

MOBİL UYGULAMA GELİŞTİRME, YAYIMLAMA VE EKONOMİK GELİR ETME AŞAMALARININ İNCELENMESİ: İOS VE ANDROID SİSTEMLERİNİN KARŞILAŞTIRMASI

Tahsin ÇİLOĞLU
Bartın Üniversitesi, Türkiye
tah-sin@hotmail.com
<http://orcid.org/0000-0001-6352-757x>

Eren ÖZEREN
Bartın Üniversitesi, Türkiye
erenozeren78@gmail.com
<http://orcid.org/0000-0001-9444-9009>

Ahmet Berk ÜSTÜN
Bartın Üniversitesi, Türkiye
ustun.ab@gmail.com
<http://orcid.org/0000-0002-1640-4291>

ÖZ

Apple şirketinin geliştirdiği akıllı telefonların yayılması ve Google'ın Android işletim sistemini piyasaya sürmesinden sonra mobil uygulamalar hızlı bir şekilde artış göstermiştir. Bu bağlamda, hızla gelişen bilişim dünyasının önde gelen mobil uygulama platformlarından Android işletim sistemi ve iOS işletim sistemi tabanlı uygulamaların yazılım geliştirme açısından karşılaştırılması, uygulama geliştiricilerin bu platformlarda uygulamalarını yayımlama aşamalarının kıyaslanması ve yayımladıktan sonra bu uygulamalardan gelir elde etmek için izlenilmesi gereken yolların karşılaştırılması yapılmıştır. Geliştirme aşamasında, Android işletim sistemi ve iOS İşletim sisteminde kullanılan yaygın programlama dilleri hakkında bilgi verilmiştir. Mobil uygulamaların yayımlanması aşamasında Google uygulama marketlerinde (Google Play) yayımlama basamakları, Apple Uygulama Marketlerinde (AppStore) yayımlama basamakları anlatılmıştır. Ekonomik gelir sürecinde ise mobil uygulama geliştiricilerin yazmış oldukları uygulamaların mağazada yer aldıktan sonra nasıl gelir sağlayabileceği açıklanmıştır. Android ve iOS tabanlı uygulamaların geliştirme, yayınlama ve gelir elde etme süreçlerinde birbirine kıyasla artı ve eksi yönleri mevcuttur. Yapılan çalışmanın Android ve iOS tabanlı mobil uygulama geliştiricilerine yol gösterici olacağı öngörülmektedir.

Anahtar Kelimeler: Mobil Uygulama Geliştirme, Mobil Uygulama Yayımlama, Uygulamadan Gelir Elde Etme, Android, IOS

INVESTIGATION OF MOBILE APPLICATION DEVELOPMENT, PUBLISHING AND ECONOMIC INCOME STAGES: COMPARISON OF iOS AND ANDROID SYSTEMS

ABSTRACT

The number of mobile applications has rapidly increased after the pervasiveness of smartphones developed by Apple Inc. and the release of Google's Android operating system on the market. The purpose of this study is to make the comparison of mobile applications developed on Android operating system and iOS operating system that are the leading mobile application platforms of the rapidly advancing IT world in terms of software development, the phase of publishing the developed

applications on these platforms and the steps that should be followed to earn income from these applications after publishing them. For the development stage, information was given about programming languages commonly used in the Android operating system and iOS operating system. For the publishing of mobile applications, the steps of publishing mobile applications on Google application markets (Google Play) and Apple Application Markets (AppStore) were explicitly elucidated. For the economic income stage, it is explained how the applications created by mobile app developers can generate revenue after they are on the application market. Android and iOS based mobile applications have positive and negative aspects compared to each other in the stage of development, publishing and revenue generation. It is predicted that the study will guide software developers who create Android and iOS based mobile applications.

Keywords: *Mobile Application Development, Mobile App Publishing, Earning Revenue From Apps, Android, IOS*

GİRİŞ

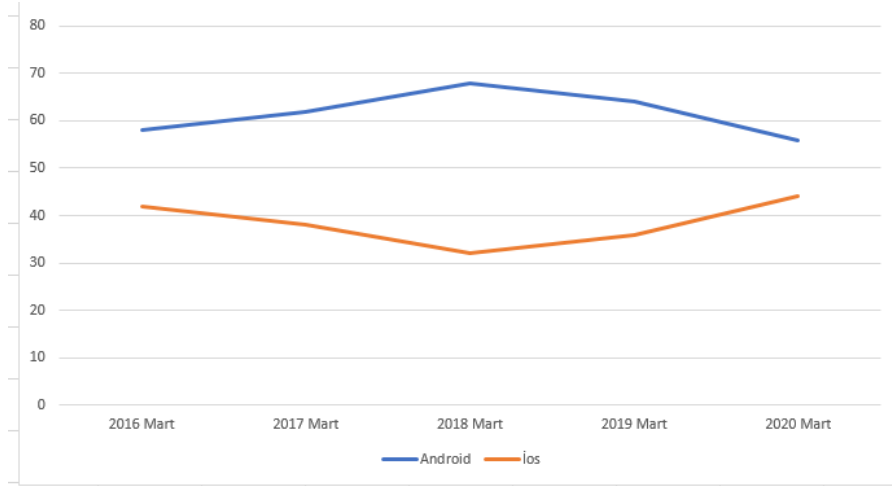
Akıllı telefonların taşınabilir kişisel asistanlar haline gelmesi sonucunda bilişim teknolojisi ile etkileşim süresi ve mobil sistemlere ilgi artmıştır (Aslay, Yıldız, Baran ve Akar, 2018). Mobil uygulamalar, mobil cihazlarda kullanılmak üzere geliştirilen yazılımlar olup, kullanılan teknoloji ve cihaz donanımları bakımından masaüstü uygulamalardan ayrılır. Güncel mobil cihazlar, kişisel bilgisayarlarda kullanılan birçok uygulamayı kullanıcılara sunarak kişisel bilgisayarlara olan ihtiyacı azaltmaktadır (Kabakuş, Doğru ve Çetin, 2015). Bu cihazlarda en yaygın olarak kullanılan işletim sistemi Google firmasına ait Android işletim sistemi ve Apple firmasına ait iOS işletim sistemidir. Android işletim sistemi 2003 yılında Andy Rubin, Rich Miner, Nick Sears ve Chris White tarafından California’da kurulmuş ve Google, Android’i 2005 yılında satın almıştır (Tufan, Baykara, Güler ve Avcı, 2012). Android işletim sistemi 2008’de 1.0 versiyonuyla kullanıma sunulmuştur (Çakır ve Tüminçin, 2017). iOS, Apple Inc. tarafından geliştirilmiş ve 2007’de kullanıma sunulmuş işletim sistemidir (Özdemir, Akadal, Çelik ve Reis, 2013). iOS işletim sistemi Linux tabanlı çalışan bir işletim sistemidir (Büyükgöze, 2019). Eski adı iPhone OS olan bu işletim sistemi ilk olarak iPhone için geliştirmiştir (Özkoçak, 2016). Daha sonra iPod Touch ve iPad’de de kullanılmıştır. Mobil cihazlarda kullanılan işletim sistemine göre hizmet veren uygulamalara ve uygulamalarla ilgili güncellemelere Android işletim sistemi için Google Play Store’dan, iOS işletim sistemi için Apple Store’dan ulaşılabilir.

Uygulamaların yayımlanacağı platformların kendine özgü kısıtları vardır. Akıllı telefonlar gibi taşınabilir cihazlar için geliştirilen ve açık kaynak kodlu mobil işletim sistemi olan Android, Linux tabanlıdır. Android platformu, izin temelli bir güvenlik modeli olan Linux işletim sistemi güvenlik modelini temel almıştır. Bu güvenlik modeline göre; kullanıcılar izni olmadan başka kullanıcıların dosyalarına erişim sağlayamaz. Android platformu, her kullanıcının kendi güvenliğini kendi izinleri ile kontrol altında tutmasına izin verirken; iOS platformu denetimi sıkı tutup Uygulama Programlama Arayüz (Application Programming Interface, API)’lerini kullanarak güvenli kod yazmaya zorlamakta ve bu şekilde zararlı uygulamaların uygulama mağazasına girmesini engellemektedir (Büyükgöze, 2019). iOS işletim sistemi kapalı sistem olarak çalışmaktadır. Buna göre iOS işletim sisteminde, uygulama geliştiriciler kodlarda değişiklik yapamazlar (Aytekin, Ayaz, Tüminçin ve Bektaş, 2019). iOS işletim sistemine kurulacak olan uygulamalar sıkı bir inceleme sürecine girerler ve denetimlerden geçen uygulamalar onaylandıktan sonra marketlerde yerini alırlar. Marketlere eklenen uygulamalar bu aşamadan sonra gelir sağlamaya başlarlar.

Analitik şirketi Appfigures’in 2017 yılında hazırlamış olduğu raporda Apple şirketine ait App Store’un ve Google şirketine ait Google Play Store’un uygulama sayısı 2017’ye kadar sürekli olarak arttığı belirtilmiştir. Google Play, yaklaşık %30 büyüme ve 3,6 milyondan fazla uygulama ile oldukça iyi bir oranda büyümeye devam ederken, App Store’un çok farklı bir yöne doğru ilerlediği raporda belirtilmiştir. 2017 yılının sonunda, mevcut olan 2,2 milyon uygulamaya kıyasla %5’lik bir düşüşle App Store’da yaklaşık 2,1 milyon iOS uygulamasının mevcut olduğu görülmüştür. Raporda bu düşüşün nedeni olarak, Apple’ın inceleme yönergelerinin daha sıkı uygulanmasının yanı sıra 64 bit mimariyi destekleyecek şekilde güncellenmemiş birçok eski uygulamayı ortadan kaldıran teknik bir değişikliğinin sebebiyet verdiği kanaatine varılmıştır. Yine bu rapora göre düşüşün tek nedeninin bu olmadığı, mobil

uygulama geliştiricilerin platform engelini aşarak uygulamaların platformlar arasında taşındığı belirtilmiştir. Raporda, iOS'tan Android'e geçen uygulama sayısı yaklaşık 17,5k, Android'ten iOS a geçen uygulama sayısı yaklaşık 7,5k'tır. 2017 yılında Android uygulamalarına geçen uygulama sayısı, Android uygulamalarının App Store'a geldiğinin iki katından fazla olduğu ve toplamda yaklaşık olarak 25k platform değişikliğinin çok büyük bir değer olmamasına karşın, geliştiricilerin platform engelini aştığı ve uygulamalarını orijinal sürümün ötesine genişlettiği görülmüştür.

Şekil 1'de Araştırma şirketi Consumer Intelligence Research Partners, LLC (CIRP, 2020) tarafından paylaşılan raporda iOS ve Android işletim sistemlerinin kullanım oranları verilmiştir.



Şekil 1. iOS ve Android Kullanım Oranları (Cirp, 2020)

Şekil 1'de, 2020 yılının ilk çeyreğinde iOS'un kullanım oranının yüzde 44, Android bir cihazın kullanım oranının yüzde 56 olduğu görülmektedir. Ayrıca CIRP araştırma şirketinin raporunda 2018 yılından sonra iOS kullanıcılarının sayısında gözle görülür bir artış olduğu belirtilmiştir. 2018'den sonra Android kullanıcılarının sayısı azalmış olsa da iOS ve Android kullanıcı sayıları karşılaştırıldığında, Android'i kullananların sayısı daha fazladır.

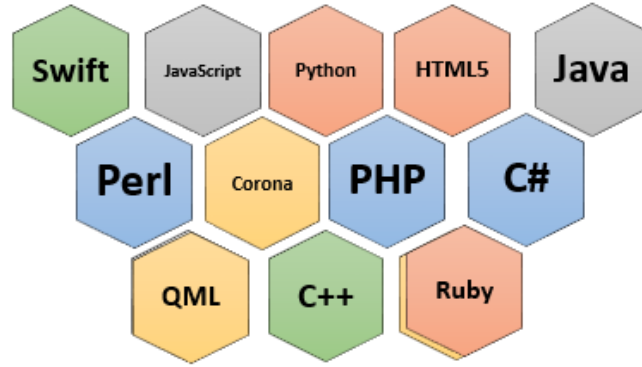
MOBİL UYGULAMA GELİŞTİRME AŞAMALARI

Cihaz kullanım sürelerinin artması, her yaşa hitap etmesi ve günlük hayatta kolaylıklar sağlaması gibi nedenlerle mobil uygulamaların sayısı her geçen gün artmaktadır. Günümüzde gelişen akıllı cihazlara mobil uygulama yazmak için bazı programlama dillerinin bilinmesi gereklidir. Geliştirilen uygulamaların kaliteli, ilgi çekici ve etkili olması için programlama dillerinin yanında uygulamaların ara yüzünde kullanılacak görsel materyallerin materyal tasarım ilkelerine göre oluşturulması gerekir. Ayrıca uygulamanın tasarımı, uygulama marketlerinde üst sıraya ilerlemesini sağlayacağı için maddi gelir elde edilmesini kolaylaştırabilir. Mobil uygulamalar her türlü ürün ve süreci daha kapsayıcı kılmayı amaçlayan evrensel tasarım ilkeleriyle hazırlanmalıdır (Şenel, Şenel ve Günaydın, 2019). Mobil uygulamanın kullanımı basit ve anlaşılır olmalı, hedef kitledeki çeşitlilik göz önünde bulundurularak rahat kullanılabilir şekilde hazırlanmalıdır. Kolay kullanılabilir, anlaşılır ve uyum sağlanabilen arayüz, kullanıcı dostu olduğundan kullanıcı ile uygulamanın etkileşimini artırır (Evren, 2016). Kullanıcı arayüzleri, insan merkezli tasarlanmalıdır (Akyol, 2014). Renk uyumu ve farklı tasarım öğeleri kullanılarak tasarımlar daha çekici hale getirilebilir. Tüm bunları sağladıktan sonra uygulamanın farklı platformlarda problemsiz çalışıp çalışmadığı kontrolünün yapılması gerekir. Farklı cihazlarda (tablet, akıllı telefon, masaüstü) kullanılan görsellerin ve metinlerin yerleşiminin yeniden şekillenmesinin sağlanması gerekir.

Android ve iOS'ta çalışan uygulamaların API'leri aracılığıyla kullanıcıların, telefonlarında depoladıkları kişisel bilgilere erişim sağlanarak kullanıcı davranışları hakkında bilgi toplanabilir (Thurm ve Kane, 2010). Uygulama geliştiricilerin, kullanıcıların telefonlarında eriştikleri verileri üçüncü bir kurum, kuruluş ve uygulama geliştiriciyle paylaşmaması gerekmektedir. Farklı platforma uygulama yazan geliştiriciler tarafından kullanıcılardan bir onay almadan kullanıcı verilerinin kullanılması tehlike

arz etmektedir. Örneğin 2011 yılında bir uygulama geliştiricinin Facebook'un mobil uygulamasını kullanarak bir kullanıcının adres defterini ve Facebook hesabını kopyaladığı ve bu durumu kullanıcının öfkeyle karşıladığı görülmüştür (Arthur, 2011). Farklı platformlarda oluşturulan uygulamaların izinlerine dikkat edilmesi gerekmektedir. Arthur'un (2011) araştırmasındaki gibi bir uygulama geliştiricisi, uygulamasında hangi izinlerin alınması gerektiğini ve kullanıcıların mobil cihazlarındaki kişisel verilerine izinsiz ulaşmaması gerektiğini bilmelidir. Mobil uygulama geliştirmek için kullanılacak programlama dilleri Şekil 2'de gösterilmiştir.

Şekil 2. Mobil Uygulamalarda Kullanılabilecek Programlama Dilleri



Kaynak: (URL 6)

Şekil 2'de görüldüğü gibi bir mobil uygulama geliştirmek istenildiğinde Swift, Perl, Qml, C++, C#, Ruby, Java, Html 5, Javascript, Python, Php ya da Corona dillerinden birinin bilinmesi gerekir.

iOS MOBİL UYGULAMA GELİŞTİRME

iOS mobil uygulama geliştirilirken Apple 2020 verilerine göre; uygulama onayının sorunsuz alınabilmesi için, süreci yavaşlatabilecek veya bir reddi tetikleyebilecek yaygın yanlış adımlar kontrol edilmelidir. Kontrol süreci uygulamanın sorunsuz yayımlanması için iyi bir başlangıçtır. Apple 2020 verilerine göre uygulama geliştirme aşamasında geliştiricilerin dikkat etmesi gereken aşamalar aşağıda sırayla verilmiştir.

- Uygulama çökme ve hatalara karşı test edilmelidir.
- Uygulama bilgilerinin ve uygulama üst verilerinin eksiksiz, doğru ve güncel olduğu teyit edilmelidir.
- Etkin bir demo hesabı ve giriş bilgileri ile uygulamayı incelemek için gerekli olabilecek diğer donanım veya kaynaklar (ör. Giriş bilgileri veya örnek bir QR kodu) sağlanmalıdır.
- İnceleme sırasında canlı ve erişilebilir olmaları için arka uç hizmetleri etkinleştirilmelidir.
- Uygun olmayan destekleyici özelliklerin ve uygulama içi satın alımların ayrıntılı açıklamalarını, uygun olduğunda destekleyici belgeler de dahil olmak üzere uygulama incelemesi notlarına eklenmelidir.
- Uygulamanın iPhone arayüzüne uygun olup olmadığı kontrol edilemelidir.

Bu adımlar izlenerek kontrol yapıldığı takdirde daha hızlı bir şekilde uygulama yayımlanma aşamasına geçilir. Ayrıca Apple, uygulamanın tasarım yönünden basit, rafine, yenilikçi ve kullanımı kolay olmasına önem verir. Uygulama bir süre sonra amaçlandığı gibi çalışmıyorsa veya artık aktif olarak desteklenmiyorsa App Store'dan kaldırılabilir.

Mobil uygulama geliştirmek için farklı programlama dilleri vardır ve her bir dilin de kendine özgü yazım kuralları vardır. Dile özgü komutların yazılması sürecine kodlama ya da programlama; ortaya çıkan son ürüne ise program ya da uygulama denir (Ersoy, Madran ve Gülbahar, 2011). Apple programlaya Objective-C ile başlamış daha sonra Swift adlı kendi ürettiği yazılım diliyle devam etmiştir (Güngör, 2017).

OBJECTIVE C

Objective-C programlama dili 1988 yılında lisanslanan Apple firmasının ürünleri (Mac bilgisayar) üzerinden uygulama geliştirilebilen yetenekli bir programlama dilidir. Mac bilgisayarın olmadığı durumlarda sanal bilgisayar üzerine kurulan OSX üzerinden Objective-C ile çalışılabilir. Temel dile nesne yönelimli özellikler ekleyen C programlama dilinin bir üst kümesidir (Egele, Kruegel, Kirde ve Vigna, 2011).

SWIFT

Object-C dilinin gramerinin yazılımcılara zor gelmesi sebebi ile 2014 yılında Swift isminde yeni bir dil yayımlamıştır (Emre ve Yeten, 2018). Swift, Linux'ta çalışan uygulamaların da yazılabildiği, kullanımı kolay ve açık kaynaklı, Objective-C, Rust, Haskell, Ruby, Python gibi birçok programlama dilinden yararlanılarak yazılmış bir programlama dilidir. Swift dünyanın en popüler programlama dillerinden biri olmasına ve deneyimli geliştiriciler sözde Swift'i anlaşılması ve benimsenmesi kolay bir dil bulmalarına rağmen araç setindeki hatalar ve anlaşılması zor veya yardımcı olmayan hata mesajları yüzünden birçok sorun olduğunu ve ayrıca Swift derleyicisinin acil olarak iyileştirilmesi gerektiğini öne sürmüşlerdir (Rebouças ve diğ., 2016).

SWIFT İLE OBJECTIVE C BENZERLİKLERİ

Objective C dili ve Swift dili arasındaki benzerlikler sayesinde herhangi birini bilen bir uygulama geliştiricisi için diğer dili öğrenmek de kolay olacaktır. Uygulama geliştiricilerin bu diller arasındaki benzerlikleri incelemeleri bu doğrultuda mobil uygulama geliştirilmesine fayda sağlayacaktır. Bu benzerlikler sırasıyla aşağıda belirtilmiş olup maddeler şeklinde sıralanmıştır.

- Basit sayısal türler (Int, Float, Double) Objective C de olduğu gibi Swift'te de bulunmaktadır.
- Benzer for...in numaralandırma sözdizimlerine sahiptirler.
- Apple ürünleri hem Swift hem de Objective C ile geliştirilmeye devam etmektedir.

Objective C ile Swift arasındaki bu benzerlikler sayesinde, uygulama geliştiriciler iki dilden birini bildikleri takdirde daha hızlı ve kolay bir şekilde diğer dili öğrenebilir. Uygulama geliştiriciler iki dili öğrenmesi mobil uygulama yazım sürecinde daha pratik ve hızlı programlama hususunda fayda sağlayabilir.

SWIFT İLE OBJECTIVE C FARKLILIKLARI

Uygulama geliştiricileri için hangi programlama dili pratik ise o programla alanına yönelmesi faydalı olacaktır. Fakat, belirli bir programla dili kullanmıyorsa, Swift ile Objective C arasında benzerlikler olduğu gibi farklılıklar da mevcuttur ve bu farklılıkları inceleyerek programlama dilini seçimi yapabilir. Bu farklılıklar 5 ana kategori şeklinde incelenebilir. Swift ve Objective C arasındaki farklılıklar, aşağıda maddeler halinde sunulmuştur.

- Birçok programlama dilinde noktalı virgül kullanımı zorunlu olmasına rağmen; Swift'de satırlar noktalı virgül (;) ile bitmek zorunda değildir.
- Objective C de bulunan Başlık dosyaları Swift'te bulunmamaktadır.
- Swift'te String ifadeler Unicode'u tamamiyle destekler.
- Syntax yapısı Objective C 'ye göre oldukça kolay ve anlaşılması basittir.
- Hafıza/Bellek Yönetimi Swift'te daha esnek hale getirilmiştir. Objective C deki gibi geliştiricilere zaman kaybettirmeden kodlarını rahat ve güvenli bir şekilde yazmalarına olanak sağlamıştır.

Swift hem yazması hem de okuması birçok programlama diline göre daha kolay olan bir dildir. Objective-C ile karşılaştırıldığı zaman, aynı işlemleri yapmak için yazılması gereken kod sayısı önemli ölçüde azalır.

Bu nedenle mobil uygulama geliştiricileri için bu basamakları dikkate alarak mobil uygulama yazmaları zaman tasarrufu sağlayacaktır.

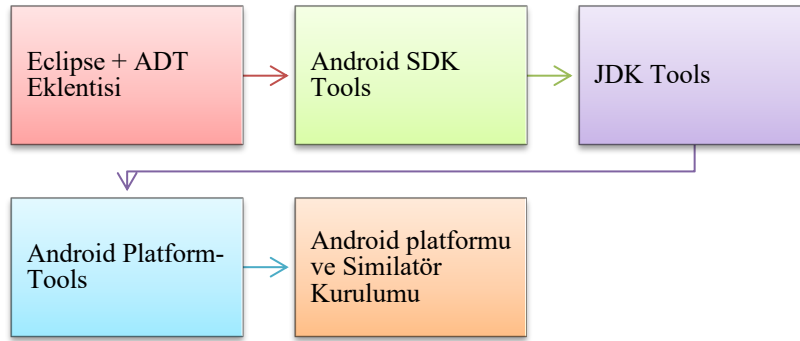
ANDROİD MOBİL UYGULAMA GELİŞTİRME

Android işletim sistemi Google tarafından geliştirilen, kaynak kodları açık olan ve günümüzün tablet bilgisayarlarında en yaygın kullanıma sahip işletim sistemidir (Arslan, Gülnar, 2014). Android işletim sistemi için mobil uygulama geliştirirken Android 2020 resmi sitesindeki verilere göre; yüksek kaliteli uygulamalar sunabilmek için uygulamaların çeşitli cihazlar üzerinde test edilmesi gerekir. Lansman öncesinde, düzen sorunlarını belirleyebilen, kilitlenme teşhisi koyabilen, güvenlik açıklarını bulabilen ve daha birçok işlemi gerçekleştirebilen otomatik testler gerçek cihazlar üzerinde çalıştırılır. Ardından, kalite ile ilgili sorunları daha derinlemesine araştırmak için Firebase Test Lab'de özel testler uygulanır. Aşağıda Android uygulama geliştiricilerinin algoritmik olarak izlemesi gereken adımlar verilmiştir.

- Hangi yüzeylerin diğerlerinin önünde olduğunu belirtmek için gölgeler kullanılmalıdır. Bu işlem, görüntüde hiyerarşi oluşturmaya yardımcı olur.
- Materyal tasarımı basamakları kullanılarak görüntüler, renkler, metrik ve ana çizgiler animasyonlu geçişler yapıldıktan sonra tasarım tamamlanır.
- Gösterilen etkileşimli yüzeyler giriş sırasında kullanıcının parmağıyla buluşacak şekilde dokununca yüklenmeli ve dokunulan yüzeylerin butonlarının çalışmasına dikkat edilmelidir.

Bu adımlar izlendiğinde uygulama görsel olarak etkili olacağı için GooglePlay'e hızlı bir şekilde eklenir ayrıca ekonomik gelir getirme süreci diğer uygulamalara göre daha hızlı olur. Android, 2020'ye göre uygulama geliştirmek isteyen bir kişinin kurması gereken eklentiler ve araçlar vardır. Bu eklentiler ve araçlar Şekil 3'te verilmiştir.

Şekil 3. Android Uygulama Geliştirmek için Gerekli Program ve Eklentiler



Kaynak: (URL 7)

Şekil 3'te de görüldüğü gibi Android uygulama geliştirmek isteyen geliştirici, öncelik sırasıyla Android Development Tools (ADT) eklentisini, The Android Software Development Kit (SDK) Tools'u, The Java Development Kit (JDK) Tools, Android Geliştirme Ortamı Tools'unu ve uygulama geliştirmek istediği Android platformunu bilgisayarına kurması gereklidir.

JAVA

Java, programlama dili ile Android platformu üzerinde mobil uygulama hazırlamak için araçlar sunan bir geliştirme aracıdır (Sonuç, Ortakçı ve Elen, 2013). Android işletim sistemi, Java ile yazıldığı için Android uygulamalarda çok tercih edilen bir dildir. Java programlama dilinin en önemli özelliklerinden biri de çoklu işlem (multithread) mimarisini destekliyor olmasıdır (Arslan ve Gülnar, 2014). Bu mimari, bir bilgisayar programının komutlarının çalışmasını kontrol eder ve iyileştirilmesini sağlar.

HTML5

HTML5, HTML4.1'den sonra geliştirilen web sayfalarındaki karmaşıklığı azaltmaya yardımcı olan ve web sayfa kodlarındaki karmaşıklığı azaltan yeni web elemanları kullanımını sağlayan bir teknolojidir (Durgut ve Çakır, 2013). Mobil cihazlar üzerinde çalışan web uygulamalarını geliştirmek için HTML5 kullanılabilir. Bu uygulamaların çalışması; web sunucusu ile etkileşim ile veya sadece cihazda olabilir.

Android ve iOS işletim sistemleri için uygulama geliştirme aşaması ile ilgili bilgiler Tablo 1'de verilmiştir.

Tablo 1. Android ve iOS'un Uygulama Geliştirme Açısından Karşılaştırılması

| Özellik | ANDROID | iOS |
|--|--|----------------------|
| İşletim Sistemi Tabanı | Linux | Linux |
| İşletim Sistemi | Açık Kaynak Kodlu | Kapalı Kaynak Kodlu |
| Hangi Programlama Dili Kullanılabilir | Java, Objective C, Swift, Html 5, Css3, JavaScript | Swift ve Objective C |
| GUI (Graphical User Interface) | XML | Xcode |
| IDE (Integrated Development Environment) | Eclipse 3.5 | Xcode |

Tablo 1'de Android ve iOS işletim sistemleri için mobil uygulama geliştirmede etkili olan başlıklar göz önünde bulundurularak karşılaştırma yapılmıştır. İşletim sistemlerinin kapalı sistem ve açık kaynak kodlu olmasının avantaj ve dezavantajları vardır. iOS kapalı bir sistem olduğundan dış saldırılara karşı daha güvenlidir, fakat bu durum dışarıdan müdahaleleri zorlaştırmaktadır. iOS tabanlı bir mobil cihaza bluetoothla veya USB ile video, mp3 ve belgelerin yüklenmesi zordur. Android işletim sistemi ise açık kaynak kodlu olduğundan dış saldırılara karşı iOS işletim sistemi kadar güvenli değildir. Ancak USB ve bluetoothla mp3, video ve belgelerin mobil cihaza yüklenmesi iOS işletim sistemi kadar zor değildir. Benenson, Gassmann ve Reinfelder (2013) tarafından 700 öğrenci üzerinde yapılan araştırmada, teknik özellikleri ön planda tutan kullanıcıların daha çok Android telefonları tercih ettiği sonucuna ulaşılmıştır. Araştırmada, telefonlarında virüs koruma programı yüklü olan kullanıcıların Android telefon kullanma oranının fazla olduğu sonucuna da ulaşılmıştır. Gizlilik konusunda daha bilinçli kullanıcıların iOS işletim sistemini kullanmayı tercih ettiği görülmüştür. Virüs içeren uygulamaların Apple'ın inceleme sürecinden geçemeyeceği düşünülmektedir. Her uygulama geliştirme platformunun üstünlükleri ve sınırlılıkları vardır. Amaçlara, bütçeye ve sahip olunan teknoloji altyapısına göre bir tercih yapılır. Hangi platform gereksinime ve teknolojik alt yapıya uygunsu o platform tercih edilmelidir (Keskin ve Kılınc, 2015).

MOBİL PLATFORMLAR

Yerel (Native) Mobil Platformlar

Native mobil uygulama, ilgili cihaza özgü dille yazılan -iOS için Objective-C, Android için Java ve sadece ilgili cihazda çalışacak şekilde derlenen mobil uygulamadır. Belirli bir platform için yazıldığından, işletim sistemi özellikleriyle ve bu platformda genellikle yüklü olan diğer yazılımlarla etkileşimde bulunabilir ve onlardan yararlanabilir. Yerel (Native) uygulamalar, cihazın tüm fonksiyonlarına erişebilirler. Yerel bileşenler çalışma anında dinamik olarak yüklenen, paylaşılan kütüphanelerdir (Yan ve Yin, 2012). Hızlı açılıp, hızlı kapanır. İnternetli veya internetsiz çalışabilirler. Facebook ve Twitter native uygulamalara örnektir.

Melez (Hybrid) Mobil Platformlar

Bir mobil platform için yazılan kodun başka bir platformda kullanılmaması yerel uygulamaların bakımı ve geliştirilmesi aşamasında mobil geliştiriciler için zorluk oluşturmaktadır. Hybrid, geliştiricilerin web tabanlı bir mobil uygulama oluşturmaya izin vermektedir (Georgiev, Jana ve Shmatikov, 2014). Yerel uygulama mantığında yüklenmesine rağmen içinde bir web uygulaması açılır. Hybrid tipi uygulamaların en önemli artışı Android ve iOS işletim sistemlerine aynı anda uyum sağlayabilmesidir. Tek kod ile bütün platformlara uygulama geliştirilebilir.

Çapraz (Cross) Mobil Platformlar

Çapraz platform uygulaması, farklı mobil platform üzerinde çalışan uygulamalardır (Perchat, Desertot, ve Lecomte, 2014). Mobil cihazlarda çeşitlilik arttıkça sorunsuz çalışan mobil uygulamalara ihtiyaç da artar. Çapraz platformlar bu çeşitlilik için en uygun çözümdür. Farklı platformlar için birden çok uygulama geliştirmek; kaynak, zaman ve çaba açısından uygulama geliştiriciler için dezavantaj oluşturduğundan çapraz platformlar tercih sebebi olmuştur. Uygulama performansları bakımından hibrit platformların çok üzerindedir (Keskin ve Kılınç, 2015).

Çapraz platformların faydaları:

- Yazılan kod tekrar kullanılabilirliği için dağıtım ve bakımı kolaydır.
- Her platform için tekrar kod yazılması gerekmediğinden hız kazandırır.
- Tek bir kod tabanını birden fazla platformda çalıştırması nedeniyle ekonomiktir.
- İşletmeler için farklı platformlardan erişim sağlanabilir olması nedeniyle kullanıcı kitlesinin geniş olmasını sağlar.

Duyarlı (Responsive) Mobil Platformlar

Web tarayıcıların destelediği kod yapısı ve çalışma biçimi farklı olduğu için bir web tarayıcısı üzerinde test edilen web uygulaması, test edildiği web tarayıcısında sorunsuz çalışabilir fakat farklı tarayıcılarda görünümde bozulmalar olabilir (Oyucu ve Polat, 2018). Bu soruna çözüm olarak duyarlı tasarım (Responsive Design) yaklaşımı geliştirilmiştir. Duyarlı tasarım; bir web sitesinin, her tür cihazın ekran çözünürlüğüne göre yeniden şekillendirilerek cihaza uyumlu görüntülenmesini sağlamaktadır. Duyarlı tasarım (Responsive Design), teknolojik gelişmeler sonucu ortaya çıkan farklı ekran ölçülerine uyum sağlayan farklı mobil cihazlara uyum sağlayabilen esnek bir web tasarım anlayışıdır. (Çatal ve Kürşad, 2015). Duyarlı tasarım tekniğiyle oluşturulan web sitesi, görüntülediği cihaza karşı hassasiyet gösterir ve o cihaz için tasarlanmış gibi çalışır

Duyarlı (Responsive) platformların avantajları:

- Responsive tasarımları daha iyi kullanıcı deneyimine sahip oldukları için arama motorunda üst sıralarda çıkarlar.
- İyi hazırlanmış mobil uyumlu web sitelerinde çok istisnai durumlar hariç mobil siteye ihtiyaç duyulmaz.
- Mobil uyumlu web siteler, yani duyarlı web tasarım, cihazın boyutlarına göre duyarlı olup şekil değiştirerek adapte olan tasarımlar olduğu için aynı URL tek site olacaktır. Tek site olması yönetimi ve güncelleme işlemlerini tek yönetim paneli üzerinden yapmaya olanak sağladığı için zamandan ve maliyetten ciddi tasarruf sağlar.

YAYIMLAMA AŞAMALARI

Geliştirilen mobil uygulamalar Google uygulama marketlerinde (Google Play) ve Apple uygulama marketinde (App Store) yayımlanabilir. Domínguez-Álvarez ve Gorla (2019)'in her iki platformdaki en popüler uygulamaların yayımlanma sıklığını karşılaştırdıkları araştırmada, uygulama geliştiricilerin

Android uygulamaları, iOS uygulamalardan daha sık yayımlama eğiliminde olduğu sonucuna varılmıştır.

Google Uygulama Marketlerinde (Google Play) Yayımlama

Google uygulama marketler, Android işletim sistemi tabanlı akıllı telefonlar için yazılımlar barındıran ve bu yazılımların dağıtımını için kullanılan mekanizmadır (Büyüköze, 2019). Uygulamayı yayımlamak için gerekli olan bilgiler aşağıda verilmiştir.

- Uygulamaya ait APK dosyasının bilgisayara indirilmesi gerekir.
- Geliştirici hesabı oluşturulmalıdır. Bunun için Gmail adresiyle kayıt olunup yükleme adımları izlenmelidir.
- Ödeme bir kereye mahsus 25 dolardır.
- Uygulama Google yetkilileri tarafından incelenip kabul edilirse, e-posta yoluyla kabul bilgisi iletilir ve uygulama Google Play’de yerini alır.

Google uygulama marketlerinden de mobil uygulamasını yayımlamak isteyen bir uygulama geliştiricisinin yukarıda belirtilen basamakları inceleyerek ve bu aşamalara dikkat ederek uygulamasını hem daha hızlı bir şekilde yayımlanmasına fayda sağlayacak hem de ekonomik gelir elde etmek için kazanç sağlayacaktır. Mobil uygulama geliştiricisinin, uygulamasının ne kadar kısa zamanda yayımlanması o denli ekonomik gelir elde etme süreci için de fayda sağlayacaktır.

Apple Uygulama Marketlerinde (Appstore) Yayımlama

AppStore, Apple’ın sahibi olduğu ve yönettiği çevrimiçi uygulama mağazasıdır. Geliştiricilerin meydana getirmiş olduğu ücretli ve ücretsiz yazılımları içinde barındırır. AppStore’da uygulama yayımlamak için gerekli olan bilgiler aşağıda verilmiştir.

- Uygulamanın IPA adlı dosyası elde edilmelidir.
- Uygulamayı AppStore’da dağıtmak için bir Apple Developer hesabına abone olunması gerekir.
- Apple Geliştirici Programı yıllık ücreti 99 USD’dir.
- Uygulama geliştirici yayımlama aşamasına gelmeden önce uygulamasını özenle test etmelidir.

Bunlarla birlikte uygulamanın yayımlanmasının kabul edilmesi için net bir şekilde açıklanması, verilen bilgiler ve sunulan hizmetin birbirini desteklemesi, geniş bir kitleye hitap etmesi, fayda sağlayan bir uygulama olması, uygulamanın tamamlanmış tüm özellikleri kullanılabilir halde olması gereklidir. Ayrıca uygulama bir süre sonra amaçlandığı gibi çalışmıyor veya artık aktif olarak desteklenmiyorsa AppStore’den kaldırılır. Android ve iOS işletim sistemleri için uygulama yayımlama aşamasında platformların özellikleri Tablo 2’de verilmiştir.

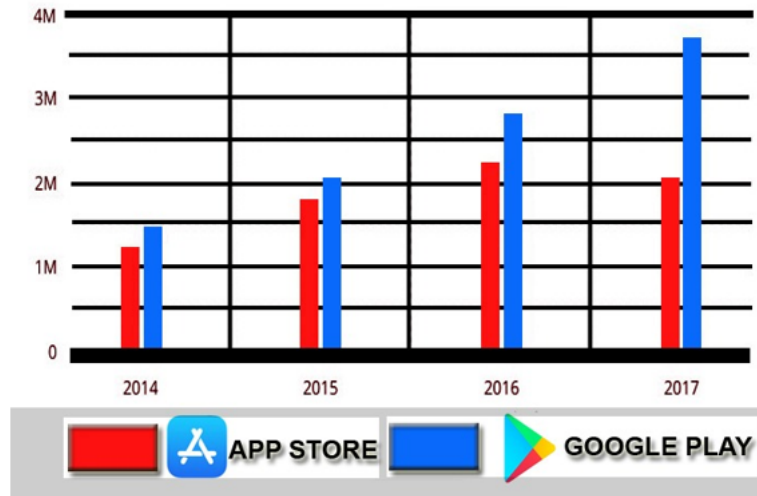
Tablo 2.Android ve iOS’un Uygulama Yayımlama Açısından Karşılaştırılması

| İnceleme Başlık Adı | ANDROID | iOS |
|--------------------------------|---|---|
| Yayımlama İzinleri | Yumuşak | Katı |
| Yayımlama Aşaması (\$) | 25(Bir kereye mahsus) | 99(yıllık) |
| 2017 Uygulama Sayıları | 3,6 Milyon | 2,1 Milyon |
| Uygulama Mağazaları | Google Play | APP Store |
| Uygulama Mağazaları Web Adresi | https://play.google.com/ | https://apps.apple.com/ |

Tablo 2’de uygulamaların yayımlanma aşamasıyla ilgili özelliklerinin iOS ve Android platformlarında karşılaştırılması yapılmıştır. Google Play’de bir uygulama yayımlanmak istendiğinde uygulama gönderildikten birkaç saat sonra mağazada yerini alır. Google Play var olan uygulamanın kontrolünü bir dizi şikâyet gelmediği sürece yapmaz. Buna karşılık iOS’da yayımlanması istenen uygulamalar için inceleme süreci daha katıdır (Chen, Xu, Zhou ve Zhu, 2013). AppStore inceleme yönergelerine göre içerik incelenerek uygun olmayan kısımlar uygun hale getirildikten sonra kullanıcılara sunulur. Ayrıca, Android uygulama için bir kereye mahsus 25\$ ödeme yapılırken, iOS uygulama geliştiricilerinin yıllık 99\$ ödeme yapmaları gerekir. Aynı zamanda, yazılan uygulamanın daha hızlı ve kolay Google Play ve App Store’a eklenmesi için belirtilen marketlerde de uygulama sayısının dikkate alınması yararlı olacaktır.

AppStore ve GooglePlay’de bulunan uygulama sayılarının 2014 – 2017 yılları arasında büyüme durumu Tablo 3’te verilmiştir.

Tablo3.Uygulama Mağazaların Yıllara Göre Büyümesi

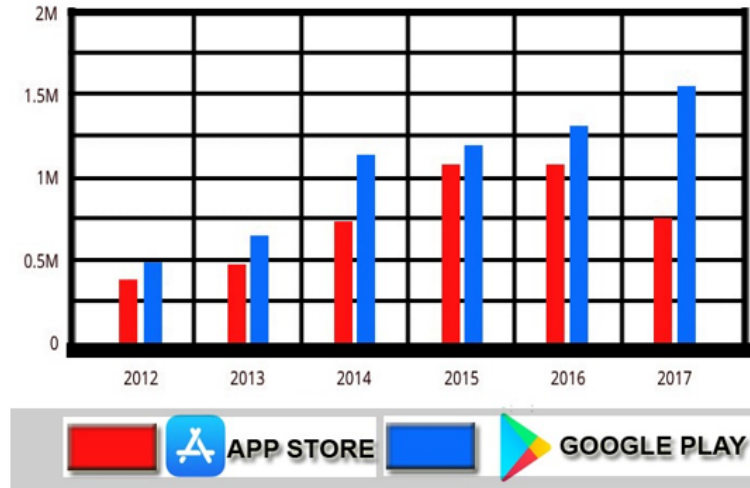


Kaynak: (URL 8)

Tablo 3’te yıllara göre uygulama mağazalarının (App Store ve Google Play) büyüme durumları verilmiştir. Grafik incelendiğinde; 2014 yılından 2017 yılına kadar Google Play’deki uygulamaların App Store’a göre daha fazla olduğu görülmektedir. 2014 yılından 2016 yılına kadar App Store ve Google Play’deki uygulamalar birbirine yakinken 2017 yılında uygulama sayısı bakımından Google Play’in App Store’a göre artışının daha fazla olduğu muhtemel sonuç olarak görülmektedir. Uygulama geliştirmek isteyen birinin marketlerde ki uygulama sayılarına ve yıllara göre artışına dikkat etmesi, uygulamanın yayımlanmasını daha hızlı ve kolay hale getirir. Hatta herhangi bir konuda uygulama yazmak istediğinizde Google Play ve App Store’de ki olduğunuz tarihten geçmişe dönük var olan uygulamaları incelemeniz diğer uygulama geliştiricilerin önüne geçmenizde fayda sağlayabilir.

2012-2017 yılları arasında AppStore ve GooglePlay e eklenen yeni uygulamaların sayısı Tablo 4 ‘te verilmiştir.

Tablo 4.Yıllara Göre Yeni Uygulamalar



Kaynak: (URL 9)

Tablo 4'teki grafik incelendiğinde; 2012 yılından 2017 yılına kadar Google Play'in yeni uygulama sayısı App Store'a göre daha fazladır. Ayrıca 2017 yılında App Store ve Google Play'deki yeni uygulama sayıları incelendiğinde, diğer yıllara göre aralarında ki farkın en fazlaya çıktığı görülmektedir. 2017 yılında Google Play uygulamaları 1,5 milyondan fazla olurken, App Store'daki uygulama sayısı 750 binlerde kalmıştır. Neredeyse 2017 yılında yeni uygulama sayıları bakımından Google Play, App Store'a göre 2 kat artış göstermiştir.

EKONOMİK GELİR SÜRECİ

Mobil hizmetler, insanların hayatına kolaylıklar sağladığı ve kullanımı yaygınlaştığı için birçok alanda mobil uygulama geliştirilmektedir. Özellikle son yıllarda kullanıcıların banka işlemlerinden alışverişe kadar birçok işlerini internet üzerinden yaptığı bir mobil çağ dönemine girilmiştir. Mobil uygulamaların çok kullanıldığı bu dönemde, bu yoldan gelir elde etmek de oldukça yaygındır. Özellikle, insanların hayatına kolaylık sağlayan, kullanımı kolay ve daha önceden yapılmamış bir mobil uygulama yaparak daha çok dikkat çekmek ve para kazanmak mümkündür. Hazırlanan mobil uygulama uygun platformun uygulama mağazasında yayımlanarak, dünyanın başka bir noktasındaki kullanıcının kullanması sağlandığında gelir elde etmek de mümkündür (Ertemel ve Çudin, 2018). Bu durumda küçük ölçekli girişimler ve bireysel yazılımcılar bile mobil uygulamalardan gelir elde edebilir.

Uygulama mağazaları satış ve prestijlerinde yüksek kalite düzeyini korumak ister; düşük kaliteli uygulamalar pazara kabul edilmeyebilir veya servis sağlayıcısı tarafından kaldırılabilir (Gezici, 2018). Bu durumda da düşük kalitedeki bir uygulamadan da düzenli gelir elde etmek mümkün olmaz. Müşterileriyle ilişki kurma yöntemi olarak mobil ortamları seçen firmalar mobil uygulama yoluyla reklam verirler. Mobil pazardaki kazancın büyük bir bölümü, mobil pazarlama yapan bu firmaların verdiği reklamlardan kaynaklanır (Yıldırım, 2010).

Android ve iOS işletim sistemlerinin karşılaştırılması ile ilgili yapılan araştırmada, uygulama marketlerindeki aynı uygulamaların fiyat bilgileri ve bu iki platformun özellikleri ile ilgili bilgiler tablolarda verilmiştir. GooglePlay ve AppStore uygulama marketinde de bulunan bazı uygulamaların fiyat bilgileri Tablo 5'te verilmiştir.

Tablo 5.Google Play ve App Store Uygulama Fiyat Karşılaştırması

| Uygulama / Oyun Adı | ANDROID | iOS |
|---------------------------|---------|---------|
| Minecraft | 24,99 ₺ | 49,99 ₺ |
| Stardew Valley | 54,99 ₺ | 54,99 ₺ |
| Monument Valley | 10,99 ₺ | 26,99 ₺ |
| Riptide GP2 | 7,99 ₺ | 19,99 ₺ |
| A Dance of Fire and Ice | 10,99 ₺ | 12,99 ₺ |
| RFS Real Flight Simulator | 2,99 ₺ | 2,99 ₺ |
| PAW Patrol: Air & Sea | 11,99 ₺ | 19,99 ₺ |
| My City: After School | 11,99 ₺ | 19,99 ₺ |
| Bridge Constructor Portal | 9,99 ₺ | 34,99 ₺ |
| Age of Civilizations II | 15,49 ₺ | 19,99 ₺ |
| Monument Valley | 10,99 ₺ | 26,99 ₺ |

Tablo 5 verileri Mayıs, 2020 tarihinde elde edilmiş olup, bazı simülör uygulamalarının ve oyunların GooglePlay ve AppStore’da fiyat bilgileri karşılaştırılmıştır. Marketlerden alınan bilgilere göre örneklik niteliği taşıyan uygulamaların büyük bir kısmının iOS işletim sistemi için daha pahalı olduğu, geriye kalan kısmının ise iOS ve Android işletim sistemi için aynı ücret olduğu anlaşılmaktadır. Fakat Android marketlerde bir kategoride yer alan benzer uygulamaların iOS market kategorisine göre daha fazla olması, uygulama fiyatlarındaki azalma nedeni olarak gösterilebilir. Bunun neticesinde aynı kategori için yapılan bir uygulamanın iOS’da sayısının az olması uygulama geliştiricileri için avantaj sağlayabilir. Aynı uygulamanın belirlenen iki marketteki fiyatları araştırıldığında iOS işletim sistemi için geliştirilen uygulamaların daha fazla olduğu anlaşılmaktadır. Genel anlamda bütün uygulamalar fiyat açısından incelendiğinde, iOS işletim sistemi için tasarlanan uygulamaların fiyatlarının daha fazla olduğu belirtilebilir. Bunun neticesinde marketlerde satışa sunulan uygulama sayısı ve uygulama kategorilerinde yer alan benzer uygulamalar göz önünde bulundurulduğunda, daha fazla kazanç sağlamak isteniliyorsa iOS’da çalışabilecek bir uygulama geliştirmenin daha kazançlı olabileceği söylenebilir. Android ve iOS platformları için geliştirme, yayımlama ve ekonomik gelir sürecine dair bulgular Tablo 6’da verilmiştir.

Tablo 6.Android ve iOS Karşılaştırması

| İnceleme Başlık Adı | ANDROID | iOS |
|---|---|---|
| Kuruluş Tarihi | 2003 | 2007 |
| Kim Tarafından Geliştirildi | Andy Rubin, Rich Miner, Nick Sears ve Chris White, 2005 yılında Google satın aldı | Apple |
| İşletim Sistemi Tabanı | Linux | Linux |
| İşletim Sistemi | Açık Kaynak Kodlu | Kapalı Kaynak Kodlu |
| Yayımlama İzinleri | Yumuşak | Katı |
| Yayımlama Aşaması (\$) | 25(Bir kereye mahsus) | 99(yıllık) |
| Ekonomik Gelir | Reklam ve Uygulama İçi Satın Almalar | Reklam ve Uygulama İçi Satın Almalar |
| 2017 Uygulama Sayıları | 3,6 Milyon | 2,1 Milyon |
| Uygulama Mağazaları | Google Play | APP Store |
| Hangi Programlama Dili Kullanılabilir | Java, Objective C, Swift, Html 5, Css3, JavaScript | Swift ve Objective C |
| GUI (Graphical User Interface) | XML | Xcode |
| IDE (Integrated Development Environment) | Eclipse 3.5 | Xcode |
| Uygulama Mağazaları Web Adresi | https://play.google.com/ | https://apps.apple.com/ |

Tablo 6’da Android işletim sisteminin iOS işletim sisteminden 4 sene önce kurulduğu, iki işletim sisteminin de Linux tabanlı olduğu, Android işletim sisteminin açık kaynak kodlu ve iOS işletim sisteminin kapalı kaynak kodlu olduğu görülmektedir. Yayımlama izinlerine baktığımızda iOS işletim sisteminin yayımlama kurallarının Android işletim sistemine göre daha katı olduğu görülmektedir. Tablo 2’ye göre, Android işletim sisteminde mobil uygulama yayımlama için bir kereye mahsus olarak 25 dolar alınırken, iOS işletim sisteminde yıllık 99 dolar alınmaktadır. Bu duruma istinaden Android işletim sisteminde uygulama yayımlamanın iOS’a göre daha avantajlı olduğu söylenebilir. 2017 yılında marketlerdeki uygulama sayıları, Android işletim sisteminde 3,6 milyon, iOS işletim sisteminde 2,1 milyondur. Mobil uygulama geliştirme için kullanılan programlama dilleri çeşitliliği bakımından Android işletim sisteminin iOS işletim sistemine göre daha zengin olduğu görülmektedir.

SONUÇ

Uygulama sayıları incelendiğinde; Android markette iOS’tan daha fazla uygulama bulunduğu ve bazı uygulamaların iOS markette Android marketten daha yüksek fiyata sahip olduğu görülmektedir. Bununla beraber, araştırmada iOS mobil cihaz kullananların sayısında son yıllarda artış görüldüğü, Android cihaz kullananların sayısında ise azalma görüldüğü sonucuna ulaşılmıştır. Android marketteki uygulama sayısının iOS markete göre fazla olması uygulama satışı yönünden fazla olabileceği gibi yeni uygulama geliştiricilerinin ise uygulama sayısının az olduğu iOS markete yönelmesi iyi sonuçlar almasına faydalı olacaktır. Ayrıca araştırma da aynı uygulamanın iki marketteki uygulama fiyatları

incelendiğinde iOS marketin bu konuda daha pahalı uygulama satışları yaptığı sonucuna ulaşılmıştır. Son yıllarda iOS işletim sistemli mobil cihaz kullanıcıların sayısındaki artış sebebiyle de uygulama geliştiricilerin iOS markete yönelmesi güçlü bir olasılıktır. Bu sebeplerle uygulama geliştiricilerinin iOS platformuna yönelmesi daha avantajlı olabilir.

iOS uygulamalarına yönelik şikayetler, güncelleme sonrası artarken, Android uygulamaların uygulama özelliklerinin fonksiyonel olmamasıyla ilgili daha fazla şikâyet aldığı sonucuna varılmıştır. Uygulama sonrasında kullanıcıların uygulamaları şikayet etmesinde, iOS uygulamalarına yönelik şikayetler fazla olurken uygulama işlevselliği ve kullanılabilirlik arayüzü bakımından ise Android uygulamaları iOS uygulamalarına göre daha fazla şikayet almaktadır. Uygulama geliştiricilerin bunları dikkat ederek uygulamalarında ki eksiklerini tamamlaması ayrıca bu uygulamaların arayüz tasarımlarına özen ve dikkat göstererek sadeleştirilmesi gerekmektedir. Buna istinaden uygulama geliştiricilerin belirtilen hususlara dikkat ederek uygulama geliştirmesi ile fayda sağlanabilir. Ayrıca uygulama fonksiyonelliği bakımından iOS kullanıcılardan Android'e göre daha az şikayet aldığı bilmek, uygulama geliştiricileri için uygulama yapmayı tercih edeceği markete yönelik yönlendirici bir bilgi olacaktır. Uygulama geliştirmek isteyen bir uygulama geliştiricinin iOS marketten daha fazla gelir sağlayabileceğinden bu markete yönelmesi kendisi için yararlı olacaktır.

Kolay kullanılabilir, anlaşılır ve uyum sağlanabilen arayüz, renk uyumu ve farklı tasarım öğeleri kullanılarak daha çekici ve kullanıcı dostu hale getirilebilir. Hamilton ve Brady (2012)'ye göre bazı mobil uygulamaların görsel, işitsel ve materyal tasarımı yönünden profesyonellikleri ölçüldüğünde bu değerlendirmenin düşük olmasına bakılarak mobil uygulamaların akademik kaynak olamayacağını savunmaktadırlar. Hamilton ve Brady (2012)'a göre akademik kaynak olarak değerlendirilmeyen bazı uygulamalar arayüzü geliştirilerek görsel işitsel ve materyal tasarım basamaklarına da dikkat edilerek tekrar yazılabilir. Ayrıca uygulama geliştiricisi, uygulamasını küçük ya da büyük gruplara test ettirebilir. Bu sayede uygulamanın eksik yönleri tespit edilerek bu eksiklerin giderilmesi sağlanabilir. Uygulama geliştiricilerinin iki markete de materyal tasarım ilkelerine uygulama tasarımlarında dikkat etmesi gerekmektedir. Yazılan uygulamanın görsel ve işitsel arayüz sadeliği ne kadar iyi olursa marketlerde kullanıcılardan olumlu yönde geribildirimlerde o kadar yüksek olacak ve uygulama geliştiricilerin de hızla ekonomik gelir elde etmesi muhtemel olacaktır. Uygulama geliştiricisinin görsel ve işitsel olarak dikkat çekici ve materyal tasarım ilkelerini de dikkate alarak uygulamasını hazırlaması, materyal tasarım ilkelerine dikkat etmeden tasarlanan uygulamaların önünde olacağı gibi yazdığı uygulamada da sistemli olarak geliştirilebilecektir. Uygulama marketlerinde görsel, işitsel ve materyal tasarımı yönünden iyi tasarlanarak geliştirilen uygulamalar benzer amaçlı geliştirilen uygulamaların önüne geçecektir.

Bir uygulama geliştiricisinin hazırladığı uygulamayı platform bakımından süreklilik arz edecek bir dil seçerek geliştirmesi, geliştirici için yararlı olacaktır. Android işletim sisteminde programlama dili olarak Java, Objective C, Swift, Html 5, Css3, JavaScript kullanılırken, iOS'ta Swift ve Objective C kullanılmaktadır. Uygulama dili bakımından Android işletim sisteminde daha fazla dil kullanılırken iOS işletim sistemi için bu sayı azdır. Uygulama dil sayısının artması uygulama geliştiricinin sayısının hedef kitlesinin artmasıyla doğru orantılı olduğundan Android markete uygulama geliştiren uygulama geliştiricilerin daha fazla olacağı düşünülmektedir. Uygulama dil çeşitliliğinin artması uygulama geliştiricilerine fayda getireceğinden çeşitliliğin olması o derece de önemlidir. Ayrıca Android ve iOS'da ortak olarak Objective C ve Swift dillerinin kullanılması, bu iki dili bilen uygulama geliştiricilerinin her iki markete de uygulama yazmasına kolaylık sağlayacaktır.

Yayımlama aşamasında Android'e bir kereye mahsus alınan 25 dolar yayımlama ücreti, iOS'da ise yayımlama ücreti olarak yıllık 99 dolar olmasından dolayı, uygulama geliştiricilerin tercih sebebi Android olabilir. Ayrıca 2017 yıllarında ki marketlerde ki uygulama sayıları incelendiğinde Android markette uygulama sayısı 3,6 milyon iken iOS markette bu sayı 2,6 milyondur. Android markette uygulama sayılarının fazla olması nedeniyle iOS marketine göre uygulama çeşitliliği daha fazla olduğundan iOS markete uygulama yazmak ve bundan ekonomik gelir elde etmek daha hızlı olacaktır.

Geliştirilen bir uygulama ilk aşamada ücretli olarak kullanıma sunulabilir ve bu sayede gelir elde edilebilir. Uygulama, uygulama mağazasında ücretsiz olarak yayımlanabilir ve kullanıcılar uygulamaya rağbet ettikten sonra ek içerikler için ücret talep edilebilir veya uygulamanın belli bir kısmını ücretsiz sunarak ek özellikler için ücret talep edilip gelir elde edilebilir. Ücretli mobil uygulamaların zaman sınırı

olmadan, özellikleri kısıtlanmış ücretsiz deneme sürümlerinin tanınırlık sağlamada verimli olduğu görülmüştür (Ertemel ve Çudin, 2018). Mobil uygulamaya reklam olarak bu reklam boyutlarına ve reklamda gösterilen materyallere (yazı, görsel, video, animasyonlu görsel) göre farklı kazançlar sağlanmaktadır. Mobil reklamlar sayesinde işletmeler hedefledikleri kitlelere hızlı ve kolay bir şekilde ulaşabilirler (Durucasu ve Aksoy, 2016). Mobil reklamlar dijital reklam yatırımları arasında en büyük artış gösteren reklam alanıdır (Kocatürk, 2018). Bu sebeple işletmeler mobil reklamları tercih ederler. Aynı zamanda gösterim ve tıklama başına da kazanç oranları değişmektedir. Bir başka gelir elde etme yolu ise uygulama iyi bir indirme ve etkin kullanım sayısına ulaştığında yatırımcıların uygulamayı satın almak istemeleri ya da yazılım konusunda kendi projelerine destek istemeleri şeklindedir.

Mobil uygulama geliştiricilerin geliştirdiği uygulamadan ekonomik gelir elde etmesi ve geliştirdiği uygulamanın daha hızlı yayınlanması için sırasıyla; geliştireceği uygulamaya başlamadan önce marketleri gözden geçirip aklındaki uygulamanın olup olmadığına bakması, aklındaki uygulamanın daha önce yazılıp yazılmadığına bakması gerekmektedir. Ayrıca yazacağı uygulamanın marketler de hangi kategoride yayınlanacağını ve o kategoride toplam uygulama sayılarını dikkate alması gerekmektedir. Uygulama geliştiricilerin Android'e yönelik mi yoksa iOS'a yönelik mi uygulama geliştireceğine karar vermesi gerekmektedir. Uygulama arayüzü geliştirirken materyal tasarım basamaklarını da dikkate alarak sade bir model oluşturması gerekmektedir. Oluşturulan bu modelin hedef kitledeki kullanıcılara test ettirilerek geri yapılan dönütlere göre düzenlemeler yapılması gerekmektedir. Uygulama geliştiricisinin geliştireceği uygulama için marketlerde yayınlanma ve ekonomik gelir elde etme süreç basamaklarını göz önünde bulundurması, hangi kategoride uygulama geliştireceğine ve hangi markete yönelik uygulamayı geliştireceğine karar vermesine yardımcı olacaktır.

İşletim sistemi ve cihaz çeşitliliğinden dolayı uygulama geliştiricileri farklı uygulama geliştirme ortamları ve farklı programlama dilleri kullanmak durumunda kalmışlardır. Uygulamanın güncellenmesi ve geliştirilmesi aşamalarında farklı platformlarda değişiklik yapmak çok büyük sorun olduğu için bazı uygulamaların sadece iOS veya Android sürümü geliştirilmiştir (Canbazoglu ve Koçer, 2016). Mobil yazılım geliştirme ortamlarının ve araçlarının birbirlerine göre farklı avantajları ve dezavantajları olduğu görülmektedir. Mobil uygulama geliştirme aracı seçim aşamasında; tanımlanan kriterlere uygun olanlar arasında makul görünen seçenekleri küçük pilot uygulamalarda deneyimleyip seçim yapılabilir (Tunali ve Zafer, 2015).

Farklı platformlarda yayımlanmak üzere bir uygulama geliştirilip, bu uygulama yayımlandıktan sonra belirli zaman kesitinde uygulamanın marketlerden indirilme sayıları, uygulamanın mobil cihazlarında kullanım sürelerinin tutulup kullanıcının uygulamada ne kadar vakit geçirdiği verisi, yapılan uygulamanın Android ve iOS'ta ne kadar yer kapladığı, uygulamanın mobil cihazın ne kadar belleğini (RAM) kullandığı, yapılan uygulamanın kesitsel zaman diliminde ekonomik gelir elde etme süreçleri analiz edilerek platform karşılaştırması yapılabilir. Yazılan mobil uygulamanın bu konuda daha önceden yazılmış mobil uygulamalar ile ne kadar benzerlik gösterdiğine dikkat edilmesi gerekmektedir. Halihazırda yazılan bir uygulamanın geliştiricisinden izin ve telif hakkı alınarak mevcut uygulamaya yeni özellikler katılabilir.

KAYNAKÇA

Akyol, G. (2014). Mobil Uygulama Arayüzlerinin İncelenmesi Üzerine Bir Çalışma. ISITES2014, Karabuk.

Arslan, B., & Gülnar, S. (2014). Java-android yazılım mimarisi: bir masaüstü ile çoklu tablet bilgisayar haberleşme uygulaması, Akademik Bilişim'14 - XVI. Akademik Bilişim Konferansı Bildirileri, Mersin Üniversitesi.

Aslay, F., Yıldız, E., Baran, A., & Akar, F. (2018). Mobil işletim sistemlerinin kullanıcı açısından değerlendirilmesi: android-iOS. S. Salman, R. Karapınar, D. Kavak, A. Kılıçer(Ed.), *Mühendislik Bilimlerinde Güncel Akademik Çalışmalar* (ss 46-62). Cetinje, Montenegro.

Aytekin, A., Ayaz, A., Tüminçin, F., & Bektaş, E. (2019). Mobil cihazları etkileyen zararlı yazılımlar ve korunma yöntemleri. International Social Research and Behavioral Sciences Symposium.

Benenson, Z., Gassmann, F., & Reinfelder, L. (2013). Android and iOS users' differences concerning security and privacy. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems* (pp. 817-822).

Büyükgoze, S. (2019). *Mobil uygulama marketlerinin güvenlik modeli incelemeleri*. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 12(1), 9-18.

Canbazoglu, E., & Koçer, A. (2016). Phonegap ile çoklu mobil ortamlara uygulama geliştirme: Web servis örneği. 18. *Akademik Bilişim Konferansı'nda sunulan bildiri, Aydın, Türkiye*.

Chen, Y., Xu, H., Zhou, Y., & Zhu, S. (2013, May). Is this app safe for children? A comparison study of maturity ratings on Android and iOS applications. In *Proceedings Of The 22nd International Conference On World Wide Web* (pp. 201-212).

Çakır, F. S., & Tüminçin, F. (2017). *Güvenli iletişim açısından akıllı cihaz işletim sistemlerinin avantaj ve dezavantajları*. Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi, 4(12), 203-218.

Çatal, D., & Kürşad, D. (2015). *Duyarlı web tasarımı*. Art-e Sanat Dergisi, 8(15), 95-109.

Domínguez-Álvarez, D., & Gorla, A. (2019, August). Release practices for iOS and Android apps. In *Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics* (pp. 15-18).

Durgut, R., & Çakır, A. (2013). *HTML5'in Sunduğu Yenilikler ve Bir Örnek Uygulama*. Akademik Bilişim 2013 – XV. Akademik Bilişim Konferansı Bildirileri, Akdeniz Üniversitesi, Antalya.

Durucasu, R. R., & Aksoy, N. F. (2016). *Sms reklamları ve mobil bilgi servislerine yönelik tüketici tutumunu belirleyen bir araştırma*. Aksaray Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 8(2), 1-13.

Egele, M., Kruegel, C., Kirde, E., & Vigna, G. (2011, February). PiOS: detecting privacy leaks in iOS applications. *NDSS* (pp. 177-183).

Erol, E., Yeten, F. (2018). *Bilişim toplumu bağlamında mobil uygulama geliştirme platformu olarak Xcode*. Yeni Medya Elektronik Dergisi, 2(3), 160-167.

Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). Programlama dilleri öğretimine bir model önerisi: robot programlama. *Akademik Bilişim*, 11.

Ertemel, A. V., & Çudin, D. (2018). Akıllı telefon kullanıcılarının ücretli uygulama satın alma niyetini etkileyen faktörler. *Istanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 17(33), 153-171.

Evren, F. B. (2016). Grafik arayüzlerin tasarım ve kullanılabilirlik açısından incelenmesi: Android ve iOS. *The Turkish Online Journal of Design, Art and Communication*, 6(4), 400-418.

Georgiev, M., Jana, S., & Shmatikov, V. (2014, February). Breaking and fixing origin-based access control in hybrid web/mobile application frameworks. In *NDSS symposium* (Vol. 2014, p. 1). NIH Public Access.

Gezici, B. (2018). *Mobil Uygulamaların Evriminde Kalitenin Gelişimi* (Master's thesis, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü).

Güngör, Ö. (2017). *IOS tabanlı mobil cihazlar için yapay zeka yöntemi ile görüntü sıkıştırma ve iletim uygulaması* (Master's thesis, İzmir Katip Çelebi Üniversitesi Fen Bilimleri Enstitüsü).

Hamilton, A. D., & Brady, R. W. (2012). Medical professional involvement in smartphone'apps' in dermatology. *British journal of dermatology (1951)*, 167(1), 220-221.

Kabakuş, A. T., Doğru, İ. A., & Çetin, A. (2015). *Android kötüçül yazılım tespit ve koruma sistemleri*. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 31(1), 9-16.

Keskin, N. Ö., & Kılınc, A. G. H. (2015). *Mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırılması ve örnek uygulamalar*. Açıköğretim Uygulamaları ve Araştırmaları Dergisi, 1(3), 68-90.

- Oyucu, S., & Polat, H. (2018). *Duyarlı tasarım ile bir M2M platformunun gerçekleştirilmesi*. Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 6(2), 259-268.
- Özdemir, Ş., Akadal, E., Çelik, S., & Reis, A. (2013). Uygulama marketlerinin eğitim kategorisi altındaki uygulamalarının incelenmesi. Akademik Bilişim, Antalya, Türkiye.
- Özkoçak, Y. (2016). Türkiye'de akıllı telefon kullanıcılarının oyalanma amaçlı tercih ettikleri mobil uygulamalar. Global Media Journal: Turkish Edition, 6(12).
- Perchat, J., Desertot, M., & Lecomte, S. (2014). Common Framework: A hybrid approach to integrate cross-platform components in mobile application. Journal Of Computer Science, 10(11), 2165.
- Rebouças, M., Pinto, G., Ebert, F., Torres, W., Serebrenik, A., & Castor, F. (2016, March). An empirical study on the usage of the swift programming language. In 2016 IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER) (Vol. 1, pp. 634-638). IEEE.
- Sonuç, E., Ortakçı, Y., & Elen, A. (2013). Karabük Üniversitesi Bilgi Sistemi Android Uygulaması. Akademik Bilişim Konferansı. Akdeniz Üniversitesi, Türkiye, 23-25.
- Şenel, S., Şenel, H. C., & Günaydın, S. (2019). *Herkes için mobil öğrenme: Mobil uygulamaların evrensel tasarım ilkelerine göre incelenmesi*. Ankara Üniversitesi Eğitim Bilimleri Fakültesi Özel Eğitim Dergisi, 20(1), 73-92.
- Thurm, S., & Kane, Y. I. (2010). *Your apps are watching you*. The Wall Street Journal, 17(1).
- Tunali, V., & Zafer, S. (2015). Comparison of popular cross-platform mobile application development tools. *Celal Bayar University: Faculty of Technology Department of Software Engineering, Maltepe University: Faculty of Engineering and Natural Sciences Department of Software Engineering*.
- Tufan, M., Baykara, M., Güler, Z., & Avcı, E. (2012). Açık Kaynak Mobil İşletim Sistemi: Android İşletim Sistemi.
- Yan, L. K., & Yin, H. (2012). Droidscape: Seamlessly reconstructing the {OS} and dalvik semantic views for dynamic android malware analysis. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)* (pp. 569-584).
- Yıldırım, M. (2010). Tüketicilerin yaşam biçimlerine göre sms reklamlarına karşı tutumlarının incelenmesi: İstanbul ilinde bir pilot araştırma. Yayınlanmamış Yüksek Lisans Tezi. İstanbul Üniversitesi, SBE, İstanbul.

ELEKTRONİK KAYNAKLAR

- URL 1: Android (2020) Android Studio Yönergeleri, [çevrim-içi: <https://developer.android.com/distribute/best-practices/develop/build-with-android-studio>], Erişim Tarihi:07.05.2020'de alınmıştır.
- URL 2: Apple (2020) App Store İnceleme Yönergeleri, [çevrim-içi: <https://developer.apple.com/app-store/review/guidelines/#introduction>], Erişim Tarihi: 07.05.2020'de alınmıştır.
- URL 3: CIRP (2020). Apple, Google Must Collaborate on Covid-19 App, [çevrim-içi: <https://www.files.constantcontact.com/150f9af2201/22099883-2dfc-4f20-a5a0-13889875ccd6.pdf>], Erişim tarihi: 09.05.2020'de alınmıştır.
- URL 4: Arthur, C. (2011). "Is your private phone number on Facebook? Probably. And so are your friends' ." [çevrim-içi: <https://www.theguardian.com/technology/blog/2010/oct/06/facebook-privacy-phone-numbers-upload>], Erişim tarihi 08.05.2020'nde alınmıştır.
- URL 5: Kocatürk, B. (2018). Reklam ve sponsorluk harcamalarının Türkiye'deki gelişimi, reklam stratejileri, vergi sistemindeki yeri ve önemi, Pamukkale Üniversitesi [çevrim-içi: <http://acikerisim.pau.edu.tr/xmlui/bitstream/handle/11499/2554/Berna%20KOCAT%C3%9CRK.pdf?sequence=1&isAllowed=y>], Erişim tarihi: 09.08.2020.
- URL 6: <https://smartpro.com.tr/mobil-programlama-dilleri/> Erişim Tarihi: 01.07.2020'de alınmıştır.

URL 7: <https://developer.android.com/distribute/best-practices/develop/build-with-android-studio>
Eriřim Tarihi: 07.05.2020'de alınmıřtır.

URL 8 ve URL 9: <https://blog.appfigures.com/ios-developers-ship-less-apps-for-first-time/> Eriřim
Tarihi: 01.07.2020'de alınmıřtır.