

Araştırma Makalesi / Research Article

**Mobil Cihazlar için Uygulama Geliştirmekte Kullanılan Platformların ve Dillerin
Karşılaştırılması**

Sait Mahmut ÇINAR^{1*}, Hilmi BİLİCİ²

¹Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi, Elektrik Mühendisliği Bölümü, Afyonkarahisar, Türkiye,

ORCID ID: <https://orcid.org/0000-0002-8683-3884>

² Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, İnternet ve Bilişim Teknolojileri Yönetim ABD, Afyonkarahisar, Türkiye

Geliş/ Recieved: 30.06.2020;

Kabul / Accepted: 12.07.2020

ÖZET: 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 kullanılacak çok sayıda platform mevcut olup söz konusu 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) ve platformun ücretli ya da ücretsiz olması gibi sıralanabilmektedir. Bu makalede 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.

Anahtar Kelimeler: Mobil uygulama, Mobil uygulama geliştirme dilleri, Mobil duyarlı web sitesi.

*Sorumlu yazar / Corresponding author: smcinar@aku.edu.tr

Bu makaleye atıf yapmak için /To cite this article

Çınar, S.M., Bilici, H. (2020). Mobil Cihazlar için Uygulama Geliştirmekte Kullanılan Platformların ve Dillerin Karşılaştırılması. Journal of Materials and Mechatronics: A (JournalMM), 1(1), 42-54.

Comparison of The Platforms and Languages Used on The Developing Mobile Applications for Mobile Devices

ABSTRACT: 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 paper, 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.

Keywords: Mobile Application, Mobile application developing languages, Mobile Responsive Web Sites.

1. GİRİŞ

Bilişim Teknolojileri ve İletişim Kurumu tarafından 2017 yılı üçüncü çeyreği için yayınlanan “Türkiye Elektronik Haberleşme Sektörü 3 Aylık Pazar Verileri Raporu” verilerine göre 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 (İnt. Ky. 1). Ayrıca 3G ve 4.5G hizmeti kullanan abonelerin 2017 Aralık ayı itibarıyla %97’sinin akıllı telefonlara sahip oldukları tahmin edilmektedir. Bu durum, kullanıcıların web sayfalarını ziyaret etmek, alışveriş yapmak ve sosyal medya kullanımı gibi işlemlerini bilgisayarlar yerine mobil aygıtlar üzerinden gerçekleştirdiğinin bir göstergesi olarak değerlendirilebilir.

Ülkemizde 2018’de bir yılda indirilen mobil uygulama sayısının 400 milyona çıkması beklendiği düşünüldüğünde ve milyarlarca cihazın kullandığı bu işletim sistemleri için üretilen yazılımlardan hangisini kullanmalı ve nereden başlamalı soruları, mobil uygulama geliştirici olmak isteyenler ve bu makalenin için çıkış noktası niteliğindedir.



Şekil 1. Geçmişten günümüze mobil cihazlar

Mobil cihazlar; akıllı telefonlar, tabletler, e-kitaplar, telefonlar, PDA'lar ve netbook'ları içeren ve çeşitli işlevleri gerçekleştiren cihazlardır (Şekil 1). Akıllı telefonlar, 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.

2018'in ikinci çeyreğinde İnternet, sosyal medya ve mobil kullanıcı istatistiklerine bakıldığında; 4.08 milyar internet kullanıcısı, 3.29 milyar sosyal medya kullanıcısı, 5.06 milyar mobil kullanıcısı ve 3.08 milyar mobil sosyal medya kullanıcısı, dünya nüfusunun sırasıyla %54, %43, %66 ve %41'ine karşılık geldiği görülmektedir (İnt. Ky.1).



Şekil 2. Mobil işletim sistemleri

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' de logoları görülen mobil işletim sistemleri Android, iOS ve Blackberry gibi işletim sistemleridir.

İnsanlar günümüzde ortalama olarak 2-5 saatlerini mobil cihazlar ile internette vakit geçirmekte ve bir insan günde 150 ila 500 defa arasında telefonunu kontrol etme ihtiyacı hissetmektedir. 2009 yılında web site trafiklerinin mobil cihaz payı %1 bile değilken, 2018 yılı itibariyle web site trafiklerinin yarıya yakını mobil cihazlar üzerinden sağlanmaktadır. Ayrıca mobil cihazlar ile internet kullanım oranı artarak devam etmektedir. Cep telefonlarında harcanan zamanın %80'i ise mobil uygulamalarda geçirilmektedir (İnt. Ky. 2). Yine istatistiklere göre mobil cihaz kullanıcılarının %79'u ürün siparişi için cihazlarını kullanmaktadır. Bu veriler göz önünde bulundurulduğunda günümüz mobil uygulamaları her ölçekte girişimci için önem arz etmektedir. Ticari açıdan bakıldığında web sitelerine oranla mobil uygulamalar, müşterilerine daha yakın olduğu açıktır.

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 verilemediği ve ülkemizde mobil yazılımlar konusunda bir boşluk olduğu görülmektedir. Konunun bu derce önemine karşılık, şimdiye kadar kapsamlı bir karşılaştırma bibliyografyası hazırlanmamıştır.

Charland ve LeRoux, 9 ayrı mobil işletim sisteminde yazılım hazırlamak için gerekli platformları sunmuş ve sadece JavaScript betik dilinin değerlendirilmesini yapmıştır(Charland ve LeRoux 2011) Klein çalışmasında sadece taşınabilir aygıtlar için web sitesi mi yoksa mobil uygulama geliştirme mi sorularına yanıtlar aramış ve mobil uygulama geliştirmenin artıları ve eksileri üzerinde durmuştur. (İnt. Ky. 3). Demir ve Akpınar makalelerinde lisans öğrencilerinin mobil cihaz öğrenmeye yönelik tutumlarını ölçebilecek bir ölçek geliştirilmiş ve bu ölçeği mobil cihazların eğitime olan katkısını araştırmakta kullanmışlardır (Demir ve Akpınar 2016). Bu konuda en kapsamlı çalışmanın ise Keskin ve Kılınç tarafından mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırıldığı ve örnek uygulamaların verildiği çalışma olduğu görülmektedir (Keskin ve Kılınç 2015). Fakat adı geçen makalede sadece mobil öğrenme uygulamalarını geliştirmeye yönelik bir çalışma yapılmıştır. Gezgin ve Bağmen tarafından hazırlanan başka bir çalışmada mobil öğrenme

uygulamalarının geliştirilmesinde kullanılan yöntemler ve platformların karşılaştırılmaları yapılmıştır (Gezgin ve Bağmen 2014).

Bu çalışmada ise mobil uygulama geliştirmekte kullanılan elliden fazla çerçeve, ondan fazla geliştirme platformu ve editör incelenmiş, öğrenme olanakları, süreleri, avantajları ve dezavantajları ayrıntılı biçimde karşılaştırılmıştır. Ayrıca karşılaştırılması yapılan platformlarla temel seviyesinde uygulamalar tasarlanarak mobil programlamaya başlayacak kişilere yol gösterici bir kaynak oluşturulmaya çalışılmıştır.

Makalenin “2. MATERYAL VE YÖNTEM” başlığı altında; mobil programlamada kullanılan diller, çerçeveler (frameworks), çevrimiçi program geliştirme platformları, mobil oyun geliştirme çerçeveleri ve örnek bir uygulama sunulmuştur. Devamında “3. BULGULAR VE TARTIŞMA” başlığında programlama dilleri, çerçeveler, çevrimiçi (online) program geliştirme platformlar ve mobil oyun geliştirme çerçevelerinin tablolar eşliğinde karşılaştırmaları yapılmıştır. Son olarak “4. SONUÇ” bölümünde elde edilen karşılaştırma sonuçları üzerinde analizler ve geleceğe yönelik projeksiyonlar sunulmuştur.

2. MATERYAL VE YÖNTEM

Bu bölüm; mobil programlamada kullanılan diller, çerçeveler, çevrimiçi program geliştirme platformları, mobil oyun geliştirme çerçeveleri ve örnek bir uygulamanın sunulduğu beş alt başlıkta düzenlenmiştir. Mobil uygulamalarda kullanılan diller kısmında; mobil platformlara göre ve proje yapısına göre programlama dilleri ayrıntılı olarak verilmiştir. Çerçeveler alt başlığında bilgisayara kurulabilen 20 çerçeve ismi verilip genel özellikleri tanıtılmıştır. Çevrimiçi program geliştirme platformları alt bölümünde ise 15 platform ismi üzerinde durulup bunlarla ilgili bilgiler verilmiştir. Mobil oyun geliştirme çerçeveleri başlığı altında 15 oyun geliştirme çerçevesi tanıtılmıştır. Son olarak örnek bir uygulama başlığında temel seviye bir uygulama kodu verilmiş ve geliştirme süreci anlatılmıştır. Mobil uygulama geliştirmekte kullanılan programlar, çerçeveler, çevrimiçi platformlar, mobil oyun geliştirme çerçeveleri ve bunlarla ilgili örnek uygulamaların ayrıntılarına (Bilici 2019)’dan ulaşılabilir.

2.1 Mobil programlamada kullanılan diller

Mobil uygulamalarda performans, görünüm ve etkileşim (kullanıcı dostu ara yüzler), 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 1’de verilmiştir.

Çizelge 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++, Phyton, 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

Çizelge 1’de mobil platform tipine göre en çok kullanılan programlama dilleri ve bu programlama aracının nereden temin edilebileceği görülmektedir. Mobil platformların neredeyse tamamında C tabanlı diller kullanıldığı görülmekle birlikte Java tabanlı yaygın kullanıma sahip diller de dikkat çekmektedir.

2.1.1 Proje yapısına göre programla dilleri:

Proje yapısına göre programlama dillerini; Mobil web uygulaması, Yerel (native) ve Melez (hybrid) uygulama geliştirme şeklinde üç kısımda incelemek mümkündür.

Mobil web uygulaması geliştirme: Uygulamanın HTML5, CSS3, Javascript gibi web teknolojileri ile yazılarak tarayıcılarda çalışacak halde sunulması işlemidir. Mobil cihazların tarayıcıları ile siteye girildiğinde tarayıcı istemciye göre (user agent) mobil tarayıcı olduğunu anlayıp ona göre web sitesi olarak değil de web uygulaması olarak açılması sağlanır. Duyarlı (responsive) web site tasarımı tercih edilebilmesi hem daha pratik hem de daha az maliyetli olmaktadır.

Yerel (native) uygulama geliştirme: Mobil uygulamaların farklı özellikteki cihazlar (Apple IOS, Google Android, RIM BlackBerry ve Windows Mobile gibi mobil işletim sistemlerine sahip) için ayrı ayrı yazılmasına, yazılımsal ve donanımsal olanaklarına doğrudan erişmesine yerel uygulama geliştirme yöntemi denmektedir. Yerel uygulamalar cihazın kendi ana platformu tarafından desteklenmiş bir dil ile yazılmaktadır. Örneğin; Java ile geliştirilen Android uygulamaları veya Objective C veya Swift ile geliştirilen iOS uygulamaları gibi.

Melez (hybrid) uygulama geliştirme: Tek kod ile yazılan ve birden fazla platformda (Apple IOS, Android, Windows, vb.) çalışabilen uygulamalara melez uygulamalar denir. Bu uygulamalar HTML5, CSS3 ve JavaScript’in gücü ve yaygınlığı üzerine kurularak mobil cihazlarda tercih edilmektedir. Yerel uygulamalar gibi sistem kaynaklarına doğrudan değil de bir ara katman sayesinde erişilmektedir. Çapraz (cross-platform) uygulama geliştirme platformu olarak da bilinen bu geliştirme yönteminde, tüm platformlar için tek dil kullanarak yerel uygulamalar oluşturulmakta ve aynı sonuç üretilmektedir.

Çizelge 2. Yerel ve melez mobil uygulamaların karşılaştırılması

Yerel Uygulamalar	Melez Uygulamalar
Platforma bağlı geliştirme	Platformlardan bağımsız geliştirme
Yüksek performans	Sınırlı performans
Geliştirme ve bakım yüksek maliyetlidir	Geliştirme ve bakım düşük maliyetlidir
Uygulama geliştirme karmaşık ve zaman alıcıdır	Hızlı geliştirme ve bakım kolaylığı
Platforma özgü dil ve araçları bilmek gereklidir	HTML, CSS, JS bilmek yeterlidir
Geliştirmiş olduğu platformdaki API'lere ulaşır	Cordova API ile cihaz özelliklerine erişme kullanma imkânı sağlar.

Çizelge 2’de Yerel veya melez uygulama geliştirme süreciyle ilgili karşılaştırmalar ve uygulama geliştirmenin birbirine göre üstün ve zayıf yönleri açıkça sunulmuştur. Örneğin yüksek performans gerektiren uygulamaların yerel olarak tasarlanması yerinde olacakken hızlı geliştirme ve bakım kolaylığı tercih ediliyorsa melez uygulama tasarımı tercih edilebilir.

2.1.2 Mobil platforma göre programla dilleri:

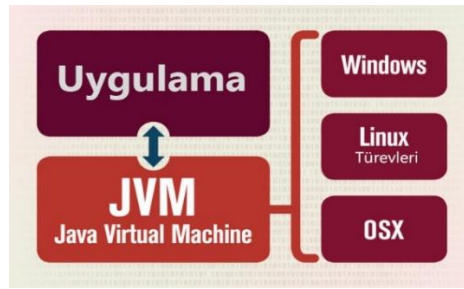
Burada Android, IOS, BlackBerry OS ve Windows phone platformları için altı farklı programla dili verilecektir.

Android platformu için programlama dilleri: Android tabanlı sistemler için yazılım geliştirme süreci aşağıdaki gibi sıralanabilmektedir;

- Java dosyalarında, program mantığı/uygulamanın yapması istenilenler tanımlanır.
- XML dosyalarında sayfa düzeni (mizanpaj)/görünümü tasarlanır.
- Uygulama yazıldıktan sonra, tüm dosyaları derlemek ve bunları Android cihazlarda çalıştırabilecek ve/veya Google Play'de yayımlayabilecek bir .apk dosyasında bir araya getirmek için derleme aracını kullanmak gerekir.
- Tümüleşik bir geliştirme ortamı (Integrated Development Enviroment-IDE) ile birlikte Android için bir uygulama oluşturmakta kullanılan tüm yardımcı programlar ve dosyaları sunabilir. IDE, kod dosyalarınızı düzenlemek, derlemek ve çalıştırmak için açtığınız bir programdır.
- Önceleri, Android geliştirme için standart IDE Eclipse kullanılmaktayken şimdilerde bir Google ürünü olan ve daha işlevsel bir Android Studio platformu kullanılmaktadır.

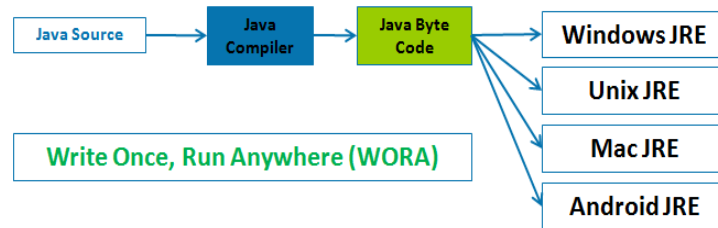
Android platformunda Java ve Kotlin dilleri kullanarak uygulama geliştirme ayrıntıları aşağıda verilmektedir.

Java Programlama dili ile yerel olarak Android uygulama geliştirmek mümkündür (Şekil 3). Java ile programlama yaparken Java sanal makinesi (Java Virtual Machine-JVM), Java çalışma zamanı ortamı (Java Runtime Environment-JRE) ve Java geliştirme kiti (Java Development Kit-JDK) terimlerinin bilinmesi gereklidir.



Şekil 3. İşletim sistemleri ve JVM

JVM, Java'nın en temel ve en cazip özelliği olan her işletim sisteminde çalışabilmesini sağlayan yapıdır. *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 (<https://www.javaworld.com/article/3272244/core-java/what-is-the-jvm-introducing-the-java-virtual-machine.html> Erişim: 27/12/2018). Hangi işletim sistemi veya donanımın mevcut olduğuna bakılmaksızın *JVM*, programların içinde çalışması için öngörülebilir bir ortam oluşturur. *JVM*, bir kere yaz, her yerde çalıştır (Write Once, Run Anywhere-WORA) prensibiyle çalışır (Şekil 4).



Şekil 4. WORA prensip blok şeması

JRE, Java teknolojisiyle yazılmış uygulamaları ve küçük uygulamaları (applet) çalıştıran bilgisayar kullanıcılarına gerekli ücretsiz temin edilen bir JVM uygulamasıdır. *JRE*, Java programlarını çalıştırmak için Java sınıflarından oluşan bir kütüphaneye sahip bir uygulamadır.

JDK, Java geliştiricilerine yönelik Oracle firması tarafından hazırlanmış bir üründür. Java'nın kullanıma sunulmasından beri en geniş kullanım alanına sahip *JDK*'dir. *JDK*, *JRE*, Java Compiler, hata ayıklama aracı ve diğer gerekli araçlardan oluşur. Yazılan “.java” uzantılı kod derlenip JVM'in okuyabileceği “.class” uzantılı hale getirilmesi için *JDK*'ya ihtiyaç duyulur. *JDK* ilk olarak 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 içinde barındırır.

Kotlin, 2010 yılında Rus şirketi JetBrains'ten tarafından hazırlanmaya başlanan genç bir dildir ve 2011 yılında “JVM Language Summit” etkinliğinde duyurulmuştur. Google I/O 2017 konferansında, Android geliştirme ekibi, Kotlin'in Android uygulamalarının geliştirilmesi için resmi destek aldığını bildirmiştir. Kotlin Java'ya göre daha kısa ve daha özgün şekilde kodlama yapılabilir. Kotlin programlama dili ile JVM uygulamaları (Server-side uygulamalar), Android uygulamaları, browser uygulamaları (JavaScript tabanlı web uygulamaları), yerel uygulamalar (MacOS, iOS ve gömülü sistemler uygulamaları) geliştirilebilmektedir. Java, C ve C++ gibi, Kotlin statik olarak yazılmış bir dildir. Hem nesne yönelimli hem de prosedürel programlamayı destekler.

IOS platformu için programlama dilleri: iOS, Apple firmasının mobil platformlar için geliştirdiği bir mobil işletim sistemidir. Çekirdeğini MAC OSX den alan iOS içinde dört katman bulundurmaktadır;

- Core OS tabakası, Sockets ve dosya sisteminin olduğu nesneye yönelik olmayan katmandır.
- Core Servisleri tabakası, nesneye yönelik ara yüzlerle bu işletim sisteminin çoğunu kapsadığı katmandır.
- Medya tabakası, ses, müzik, video gibi dosyaların çalıştırıldığı katmandır.
- 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 kelimesinin kısaltması olan bu sistem nesnelere yaratıldıktan sonra onların hafızadaki durumunu düzenleyen sistemdir. Bu sistem sayesinde her seferinde nesnelere elle bellekten çıkarılmasına gerek kalmamakta ve programcıya büyük bir kolaylık sağlamaktadır.

Objective-C, C tabanlı, yansımali ve nesne yönelimli bir programlama dilidir. 1983 yılında geliştirilmeye başlanan dil ObjC, Objective C ve Obj-C olarak da isimlendirilmektedir. 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 bağlamayı sağlar. Aynı zamanda nesne grafik yönetimi ve nesne hazırlıkları için dil seviyesi desteği de ekler.

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.

BlackBerry OS platformu için programlama dilleri: 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 simülatörde hem de doğrudan cihazda yapılabilir.

Windows Phone platformu için programlama dilleri: 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. Ortak çekirdek ve uygulama yürütme sistemi sayesinde, bir kez yazılmış olan bir UWP uygulaması Windows 10 çalıştıran tüm cihazlarda çalışacaktır.

2.2 Çerçeveseler

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 makale ile popüler geliştiricilerden Android ve IOS platformlarına uygulamalar geliştirmek için kullanılan popüler çerçeveleri bir arada ve karşılaştırılmalı olarak ele alınmaktadır. Çerçevelerin genel özelliği Android, iOS ve Windows Universal platformları için uygulama yapılabilmesidir.

Bilgisayara kurulumu yapılabilen çerçeveler; NativeScript, Flutter, Cordova, Appcelerator, Titanium, Ionic, Alpha Anywhere, Xamarin, PhoneGap, Sencha Touch, jQueryMobile, IBM Worklight, Corona, ENYO, LUNGO, React Native, Kivy, JQTouch, Alpha Anywhere, Xojo ve RhoMobile şeklinde sıralanabilmektedir.

2.3 Çevrimiçi 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 sunmaktadır. Mobil uygulama pazarına girmek için binlerce dolar ve aylarca çalışmaya gerek yoktur. Hızlı bir şekilde ve özel programlama bilgisine sahip olmadan uygun bir bütçe içinde uygulama oluşturmanıza yardımcı olacak birçok mobil platform bulunmaktadır. HTML5, Javascript, CSS3 programlama dillerinin mobil web alanında karşılaştırılması ve hibrit mobil uygulama görüşlerinin pek çok platformda farklı şekilde sunulması kafaları karıştırmaktadır. 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. Çok bilinen çevrimiçi mobil uygulama oluşturma platformları; Appery.io, Mobileroadie.com, The AppBuilder, Good Barber, ShoutEm, Appy Pie, AppMachine, GameSalad, BiznessApps, AppMakr, App Press, EachScape, iBuildApp, QuickBase, Salesforce1 ve Lightning şeklinde sıralanabilmektedir.

2.4 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; Unity, Unreal Engine, Corona SDK – The 2D Game Engines, SpriteKit – 2D sprite-based games, Marmalade SDK, Buildbox, AppGameKit, Construct 2, Fusion, GameMaker Studio, CocoonJS, MonoGame, Amazon Lumberyard, Cocos2D-x ve Haxe olarak sıralanabilir.

2.5 Örnek bir uygulama

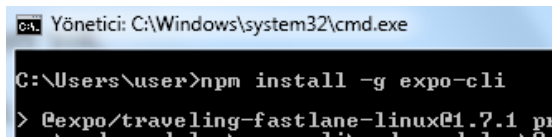
Örnek bir çerçevenin kurulum ve geliştirme süreci;

React native Facebook firması tarafından 2015 yılında geliştirilmiş hem Android hemde IOS'a hizmet verebilen yerel bir mobil platformudur. Aynı zamanda bir JavaScript kütüphanesidir. Geliştiricilerin işlemleri daha kolay ve hızlı yapmaları amaçlanmıştır. React native kurulumu aşağıdaki gibi gerçekleştirilebilir.

1. Npm kullanarak komut satırına şu kod yazılır: npm install -g expo-cli

2. Ardından "AwesomeProject" adlı yeni bir React Native projesi oluşturmak için sıra ile aşağıdaki komutları çalıştırılır (Şekil 5);

```
expo init AwesomeProject
cd AwesomeProject
npm start #you can also use: expo start
```



Şekil 5. Komut tabanlı ara yüz kullanımı

Uygulamayı çalıştırması için Expo istemci uygulamasını iOS veya Android telefona yüklenmesi ve bilgisayarla aynı kablosuz ağa bağlanması gerekir. Android'de, projeyi açmak için terminaldeki QR kodunu taramak için Expo uygulaması kullanılır. IOS'ta ise bağlantı almak için ekrandaki talimatlar izlenir:

Örnek kodlama aşağıda verilmiştir.

```
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',
  }
});
```

Uygulama başarıyla çalıştırıldıktan sonra değişiklikler yapılabilir. Tercih edilen metin editöründe App.js dosyasını açılır ve düzenlenir. Değişiklikleri kaydettikten sonra uygulama otomatik olarak yeniden yüklenmelidir. Kodda değişiklik yapıp kaydedildiğinde anında cep telefondaki uygulama da güncellenen sürüm gösterilmektedir.

3. BULGULAR VE TARTIŞMA

Bu makalede Android, IOS ve diğer platformlara ait uygulama geliştirmek için kullanılan popüler çerçeveler, diller ve veri tabanları karşılaştırılmalı olarak sunulmuştur.

Çizelge 3’de çevrimiçi ve çevrimdışı çerçevelerin karşılaştırılması sunulmaktadır. Çizelge incelendiğinde hızlı ve tek bir uygulama geliştirilmesi gerektiğinde çevrimiçi 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.

Çizelge 3. Çevrimiçi ve çevrimdışı 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	Çerçeveyi 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

3.1 Mobil Geliştirme Platformları için Dikkat Edilecek Noktalar

1. Programlama dili: Çok fazla 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. Arka uç ve ön uç (Back-end - front-end): Bir geliştirici 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 odaklı 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 daralacaktır.

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, uygulamanın 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. Simülatörler uygulamanın davranışını çoğaltmaya çalışır, emülatörler 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 eder.

6. Hız önemlidir ve giriş animasyonlarını kısa tutmak, 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.

3.2 Mobil Geliştirme Platformları için Dikkat Edilecek Noktalar

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 uygulamalar her cihazda test edilir,
- Arka uç (Back-end) yetenekleri yönetilir ve kontrol edilir
- 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.

Çizelge 4'de mobil işletim sistemlerine göre geliştirme platformları ve programlama dilleri karşılaştırmalı olarak sunulmuştur. Hem Android hem de iOS mobil işletim sistemleri için çok sayıda geliştirme platformu ve programla bulunmaktadır ve Android yazılımlar hemen hemen her cihazda çalışma kabiliyetine sahipken iOS uygulamalar sadece Apple firmasının ürünlerinde çalışabilmektedir.

Çizelge 4. 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	Çerçeve ve açık kaynak
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 tarihi: 16/05/2019

**Morgan Stanley Medya raporları.

*** 2017 Dünya geneli mağaza uygulama mağazası gelirleri.

Çizelge 5'te Ionic, Cordova ve PhoneGap geliştirme ortamlarının karşılaştırması verilmiştir. Ücretsiz olarak kullanılabilen bu üç platformun birbirlerine göre üstün ve zayıf yönleri bulunmaktadır. Bu üç geliştirme platformunun ortak noktaları HTML, CSS ve JavaScript ile geliştirilmesine olanak tanınmaktadır.

Çizelge 5. Ionic, Cordova ve PhoneGap ortamlarının karşılaştırılması

	 Ionic	 Apache Cordova	 PhoneGap
Nedir?	Açık kaynak kodlu ve ücretsiz, 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.		
Neden seçilmeli?	<ul style="list-style-type: none"> Hızlı prototipleme Ücretsiz Tasarım Süreklilik Java Script 	<ul style="list-style-type: none"> Eklenti çokluğu Topluluk Kolay erişim Ücretsiz Java Script 	<ul style="list-style-type: none"> Adobe tarafından desteklenmesi Ücretsiz Kolay Java Script
Eksiler	<ul style="list-style-type: none"> Yüksek performanslı veya kullanıcı ara yüzü yoğun uygulamalar için uygun değil Oyun geliştirme amaçlı değil 	<ul style="list-style-type: none"> Bir yerel uygulama kadar hızlı değil 	<ul style="list-style-type: none"> Bir yerel uygulama kadar iyi değil
Alternatifleri	<ul style="list-style-type: none"> 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. 		

4. SONUÇ

Bu çalışmada 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 veri tabanı incelenerek kurulumları, öğrenme olanakları ve süreleri, avantajları ve dezavantajlarına 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 yazılımcıların tüm taşınabilir cihazları, özellikleri, mobil işletim sistemlerini, uygulama geliştirme için kullanılan dilleri ve platformları iyi tanınması gerekmektedir.

5. KAYNAKLAR

- Bilici, H., Mobil Cihazlara Uygulama Geliştirmek için Kullanılan Platformların ve Dillerin Karşılaştırılması, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi (Basılmış), 2019.
- Charland A., LeRoux B., Mobile application development: web vs. native, Communications of the ACM Magazine, 54 (5), 49-53, 2011.
- Demir K., Akpınar E., Mobil Öğrenmeye Yönelik Tutum Ölçeği Geliştirme Çalışması, Eğitim Teknolojisi Kuram ve Uygulama, 6 (1), 59-79, 2016.
- Gezgin D.M., Bağmen E., Mobil Öğrenme Uygulamalarının Geliştirilmesinde Yöntemler ve Platformlar, 2nd International Instructional Technologies & Teacher Education Symposium, 20-22 Mayıs 2014, Afyonkarahisar.
- İnt. Ky. 1. <https://www.btk.gov.tr/uploads/pages/slug/2017-q3-5a5f1e2baba7b.pdf> (Erişim Tarihi: 02/05/2019).
- İnt. Ky. 2. <https://wearesocial.com/blog/2018/01/global-digital-report-2018> (Erişim Tarihi:12/12/2018).
- İnt. Ky. 3. <http://www.adobe.com/inspire/2012/02/mobile-websites-vs-mobile-apps.html>. Erişim tarihi: 20.05.2019.
- Keskin, N.Ö., Kılınç, A.G.H., Mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırılması ve örnek uygulamalar, Açıköğretim Uygulamaları ve Araştırmaları Dergisi, 1(3), 68-90, 2015.