçeve, 15 çevrim içi çerçeve, 9 kütüphane ve 5 veritabanı incelenerek kurulumları, öğrenme olanakları ve süreleri, avantajları, dezavantajları, karşılaştırılmalı olarak yer almıştır.



3. MATERYAL ve METOT

Mobil endüstri, akıllı telefonlar ve tabletler gibi artan sayıda mobil cihazla hızla gelişmektedir. Günümüzde insanlar mobil cihazlardan her zamankinden daha fazla hizmet almaktadır. Mobil çözümler işletmeleri, kuruluşları ve artan müşteri ve iş ortaklarını doğrudan etkilemektedir.

3.1 Mobil Uygulama Geliştirme Platformları

Mobil uygulamalarda şu an için performans, görünüm ve etkileşim (kullanıcı dostu arayüzler), uygulama güvenliği, anlaşılabilirlik, stabilite, geliştirme, test ve bakım süreçleri, yeni geliştirmelere destek, kullanılacak platform ve eklenti desteği en önemli ölçütlerdir. İşletim sistemlerine göre kullanılan programlama dilleri Çizelge 3.1’de verilmiştir.

Çizelge 3.1 Mobil platforma göre programlama dilleri

Mobil Platformlar	Programlama Dilleri	Uygulama mağazası
Apple IOS	C, Objective C	Apple Store
Google Android	Java	Google Play Store
Windows Mobile	.NET, C#	Windows Store
Window 7 Phone	.NET, C#	Windows Store
Symbian	C, C++, Python, HTML/ CSS/JS	Nokia Store, Ovi store
RIM BlackBerry	Java (J2ME)	BlackBerry App World
HP Palm WebOS	HTML/CSS/JS	HP store
Samsung Bada	C, C++, HTML/CSS/JS	Samsung Apps
MeeGo	C++	Apps for MeeGo

Projenin yapısına göre uygulama geliştirme:

- 1- Mobil Web Uygulaması
- 2- Native (Yerel)
- 3- Hybrid (Melez)

3.1.1 Mobil Web Uygulaması Geliştirme:

Uygulamanın HTML5, CSS3, Javascript gibi web teknolojileri ile yazılarak tarayıcılarda çalışacak halde sunulması işlemidir. Mobil cihazların tarayıcıları ile siteye girdiğimiz zaman tarayıcı istemciye göre (user agent) mobil tarayıcı olduğunu anlayıp ona göre web sitesi olarak değil de web uygulaması olarak açılması sağlanır. Responsive web site tasarımı 'da tercih edilebilmesi hem daha pratik hem de daha az maliyetli olmaktadır. Farkı uygulama mağazalarından indirilemeyip tarayıcılar vasıtası ile erişilmesidir.

3.1.2 Native (Yerel) Uygulama Geliştirme:

Günümüzde var olan mobil aygıtları incelediğimizde Apple IOS, Google Android, RIM BlackBerry ve Windows Mobile gibi mobil işletim sistemlerinin kullanıldığını görmekteyiz. Bu sistemler için geliştirilen uygulamalar ayrı ayrı mağazalarından erişilebilmektedir. Mobil uygulamaların farklı özellikteki bu cihazlar için ayrı ayrı yazılmasına, yazılımsal ve donanımsal olanaklarına doğrudan erişmesine yerel (Native) uygulama geliştirme yöntemi denmektedir. Native uygulamalar cihazın kendi ana platformu tarafından desteklenmiş bir dil ile yazılır. Örneğin Java ile geliştirilen Android uygulamaları Objective C veya Swift ile geliştirilen iOS uygulamaları gibi.

3.1.3 Hybrid (Melez) Uygulama Geliştirme:

Tek kod ile yazılan ve birden fazla platformda (IOS, Android, Windows) çalışabilen uygulamalara Hybrid uygulamalar denir. Bu uygulamalar HTML5, CSS3 ve JavaScript'in gücü ve yaygınlığı üzerine kurularak mobil cihazlarda tercih edilmektedir. Native uygulamalar gibi sistem kaynaklarına doğrudan değil de bir ara katman sayesinde erişir. Çarpaz (cross-platform) uygulama geliştirme platformu olarakta bilinen bu geliştirme yönteminde, tüm platformlar için tek dil kullanarak native uygulamalar oluşturulur ve aynı sonuç üretilir.

Hybrid uygulama bir kez kod yazıp tüm platformlar için (iOS, Android, Windows)

native uygulama geliştirebilme isteğidir. Uygulama bir kez yazılacak ve tüm platformlar için uygulama çalışır hale gelecek. Uygulama ileride güncellenmek istenirse yine aynı kod üzerinden geliştirmeye devam edip, tek bir kod ile tüm platformlara native uygulama hazırlanmış olur. Geliştirilen bu uygulama App Store, Google Play gibi tüm uygulama mağazalarına sunulabilir. Detaylı karşılaştırma Çizelge 3.2’de verilmiştir.

Çizelge 3. 2 Native ve Hybrid mobil uygulamaların karşılaştırılması

Native Uygulamalar	Hybrid Uygulamalar
Platforma bağlı geliştirme	Platformlardan bağımsız geliştirme
Yüksek performans	Sınırlı performans
Geliştirme ve bakım maliyetlidir	Geliştirme ve bakım az maliyetlidir
Uygulama geliştirme karmaşık ve zaman alır	Hızlı geliştirme ve bakım
Her platformun kendine özgü dilini ve araçlarını öğrenmek zorundasın	HTML, CSS, JS bilmek yeterlidir
Geliştirmiş olduğu platformdaki API'lere ulaşır	Cordova API ile cihaz özelliklerine erişme kullanma imkânı sağlar.

Peki, hangisini öğrenmek ya da hangisi ile uygulamamızı geliştirmek daha doğrusu? Hangi programcıya sorulsa kendi bildiği teknolojiyi savunmakta ve daha avantajlı olduğunu anlatmaktadır. Mobil Web siteleri tam olarak bir mobil uygulama sayılmazlar. Mobil cihazların tarayıcıları üzerinden erişilen mobil uygulamalara benzetilmiş web site yapılarıdır. Zaten uygulama gibi indirilip kurulamazlar. Sadece tarayıcılar üzerinden erişilip cihaz özelliklerinin (sensör, kamera, veri tabanı) tamamını da kullanamazlar. İnternet bağlantısı olmayınca (çevrimdışı) zaten erişim olamaz. Bu yüzden Mobil web siteleri biraz daha konu dışındadır ve Mobil Web uygulaması yazmayı düşünenler daha çok Responsive site tasarımı üzerinde durmalıdır. Responsive site tasarımları sayesinde hem daha az maliyetli hemde web sitesinin tüm içeriğini sunan bir amaç hedeflenmiş olunur.

3.2 Platform ve Programlama Dili Seçimi

3.2.1 Android Programlama

Android tabanlı sistemler için yazılım geliştirme süreci:

- Java dosyalarında, programın mantığını - uygulamanın ne yapmasını istediği açıklanır.
- XML dosyalarında sayfa düzeni (mizanpaj) - görünümleri tasarlanır.
- Uygulama yazıldıktan sonra, tüm dosyaları derlemek ve bunları Android cihazlarda çalıştırabilecek ve / veya Google Play'de yayınlatabilecek bir .apk dosyasında bir araya getirmek için derleme aracını kullanmak gerekir.
- Entegre bir geliştirme ortamı (IDE) ile birlikte Android için bir uygulama oluşturmak için kullanılan tüm yardımcı programlar ve dosyalar. Bir IDE, kod dosyalarınızı düzenlemek, derlemek ve çalıştırmak için açtığınız bir programdır.
- Önceleri, Android geliştirme için standart IDE Eclipse kullanılır idi, ancak şimdi bir Google ürünü olan ve daha işlevsel bir Android Studio platformu kullanılıyor.

3.2.1.1 Java ile Android Programlama

Java Programlama dili ile Native olarak android uygulama geliştirmek mümkündür. Java ile programlama yaparken duyacağımız JVM, JRE ve JDK terimlerini bilmemiz gerekli.

JVM (Java Virtual Machine – Java Sanal Makinesi)

Java'nın en temel ve en cazip özelliği her işletim sisteminde çalışabilmesidir, bunu sağlayan yapı ise JVM'dir. Java platformunun bağımsızlığı temel olarak Java sanal makinesinden (JVM) oluşur. JVM nesne modülü formatı olan class uzantılı sınıf dosyalarının belleğe yüklenmesi, kontrol edilmesi ve çalıştırılması için gereken hizmetleri sunan sanal bir programdır. Temel görevi sınıf dosyalarını yorumlamaktır (İnt. Kyn. 10).

JRE (Java Runtime Environment- Java Çalışma Zamanı Ortamı)

Java teknolojisiyle yazılmış uygulamaları ve küçük uygulamaları (applet) çalıştıran bilgisayar kullanıcılarına gerekli ücretsiz temin edilen bir Java Sanal Makine uygulamasıdır. Java Runtime Environment, java programlarını çalıştırmak için Java sınıflarından oluşan bir kütüphaneye sahip bir uygulamadır.

JDK (Java Development Kit - Java Geliştirme Kiti)

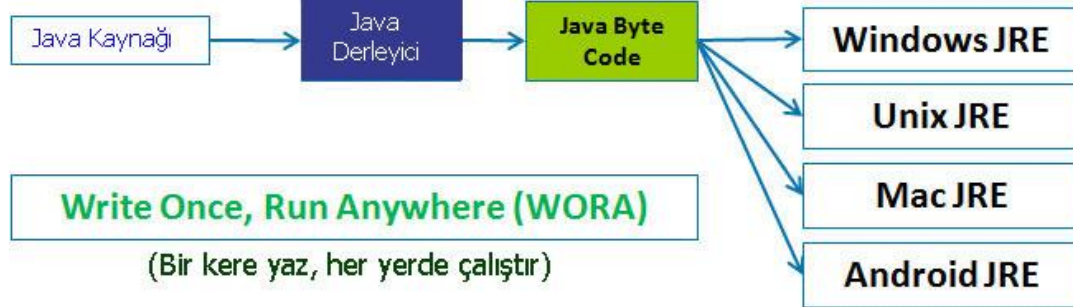
Java Geliştirme Kiti Java geliştiricilerine yönelik Oracle firması tarafından hazırlanmış bir üründür. Java'nın kullanıma sunulmasından beri en geniş kullanım alanına sahip Java geliştirme kitidir.

JDK; JRE, Java Compiler, Debug Tool (hata ayıklama) ve diğer gerekli araçlardan oluşur. Yazmış olduğumuz “.java” uzantılı kodumuzu derleyip JVM'in okuyabileceği “.class” uzantılı hale getirebilmemiz için **Java Geliştirme Kiti'ne** ihtiyaç duyarız. Java Geliştirme Paketi (JDK) Java kodunu bytecode'a çevirir, onu JVM'ye yükler ve farklı işletim sistemlerinde çalıştırır. Bunun yanında Java uygulamaları geliştirmek ve çalıştırmak için gerekli diğer tüm araçları da barındırır.

3.2.1.2 Java programlama dili

Java dili, açık kaynak kodlu, platformdan bağımsız, yüksek performanslı, çok işlevli, yüksek seviye bugüne kadar kullanılan en yaygın ve popüler programlama dillerinden biridir. Dilin ilk versiyonu 1996 yılında Sun Microsystems'ın mühendisi James Gosling tarafından yazılmıştır ve ardından Oracle tarafından desteklenmiştir. Java, çeşitli görevler için kullanılacak evrensel bir programlama dili olarak tasarlanmıştır. Ve şimdi Java dili uzun bir yol katederek, birçok farklı sürümü yayınlanmıştır En son sürümü, Eylül 2018'de yayımlanan Java 11'dir. Java, basit bir evrensel dilden bütün bir platforma ve çeşitli görevlerde kullanılan çeşitli teknolojileri birleştiren bir ekosisteme gelişti: masaüstü uygulamaları oluşturmaktan büyük web portalları ve servisleri yazmaya kadar. Ayrıca, Java dili sıradan bilgisayarlar, tabletler, akıllı telefonlar ve cep

telefonları ve hatta ev aletleri için yazılım oluşturmada kullanılır. Android mobil işletim sisteminin popülerliği de dilin gelişiminde katkısı olmuştur.



Şekil 3.1 WORA prensibi

Şekil 3.1’de belirtildiği gibi Java, WORA (**W**rite **O**nce, **R**un **A**n anywhere: Bir kere yaz, her yerde çalıştır) prensibi ile çalışır. Hangi işletim sistemi veya donanımın mevcut olduğuna bakılmaksızın, JVM, programların içinde çalışması için öngörülebilir bir ortam oluşturur.

Bir Java yazılımı geliştirilmesinde Programcı Java kodunu yazar (dosya .java uzatısı ile kaydedilir. Bu kod bir Java derleyicisi ile derlenir. Sonuçta byte code adı verilen (.class uzantılı byte code) bir tür makine kodu oluşur. Platform bağımsızlığını sağlayan şey byte code’dur. Çünkü bir kere bytecode oluştuktan sonra yazılım sanal makine içeren tüm işletim sistemlerinde sorunsuz çalışır. Bu “byte code” Java Sanal Makinesi tarafından satı satır yürütülür. Bu byte kod taşınabilme özeliğine sahiptir.

Bazı java aileleri ve yapabilecekleri aşağıda sıralanmıştır (İnt.Kyn.8):

j2se (JAVA to Standard Edition): Java’nın standart sürümüdür ve en temel fonksiyon ve özelliklerini içerir. Genelde java programlama dili denilince kastedilen API’nin ismidir.

j2ee (JAVA to Enterprise Edition): Java’nın gelişmiş sürümüdür. Genelde web programlamayı içeren JSP gibi eklentiler bu paketin içerisindedir.

j2me (JAVA to micro edition): Java’nın küçültülmüş sürümüdür. Standart sürümü olan

J2SE'ye göre daha hafif ve daha hızlı olarak tasarlanmıştır. Burada tam bir alt küme olarak düşünmek ve j2se'de bulunan herşeyin j2me'de de olacağını sanmak bir hatadır. J2ME içerisinde programlanan ortama göre (örneğin cep telefonu) özel olarak kullanılan ilave sınıf ve kütüphaneler bulunmaktadır.

j2card (JAVA to card): JAVA'nın akıllı kartlar (smart cards) için geliştirilen paketinin ismidir.

```
public class Program
{
    public static void main (String args[])
    {
        System.out.println("Hello Java!");
    }
}
```

Java nesne yönelimli bir dildir, bu nedenle tüm program etkileşimli sınıflar kümesi olarak temsil edilir. Bu durumda, bir Program sınıfı tanımlanır. Bir sınıf tanımlarken, önce bu erişimin herkes tarafından erişilebilir olacağını belirten bir genel erişim değiştiricisi tanımlanır, yani o komut satırından başlatılabilir. Daha sonra sınıf anahtar kelimesi ve daha sonra sınıf adı gelir. Yani, sınıf Program olarak adlandırılır. Kıvrımlı parantez içinde adından sonra sınıfın içeriğidir.

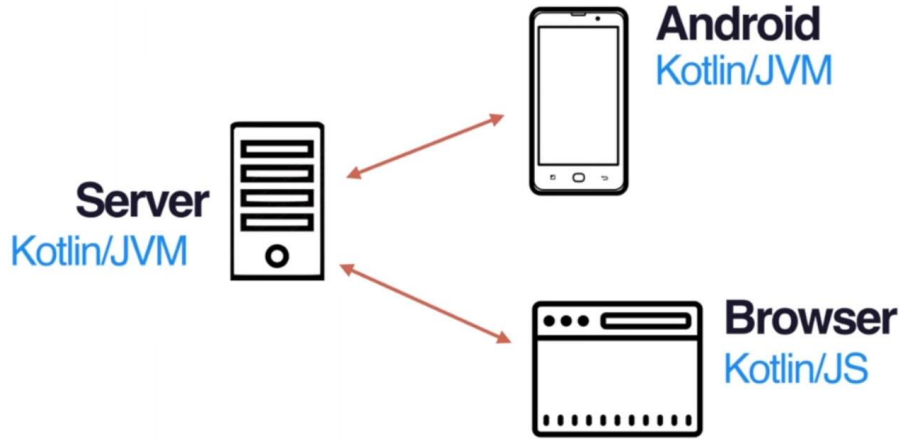
3.2.1.3 Kotlin ile Android Programlama

Kotlin, 2010 yılında Rus şirketi JetBrains'ten tarafından hazırlanmaya başlanan genç bir dildir. 2011 yılında “JVM Language Summit” etkinliğinde duyurulmuştur. Google I / O 2017 konferansında, Android geliştirme ekibi, Kotlin'in Android uygulamalarının geliştirilmesi için resmi destek aldığını bildirdi. Proje açık kaynak kodludur. Github adresini ziyaret ederek projeye destek olabilir veya inceleyebilirsiniz. Adı Rusya'da bulunan Kotlin adasından gelmektedir. “Java varken neden böyle bir dil geliştirdiniz?” sorusuna firma “daha özlü kodlar yazmayı sağlamak için” demiştir. Yani Kotlin ile Java'ya göre daha kısa ve daha özgün şekilde kodlama yapılabilmektedir. Kotlin programlama dili ile JVM uygulamaları (Server-side Uygulamalar), Android Uygulamaları, Browser uygulamaları (JavaScript tabanlı Web Uygulamaları), Native Uygulamalar (MacOS, iOS ve Gömülü sistemler Uygulamaları) geliştirilebilir (Şekil

3.5). Java, C ve C ++ gibi, Kotlin statik olarak yazılmış bir dildir. Hem nesne yönelimli hem de prosedürel programlamayı destekler. Yukarıda belirtilen dillere benzer şekilde, bir Kotlin programının ana kodu, bir main komut satırı argüman dizisinden geçirilen bir işleve yazılır:

```
package merhaba  
  
fun main(args: Array < String > ) {  
    val scope = "Dünya"  
    println("Merhaba, $scope!") // noktalı virgöl isteğe bağlıdır  
}
```

Kotlin/Anywhere



Kotlin ile Sunucu taraflı, Android için, Tarayıcı uygulaması ve Native uygulamalar yazmak mümkündür.

Şekil 3.2 Kotlin ile Native uygulamalar yapma

Kotlin ve Java-Android İlişkisi

Kotlin'in başlıca özellikleri ve faydaları şunlardır:

- JVM bayt kodu veya JavaScript ile derlenir.
- Mevcut tüm Java çerçevelerini ve kitaplıklarını kullanabilir.
- Kotlin'de null hatası almak zordur.
- Dil öğrenmek çok kolaydır.
- Kaynak kodu açıktır.
- Doğru işlev türleri vardır.

- Null-safety, Smart casts, String templates, Properties, Primary constructors, Range, Operator overloading, Data Classes gibi boş güvenlik ve veri sınıfları Kotlin de olan Java da olmayan özellikler arasındadır.
- Kotlin’ de akıllı tip dönüşümleri vardır.
- Kotlin’ de her şey bir nesnedir.
- Sunucu (Server) ve istemci (Client) tabanlı web uygulamaları geliştirmeyi desteklemektedir.
- Java ile yüzde yüz uyumlu çalışırla bilirlilik.

Kotlin dilini öne çıkaran en önemli etkenlerden birisi Google firmasının Android Geliştirme bölümünün bu dile güvenmesi ve Android uygulamaları geliştirmek için desteklemelerini duyurmasıdır.

➤ **Kotlin, RxJava ve RxAndroid**

ReactiveX ilk olarak Microsoft tarafından ortaya çıkarılmış bir kütüphanedir. Observable yaklaşımı (*Bilinen yordamsal (prosedürel) programlamada olduğu gibi ardışık olarak işlenmiş kodsall süreçleri adım adım yerine getirip tepki göstermekten ziyade, olay tabanlı duyarlılık sergilemekte olan ve belli başlı olaylar cereyan ettiği zaman refleks göstermesine observable yaklaşımı denir*) kullanan ardışık ve olay tabanlı programlar içindir. RxJava da ReactiveX’in Java Virtual Machine için gerçeklemesi (implementation) aslında. Java 6 ve üzerinde kullanabiliyor.

Sınıf beyanı:

Kotlin aslında Java dili ile etkileşime girme ihtiyacı ile tasarlanmış bir dildir. Mevcut Java kodu, Kotlin'den doğal bir şekilde çağrılabilir ve Kotlin kodu da Java'da sorunsuz bir şekilde kullanılabilir.

```
class Person(  
    id: Int,  
    name: String,  
    surname: String
```

```
) : Human(id, name),  
KotlinMaker {  
// ...  
}
```

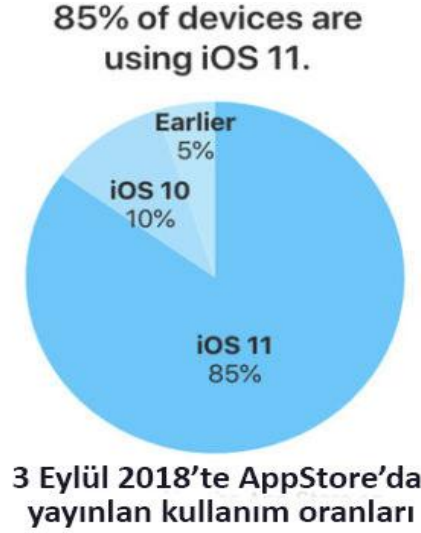
3.2.2 IOS Programlama

iOS, Apple firmasının mobil platformlar için geliştirdiği bir mobil işletim sistemidir. Çekirdeğini MAC OSX den alan iOS içinde 4 katman bulundurmaktadır:

- 1- Core OS tabakası (Sockets ve dosya sisteminin olduğu nesneye yönelik olmayan katmandır),
- 2- Core Servisleri tabakası (nesneye yönelik ara yüzlerle bu işletim sisteminin çoğunu kapsadığı katmandır),
- 3- Medya tabakası (Ses, müzik, video gibi dosyaların çalıştırıldığı katmandır.),
- 4- Cocoa Touch tabakası (Dokunmatik ekran üzerinde yapılan parmak hareketlerini algılayan bir yapıya sahip ve nesneye yönelik bir ara yüzün olduğu en üst katmandır).

Programcılar uygulama geliştirmeye başlamadan önce işletim sistemi sürümlerinin kullanım oranlarını mutlaka gözden geçirmelidir. Çünkü programlarda hafıza yönetimi (Garbage Collection) çok önemlidir. Özellikle bazı sürümlerden sonra ortaya çıkan yapılar geliştiricinin üstünden yükü kaldırır. Örneğin iOS 7 ile birlikte standart haline gelen “ARC” gibi. Automatic Reference Counting kelimelerinin kısaltması olan bu sistem nesnelerin yaratıldıktan sonra onların hafızadaki durumunu düzenleyen sistemdir. Bu sistem sayesinde her seferinde nesneleri elle bellekten çıkarmamıza gerek kalmayacaktır. Bu da programcıya büyük bir kolaylık sağlamaktadır.

iOS işletim sistemleri için programlama dili olarak Objective-C kullanılmaktadır. Dil genel olarak C tabanlıdır. Fakat C dilinden oldukça farklı özellikleri de vardır.



Şekil 3.3 iOS işletim sistemleri kullanım oranları

3.2.2.1 Objective-c Programlama Dili

Objective-C, C dilinin üzerine yazılmış, yansımali, nesne yönelimli bir programlama dilidir. 1983 yılında geliştirilmeye başlanan dil ObjC, Objective C ve Obj-C olarak da isimlendirilir. 26 Ekim 2007 tarihinden itibaren Apple firması tarafından geliştirilmeye başlanan dil OpenStep standardı üzerine kurulu olan Mac OS X ve GNUstep işletim sistemlerinde kullanılmaktadır. Objective-C, C'nin syntax'ı(sözdizimini), ilkel türleri ve akış kontrol deyimlerini miras alır. Sınıfları ve yöntemleri tanımlamak için söz dizimi ekler. Objective-C, dinamik yazma ve bağlama sağlar. Aynı zamanda nesne grafik yönetimi ve nesne hazırlıkları için dil seviyesi desteği de ekler.

TIOBE indekslerine göre iOS programlama için yazılım geliştiricilerin yıllarca ilk tercihi olan ve C programlama dili esaslarına dayanan Objective-C listede geçen yıla (2018) göre iki sıra yükselmiş olmasına rağmen popülerlik bakımından sonra çıkan Swift'in gerisindedir (İnt. Kyn.9).

Objective-C'deki modül dosyaları “.m” uzantısına sahiptir. Başlık dosyaları “.h” dır. Objective-C'de oluşturulan tüm nesneler dinamik belleğe yerleştirilmelidir. Bu nedenle, herhangi bir sınıftaki bir nesnenin bir göstergesi olan id türü özel bir önem taşır. Aşağıdaki örnekte, bir for döngüsü içinde tanımlanmış bir x değişkeni var. Bu, x'in o

blokta yerel olduğu anlamına gelir. For döngüsü içinde x'e erişebilir ancak başka hiçbir yerde erişilmezdir.

```
#import
int main (int argc, const char * argv[]) {
    int i;
    for (i=0; i<10; i++) {
        int x = 10;
        NSLog(@"x = %i", x);
    }
    NSLog(@"x = %i", x); // x'in sınırları dışında kullanılmış
}
```

3.2.2.2 Swift Programlama Dili

Swift programlama dili 2 Haziran 2014 yılında Apple tarafından düzenlenen WWDC (Apple etkinlikleri - <https://developer.apple.com/wwdc/>) konferansında duyurulan, iOS ve OS X platformlarına uygulamalar geliştirmek için oluşturulmuş, öğrenilmesi kolay, nesne yönelimli ve güçlü bir dildir. Swift, C ve Objective-C'deki en iyileri bir araya getiren iOS ve OS X sistemlerine uygulamalar geliştirmek için tasarlanan yeni bir programlama dilidir. Xcode derleyicisi kullanarak tasarlanabilir. Swift dilinin hazırlanmasında sadece Objective-C, değil Rust, Haskell, Ruby, Python gibi birçok programlama dilinden faydalanılmıştır. Swift, kullanımı kolay ve açık kaynak kodlu bir programlama dilidir. Birçok üniversite ve eğitim kurumlarında, bilgisayar programlama derslerinde öğretiliyor olması geleceğini güçlendirmektedir. Swift, gerçek zamanlı geri bildirim olanağı sağlayan ve mevcut Objective-C koduna sorunsuz bir şekilde monte edilebilir.



Şekil 3.4 Swift dili ile yazılmış bazı popüler uygulamalar

Şekil 3.4’te Swift dili ile yazılmış bazı popüler uygulamalar görülmektedir. Daha detaylı bilgi için <https://www.apple.com/tr/swift/> adresi ziyaret edilebilir. Apple, Swift’in Objective-C’den 2,6 kat, Python’dan 8,4 kat daha hızlı olduğunu söylüyor. Kısacası dilin ana felsefesi sadeliği, hızı ve güvenliğidir. Detaylı Swift, Objective C karşılaştırması Çizelge 3.3’te verilmiştir.

Çizelge 3.3 Swift, Objective C karşılaştırması

Benzerlikler	Farklılıklar
Int, Float, Double gibi sayısal türleri	Swift’te noktalı virgül (;) kullanımı zorunlu değil
for...in numaralandırma söz dizimleri	Başlık dosyaları Swift’te bulunmamakta
Apple ürünleri için halen geliştirmeye devam etmeleri	Swift’te String ifadeler Unicode’u tamamı ile desteklenir Syntax yapısı Objective C ‘ye göre oldukça basit ve anlaşılması kolaydır. Hafıza/Bellek Yönetimi (Memory Management) Swift’te daha esnek

Swift geliştiricileri bu programlama dili ile telefonlar, akıllı saatler ve Apple bilgisayarlar için mobil uygulamalar yazmak için kullanır. Google’ın Android’i tamamen Swift’e transfer etmek istediğine dair bir söylenti bile vardı. Önemli olan nokta, 2016 sonbaharında Apple’ın herkesin iOS için uygulamaları nasıl programlayacağını öğrenebileceği bir öğrenim merkezinin açılışına katkıda bulunması. Ancak hepsi bu

kadar değil, Swift zaten birçok lider eğitim kurumunun müfredatına dâhil edildi. Bu programlama dili büyük bir potansiyele sahip olduğunu gösteriyor.

3.2.2.3 BlackBerry OS Programlama

BlackBerry işletim sistemi, 1999-2013 yılları arasında Java tabanlı BlackBerry marka akıllı telefonlar için geliştirdiği ve halen 10.2.1 sürümü olan mobil işletim sistemidir. Kanadalı şirket RIM (Research In Motion) tarafından üretilen BlackBerry Geliştirme araçlarını ücretsiz sağlamaktadır. Uygulama hata ayıklama işlemi hem simülörde hem de doğrudan cihazda yapılabilir. Uygulamayı cihazda çalıştırabilmek için, .cod uzantılı derlenmiş dosya, RIM tarafından sağlanan özel bir elektronik anahtarla imzalanmalıdır. Anahtarı edinme süreci çok basittir. RIM web sitesindeki (<https://www.blackberry.com/SignedKeys/codesigning.html>) formu (Şekil 3.5) doldurarak bu anahtara ücretsiz sahip olunabilir. BlackBerry akıllı telefonları için uygulama geliştirme, aşağıdaki araçlar kullanılarak gerçekleştirilebilir:

BlackBerry Code Signing Keys

Jump

- Getting Started
- BlackBerry World Vendor Portal
- Platform & Developer Services
- Developer Resources
- Code Signing
- Why Develop for BlackBerry?

BlackBerryKeys Order Form

For BlackBerry NDK, AIR, and Android app developers

- If you're developing apps using the BlackBerry 10 Native SDK, AIR, and Android you can also request your BlackBerry ID token from the Deployment Setup Wizard in the BlackBerry Momentics IDE.
- Make sure you have the latest development tools available for your platform:
 - * Native SDK * AIR tools * Android tools * PlayBook

I need to sign apps:

- ☐ For BlackBerry 10 apps developed using a BlackBerry 10 SDK 10.2 or higher, or BlackBerry 10 WebWorks SDK 2.0 and higher, or any apps repackaged using the BlackBerry repackaging tools for Android version 1.8.1 or higher
- ☐ For BlackBerry PlayBook apps or BlackBerry 10 apps developed using a 10.1 or lower SDK or BlackBerry 10 WebWorks SDK below 2.0.
- ☐ For BlackBerry OS 7.x and lower

* Indicate a required field

Personal Information

First name*

Last name*

Company*

Email*

Country*

Registration PIN

Your PIN can be any 6-10 digit, lowercase, alphanumeric code. Your PIN protects against usage of your Code Signing Keys by unauthorized parties, to keep it safe. BlackBerry reserves the right to request that you choose another PIN if deemed unsuitable.

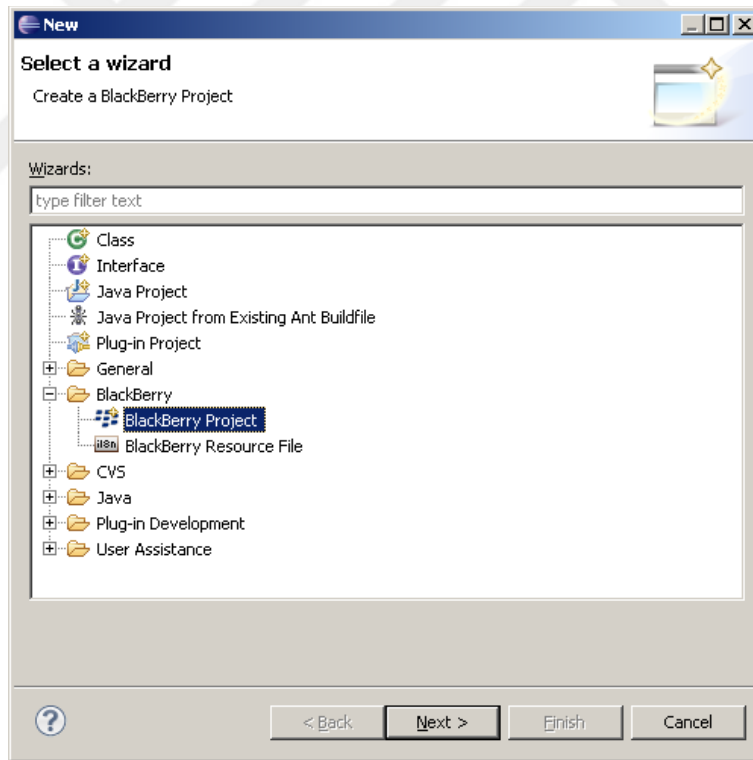
Şekil 3.5 BlackBerry anahtar edinme sayfası

1- BlackBerry Java Geliştirme Ortamı (BlackBerry JDE)- Java'da BlackBerry

uygulamaları geliştiren ilk araçlardan biriydi. Biraz karmaşık bir uygulama geliştirmek oldukça zordur. Eclipse, Netbeans, IDEA ve diğerleri gibi modern IDE'lerin donatıldığı gibi pek fazla kullanışlı özellikler yoktur.

2- Eclipse için BlackBerry JDE Eklentisi-Eclipse için bir eklenti, yükleyerek BlackBerry için uygulamalar oluşturabilir ve derleme yapabilirsiniz. BlackBerry JDE'den çok daha kullanışlı bir araçtır. Yüklemeden önce, eklentinin gerektirdiği Eclipse sürümünü yüklemeniz gerekir. BlackBerry JDE Plugin'in kurulumu sırasında kurulu Eclipse yolu istenecektir.

3- HTML, CSS ve JavaScript kullanılarak yazılmış ve BlackBerry tarayıcısında çalışan "Tarayıcı Uygulamaları" adlı bir araç seti. Aşağıda BlackBerry JDE Plugin tarafından desteklenen Eclipse sürümünü yüklenmesi ile ilgili bir örnek verilmiştir (Şekil 3.6, Şekil 3.7 ve Şekil 3.8):



Şekil 3.6 Eclipse ile BlackBerry sistemine kod yazma

BlackBerry Application Descriptor - Application

General Information

[This section describes the general BlackBerry properties used in this application.]

Title:	MyFirstApp
Version:	1.0.0
Vendor:	Bilici
Description:	My first application
Application type:	BlackBerry Application
Name of main MIDlet class:	
Application argument:	

☐ Auto-run on startup

Startup tier: 7

☐ Do not display the application icon on the BlackBerry home screen

Şekil 3.7 BlackBerry için ilk uygulama

Bir kodlama örneği:

```
package test.myfirstapp.view;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.container.MainScreen;
public class MyAppScreen extends MainScreen {
    public MyAppScreen() {
        // Varsayılan menüyü oluşturuyoruz
        super(DEFAULT_MENU | DEFAULT_CLOSE); // Başlık ayarlama
        setTitle("İlk uygulamam");
        // Düzenlenemez bir alana (Label) ekliyoruz.
        this.add(new LabelField("Merhaba Dünya!")); }
    // Ekranın kapatılması ayarları
    public boolean onClose() { // Ekran kapandığında uygulamayı kapat
        System.exit(0);
        // true, ekranın kapanacağını onaylar
        return true;
    }
}
```




Şekil 3.8 Eclipse ve BlackBerry JDE eklentisi ile ilk programlama

3.2.2.4 Windows Phone Programlama

Windows Phone olarak adlandırılan Microsoft'un telefonlar ve diğer akıllı ürünleri için kullandığı işletim sistemi Windows Mobile olarak Evrensel tüm platformlar için kullanılan işletim sistemine dönüşmüştür.

Mimariye Genel Bakış

Ortak çekirdek ve uygulama yürütme sistemi sayesinde, bir kez yazılmış olan bir UWP uygulaması Windows 10 çalıştıran tüm cihazlarda çalışacaktır. Bu aşağıdakileri içerir:

- Masaüstü bilgisayarlar;
- Sunucu sistemleri - Windows Server 2016;
- Dizüstü bilgisayarlar - MS Surface Book;
- Tabletler - MS Surface Pro;
- Akıllı telefonlar - Lumia;
- Oyun konsolları - Xbox One;
- Artırılmış Gerçeklik Camları - MS HoloLens;
- Duvar plakaları - MS Surface Hub;

- Akıllı Saat - MS Band 2.

Geliştirici Araçları ve süreç

- Windows 10 Mobile için uygulama yazmak, hata ayıklamak, dağıtmak ve test etmek için Visual Studio 2017 programı kullanılıyor.
- Windows 10 Anniversary Update SDK yüklü olmalıdır.
- Yine Windows Ink API'si ve diğer API'ler sayesinde daha hızlı ve basitçe programlama yapılabilir.
- Visual C # en çok kullanılan dildir. Visual Basic, Visual C ++ veya JavaScript kullanarak da mobil programlama yapılabilir.
- Eski projeleri **Desktop App Converter** programı sayesinde sorunsuz bir şekilde dönüştürme imkânı vardır.

3.3. Çerçeveler

Çerçevenin kullanımı, geliştirme çaba maliyetlerini azaltarak verimliliği arttırmaktır. İdeal olarak, geliştiricilere, bir uygulamayı geliştirirken gerçekten önemli olan sorunları çözme konusunda zaman kazandırmaktır.

Dünyanın dört bir yanındaki 200 ülkede milyonlarca akıllı telefon insanların kullanımındadır. Bu tez ile popüler geliştiricilerden Android ve IOS platformlarına uygulamalar geliştirmek için kullanılan popüler Frameworkleri (çerçeve) bir arada ve karşılaştırılmalı olarak ele alınmaktadır. Çalışmada kurulumu yapılabilen ya da çevrim içi yapılar ele alındığı gibi kod bilgisine ihtiyaç duyulmadan profesyonel uygulama geliştirme sistemleri de anlatılacaktır.

Çerçevelerin kurulumu, öğrenme olanakları ve süreleri, avantajları, dezavantajları, karşılaştırılmalı olarak yer alacaktır. Çerçevelerin genel özelliği Android, iOS ve Windows Universal platformları için uygulama yapmalarıdır.

3.3.1 NativeScript

Telerik firması tarafından geliştirilen javascript kullanarak Android, iOS ve Windows Universal platformları için native uygulamalar yazılmasını sağlayan frameworktür. NativeScript uygulamaları, Javascript veya TypeScript gibi platformdan bağımsız dillerde geliştirilmiştir.

NativeScript, AngularJS çerçevesi için tam destek sağlar. NativeScript platformu ile hazırlanan mobil uygulamalar, Xcode veya Android Studio'da düzenlenebildikleri gibi platform API'lerine tam erişim sağlayabilir. Geliştiriciler, uygulamalarına başka katmanlar oluşturmadan Cocoapods, Android Arsenal, Maven ve npm.js gibi kaynaklardan üçüncü taraf kütüphaneleri de dahil edebilirler.

Sadece Javascript kodları yazarak yerel (native) mobil uygulamalar geliştirile biliniyor. Üstelik yazılan uygulamaları sadece Android ortamında değil, aynı zamanda iOS ve Windows Universal ortamlarında da çalıştırabiliyoruz. Yani tek kodla farklı platform için de uygulama yazılmış oluyor (Şekil 3.4). NativeScript gereksinimlerini yüklemek ve Windows'ta NativeScript uygulamaları oluşturmaya ve çalıştırmaya başlamak için adım adım kurulum aşaması:

Windows sistemlerinde, Android uygulamaları geliştirmek için yalnızca NativeScript CLI kullanabilirsiniz. Bunun nedeni, NativeScript CLI'nın, yalnızca macOS işletim sisteminde bulunan iOS uygulamalarını oluşturmak için Xcode kullanmasıdır. Windows'ta iOS uygulamaları oluşturmakla ilgileniyorsanız, NativeScript Sidekick'i denemek isteyebilirsiniz. NativeScript Sidekick, bulutta iOS ve Android sürümleri gerçekleştiren, bu sistem gereksinimlerini tamamlama gereksinimini ortadan kaldıran ve Windows'ta iOS için oluşturmanıza izin veren bir hizmet de dahil olmak üzere NativeScript uygulamaları için güçlü araçlar sunar. Kurulum için Windows 7 Service Pack 1 veya üstü işletim sistemine sahip olmak gerekmektedir.

3.3.2 Flutter

Flutter, Google firması tarafından geliştirilen açık kaynak kodlu (Open Source Code) bir mobil uygulama geliştirme SDK'sıdır. Android ve iOS için uygulama geliştirmek ve Google Fuchsia işletim sistemi için uygulama geliştirilebilir.

Google, yeni açık kaynak kodlu geliştirme platformu Flutter'ı Mobil Dünya Kongresi'nde (Mobile World Congress - MWC) tanıtımını yaptı. Flutter, tamamen ücretsiz olup tüm geliştiricilerin de katkı sağlayabileceği bir yazılım geliştirme kitidir. Flutter'ı kullanarak yazılımcılar, hem Android hem de iOS'ta istenilen uygulamalar geliştirebilir. Hem zamandan hem de maliyetten kazanç sağlamak için mobiluygulama geliştiricileri çapraz platformlu uygulamalar geliştirebilir.

Ara yüz tasarımcıları ise Flutter platformunu kullanarak Android ve iOS'a uygun, daha başarılı uygulamalar hazırlayabilir. Flutter'ı öğrenmek zor değildir Dart programlama diline benzerlikleri vardır. Android ve iOS için ayrı ayrı dilleri öğrenmek zorunda kalınmıyor.

Flutter'ın Hedefleri

- 1- Hızlı yazılım geliştirme (Fast Development),
- 2- Etkileyici, Ara yüzler (Expressive, beautiful UIs) ile zengin içerikli, değişik tasarımlar ortaya çıkarmak.
- 3- Kullanışlı ve modern bir framework sunmak (Modern, reactive framework)

Flutter'ın Genel Özellikleri

- Flutter, esnek bir yazılım geliştirme kitidir. Windows, Linux ya da Mac işletim sistemlerinde sorunsuz çalışır.
- Flutter, Mobile-first 2D render motoru, react-style framework'ü, Android ve iOS platformunda kullanılan araç (widget) desteğine sahiptir.

- Flutter, yapısında *Material Design* ve *Cupertino* (iOS tarzı) widget'ler, temalar yardımıyla her iki platform için de uygun uygulamalar geliştirmeyi sağlar.
- Flutter birçok geliştirme ortamında bulunmayan *Hot reload* adında bir gerçek zamanlı düzenleme özelliğine sahiptir. Hot Reload ile geliştiriciler uygulamalarını hazırlar veya düzenlerken, gerçek zamanlı olarak yaptıkları değişikliklerin uygulama üzerindeki yansımalarını anında görebilirsiniz.
- **Flutter** ile Android Jelly Bean v16 sürümü üzeri ve daha sonraki sürümler için mobil uygulamalar geliştirilebilir. iOS tarafında ise iOS 8 ve sonraki sürümleri için uygulamalar geliştirebilir. Donanım tarafına bakıldığında Android'in ARM işlemcilerini destekleyen SDK, iOS tarafında ise 64-bit işlemci mimarisini destekleyen iPhone5s ve üzeri cihazlara uygun mobil uygulamalar geliştirebilir.
- **Flutter** ile ilgili en güncel ve yeni kaynaklar kendi sitesinde bulunan Flutter dokümanlarıdır. <https://flutter.io/docs> adresinden erişilebilir.

3.3.3 Cordova

21 Nisan 2015 tarihinde, Cordova ekibi (eski adı ile PhoneGap), web geliştiricilerin aynı anda birkaç platform için mobil uygulamalar geliştirmesine olanak tanıyan araçlarının 5.0 sürümünü yayımladı. Cordova, mobil uygulamaya bir tarayıcı yerleştirerek farklı platformlarda mobil uygulamalar geliştirmenize izin veren bir platformdur. Dolayısıyla, uygulama aslında tek bir siteyi gösteren bir mini tarayıcıdır. İndirmeyi hızlandırmak için tüm kaynaklar uygulamanın dağıtılmış paketine yerleştirilebilir ve gerekirse sunucudan indirilebilir.

Kısaca Cordova (<http://cordova.apache.org/>), yazılan HTML, CSS ve JS kodlarını mobil uygulama olarak ekrana çıkaran bir framework'tür. Açık kaynak kodlu ve tamamen ücretsiz arkasında Apache'nin bulunduğu bir yapıdır.

Uygulama oluşturma

Kurulumu yaptıktan sonra (Ek-2) Cordova ile basit bir Hello World uygulaması geliştirmeye hazırdır. Yeni bir Cordova uygulaması oluşturmak için bir terminal açın ve

Cordova uygulaması oluşturma komutunu çalıştırın:

cordova create photosharer com.yourname.photosharer PhotoSharer

Kod:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Security-Policy"
content="default-src 'self' data: gap: https://ssl.gstatic.com
'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src *">
    <meta name="format-detection" content="telephone=no">
    <meta name="msapplication-tap-highlight" content="no">
    <meta name="viewport" content="user-scalable=no,
initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-
width">

    <link rel="stylesheet" type="text/css"
href="css/index.css">
    <title>Merhaba Dünya</title>
  </head>
  <body>
    <div class="app">
      <h1>Apache Cordova</h1>
      <div id="deviceready" class="blink">
        <p class="event listening">Aygıt a
bağlanılıyor</p>
        <p class="event received">Aygıt hazır</p>
      </div>
    </div>

    <script type="text/javascript"
src="cordova.js"></script>
    <script type="text/javascript"
src="js/index.js"></script>
  </body>
</html>
```

İkinci komut dosyası index.js dosyasını içerir. Bu, yeni bir Cordova projesi oluştururken varsayılan olarak oluşturulan belirli bir uygulama dosyasıdır. Belirli bir olayda belirli bir kodu yürütür. Bu dosyayı açarsanız, receivedEvent gizleme fonksiyonu ve yukarıda tanımladığımız iki elementi göreceksiniz. Sonra onDeviceReady olayı çağırılır. Böylece, fonksiyon etkinlik tanımlayıcı receivedEvent olacaktır aygıt hazırda geçirilir.

```
onDeviceReady: function() {
```

```

    app.receivedEvent('deviceready');
},

receivedEvent: function(id) {

    var parentElement = document.getElementById(id);
    var listeningElement =
parentElement.querySelector('.listening');
    var receivedElement =
parentElement.querySelector('.received');

    listeningElement.setAttribute('style', 'display:none;');
    receivedElement.setAttribute('style', 'display:block;');

    console.log('Received Event: ' + id);

}

```

Uygulamayı çalıştırmak için öncelikle Ratchet mobil web uygulamaları oluşturmak yani bir ön platform eklenmelidir. İndirme sayfasından veya komut kullanarak Bower kurulumu yapılır:

```
bower install ratchet
```

3.3.4 Appcelerator Titanium

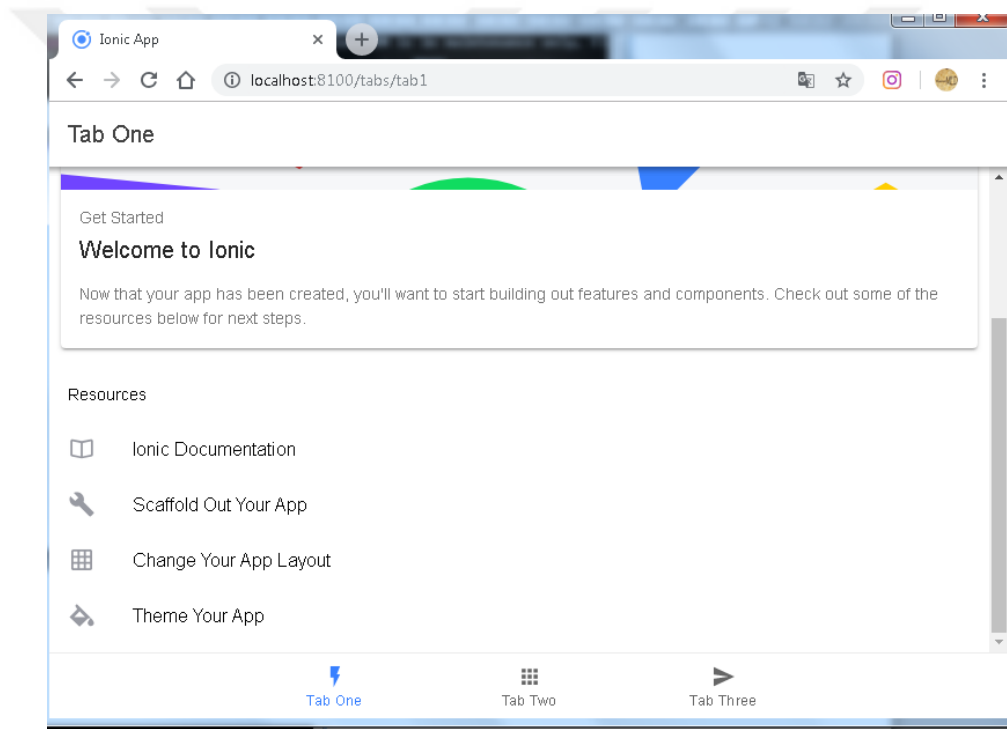
Titanium SDK, Apache 2 lisansı altında, JavaScript (önyükleme için HTML + CSS) ve JavaScript API'leri ile mobil platformların sahip olduğu yerel API'leri kullanmadan mobil uygulamalar geliştirme imkânı sağlayan bir platformdur. Asıl hedeflenen geliştiricinin, Android Platformu için Java veya Kotlin, iOS platformu için Objective-C veya Swift öğrenmesine gerek kalmadan JavaScript bilgisi ile mobil uygulama geliştirmesini sağlamaktır.

Titanium uygulamalarında bir platform için yazılan JavaScript kodunu, %80 oranında değiştirmeden başka platformlarda da kullanılabilir. Bunları yaparkende hiç bir şekilde lisans ücreti ödenmeyecektir. Çünkü Appcelerator, Titanium'u Apache 2 lisansı altında Kişisel ve Ticari kullanıma sunmuştur. Titanium ile Android, iOS ve Blackberry işletim sistemlerinde uygulamalar geliştirilebilir. Titanium Studio'yu çalıştırmak için Oracle JDK ve Node.js yüklemek gereklidir. Titanium Studio ve SDK (yazılım geliştirme kiti)

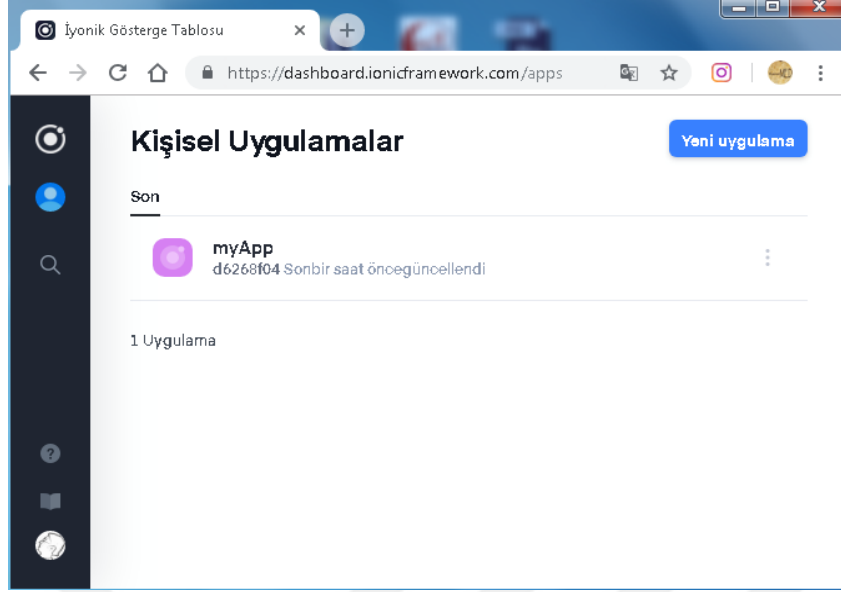
kurulmalıdır. Titanium Studio uygulaması, Eclipse üzerine inşa edilmiş bir geliştirme ortamlarından birisidir. Daha önce Eclipse ortamını kullananlara tanıdık. Titanium SDK ise JavaScript kodunun yerel koda dönüştüğü yerdir.

3.3.5 Ionic

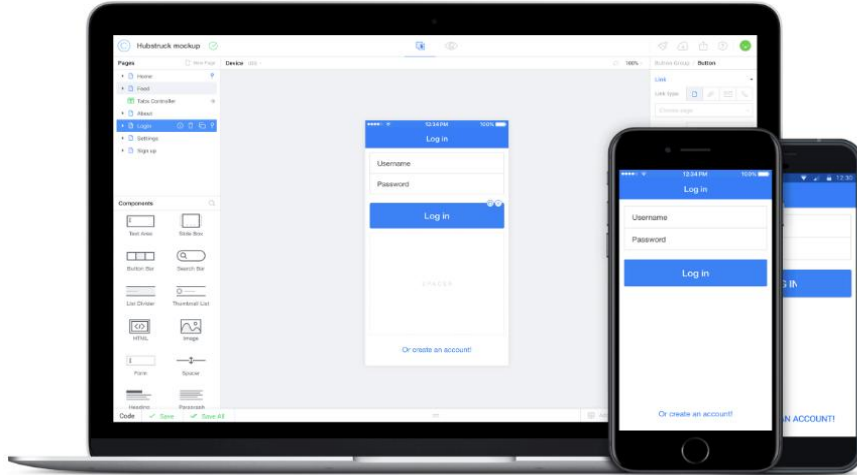
Ionic, mobil uygulamalar geliştirmek için HTML5 tabanlı hazırlanmış açık kaynak kodlu ve uç kullanıcılar için hazırlanmış bir framework'tür. İki farklı kurulumu vardır. Kurulum yapıldıktan sonra uygulamalar hazırlanabilir. Tarayıcı üzerinde ki çalışan hali Şekil 3.9'da görülmektedir.



Şekil 3.9 Tarayıcı üzerinde çalışan hali



Şekil 3.10 Ionic kişisel uygulamaların ekranı



Şekil 3. 11 Tasarım ara yüzü ile çalışma

Ionic Kullanımı Artıları:

- 1- Ionic angular.js ile phonegap için hazırlanmış bir framework olduğu için en büyük artılarından birisi kodu bir sefer yazmanıza rağmen otomatik olarak android ve ios tasarım düzenlerine bu tasarımı uyarlanır.
- 2- Kodları HTML/CSS/JS ile yazıldığı için uygulamaları test ederken emulator kullanmak yerine tarayıcı (browser) üstünde test etmesidir.
- 3- Uygulama yapmak çok hızlı ve kolaydır.

3.3.6 Alpha Anywhere

Mobil uygulama geliřtiricileri arasında hızlı uygulama geliřtirme platformu olarak anılan Alpha Anywhere, çevrimiçi ve çevrimdışı çalışabilen hibrit (karma) mobil uygulamalar geliřtirme sunan bir platformdur. Platformun sunduđu SQL veri tabanı desteđi ile istenilen tüm veri tabanları için SQL etkileřimler oluşturabilir ya da Alpha Portable SQL ile uyumlu mevcut veri tabanına SQL bağlantılar kurula bilinir. Çevrimdışı mobil uygulama kullanımı sunması, hibrit uygulamalarda büyük miktarlarda sıkıştırılmış veriler için mobil dosya sistemine erişim özelliđi eklenmesi ve Mobile Optimized Forms gibi özellikleri kullanımını artırıyor. Yıllık ücreti 995 dolardır.

Resmi Web Sitesi : www.alphasoftware.com/

Alpha Yazılım Örneđi Uygulama Kütüphanesini kullanarak hızlı uygulamalar hazırlanabilir.

3.3.7 Xamarin

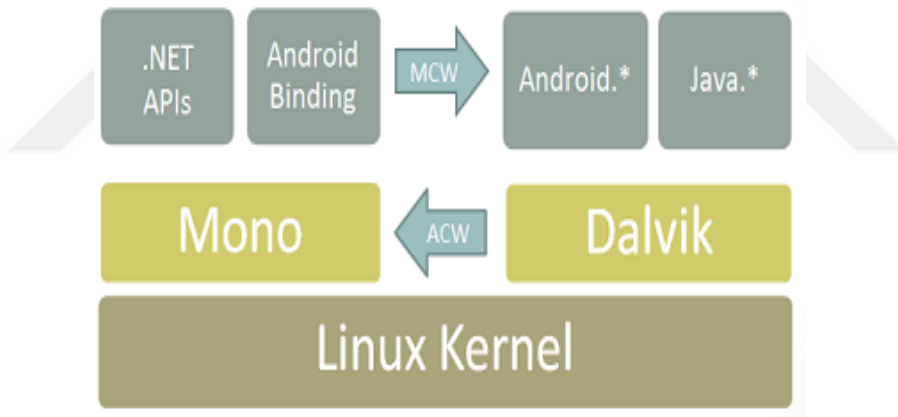
Xamarin, .NET kodu ve platforma özgü kullanıcı arabirimlerini kullanarak Android, iOS, macOS ve Windows Phone için yerel uygulamalar oluşturmanıza olanak sağlayan 2011 yılından itibaren Microsoft firmasının geliřtirdiđi bir yazılım geliřtiricisidir. Örneđin C# dilini kullanarak Cross (çapraz) Platform mobil uygulama geliřtirme imkânı vardır. Çerçeve birkaç ana bölümden oluşmaktadır:

- Xamarin.iOS, geliřtiricilere iOS SDK'ya erişim sağlayan C # için bir kütüphane;
- Xamarin.Android - geliřtiricilere Android SDK'ya erişim sağlayan C # için bir kütüphane;
- iOS ve Android için Derleyiciler;
- IDE Xamarin Studio;
- Visual Studio için Eklenti.



Şekil 3.12 Xamarin çalışma prensibi

Xamarin Nasıl Çalışır?



Şekil 3.13 Xamarin çalışma sistemi

Xamarin, C#'ı kullanarak her platform için özel yerel kullanıcı ara yüzünü tasarlama esnekliği sağlar. Tekrar kullanılan kod miktarı, ne kadar kodun tutulduğuna ve kullanıcı ara yüzüne ne kadar taşındığına bağlıdır. Xamarin birkaç ana bölümden oluşmaktadır (Şekil 3.12, Şekil 3.13):

- iOS SDK'ya erişim sağlayan C # sınıfı bir kütüphane;
- Android SDK'ya erişim sağlayan C # için sınıf kitaplığı;
- iOS ve Android için Derleyiciler;

- IDE Xamarin Studio;
- Visual Studio için Eklenti.

C# dili ile hazırlanan kodlar, her platform için farklı yerel uygulamalara derlenir.

Xamarin'i neden kullanmalıyım?

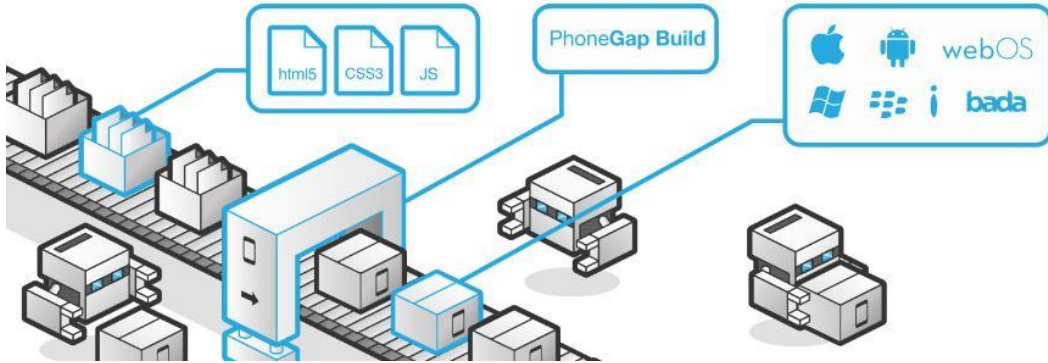
1. Xamarin, native platformların gücünü bir araya getirir. Xamarin uygulamaları modern bir dil olan C# ile geliştirilmiş ve Objective-C ve Java üzerinde pek çok geliştirme vardır.
2. Xamarin, Objective-C, Java ve C ++ kütüphanelerini doğrudan dâhil etme imkânı sağlar. Java, Objective-C veya C ++'da bulunan birçok 3. parti kod tabanı yeniden kullanılabilirlik yeteneği kazandıracaktır.
3. Xamarin, üç büyük mobil platform (Android, iOS ve Windows) için cross platform desteği sunar. Uygulama kendi kod tabanının %90'ını paylaşıyor.
4. Xamarin mobil uygulama geliştirmenin zaman ve maliyetini düşürür.
5. Xamarin'in kendi üçüncü taraf bileşen deposu vardır.
6. Xamarin, sağlam derleme zamanı denetimi sağlar. Bu, çalışma zamanı hataları ve yüksek kaliteli uygulamalar için daha az yol açacaktır.
7. Xamarin, .NET platformunun açık kaynaklı uygulamasına dayanmaktadır

3.3.8 PhoneGap

PhoneGap web tabanlı mobil uygulamalar hazırlamayı sağlayan bir uygulama geliştirme frameworküdür(çerçevesidir). PhoneGap ile uygulama yazacağınız platformun dilini (native language) kullanmadan HTML, CSS ve Javascript (Şekil 4.13) gibi her platformda çalışan web teknolojileri ile uygulamalar yazıla bilinir.

Programları indirme adresi: <https://phonegap.com/getstarted/>

PhoneGap kullanarak Mobil uygulama geliştirme



Şekil 3.14 PhoneGap çalışma sistemi

Çerçeveyi (Framework) kurmak için hem bir bilgisayar uygulaması hem de komut satırı kullanılabilir.

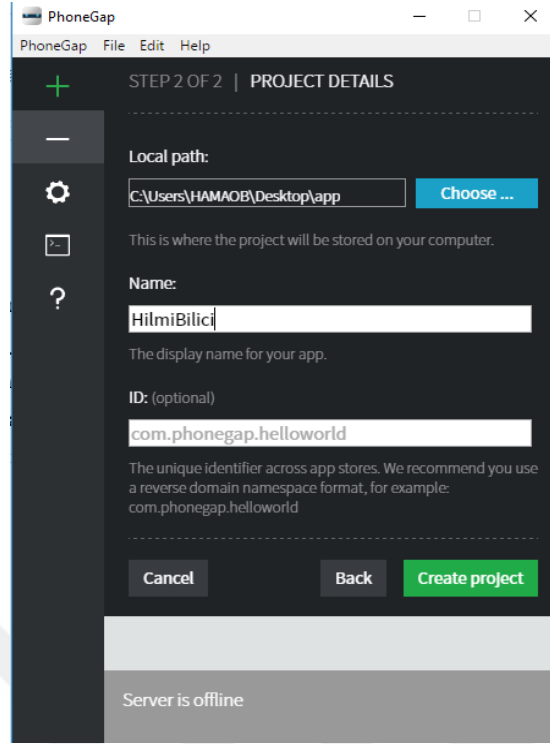
- 1- Komut satırı ara yüzü (CLI) ile başlamak için aşağıdaki komutun yazılması yeterlidir. Fakat CLI devam edebilmek için node.js ve git uygulamaları yüklü olmalıdır.

```
npm install -g phonegap
```

Ya da Şu anda beta sürümünde olan yeni PhoneGap Desktop uygulamasını indirin ve yükleyin.

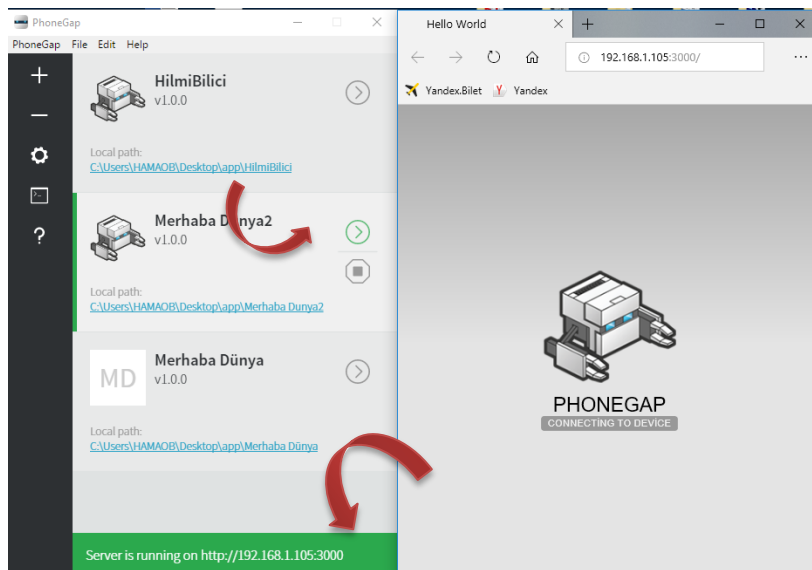
- 2- Mobil uygulamanın yüklenmesi gereklidir. PhoneGap Geliştirici uygulaması birden fazla platformda (Apple yönergeleri nedeniyle, PhoneGap Developer App iOS App Store'dan kaldırılmıştır. Yeni kullanıcılar indiremez) kullanılabilir ve PhoneGap projenizi kod imzalama veya derleme yapmadan mobil cihazınızda çalıştırmanızı sağlar.
- 3- Yeni PhoneGap uygulamasını oluşturun:

Başlatmak için, PhoneGap Desktop uygulamasını açın (1. adımda yüklenen) ve yeni bir proje oluşturun.



Şekil 3.15 PhoneGap Yeni proje oluşturma ekranı

- 4- Yeni uygulama cihazda ön izleme yapılır. PhoneGap Desktop uygulaması, projenizi barındırmak için küçük bir web sunucusu başlatır ve bir kez bu web sunucusuna bağlandığında, PhoneGap Developer uygulaması projenizi mobil cihazınızda çalıştırır.
- 5- Artık kodlama yapılabilir.



Şekil 3.16 PhoneGap uygulama çalıştırma

```

1 <!DOCTYPE html>
2 <html><head>
3 <meta charset="utf-8" />
4     <meta name="format-detection" content="telephone=no" />
5     <meta name="msapplication-tap-highlight" content="no" />
6     <meta name="viewport" content="user-scalable=no, initial-
7 scale=1, maximum-scale=1, minimum-scale=1, width=device-width"
8 />
9     <!-- This is a wide open CSP declaration. To lock this down
10 for production, see below. -->
11     <meta http-equiv="Content-Security-Policy" content="default-
12 src * 'unsafe-inline'; style-src 'self' 'unsafe-inline'; media-
13 src *; font-src * data: blob;" />
14 <title>Merhaba Dünya</title>
15 </head>
16 <body data-dom-cache="true">
17     <div id="app"></div>
18     <script type="text/javascript" src="dist/build.js"></script>
19     <script type="text/javascript" src="cordova.js"></script>
20 </body></html>

```

3.3.9 Sencha Touch

Sencha Touch (ST), iPhone, iPad ve Android odaklı web siteleri ve web servisleri geliştirmek için kullanılan bir çerçevedir (framework). ExtJS kütüphanesinin tanınmış bir tedarikçisi olan Sencha, Inc. tarafından, yüksek teknoloji ürünü tarayıcı ara yüz pazarı için geliştirilmiş ücretli bir platformdur. 400 \$ ile 1900 \$ civarı arasında paketleri vardır. ST'nin asıl amacı, mümkün olduğu kadar, standart ara yüz ve iOS ara yüzünün elemanlarının davranışını taklit ederek programcılara esnek bir API sunmaktır.

Çekirdek kütüphane JavaScript, CSS3 ve Html5 ile yazılmıştır. FireFox 4 ve IE 9, desteklenen tarayıcılar listesinde değildir. Buna göre, bir ST uygulaması ortak bir istemci-sunucu geliştirmedir, Safari veya Chrome tarayıcısı bir istemci olarak çalışır. JSON, XML ve diğer birkaç formatta "yanıt verebilecek" herhangi bir web hizmeti bir sunucu olarak davranır. Temel olarak, çerçeve yine tanıdık bir HTML5 + JS + CSS grubudur. Ancak bu durumda, ara yüzü geliştirirken JS'ye HTML5'ten daha fazla önem veriyor. Çerçeve, hem UI bileşenlerini hem de mevcut olanları temel alarak geliştirmek için esnek araçlar sunar. Herhangi bir kütüphane bileşeni aslında bir dizi yöntem ve özelliğe sahip bir ST nesnesidir. Ek- 6 'da kurulum ve uygulama oluşturma adımları anlatılmıştır.

Sencha'nın avantajları:

- Kendi bileşenlerinizi oluşturarak kendi ara yüzünüzü oluşturma esnekliği (daha katı bileşen gereksinimleri)
- Hızlı bir başlangıç için çok fazla belge olması

3.3.10 jQueryMobile

2010 yılında piyasaya sürülen dokunmatik ekran odaklı jquery takımının mobil platformlar için geliştirdiği HTML 5, CSS kullanan bir javascript framework'üdür. jQuery Mobile, PhoneGap, Worklight ve diğer mobil çerçevelerle uyumludur.

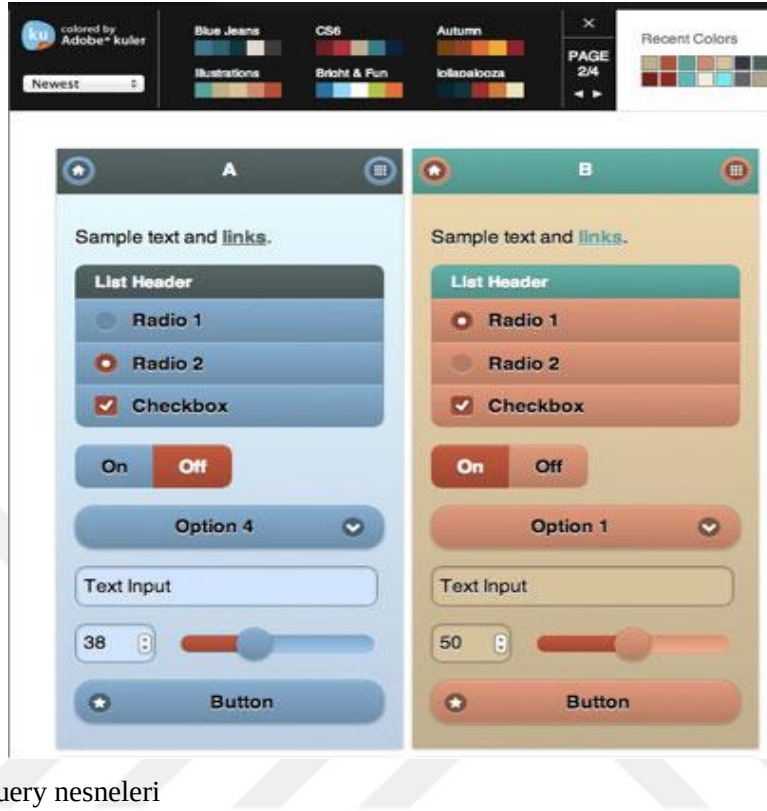
JQuery Mobile Nasıl Kullanılır?

JQuery Mobile 'ı kullanmak için bilgisayarınıza hiç bir şey kurmanıza ve depolamanıza gerek yoktur. Birer js ve CSS dosyası olan kütüphaneleri bağlayarak hemen kullanmaya başlanılabilir. Bu dosyaları kendi bilgisayarınızda da saklayıp referans göstermek sureti ile de bu mümkündür. <http://jquerymobile.com/download/> linkini tıkladığınızda, JQuery Mobile'ı kullanmak için size birçok seçenek sunar. Burada 3 temel dosyanın kullanılması gerektiği açıktır. Bunlar birincisi uzantıları .js olan JQuery, ikincisi uzantısı JQuery.Mobile ve üçüncüsü uzantısı .css olan JQuery.Mobile'dır. İfadelerin sonunda yazılan sayılar ise hangi sürüm olduklarını gösterir.

JQuery Mobile kullanmak için beş neden

- 1- Geniş tarayıcı ve cihaz kapsamı- aşamalı gelişmeler sayesinde web siteniz çok sayıda tarayıcıda ve cihazda görülebilir.
- 2- Dokunmatik giriş desteğine sahip formlar ve kullanıcı arabirimi widget'ları- form girişi için dokunmatik desteği, dokunmatik girişli bir dizi kullanıcı arabirimi widget'ı ekleyerek artırabilir.
- 3- Hazır yanıt veren web tasarımı-(Şekil 3.21 hazır nesneler) jQuery Mobile, duyarlı web tasarım örnekleri sunan bir çerçevedir.
- 4- Tarayıcı ve tema şablonları kolayca genişletilebilir ve kolayca değiştirilebilir.

- 5- Ajax çağrı cihazı navigasyon modeli- tam sayfa yenileme yerine pushstate Ajax ve HTML5 kullanarak yeni sayfalar daha hızlı yüklenir.



Şekil 3.17 JQuery nesneleri

Gerekli betikleri bağlama:

```
<link
href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css"
" rel="stylesheet" type="text/css" />
<script type="text/javascript"
src="http://code.jquery.com/jquery.min.js"></script>
<script type="text/javascript"
src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js">
</script>
```

Sayfa içi kodlama:

```
<div data-role="page" id="main_page" data-theme="b">
  <div data-role="header" >
    <h1 id="twi_acc">Ana Sayfa</h1>
```

```

        <a href="#settings" data-icon="gear" class="ui-btn-
right">Ayarlar</a>
    </div>
    <div data-role="content" >
        <ul data-role="listview">
            <li><a href="#twi-page">Örnekler</a></li>
            <li><a href="#map-page">İletişim</a></li>
            <li><a href="#search-page">Arama</a></li>
            <li><a href="#about-page">Hakkımda</a></li>
        </ul>
    </div>
    <div data-role="footer">
        <h2>© Hilmi Bilici, 2019</h2>
    </div>
</div>

```

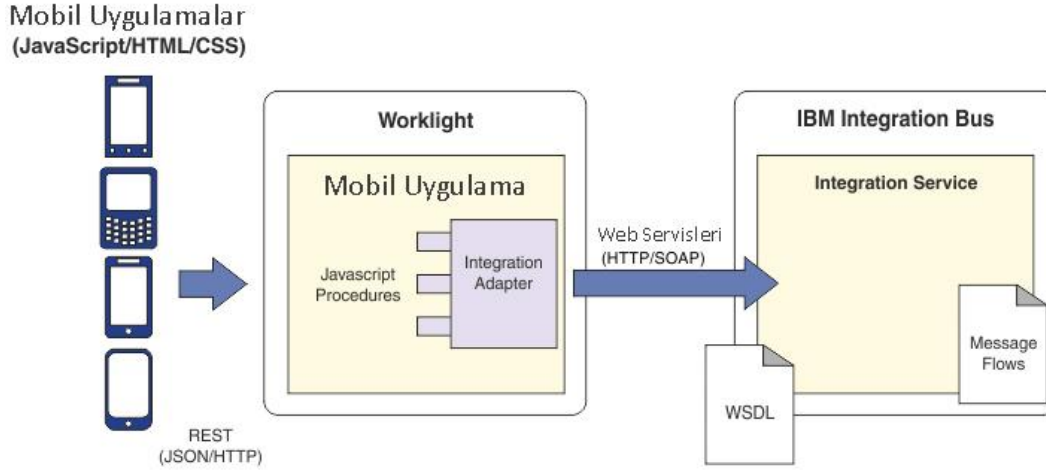


Şekil 3.18 JQuery örnek uygulama çıktı ekranı

3.3.11 IBM Worklight

IBM Worklight, arka uç hizmetlerle etkileşimde bulunan ve mobil hibrit uygulamalar oluşturmayı sağlayan ve gereken tüm araçları içeren bir mobil uygulama platformudur. 2012 yılında IBM, kuruluşların işlerini değiştirmelerine ve müşterilere mobil çözümler sunmalarına olanak sağlayan IBM Worklight adlı ilk mobil geliştirme ve entegrasyon araçlarını aldı. IBM Worklight, geliştiricilere bir uygulama oluşturmaları ve her ortam için, yani Apple iOS, Google Android, Blackberry ve Microsoft Windows Phone için taşımalarına gerek kalmadan birden fazla mobil platformda çalıştırmaları için gerçekten

açık bir yaklaşım sunar. IBM Worklight ayrıca, jQuery Mobile, Dojo Toolkit ve Sencha Touch gibi popüler kütüphaneler için uzantıları olan HTML5 ve JavaScript gibi standart teknolojileri kullanarak geliştiricinin işini kolaylaştırır.



Şekil 3.19 IBM Worklight çalışma prensibi

IBM Worklight ile Mobil Uygulama Geliştirmenin yaşam döngüsü;

- **Worklight Studio:** IBM Worklight, geliştiricilerin birden fazla işletim sistemi için mobil uygulama oluşturmasını sağlayan Eclipse tabanlı Worklight Studio adlı bir geliştirme ortamı sunar.
- **Worklight Sunucusu:** Bu bileşen, çevrimdışı şifreli depolama, birleşik Push Bildirimi ve daha fazlası için kullanılan adaptörlerle merkezi arka uç bağlantısı aracılığıyla güvenli veri iletimini etkinleştiren veya sağlayan bir çalışma zamanı sunucusudur.
- **Worklight Aygıt Çalışma Zamanı:** Aygıt çalışma zamanı doğada çapraz platformlu ve IBM Worklight Sunucusu tarafından sağlanan hizmetlere kolay erişim sağlayan zengin bir API kümesi sağlar.
- **Worklight Konsolu:** Bu, gerçek zamanlı analitik, Push Bildirim Yetkilisini Yönetme ve Mobil Sürüm Yönetimi için web'e bağlı bir ara yüzdür. Worklight Konsolu web tabanlı bir ara yüzdür ve Worklight Server ve onun dağıtılmış uygulamaları, uyarlayıcıları ve push bildirim servislerinin sürekli yönetimi için ayrılmıştır.

- **Worklight Uygulama Merkezi:** Mobil Uygulama Geliştirme ekibinin özel ihtiyaçlarını ortaya koyan çapraz platformlu bir Mobil Uygulama Mağazasıdır.

Worklight'ı kullanıcı ara yüzü oluşturmak için kullanmanın ve müşteri tarafında olduğu kadar sunucu tarafında da gelişmeyi yansıtan büyük bir avantaj var. Genel olarak geliştirici, geliştirme ve diğer ürünler kullanarak hibrid uygulamanın yaratılması için destek verirken, kullanım durumlarını tanımlamak, hata ayıklamak, işletme uygulaması için ön izleme testi yapmak için tipik değildir. Bir Worklight kullanarak, geliştirici basit mimari ve gelişmiş doğaçlamalı yapabilir.

3.3.12 Corona

Corona SDK, Android ve iOS için mobil uygulamalar yapmanızı sağlayan bir yazılım platformudur. Corona çapraz bir platformdur, bir kod yazılır ve ardından küçük değişikliklerle farklı işletim sistemlerinde çalışan hazır bir program elde edilir.

Grafik yönelimi nedeniyle, Corona öncelikle atari salonları gibi iki boyutlu oyunlar yaratmayı amaçlıyor, ancak aynı zamanda iş uygulamaları geliştirmek için gerekli araçları da içeriyor.

Corona SDK'nın artıları ve eksileri

Her platformda olduğu gibi, Corona SDK'nın kullanımı hakkında karar verilmesi gereken avantaj ve dezavantajları vardır. Ayrıca, uygulamanızın odağına da bağlıdır. Corona SDK, standart dışı bir ara yüze sahiptir. Karmaşık iş uygulamaları için çok uygun olmasa da 2D oyunlar ve basit uygulamalar için çok uygundur.

Artıları:

- Kullanımı basit Lua dili ile yazılmıştır.
- Çapraz uygulama hazırlanabilir.
- Corona, liste, kaydırıcı, düğme vb. kullanıcı etkileşimi için hemen hemen tüm gerekli ara yüz öğelerini içerir.

Eksileri:

- Dokümantasyon eksiği vardır.
- Hata ayıklama, platformun en zayıf noktasıdır.
- Yazılım kodu ve tasarım ayrımı yoktur.
- Ücretlidir.

Corona SDK sadece bir emülatördür (başka bir yazılım ve donanım ortamını taklit etme). Editör veya tam geliştirme ortamı yoktur. Bu nedenle, Lua'daki program kodunu düzenlemenizi ve hata ayıklamanızı sağlayan üçüncü taraf araçlarını kullanmanız gerekir. Araçları ve eklentileri görmek için EK-8'i incelenebilir.

3.3.13 ENYO

HTML 5 ile mobil uygulamalar oluşturmaya yarayan ücretsiz bir frameworktür (çerçeve).

Geliştirme Ortamınızı Hazırlama

Bir Enyo geliştiricisi, uygulamalarını kodlamak için kullandığı platformu ve araçları seçme konusunda büyük bir özgürlüğe sahiptir. Windows, Mac veya Linux ortamında çalışabilir. Geliştirme sürecinde kodu herhangi bir modern Web tarayıcısını kullanarak test edebilirsiniz. Dosyaları doğrudan tarayıcıya yerel dosya sisteminden yükleyerek test etmek mümkün olsa da, uygulamaya bir HTTP sunucusu üzerinden erişim önerilir. Bu yaklaşım, uzak cihazlardan test etme yeteneği de dahil olmak üzere sayısız avantaj sağlar. Kurulum ve örnekler için Ek-9'a bakınız.

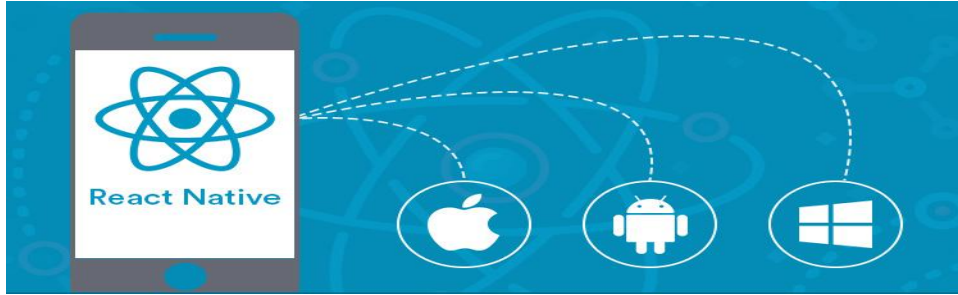
Platformlar arası ENYO çerçevesi ilk sürümleri WebOS'un bir parçası iken daha sonra ayrı bir projeye dönüşerek webOS, Apple iOS ve Android gibi mobil platformlara da uygulama hazırlayan bir yapıya büründü. ENYO topluluğu, kuruluşundan bu yana, Enyo'ya, içinde görülebilecek 100'den fazla eklenti hazırlayarak gelişmesini sağladı.

3.3.14 LUNGO

Aygıtlar arası uygulamalar tasarlamak, oluşturmak ve paylaşmak isteyen geliştiriciler için kullanılan açık kaynak kodlu bir HTML5 çerçevesidir. HTML5, CSS3 ve JavaScript gibi açık web standartlarını destekler. Cep telefonlarında, TV'lerde ve masaüstü cihazlarda tutarlı bir tarayıcı ortamı sunar. Her platform için uygulama oluşturmanın pahalı olduğu bilinmektedir, bu durum tabletlerin ve SmartTV'lerin gelmesiyle artar. Lungo hepsine uyacak uygulamalar için hazırlanmış güçlü API desteği sunar. Kurulumu ve kodlama ile ilgili bilgileri EK-10'da mevcuttur.

Bu platform kullanılarak geliştirilen uygulamalar tüm popüler işletim sistemlerinde (iOS, Android, Blackberry ve WebOs) çalışmaktadır. WebSQL, coğrafi konum, tarih veya ekran yönü gibi HTML5 özelliklerini de kullanılabilir. Ek olarak, Lungo.js tamamen modüllere dayanmaktadır ve kendi ihtiyaçlarınızı karşılayacak şekilde yapılandırılabilir.

3.3.15 React Native



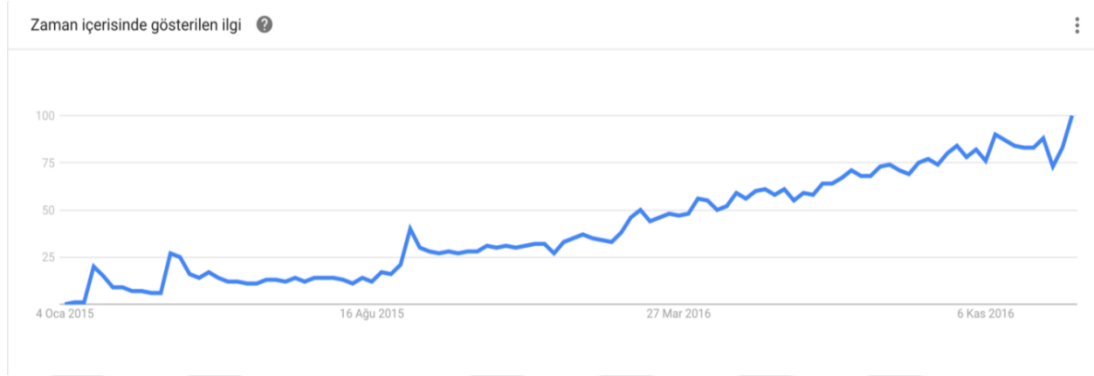
Şekil 3.20 Platformlar arası mobil uygulama geliştirme çerçevesi React Native

React native Facebook firması tarafından 2015 yılında geliştirilmiş hem android hemde IOS'a hizmet verebilen bir Native mobil platformudur. Aynı zamanda bir JavaScript kütüphanesidir. Geliştiricilerin işlemleri daha kolay, hızlı amaçlanmıştır.

Neden React Native öğrenmeliyim?

1. React Native, hibrit uygulama framework'leri arasında gösterilmektedir.

2. Uygulamalar uzaktan güncellenebilir.
3. Uygulama geliştirme hızı artar.
4. Büyük bir topluluk desteği var. Github'da sayısız tutorial uygulamalar bulunabilir.



Şekil 3.21 React native'in zamanla gösterdiği kullanım artışı

5. Facebook desteği vardır.
6. **Write Once, Run Everywhere** sloganı ile Hibrit uygulamalar gibi bir kere yazılıp bütün platformlarda çıktı alınabilir olması.
7. Native uygulamanızı yeniden yazmak zorunda değilsiniz.
8. Farklı ekran tipleri için (Flexbox) çözümler vardır.
9. İstedığınız bir nodejs modülü dahil edilebiliyor.
10. React'ı Öğrenmesi gerçekten kolaydır.
12. Babel syntax(*) yapısını kullanıyor.

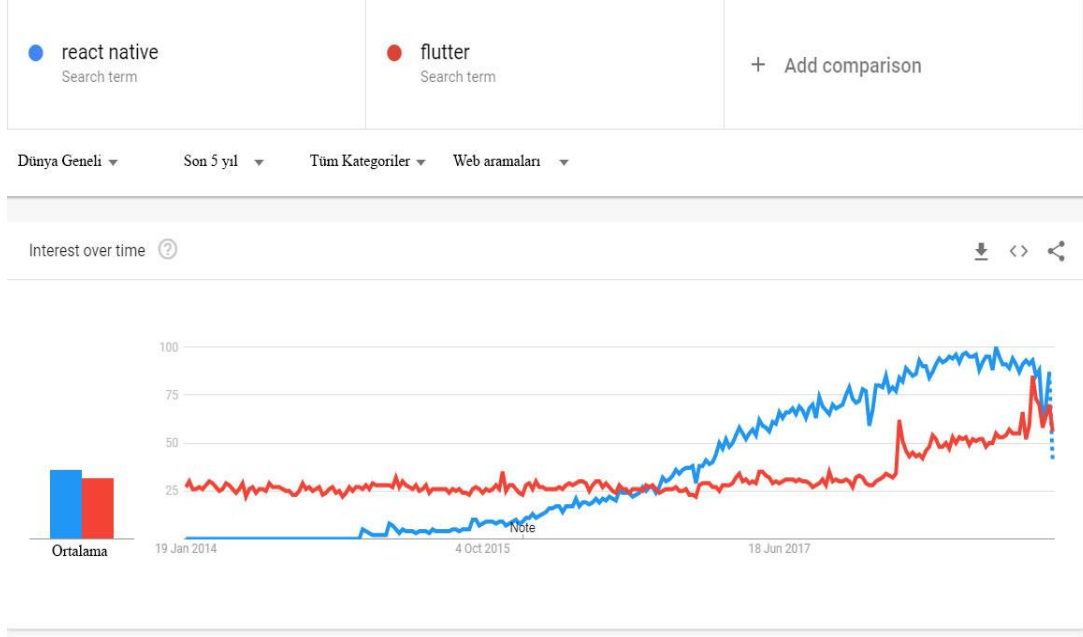
(*) Babel, ECMAScript 2015+ kodunu güncel ve eski tarayıcılarda veya ortamlarda JavaScript'in geriye dönük uyumlu bir sürümüne dönüştürmek için kullanılan bir araçtır.

Dezavantajları:

- Ionic gibi standart tasarım kalıpları (design pattern örnek: <https://medium.com/nsistanbul/design-patterns-factory-method-pattern-615457e9560b>) mevcut değildir.
- Sayfanın render işlevi aynı sayfada olduğu için front-end teknolojisi ile yazarlar ve css yazarlar çakışma yaşanması mümkündür.

Kurulum ve kodlama örnekleri için EK-11'e bakılabilir.

React Native ve Flutter Karşılaştırması: Yeniden kullanılabilir kod yapısı, kütüphane ve topluluk desteğine göre Çizelge 4.1'de detaylı bir karşılaştırma yapılmıştır.



Şekil 3.22 React Native ve Flutter karşılaştırması

3.3.16 Kivy

Kivy Açık kaynak kodlu bir Python kütüphanesidir. %90 Python ile yazılmıştır ve Mobil Uygulama programlamak için kullanılır. Ücretsiz bir yazılım olan Kivy Android, iOS, Linux, OS X ve Windows üzerinde çalışabilir. Mit lisansı şartları altında dağıtılır.

Neden Kivy?

İlk olarak, bu javascript değil Python tabanlı bir dildir. Bu nedenle, geliştirme hızı, kodun özdeyişi, uygulamalardaki değişiklikleri anında değiştirme ve izleme yeteneği, basit kod yazma yeteneği, uygulamaların çıktısını alma yolları farklıdır. İkincisi, bu %99,9 oranda gerçek bir üst platformlar arası çerçevedir ve kodunuzun bir kez yazıldığında mevcut tüm platformlarda çalışacağından emin olunabilir. Kurulum ve kodlama örnekleri EK-12'de mevcuttur.

3.3.17 JQTouch

iPhone, Android, iPod Touch ve diğer benzeri akıllı cihazlara mobil web geliştirme için David Kaneda ve Thomas Yip tarafından yazılan açık Kaynak kodlu bir Zepto / jQuery eklentisidir. Öncelikle JQTouch'ın GitHub'dan bir kopyasını indirilir. HTML'ye Komut Dosyaları ve bağlantılar ekleyerek Api özelliklerini kullanmaya başlanır.

```
<style type="text/css" media="screen">@import
"../../themes/css/jqtouch.css";</style>
<script src="../../src/lib/zepto.min.js" type="text/javascript"
charset="utf-8"></script>
<script src="../../src/jqtouch.min.js" type="text/javascript"
charset="utf-8"></script>
```

3.3.18 Alpha Anywhere

Alpha Anywhere, mobil cihazlar ve masaüstü bilgisayarlar için veri tabanı odaklı iş uygulamaları oluşturmak ve dağıtmak için çeşitli araç geliştiriciler için kullanılan bir platformdur. Hibrit mobil uygulamalar geliştirme imkânı sunan platform iki bölümden oluşur: uygulamalar oluşturmak için bir geliştirici ve dağıtımı için bir sunucu. Uygulamalar bir web tarayıcısında başlatılabilir veya bir mobil cihaza uygulama olarak yüklenebilir. Kodlama HTML5, CSS3 ve Javascript'in bir birleşimidir. GPS ve benzeri bir donanıma erişebilecek uygulamalar veya mobil cihazlarda veya grafikler ve haritalar oluşturabilmeniz için PhoneGap veya Google API'ler gibi çok çeşitli entegre araç ve kitaplıklara erişim imkanı vardır.

SQL veritabanları: Alpha Anywhere, popüler ve kurumsal SQL sistemleri için kendi sürücülerini içerir. Buna Microsoft Access, SQL Server, MySQL, Oracle, DB2 ve diğerleri de dâhildir. Bu, başka bir sistemin çalışmasını aksatmadan başka bir sistemde kullandığınız verilere bağlanmanıza olanak tanır. Örneğin, Ruby'de yazılmış bir web uygulamanız veya Visual Basic'te (veya MS Access) bir masaüstü uygulamanız varsa, bu sistemi yerinde bırakabilir ve eski bir sistemle veri paylaşan bir mobil uygulama oluşturabilirsiniz.

Çapraz Platform / Çapraz Cihaz Desteği: Alpha Anywhere, HTML5 / JavaScript uygulamaları oluşturduğundan, onları Mac / PC masaüstü ve dizüstü bilgisayarlar, akıllı telefonlar ve tabletler dâhil olmak üzere bir web tarayıcısına sahip hemen hemen her cihazda çalıştırabilirsiniz. Mobil cihazlarda, Alpha Anywhere uygulamaları kendi uygulamaları olarak kurulabilir ve cihaz donanımına erişebilir.

Rapor yazımı: Alpha Anywhere, çeşitli veri kaynaklarından karmaşık raporlar oluşturabilen bir rapor oluşturucu içerir. Raporlar PDF, HTML, Excel ve diğer dosyalar olarak oluşturulabilir. Raporlar ekranda, uygulamada görüntülenebilir veya e-posta ile otomatik olarak gönderilebilir.

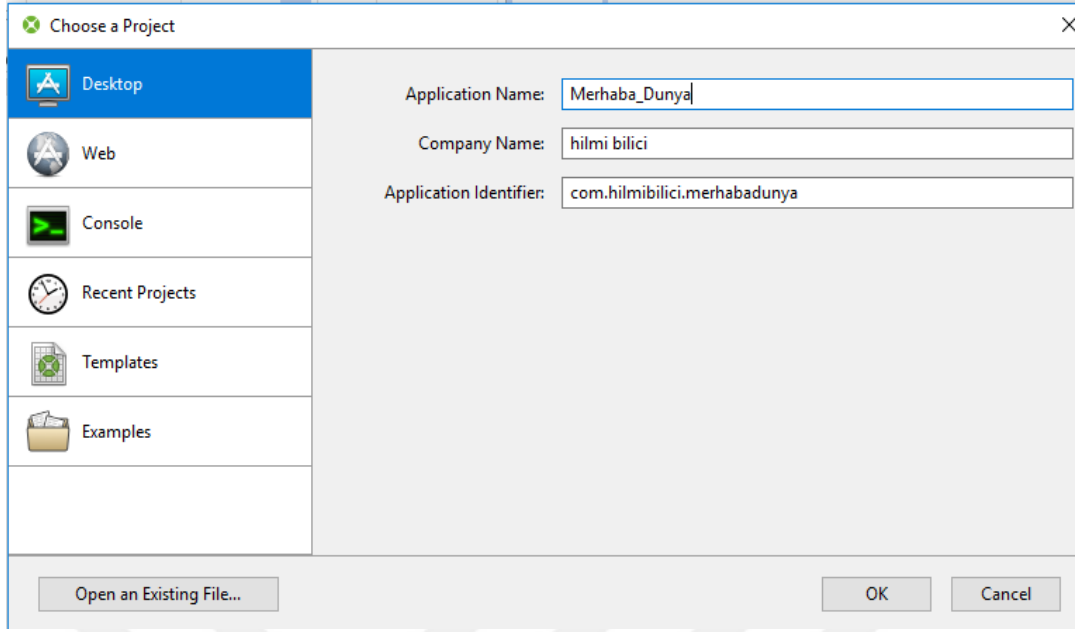
Zengin kullanıcı deneyimi: Alpha Anywhere, masaüstü / mobil ve web uygulamaları için çok zengin kullanıcı arayüzlerini destekler. Uygulamalar haritalar, çizelgeler, göstergeler ve özel kontroller içerebilir. Ayrıca, dokunmatik ekranda çizim yapmak, resimlerle ilgili notlar oluşturmak için görüntü açıklamaları, imzaları yakalamak ve tüm ses ve videoları desteklemek için mürekkep yönetimi araçları da vardır.

Güvenlik: Alpha Anywhere, rol tabanlı bir güvenlik altyapısı, şifreleme işlevleri, SSL desteği içerir ve IIS ve .NET platformunda yerleşik güvenlik özelliklerini kullanabilir. Alpha Anywhere, Yıllık 1.499 dolardan başlayan fiyatlarla satılmaktadır.

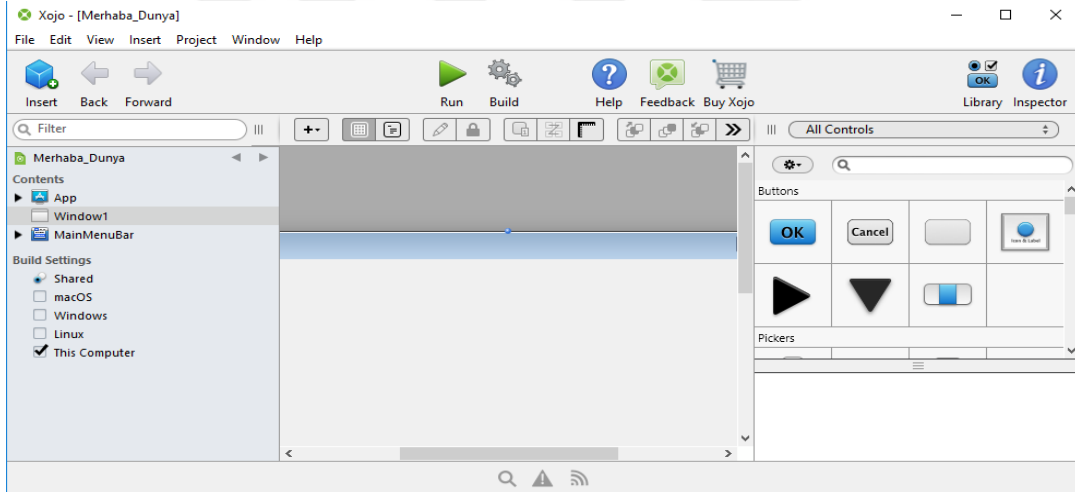
3.3.19 Xojo

Ücretli bir çapraz platform (hybrid) frameworküdür. Android, iOS, Linux, OS X ve Windows üzerinde çalışarak mobil uygulamalar geliştirilebilir. CNN, Nasa, Intel, Xerox, Cisco gibi firmaların kullanması platformu öne çıkaran artılarındanır.

Resmi web sitesi: <https://xojo.com/account/create/>



Şekil 3.23 Xojo proje oluşturma ekranı



Şekil 3.24 Xojo çalışma ekranı

3.3.20 RhoMobile

Açık kaynaklı bir topluluk tarafından desteklenen ve JavaScript, HTML5 ve CSS3 gibi modern kodlama dilleri üzerine inşa edilen RhoMobile Suite, zengin özelliklere sahip, yerel kurumsal uygulamalar oluşturmak için mevcut web geliştirme becerilerini kullanır. Gelişmiş widget'ların ve efektlerin stil kitaplıklarıyla bütünleşmesi kolaydır, bu da daha iyi görünümlü, daha iyi çalışan uygulamalar oluşturmayı basitleştirir.

RhoStudio

RhoMobile uygulamalarınızı geliřtirmenize, test etmenize ve hata ayıklamanıza izin veren güçlü bir Eclipse eklentisidir. Hem Windows PC'de hem de Mac'te çalışır ve RhoMobile uygulama geliřtirmeniz için özelleřtirilmiř bir cihaz ve platform simülatörü olan RhoSimulator içerir. **Resmi web sitesi:** <http://rms.rhomobile.com/>

Bu yazılan frameworkler dıřında řu platformlarda vardır (Bir kısmı çevrim içi (online) platformlar kısmında anlatılmıřtır.):

- AppInstitute,
- Appery.io,
- AppMachine,
- AppMakr,
- AppYourself,
- Appy Pie,
- Bizness Apps,
- Como,
- GameSalad,
- Good Barber,
- iBuildApp,
- Kalipso,
- Kony,
- LiveBlox,
- Mobile Roadie,
- Outsystems,
- TheAppBuilder,
- Viziapps

3.3.21 Mobil Oyun Geliştirme Çerçeveleri

Ücretsiz veya düşük maliyeti, oyun hizmetlerinin ve reklam hizmetlerinin uygun bağlantısı, geliştirme kolaylığı (Görsel stüdyo veya kendi geliştirme ortamının olması), kullanılan programlama dili Mobil cihazlar için oyun geliştirme ortamlarının çeşitliliğini artırmıştır. Mobil oyun geliştirme çerçeveleri:

- 1.Unity
2. Unreal Engine
3. Corona SDK – The 2D Game Engines
4. SpriteKit – 2D sprite-based games
5. Marmalade SDK
6. Buildbox
7. AppGameKit
8. Construct 2
8. Fusion
9. GameMaker Studio
- 10.CocoonJS
- 11.MonoGame
- 12.Amazon Lumberyard
- 13.Cocos2D-x
- 14.Haxe

3.4 Çevrimiçi (Online) Mobil Uygulama Oluşturma Platformları

Kod kullanımını aza indiren ya da çok az kod ile mobil yazılım geliştirme ortamı sunan mobil programlama araçları, kütüphaneler, apiler kurumsal kullanıcılara ve geliştiricilere mobil yazılım sürecinde daha hızlı ve kolay bir ortam sunuyor. Mobil uygulama pazarına girmek için binlerce dolar ve aylarca çalışmaya gerek yok. Hızlı bir şekilde ve özel programlama bilgisine sahip olmadan uygun bir bütçe içinde uygulama oluşturmanıza yardımcı olacak birçok mobil platform var.

HTML5, Javascript, CSS3 programlama dillerinin mobil web alanında karşılaştırılması ve hibrit mobil uygulama görüşlerinin pek çok platformda farklı şekilde sunulması kafaları karıştırıyor. Mobil uygulama geliştirme süreci konusunda herkes tarafından farkına varılan ve kabul edilen konunun tek geçerli çözümü ise bu platformlar değildir. Her platformun veya aracın kendi çapında çeşitli avantajları ve dezavantajları vardır.

3.4.1 Appery.io

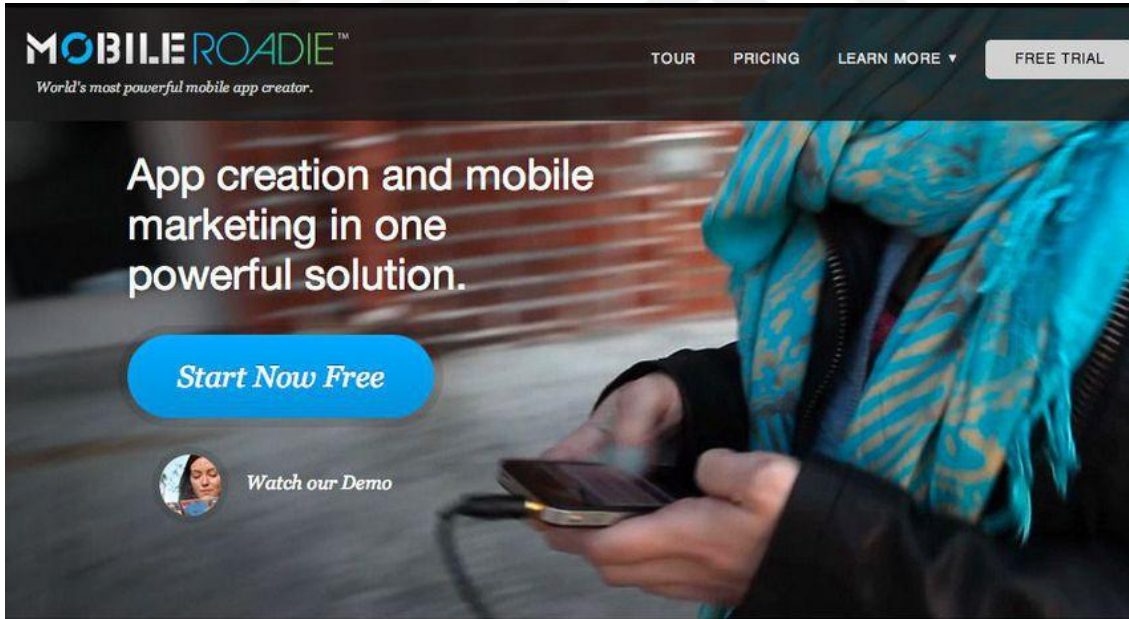


Şekil 3.25 Appery.io ana sayfası

Appery, açık kaynaklı Apache Cordova (Phone Gap) çerçevesini de içeren, gömülü bileşenlere erişim sağlayan Android, iOS ve Windows Phone'da mobil uygulamalar oluşturmak için bulut tabanlı bir hizmettir. Bu bir bulut hizmeti olduğundan, hiçbir şey yüklemek ve indirmek gerekmez ve başlaması kolaydır. Bileşenleri sürükleyip bırakarak görsel bir editör bir kullanıcı arayüzü oluşturur. Herhangi bir REST API'sine anında bir bulut veritabanı eklemek ve verileri kaydetmeniz gerekirse uygulama ile ilişkilendirmek için bağlayabilirsiniz.

Appery eklenti dizini, kendi eklentilerinizi oluşturmak için gerekli tüm işlevselliğe sahiptir. İşbirliği için iyi fırsatlar var, mobil projenizi geliştirme ekipleri, iş kullanıcıları ve tüketicilerle gerçek zamanlı olarak paylaşabilirsiniz. **Fiyat:** ücretsiz deneme sürümü vardır. Premium sürümü ise 180 \$.

3.4.2 Mobileroadie.com



Şekil 3.26 Mobil Roadie ana sayfası

Mobil Roadie; iOS ve Android için kullanılan bir uygulama tasarımcısıdır. Platform, RSS, Twitter veya anahtar kelimeleri Google Haberler'den otomatik olarak içe aktaran ve kullanıcı sohbetini gerçek zamanlı olarak güncelleyen tüm medya formatlarını destekler. Free Mobile Roadie Connect servisi, kullanıcıların göreceği gibi

uygulamanızın bir ön izlemesini gösterir. Mobil Roadie, içeriğinizin kalitesini ve uygunluğunu kontrol eder. İçeriği doğrudan uygulamaya, web sitesine veya bloga aktarabilirsiniz. API dilden bağımsızdır, böylece XML, JSON, PHP, CSV ve HTML dâhil olmak üzere çeşitli biçimlerde veri aktarabilirsiniz. Uygulamanın benzersiz bir görüntüsünü kolayca oluşturabilir, tüm platformlarda uygulayabilir ve piyasaya sürüldükten sonra pazarlama araçlarını kullanabilirsiniz. **Fiyat:** Temel sürümü ayda 125 ABD Doları ve Pro sürümü aylık 667 ABD Dolarıdır.

3.4.3 The AppBuilder



Şekil 3.27 The AppBuilder ana sayfası

The AppBuilder iki yaklaşım kullanarak çalışanlar, müşteriler, etkinlikler ve reklam araçları için uygun bir yazılım sunar. Uygulamanın yapısını özelleştirmek ve ilk içerikle doldurmak için TheAppBuilder'ın eğitim veya çevrimiçi araçlarla bir uygulama oluşturulur. Özel AppLibrary kütüphanesini kullanarak, kullanıcılar markanın tarzında tasarlanabilecek çok sayıda uygulamaya sahip bir pencereye erişebilirler. Kamusal ve özel uygulamaları şifrelerle sanal bir mağazadan göndererek ve TheAppBuilder Active Directory dizinlerinin ve mevcut kullanıcıların entegrasyonu için koşullar oluşturarak korumak mümkündür. Uygulamanın yapısını ve içeriğini güncellemek, başlattıktan

sonra bile kolaydır, çünkü sonsuz sayıda değişiklik yapabilir ve bunları tek bir tıklamayla mobil platformlarda uygulayabilir. Platform iPhone, iPad ve Android'i destekliyor ve güncelleme onaylandıktan bir dakika sonra devreye giriyor. **Fiyat:** Özelleştirilebilir bir fiyat politikası var.

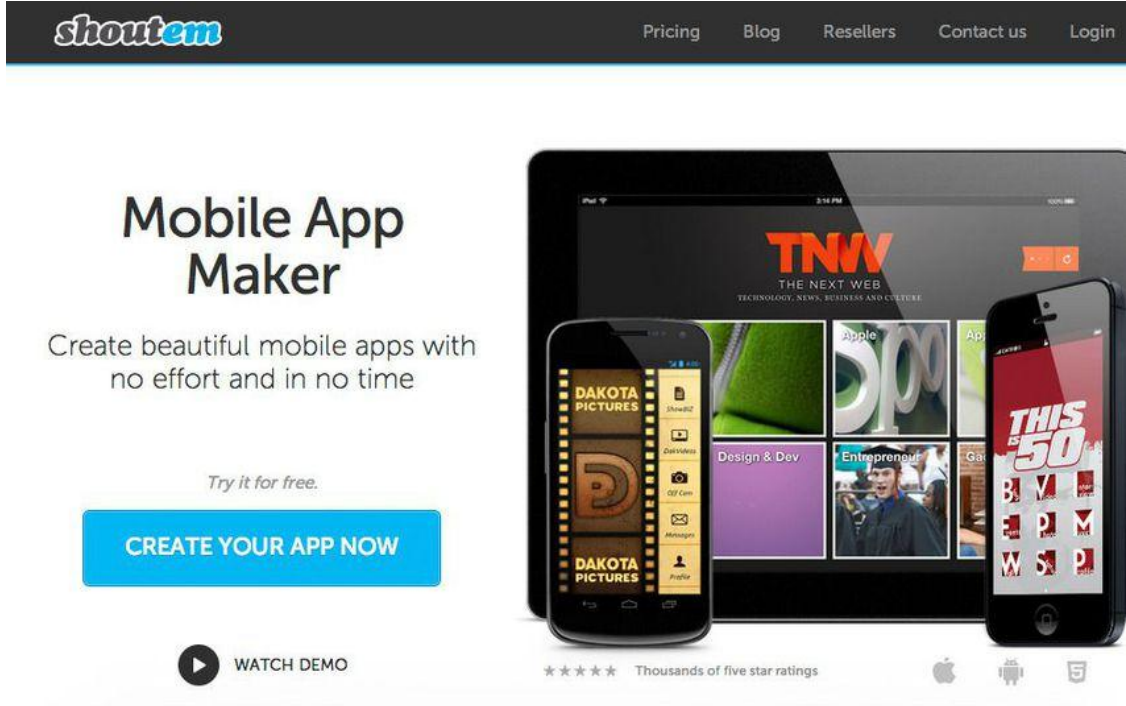
3.4.4 Good Barber



Şekil 3.28 Good Barber ana sayfası

iPhone ve Android'de uygulamalar oluşturmak için kullanılan çevrim içi bir platformdur ve uygulamanın her detayını kod yazmadan kontrol etmeyi sağlar. Dokuz renk, son derece özelleştirilebilir tasarım şemaları, 350 güzel simge ve 600'den fazla Google Fonts yazı tipi kolaylıkla başlayacaktır. iPhone ve Android uygulamalarına ek olarak, sitenin resmi mobil sürümü olmasını istiyorsanız, etki alanı ile bağlantılı olabilen, telefonda bulunan bir tarayıcı üzerinden erişim için sitenin ücretsiz bir mobil sürümü vardır. Yeni parametreleri uyguladıktan sonra her zaman görsel bir rapor alınır. Benzersiz push bildirimleri gönderilebilir ve okuyucuları faydalı kullanıcılara dönüştürerek makaleler, fotoğraflar ve videolar eklemelerine izin verilir. Video, fotoğraf, ses, şov, vb. bölümler ekleyerek uygulama özelleştirilebilir. Fiyat: Sürüm başına 16 \$, tam sürüm aylık 32 ABD dolarıdır

3.4.5 ShoutEm



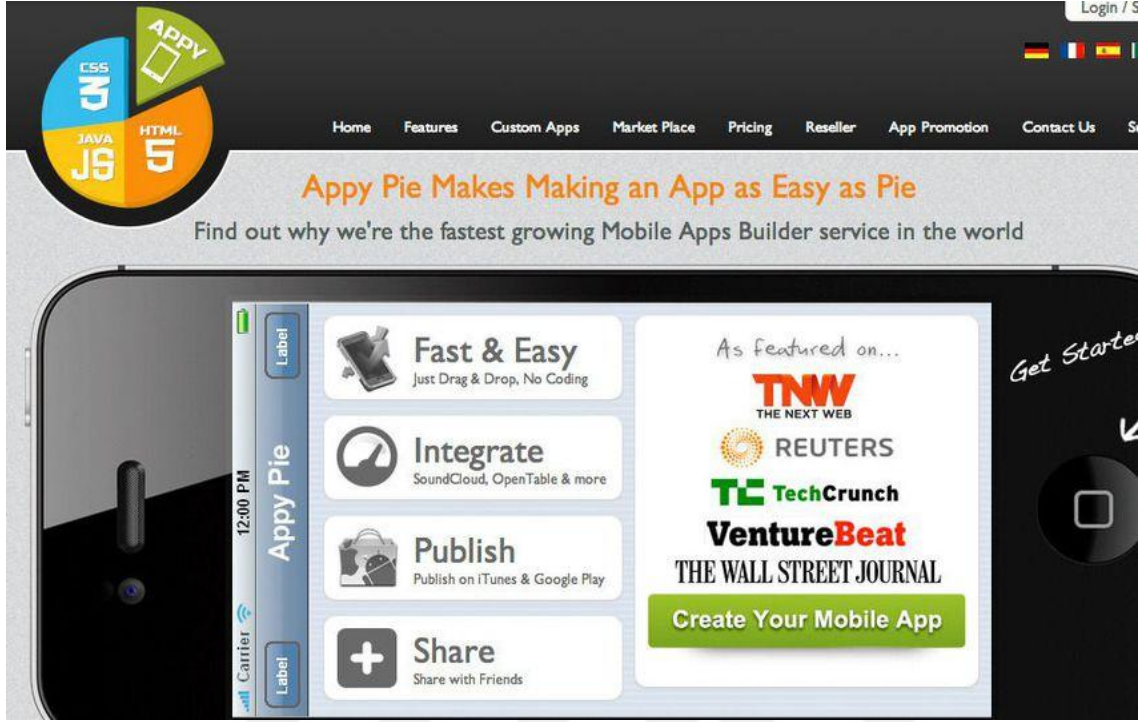
Şekil 3.29 Mobile App Maker anasayfası

Shoutem çok fazla yazılım bilmeden mobil uygulama geliştirebilecek 2008'de kullanıcıların kendi sosyal platformlarını yapmaları için kurulmuş bir platformdur. Shoutem 2011 yılında kullanıcıların yazılım ve tasarım bilgisi olmadan sadece sürükle-bırak yöntemini kullanarak Android, iOS ve HTML5 uygulamaları yapmalarına olanak sağlayan platforma dönüşmüş durumdadır.

Program işleyişi şöyle kullanıcılar kendilerine en uygun uygulama modelini seçip tasarım seçeneklerine karar veriyor. Uygun eklentilerle mobil uygulama tamamen kişiselleştirilebilir hale geliyor. Ücretlidir.

Resmi web sitesi: <http://www.shoutem.com/>

3.4.6 Appy Pie



Şekil 3.30 Appy Pie ana sayfası

Appy Pie; Programlama becerisine sahip olmayan kullanıcıların Windows, Android, iOS için uygulamalar oluşturmalarını ve bunları Google Play'de veya iTunes'da yayınlamalarını sağlayan bulut tabanlı bir araç. Hiçbir şey indirmeye veya yüklemeye gerek yok; çevrimiçi bir mobil uygulama oluşturmak için sürükle bırak yöntemi yeterlidir. Tamamlandığında, Blackberry de dahil olmak üzere tüm platformlarda çalışan HTML5 tabanlı bir hibrit uygulama elde edersiniz. Tüm kontroller gerçek zamanlı olarak gerçekleştirilir, anında bildirimler göndermek, reklamlardan para kazanmak, mevcut analizleri izlemek, GPS konumlarını izlemek, sosyal ağ haber yayınlarını, blogları, web sitelerini, ses, radyo vb.

Program planlayıcısı, özellikle bir dokunma çağrısı işlevi ve QR kodları varsa, hastaneler, güzellik salonları, spa merkezleri vb. için özellikle yararlı olacaktır. Kendi kodunuzu ve kayan çerçevelerinizi kod sayfasına eklenebilir.

Fiyat: ücretsiz ve aylık 33 \$ olan ayrıcalıklı bir sürümü var.

3.4.7 AppMachine



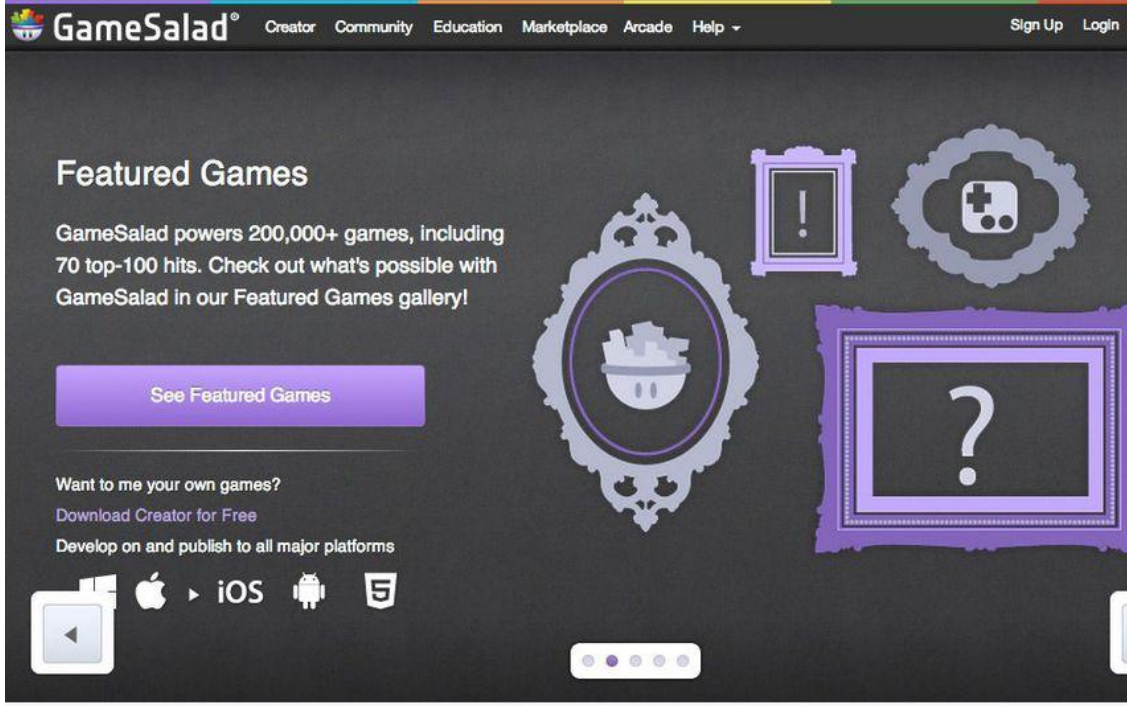
Use AppMachine to make professional apps for yourself
or your customers — fast.

Şekil 3.31 AppMachine ana sayfa ekranı

AppMachine; Profesyonel iOS ve Android uygulamaları oluşturmak ve geliştirmek için kullanımı kolay bir çevrim içi platformdur. Sürükle - bırak arayüzünü kullanarak, farklı öğeleri ve birçok işlevi, bilgiyi, fotoğrafı ve videoyu birleştirebilirsiniz. Bu öğeler, uygulamayı Facebook, Twitter ve çevrimiçi mağazalara bağlamayı sağlar. Uygulamanın tasarımı benzersiz olabilir, yapıyı tamamen kontrol edip Önizleyiciyi kullanarak her şeyi kontrol ederek gezinme yollarını, renkleri, simgeleri ve yazı tiplerini seçin. Uygulamayı oluşturulduğu gibi test edilir ve bilgisayarda, cep telefonunda veya tablette kaydedilen ilerlemeyi izlenebilir. Başvuru tamamen test edildikten ve kullanıma hazır olduktan sonra, kullanıcı verilerini analiz ederek yayınlatabilir ve tanıtılabilir. AppMachine, App Store ve Google Play'de görünmesini sağlayacak tüm özelliklere sahiptir.

Fiyat: Ödeme bir kez yapılır, ilk sürüm 499 \$, tasarımcılar ve geliştiriciler için sürüm maliyeti toplam 1299 dolardır.

3.4.8 GameSalad



Şekil 3.32 GameSalad ana sayfası

GameSalad, iOS, Android, HTML5 ve OS X dahil çeşitli platformlarda oyun oluşturmanıza ve yayınlamanıza olanak tanır. Bu, programlama dili bilgisi olmadan hızlı bir şekilde başlatılabilen sürükle ve bırak arabirimidir. Burada bir harita ve karakter editörü vardır. Tasarımı, becerilerin oluşturulması, resimler, sesler ve karakterlerin navigasyonu çoğu zaman alır. Büyük artılardan birisi yardım; bölümlerine erişebilen aktif forum topluluğunun olmasıdır.

GameSalad kısa süre önce Amazon ile bir araya gelerek, özellikle Kindle Fire platformu için uygulama içi satın alımları GameCircle API ile entegre etmek için özel olarak optimize etmiştir.

Fiyatı: Hem ücretsiz hem de yılda 299 \$'a profesyonel bir sürümü vardır.

3.4.9 BiznessApps



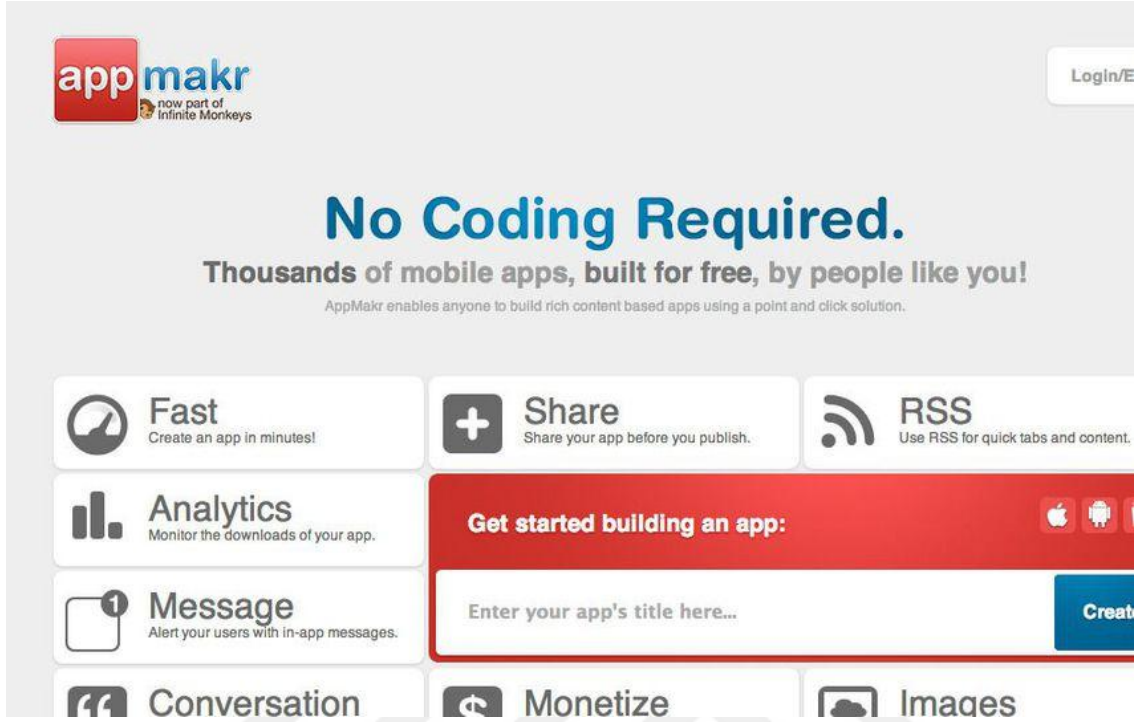
Şekil 3.33 BiznessApps ana sayfası

BiznessApps; Bu, kolayca iş uygulamaları oluşturmanıza yardımcı olan bir platformdur. Yemek siparişi, alışveriş sepeti, sadakat programları, kapsamlı analitik ve çok daha fazlası dahil olmak üzere birçok özellik sunar. Kullanımı kolay içerik yönetim sistemi dakikalar içinde bir uygulama oluşturmanıza ve kendi dahili tasarımınızı kullanarak her şeyi özelleştirmenize olanak sağlar. Ayrıca tasarım ve uygulama geliştirme sürecinin gerçek zamanlı bir önizlemesi de vardır.

Uygulamayı çevrimiçi olarak anında güncelleyebilir ve Apple veya Google'a uzun bir istek göndermeden her şeyi değiştirebilirsiniz. BiznessApps platformunu kullanarak güzel bir mobil uygulamanın nasıl yapılacağına dair haftalık web seminerleri de var ve süreci daha da kolaylaştıran yüzlerce eğitim makalesi var.

Fiyat: mobil site için başlangıç fiyatı ayda 29 dolar; Mobil uygulama için, ayda 59 dolardır.

3.4.10 AppMakr



Şekil 3.34 AppMakr ana sayfası

AppMakr; Programlamaya gerek kalmadan iOS, Android ve HTML5 uygulamaları oluşturmak için kullanılan bir platformdur. 2009 yılında kuruldu ve son zamanlarda dünyanın en büyük geliştirme platformu haline gelen Infinite Monkeys tarafından satın alındı. İstenilen kadar uygulama yapabilir, sınırsız güncelleme yapabilir, bildirim gönderebilir, yüksek çözünürlüklü fotoğraf galerileri ekleyebilir, yeni güncellemeler, müzik ve video yayını, sohbet, Google Haritalar entegrasyonu, genel etkinlik takvimi, bir iç mağaza hizmeti mevcuttur.

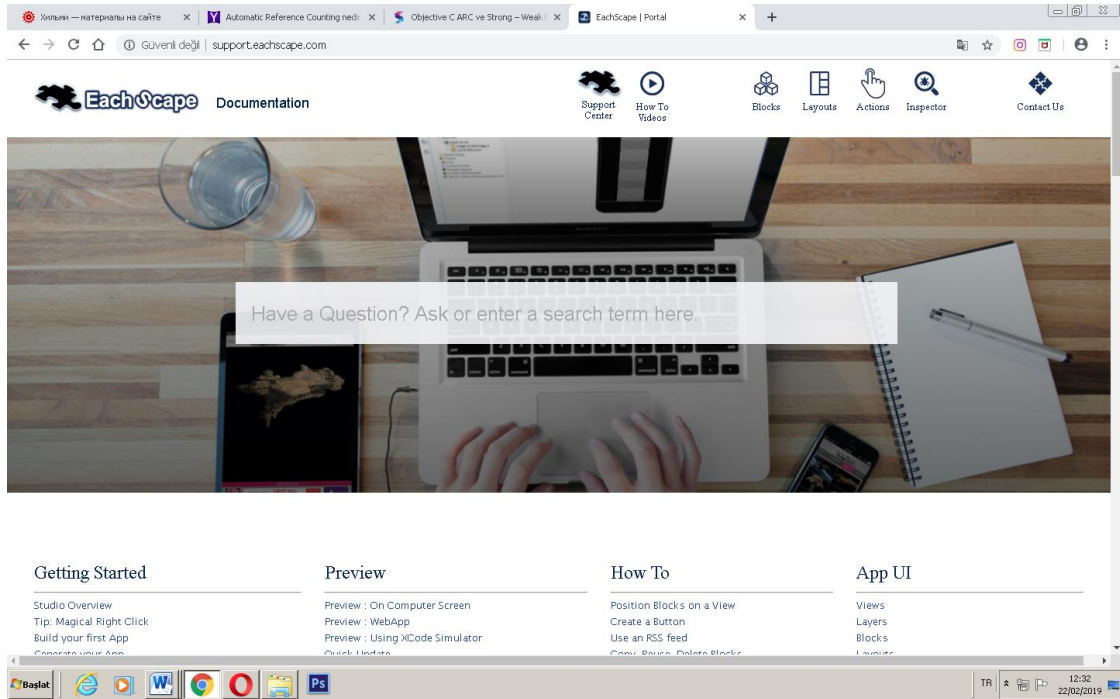
Tüm yayınları takip edebilir, bilgi paneli aracılığıyla uygulamalar, görünümlerini ve işlevlerini kolayca kişiselleştirebilir, sekmeli içerik ekleyebilir, uygulamayı gerçek zamanlı olarak önizleyebilir ve tek tıklamayla sanal mağazalarda yayınlanabilir.

Fiyatı: bir defalık ücretsiz sürümü var fakat yayın maliyeti 99 dolardır.

3.4.11 App Press

Web tabanlı, kod yazmadan uygulama oluşturma imkanı sağlayan App Press platformu iPhone, iPad ve Android ortamlarında geliştirme imkanı sağlıyor. App Press, aylık 30 - 460 dolar arasında fiyatlarla satışa sunulmuş durumdadır.

3.4.12 EachScape



Şekil 3.35 EachScape ana sayfası

Tümleşik, bulut tabanlı, sürükle-bırak yöntemi desteğiyle HTML5 web uygulamalarının yanında iOS ve Android uygulamaları geliştirilmesini sağlayan, iyi bir dokümantasyona sahip bir platformdur. Platform üzerinde geliştirilen uygulamalar için mobil destek hizmetinin sağlanıyor olması, tüm uygulamaların web ön izlemesinin gerçekleştirilebilmesi ve çevrimiçi geliştirme desteği sunulması platformun artıları arasında yer alıyor. Sürükle-bırak özelliğine sahip editör (Cloud Studio) üzerinden iOS, Android ve HTML5 uygulamaların hazırlanmasını sağlayan mimari, arabirim ve eylemlerle birlikte çeşitli bloklardan ve parçalardan oluşur. İleri düzey geliştiricilerin EachScape için yeni bloklar ve kısımlar geliştirebilmesi ve SDK paketlerini kullanabilmesi mümkündür. EachScape, başlangıç olarak ücretsiz kullanılabilir.

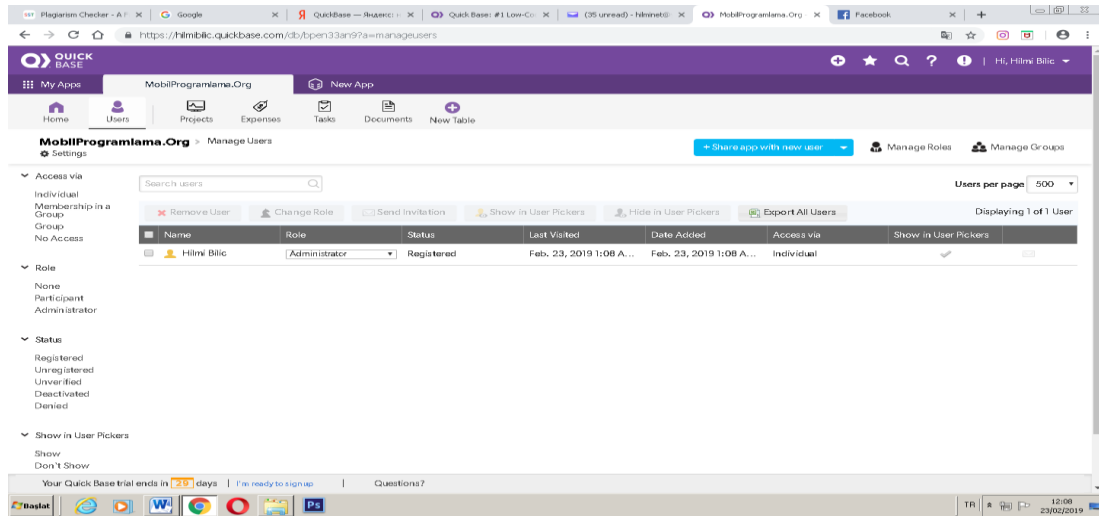
3.4.13 iBuildApp



Şekil 3.36 iBuildApp ana sayfası

iBuildApp, iPhone, iPad ve Android platformları için mobil uygulamaları, HTML5 uygulamaları (ve hatta iPad için) oluşturmak, test etmek, izlemek ve güncellemek için basit ve ucuz bir yol sunan çevrimiçi bir hizmettir. Hizmet, uygulamada metinler oluşturmanıza, RSS beslemelerini, görüntüleri, ses, video ve çok daha fazlasını gömmeye olanak tanır. iBuildApp, Aylık 5.9- 895 dolar arasında fiyatlardadır.

3.4.14 QuickBase



Şekil 3.37 QuickBase çalışma ekranı

QuickBase, web veritabanları ve mobil web veritabanı uygulamalarına yönelik mobil programlar hazırlamaya yarayan ve 800'den fazla özelleştirilebilir uygulama temasının yer aldığı kod yazma gereksinimi olmayan bir çevrim içi platformdur. Websitesi: www.quickbase.com Demo sürümleri ile denemeler yapabileceğiniz site ücretlidir.

3.4.15 Salesforce1 ve Lightning

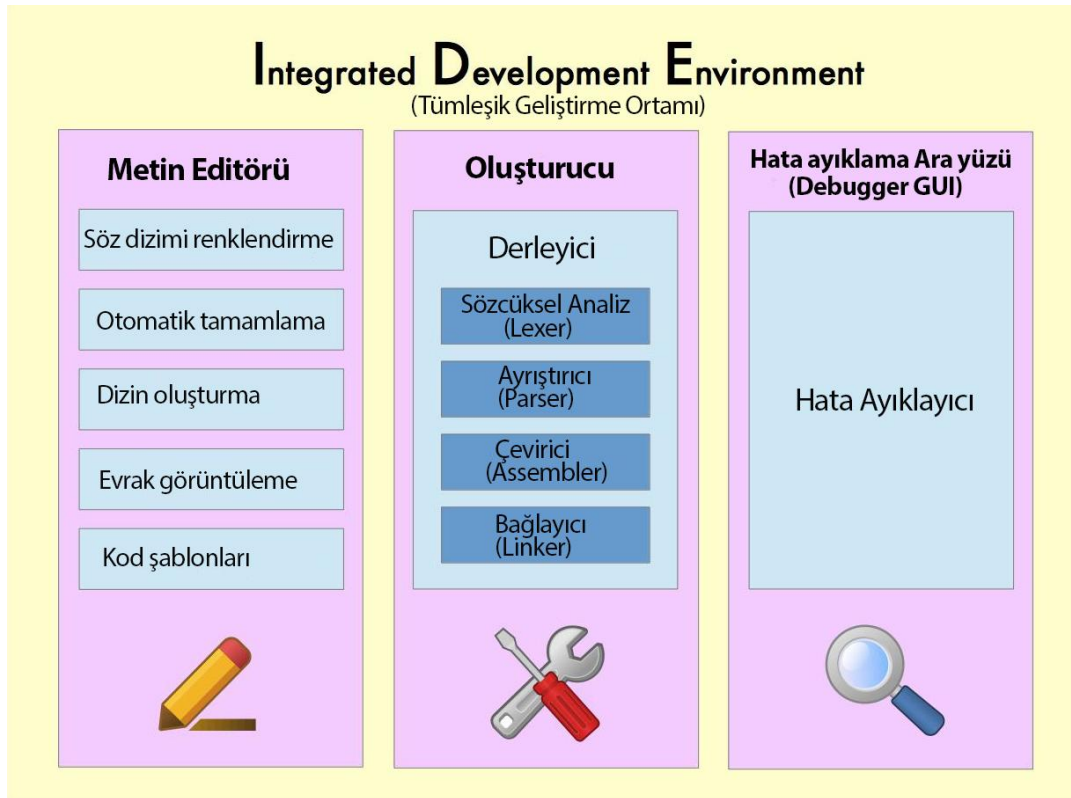
Salesforce bir Müşteri İlişkileri Yönetimi aracıdır (CRM), kuruluşun mevcut müşterileriyle olan ilişkisini takip etmek için kullanılır. Her seviyedeki CRM geliştiricisi Mobile SDK'lar aracılığı ile Salesforce verilerine hem ana uygulama hem de hibrit uygulamalardan erişim sağlanabilmesi ve Mobil Tasarım Şablonlarını (Mobile Design Templates) kullanarak mobil uygulama sayfaları hazırlaması mümkündür. **Lightning** açık kaynak kodlu bir proje olarak yürütülmektedir ve herkese açıktır. Salesforce1, aylık her bir uygulamada kullanıcı başına 25 dolardan başlayan fiyatlarla satışa sunulmuştur.

Neden kullanmalıyım?

1. Her şey bulutlarda çalışır, bu nedenle sunuculara, veritabanlarına ve bunların yönetimine bakmanıza gerek yoktur.
2. Çalışmak için sadece bir tarayıcıya ihtiyaç var, hiçbir şeyin yerel olarak kurulması gerekmiyor.
3. Geliştirme olmadan başlamanıza izin veren birçok işlevsel hazır çözüm.
4. Zengin özelleştirme yetenekleri - tam özellikli Apex programlama dili (Java klonu) ve Visualforce şablon motoru.
5. Erişim hakları (profiller, roller) nesneler ve kayıtlar düzeyinde işlevselliğin farklılaştırılması için güçlü bir sistem.
6. Alışveriş için hazır çözümler.
7. Mobil cihazlarda CRM kullanma platformu - Salesforce1.

3.5 Geliştirme Ortamları (IDE) ve Editörler

Gerçek programcı ne kadar editörlerden bağımsız olsa da kendi yapısına uygun bir yazılım geliştirme editörünü kullanmak durumundadır. Bir programcı, zamanının %80'ini yazılı kodun içinde geçirir. Bu süreyi kısaltmanın yolu da iyi ya da daha doğrusu uygun bir Tümleşik Geliştirme Ortamı (IDE) kullanmaktır.



Şekil 3.38 Tümleşik geliştirme ortamı (IDE) kavramı

Tümleşik geliştirme ortamlarının en temel özellikleri:

- Programcının hızını artırmalıdır.
- Daha kolay programlama ortamı sunmak.
- Programlama diline özgü sözdizimi renklendirmesi yapabilen kod yazım editörü barındırmalıdır.
- Kod dosyalarının hiyerarşik olarak görülebilmesini sağlamak.
- Tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcı birlikte olur.

- Yazılımın derlenmesi, bağlanması kısacası çalışmaya tümüyle hazır hale gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla diğer araçları barındırması.

En bilinen tümleşik geliştirme ortamları: Eclipse, Microsoft Visual Studio, Code: Blocks, Dev-C++, Anjuta, KDevelop, NetBeans, XCode, Android Studio'dur. Çevrim içi olarak çalışan (<https://ideone.com>) IDE dahi vardır.

Editör ise sadece metin oluşturmak için kullanılan programlardır. Programlar bir metindir. Derleme, hata ayıklama, yorumlama özellikleri olmadığı gibi herhangi bir araç kütüphanesine de sahip değildirler. Not defteri, Notepad++ çok kullanılan editörlere örnektir.

3.5.1 Eclipse

Eclipse, C ve Java dili ile yazılan açık kaynak kodlu ve özgür bir IDE (tümleşik geliştirme ortamı) 'dir. Yoğunluk olarak Java ve Java ile ilişkili teknolojiler olsa da, esnek yapısı sayesinde C ve Python gibi farklı diller için de kullanılabilir. Eclipse, Eclipse Vakfı tarafından kontrol edilen ücretsiz ve açık kaynaklı bir yazılım platformudur. Java programlama dilinde yazılmış ve yaratılmasının temel amacı, yazılım geliştirme sürecinin verimliliğini artırmaktır.

Eclipse Platformunun Özellikleri

- Çapraz platform Windows, Linux, Solaris ve Mac OS X işletim sistemleri altında çalışır.
- Eclipse kullanarak, Java, C ve C ++, PHP, Perl, Python, Cobol ve diğerleri gibi çeşitli dillerde programlanabilir.
- Başka araçlar geliştirmek için bir çerçevedir ve modüller oluşturmak için geniş bir API kümesi sunar.
- RCP (Zengin İstemci Platformu) yaklaşımını kullanan Eclipse, neredeyse her türlü istemci yazılımını oluşturmak için bir araçtır. Eclipse kurulum ve kullanımı hakkında bilgi için Ek-13 incelenebilir.

3.5.2 Android Studio

16 Mayıs 2013 tarihinde Google I / O konferansında açıklanan Android platformuyla çalışmak için entegre bir geliştirme ortamıdır.

Android Studio Kurulumu

Bilgisayarda android programlama yapabilmek için JDK (Java Development Kit), Android SDK (Software Development Kit), programları yazmak için bir editör (ör: Eclipse, Android Studio) gereklidir. <https://developer.android.com> adresinden sözü edilen Yazılım geliştirme kitini indirebilir (Çizelge 3.4 Android Studio indirilebilir paketleri) Android yazılım ile ilgili videolar, bloglar, dokümanlar vb. birçok kaynağa ulaşılabilir.

Çizelge 3.4 24.12.2018 tarihinde Android Studio indirilebilir paketleri

Platform	Android Studio paketi	Boyut
Windows (64-bit)	android-studio-ide-183.5452501-windows.exe Recommended	971 MB
	android-studio-ide-183.5452501-windows.zipNo .exe installer	1034 MB
Windows (32-bit)	android-studio-ide-183.5452501-windows32.zip No .exe installer	1034 MB
Mac(64-bit)	android-studio-ide-183.5452501-mac.dmg	1025 MB
Linux (64-bit)	android-studio-ide-183.5452501-linux.tar.gz	1036 MB

Uygun platformu indirdikten ve lisans sözleşmesini onayladıktan sonra programın kurulumu (EK-14) yapılır.

3.5.3 XCODE

Apple firması tarafından geliştirilen macOS, iOS, watchOS ve tvOS platformları için entegre yazılım geliştirme için kullanılan ortamdır. 2001 yılında yayınlanan ilk sürümünden itibaren Mac App Store üzerinden ücretsiz olarak dağıtılır. Şuan 10. Sürümü web sitesinde yayınlanmaktadır (Apple Web sayfası erişim: 14.02.2019).

Objective-c geliştirme ortamı **X-Code** dur. X-Code macOS için tümleşik geliştirme ortamıdır ve sadece MAC OSX işletim sisteminde çalışmaktadır. Sanal makine üzerinden de uygulama çalıştırılabilse de çok güçlü bir makineye sahip değilseniz yavaş ve yetersizdir. Xcode indirme sayfasına tıkladığınızda “Apple Geliştiricisi uygulamasına giriş yapın” uyarısı ile karşılaşılır.

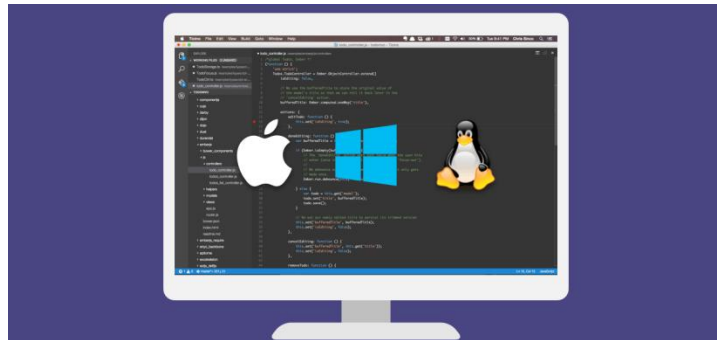
Resmi indirme sayfası: <https://developer.apple.com/xcode/>

iOS işletim sistemleri için native olmayan bir durumda HTML5, CSS ve Java-Script ile geliştirilen Web uygulama halidir. Fakat Windows işletim sisteminde yazsanız bile AppStore’a eklemek için ve derleme yaparken yine Mac bir bilgisayara ihtiyaç duyacaksınız. iOS yazılımları geliştirmeye başladığınızda Android’te yazılım geliştirme ile ilgilendiyseniz simülatörün oldukça hızlı olduğu fark edilecektir. Bunun sebebi ise çekirdek mimari olarak MAC OSX ve IOS’un aynı temelleri kullanmasıdır. Android’de arada bir de Dalvik VM bulunur bu da simülatör olarak Android programlamayı oldukça yavaşlatmaktadır.

3.5.4 Adobe Air

Adobe® AIR® teknolojisi, geliştiricilerin iOS, Android, Windows ve Mac OS gibi büyük platformlar için platformlar arası oyunlar / uygulamalar oluşturmasını ve paketlemesini sağlar.

3.5.5 Visual Studio Code



Şekil 3.39 Visual Studio Code

Linux, OS X ve Windows için kullanılan bir kod editörüdür. Yazılım tarihi boyunca, Microsoft birçok geliştirme aracı yayımladı. Ancak bu, daha büyük ve güçlü bir IDE birleştirmesi oldu. Bu ürün yirmi yıldan fazla bir süredir geliştirilmektedir ve çeşitli işlevleri yerine getirmiştir. Bu araç setini seven birçok kişi ve bazen Visual Studio'nun diğer platformlara aktarılıp aktarılmayacağını da sormuştur.

Visual Studio Code, sadece bir editördür, fakat uzantılara dayanan IDE özelliklerine sahiptir. Zaten, ASP.NET 5 veya Node.js web projeleri oluşturmak için Visual Studio kodu kullanılabilir.

3.5.6 Visual Studio .NET

Microsoft firmasına ait bir yazılım geliştirme ortamıdır. Windows Forms teknolojisinin yanı sıra web siteleri, web uygulamaları, tüm platformlar için yerel ve yönetilen koddaki web hizmetleri gibi web ara yüzlerini de içeren grafik ara yüzü uygulamalar geliştirmenize olanak sağlar. Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Kompakt Çerçeve ve Silverlight tarafından desteklenir. Visual Studio'da platformlar arası programlar için C++ dili ve farklı diller kullanılabilir. Visual Studio kullanılan güncel sürümü Visual Studio 2017'dir. Fakat 2019 ile ilgili haberler Microsoft'un resmi sitesinde duyurulmaktadır. Visual Studio'nun her yeni sürümü genellikle önceki türlerinin projeler, dosyalar ve diğer varlıklardan oluşan dosyaları desteklemektedir.

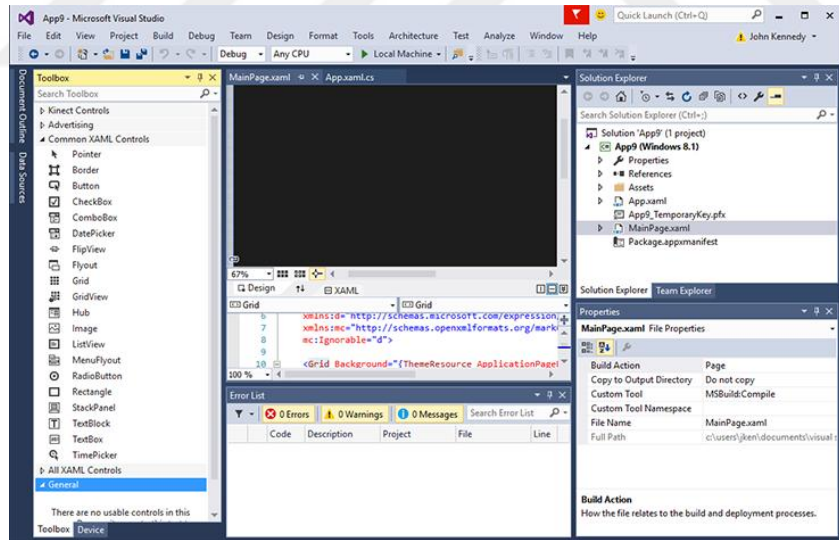
Windows 10 ile birlikte gelen Universal Windows Platform (UWP) uygulama yapısı ile artık farklı platformlarda bile çalışması mümkün olacak. Yani tek bir uygulama bundan sonra telefondan masaüstü bilgisayara, tableten “internet of things” diye bilinen nesnelerin interneti cihazlarına ve hatta oyun konsolu XBOX gibi farklı platformlarda çalışabilecek. Bununla da kalmayıp “Hololens” denilen 3 boyutlu hologram görmenizi sağlayan gözlüklerle de bu uygulamaların çalışması mümkün hale gelecek.



Şekil 3.40 Evrensel Windows platformu UWP

Hangi diller ile uygulama hazırlanabilir?

UWP uygulamalarını C++, C#, Microsoft Visual Basic ve JavaScript kullanarak geliştirebilirsiniz. JavaScript, kullanıcı ara yüzü için HTML5 işaretleme kullanır ve diğer diller, kullanıcı arabirimlerini tanımlamak için *Genişletilebilir Uygulama İşaretleme Dili* (XAML) adlı bir işaretleme dili kullanır.



Şekil 3.41 Microsoft Visual Studio ekranı

3.5.7 MIT APP INVENTOR

MIT App Inventor, Google tarafından geliştirilmiş, şimdilik sadece Android platformları için sürükle-bırak metoduyla program geliştirilebilmesine olanak tanıyan bir blok programlama dilidir. (Uygulama web sitesinde IOS platformlarla ilgili çalışmaların devam ettiği açıklaması yapılmıştır.) Akıllı telefonlar ve tabletler için tamamen işlevsel uygulamalar geliştirilebilmesine olanak tanır. Hemen her yaş gurubunun özellikle çocukların rahatlıkla kullanabileceği sezgisel-görsel bir programlama ortamıdır. Bu uygulama geliştirme ortamına uyum sağlamak ve temel seviyede bir uygulama geliştirmek 30 dakika gibi kısa bir sürede mümkün olabilmektedir. Platformun blok tabanlı olması, geleneksel programlama ortamlarına göre, çok daha az sürede karmaşık ve yüksek etkili uygulamaların oluşturulmasını kolaylaştırır.

App Inventer ile neler tasarlanabilir:

- Oyun
- Eğitim uygulamaları
- Konum belirleme ihtiyacı olan uygulamalar. (GPS)
- Ses Tanıma, Görüntü İşleme, Parmak İzi Okuma ve Yüz Tanıma gibi ileri seviye uygulamalar.
- Mesajlaşma uygulamaları.
- Robotik sistemleri kontrol eden uygulamalar.
- Web etkileşimli uygulamalar. (API)

Platformun artıları:

- Platform çevrimiçi veya çevrimdışı olarak kullanılabilir.
- Hazırlanan uygulamalar kolayca bir android cihazda gerçek zamanlı olarak görüntülenebilir ve çalıştırılabilir.
- Diğer uygulama geliştirme yazılımlarının hemen hemen bütün yeteneklerine sahiptir.

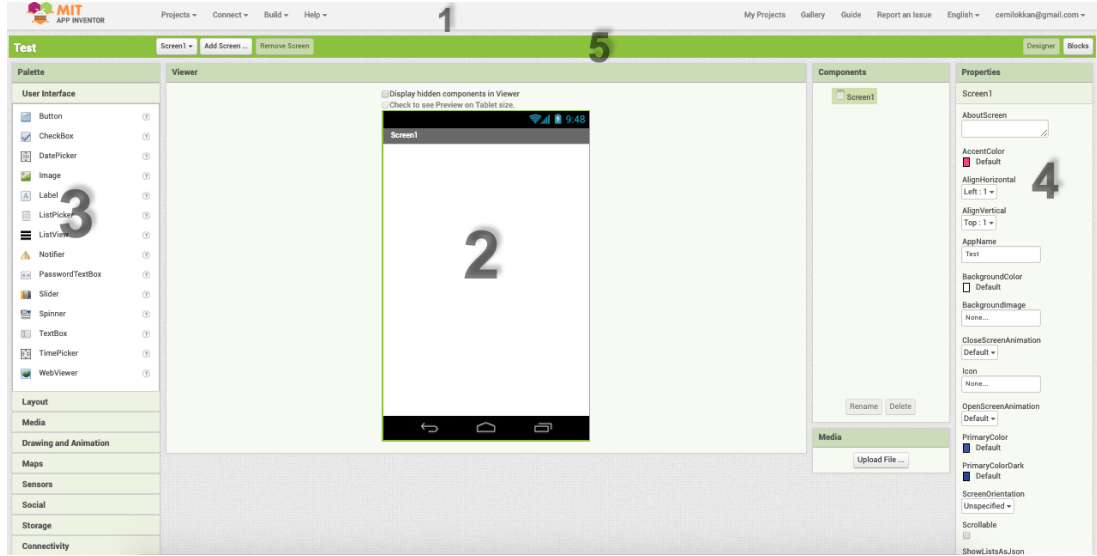
- Kodlama komutlarla değil bloklarla gerçekleştirilir.
- Platformu kullanabilmek için bir Google hesabı yeterlidir.
- Hazırlanan uygulamaların paketlenmesi oldukça hızlı ve kolaydır.
- Bilgisayara indirip birçok örnek uygulamanın bulunduğu bir galeri seçeneği mevcuttur.
- Eklenti özelliği ile uygulamaya istenilen eklentiler kolayca yüklenebilir ve kullanılabilir.
- Uygulama geliştirirken komutların ezbere bilinmesine gerek yoktur. Herhangi bir komut bloğu seçildiğinde kullanılacak komut blokları ekranda görünürdür.
- Hata yapma ihtimali düşüktür. Bloklar uygun olmayan değerlere tanımlandıklarında anında uyarı verirler.
- Öğrenme kolay ve hızlıdır.
- Uygulamanın test edilmesi kolay ve hızlıdır.
- Test aşamasında uygulamada yapılan değişiklikler anında test uygulamasına yansır.
- Tüm özellikler ücretsizdir.

Platformun eksileri:

- Türkçe dil desteği mevcut değil.
- Kompleks uygulamalarda komut bloğu sayısının artması uygulamada hata aramayı zorlaştırır.

App Inventor başlangıç ekranı

App inventor platformunda uygulama geliştirirken iki ekran kullanılır. Bunlardan birincisi “Designer” ekranı diğer ise “Blocks” ekranıdır. Designer ekranında arayüz tasarım işlemleri Blocks ekranında ise kod bloklarının kurulması gerçekleştirilir. Aşağıda ilk açılışta karşımıza gelecek olan “Designer” penceresi görülmektedir.



Şekil 3.42 App Inventor ekranı

1 → Menü Çubuğu: Burada yeni proje oluşturma, hâlihazırdaki projelerin listesini görme, herhangi bir proje dosyasını içe aktarma, hazırlanan uygulamayı emülatör ya da gerçek bir cihaz üzerinde test etme ve paketleme seçeneklerinin yanı sıra galeri ve dil seçenekleri de bu menü içerisinde yer alır.

2 → Screen: Bu kısım görsel öğelerin yerleştirildiği ve düzenlendiği kullanıcı arayüzünün oluşturulduğu bölümdür.

3 → Palet: Bu kısım uygulama geliştirirken kullanılabilecek buton, label, checkbox, media öğeleri gibi temel bileşenlerden gelişmiş uygulamaların ihtiyaç duyabileceği sensör, harita, veritabanı bağlantıları gibi gelişmiş öğeleri barındıran kısımdır. Burada ilgili başlık tıklanarak kullanılabilecek öğelerin listesine erişilebilir. Eklenmek istenilen öğe fare imleciyle sürükleyip bırak yöntemi ile çalışma ekranına taşınır.

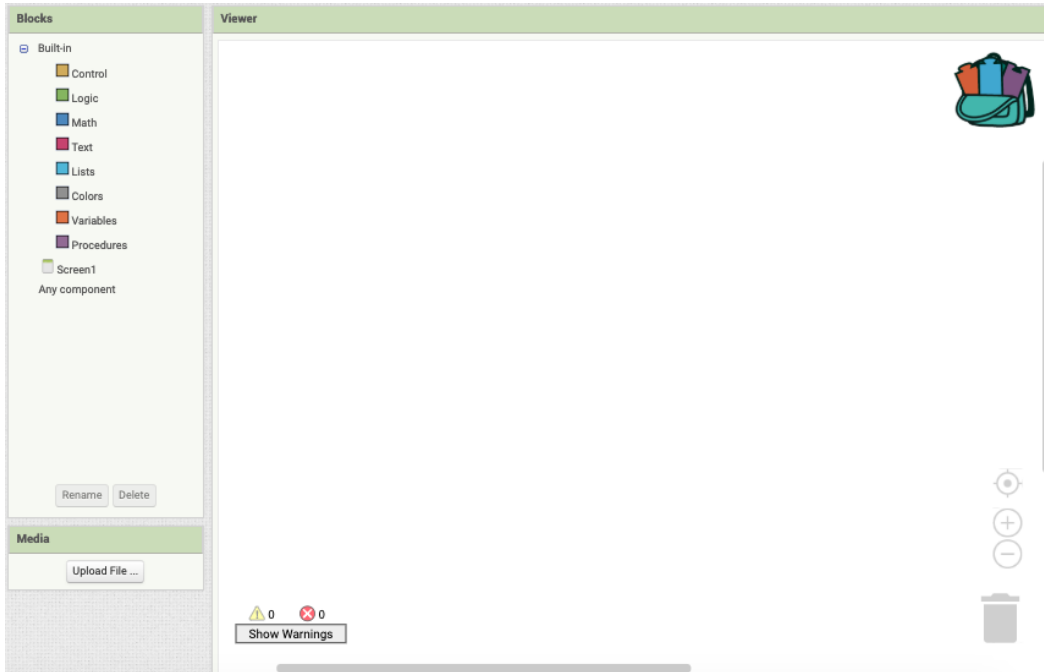
4 → Properties: Bu kısımda çalışma ekranı üzerinde bulunan herhangi bir öğenin değiştirilebilir özellikleri görüntülenebilir ve istenen değişiklikler yapılabilir. Özelliklerine erişilmek istenen öğenin çalışma alanında fare yardımıyla seçilmesi gerekir.

5 → Bu numaralı kısımda uygulamanın ismi, uygulamaya yeni ekranlar ekleme ve

bunlar arasında geiş yapma ve App Inventor’ da tasarım ve kod blokları ekranı arasında geiş yapılabilmesini saėlayan Tasarımcı (Designer) ve Bloklar (Blocks) düğmeleri bulunur.

Bloklar (blocks) ekranı

Blocks ekranına geiş yapıldığında (Şekil 4.52) ekranın sol tarafında yer alan menüde hazırlanan uygulamanın arka planında işlemleri gerçekleştirecek olan komut blokları gruplar halinde yer alır.



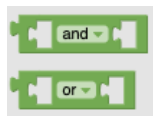
Şekil 3.43 Blocks ekranı

CONTROL: Bu başlık altında döngü, şartlı yönlendirme gibi uygulamanın çalışmasını yönlendiren komut blokları yer alır.



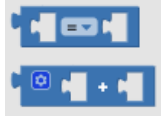
Örnek Blok:

LOGIC: Bu başlık altında true/false ve and/or gibi mantıksal işlemlerde kullanılan komut blokları yer alır.



Örnek Blok:

MATH: Bu başlık altında birçok matematik ve geometri işlemlerinin gerçekleştirilebileceği komut blokları yer alır.



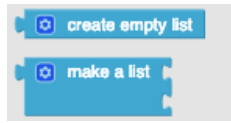
Örnek Blok:

TEXT: Bu başlık altında metin işlemlerinde kullanılan komut blokları yer alır.



Örnek Blok :

LISTS: Liste işlemleri komut blokları yer alır.



Örnek Blok :

COLORS: Renk komut blokları yer alır.



Örnek Blok :

VARIABLES: Değişken tanımlama işlemleri için kullanılacak komut blokları yer alır.



Örnek Blok :

PROCEDURES: Kullanıcı tarafından diğer komut blokları kullanılarak oluşturulan prosedürler burada yer alır.



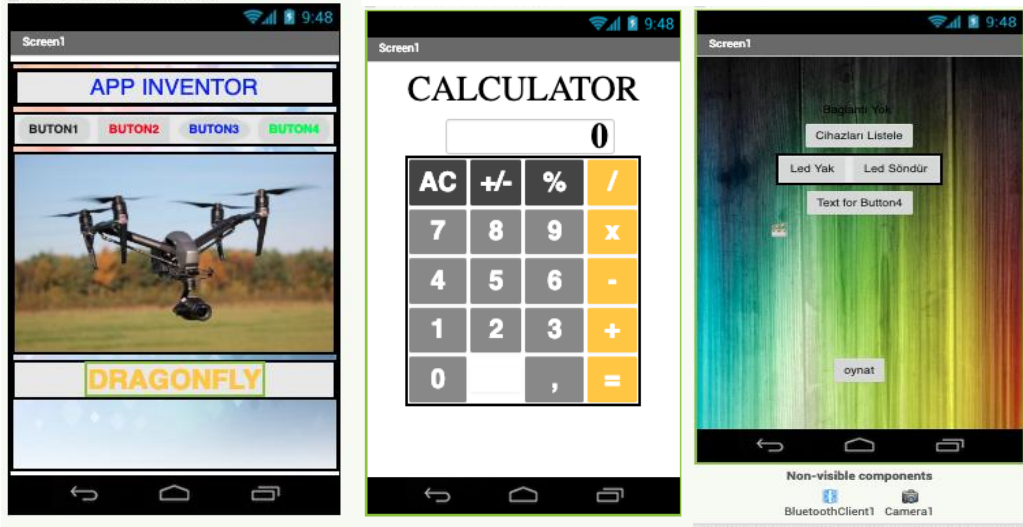
Örnek Blok :

Mit app Inventor uygulama geliştirme ve test aşamaları

1. Ara yüz tasarımı

Arayüz tasarımı Designer penceresinde Palette kutusu altındaki öğeler kullanılarak öğelerin çalışma ekranında istenilen yere konumlandırılarak, properties penceresinden özelliklerinde istenen değişiklikler yapılarak tamamlanır. Ayrıca arayüzde kullanılan öğelerin buton, ekran gibi... resim ataması yapılabilir.

Örnek Arayüz Tasarımları Şekil 3.48’te verilmiştir:

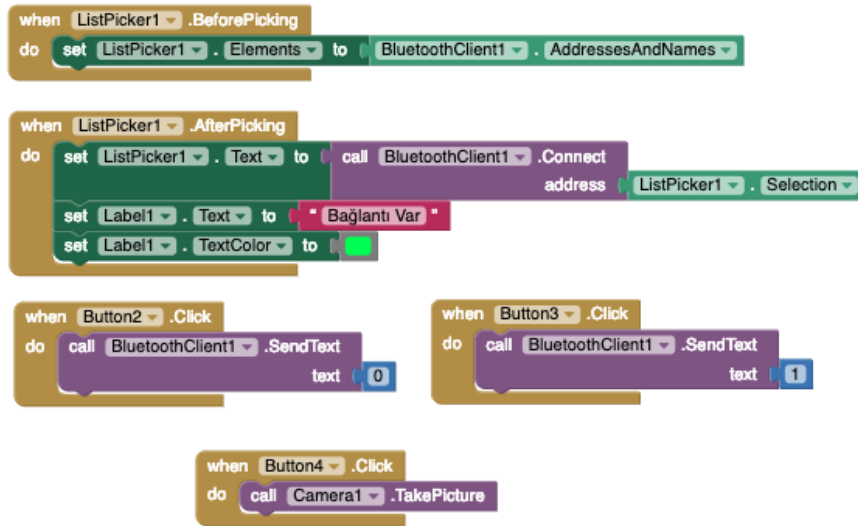


Şekil 3.44 App Inventor örnek tasarımlar

2. Kod bloklarının bileşenlere bağlanması

Arayüz tasarımı bitirildikten sonra “Blocks” ekranı kullanılarak arayüzdeki bileşenlere verilen isimler kullanılarak işlev ataması yapılır.

Örnek Komut Bloğu:



Şekil 3.45 App Inventor komut blokları

Örnek komut bloğunda (Şekil 3.45) görüldüğü gibi komutların belirli bir sıra ile kullanılma zorunluluğu bulunmamakta olup okunması kolaydır ve bu kolaylık programcının komutlara hâkimiyetini artırır.

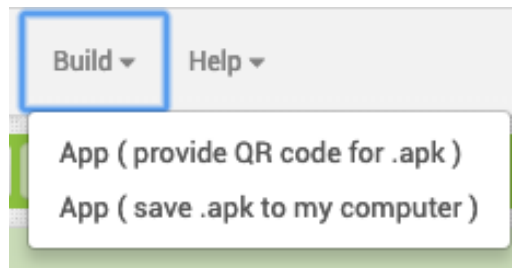
3. Uygulamanın test edilmesi

App Inventor platformunda hazırlanan bir uygulama birden farklı şekilde test edilebilir.

- Dâhili emülatör kullanarak;
- **Menü Çubuğunda>Connect>Emulator**
- Android işletim sistemli cihazınıza MIT AI2 Companion test uygulamasını yükleyerek hazırladığınız uygulamayı telefondan veya tablettten test edilebilir. Aşağıdaki komutu çalıştırdığınızda ekrana bir karekod gelir. Cihazınızdan test uygulamasını çalıştırdığınızda bu karekodu okutabilir veya aynı ekrana gelen kod bilgisini girerek uygulamayı test edilebilir.
- **Menü Çubuğunda>Connect>AI Companion**
- Cihazınızı USB den bilgisayara bağlayarak;
- **Menü Çubuğu>Connect>USB**

4. Uygulamanın paketlenmesi

Yine bu işlemde oldukça kolay ve hızlıdır. Paketleme işlemi Menü çubuğundaki “Build” komutundan yapılır. Buradaki ilk komut paket için 2 saat boyunca geçerli bir karekod üretir ve bu karekod kullanılarak istenilen cihaza yükleme yapılabilir. İkinci komut ise uygulamanın paketlenerek bilgisayara kaydedilmesini sağlar.



Şekil 3.46 Apk uzantılı dosya olarak kayıt etme

Şekil 3.46 da gösterildiği gibi iki türlü. apk çıktısı elde edilebilir. Birincisi QR kod oluşumu, ikincisi bilgisayara. apk olarak kurulum dosyasını kaydetme.

3.6 Kütüphaneler ve Api Kavramı

İyi bir geliştirici asla daha önce yapılmış bir şeyi yapmak için uğraşmaz. Var olanı geliştirmeyi amaçlar. Bu bağlamda Mobil uygulama geliştiricileri, kendi uygulamalarında özgürce kullanabileceği canlı ve güçlü bir özgür yazılım kütüphane eko sistemine sahiptir. Geliştiriciler bu kütüphaneleri diledikleri gibi kullanabilir. Bunun yanında gelişimlerine destek olabilir. Açık kaynak kütüphaneler, yazılımcıların işlerini hızlandırmak ve uygulamalarına işe yarar yeni özellikleri daha kolay eklemek için sıkça başvurdukları kaynakların başında gelir. Mobil yazılım geliştirme ile ilgileniyorsanız ve belli bir seviyeye geldiyseniz, siz de bazı açık kaynak kütüphanelere ihtiyaç duyabilirsiniz.

3.6.1 Kütüphane Kavramı

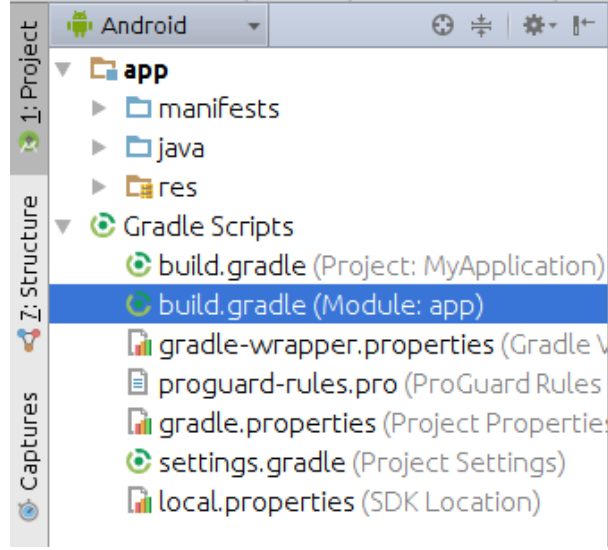
Birçok IDE yazılımcılara sunduğu kütüphaneler ile büyük kolaylıklar sağlıyor. Android Studio da geliştiricilerine birçok kütüphane ile destek veriyor. Küçük ayarlamalarla kütüphaneler projeye uyarlanıp uzun zamanda yapılacak işler kısa sürelere indirilebiliyor.

Nasıl Eklenir: Android Studio’da kütüphane dâhil etmek çok kolaydır. Yapılması gereken tek şey, ilgili kütüphanenin gradle kodunu “build.gradle” dosyasına eklemektir. Bu işlemden sonra eklenen kütüphanenin imkânlarından faydalanmak mümkündür.

3.6.1.1-Android Volley Kütüphanesi

Volley kütüphanesi kısaca sunucuda ki veri tabanı ile bağlantı kurarak veri haberleşmesi yarayan bir HTTP kütüphanesidir. json, xml veri alışverişi yapmasına olanak sağlıyor. Eklemek için build.gradle(Module: app) yazan yere tıklayıp aşağıdaki kod satırını eklenir:

```
dependencies {...  
    Compile 'com.android.volley:volley:1.1.0'  
}
```

Şekil 3.47 Build.gradle modülü

- JSON verilerine erişim, http üzerinden resim indirme veya yükleme, sunucu veri tabanına veri kaydı yapma gibi birçok işlemi kolay bir şekilde yapılır. Hatta indirilen resimler bellekte ya da disk üzerinde cache'lenebilir.
- Tüm networking işlemlerini otomatik olarak zamanlar.
- İstekleri kuyruk yapısı ile öncelik sırasına koyar.
- REST servislerinize bağlanabilir.
- JSON ya da XML cevapları otomatik olarak istediğiniz sınıflara çevrilebilir.
- Aynı anda çoklu bağlantı (multithread) yapar.
- Özelleştirme olanağı sağlar.

3.6.1.2 Picasso

Picasso Kütüphanesi Android uygulamalarında internetten resim indirip kolay yoldan imageviewe atmaya yardımcı olan bir kütüphanedir.

Web sitesi: <http://square.github.io/picasso/>

Ekleme için build.gradle(Module: app) yazan yere tıklayıp şu kod satırını eklenir:
compile 'com.squareup.picasso:picasso:2.4.0'

Görüntü URL'si alınır.

Görüntüyü indirmek için **AsyncTask** oluşturulur.

AsyncTask çalıştırılır.

Sonuç bitmap olarak saklanır. Bu bitmap kaynak olarak ayarlanır. Cache (Önbellek) görüntü saklanır.

```
1 Picasso.with(context).load("http://hilmibilici.com/DvpvklR.png").into(imageView);
```

3.6.1.3 Glide

Android uygulamalara resim eklemek için kullanılan, internetteki bir resmi indirip imageView'de göstermeye yarayan, indirilen görseli önbelleğe alma, yeniden boyutlandırma özelliklerine sahip kütüphanedir. Her hangi bir hata durumunda çıkacak olan görseli belirleme, görseli döndürme, sadece network üzerinden değil cihaz içerisinde bulunan bir görseli de gösterebilme gibi özelliklere sahip olan resim kütüphanesidir. Çizelge 3.5'de karşılaştırılması yapılan bu işlemler Picasso ile de yapılabilir ancak **AsyncTask** ve **Bitmap** işlemleriyle uğraşmak, kodları düzenlemek performans kaybına neden olacaktır.

Çizelge 3.5 Glide Picasso kütüphanelerinin karşılaştırılması

	Glide	Picasso
ListView Performansı	Performansı iyi	Bekleme yapıyor, yüklenirken boş resim görmek hoş olmuyor.
GridView (Image) Performansı	Çok resim bulunan bir activity içindeki grid de performanslı değil.	Performanslı
JPEG	Var	Var
GIF desteği	Var	Yok

3.6.1.4 ButterKnife

Butterknife kısa kodlar ile kod karmaşıklığını daha basite indirgemek ve uygulamaya nesne eklemek için kullanılan bir **View Injector** kütüphanesidir. Kısa çözüm yolları sunar, kod tekrarından kurtarır. AndroidManifest.xml dosyamızda build.gradle kısmına eklenecek kod: `compile 'com.jakewharton:butterknife:7.0.1'`

İlgili web Siteleri: <https://github.com/JakeWharton/butterknife>,
<http://jakewharton.github.io/butterknife/>

3.6.1.5 Gson

GSON kütüphanesi, Google programcıları tarafından geliştirilmiştir ve JSON nesnelerini Java nesnelere dönüştürür ve bunun tersi de geçerlidir.

Ana sayfa: <https://github.com/google/gson> Ekleme için: `compile 'com.google.code.gson:gson:2.2.+'`

3.6.1.6 Retrofit

Veri çekme ve network işlemleri yapmak için kullanılan Volley kütüphanesinden daha performanslı ve hızlı HTTP kütüphanesidir. İkinci sürümü yayınlanmıştır.

Faydaları:

- 1) Artık API isteklerini kodda ayrı bir akışa vermek gerekli değildir - Retrofit bunu kendi yapar.
- 2) Kodun uzunluğunu önemli ölçüde azaltır ve buna bağlı olarak hızlandırır,
- 3) İstekleri dinamik olarak oluşturur,
- 4) JSON'u nesnelere otomatik olarak dönüştürür (Gson kütüphanesini kullanarak)
- 5) Hataları işler
- 6) Dosyaları aktarabilme

Kullanımı: `compile 'com.squareup.retrofit:retrofit:1.9.0'`

3.6.1.7 ActiveAndroid

ActiveAndroid, Android için bir ORM(Object Relational Mapping), SQL sorguları yazmadan bir veri tabanı ile çalışmanıza izin veren bir kütüphanedir. Kütüphane, nesneleri bir tabloya kaydedebilir ve bunları nesneler olarak da alabilir. Projeye ekledikten sonra AndroidManifest.xml dosyasına bildirmek gerekiyor.

3.6.1.8 Dagger 2

Dagger 2, Android ve java için hazırlanan bir dependency injection (bağımlılık enjeksiyonu: bir nesnenin başka bir nesnenin bağımlılıklarını sağladığı bir teknik) kütüphanesidir. Sınıfların birbirine olan bağımlılıklarını azaltmak için kullanılır.

Ana sayfa: <https://google.github.io/dagger/>

Diğer çok kullanılan kütüphaneler ise şunlardır:

Android için:

- 1-FragmentManager
- 2-PRDownloader
- 3-ExpansionPanel
- 4-AnimatedPieView
- 5-Cipher.so
- 6-WindowImageView
- 7-Android-Indefinite-Pager-Indicator
- 8-CalendarPicker

IOS için:

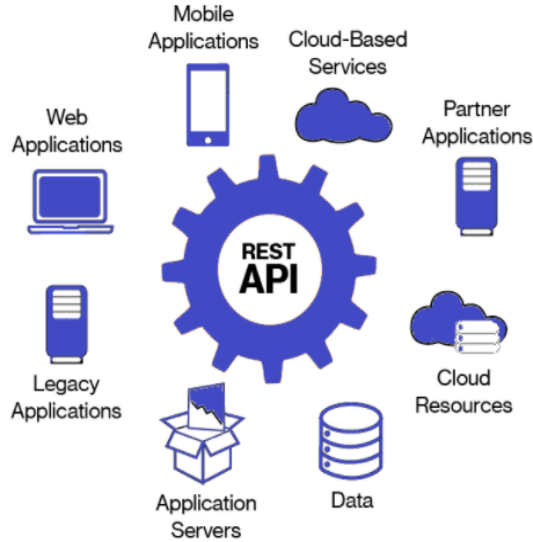
1. IGListKit by Instagram Engineering
2. Realm by realm.io
3. Moya by Ash Furrow
4. SwiftyJSON by Pinglin Tang
5. Valet by Square
6. Firebase Analytics by Google Developers
7. AsyncDisplayKit
8. DZNEmptyDataSet
9. Chameleon by Vicc Alexander
10. PermissionScope by Nick O'Neill

3.6.2 Apiler

Api (Application Programming Interface) çalışılan uygulama ile farklı bir uygulama arasında iletişime geçmesini sağlayan yapıdır. Diğer bir deyişle API, bir uygulamanın sahip olduğu işlevlerin ve yeteneklerin başka bir uygulama tarafından kullanılabilmesine olanak sağlayan bir bileşendir. Örnek vermek gerekirse uygulamanın içine eklenecek Google Haritalar API'si için yöntem şu şekilde olmaktadır:

```
<iframe
  width="600"
  height="450"
  frameborder="0" style="border:0"
  src="https://www.google.com/maps/embed/v1/place?key= Api Key
buraya &q=Merkez mahallesi" allowfullscreen>
</iframe>
```

Örnek bir api uygulaması:



Şekil 3.48 Rest API kavramı

REST(Representational State Transfer) istemci-sunucu arasında hızlı ve kolay bir şekilde haberleşme sağlayan bir servis çözümüdür.

3.7 Mobil Uygulamalarda Veri Tabanı Kullanımı



Şekil 3.49 Mobil uygulamalar için kullanılacak veri tabanları

Son yıllarda akıllı telefonların gelişmesiyle birlikte, geliştiricilerin yaşamlarını kolaylaştıracak, maksimum performans ve kalitelerini sağlayacak birçok araç geliştirilmiştir. Bugün App Store'da lider konumda olmak kolay bir iş değildir. Uygulama milyonlarca kullanıcı tarafından kullanıldığında, uygulamayı izlemek gerekir. Bu nedenle, birçok geliştiricinin karşılaştığı sorunlardan biri Veritabanıdır. Android platformu ya da diğer mobil platformlar gibi veritabanı olarak SQLite kullanımını tercih etse de farklı birçok veri tabanı vardır. SQL komutlarını çalıştırabilmesi ve düşük kapasiteli cihazlarda kolayca çalışabilmesi *SQLite*'ı Android ve iOS platformlarında ilk seçenek olarak akla getiriyor.

3.7.1 SQLite

SQLite, oldukça eksiksiz bir SQL komutları kümesini destekleyen ve kaynak kodlarında (C cinsinden) bulunan gömülü bir platformlar arası veritabanıdır. SQLite açık kaynak kodlu olup, kullanımında hiçbir kısıtlama yoktur. Web sitesi : <http://sqlite.org>

3.7.2 Realm DB

Realm, iOS (Swift & Objective-C'de mevcut) ve Android için çapraz platformlu bir mobil veritabanıdır. Web sitesi : <http://realm.io>

Avantajları:

- Kurulumu kolaydır
- Hızlı
- Çapraz platform
- Ölçeklendirilebilir
- Dokümantasyon
- Güvenilir
- Ücretsiz

3.7.3 ORMLite

Android platformu için ORM (Nesne-İlişkisel Eşleme) yaklaşımının veritabanında çalışmak üzere uygulanmasına izin veren çözümdür.

Avantajları:

- İş için yansıma kullanmaz
- Android için maksimum performans, ORM,
- Minimum bellek kullanımı
- Kütüphanenin küçük boyutunun Android projesinde büyük bir etkisi yoktur.

Dezavantajları:

- Bir şema güncellenirken veri aktarımı için bir mekanizma yoktur.
- Ek açıklama sınıflarının imkânsızlığı ve yeni varlıklar oluşturmak için algoritmayı tanımlamak ihtiyacı vardır.

3.7.4 Berkeley DB

Kütüphane olarak uygulanan yüksek performanslı gömülü veritabanı yönetim sistemidir. Berkeley DB ilişkisel olmayan bir veritabanıdır - anahtar-değer çiftlerini bayt dizileri olarak depolar ve tek bir anahtar için birden çok değeri destekler. Berkeley

DB, çoğu UNIX benzeri sistem, Windows ve aynı zamanda gerçek zamanlı işletim sistemleri de dahil olmak üzere farklı işletim sistemleri altındaki çeşitli donanımlarda 256 terabayt veritabanlarını eşzamanlı olarak yöneten binlerce işlem veya iş parçacığına hizmet verebilir. Berkeley DB Kütüphanesi veritabanlarına çoklu kullanıcı erişimini destekler. Çevrimdışı uygulamalar koduna, bir dizi ortak uygulamaya veya müşteri isteklerini işleyen ve bunlara uygun olarak veritabanı işlemleri gerçekleştiren sunuculara bağlanabilir.

3.7.5 Mongo DB

MongoDB, C ++ dilinde yazılmış ve tablo şemasının açıklamasını gerektirmeyen açık kaynaklı bir belge yönetim veritabanı sistemidir (DBMS). MongoDB, JSON benzeri belgeler içeren bir NoSQL veritabanı olarak sınıflandırılmıştır. Önemli avantajlarından biri, MongoDB işlevselliğinin birkaç fiziksel sunucuya birkaç veritabanı yerleştirmenize izin vermesidir ve bu veritabanları kendi aralarında kolayca veri alışverişinde bulunabilir ve bütünlüğü koruyabilir. Tabii ki bazı küçük kusurları da bulunmaktadır:

- MongoDB belge odaklı bir veritabanıdır, ancak bunun için ödeme yapmak gerekli. Geleneksel DBMS'den farklı olarak, bir katılım işleci yoktur.
- İzolasyon kavramı ve işlem diye bir şey yoktur.

3.8 Mobil Uygulamalarda Resim Kullanımı

Mobil uygulamaların en önemli özelliklerden birisi de nasıl görüldüğüdür. Her ne kadar günümüzde mobil uygulama tasarımı çok zor olmasa da, mobil uygulama geliştiricileri uygulamalarının kullanılabilirliğini arttırmak için tasarıma mutlaka dikkat etmelidir. Platformu ne olursa olsun akıllı telefonlar kullanıcılarına yüksek piksel özellikleri ve çözünürlük sunmalıdır. Mobil uygulama geliştiricileri dokunmatik ekran üzerinde yer alacak uygulama özelliklerini ekranlarda rahat kullanım alanı bırakacak şekilde tasarlamalıdır. Uber'den Vine'a mobil uygulamaların gelişimi sayesinde, program tasarımındaki eğilimler çok hızlı bir şekilde değişiyor. Mobil uygulama tasarımındaki mevcut eğilimler şu şekildedir:

- ✓ **Büyük telefon ekranları:** Gezinme öğelerinin, bir kişinin başparmağıyla kolayca erişebileceği şekilde yerleştirilmesi gerektiği, kullanıcıların yaklaşık% 10'unun solak olduğu gerçeği işi karmaşıklştırsa da uygun ekran çözünürlüklerine göre tasarım yapılmalıdır.
- ✓ **Kaydırma Hareketleri:** Dokunmatik hassasiyetin artması ile büyültme küçültme hareketlerine dikkat etmelidir.
- ✓ **Giyilebilir aksesuarların uygulama tasarımı üzerinde artan bir etkisi:** Akıllı saatlerin küçük ekranları, büyük tablet ekranlarının tam tersi şekildedir. İnceleme firması International Data Corporation, bu yıl (2019) tedarikçilerinin bileğe takılan aksesuarların 45 milyon giyilebilir cihaz tedarik edeceğini öngörüyor. Başlangıçta fitness aleti olarak geliştirilen bu cihazlar, şimdi Apple Watch ve Android Wear'den çıkan tüm büyük pazar alanlarını ele geçiriyor. Giyilebilir cihazlar artık nabız ve uykuyu izleyen mekanizmalara sahip adım sayıcılar değildir. Gerçekten de, kullanıcılara çalışma seyahatinin ne kadar süreceğini, yarın için hava tahminlerini göstermesini ya da eve gelmeden önce su ısıtıcısını kaynatmayı hatırlatmasını sağlayabilen güvenilir akıllı araçlar haline gelmektedir. Ve bu teknoloji oldukça yeni olduğu için kullanıcıların geri bildirimlerine ve isteklerine dikkat etmek ve ilerideki güncellemeler üzerinde çalışmak son derece önemlidir. Firmalar, giyilebilir cihazlar için mobil uygulamaların gelecekteki görünümü hakkında varsayımlarda bulunabilirken, gerçekte tasarımlarını belirleyecek olan kullanıcılar olacaktır.
- ✓ **Katmanlı düzenler** Düz bir tasarımda olduğu gibi, çok katmanlı veya materyal tasarımının elemanları doğal dokular veya bir banal parıltı veya parlaklığı taklit etmek yerine "monolitik" renklerle doldurulmuş doğada geometriktir. Ancak skepomorfizm, doğal dünyadan metaforları, özellikle de nesnelerin birbirleriyle nasıl etkileşimde bulunduğunu ödünç aldı. Bu, gölgelerde ve diğer arka plan ışığı efektlerinde görülür.

Google Materyal Tasarımı İlkeleri: Düz tasarımı doğal fizik ile birleştirerek, kullanıcılar ekrandaki öğelerin nasıl çalıştığını daha sezgisel olarak anlayabilirler.

Cihazınızdaki nesneleri deęiřtirdięinizde, k t leleri ve eylemsizlięi vardır. Ekranda hareket ettirdięinizde veya arkalarındaki elementlere g lge d ř rd klerinde kitle oldukları izlenimini verebilir.

✓ **Daha fazla hareket:** Akıllı telefonlarda, NASA'nın ilk adamı aya g ndermeyi bařarmasının yanı sıra 1997'de satranta Garry Kasparov'u yenen S per Mavi s per bilgisayardakinden daha fazla performans vardır. Ve mobil cihazlar daha da akıllı hale geliyor. Animasyon mobil uygulama tasarımı iin birok avantaj sunar. Kullanıcının dikkatini belirli bir nesneye ekmeye katkıda bulunabilir, gerekli eyleme  nc l k edebilir veya daha keyifli ve řařırtıcı bir kullanıcı deneyimi yaratabilir.

✓ **Basit yumuřak renkler:** Mobil uygulamaların basit bir tasarımı baęlamında, keskin bir renk karřıtlıęının g ze daha fazla dikkat etmesi gereken daha az daęınık bir deneyim seeneęi tercih edilir.

✓ **Tekst re:** Son zamanlarda, mobil uygulamalardaki fontların yalnızca d ř k  z n rl kl  ekranlarda okunabilen g venli web fontlarıyla temsil edildięi anlařılıyor. Son iki yılda, iOS ve Android iřletim sistemlerini daha dinamik,  leklenebilir ve okunabilir yazı tipleri iin optimize ediyor. B y k ekranların ve teknolojik yeniliklerin bir bařka yan etkisi de (TypeKit sayesinde), mobil uygulamalarınızın tasarımına g zellik ve etkililik katmak iin fontların kullanılmasıdır. Metninize arka planda b y k ve modaya uygun bir g r nt  eřlik edip etmedięi veya negatif bořluklarla evrili olmasına bakılmaksızın, yazı tipini fikrinizi g l  bir minimalist biimde iletmek iin kullanılabilir.

✓ **Dahili bulanıklık:** Mobil uygulamaların tasarımındaki trendlerden biri, "yarı saydam" uygulamaların kullanım rahatlıęına katkıda bulunan arka planın bulanıklařmasıdır. Kullanıcı aray z n  kapatan mobil uygulamaların temiz ve b t nsel bir g r n me sahip olmalarına raęmen, yine de kullanılabilirlik sorunlarına neden olabilirler. Mobil cihazlar iin kullanıcı tarafından seilen arka plan resmine, simgelerin ve widget'ların (minik araların) sayısına baęlı olarak, saydamlıęın olmaması uygulamanızı daha zor anlařmaya neden olabilir.

✓ **Mobil uygulamaların uygun tasarımı:** Modern tasarım akımlarının aslında uygulamaları daha kullanıcı dostu hale getirdiği görülmektedir. Geniş elemanlar ve yazı tiplerine sahip basit katmanlar görsel aygıt için daha az çaba gerektirir. Typekit'in, yazı tipi kalitesinden ödün vermeden mobil uygulamaların tasarımında daha canlı metinler kullanmanıza izin vermesinin yanı sıra, programlarımızı görme engelli kişiler için de kullanımını kolaylaştırmaktadır.

✓ **Prototipleme:** Kullandığımız her uygulama bir zamanlar prototip halindeydi. Birçok deney prototipi, birkaç kâğıda basılmış veya PDF olarak kaydedilmiş basit eskizlerle başlamıştır. Artık mobil uygulamaların tasarımı daha karmaşık hale geldiğinden, sadece birkaç statik düzen göstermesi ve müşterinin (veya geliştiricinin) buradaki amaçlanan hareketleri sunması yeterli değildir.

✓ **CSS:** CSS etkileri (gradyanlar, gölgeler vb.) ve CSS animasyonları, genellikle bir resim dosyasının gerektiğinden daha az sayıda bayt ile her çözünürlükte ve züm düzeyinde her zaman net görünen, çözünürlükten bağımsız varlıklar oluşturmak için kullanılabilir. **Web yazı tipleri** metin seçme, arama ve yeniden boyutlandırma yeteneğini korurken güzel yazı biçimleri kullanılabilmesini sağlar. Bu, kullanılabilirlik açısından önemli bir iyileştirmedir.

Vektör resimler, geometrik şekillerden oluşan resimler için idealdir. Vektör resimler züm ve çözünürlükten bağımsızdır. Tarama resimler, birçok düzensiz şekil ve ayrıntı bulunan karmaşık sahneler için kullanılmalıdır.

4. BULGULAR

Bu tez ile popüler mobil uygulama geliştiricilerden Android, OS ve diğer platformlara ait uygulamalar geliştirmek için kullanılan popüler frameworkler (çerçeveler), diller, veritabanları bir arada ve karşılaştırılmalı olarak sunulmuştur.

Bu bölümde anlatımı veya tanımı yapılan mobil geliştirme platformlarının elde edilen sonuçları karşılaştırmalı olarak verilmiştir.

Çizelge 4.1 Çevrim dışı ve içi çerçevelerin karşılaştırılması

	Bilgisayara yüklenen Çerçeveler	Çevrim içi çerçeveler
Ücret	Birçoğu açık kaynak kodlu ve ücretsiz	Hemen hemen tamamı ücretli
Kurulum	Karmaşık yükleme işlemleri var.	Yükleme gerektirmiyor.
Eğitim	Framework'ü kullanabilmek için bir süre alışmak ve kod öğrenmek gerekiyor.	Sürükle-bırak çalışma prensibi var.
Platform desteği	Tüm mobil işletim sistemlerine uygun uygulama geliştirilebilir.	Ağırlıklı olarak Android ve iOS işletim sistemleri üzerinde durulmuş.
Topluluk	Çoğunda büyük topluluk desteği mevcuttur.	Topluluk desteği mevcut değil.
Programlama Dili	HTML, CSS ve JavaScript öğrenmek gereklidir.	Herhangi bir dil öğrenmek gerekli değildir.
Neden seçilmeli	• Çoğu ücretsiz • Tasarım özgünlüğü • Süreklilik • Topluluk • Java Script • Büyük firmaların desteği	• Hızlı prototipleme • Kurulum yok • Dil öğrenme çabası yok • Kolay • Sürükle bırak

Çizelge 4.1 incelendiğinde hızlı ve tek bir uygulama geliştirilmesi gerektiğinde online

mobil çerçevelerin daha uygun olduğu fakat bu işi profesyonelce gerçekleştirmek ve geniş kapsamlı uygulamalar hazırlamak için çevrim dışı çerçevelerin ve programlama dillerinin öğrenilmesi gerektiği görülmektedir.

Ionic - Apache Cordova – PhoneGap platformlar arası Mobil yazılım geliştirme karşılaştırması:

Çizelge 4. 2 Ionic, Cordova ve PhoneGap ortamlarının karşılaştırılması*

	 Ionic	 Apache Cordova	 PhoneGap
Nedir?	Açık kaynak kodlu ve ücretsiz Ionic, yüksek düzeyde etkileşimli uygulamalar oluşturmak için mobil olarak optimize edilmiş HTML, CSS ve JS'ten oluşan bir kütüphane sunar.	Cordova, mobil uygulamaya bir tarayıcı yerleştirerek farklı platformlarda mobil uygulamalar geliştirmenize izin veren bir platformdur.	PhoneGap web tabanlı mobil uygulamalar hazırlamayı sağlayan bir uygulama geliştirme frameworküdür.
Ortak Nokta	HTML, CSS ve JavaScript ile geliştirilmesine olanak tanır.	HTML, CSS ve JavaScript ile geliştirilmesine olanak tanır.	HTML, CSS ve JavaScript ile geliştirilmesine olanak tanır.
Neden seçilmeli?	• Hızlı prototipleme • Ücretsiz • Tasarım • Süreklilik • Java Script	• Eklenti çokluğu • Topluluk • Kolay erişim • Ücretsiz • Java Script	• Adobe tarafından desteklenmesi • Ücretsiz • Kolay • Java Script
Eksiler	• Yüksek performanslı veya kullanıcı arayüzü yoğun uygulamalar için uygun değil • Oyun geliştirme amaçlı değil	• Bir yerel uygulama kadar hızlı değil	• Bir yerel uygulama kadar iyi değil
Alternatifleri	• React Native– React ile yerel (native) uygulama oluşturula bilinir. • Xamarin - C# ile iOS, Android ve Mac uygulamaları oluşturula bilinir. • NativeScript- JavaScript ile gerçek yerel uygulamalar oluşturula bilinir. • Expo-yerel uygulamalar oluşturula bilinir.		

Github ** İstatistikle ri	Yıldız: 36 K Fork*** : 12 K	Yıldız: 713 Fork: 313	Yıldız: 4,2 K Fork: 976
--	--------------------------------	--------------------------	----------------------------

* Karşılaştırma yapılırken <https://stackshare.io> sitesinden faydalanılmıştır

** GitHub, git yazılımı ile entegre olmuş dünyanın çeşitli ülkelerinden, projenize farklı bir kaç kişi ekleyerek takım çalışması yapabileceğiniz bir depolama alanıdır.

*** Fork; projede yapılan değişiklikliği ifade eder.

Çizelge 4.3 Mobil işletim sistemleri kıyaslaması

	Android	iOS	Kaios	Windows Mobile	Diğer
Hedef Kitle *	%74.85	%22.94	0.81%	% 0.28	1.12
Geliştirme Platformları	Android studio, E-Clipse, çerçeveler, online çerçeveler	Xcode, çerçeveler, online çerçeveler	Phyton	.NET Studio	Çerçeveler
Geliştirme dilleri	Kotlin, Java, CSS,	Objective-C, Swift, CSS, Java	CSS, JavaScript, php, phyton, ruby	C#	JavaScript, HTML, CSS
Topluluk	Çerçevelerin topluluk desteği mevcut.	Çerçevelerin topluluk desteği mevcut.			
Destek	Google firması	Apple firması	Google firması	Microsoft firması	
Kullanıcı sadakati **	%70	% 92	-	-	-
Uygulama Geliri ***	\$ 20.1 B	\$ 38.5B			
Geliştirme karmaşıklığı	Hemen hemen her marka ürüne yüklenebiliyor	Sadece Apple firması ürünleri	Alcatel, Nokia, WizPhone .	Microsoft	

*<http://gs.statcounter.com/os-market-share/mobile/worldwide> Erişim: 16/05/2019

**Morgan Stanley Medya raporları

*** 2017 Dünya geneli mağaza uygulama mağazası gelirleri

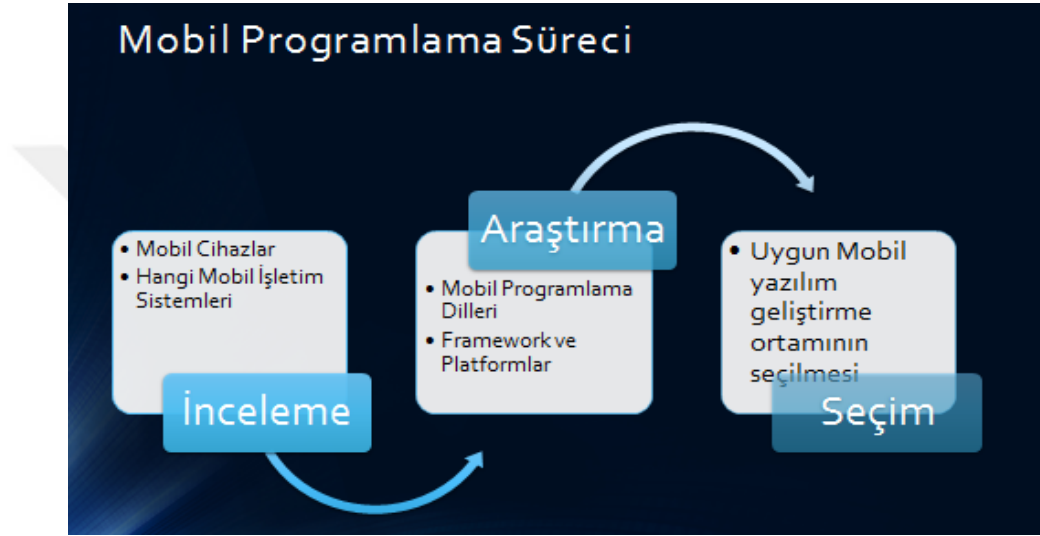
Çizelge 4.4 Flutter ve React Native karşılaştırması

	Flutter	React Native
Yeniden kullanılabilir kod	Flutter, kodun üzerine yazılmasına olanak sağlar. Kodu geri dönüştürmeyi planlıyorsanız, Yeniden Kullanılabilirliğe izin verdiği için Flutter iyi bir seçenektir.	React Native, kodu yeniden kullanmaya izin verir, ancak birkaç temel bileşenle sınırlıdır. Platformları stilize etmek React Native ile çok daha fazla zaman alır.
Üçüncü Parti Kütüphaneler	Çok sayıda üçüncü taraf paketi mevcuttur, Flutter topluluğu etkindir.	React Native popüler bir çerçeve olduğundan, bazı dinamik özellikler ekleyerek uygulamanın kullanılabilirliğini artırmak için daha fazla üçüncü taraf paketine sahiptir.
Popülerlik ve Kapsam	30k Github yıldızıyla, Flutter sektörde trend olarak gösteriliyor. Google, çerçeveyi tanıtırken, çerçevenin geliştiricinin topluluğu arasında popüler olmasını sağlamak istiyor.	Native'i, 65k Github yıldızı, web geliştirici dili JavaScript ile birlikte React Kütüphanesi ile geliştiriciler arasında daha popüler hale getiriyor.
Topluluk	Flutter sektörde yeni olduğu için, mobil uygulamayı geliştirmeye başlamak için bazı yararlı kaynaklar olan çevrimiçi olarak mevcut çok sayıda öğretici yazı ve kılavuz bulunmaktadır.	React Native sektörde popüler ve eski olduğundan, çok sayıda topluluk ve çevrimiçi destek alıyor. JavaScript konusunda uzmanlığa sahip yetenekli geliştiriciler, geliştirme süreciyle ilgili herhangi bir soruna anında çözüm bulabilmenizi sağlar.

4.1 Mobil Uygulama Geliştiricilerinin Dikkat Etmesi Gereken Noktalar

Uygulamayı bilgisayarınızda ve çevrim içi olarak hazırlarsanızda son kullanıcıya ulaşana kadar bazı adımlardan geçer. Yayınlama veya satış için gerekli adımlar bu bölümde anlatılmıştır.

4.1.1 Mobil Uygulama Geliştiricilerinin Dikkat Etmesi Gereken Noktalar



Şekil 4.1 Mobile programlama süreci

Mobil Geliştirme Platformları için Dikkat Edilecek Noktalar

1. **Programlama dili:** Çok fazla entegre geliştirme ortamları (IDE) vardır. Mobil geliştirme platformları genellikle bir programlama diline özgüdür. Nihai ürün alıcıları genellikle mobil uygulamalarını oluştururken kullanmayı düşündükleri dile bağlıdır. Çok dilli destek sunan birçok platformlar olmasına rağmen, amaçlanan kodlama dilini bilmek platform seçeneklerinizi etkili bir şekilde kullanmayı sağlar. Doğru dil ve platform seçimi çok önemlidir.
2. **Back-end - front-end (Arka uç ve ön uç):** Bir geliştirici olarak role bağlı olarak, ön uç ya da arka uç geliştirmeye özgü bir platform düşünülebilir. Amaç ya kullanıcı ara yüzüne hâkim olmak ya da veri erişim katmanları oluşturmaksa, arama odaklanmış platformlar veya muhtemelen hizmet (MBaaS) ürünü olarak

mobil bir arka uç oluşturmak için daralır. Ön uç ve arka uç entegrasyon özelliklerine sahip bir platform arıyorsanız, arama daha da daralır.

3. **Platformlar arası (Cross-platform):** Uygulamanın dağıtım yöntemi, bir mobil geliştirme platformu seçilmesinde kilit bir faktördür. Birçok ürün, özellikle iOS veya Android gibi yerel bir işletim sisteminde çalışır. Diğer ürünler, uygulamaların çeşitli cihazlarda çalışmasına izin veren platformlar arası dağıtım özelliklerine sahiptir. Ürünün sonunda belirli bir işletim sisteminde mi yoksa birden fazla işletim sisteminde mi çalışacağını bilmek, sizi bir mobil geliştirme platformu seçmeye daha da yaklaştırmalıdır.
4. **Ölçeklenebilirlik:** Arka uca özel veya tümleşik platformlara bakıyorsanız, ölçeklenebilirlik daha ağır basabilir. Geliştiriciler genellikle uygulamalarını bir platformda derler ve sonunda ölçeklenebilir bir platforma aktarırlar. Ancak bazı platformlar, cihazlar arasında ve bulut depolamada yerel özellikler olarak işlev görmesi için ölçeklenebilirlik olanakları sunar. Yerel bir geliştirme kiti seçerseniz, geliştiricilerin uygulamalarını benimsemeyi umdukları her cihaz için yeniden yazmaları gerekir. Ancak bazı geliştiriciler yalnızca tek aygıt işlevselliğini arzu edebilir. Her iki durumda da, ölçeklenebilirlik gereksinimlerinizi bilmek, Mobil Geliştirme Platformu Aramanızı daraltmanıza yardımcı olacaktır.
5. **Emülatör kullanılabilirliği:** Emülatörler, uygulamanızın işlevselliğini test etmede kilit bir bileşendir. Birçok mobil geliştirme platformu, pek çoğu olmasa da, mobil testler için yerleşik emülatörler sunar. Geliştirme platformunda emülatör sunmayanların bazıları, uygulamaları test etmek için üçüncü taraf emülatör araçlarını gerektirir. Bazıları simülatör sunar, fakat emülatör değil. Simülatörler uygulamanızın davranışını çoğaltmaya çalışır, öykünücüler ise bir uygulamanın iç işlevlerini ve genel davranışını çoğaltır. Emülatörler sıklıkla tercih edilir, çünkü yeniden oluşturma tam olarak uygulamanın çalışacağı gibi davranır. Simülatörler uygulamanın fonksiyonlarını taklit ederek, aslında çalıştırmanın aksine taklit eder.
6. Hız önemlidir. Giriş animasyonlarını kısa tutmak ve gereksiz kodlamayı ve kullanılmayan öğeleri çıkartmak hızı artırmayı sağlar.

7. Kullanılan grafikler yazılım ile uyumlu olmalıdır.
8. Uyumu korumak, son kullanıcı testlerini iyi yapmak lazımdır.
9. Cihazların güvenliğine dikkat etmelidir. Gereksiz kod parçaları, gereksiz zamana bu da ısınmaya sebep olabileceği için kaldırılmalıdır.
10. Reklam kullanımına dikkat edilmelidir.

4.1.2 Mobil Geliştirme Platformlarının Temel Faydaları

Mobil Geliştirme Platformları ile şunlar yapılabilir:

- Mobil uygulamalar yazılır, yönetilir ve dağıtımı yapılır
- Geliştirme sürecini hızlandırmak için API'ler, çerçeveler ve araçlar kullanılır
- UI / UX gelişimi için tasarım araçlarını kullanılır
- Emülatörler ve simülatörler aracılığıyla mobil uygulamaları her cihazda test edilir
- Arka uç (Back-end) yetenekleri yönetilir ve kontrol etme
- Veri saklama ve şifreleme olanakları,
- Mobil uygulamaları platformlar ve cihazlar arasında ölçeklendirme,
- Depolama, test etme, hata ayıklama ve diğer görevler için üçüncü taraf araçlarla entegre edebilme imkanları vardır.

5. TARTIŞMA, SONUÇ VE ÖNERİLER

Bu araştırmada mobil cihazlar için kullanılan 9 işletim sistemi ve sistemlere programlama yapabilen 13 programlama dili 7 IDE ve editör, 18 çerçeve, 15 çevrim içi çerçeve, 9 kütüphane ve 5 veritabanı incelenerek kurulumları, öğrenme olanakları ve süreleri, avantajları, dezavantajları, karşılaştırılmalı olarak yer verilmiştir.

Bu çalışma, günümüz toplumlarının yaşam, haberleşme ve hatta eğlence alışkanlıklarının değişiminde büyük paya sahip olan mobil cihazlara uygulama geliştirme sürecini öğrenmek isteyen kişilerin nereden başlayacaklarını detaylı olarak incelemektedir. Araştırmada akıllı telefonların, tabletlerin, diğer taşınabilir aygıtların ve dolayısıyla tüm mobil uygulamaların nasıl hazırlanacağı konusunda ki eğilimlerin belirlenmesi ve en uygun seçeneğin belirlenmesini amaçlamaktadır.

Ortaya konan bu çalışma, mobil uygulama hazırlama yaklaşımlarının değerlendirilmesi ve seçilmesi açısından önem arz etmektedir. Bu bağlamda, mobil uygulama hazırlamayı düşünenler öncelikle hedef kitlenin beklentisini doğru analiz etmelidir. Bunun neticesinde mobil uygulama geliştiricileri uygun platformu ve dili seçmelidir. Mobil uygulamanın güvenilirliği, doğru çalışması, hızlı yayılımı için öncelikle dilin, çerçevenin ve hedef kitlenin doğru tespit edilmesi büyük önem taşımaktadır.

Özetle, genç nüfusa sahip bir ülke olarak Türkiye mobil uygulama geliştirme pazarında başarılı olması için yazımcıların tüm taşınabilir cihazları, özellikleri, mobil işletim sistemlerini, uygulama geliştirme için kullanılan dilleri ve platformları iyi tanıması gerekmektedir.

6. KAYNAKLAR

- Burnette E. (2009), Hello, Android, Introducing Google's Mobile Development Platform, 30-40
- Charland, A., & LeRoux, B., (2011). Mobile application development web vs. native. *Communications of the ACM Magazine*, **54**(5), 49-53.
- Darwin, F. I.(2015) Android Cookbook, Problems and solutions for android Developers, 73-79,
- Fodor A.G., Covaci B. V.(2016) E-learning World (*eLearning Mobile App for Android and Ios*) **2** 10-18.
- Narman A. E., İ. (2017). Android Studio ile Programlama, **13**: 259-263. İmran Ö. Swift ile İOS Programlama Temelleri 1: 20-50
- Michael M., (2009) Why Develop for Android? Android Overview and Android UI, 48-68
- Özdamar Keskin, N. ve Kılınç, H. (2015). Mobil öğrenme uygulamalarına 68 yönelik geliştirme platformlarının karşılaştırılması ve örnek uygulamalar. *AUAd*, **1**(3), 68-90.

İnternet Kaynakları

- 1) ABI. (2014). Q4 2013 Smartphone OS Results: Is Google Losing Control of the Android Ecosystem? <https://www.abiresearch.com/press/q4-2013-smartphone-os-results-is-google-losing-con/> Erişim: 10.05.2019
- 2) Klein, D. (2012). “How to decide: Mobile websites vs. mobile apps”. <http://web.archive.org/web/20120213091207/http://www.adobe.com/newsletters/inspire/february2012/articles/article7/index.html?trackingid>, Erişim: 20.05.2019
- 3) Android Auto Complete Text View Design Resmi destek sitesi, <http://developer.android.com/reference/android/widget/AutoCompleteTextView.html>, Erişim 10.10.2018
- 4) “Bilgi Teknolojileri ve İletişim Kurumu Pazar verileri raporu” <https://www.btk.gov.tr/uploads/pages/slug/2017-q3-5a5f1e2baba7b.pdf> Erişim: 02.05.2019
- 5) “Global digital report 2018” <https://wearesocial.com/blog/2018/01/global-digital-report-2018> Erişim:12.12.2018
- 6) “Mobile & Tablet Operating System Market Share Turkey” <http://gs.statcounter.com/os-market-share/mobile-tablet/turkey> Erişim:14.12.2018
- 7) “İnternet kullanımı ve sosyal medya istatistikleri” <https://dijilopedi.com/2018-internet-kullanimi-ve-sosyal-medya-istatistikleri/> Erişim: 28.11.2018
- 8) “Google I/O is an annual developer festival” <https://events.google.com/io2017/> Erişim: 17.12.2018
- 9) “TIOBE İstatistikleri” <https://www.tiobe.com/tiobe-index/> Erişim:17.12.2018
- 10) Tyson M., “What is the JVM? Introducing the Java Virtual Machine” <https://www.javaworld.com/article/3272244/core-java/what-is-the-jvm-introducing-the-java-virtual-machine.html> Erişim: 27.12.2018
- 11) “Geleceği yazarlar platformu” <https://gelecegiyazanlar.turkcell.com.tr/konu/app-inventor>, Erişim: 10.12.2018
- 12) “Android™ Notes for Professionals book” <https://books.goalkicker.com/AndroidBook>, Erişim: 21.10.2018
- 13) Alkan D., “Java Programlama Dili” <https://denizalkan.com.tr/tum-kategoriler/java-programlama-dili/>, Erişim: 08.12.2018

- 14) “BlackBerry Keys Order Form”
<https://www.blackberry.com/SignedKeys/codesigning.html>, Erişim: 24.01.2019
- 15) “Android API düzeyleri” <https://docs.microsoft.com/tr-tr/xamarin/android/app-fundamentals/android-api-levels?tabs=windows>, Erişim: 02.02.2019
- 16) “Ext JS 6.0.0 Classic Toolkit” https://docs.sencha.com/ext/6.0.0/guides/tutorials/login_app.html, Erişim: 03.03.2019
- 17) Merjevich V., “Делаем приложение на Corona SDK”
<https://webref.ru/mobile/corona/>, (Rusça) Erişim: 05.02.2019
- 18) Değirmenci E., “Design Patterns (Tasarım Kalıpları)”
<https://medium.com/nsistanbul/design-patterns-factory-method-pattern-615457e9560b>, Erişim: 08.02.2019
- 19) Mucuk M., “React Native Nedir?” <https://medium.com/turkce/react-native-nedir-cd2ec1e38154>, Erişim: 10.04.2019
- 20) “React Native Elements” <https://react-native-training.github.io/react-native-elements/>, Erişim: 11.04.2019
- 21) Saring J., “11 React Native Component Libraries you Should Know in 2019”
<https://blog.bitsrc.io/11-react-native-component-libraries-you-should-know-in-2018-71d2a8e33312> Erişim: 05.05.2019
- 22) Choosing a Programming Language for Windows Mobile Development
[https://docs.microsoft.com/en-us/previous-versions/bb677133\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/bb677133(v=msdn.10)) Erişim: 06.05.2019
- 23) Kedit Part 1: Configuring Android Studio with Kotlin
<https://android.jlelse.eu/learn-kotlin-while-developing-an-android-app-part-1-e0f51fc1a8b3> Erişim: 10.05.2019
- 24) How to Create a RESTful API For Your Mobile App (Fast) – BuildFire
<https://buildfire.com/create-restful-api-mobile-app/> Erişim: Mayıs 2019
- 25) Посторонним В.: Пишем приложение для Android в Appcelerator (Titanium Studio) <http://blog.sci-smart.ru/2012/10/android-appcelerator-titanium-studio.html>
Erişim: 12.05.2019
- 26) Stefanova C., Top 15 Mobile Game Engines & Development Platforms / Tools in 2018
<https://thetool.io/2018/mobile-game-development-platforms>, Erişim: 22.06.2019

ÖZGEÇMİŞ

Adı Soyadı : Hilmi BİLİCİ
Doğum Yeri ve Tarihi : Elbistan / 10.04.1981
Yabancı Dili : Rusça, İngilizce
İletişim (Telefon/e-posta) : 0505 235 75 85 / hilminet@yahoo.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Elbistan Sağlık Meslek Lisesi, (1995-1999)
Lisans : Kırgızistan Türkiye Manas Üniversitesi, Bilgisayar
Mühendisliği Bölümü, (1999-2004)

Diğer konular :

EKLER

Bu bölümde anlatımı yapılan çerçevelerin(frameworklerin), programların kurulumları ve kullanımı hakkında ek bilgiler verilecektir.

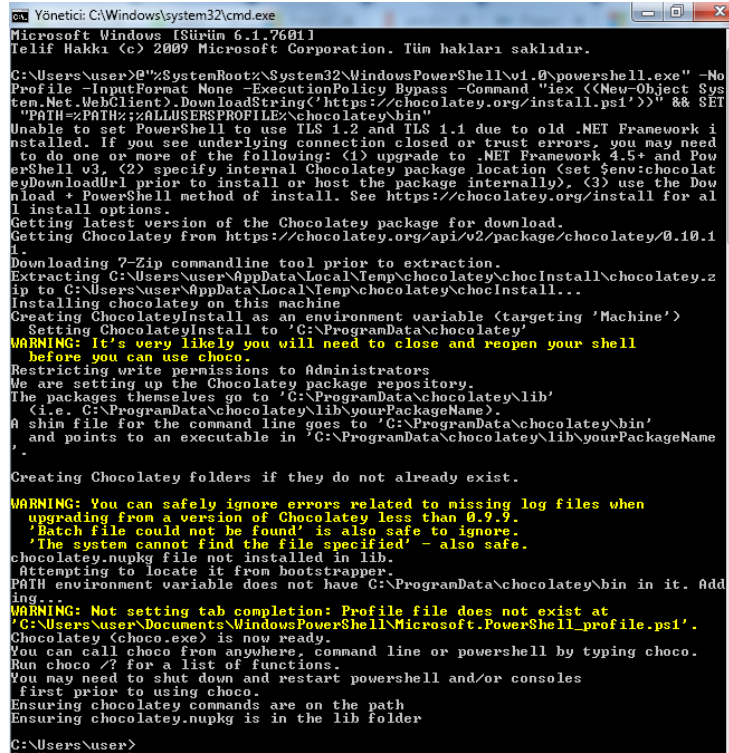
EK-1 NativeScript Kurulum adımları

NativeScript'i Windows makinenize ayarlamak için aşağıdaki adımları izlenir: (<https://docs.nativescript.org> sitesindeki İngilizce kurulum makalesinden istifade edilmiştir. Erişim: 25.01.2019.)

- Gereksinimlerin kurulumunu ve yapılandırmasını kolaylaştırmak için Chocolatey'i (<https://chocolatey.org/install>) kurulur.
- Komut istemini Yönetici olarak çalıştırılır.

Aşağıdaki betiği kopyalayıp komut istemine yapıştırın, kurulum gerçekleşecektir.

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command "iex  
(  
(new-object  
net.webclient).DownloadString('https://chocolatey.org/install.ps1'))"  
&& SET PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin
```



```
Yönetici: C:\Windows\system32\cmd.exe  
Microsoft Windows [Sürüm 6.1.7601]  
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.  
  
C:\Users\user>@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -No  
Profile -InputFormat None -ExecutionPolicy Bypass -Command "iex (<(New-Object Sys  
tem.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET  
"PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"  
Unable to set PowerShell to use TLS 1.2 and TLS 1.1 due to old .NET Framework i  
nstalled. If you see underlying connection closed or trust errors, you may need  
to do one or more of the following: (1) upgrade to .NET Framework 4.5+ and Pow  
erShell v3, (2) specify internal Chocolatey package location (set $env:chocolat  
eyDownloadUrl prior to install or host the package internally), (3) use the Dow  
nload + PowerShell method of install. See https://chocolatey.org/install for al  
l install options.  
Getting latest version of the Chocolatey package for download.  
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.1  
1.  
Downloading 7-Zip commandline tool prior to extraction.  
Extracting C:\Users\user\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.z  
ip to C:\Users\user\AppData\Local\Temp\chocolatey\chocInstall...  
Installing chocolatey on this machine  
Creating ChocolateyInstall as an environment variable (targeting 'Machine')  
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'  
WARNING: It's very likely you will need to close and reopen your shell  
before you can use choco.  
Restricting write permissions to Administrators  
We are setting up the Chocolatey package repository.  
The packages themselves go to 'C:\ProgramData\chocolatey\lib'  
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).  
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'  
, and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName  
'.  
Creating Chocolatey folders if they do not already exist.  
  
WARNING: You can safely ignore errors related to missing log files when  
upgrading from a version of Chocolatey less than 0.9.9.  
'Batch file could not be found' is also safe to ignore.  
'The system cannot find the file specified' - also safe.  
chocolatey.nupkg file not installed in lib.  
Attempting to locate it from bootstrapper.  
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Add  
ing...  
WARNING: Not setting tab completion: Profile file does not exist at  
'C:\Users\user\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.  
Chocolatey (choco.exe) is now ready.  
You can call choco from anywhere, command line or powershell by typing choco.  
Run choco /? for a list of functions.  
You may need to shut down and restart powershell and/or consoles  
first prior to using choco.  
Ensuring chocolatey commands are on the path  
Ensuring chocolatey.nupkg is in the lib folder  
  
C:\Users\user>
```

Şekil 6. 1 Chocolatey komut satırında kurulumu

Google Chrome'u yükleyin (NativeScript uygulamalarında hata ayıklamak için gereklidir). Komut isteminde aşağıdaki komutu çalıştırın:

choco install googlechrome -y

3. Node.js LTS'yi (LTS: Long-Term Support- Uzun süreli destek) yüklenir. Komut isteminde şu komut çalıştırılır: *choco install nodejs-lts -y*

```
C:\Users\user>choco install nodejs-lts -y
Chocolatey v0.10.11
Installing the following packages:
nodejs-lts
By installing you accept licenses for the packages.
Progress: Downloading nodejs-lts 10.15.0... 100%
nodejs-lts v10.15.0 [Approved]
nodejs-lts package files install completed. Performing other installation steps.
Installing 64 bit version
Installing nodejs-lts...
```

Şekil 6.2 Node.Js kurulumu

4. JDK 8 ayarlanır:

Komut isteminde aşağıdaki komutu çalıştırın.

choco install adoptopenjdk --version 8.192

```
Yönetici: C:\Windows\system32\cmd.exe
C:\Users\user>choco install adoptopenjdk --version 8.192
Chocolatey v0.10.11
Installing the following packages:
adoptopenjdk
By installing you accept licenses for the packages.
Progress: Downloading adoptopenjdk 8.192... 100%
adoptopenjdk v8.192 [Approved]
adoptopenjdk package files install completed. Performing other installation steps.
The package adoptopenjdk wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script? (Y/N/No/Pl/print): y
Downloading adoptopenjdk
from 'https://github.com/AdoptOpenJDK/openjdk8-binaries/releases/download/jdk8u192-b12/OpenJDK8U-jdk_x64_windows_hotspot_8u192b12.zip'
Progress: 100% - Completed download of C:\Users\user\AppData\Local\Temp\chocolatey\adoptopenjdk\8.192\OpenJDK8U-jdk_x64_windows_hotspot_8u192b12.zip (72.92 MB).
Download of OpenJDK8U-jdk_x64_windows_hotspot_8u192b12.zip (72.92 MB) completed.
Extracting C:\Users\user\AppData\Local\Temp\chocolatey\adoptopenjdk\8.192\OpenJDK8U-jdk_x64_windows_hotspot_8u192b12.zip to C:\Program Files\AdoptOpenJDK...
C:\Program Files\AdoptOpenJDK
PATH environment variable does not have C:\Program Files\AdoptOpenJDK\jdk8u192-b12\bin in it. Adding...
Environment Vars (like PATH) have changed. Close/reopen your shell to see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of adoptopenjdk was successful.
Software installed to 'C:\Program Files\AdoptOpenJDK'
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
C:\Users\user>
```

Şekil 6.3 JDK kurulumu

5. Android SDK yüklenir.

Komut isteminde aşağıdaki komutu çalıştırılır.

```
choco install android-sdk -y
```

Komut istemini yeniden başlatılır ve **Android SDK Kurulumu** yapılır.

6. Android SDK Platformu 28, Android SDK Oluşturma Araçları 28.0.3 veya sonraki sürümler, Android Destek Deposu, Google Deposu ve ihtiyaç duyabilecek diğer tüm SDK'lar için tüm paketleri yüklenir. Alternatif olarak, gerekli tüm paketleri kuracak olan aşağıdaki komutu kullanabilir.

```
"%ANDROID_HOME%\tools\bin\sdkmanager" "emulator" "platform-tools"  
"platforms;android-28" "build-tools;28.0.3" "extras;android;m2repository"  
"extras;google;m2repository"
```

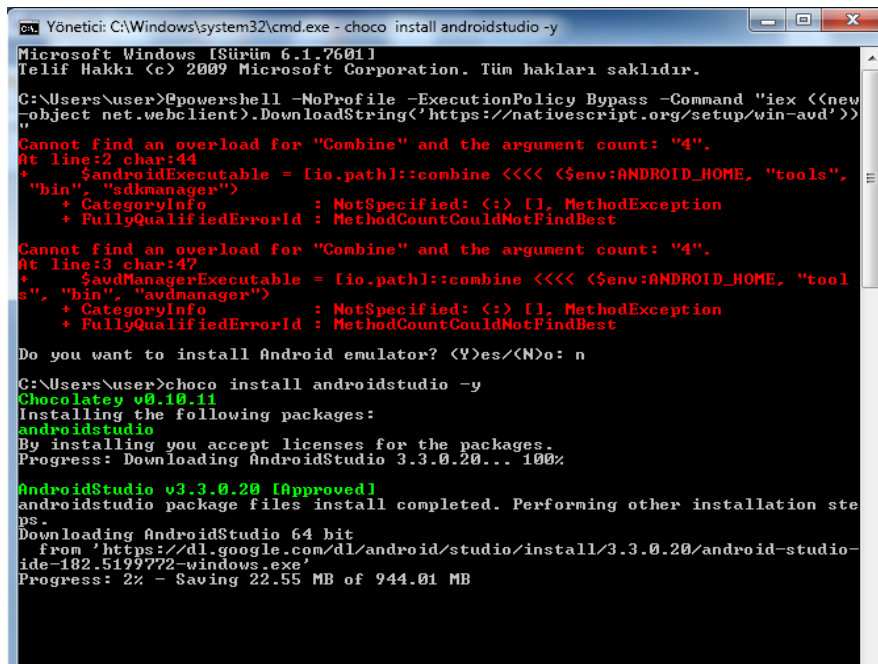
7. Android sanal aygıtlarını (AVD'ler) yükleyin. Bunu yapmanın birden fazla yolu var, o yüzden sadece birini seçilir:

a. Yönetici olarak Komut İstemi'nde aşağıdaki komutu uygulayın:

```
@powershell -NoProfile -ExecutionPolicy Bypass -Command "iex ((new-object net.webclient).DownloadString('https://nativescript.org/setup/win-avd'))"
```

b. Android Studio'yu kullanın ve Android sanal cihazlarını (AVD'ler) oradan yükleyin:

Komut isteminde Yönetici olarak aşağıdaki komutu çalıştırarak Android Studio'yu yüklenir: `choco install androidstudio -y`



```
Yönetici: C:\Windows\system32\cmd.exe - choco install androidstudio -y  
Microsoft Windows [Sürüm 6.1.7601]  
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.  
C:\Users\user>@powershell -NoProfile -ExecutionPolicy Bypass -Command "iex ((new-object net.webclient).DownloadString('https://nativescript.org/setup/win-avd'))"  
Cannot find an overload for "Combine" and the argument count: "4".  
At line:2 char:44  
+ $androidExecutable = [io.path]::combine <<<< ($env:ANDROID_HOME, "tools",  
+ "bin", "sdkmanager") : NotSpecified: (<:) [], MethodException  
+ CategoryInfo          : NotSpecified: (<:) [], MethodException  
+ FullyQualifiedErrorId : MethodCountCouldNotFindBest  
Cannot find an overload for "Combine" and the argument count: "4".  
At line:3 char:47  
+ $avdManagerExecutable = [io.path]::combine <<<< ($env:ANDROID_HOME, "tool  
s", "bin", "avdmanager") : NotSpecified: (<:) [], MethodException  
+ CategoryInfo          : NotSpecified: (<:) [], MethodException  
+ FullyQualifiedErrorId : MethodCountCouldNotFindBest  
Do you want to install Android emulator? (Y)es/(N)o: n  
C:\Users\user>choco install androidstudio -y  
Chocolatey v0.10.11  
Installing the following packages:  
androidstudio  
By installing you accept licenses for the packages.  
Progress: Downloading AndroidStudio 3.3.0.20... 100%  
AndroidStudio v3.3.0.20 [Approved]  
androidstudio package files install completed. Performing other installation steps.  
Downloading AndroidStudio 64 bit  
from 'https://dl.google.com/dl/android/studio/install/3.3.0.20/android-studio-  
ide-182.5199772-windows.exe'  
Progress: 2% - Saving 22.55 MB of 944.01 MB
```

Şekil 6.4 Android Studio Kurulumu

8. NativeScript CLI'yi yükleyin. Aşağıdaki komutu çalıştırılır: `npm i -g nativescript`

9. Sistemin doğru yapılandırılıp yapılandırılmadığını kontrol etmek için şu komut çalıştırılır: `tns doctor`

"Hiçbir sorun tespit edilemedi" seçeneğini görülürse artık kodlamaya başlanabilir.

```
Your components are up-to-date:
nativescript,tns-core-modules,tns-android,tns-ios
No issues were detected.
C:\Users\>
```

Şekil 6.5 Tns doctor

Eğer herhangi bir hata mesajı alırsa “tns doctor” komutu yardımı ile eksik kalan yeri yapılandırabilir.

Proje oluşturma:

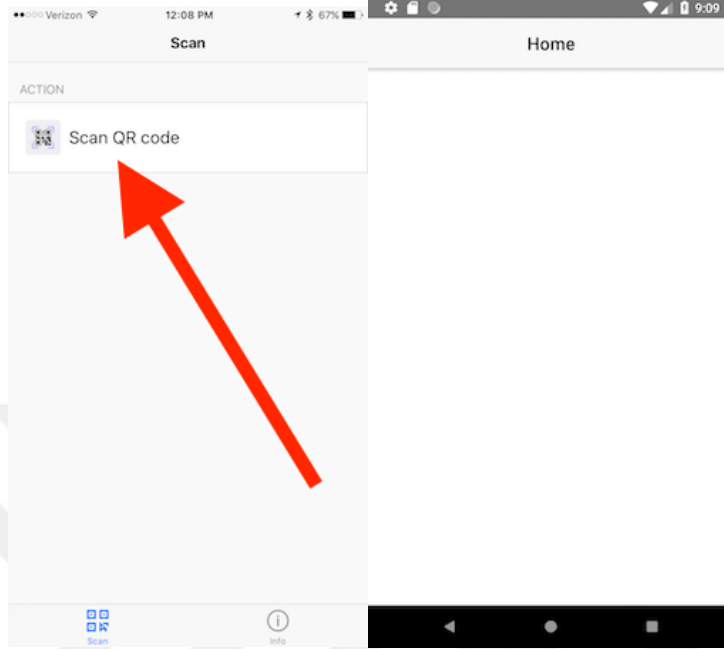
NativeScript'te, NativeScript CLI'yi kullanarak uygulamalar oluşturup çalıştırınız. Kurulumdan sonra, NativeScript CLI, terminalinizde veya komut isteminizde komut olarak bulunur; tns, Telerik Native Script 'in kısaltılmasıdır. Terminal veya komut isteminizi açın ve yeni bir NativeScript uygulaması oluşturmak için şu komutu çalıştırılır: `tns create HelloWorld --template tns-template-blank`

Artık bir uygulama oluşturduğuna göre, yeni uygulamayı nasıl çalıştırılacağını ve bir cihazda nasıl çalıştığını görelim.



Şekil 6.6 QR kodu oluşumu

Cihaza NativeScript Playground ve NativeScript Privew uygulamalarını kurduktan sonra QR kodunu tara seçeneğine dokununuz ve terminalinizde veya komut isteminizde görünen QR kodunu tarayınız.



Şekil 6.7 Telefonda QR code tarama

Uygulamaya basit bir resim eklenmesi; livesync'in nasıl çalıştığı görülür. Uygulamanın app/main-page.xml ya da (C:\Users\user\HelloWorld\app\app-root.xml) dosyasını açın ve içeriğini aşağıdaki kodla değiştirilmesi gereklidir:

```
<Page loaded="pageLoaded">
  <ActionBar title="My App" class="action-bar"></ActionBar>
  <Image src="https://user-images.githubusercontent.com/544280/42960643-66d498ac-8b5a-11e8-8946-7224eefea6a5.jpg"></Image>
</Page>
```

Emülatör yeni görüntüyü yenilemeli ve görüntülemelidir:



Şekil 6.8 Emülatörden gelen myApp çıktısı

Sonra, uygulamanın app/app.css dosyasını açın ve aşağıdaki kodu dosyanın altına yapıştırın.

```
@keyframes spin {  
  from { transform: rotate(0); }  
  to { transform: rotate(360); }  
}  
Image {  
  animation-name: spin;  
  animation-duration: 3s;  
  animation-iteration-count: infinite;  
  animation-timing-function: linear;  
}
```

Emülatör yenilendiğinde çılgınca dönen bir elma görünmelidir:

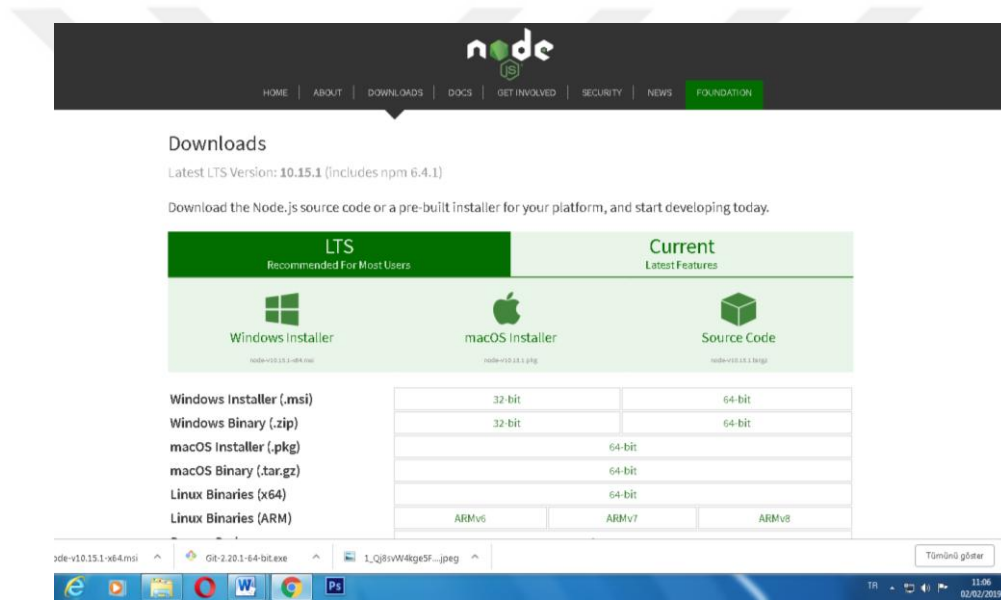


Şekil 6.9 Güncellenmiş myApp çıktısı

EK-2 Cordova yükleme adımları

- 1) <https://git-scm.com/downloads> adresinden işletim sistemi için istenilen yükleme seçeneğini seçilir.
- 2) Node.js: Node.js, javascript üzerinde çalışan bir web uygulamasıdır. Geliştiriciler tarafından araçları ve kitaplıkları dağıtmak için kullanılan Node Package Manager veya npm adlı bir paket yöneticisi ile birlikte gelir. Cordova ve eklentileri npm kullanılarak dağıtılmıştır, bu yüzden yüklenmelidir.

Resmi Yükleme Sitesi: <https://nodejs.org/en/download/>



Şekil 6.10 Node Js yükleme ekranı

- 3) Java: Yerel Android uygulamaları Java ile yazılmıştır, bu yüzden yüklenmelidir. Java'yı yüklemek için indirme sayfasına (<https://java.com/en/download/>) gidilir ve ücretsiz olarak Java İndir düğmesine tıklayın, işletim sistemi için istenilen yükleme seçeneği seçilir. İşletim sisteminizin yükleyicisini indirip Java yüklenir.
- 4) Android SDK: Uygulamayı bir Android cihaza dağıtılacağından, Android Yazılım Geliştirme Kitini (SDK) kurmak gerekiyor. Git Android Studio ve SDK Araçları (<https://developer.android.com/studio>) ve Tek işletim sisteminiz için SDK Araçları

indirin. Windows için, gerekli paketleri seçmek ve yüklemek için kullanabilecek hazır bir yükleyici vardır.

- 5) Cordova yükleme: Gerekli bileşenleri bilgisayarınıza yükledikten sonra, Cordova'nın kurulumuna devam edebilirsiniz. Aşağıdaki komutu komut satırından çalıştırılır: *sudo npm install -g cordova*

Ya da Komut satırı kullanarak ikinci yöntem:

Kurulum için komut: *npm install -g cordova*

Proje için Komut: *cordova create merhaba com.ornek.merhaba merhabaDunya*

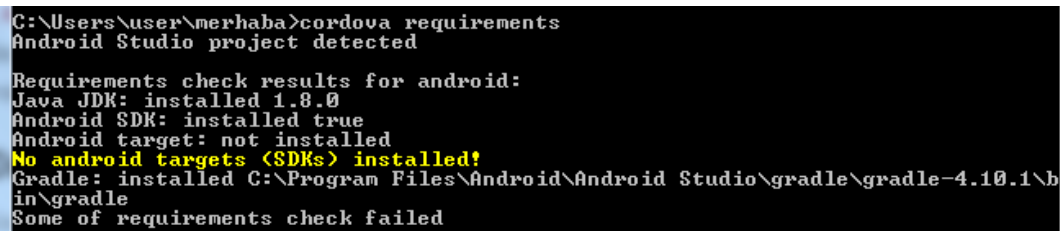
Proje klasörüne girme: *cd merhaba*

Klasör içeriğini görme komutu: *dir*

Platformların Eklenmesi

Projenin hangi platformda/işletim sisteminde çalışacağını belirlemek için ilgili platform adı bu kodlar yardımı ile eklenir:

- Android işletim sistemi için: *cordova platform add android*
- OS işletim sistemi için: *cordova platform add ios*
- Platform kontrol komutu: *cordova platform ls*
- Derleme için ön gereksinim kontrolü komutu: *cordova requirements*



```
C:\Users\user\merhaba>cordova requirements
Android Studio project detected

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed true
Android target: not installed
No android targets (SDKs) installed!
Gradle: installed C:\Program Files\Android\Android Studio\gradle\gradle-4.10.1\bin\gradle
Some of requirements check failed
```

Şekil 6.11 Platform kontrolleri

Platform derlemesi yapmak için gerekli kod: *cordova build*

Uygulamanın Test edilmesi: *cordova emulate android*

```
Yönetici: C:\Windows\system32\cmd.exe - cordova build
C:\Users\user\merhaba>cordova build
Android Studio project detected
ANDROID_HOME=C:\Android\android-sdk
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_201
studio
Welcome to Gradle 4.10.1!

Here are the highlights of this release:
- Incremental Java compilation by default
- Periodic Gradle caches cleanup
- Gradle Kotlin DSL 1.0-RC6
- Nested included builds
- SNAPSHOT plugin versions in the 'plugins {}' block

For more details see https://docs.gradle.org/4.10.1/release-notes.html

Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details

BUILD SUCCESSFUL in 24s
1 actionable task: 1 executed
Subproject Path: CordovaLib
Subproject Path: app
Downloading https://services.gradle.org/distributions/gradle-4.1-all.zip
.....
```

Şekil 6.12 Gereksinim kontrolleri

EK-3 Titanium'u yükleme adımları:

Öncelikle Appcelerator'un sitesine gidilerek (www.appcelerator.com) bir hesap oluşturulur. Download Titanium Studio başlığı altındakilerden işletim sistemine uygun paketi indirilebilir ya da komut tabanlı kurulum yapıla bilinir.

- **Node.js**-<https://nodejs.org/#download> sitesinden indirip yükleyin
- **JDK**-<https://appcelerator.com/jdk> adresinden yükleyin
- **Android**- Android platformu için geliştiriliyorsa, Android için Kurulum kılavuzunu kullanın
- **Xcode**- Mac kullanıcıları için, Apple App Store'dan XCode'u yükleyin. İstendiğinde tüm ek bileşenleri açın ve yükleyin
- **Windows platformu**-Windows Platformu için Kurulum kılavuzunu kullanın

Komut Tabanlı Ara yüz (CLI) Kurulum

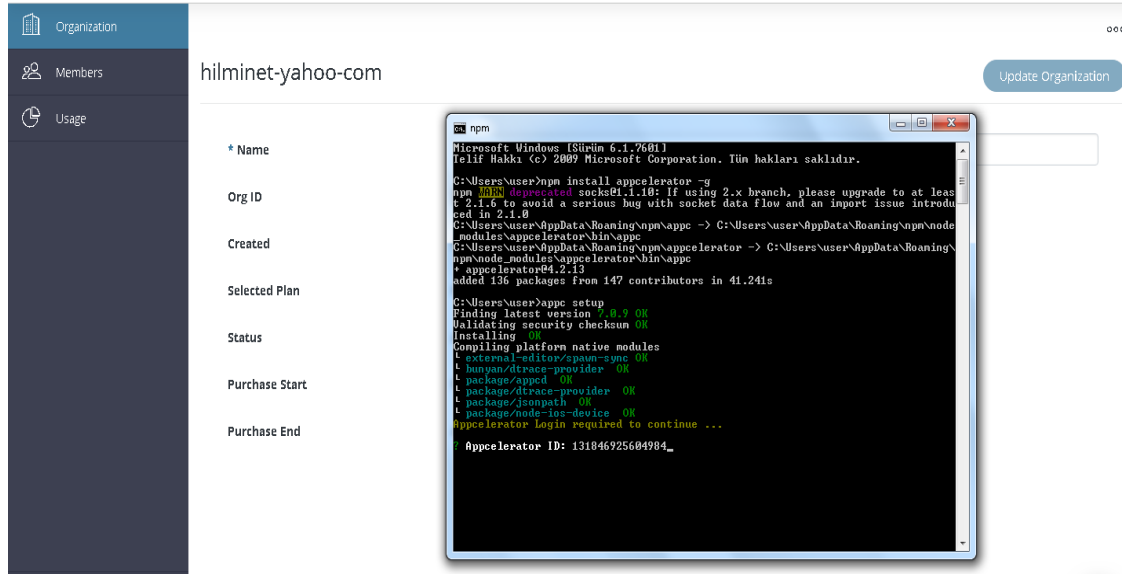
- 1- Appcelerator indirilmesi : `npm install appcelerator -g`

```
Yönetici: C:\Windows\system32\cmd.exe - appc setup
Microsoft Windows [Sürüm 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\user>npm install appcelerator -g
npm WARN deprecated socks@1.1.10: If using 2.x branch, please upgrade to at least 2.1.6 to avoid a serious bug with socket data flow and an import issue introduced in 2.1.0
C:\Users\user\AppData\Roaming\npm\appc -> C:\Users\user\AppData\Roaming\npm\node_modules\appcelerator\bin\appc
C:\Users\user\AppData\Roaming\npm\appc -> C:\Users\user\AppData\Roaming\npm\node_modules\appcelerator\bin\appc
+ appcelerator@4.2.13
added 136 packages from 147 contributors in 41.241s
```

Şekil 6.13 Appcelerator indirme

2- Kurulum: *appc setup*



Şekil 6.14 Appcelerator kurulum

3- İlk uygulama oluşturma:

appc new --type app --name MyFirstApp --id com.appcelerator.MyFirstApp

4- Uygulama klasörüne girme: *cd MyFirstApp*

5- Android için Çalıştırma: *appc run --platform android*

Windows Tabanlı Kurulum

Öncelikle Git ve Node.js kurulu olmalıdır. Değilse Studio Oracle JDK ile birlikte bunları da yüklemeniz için sizi uyarır.

Titanium Studio Başlatma

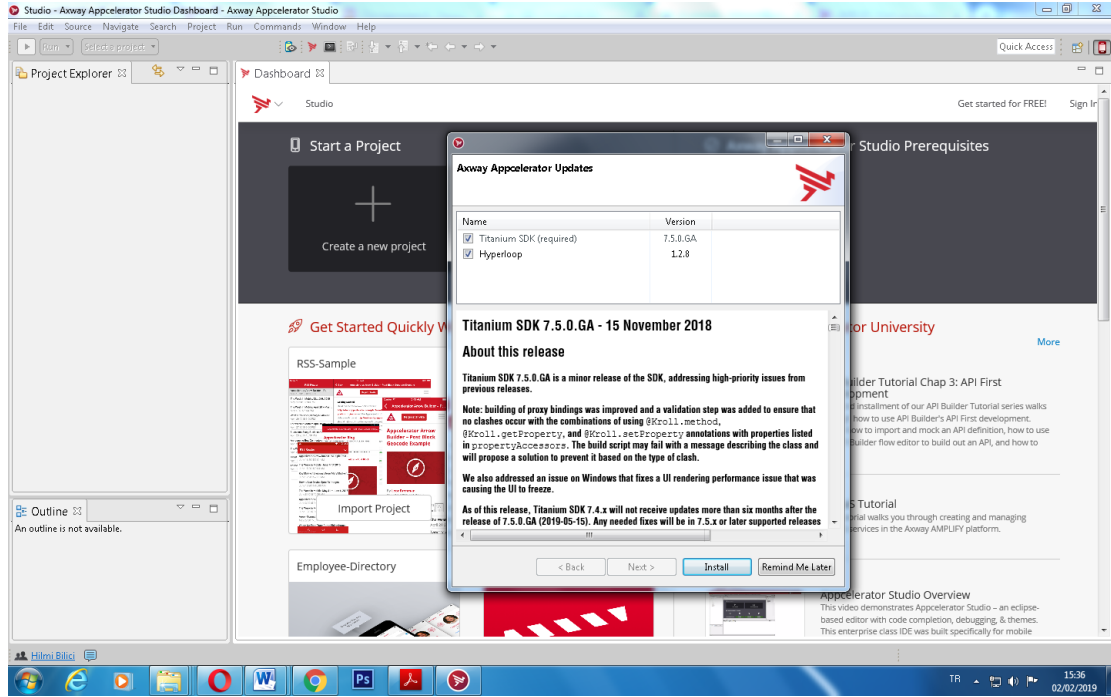
Workspace (Çalışma Alanı) Klasörü Seçme

Titanium Studio uygulaması ilk çalıştırdığında bir çalışma alanı seçilmesi gerekecektir. Bunu varsayılan olarak bırakabileceğiniz gibi kendinizde bir klasör yolu belirleyerek yapabilirsiniz.

Appcelerator Hesabına Giriş Yapma

Studio'yu indirirken Appcelerator sitesinde bir hesap oluşturmuştuk bu oluşturulan hesabı buraya yazıyoruz. Bunları yaptıktan sonra Titanium Studio programı çalışacaktır.

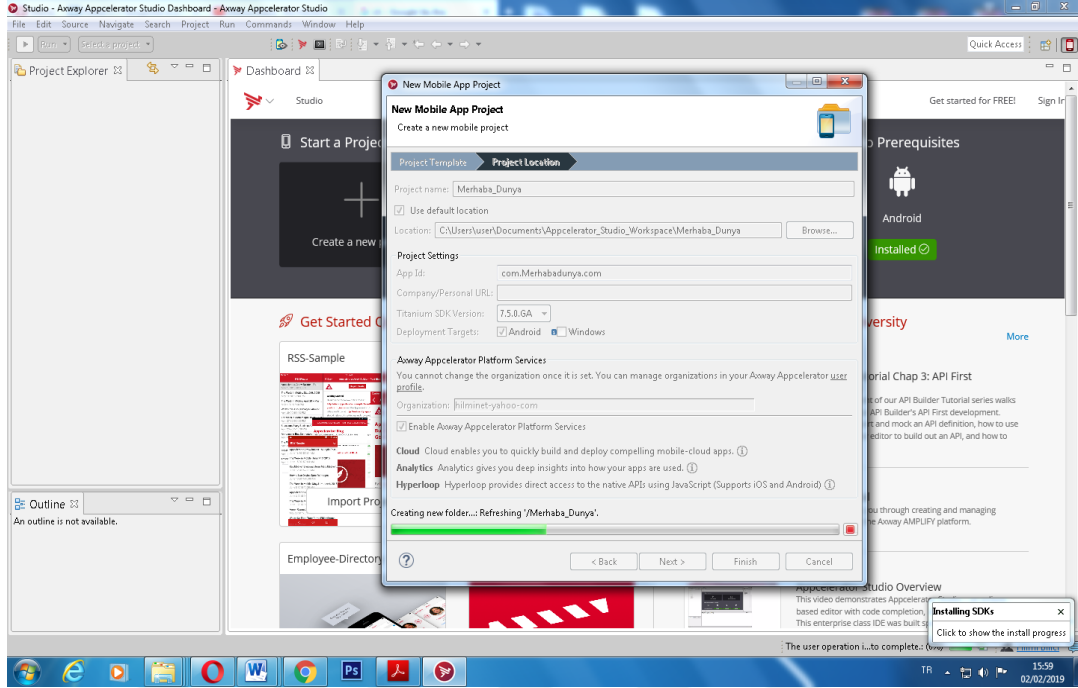
Yerel geliştirme yapabilmemiz için gerekli olan platformların SDK'larını yüklememiz gerekmektedir. Örneğin, Android uygulaması geliştirmek için; Android SDK ve Toolchain, iOS uygulaması geliştirmek için; Xcode'a ihtiyaç duyulmaktadır.



Şekil 6.15 Axway studio proje başlatma ekranı

NOT: Eğer açılmadıysa Studio DashBoard'u açmak için menüde bulunan Appcelerator Logosuna tıklamak gereklidir.

Kurulum yapıldıktan sonra ilk uygulamayı geliştirme ekranı:



Şekil 6.16 İlk uygulamayı hazırlama

İlk kodlama:

```
var win1 = Ti.UI.createWindow({
    backgroundColor:"white",
    title:"Merhaba Dünya"
});
var labell = Ti.UI.createLabel({
    text:"Merhaba Dünya",
    color:"black"
});
win1.add(labell);
win1.open();
```

EK-4 IONIC kurulumu

- 1- Komut Tabanlı Ara yüz (CLI) Kurulumu için Komut satırına aşağıdaki kodu yazılır (Şekil 6.17’te görüldüğü gibi): `npm install -g ionic`

```
C:\Users\user>npm install -g ionic
C:\Users\user\AppData\Roaming\npm\ionic -> C:\Users\user\AppData\Roaming\npm\node_modules\ionic\bin\ionic
+ ionic@4.10.1
added 262 packages from 175 contributors in 79.327s
```

Şekil 6.17 Ionic komut tabanlı kurulumu başlatma

2- Hazır uygulama şablonlarından birini veya yeni bir başlangıç yapmak için boş olanı kullanarak bir Ionic Uygulaması oluşturulur: `ionic start myApp tabs`

3- Uygulama klasörüne girme ve tarayıcıdan çalıştırma:

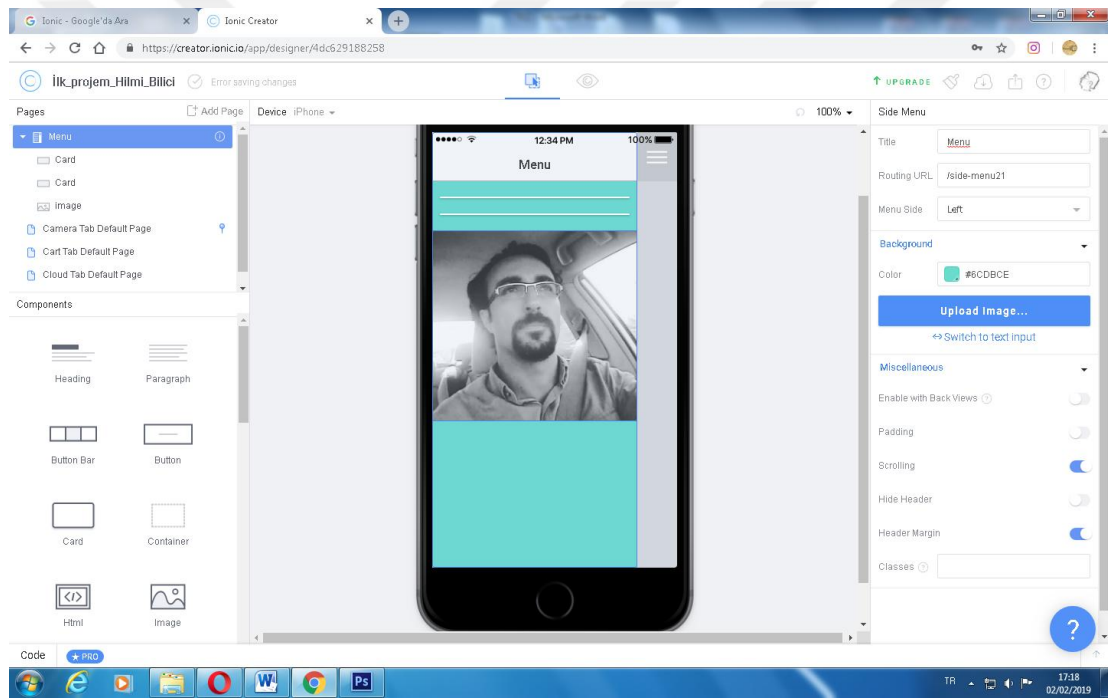
```
cd myApp
```

```
ionic serve
```

Gerçek zamanlı güncellemeler ve daha fazlası için kodu Ionic Appflow'a getirilir:

```
git push ionic master
```

Ionic Creator Proje oluşturma ve çalışma Ekranı Şekil 6.15'te görülmektedir.

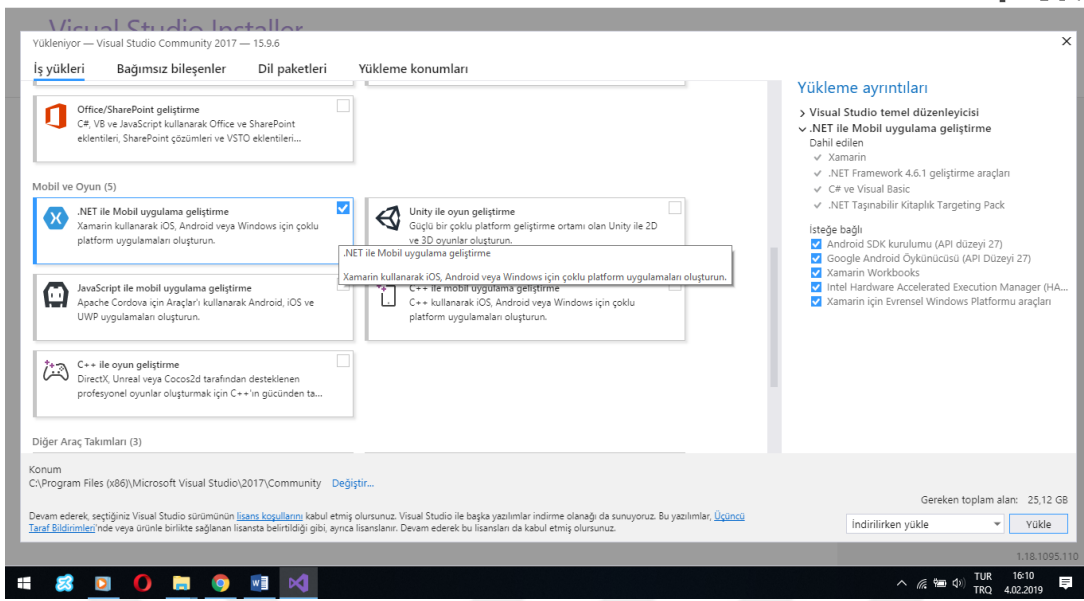


Şekil 6.18 Ionic çalışma ekranı

EK-5 Xamarin'i Yükleme

1- Visual Studio 2017 Community, Visual Studio Professional veya Visual Studio Enterprise'nden indirilir.

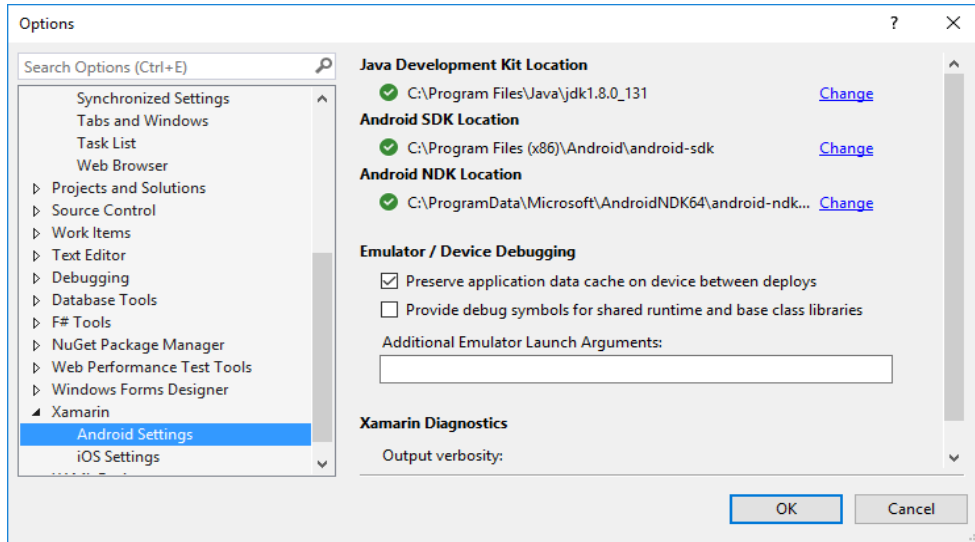
Yükleme ekranında, NET ile Mobil Geliştirme seçimi işaretlenir ve Yükleme ayrıntıları seçilir:



Şekil 6.19 .NET ile mobil uygulama geliştirme

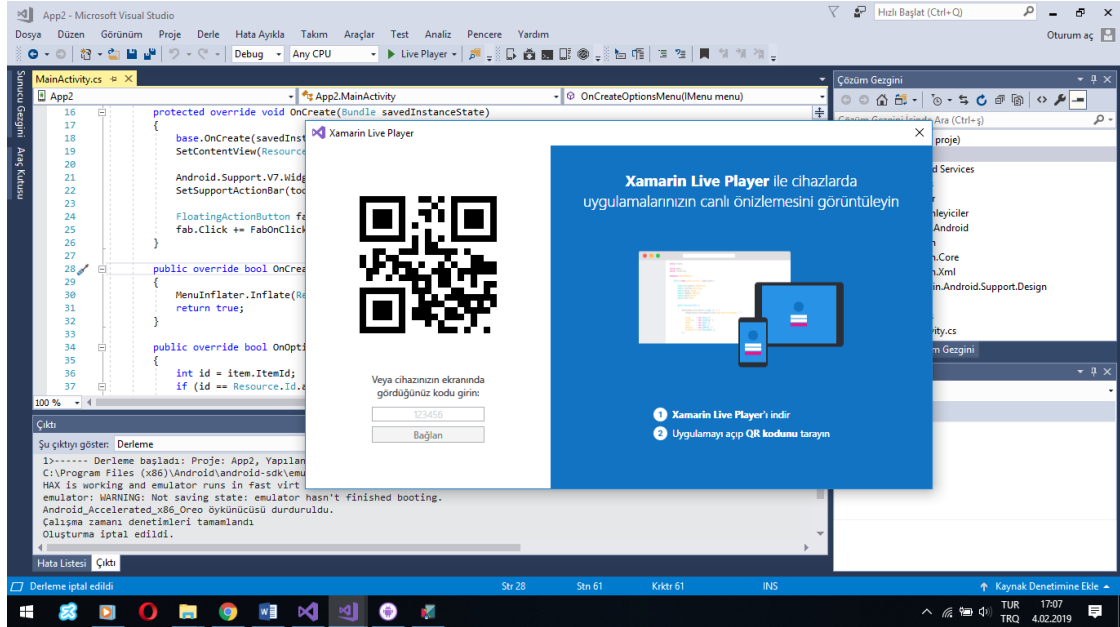
Visual Studio 2017'ye Xamarin ekleme ve Android Yapılandırma

Xamarin ile Android uygulamaları oluşturmak için Java Development Kit (JDK) ve Android SDK'sından yararlanır. Visual Studio yükleyicisi, araçları, varsayılan konumlarda yerleştirir ve geliştirme ortamına uygun yolu yapılandırılır (Şekil 6.20).

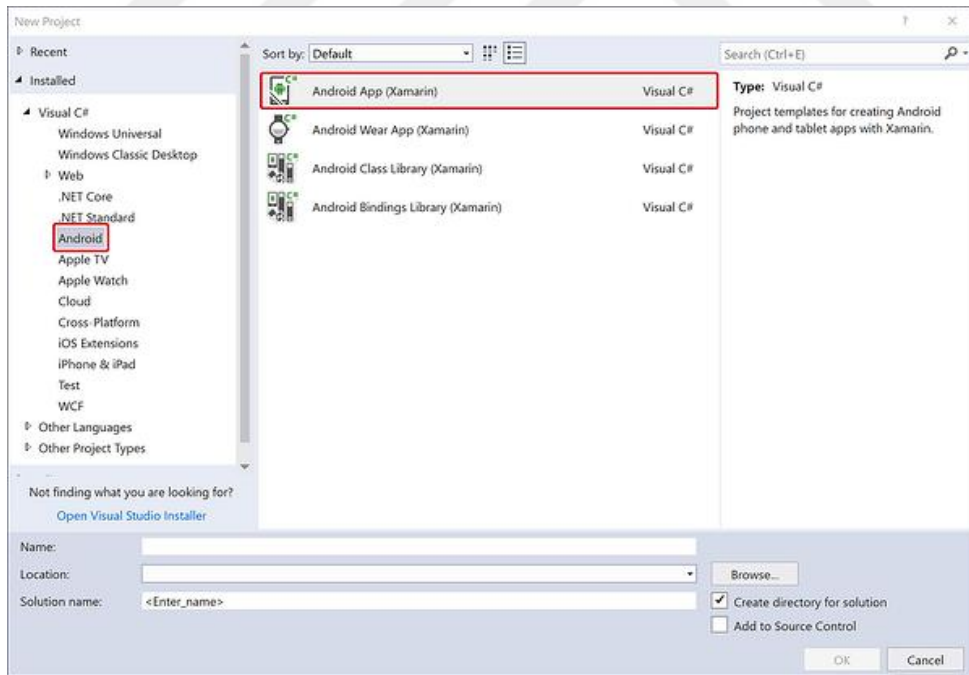


Şekil 6.20 Android yapılandırma ekranı

Android sürümleri ile ilgili detaylı bilgi adresi: <https://docs.microsoft.com/tr->



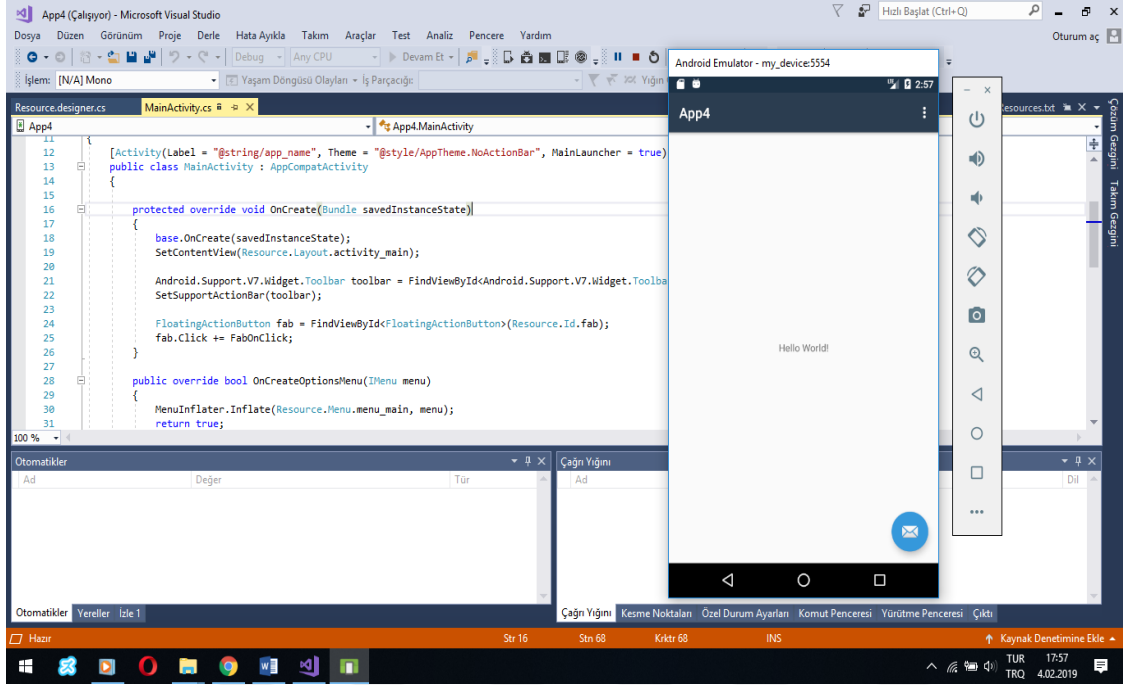
Şekil 6.21 Uygulama telefona yüklenen live player ile canlı görünüm sağlanabilir.



Şekil 6.22 Android Xamarin uygulama başlatma ekranı

Xamarin Android en düşük API düzeyi seçmeden önce bu API düzeyine karşılık gelen Android SDK platform sürümünü yüklemek gereklidir. Android, iOS, Mac, tvOS,

watchOS, Windows, Windows Phone, Xamarin Live Player ve Xamarin.Forms ile ilgili örnekleri <https://developer.xamarin.com/samples-all/> adresinde bulunabilir.



Şekil 6.23 Emulator Ekranı

EK-6 İlk Sencha Ext JS Uygulamasını Oluşturma:

Windows platformunda Sencha Cmd kullanarak ilk Ext JS 6 uygulamasını oluşturacağız. Sencha Cmd ve GPL Ext JS SDK'yı indirip kurmak gereklidir.

1. Adım - Uygulama oluşturma

Sencha CMD komut satırına aşağıdaki kod yazılır:

```
sencha -sdk /path/to/ExtSDK generate app -classic TutorialApp ./TutorialApp
```

Elde edilen TutorialApp, giriş / çıkış uygulamasına temel olarak kullanılacak tamamen işlevsel bir Cmd uygulamasıdır. Web sunucusundaki (Sencha localhost üzerinde çalışır. IIS yapılandırması gereklidir) uygulamanın konumuna giderek bu uygulamayı görüntüleyebilirsiniz. Örneğin: <http://localhost/TutorialApp>

2.Adım - Login View Bileşenlerini Oluşturun

Uygulamanın "app / view /" klasörüne gidin. Varsayılan olarak oluşturulan "Main" klasörü görmelisiniz. Bu klasör Main.js,MainController.js veMainModel.js dosyalarını içerir. "App / view / login /" klasörü oluşturulduktan sonra, aşağıdaki dosyaları belirtilen klasöre eklemek gereklidir:

- Login.js

- LoginController.js

3.Adım - mainView'ı devre dışı bırakın

Görünüm eklentisinden "TutorialApp.view.main.Main" devre dışı bırakılır.

4.Adım- Giriş Penceresini Yaratın

"{appRoot}/app/view/login/Login.js" oluşturup dosyaya aşağıdaki kodu eklenir:

```
Ext.define('TutorialApp.view.login.Login', {  
    extend: 'Ext.window.Window',  
    xtype: 'login',  
    requires: [  
        'TutorialApp.view.login.LoginController',  
        'Ext.form.Panel'  
    ],  
    controller: 'login',  
    bodyPadding: 10,  
    title: 'Login Window',  
    closable: false,  
    autoShow: true,  
    items: {  
        xtype: 'form',  
        reference: 'form',  
        items: [{  
            xtype: 'textfield',  
            name: 'username',  
            fieldLabel: 'Username',  
            allowBlank: false  
        }, {  
            xtype: 'textfield',  
            name: 'password',  
            inputType: 'password',  
            fieldLabel: 'Password',  
            allowBlank: false  
        }  
    ]  
});
```



```

    }, {
      xtype: 'displayfield',
      hideEmptyLabel: false,
      value: 'Enter any non-blank password'
    }],
    buttons: [{
      text: 'Login',
      formBind: true,
      listeners: {
        click: 'onLoginClick'
      }
    }]
  }
});

```

5.Adım - Giriş Mantığı Ekleme

{appRoot}/app/view/login/LoginController.js dosyası oluşturulur:

```

Ext.define('TutorialApp.view.login.LoginController', {
  extend: 'Ext.app.ViewController',
  alias: 'controller.login',
  onLoginClick: function() {
    // kullanıcının kimlik bilgilerini doğrulamak için
    // LocalStorage değerini true olarak ayarlayın.
    localStorage.setItem("TutorialLoggedIn", true);
    // Giriş Penceresini Kapat
    this.getView().destroy();
    // Ana görünümü, görünüm alanına ekleyin
    Ext.create({
      xtype: 'app-main'
    });
  }
});

```

Sonuçta ortaya çıkan {appRoot}/app/view/main/Main.js dosyası çalıştırılır. (Sencha resmi sitesinden faydalanılmıştır.(İnt. Kyn.13)

EK-7 Basit bir IBM Worklight Uygulaması Oluşturma

1. IBM Worklight Studio'yu başlatın.

Programı indirme adresi: <https://www.ibm.com/account/reg/us-en/signup?formid=mrs->

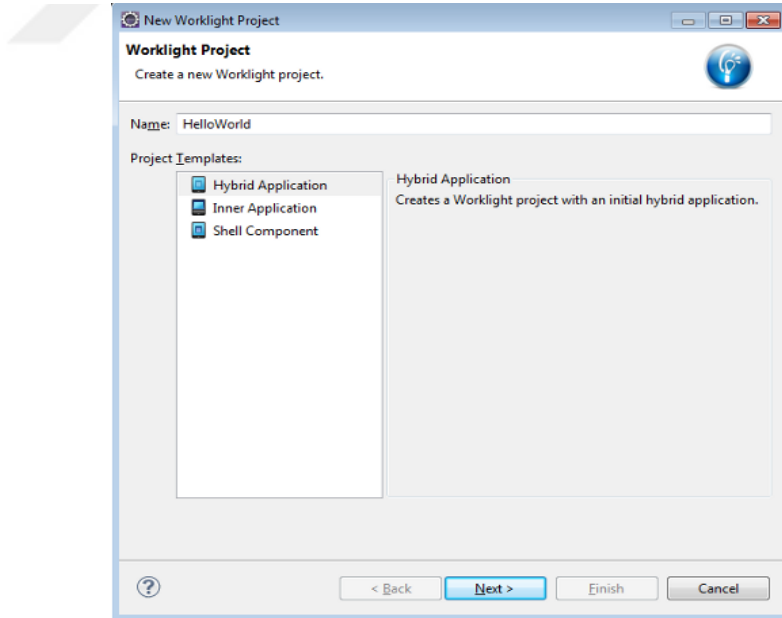
form-4136&S_CMP=web_dw_rt_swd

MobileFirst Sunucusunu yerel iş istasyonunuzda kullanmak için, Java Developer Kit (JDK) 7 veya 8 programının yüklenmesi gerekir.

Güncel Java sürümünü Bulma:

Java 7 güncellemesi 40. sürümden itibaren, Windows Başlat menüsünü kullanarak bulabilirsiniz.

2. **Windows Başlat** menüsünü açılır
3. **Programlar**'a tıklanır
4. **Java** program listesi bulunur
5. Java sürümünü görmek için **Java Hakkında**'ya tıklanır :
https://www.java.com/tr/download/help/version_manual.xml
6. **Yeni Worklight Project**'i seçilir:
7. Aşağıdaki ekran görüntüsünde görüntülenen iletişim kutusunda, proje şablonları tanımlanan uygulamanın türü olarak **Hibrid Uygulama** seçilip, mobil projenin adı *MerhabaDünya* olarak yazılır.

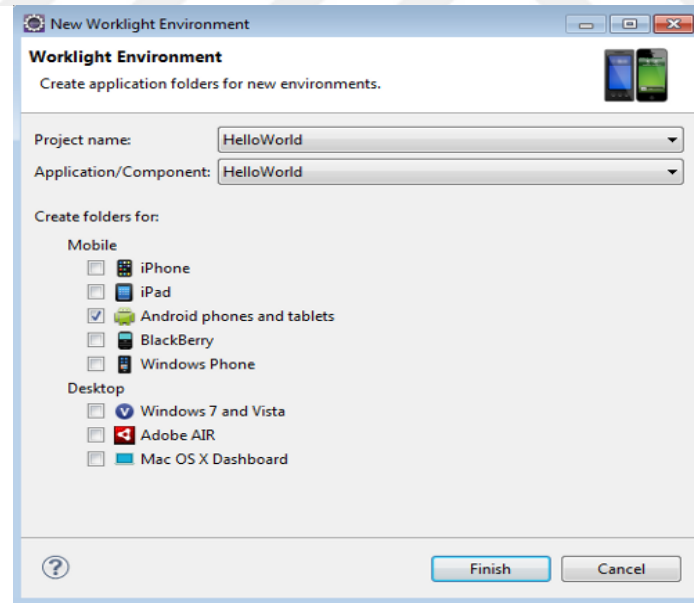


Şekil 6.24 Worklight Hybrid proje oluşturma ekranı

- 8- **Hibrit Uygulama** için başka bir iletişim kutusu görülecektir. Şimdilik onay kutularını işaretlessiz bırakın; Bunlar, desteklenen JavaScript kitaplıklarını uygulamaya genişletmek için kullanılır.
- 9- **Bitir**'i tıkladıktan sonra, aşağıdaki ekran görüntüsünde gösterildiği gibi, projenin **Proje Gezini**'nde tasarım açısından yaratıldığını görülecektir:

Bir proje eklemek

IBM Worklight Studio ve eklentinin hayatınızı nasıl daha da kolaylaştıracağını görme zamanı geldi. Verimli ara yüzü içinde sadece birkaç tıklama ile çapraz geliştirme ortamı elde etmek için gerekli araçları sağlar. Android, iPhone veya başka bir platform için ortam eklemek için, adaptörlerin yanındaki *Uygulamalar* klasörüne sağ tıklayın ve **Yeni Çalışma Ortamı**. Bir uygulama oluşturmanız gereken, şu anda desteklenen ortamlar için onay kutularını içeren bir iletişim kutusu görüntülendiğini görülür. Aşağıdaki ekran görüntüsü bu özelliği göstermektedir; bu uygulama için bir Android ortamı eklenir:



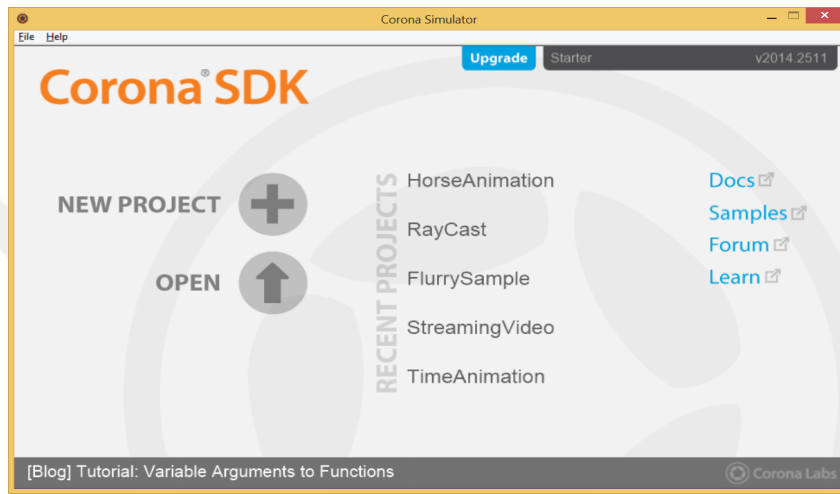
Şekil 6.25 Worklight Environment seçim penceresi

Hataların çözümleri ve örnekler için bu adres ziyaret edilebilir:

<https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/issues>

EK-8 Corona SDK Kurulum ve Araçlar

Programı Windows veya Mac için şu adresten: <https://developer.coronalabs.com/downloads/corona-sdk> indirilebilir. Windows programı Java 6 Development Kit'in yüklü bir 32 bit sürümünü gerektirir. İhtiyaç duyulan 32 bit sürüm ve altıncı sürüm! Aksi takdirde, Corona SDK kurulmayacak ve başlamayacaktır.



Şekil 6.26 Corona program penceresi

Corona'yı ilk kez başlatıldığında, incelemeye, örneklerin grafik motorunun temel özelliklerini gösterdiği “Örnekler” bağlantısından başlamak iyi olur.

ZeroBrane Studio (<http://studio.zerobrane.com>)

Windows ve MacOS için kompakt, ücretsiz geliştirme ortamıdır. Corona SDK'yı hedef alan kod vurgulama, çevrilen komutların otomatik olarak tamamlanması ve yerleşik hata ayıklayıcı (debugger) hizmeti bulunmaktadır.

Lua planör (<http://www.mydevelopersgames.com/Glider/>)

Birçok özelliğe sahip güçlü bir geliştirme ortamıdır. Doğal olarak, bu platform için ödeme zorunluluğu vardır. Deneme sürümünü Windows ve MacOS'tan indirebilir. Bahsedilen uzmanlık programlarına ek olarak, bilinen herhangi bir geliştirme ortamı da uygundur, sadece Lua eklentisi ile desteklenir.

Eclipse Eklentisi (<http://www.eclipse.org/ldt/>) Eclipse IDE için kullanılan bir eklentidir. Lua sözdizimi, vurgulama, kod hata ayıklama, otomatik tamamlama vb. özellikleri vardır.

öneririz. Bu yaklaşım, uzak cihazlardan test etme yeteneği de dahil olmak üzere sayısız avantaj sağlar. Yerel web sunucu hizmeti için: `npm install -g serve`

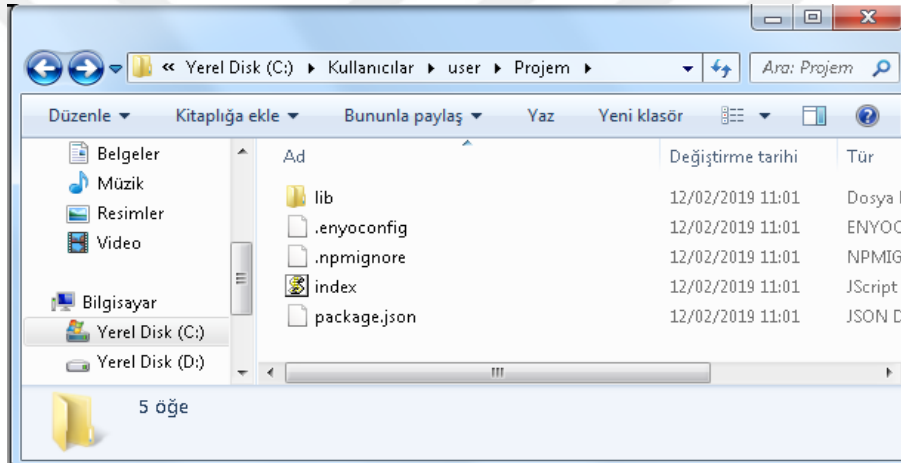
Bir Uygulama Oluşturma

Örnek bir, uygulama oluşturulması: `enyo init Projem`

```
C:\Users\user> enyo init Projem
cloning enyo and checking out 2.7.0
```

Şekil 6.28 Uygulama oluşturma

Dosyaların oluşturulduğu dizin:



Şekil 6.29 Dizin içindeki dosyalama

index.js dosyasının içeriğini aşağıdaki kodla değiştirilir:

```
var Control = require('enyo/Control');
var ready = require('enyo/ready');
var HelloWorld = Control.kind({
  content: 'Merhaba, Dünya!'
});
ready(function () {
  new HelloWorld().renderInto(document.body);
});
```

aşağıdaki komut build/app.css ve içerisinde çıktı yaratacaktır `build/app.js`.

```
enyo pack --production -d ./build --no-inline-css --css-
outfile app.css --no-inline-js --js-outfile app.js
```

Çıktı dizinlerinin oluşması için kullanılan komut: `enyo pack`

EK-10 Lungo framework kurulumu

Başlamak için komut satırına aşağıdaki kod yazılır: *npm i lungo*

Lungo yapısında bunları içermelidir:

```
< link    rel = " stylesheet "    href = " components /  
lungo.brownie / lungo.css " >  
< link    rel = " stylesheet "    href = " components / lungo.icon /  
lungo.icon.css " >  
< link    rel = " stylesheet "    href = " components /  
lungo.brownie / lungo.theme.css " >  
< script  src = " components / quojs / quo.js " > </ script >  
< script  src = " components / lungo / lungo.js " > </ script >
```

Her bölüm ve makale benzersiz bir kimlik içermelidir:

```
<section id="main">  
  <article id="main-article">  
    içeriğiniz  
  </article>  
</section>
```

Lungo'yu başlatan JavaScript işlevi:

```
Lungo.init({});  
//uygulama kaynaklarını yükle  
Lungo.init({  
  name: 'example',  
  resources: ['section_to_load.html']  
});
```

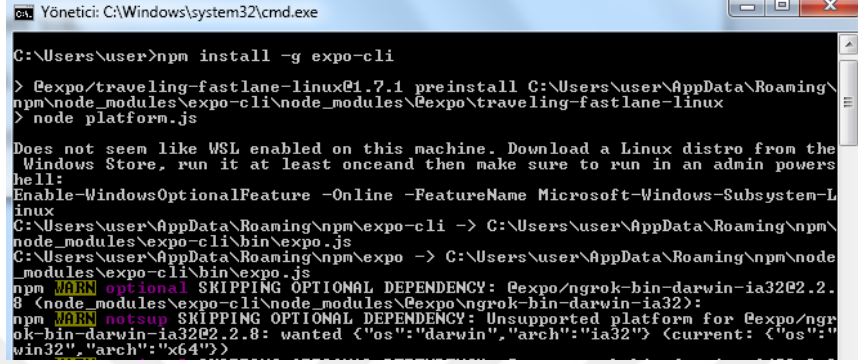
Kurulumu ve kodlama için web sitesi: <https://github.com/tapquo/Lungo.js/>

Not: Resmi web sitesinin kapanmış olması bu çerçeve ile ilgili olumsuz bir hava estirmiştir.

EK-11 React native Kurulumu:

- 1- Npm kullanarak komut satırına aşağıdaki kod yazılır:

```
npm install -g expo-cli
```



```
Yönetici: C:\Windows\system32\cmd.exe
C:\Users\user>npm install -g expo-cli
> @expo/traveling-fastlane-linux@1.7.1 preinstall C:\Users\user\AppData\Roaming\npm\node_modules\expo-cli\node_modules\expo\traveling-fastlane-linux
> node platform.js
Does not seem like WSL enabled on this machine. Download a Linux distro from the Windows Store, run it at least once and then make sure to run in an admin powershell:
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
C:\Users\user\AppData\Roaming\npm\expo-cli -> C:\Users\user\AppData\Roaming\npm\node_modules\expo-cli\bin\expo.js
C:\Users\user\AppData\Roaming\npm\expo -> C:\Users\user\AppData\Roaming\npm\node_modules\expo-cli\bin\expo.js
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @expo/ngrok-bin-darwin-ia32@2.2.8 (node_modules\expo-cli\node_modules\expo\ngrok-bin-darwin-ia32):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for @expo/ngrok-bin-darwin-ia32@2.2.8: wanted {"os":"darwin","arch":"ia32"} (current: {"os":"win32","arch":"x64"})
```

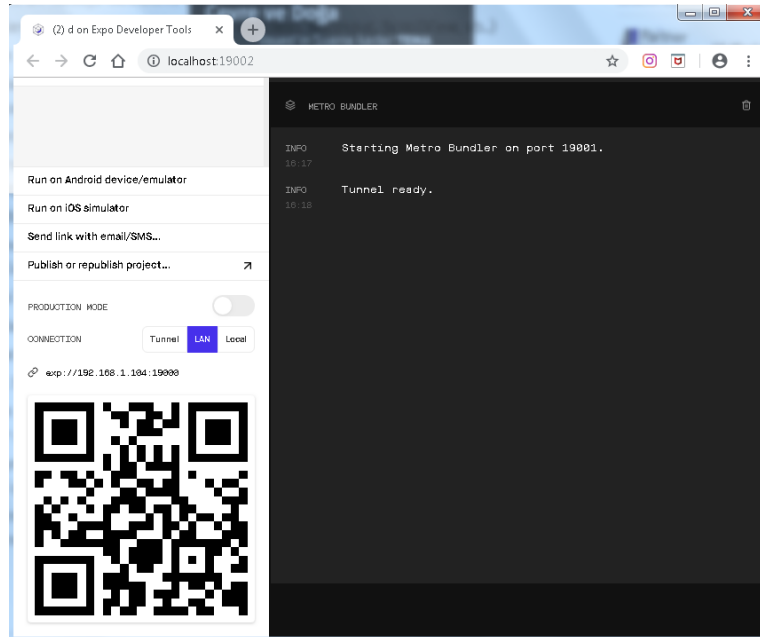
Şekil 6.30 Expo kurulumu

- 2- Ardından "AwesomeProject" adlı yeni bir React Native projesi oluşturmak için sıra ile aşağıdaki komutları çalıştırılır:

```
expo init AwesomeProject
```

```
cd AwesomeProject
```

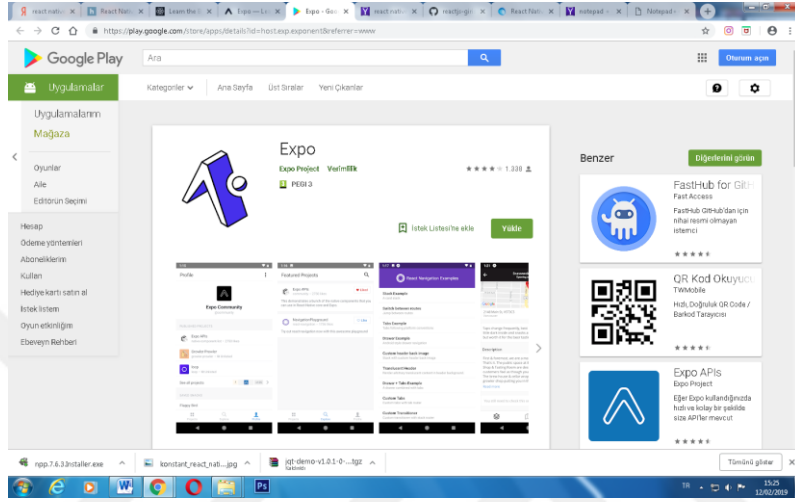
```
npm start #you can also use: expo start
```



Şekil 6.31 Karekod oluşturma

Uygulamayı çalıştırma

Expo istemci uygulamasını iOS veya Android telefonunuza yükleyin ve bilgisayarınızla aynı kablosuz ağı bağlanın. Android'de, projenizi açmak için terminalinizdeki QR kodunu taramak için Expo uygulamasını kullanın. IOS'ta bağlantı almak için ekrandaki talimatları izleyin.



Şekil 6.32 React Native çalıştırmak için kurulacak telefon uygulaması

Örnek Kodlama:

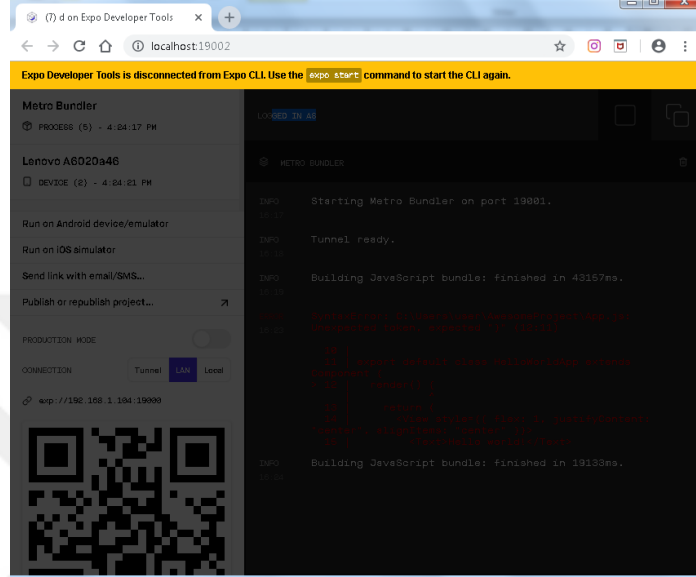
```
import React from 'react';

import { StyleSheet, Text, View } from 'react-native';
export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Text>Merhaba Dünya</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  }
});
```

Uygulamayı deęiřtirme

Uygulamayı bařarıyla alıřtırdıđınıza gre, artık deęiřtirilebilir. Tercih edilen metin editrnde *App.js* dosyasını aılır ve dzenlenir. Deęiřiklikleri kaydettikten sonra uygulama otomatik olarak yeniden yklenmelidir.

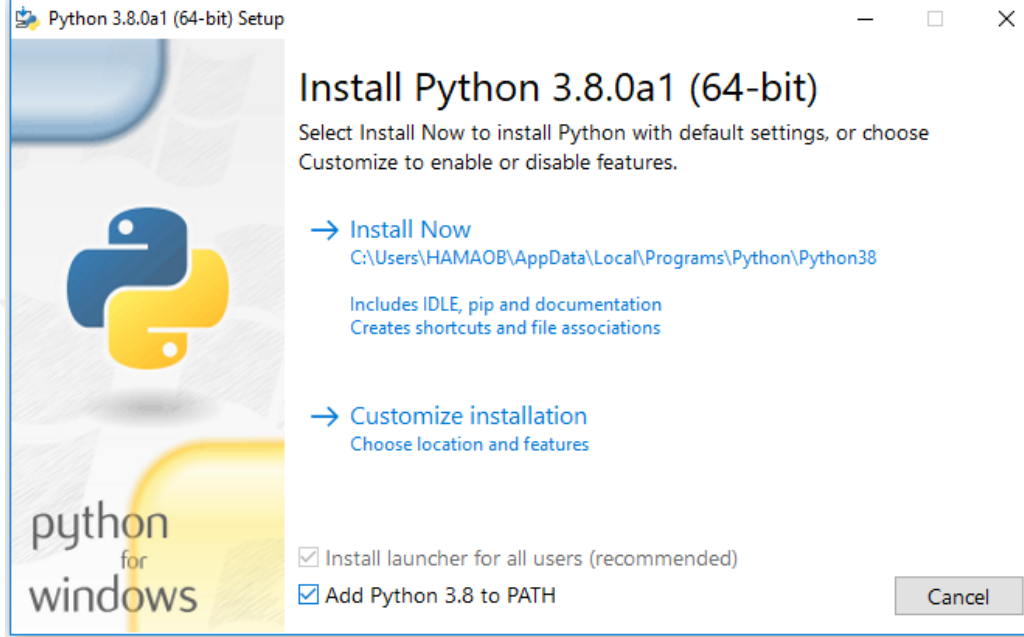


řekil 6.33 Kod deęiřiklięi ve gncelleme ekranı

Kodda deęiřiklik yapılıp kaydedildięinde anında cep telefondaki uygulama da gncellenen srm gsteriliyor (řekil 6.50). React Native đrenmek ve bařlamak istiyorsanız facebook sayfasını tıklayarak ulařılabilir: <https://facebook.github.io/react-native/>

EK-12 Kivy Kurulumu:

Önce Python uygulaması indirilerek yüklenmelidir. “Add Python Path” kutucuğu işaretlenmelidir.



Şekil 6.34 Python kurulum ekranı

Python’ı yükledikten sonra, komut satırını açıp, komut satırına “python --version” yazarak onun kullanılabilir olduğundan emin olmak gereklidir.

NOT: Komut satırından Python’ı kullanabilmek için yükleme esnasında Add Python 3.5 to PATH seçeneği seçilmelidir. Aşağıda belirtilen komutu windows’un komut satırına yazılır:

```
python -m pip install
```

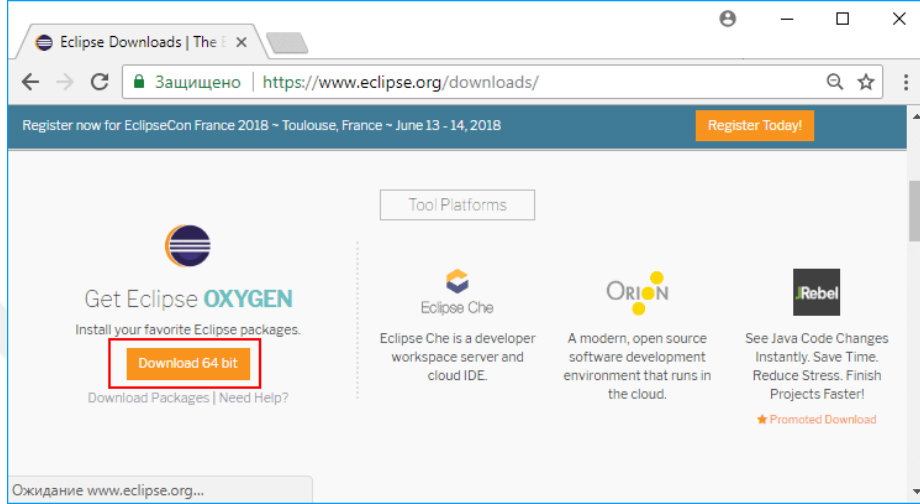
Pip’in ve wheel’in son sürümleri olması gerekli, eğer son sürümler yoksa zaten uyarı verecek yükleme için yönlendirecektir:

Kivy kurulumu için gerekli komut: `python -m pip install kivy`

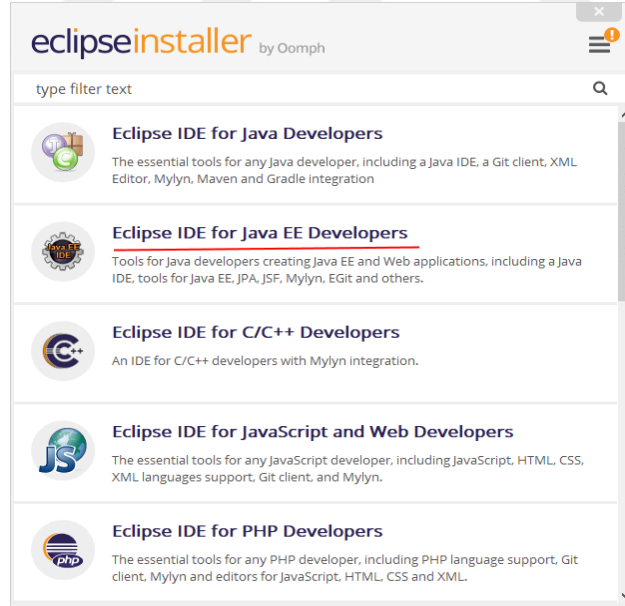
İnternette Python programı haricinde herhangi bir şey indirmeye gerek yoktur. Kurulum işlemi için gerekli olan yazılımlar pip ile komut tabanlı ara yüzde indirilecektir.

EK 13: Eclipse Kurulum ve kullanım

Eclipse kurmak için İlk önce, <https://www.eclipse.org/downloads/> adresinde bulunabilecek en yeni sürümünü indirmek gereklidir.



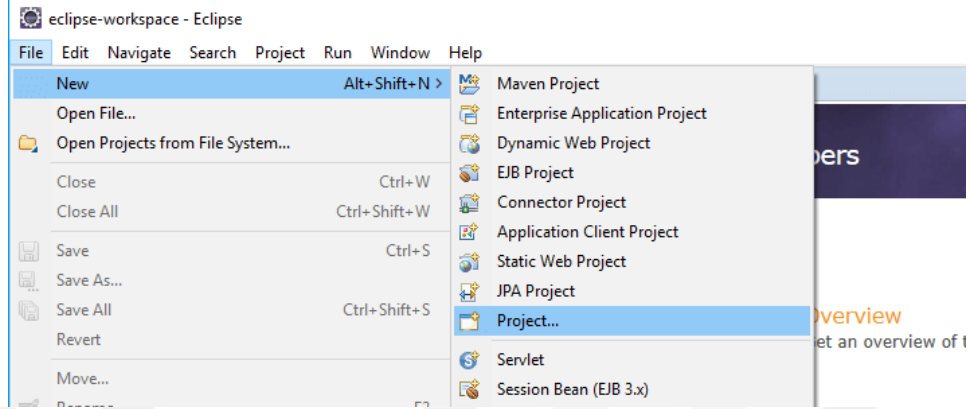
Şekil 6.35 Eclipse kurulumu



Şekil 6.36 Eclipse kurulumunda IDE seçimi

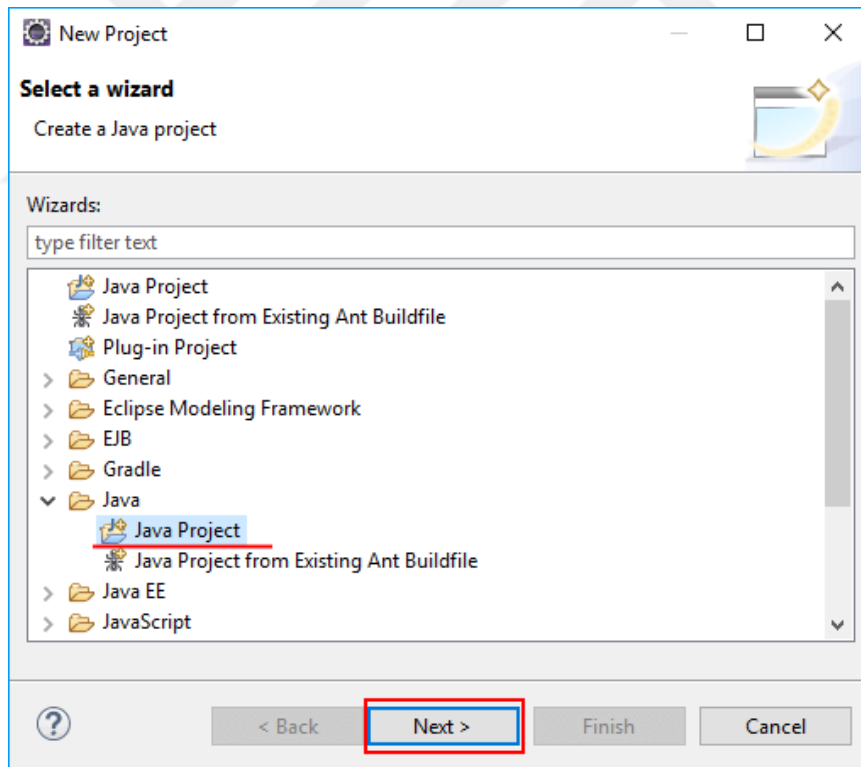
Kurulum yapar iken; Java Geliştiricileri için Eclipse IDE, Java EE Geliştiricileri için Eclipse IDE seçenekleri vardır. İlk seçenek Java ile çalışmak için temel özellikler sağlar ve ikincisi de web uygulamaları ve servisleri oluşturmanıza olanak sağlar. Aslında,

ikinci seçenek birinci seçeneği de kapsar. Bu nedenle, Java EE Geliştiricileri için Eclipse IDE'yi seçilmelidir.



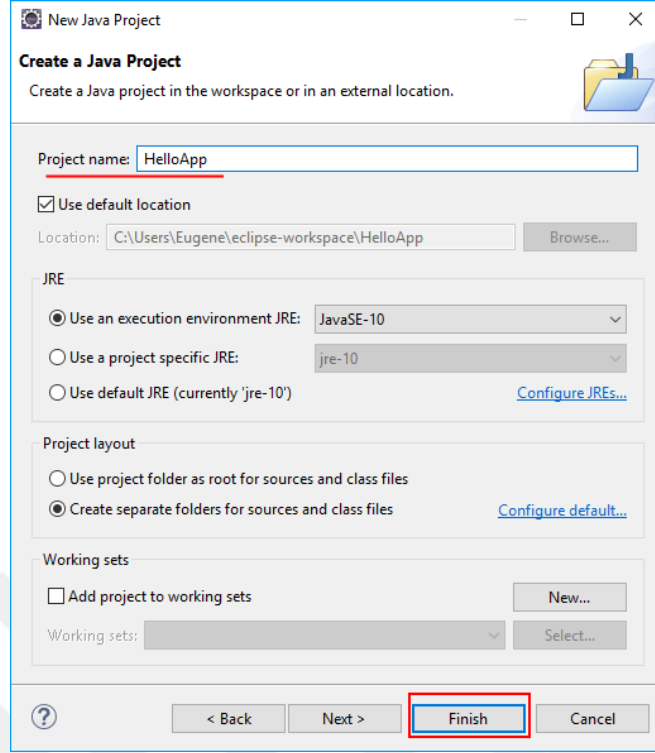
Şekil 6.37 Yeni proje oluşturma

Yeni bir Proje seçimi için New>Project sekmesine tıklanır.



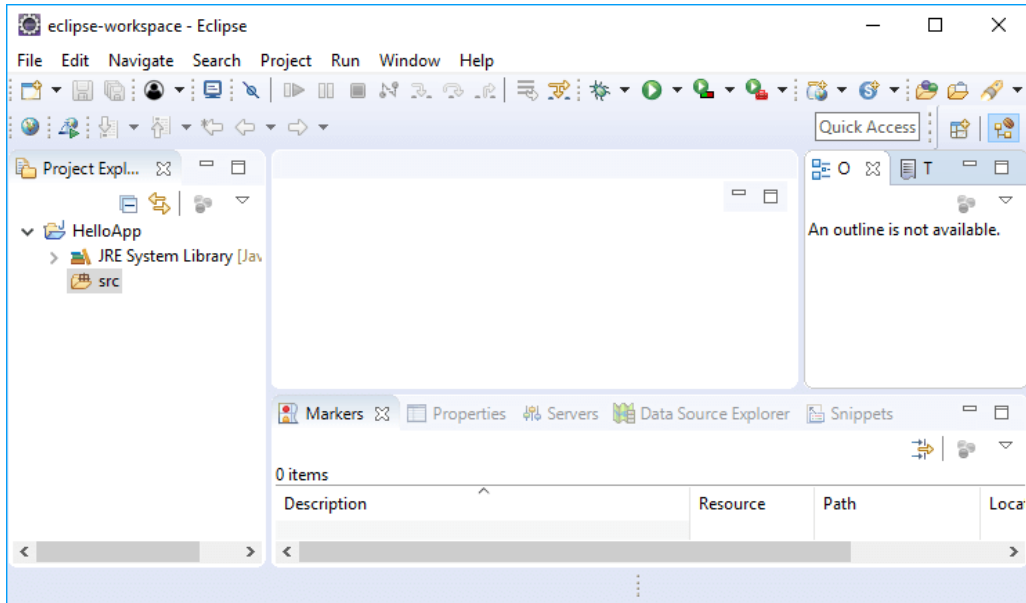
Şekil 6.38 Proje oluşturma sihirbazı

Ardından proje ayarlarını yapmanız istenecektir. Özellikle, Proje Adı alanında projenin adını belirtmeniz gerekir.



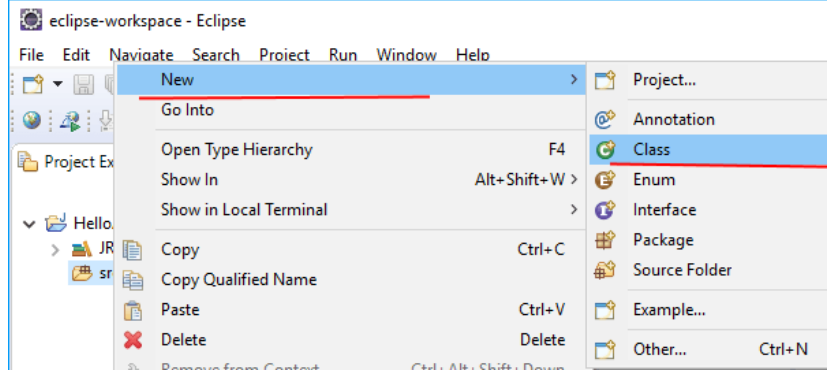
Şekil 6.39 İlk uygulama oluşturma

Proje adı yazıldıktan sonra Diğer tüm ayarlar varsayılan olarak bırakılabilir ve sonunda bir proje oluşturmak için Son düğmesine tıklanır.



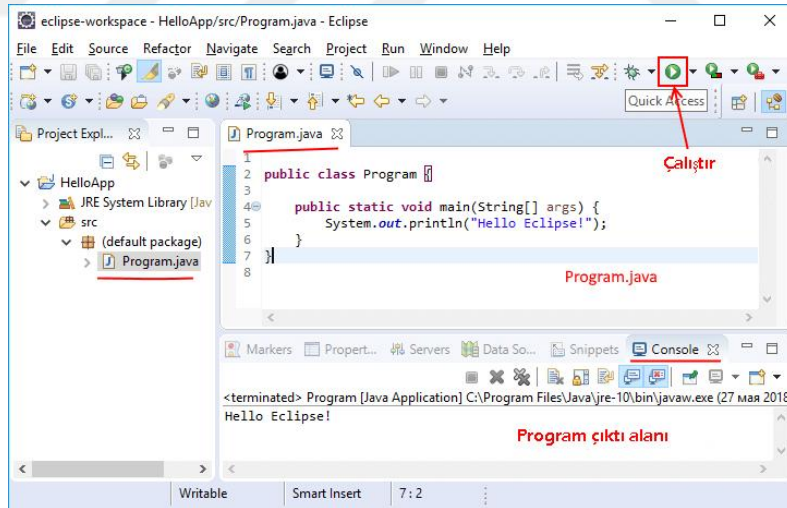
Şekil 6.40 Eclipse çalışma ekranı

Projedeki tüm Java kaynak dosyaları **src** klasörüne yerleştirilir. Üzerine farenin sağ tuşu ile tıkla ve bağlam menüsünden New -> Class'ı seçilir:



Şekil 6.41 Yeni sınıf oluşturma

Sınıf kurulum penceresinde Ad alanında, sınıfın adını belirtmek gerekir. Diyelim ki *Program* olarak adlandırılacak. Diğer tüm ayarlar varsayılan olarak bırakılabilir. Bundan sonra, proje Program.java proje dosyasına eklenecek ve Eclipse'nin orta kısmında kodunu görülebilecektir.



Şekil 6.42 Çalışma Ekranı

Çalıştır komu verildikten sonra konsol alanında program çıktısı görülebilir.

EK-14 Visual Studio Code Yükleme:

Mac os x

1. Mac OS X için Visual Studio kodunu indirin
2. VSCode-osx.zip arşivine çift tıklayın.
3. Visual Studio Code.app'ı Uygulamalar klasörüne sürükleyin
4. Seçenekler içindeki “Dock'ta Tut” u seçerek Dock'a Visual Studio Kodu ekleyin

İpucu: Visual Studio Kodunun terminalden başlamasını istiyorsanız, `~ / .bash_profile` dosyasına ekleyin.

```
code () { VSCODE_CWD="$PWD" open -n -b "com.microsoft.VSCode" --args $* ;}
```

Linux

1. Linux için Visual Studio Kodunu İndirin
2. Yeni bir klasör oluşturun ve VSCode-linux-x64.zip arşivinin içeriğini açın
3. Kod üzerine çift tıklayın.

İpucu: Visual Studio Code kurulumunun terminalden başlamasını istiyorsanız, şu bağlantı kullanılır: `sudo ln -s /path/to/vscode/Code /usr/local/bin/code`

Windows

1. Windows için Visual Studio Kodunu İndirin:
<https://code.visualstudio.com/docs?dv=win>
2. Kurulumu başlatmak için indirilen VSCodeSetup.exe dosyasına çift tıklayın.

Visual studio Code’u denemek için en kolay yol bir Node web projesi oluşturmaktır. Bunu yapmak için, Node bilgisayarınıza kurun (yukarıdaki linke tıklayın) ve komut satırında aşağıdaki komutları çalıştırın:

```
npm install -g express
```

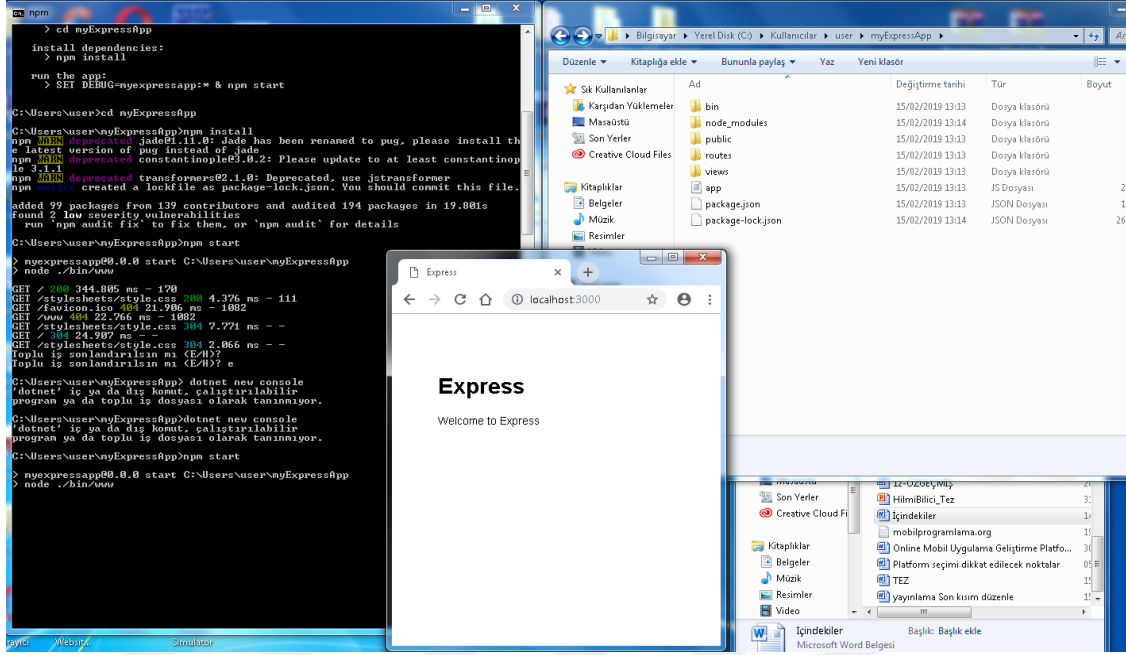
```
npm install -g express-generator
```

```
express myExpressApp
```

```
cd myExpressApp
```

```
npm install
```

npm start komutu ile çalıştırın ve tarayıcıyı <http://localhost:3000> adresinden başlatılır.



Şekil 6.43 İlk projenin çalıştırılması

EK-15 Yayınlama

Uygulamaların son kullanıcılara ulaşabilmesi için paketleme ve yayınlama işlemlerinin yapılması gereklidir.

EK 15.1 Android Uygulamasının Yayınlaması (Disturbition)

Bu bölümde uygulamanın son adımı olan Google Play’de yayınlaması anlatılacaktır.

Google Play’e Kayıt

Google Play’de uygulamanızı yayınlamak için öncelikle bir Google hesabına gerek vardır. Herhangi bir Gmail hesabınız varsa onu kullanabilir ya da yeni bir hesap oluşturarak başlanabilir. Daha sonra play.google.com adresinden sadece bir defaya özgü olan 25 dolarlık program ücretini ödedikten sonra geliştirici programına katılım sağlanır.

Sertifika ve Keypair oluşturma

Uygulamanın Google Play'e yüklenmesinden önce dijital olarak imzalanması gereklidir. Bunun için ise bir sertifika ve keypair'e (anahtar çifti) ihtiyaç vardır. Oluşturmak için Java SDK içinde bulunan keytool'u kullanmak gereklidir.

Genel formu ise şöyledir:

```
keytool -genkeypair -v -keystore path/file_name -alias  
alias_name -keyalg  
RSA -sigalg SHA1withRSA -validity 10000
```

burda path/filename keystore'un nerede saklanacağını gösterir.

Bir uygulama için girilenler,

```
keytool -genkeypair -v -keystore android.keystore -alias  
helloworldhb -keyalg  
RSA -sigalg SHA1withRSA -validity 10000
```

Keystore'u (özel anahtarların deposu) doğrulama

Uygulamayı paketlemeden önce keystore'un doğru olduğundan emin olmanız gerekecektir. Çünkü hatalı bir keystore'da uygulamanız marketten indirildiğinde çalışmayacaktır. Bunun için ise şunu kullanılabilir; `keytool -list -v -keystore path/file_name`

Uygulamayı İmzalama ve Paketleme

Project Explorer'dan projeyi seçtikten sonra sağ tıklayıp Android App Store dedikten sonra karşımıza şu ekran gelecektir.

Uygulamanın nereye çıkarılacağını, Keystore'un bulunduğu yeri keystore şifresini ve alias (takma) ismini girdikten sonra "Tamam" diyerek apk'nın belirtilen konuma çıkarılması beklenir. Daha sonra ise uygulamanızın doğru imzalandığını ve yükleme için hazır olduğunu doğrulamak için ise uygulamanızın yolunu girerek doğrulanır.

Bu işlemi yaptıktan sonra jar verified mesajını görülmelidir. Görüldüğü takdirde uygulama Google Play’de yayınlamak için hazırdır.

```
jarsigner -verify -verbose path/yourapp.apk
```

EK 15.2 iOS için Paketleme (Deploy)

Apple Geliştirici Hesabına kayıt: Uygulamayı App Store’da dağıtmak veya bir cihaz üzerinde test etmek için Apple geliştirici hesabına kayıt olmak gerekmektedir.

Apple Geliştirici hesabına kayıt olma;

- <https://developer.apple.com/membercenter/> adresine giderek Apple geliştirici hesabında Oturum açılır.
- Seçenekler arasından iOS Provisioning Portal’ı seçilir. Bu seçenek altında uygulamanın bütün ayarları bulunur.
- Hazırlık (provisioning) portal’ındaki Menüden App ID’lerinden Yeni ID’ye (New App ID) tıklanır.
- Uygulamanın detayları buraya girilir. Burada önemli olan ise Paket yapma ID’si (Bundle Identifier) ile tiapp.xml’deki uygulam id’sinin (application id) aynı olmasıdır.

Development/Distribution Sertifikalarını Yükleme

iOS uygulamalarını çalıştırmak ve yüklemek için sertifikaların yüklenmesi lazımdır.

Sertifakaların (Development/Distribution) yüklenmesi:

- Apple developer sitesinden iOS provisioning portal’a giriş yapılır.
- Kodu test etmek istenilen her cihazı Apple Developer’a kayıt etmek gerekmektedir. Menüden Aygıtları (Devices) seçerek Yeni Aygıt’a (Add Devices) tıklanır.
- Cihazların benzersiz ID’sini (Unique Device ID (UUID)) bu alana eklemek gereklidir.
- Hazırlık (Provisioning) menüsünden Development sekmesine tıklanır.

- Yeni görüntü (New Profile) düğmesine tıklanır. Burada yeni bir görüntü oluşturmak için bir form açılacaktır.
- Development sertifikasına isim verdikten sonra Uygulama ID'nizi (App ID) seçilir ve uygulamanın çalıştırılmak istendiği cihazlar seçilir.
- Daha sonra ise oluşturulan sertifika indirilir.

Cihaz Üzerinde Test işlemi

- iOS cihazı Mac bilgisayara bağlanır.
- Project Explorer menüsünden projeyi seçilip. Çalıştır (Run) düğmesine basıp iOS Devices'a tıklayıp kayıtlı cihazlardan bağlı olanı seçilir.
- İlk açıldığında iOS SDK sürümü seçilmesi gerekmektedir
- Hazırlık (Provisioning) sekmesi altında ise indirilen Development Certificate seçilir.
- Bitir(Finish) düğmesine bastıktan sonra ise uygulama cihaza yüklenecektir.
- Sertifikalar (Certificates) sekmesi altında oluşturulan sertifikalar seçilir.

iOS Uygulamasının Yayınlanması (Distribution)

Bu kısımda, iOS uygulamasının App Store'da yayınlanması anlatılacaktır. Apple geliştiricilere uygulamalarını dağıtmak için üç farklı yol sunar. Bunlar:

1. App Store veya iTunes Store için yayınlama,
 2. **In House:** Uygulamalarınızı belirli bir grup için yayınlama,
 3. **AdHoc:** Test için belirli bir sayıda cihaz için yayınlama.
-
1. Şimdiye kadar oluşturulmadı ise bir “distribution sertifikası” oluşturmak ve indirmek gereklidir.
 2. Uygulamayı yayınlamadan önce uygulamayı ve tanıtım dökümanlarını iTunes Connect'e kayıt etmek gereklidir.
 3. Uygulama oluşturmak için, Apple ID'sini kullanarak iTunes ortam oynatıcısına (<https://itunesconnect.apple.com/>) erişim sağlanır.
 4. Mevcut seçeneklerden Manage Your Application'i seçilir, ardından Ekle/Yeni

uygulama (Add NewApp) düğmesine tıklanır.

5. Uygulamanın ayrıntılarını girilir. SKU sayısı (ürün çeşit sayısı) uygulamanın kimliğidir. Daha sonra uygulamanın tanıtım dökümanları, iletişim bilgileri, uygulama simgesi, ekran görüntüleri vb eklenir.

6. Bu işlem de tamamlandıktan sonra, uygulama yüklemeye hazır hale gelecektir (Prepare for Upload). Bu duruma geldiğinizde yükleme bekleniyor durumu yazacaktır. (Waiting for Upload). iTunes Connect'den uygulamayı seçip Current Version panelinden Detayları görüntüle (View Details) seçilir. Sürüm ayrıntı (Version Detail) ekranında Kod yüklemeye hazır (Ready to Upload Binary) düğmesine tıklanır. Artık uygulama gönderilebilir.

7. Bu işlemleri tamamladıktan sonra, Titanium Studio'ya dönün. Project Explorer Menüsünden Yayınla (Publish) butonuna tıklanır. Açılan menüden Distribute-Apple iTunes Store'u seçilir.

8. Açılan pencerelerde ileri düğmesi tıklanır. Provisioning açılır penceresinde ise oluşturulan ve indirilen Distribution sertifikası eğer listede yoksa göstermek gerekmektedir. Bütün ayarları doğrulayıp onayladıktan sonra Yayınlaya tıklanır.

9. Titanium'da terminali kontrol edilir.

```
[TRACE] : __ _** BUILD SUCCEEDED **
```

```
[INFO] : __ _Finished building the application in 45s 563ms
```

```
[INFO] : __ _Packaging for App Store distribution
```

Eğer bu mesaj görülürse uygulamanız yükleme için hazırdır. Unutulmamalıdır ki paketlenmiş uygulama simülatör'de ya da cihazda çalıştırmamalıdır.

10. XCode'dan Windows>Organiser ve açılan ekrandan Arşivler (Archives) düğmesine tıklanır.

11. Uygulamayı seçip Doğrulama'ya (Validate) tıklanır.

12. Herhangi bir sorun yok ise Dağıt (Distribute) düğmesine tıklanır. iOS App Store Mağazasına gönder (Submit to the iOS App Store'u) seçilir ve ileri düğmesine tıklanır. Kullanıcı girişi yapıldıktan sonra ileri düğmesine tıklanır.

13. Uygulama böylece yüklenmiş ve Apple'in incelemesi için sıraya alınmış olur.