

**T.C.
Kocaeli Üniversitesi
Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği Anabilim Dalı**

Mobil Cihazlar için Uygulama Geliştirme

**Abdullah Ali GÜN
Emre ÇETİNDİR
Muhammed Furkan ÇELİK
Mehmet Arda YÜKSEL**

GÜZ DÖNEMİ 1.YARIYIL ÖDEVİ

**DANIŞMAN-ÖĞRETİM GÖREVLİSİ
Dr. Serdar SOLAK**

Kocaeli 2019

Mobil Cihazlar için Uygulama Geliştirme

Application Development For Mobile Devices

Abdullah Ali GÜN, Emre ÇETİNDİR, Muhammed Furkan ÇELİK,
Mehmet Arda YÜKSEL

aagun16@gmail.com , emrecetindemir41@outlook.com, --, m.arda.yuksel@gmail.com

Bilişim Sistemleri Mühendisliği

Kocaeli Üniversitesi

ÖZET

Mobil uygulamaların geliştirilme basamaklarından bahsedilmiştir. Bunlar fikir üretmek, araştırma yapma, kesinleştirme, tasarım, yazılım ve test aşamalarından oluşur. Bu uygulamaların geliştirilmesi için gerekli yazılım kitleleri(ortamları) hakkında bilgiler verilmiştir. Android Studio, Xcode, Eclipse, Netbeans vb. gibi geliştirme ortamları ve Xamarin, Ionic, React Native, Cordova vb. gibi geliştirme kitlelerinden detaylıca bahsedilmiştir. Daha sonra bu geliştirme ortamlarında ve geliştirme kitlelerinde kullanabildiğimiz programlama Dillerinden bahsedilmiştir: Java, Kotlin, Objective-C, Swift bu dillere örnektir. Araştırmalar sonucu bulunan çeşitli istatistiksel bulgulara dayanarak mobil uygulamalar konusunda trendlerin hangi yönde ilerleyebileceği açıklandı.

Anahtar kelimeler: Mobil Uygulama Geliştirme, Android,

iOS, Xamarin, React, Swift, Apache Cordova, Ionic

ABSTRACT

The development stages of mobile applications are mentioned. These include idea generation, research, finalization, design, software and testing. Information about the software kits (environments) required for these applications are required. Android Studio, Xcode, Eclipse, Netbeans etc. development environments such as Xamarin, Ionic, React Native, Cordova etc. development kits such as. Then, the programming languages that we can use in these development environments and development kits are mentioned: Java, Kotlin, Objective-C, Swift are examples of these languages.

Based on the various statistical findings found as a result of the research, it was announced in which direction the trends in mobile applications could proceed.

Keywords: Mobile App Development, Android,

iOS, Xamarin, React, Swift, Apache Cordova, Ionic

1.GİRİŞ

Mobil uygulama sektörünün popülerleşmesi ve büyümesi ile oluşan mobil uygulama ve teknoloji ihtiyacı gün geçtikçe artmaktadır ve çeşitlenmektedir. Bu uygulamaların tasarımını ve özellikle yazılımını oluşturan uygulama geliştiricileri için hangi geliştirme ortamı, yöntemi ve programlama dilleri ile çalışma noktasında soru işaretleri oluşmaktadır. Yeni çıkan ortamların kullanılması veya hangi ortamların ne amaçla çıktıkları da yine büyük bir sorun oluşturmaktadır. Bu çalışma da tam olarak bu noktada yazılım geliştiricilerine temel seviyede yol göstermek, en azından bir fikir edinmelerini sağlamak amacıyla yazılmıştır. Bu yazıda birçok geliştirme ortamının ve programlama dillerinin tam olarak hangi amaç doğrultusunda kimler için kullanılabileceği açıklanmıştır. Sonrasında ise oluşan uygulamaların son yıllarda hangi kategorilerde ağırlıkla yer aldığını ve hangi yönde ilerlediğini tespit edip gelecekteki popüler uygulama türlerini ve barındırdıkları teknolojilerini tahmin etmeye çalıştık. Daha sonra bu uygulamaların konularını araştırma sonucunda bulduğumuz istatistik kaynaklarına göre tarif ettik.

Sonuç olarak bu araştırma kapsamında mobil uygulama geliştiricilerinin ne olduğundan, nasıl yapılabileceğinden ve gerekli ortamların ne olduğundan bahsedilmiştir. Ardından bu araştırma kapsamında gelecekte yeni nesil mobil cihazlar ile birlikte uygulama kategorileri tespit edilmiştir.[4]

2.UYGULAMA GELİŞTİRME AŞAMALARI

2.1. FİKİR ÜRETMEK

Bir uygulama geliştirmek için en önemli birincil aşamadır. Neticede her başarılı uygulama bir FİKİR ile ortaya çıkmıştır. Geliştirilmesi planlanan uygulamanın bir sorunu çözmesi veya bir ihtiyacı karşılaması gerekir. Fikrin uygulanabilir ve özgün olması da önemli bir başka kriterdir. Bu aşamada

FREEWRITING adı verilen yöntemin kullanılması tavsiye edilir.

1.1.1. FREEWRITING

Uygulama ile alakalı düşünülen, akla gelen her şeyin bir kâğıda dökülmesi ve daha sonrasında toplanmasıdır. Burada "her şey" derken tam anlamıyla her şey olmalıdır, çünkü bu yöntem uygulama geliştirirken tahmin edilemeyen pek çok duruma hazırlıklı olmayı sağlar. Bir başka deyişle bütün bir süreç için, geliştirmek istediğimiz uygulamanın kafamızda belirginleşmesine yarar.[1][7][14]

2.2. ARAŞTIRMA

Bir fikir oluşturulduğunda, uygulamanın başından geçen tüm süreci planlamak gerekir. Uygulama geliştirme sürecinde harcanan vakit ve para kaybını en aza indirmek için daha önce yapılan hatalardan ders çıkarmak gerekir. Bu hataların hepsini tek başına yapmaktansa daha önce yapılan hatalara bakmak çok işe yarar. Başlamak için en iyi yerlerden biri rekabeti belirlemektir. Başka bir uygulamanın da benzer bir amaca hizmet edip etmediğine bakmalı ve aşağıdakiler aranmalıdır:

Yükleme sayısı: Bu uygulamaları kullanan biri var mı,

Derecelendirmeler ve incelemeler: İnsanların bu uygulamaları sevip sevmediğini ve neleri sevdiklerini / beğenmedikleri,

Uygulama geçmişi: Bu uygulamaların zaman içinde nasıl değiştiğini ve yol boyunca ne tür zorluklarla karşılaştıkları, Kullanıcı tabanını büyütmek için neler yaptıklarını,

İncelemek gerekir. [1][7][14][18]

2.3. KESİNLEŞTİRMEK

2.3.1. Uygulama Hedef Kitle

Bir uygulama geliştirirken hedef kitlenin kimler olduğunun bilinmesi gerekir. Bu aşama, uygulamadan özellikle maddi bir beklenti varsa daha da önem arz eder. (Küçük bir çocuk ile yetişkin bir insanın mevcut uygulamaya yatırabilecekleri para miktarı çok farklıdır.) Uygulama ile sunulan faydaların, hizmetlerin, ürünlerin ihtiyaçları nasıl ve ne ölçüde karşıladığını göz önünde bulundurmak çok önemlidir. uygulama, beklentileri ne kadar iyi karşılırsa, hedef kitleyi yakalamak da o kadar kolaylaşır.

Örneğin bir bankanın mobil uygulamasında; Açıkça uygulamanın en çok hitap ettiği kesim, kendi müşterileridir. Bir başka örnek de mobil oyunlarda, nerdeyse 7 den 70 e pek bir sınır yoktur.

2.3.2. Mobil Platformlar ve Desteklenecek Cihazlar

Mobil platformlar ve cihazlar; donanımlarının performansı bataryalarının durumu, dayanıklılıklarını ve bağlanabilir çevre birimlerini de göz önünde bulundurarak seçilmelidir. Bunların dışında kapsama alanı, cihaz desteği ve diğer özellikleri de göz önünde bulundurmak gerekir. Uygulamanın hangi platformlarda ve ne tür cihazlarda çalışacağını belirleyerek

uygulamanın geliştirilme sürecinde ulaşılabilecek sınırların belirlenmesi sağlanır.

2.3.3. Gelir Modeli

Uygulamadan kazanç sağlamanın çeşitli yöntemleri vardır. Bu yöntemlerden hangisinin geliştirilen uygulama için en iyisi olduğunu belirlemek için pazarda bulunan alternatif uygulamalarda hangi gelir modellerinin kullanıldığı incelenmelidir. Ayrıca, uygulamayı pazara sürdükten sonra hangi aşamada para kazanmaya başlanılacağı da düşünülmesi gerekir.

Uygulamadan para kazanma yöntemleri:

- Reklamlar: Reklamlar, mobil uygulamalarda en çok tercih edilen gelir modelidir. Reklamlar tam ekran veya banner şeklinde olabilir. Ayrıca tam ekran reklamlar ise geçilebilen ve geçilemeyen reklamlar olmak üzere ikiye ayrılır. İzleme, tıklama ve etkileşim başına gelir elde edilir.
- Ücretli indirme: En az tercih edilen gelir modelidir. Yüksek rekabet sebebiyle uygulamanın tanıtımı iyi bir şekilde yapılmazsa ve kullanıcılar ikna edilemezse bu yöntem bir getiri sağlamaz. Ayrıca uygulamanın korsan olarak dağıtılması en büyük sorundur. Bu yöntemi sadece korsan korumalı platformlarda kullanılmaya özen gösterilmelidir. (Android üzerinde APK sorunu varken, iPhone, iPad gibi iOS işletim sistemli cihazlarda böyle bir sorunun olmaması.)
- Uygulama içi satın alma: Ücretli indirme yerine tercih edilebilen bir gelir modelidir. Bu gelir modeliyle uygulama ücretsiz sunulur ancak uygulamanın içerisinde, kullanıcının cazip teklifler ile para ödemesi sağlanır.

Burada dikkat edilmesi gereken husus ise kullanıcıya sunulan ücretli içeriğin gerçek anlamda etkileyici olması gerektiğidir. Bu gelir modeline daha çok oyunlarda karşılaşılr. Üyelik teklifleri de bu kategoridendir.

Netflix abonelikleri, Youtube Premium vb. sistemler bunlara örnektir[1][2]

2.4. UYGULAMAYI TASARLAMAK

Tasarlama aşamasında uygulamaya görsel ve işitsel şekil/form kazandırılır. Bu aşamalar aşağıdaki gibidir:

- Tasarım Araştırması : Bu adım için daha önce de yapıldığı üzere, varsa halihazırda bulunan uygulamaların tasarımlarını, kullanıcı ara yüzlerini (UI), logolarını vs incelemek ile başlanmalıdır.

Burada dikkat edilmesi gereken husus tasarımın içeriğin üzerine inşa ediliyor, içeriğe göre yapılıyor olmasıdır. Bu aşamada içeriğin halen tamamlanmamış olması tasarımın sürekli değişmesine sebep olup gereksiz zaman ve para kaybına yol açacaktır. Bir müzik uygulamasının arayüz tasarımı ile haber uygulamasının arayüz tasarımı aynı olamaz. Bu yüzden, içeriğin tasarım süreçlerinden önce tamamlanması önemlidir.

Örnek toplamak için kullanılabilecek siteler: [10][29][30]

- Akış Şeması (Flowchart) : “Akış şeması: mobil uygulamanın içindeki ekranlar arasındaki geçişleri, hangi ekrandan hangi ekrana geçileceğini ve uygulama içindeki diğer navigasyonel fikirlerin sürecinin her bir aşamasını (ara yüzünün) oklar ile birbirlerine bağlanarak tasarım süreci başlamadan önce görsel olarak düzenlenmesidir.”

Akış Şeması için kullanılabilecek programlar:
[31][32][33][34]

- Digital branding : Akış şemasının ardından tüm tasarım sürecini biraz daha kolaylaştıracak bir logo tasarlamak isabetli olacaktır. Bu logo, uygulamanın açık ya da kapalı olduğu her an onu simgeleyecek ana görsel olacaktır. Dolayısıyla diğer uygulamaların logolarının da yanında renk, şekil ve çizim olarak uygun bir tasarıma sahip olmalıdır. Logonun tasarlanmasının ardından diğer görsel içeriği de buna göre yapmak gerekir.
- Wireframing : Wireframe Türkçe’ye çevirisi telkafes olsa da aslında geliştirilen mobil uygulamanın arayüzünün şematik bir planıdır. Bir diğer deyişle tasarımın görsel bir iskeletidir. Burada wireframe, elle çizilebileceği gibi artık dijital ortamda da çeşitli araçlar ile hazırlanabilir. Wireframe görsel elemanların nerede olduklarını genel hatlarıyla göstermek için kullanılır.

Bazı Wireframe araçları- Örnekler için kaynak site:
[35][36][37][38]

- UI Kiti ve arayüz detaylarının tasarımı : User Interface yani Kullanıcı Arayüzünün, daha önce birbirleriyle genel hatlarıyla etkileşimleri belirlenen(wireframe ile) görsel elemanların kendilerinin tasarlanmasıdır. Bu tasarımı, daha önce yapılmış uygulamanın ana logosuna göre yapmak gerekir. Bu tasarım yapılırken kullanılabilecek birtakım siteler elbette mevcuttur fakat bu durum özgün görseller hazırlamanın önüne geçer. Arayüzü tasarlarken renkleri, şekilleri, çerçeveleri gibi tüm görsel detayları iyi düşünmek gerekir. Bu süreçte de tutarlı bir tasarım yapmak tüm ikonların beraber çalışmasını kolaylaştırır. (Tüm bu tasarımlar neticede (UX) kullanıcı deneyimini birincil derecede etkileyen faktördür.)([1][9][11])

Uygulama tasarlamaya yönelik bazı web siteleri-Yaygın mobil tasarım modelleri: [13][39]

2.5. Uygulama Geliştirme Yaklaşımını Belirlemek (HYBRID-NATIVE-WEB)

Tasarım tamamlandıktan sonra asıl işlemlerin yapıldığı, fonksiyonların gerçekleştirildiği kısım yazılımdır. Her tıklanan butonun altında yapılacak işlemlerin cihazı yormadan hızlı çalışması gerekir. Bunu yapabilmek için algoritma yapısının iyi oluşturulmuş olması gerekir.

2.5.1. IDE Nedir? (Integrated Development Environment – Tümlşik Geliştirme Ortamı) (Tanım-1)

“IDE bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür.”

2.5.2. Native (Yerel)

Platforma özel olarak yani platformun kendine has IDE ve Dilleriyle uygulama geliştirilmesidir. Örnek olarak; IOS platformuna Native olarak uygulama geliştirmek için Apple’ın kendine has dili olan Swift veya Objective C bilmek ve IDE olarak Xcode kullanmak gerekir. Aynı zamanda muhakkak bir Mac OSX işletim sistemli bir bilgisayar da bulunmalıdır.

Android platformuna Native olarak uygulama geliştirmek için Google’ın kendine has dili olan Kotlin veya Java bilmek ve IDE olarak da Android Studio (ya da Eclipse, Netbeans) kullanmak gerekir.

2.5.3. Hybrid (Karma-Melez)

Hybrid uygulama ise platforma özel ayrı ayrı proje geliştirmeye gerek kalmaksızın tek kod ile yazılan ve çeşitli platformlarda çalışabilen uygulamalardır. Bu tip uygulamalar temelinde C#, HTML, CSS, Javascript, dilleri kullanılarak ortaya çıkarılır. Sonuç olarak Webview şeklinde isimlendirilen bir tarayıcı sayesinde aktif hale getirilir. Cross-platform (çarpaz platform) olarak da adlandırılan bu uygulama geliştirme yaklaşımında, tüm platformlar için hemen hemen aynı sonuç üretilir ve tek bir dil kullanarak native benzeri bir uygulama oluşturulur.

Hybrit uygulama geliştirmek için bir çok IDE bulunmaktadır, ReactNative, Xamarin, Cordova, Ionic, PhoneGap, Titanyum, Smartface... gibi. Her IDE’nin kendine özgü farklı kullanım şekilleri, farklı geliştirme yöntemleri ve dilleri bulunmaktadır.

2.5.4. Web Uygulama Geliştirme

Bu yöntemle geliştirilen uygulamalar, sadece ağ bağlantısı üzerinden HTTP (Hypertext Transfer Protocol) kullanarak çalıştırılabilirler.Web tabanlı uygulamalar esasında cihazın hafızasında yer etmez ve internet tarayıcısını kullanarak aktif bir şekilde kullanılırlar.Ancak bazı web tabanlı geliştirilen uygulamalar istemci-sunucu ilişkisi içinde çalışarak cihaza az da olsa bir kod parçasının yüklenmesini gerektirebilir. Buna

karşın yine de asıl işlem sunucu tarafında gerçekleştirilir ve tamamlanır. İnternet üzerinden aktif bir şekilde çalışabilmeleri en önemli özelliklerindendir. Çünkü bu sayede mekan, zaman, cihaz ayırt etmeksizin çeşitli amaçlar doğrultusunda internete bağlantı sağlandığı sürece her türlü hizmet ve işlemler yürütülebilir haldedir.

Böyle uygulamalara aynı zamanda internet yazılımları, uygulamaları da denebilir. Web tabanlı uygulamalar sadece programlama dilleri kullanarak geliştirilebilirler. Bunun için birçok programlama dili mevcuttur. Ancak sonuç olarak farklı diller ile aynı işleve sahip uygulamalar geliştirilir. İnternet tarayıcısı üzerinden kullanılan ve http protokolü kapsamında haberleşmeyi sağlayan tüm yazılımlar bu tarz uygulamalara örnek olarak verilebilir.

Web tabanlı uygulamaların bir diğer özelliği performans konusunda esneklik sağlayabilmesidir. Yani hangi tarayıcı olursa olsun çalışabilmektedirler. Ayrıca bu uygulamalara herhangi bir güncelleme yapılacağı zaman tüm kullanıcıların cihazlarını güncellemektense sadece sunucunun olduğu bilgisayar güncellenerek çok gereksiz ve uzun ikinci bir işlem yapılmadan çözüme ulaşılır. Bir diğer avantajı ise diğer uygulama geliştirme yaklaşımlarında yaşanabilecek sistem çökme sorununun Web tabanlı uygulamalarda olmamasıdır. Gerekli olduğu takdirde bir başka cihaz ile aynı işlemlere devam edilebilir. Bunun nedeni ise daha önce bahsedildiği üzere web tabanlı uygulamaların adı üstünde internet üzerinden aktif olarak çalışmalarıdır

Web tabanlı uygulamalar kişilerin bireysel siteler yapabilmelerinden büyük çaplı e-ticaret sitelerine varana kadar tek bir merkezde yönetim ve denetim ile işlevsel çözümler oluşturur.

2.5.4. HYBRID VS NATIVE

Hybrid uygulama geliştiricinin artıları-Native'nin eksileri

- En büyük avantajı, tek kod yazılarak tek proje ile tüm platformlarda çalışabilmesidir. “write once, run everywhere”
- Diğer avantajları da çoğunlukla bu ilk özellikten ötürü meydana gelir.
- Bakım ve güncelleme maliyetleri diğer uygulama geliştirme yöntemlerine göre daha düşüktür.
- Çünkü platformlar üstünde ayrı ayrı güncelleme yapmaktansa tek kod üzerinden
- uygulama güncellenir.
- Her bir platformun kendine has IDE ve dillerini bilmek yerine tek dil ve ortam bilindiği
- takdirde uygulama yayınlamak mümkündür.
- Tek proje olması dolayısıyla uygulamalar daha hızlı geliştirilir ve piyasaya daha çabuk sürülür.
- Uygulamayı pazara sürmenin ardından uygulamanın güncellemeleri çok daha kolay bir şekilde yapılabilir.
- Hem çevrimiçi hem de çevrimdışı çalıştırılabilir.

Hybrid uygulama geliştiricinin eksileri-Native'nin artıları

- Native uygulamalarda, platforma ayrı ayrı kod, tasarım, gibi aşamaların
- özel bir şekilde gerçekleştirilmesinden ötürü tek proje ile geliştirilen Hybrid uygulamalar, kendilerinin her ne kadar güzel avantajları olsa da bazı konularda Native uygulamaların gerisinde kalmaktadır.
- Geliştirilen uygulamanın performansında Native olarak geliştirilmiş bir uygulamaya göre
- gözle görülür yavaşlıklar meydana gelebilir.
- Hybrid uygulamalar gün geçtikçe iyileşmelerine rağmen, yine de performans konusunda
- Native uygulamaların gerisindedirler.
- Hybrid uygulamalar aynı zamanda görsel olarak da native uygulamaların gerisindedirler.
- Hybrid uygulamalar tek projeden meydana geldiği için yani ortak bir tasarım yapılmasından
- ötürü arayüzdeki butonlar, kontroller, şekiller; bunların yerleri ve görüntüleri
- kullanıcı gözünden biraz yetersiz kalabilir.
- iOS ve Android, birçok yönden birbirlerinden farklı iki platformdur. Yani hem görsel
- hem de arkaplanda işleyen yazılımlar Native olarak geliştirildiği takdirde uyumu
- ve işleyişi daha iyi sağlanmasına
- karşılık Hybrid uygulamalarda birtakım uyumsuzluklar meydana gelebilir.
- Apple ve Google firmalarının her yeni çıkarttığı yeniliklerde(güncelleme vs.)
- yukarıda da bahsedildiği üzere ortaya çıkan uyumsuzluklar Hybrid yönteminde kolaylıkla giderilemeyebilir.
- Benzer bir sebepten ötürü yine işletim sistemi güncellendiği takdirde bu uygulamalar bazen
- uygulamanın görsel olarak arayüzünü ve davranışını da etkileyebilir.
- Hybrid uygulamalarda bu durum sebebiyle uyumsuzluklar çıkarken
- Native uygulamalara pek yansımaz.

Karar verirken kısaca aşağıdaki durumlar incelenmelidir:

- Eğer uygulama çok büyük kitleler için geliştiriliyorsa,
- Maddi olarak bütçe geniş ise,
- Uygulamada basit bir şekilde sadece veri alıp göstermenin ötesinde
- bir iş yapılacaksa,
- Uygulama telefonun tüm imkanlarını sonuna kadar çok etkin bir şekilde kullanacak ise,
- Uygulama hem tasarım hem de performans anlamında üstün olması isteniyorsa;
- Native uygulama seçimi daha isabetli olacaktır.
- Uygulamanın hitap ettiği kesim çok geniş değilse,
- Maddi olarak bütçe az ise,
- Uygulamada çok karmaşık olmayan birtakım işlemler yapılacaksa(gazetelerin basit bir şekilde haber gösterimi yapması gibi.)
- Uygulamanın arayüz gibi kısımlarının görsel olarak öneminin çok da bulunmadığı durumlarda,
- Çok daha kısa bir sürede uygulama geliştirilmesi isteniyorsa,

- Hybrid uygulama seçimi daha isabetli olacaktır. [1][3][5][8][20]

2.6. UYGULAMAYI TEST ETMEK

Kullanıcı Deneyimi (UX-User Experience)'ni en iyi hale getirebilmek adına yazılan kodları ve tüm arayüzü, kullanıcı gözüyle incelemek ve defalarca test etmek gerekir. [1][19]

3. Uygulama Geliştirme Ortamları

Tümleşik Geliştirme Ortamları (IDE-Integrated Development Environment) (Tanım-2)

“Bünyesinde programlama editöründen test uygulamasına, tasarım oluşturmaktan hata ayıklama programlarına kadar pek çok özelliği içeren paketlere IDE denir.”

Hybrid için; Apache, Cordova, React Native, Build Fire, Xamarin, IonicA

Native için; Android Studio, Eclipse, Netbeans, Xcode [3]

3.1. Native Uygulama Geliştirme Ortamları

3.1.1. Android Studio

“Android uygulamalarının geliştirildiği, üst seviye özelliklere sahip ve Google tarafından da önerilen resmi programlama aracıdır.” Android Studio bazı Apple, Linux ve Windows işletim sistemleriyle uyumu bulunmaktadır. Özellikle Google firmasının çeşitli destekleriyle 2013 yılından bu yana Native olarak Android platformuna uygulama geliştirmede büyük öneme sahiptir.Projelerde kullanılabilecek birçok araç seti bulundurulur.Temeli IntelliJ IDEA’ya dayanmaktadır.

(IntelliJ IDEA JetBrains tarafından geliştirilmiş Java ile yazılım geliştirmek için kullanılan bir IDE’dir.)

Dolayısıyla AS’nin ilk dili Java’dır.

- Gradle tabanlı, esnek proje oluşturma sistemidir.
- (Gradle, yazılım geliştirme sürecinde ihtiyaç duyulan birtakım kütüphaneleri geliştiricinin tek tek indirmesine gerek kalmadan bunu geliştirici yerine yapan bir sistemdir.)
- Android Studio’nun geliştiricilere temelde sunduğu özellikler:

“Android Emülatör, sanal bir Android'dir. Windows başta olmak üzere işletim sistemlerine kurulan bu emülatörler, sanal bir cihaz kullanmanıza olanak tanır. Android emülatörler sanal telefonlar ya da sanal tabletlerdir. Windows için android

emülatör demek windows içerisinde sanal android cihaz çalıştırmak demektir.”

- Hızlı ve zengin özellikli bir emülatöre sahiptir.
- (Android Emülatör, sanal android yüklü cihazlardır.
- Fiziksel bir cihaza sahip olmaya gerek kalmadan örneğin Windows içerisinde sanal olarak android işletim sistemi ortamını sunar.)
- APK çıktısı alınabilmektedir. (Android Package Kit, kısaca android işletim sistemindeki dosyaların temel formatıdır. Uygulamaların işlenmemiş dosya biçimidir. Windows için .exe ne ise Android için de .apk odur. Google Playstore da bulunan desteklenmiş apk lerin yanı sıra internet üzerinden de pek çok yabancı kaynak tarafından apk’ler indirilebilir.)
- Uygulamanın genel özelliklerini oluşturmaya ve örnek kodu içe aktarmaya yardımcı olacak kod şablonlarına sahiptir ve GitHub ile entegrasyonu bulunmaktadır.
- C ++ ve NDK(Native Development Kit) desteği bulunmaktadır.
- Uygulamanın ekran tasarımlarını yani kullanıcı arayüzü UI yapmada kolaylık sağlayan sürükle-bırak özellikli zengin bir editörü bulunmaktadır.
- Uygulama geliştiricisine yardımcı birçok test aracı ve frameworkleri bulunmaktadır.(Framework, yazılım iskeleti veya çerçevesi de denebilen, standart birtakım fonksiyonlar ve kütüphaneler barındıran, yazılımcıya da bunlarla alakalı olarak ekleme veya değişiklik yapma olanağı tanıyan sistemlerdir.)
- Uygulamayı pazara sürmeden önce APK imzalanması gerekmektedir ve Android Studio ile de bu işlem kolay ve güvenli bir şekilde yapılabilir.
- Google’ın kendi resmi IDE’si olmasından ötürü herhangi ek bir çaba göstermeden Google hizmetleri uygulamaya eklenebilir. [23][28]

Bunların yanında bazı olumsuz yönleri de vardır: Bir uygulama yazmak için genellikle diğer ortamlara kıyasla daha fazla satır yazmak gerekir bu da doğal olarak zaman kaybı yaşatır. Karmaşık bir uygulama ya da animasyonlarla ilgili bir uygulama yapılacağı zaman ise diğer ortamlara göre kodlaması zorlaşmaktadır. En büyük dezavantajlarından bir diğeri ise güvenlik açısından zayıf bir yapıya sahip olmasıdır. Bu güvenlik sorunu yazılan uygulamaları başkalarının da basit bir biçimde erişmesine olanak sağlar. [40][41][42]

3.1.2. Eclipse

“2001 yılında IBM tarafından ortaya çıkarılmış açık kaynak kodlu IDE’dir. Odak noktası Java ve Java ile ilişkili teknolojiler olsa da esnek yapısı sayesinde C,C++, Ruby, Python, PHP, Javascript gibi farklı diller için de kullanılabilir.Arayüzü, şık görünümü ve çok kuvvetli

özellikleriyle Java geliştiricileri genellikle Eclipse kullanmayı tercih ederler.”

Eclipse projeleri 2005 yılından itibaren kar amacı gütmeyen Eclipse Vakfı tarafından yönetilmektedir. Eclipse Vakfı kar amacı gütmeyen, üyeleri tarafından desteklenen, Eclipse açık kaynak kodlarını sağlayan ve hem açık kaynak topluluğunu ilerleten hem de tamamlayıcı ürünler ve servislerini içinde barındıran bir kuruluştur.

Eclipse Temel Özellikleri

- Başlıca özelliklerinden biri ücretsiz ve açık kaynaklı olmasıdır.
- Farklı dilleri desteklemektedir.
- Programda kod satırını kaydırma genişliği değiştirilebilir.
- Yazılan kodları tamamlayabilir.
- Eclipse'nin genişletilebilir birçok eklentisi mevcuttur. Bunlara plugin adı verilir.

(“Plug-in, kendi başına çalışabilen bir program için, genellikle çok özel bir alanda duyulan gereklilik üzerine geliştirilen, programa yeni özellikler ekleyen yazılımdır.”)

-
- Eclipse IDE içerisinde Java derleyicisi bulunur.
- Geliştiricilere sürükle-bırak özelliği ile tasarımda kolaylık sunar.
- Sonuçta toplum destekli bir IDE'dir.
- Farklı projeler arasında kolaylıkla paylaşılabilir. [25][26]

3.1.3. Netbeans

“Oracle firması tarafından geliştirilen bir geliştirme ortamıdır. 2010 yılında ise Netbeans 1.000.000 aktif kullanıcı kitlesine ulaşmıştır.”

Avantajları

- Kullanıcı arayüzünün hem hızlı hem de kolay yapısıyla çeşitli platformlar için proje yapımı sürecinde kolaylık sağlar.
- Netbeans geliştirme ortamının kurulumu kolaydır.
- Java tabanlı olmasına rağmen PHP, HTML5 ve C/C++ gibi dilleri de destekler.
- Açık kaynak koda sahiptir.
- Sürekli gelişen ve kullanımı artan bir projedir.
- Ücretsiz bir üründür.
- Proje yönetimi kolaydır.
- Windows, Linux, MacOS X, gibi birçok platformda çalışır.[25]

3.1.4. Xcode

Xcode projesi ilk defa Steve Jobs, NeXT şirketinin(Apple'ın önceki kuruluşu) başındayken “Project Builder” ismiyle ortaya çıktı. Proje, daha sonra yıllar içinde değişiklikler geçirerek 2003 yılında “Xcode” ismini aldı. Sonrasında bir süre küçük sürümlerle devam eden proje 2014 yılında Swift dilinin

tanıtılmasıyla tam anlamıyla bir iOS işletim sistemli cihazlar için uygulama geliştirmenin temelini oluşturmuştur.

“Xcode: Apple, iPad, iPhone, Apple Watch ve Mac gibi Apple ürünleri için uygulamalar oluşturmak için kullanılan ücretsiz, Apple tarafından hazırlanmış ve tek entegre geliştirme ortamıdır (IDE).” Xcode, uygulamanın oluşturulmasından, test edilmesine, optimize edilmesine ve App Store'a gönderilmesine kadar ki süreç için çeşitli geliştirme araçları sunar.

Xcode ile geliştirme yapmak için, Mac OS X yüklü bir bilgisayar yada sanal makine gereklidir. [3][23][43]

3.2. Android Geliştirme için diller

3.2.1. Java

“Java ilk olarak Sun Microsystems tarafından 1995 yılında piyasaya sürülen bir programlama dili ve bilgi işlem platformudur. Java neredeyse her türdeki ağ uygulamalarının temelini oluşturarak gömülü ve mobil uygulamalar, oyunlar, Web tabanlı içerik ve kurumsal yazılım geliştirme ve dağıtımı için küresel standarttır. Dünya genelinde 9 milyonu aşan geliştiricisiyle Java, heyecan verici uygulamaları ve hizmetleri verimli bir şekilde geliştirilmesine ve dağıtılmasına olanak tanır.”

Pek çok Java geliştiricisi ile özel bir topluluk oluşmuştur ve bu topluluk Java'yı geliştirmiş, genişletmiş, iyileştirmiş, test etmiş ve onaylamıştır. Uygulamaları karma bir şekilde çeşitli platformlarda çalışabilir hale getirerek daha fazla hizmet sunulmasını sağlamışlardır.

Java ile aslında diğer dillerde de olduğu gibi hem kurumsal hem de bireysel uygulamalar geliştirilmesi mümkündür.

Avantajları

- Java'nın en büyük avantajı platformdan bağımsız olarak çalışmasıdır. Yani işletim sistemi ne olursa olsun o işletim sistemine ait bir Java yorumlayıcısı ile Java kodları çalışacaktır.
- Günümüzdeki tüm işletim sistemleri Java'yı desteklemektedir. Windows95'ten sonra Java'yı Microsoft da desteklemiştir. Şuanda ise Sony Playstation 2'ye ve cep telefonlarına bile Java yorumlayıcısı girmiştir.
- Java yorumlayıcısının en büyük özelliği yazılan Java programını bir kere Java makine koduna çevirdikten sonra her işletim sisteminde çalışan bir dosya haline getirmesidir.
- Java'nın bir diğer avantajı ise nesne yönelimli bir dil olmasıdır. Yapı olarak C++'ı andırır. Bu yüzden daha önce bu dili bilenler için Java'yı öğrenmek daha basit bir hal alır.
- Java dilini öğrenirken en çok çalışılması gereken bölüm, nesneye dayalı programlama(Object Oriented Programming)'dır. (**Object Oriented Programming**, Nesne Yönelimli Programlama (NYP) mantıksal işlemlerden ziyade, nesnelere (object) ve nesneler üzerinde işlemlere odaklanan programlama dili modelidir. NYP'de programlar, nesnelerin birbirleriyle etkileşime geçmeleri

sağlanmasıyla tasarlanır. Java, C++, C#, Python, PHP, JavaScript, Ruby, Perl, Smalltalk, Objective-C gibi diller başlıca nesne yönelimli programlama dilleridir.”)

- Diğer bir avantaj Java'nın web tabanlı bir dil olmasıdır.

Dezavantajları

- Java'nın en büyük dezavantajı, C/C++ dillerine göre çok yavaş olması ve java yorumlayıcısına ihtiyaç duymasıdır.
- ARM işlemcileri bu noktada sorunu çözebilmektedirler.

[3][45][46][47][48][49][50]

3.2.2. Kotlin

- Kotlin, JetBrains firması tarafından 2010 yılında piyasaya sürüldü.
- Kotlin de bir statik programlama dilidir.(Statik programlama dili ise değişkenlerin önceden belirlenmesidir.)
- Kotlin, açık kaynak kodlu bir projedir.
- Projenin kaynak kodları herkese açıktır.
- Google, 2017'de Android geliştirme için Kotlin'i resmi bir dil olarak benimsemiştir

Kotlin Neden Geliştirildi?

Java varken neden bu dile ihtiyaç duyuldu ? JetBrains firması bu konu hakkında ; “Performans ve güvenliği feda etmeden Java'dan daha özlü kodlar yazmayı sağlamak için bu dili geliştirdik” cevabını vermektedir. Kotlin'in çıkış hedefi daha az satır kod ile daha güvenli bir kod yazmaktır. Buna rağmen Java hala kullanılmaktadır.

- Kotlin; Java, C# ve C++ gibi nesne yönelimli bir programlama dilidir.
- Kotlin sade ve kendine özgün olmasıyla geliştiricilerin beğenisini kazanır.
- Kotlin, Android ve Java ile tam uyumlu bir şekilde çalışmaktadır.
- İstemci-Sunucu yani web tabanlı uygulamalar geliştirilebilmektedir.
- HTML sayfalarında kullanılabilmesi için JavaScript kodlarına derlenebilir.
- Kotlin ve Java birbirlerine çevrilebilir. Program yazarken birlikte kullanılabilir.
- Kotlin dilini ön plana çıkaran özellik :

”Google Şirketinin Android Developer bölümünün bu dile güvenmeleri ve Android uygulamaları geliştirmek için desteklemeleridir.”

[3][24]

3.3. IOS Geliştirme İçin Diller

3.3.1. Swift

- Swift, 2 Haziran 2014 yılında WWDC konferansında Apple tarafından duyurulmuştur.
 - Swift, Apple'ın kendi platformlarına uygulama geliştirilmesi için oluşturulmuş, nesne yönelimli bir dildir.
 - Swift programlama Xcode geliştirme ortamı için kullanılır.
 - Swift, açık kaynaklı bir dildir.
 - Swift, Objective-C ile tam uyumludur. Birlikte kullanılabilir.
 - Swift dili, 2014 yılında çıkmasına rağmen (alternatif olmadığı için) büyük ilgi görmüştür
 - Swift, Objective C 'ye göre 2.7 kat daha hızlı çalışmaktadır.
- [3]

Apple aynı zamanda oluşturduğu bu dilin daha rahat öğrenilmesi için görsele dayalı olan Swift Playgrounds'u kullanıcılara sundu.

Swift Playgrounds: “öğrencileri ve yeni başlayanları, Apple'ın sunduğu ve profesyonel geliştiricilerin birinci sınıf uygulamalar oluşturmak için kullandığı, öğrenmesi kolay programlama dili Swift ile çalışmayı keşfetmeye teşvik eden interaktif bir arayüzle kodlamayı hayata geçiriyor. Swift Playgrounds, Apple tarafından geliştirilen programlama derslerini kapsıyor.” [43]

3.3.2. Objective C

- Objective-C, C tabanlı, nesne yönelimli bir dildir.
- Objective-C, 1983 yılından itibaren geliştirilmeye başlanmıştır.
- Objective-C, 2008 tarihinden sonra iPhone cihazlarının tanıtılmasıyla birlikte dünya üzerinde popüleritesi oldukça artmıştır.

Swift mi Objective C mi sorusunun cevabı olarak ise;

Swift 2010 yılında geliştirilmeye başlamasından dolayı sorunlara daha modern çözümler getirmektedir. Öte yandan iOS'a ya da OS X'e nasıl uygulama hazırlanacağını öğrenirken vaktin çok büyük kısmı Apple kütüphanelerini öğrenerek harcanacaktır.

Sonuçta yeterli vakit varsa Swift öğrenmekte fayda var fakat vakit kısıtlıysa Objective-C öğrenmek daha mantıklı olacaktır.

*Apple platformları için her iki programlama dili de kullanılmaktadır.

[3][44]

3.2. Hybrid Uygulama Geliştirme Ortamları

3.2.1. IONIC

- Ionic; AngularJS ve Cordova tabanlı bir framework'tür.
 - Ionic ile geliştirme yapabilmek için HTML, JavaScript ve Angular.js bilmek yeterlidir.
 - En iyi hybrid geliştirme kitlerinden biridir.
 - IonicLab ile Android ve iOS üzerinde uygulama derlemesi ve test işlemleri gerçekleştirilebilir.
 - Uygulamanın tasarımı her iki platform için de tek hamlede yapılabilir.
 - Geliştirme kitinin LiveReload özelliği sayesinde yapılan değişiklikler anında izlenebilir.
 - Web geliştiricilerinin uyum sağlaması kolaydır.
 - Uygulama modüler halde geliştirilebilir.
 - ngCordova ile pluginler Angular.js üzerinden kullanılabilir.
 - Cordova pluginleri sayesinde uygulamaya bir çok native özellikler de eklenebilir.
 - Uygulamaların arayüz tasarımlarının büyük bir kısmı Ionic Creator ile sürükleyip bırak mantığı ile yapılabilir.
 - Ionic View App sayesinde uygulamanın testlere denetleme imkanı sağlanır.
- [8][22]

3.2.2. React Native

React Native, Facebook tarafından geliştirilen, multi-platform mobil uygulama geliştirme olanağı sağlayan bir geliştirme kitidir.

React Native, 2013 Yaz'ında Facebook'un yaptığı hackathon ile front-end* olarak geliştirilmeye başlanmıştır. Yaklaşık 2 yıllık geliştirmeden sonra 2015 Ocak ayında ilk kararlı sürümü yayınlanmış bundan 2 ay sonrada GitHub üzerinden kodları açık kaynak haline getirilmiştir.

*"front-end; bir uygulamanın ara yüzünde ki tasarımsal ve operasyonel işlemlerin bütünüdür." [3]

3.2.3. Xamarin

Xamarin; C# ve .NET kullanılarak Hybrid uygulama geliştirmemizi sağlayan bir ortamdır.

Avantajları

- Xamarin ile hem iOS hemde Android uygulama yapılabilir.
- Xamarin üzerinde Swift, Objective-C veya Java ile gerçekleştirilebilecek her şey yapılabilir.
- Microsoft'un resmi olarak desteklemesiyle geliştiricilere yardım sağlar.
- Visual Studio üzerinde kullanılmaktadır.
- Geliştiriciler kendi kütüphanelerini yazabilir. Bu nedenle geniş bir kütüphanesi bulunmaktadır.
- Akıllı saatler için uygulama geliştirilebilir.

- C# bilgisi aynı şekilde Xamarin için de kullanılabilir.
- Geliştirici sayısı çok fazla olması sebebiyle kaynak anlamında oldukça önemli imkanlar sunmaktadır.
- Xamarin SDK'leri GitHub üzerinden paylaşılmaktadır.

Dezavantajları

- Xamarin ile geliştirilen uygulamalar Native geliştirme yöntemlerine göre daha fazla yer kaplamaktadır.

[46][12][27]

3.2.4. Cordova

"Apache Cordova, mobil platformlara uygulama oluşturmak için JavaScript, HTML ve CSS içeriğini kullanmalarını sağlayan açık kaynaklı bir geliştirme kitidir."

Avantajları

- Hızlı uygulama geliştirme ve prototipleme için en uygun geliştirme kitidir. WebView ile JS ve CSS kütüphanelerini kullanarak oldukça güzel uygulamalar oluşturmak mümkün.
- Çoklu platformlara uygulama geliştirme temel amacıdır.
- Native özellikler için plugin desteği mevcuttur.
- Tarayıcı üzerinden geliştirilmeye olanak sağlar.
- Öğrenim maliyeti azdır. Uygulama geliştirmek için diğer platformlarda gereken pek çok dili öğrenmektense temel web bilgileri yeterlidir.
- Geliştirme maliyeti düşüktür. Her platform için ayrı personel maliyetinin olmasını engeller, tek bir personel ile tüm platformlara uygulama geliştirilebilir. [8][21]

4. Mobil Uygulama Trendleri

Son birkaç yılda, birçok teknoloji devi mobil uygulama geliştirme endüstrisine büyük yatırımlar gerçekleştirdi. Teknoloji şirketlerinin yaptıkları araştırmaların sonuçları mobil uygulama geliştirme endüstrisinin en hızlı büyüyen endüstrilerden biri olduğunu ve gelecekte de yavaşlama belirtileri göstermediğini ortaya koydu.

Dünyamızın dijitalleşmeye yöneldiği çağda, mobil uygulamalar yaşamın her alanında yanımızdalar ve hayatımızı sürekli değiştiriyorlar. Mobil uygulamalarda oldukça etkileyici ve yenilikçi teknolojilere doğru yönelmekteyiz. Bu teknolojilerin kullanımı mobil uygulama geliştirmede kaçınılmaz eğilimler haline geldi. Bu teknolojileri kullanmak ve güncel kalmak, mobil uygulamalar için önemli bir husustur. Aşağıda mobil uygulama geliştirmenin geleceğine yön veren eğilimlerden bazıları yer almaktadır.

[52] [55]

4.1. Arttırılmış Gerçeklik ve Sanal Gerçeklik (AR & VR)

Arttırılmış Gerçeklik (AR) daha çok objeler üzerinden nesneleri 4 boyutlu görmemizi sağlayan teknolojidir. Sanal Gerçeklik (VR) ise sanal bir alanda 360° dolaşabilme olanağı sunan teknolojidir.

Teknoloji devleri her iki teknoloji için de yeni kullanım çeşitleri geliştirmektedir. Bu teknolojiler sadece oyun uygulamalarının deneyimini arttırmakla kalmayıp sosyal medya uygulamalarında da kendilerini göstermiştir. Örneğin, Snapchat ve Instagram bir insan yüzünü çeşitli dijital karakterlere dönüştürebilen AR filtrelerini piyasaya sundular. AR'nin sosyal medyada kullanılması, kullanıcılara çok eğlendikleri bir deneyim sunmuştur.

AR ve VR teknolojileri ile yapılacak daha birçok şey olduğunu düşünmekteyiz ve yıllarca gelişen bir mobil uygulama trendi halini alacaktır.

[53] [54] [60] [62]

4.2. Blockchain (Blok Zinciri) ve Mobil Ticaret

Akıllı cihazlar yaygınlaştıkça mobil ticaretin kullanımı artmıştır. Banka ve kredi kartları yerlerini akıllı cihazlara bıraktığını görmekteyiz. Temassız ödeme yöntemleri artık müşterilerin kullanmayı sevdikleri ödeme yöntemi olmuştur. Bu nedenle uygulama geliştiricilerin banka ve kredi kartlarına gerek duymadan işlemleri gerçekleştirebilecek bir mobil uygulama oluşturmaya yöneltmiştir. Ancak bu işlemlerin ve hesapların bir güvenliğe ihtiyacı olmuştur. Bu noktada ise karşımıza blockchain gelmektedir.

Blockchain'in mobil uygulamalara entegrasyonu, uygulamalarda ki ödeme yöntemlerinin en yüksek seviyede güvenli olmasını sağladı. Blockchain birçok mobil cüzdan, döviz uygulamaları ve alışveriş uygulamalarında kullanılmaktadır ve yakında her türlü uygulamada yaygınlaşacaktır.

[51] [53] [54]

4.3. Internet Of Things (IoT (Nesnelerin İnterneti))

IoT, fiziksel nesnelerin birbirleriyle veya daha büyük sistemlerle bağlantılı olduğu geniş bir ağıdır. IoT'nin mobil uygulamalarda kullanılması akıllı cihazların, akıllı telefonlarla uzaktan kontrol edilmesine olanak sağlar. Örneğin IoT teknolojisi sayesinde Nest Akıllı Termostat ile ev sıcaklığını ayarlayabilir, Canary Akıllı Güvenlik Sistemi ile evinizi davetsiz misafirlerden koruyabilir veya Petnet Akıllı Evcil Hayvan Besleyici ile siz evde olmasanız bile evcil hayvanlarınızı besleyebilirsiniz. Ayrıca IoT sayesinde sokaklar daha da akıllı hale gelecek. Bunun üzerinde çalışanlar IoT sayesinde trafikteki sıkışıklıkları ve park sorunlarının en aza indirilebileceğini düşünüyorlar.

IoT'nin sağlık, ulaşım, e-ticaret ve daha bir çok alanda uzun bir süre trend olarak devam edeceğinden şüphemiz yok.

[51] [52] [54] [55] [61]

4.4. Cloud (Bulut) Teknolojisi

Akıllı telefonlarımızın kapasiteleri biraz sınırlıdır. Bu noktada karşımıza çıkan teknoloji Cloud teknolojisidir. Cloud teknolojisi bize geniş bir depolama alanı sunar. Böylelikle depolama maliyetini de düşürmüştür. Bu teknoloji mobil uygulamalarla birleştiğinde, uygulamaların depolama özelliklerini geliştirir ve verimliliğini arttırmıştır. Önümüzdeki yıllarda bu teknoloji sayesinde depolama sorunu yaşamayacağız gibi gözüküyor.

[53] [55] [63]

4.5. Yapay Zeka (AI) ve Makine Öğrenimi

Yapay zekanın mobil uygulamalara entegrasyonu işletmelere çok para kazandırmakla kalmayıp kullanıcıların ilgisini de çekiyor. Yapay zeka karşımıza daha çok sohbetler olarak karşımıza çıkmaktadır. Çevrimiçi alışveriş ve yiyecek dağıtımı gibi hizmetleri içeren uygulamalar yapay zeka sayesinde kısa sürede müşterilerine yanıt verebilmesini sağlamaktadır. Yapay zeka ve makine öğreniminin kombinasyonu olarak karşımıza çıkan Siri ise Apple kullanıcılarının ayrılmaz bir parçası olmuş durumda. Yapay zeka ve makine öğreniminin birleşmesi gelecek yıllarda teknolojide harikalar yaratmasını beklemekteyiz.

[53] [55] [59]

4.6. (Beş)5G Teknolojisi

Önceki neslinden 10 kat daha hızlı olan 5G teknolojisi sayesinde saatlerce yüklenmesini beklediğimiz uygulamalar artık dakikalar içinde yüklenmiş olacak. Zamandan tasarruf ve iyileştirmenin yanı sıra, 5G teknolojisi diğer teknolojilerin gelişmesine de büyük ölçüde yardımcı olacaktır.

[67]

4.7. Giyilebilir Teknoloji

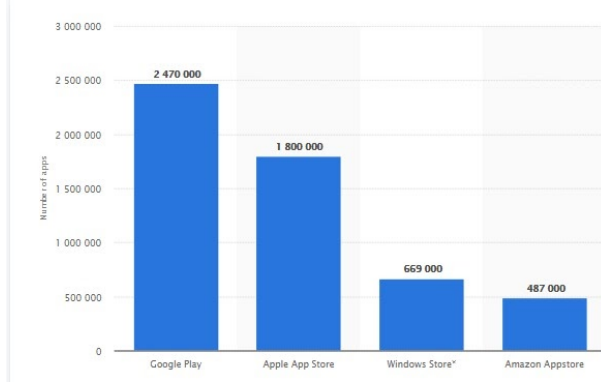
Giyilebilir cihazların ana kontrol panelleri akıllı telefonlardır. Akıllı telefonlarımıza indirdiğimiz uygulamalarla onları kontrol etmemiz çok kolay olmaktadır. Bir çok giyilebilir cihaz olsa da aralarında en popüler olanları hareket takip ekipmanları ve akıllı saatlerdir. Apple Watch ve Hololens bunlara en iyi örneklerdendir.

Giyilebilir teknolojinin en önemli özelliklerinden biri internete bağlanarak cihaz ile ağ arasında kolayca veri aktarımı yapılabilmesidir. Sağlık, spor, fitness, moda ve daha bir çok alanda yaygınlaşması beklenen bu cihazların kullanımının artması giyilebilir teknolojinin uzun bir zaman daha trend olarak devam edeceğinin göstergesidir.

[51] [54]

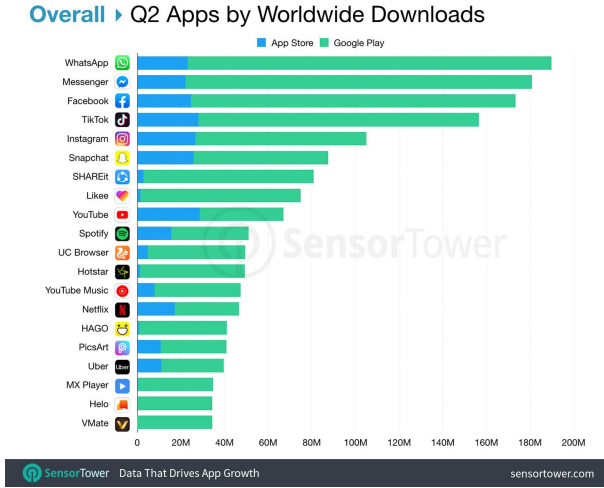
4.8. Sonuç

Tablo 1: Mobil mağazalarda yer alan güncel uygulama sayıları [64]



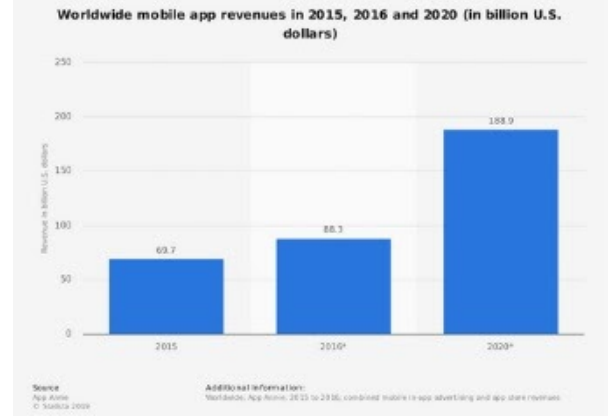
Tablo 1’de mobil mağazalarda yer alan uygulama sayıları görülmektedir. Bu istatistikler göz önüne alındığında, mobil uygulama sektörünün ne kadar büyüdüğü ve büyümeye devam edeceği görülmektedir.

Tablo 2: En çok indirilen uygulamalar ve indirilme sayıları (2019’un 3. çeyreği) [65]



Tablo 2’ye bakıldığında en çok haberleşme ve sosyal medya uygulamalarının indirildiği görülmüştür.

Tablo 3: Mobil uygulama sektöründe öngörülen gelir tablosu



Dünyada uygulama kullanımı arttıkça, mobil uygulamalardan gelirin arttığı gözlemlenmiştir.

Yakın tarihli bir araştırmada (Tablo 3), mobil uygulama geliştirme gelirlerinin 2020 yılında 188,9 milyon dolar küresel gelir sağlayacağını öngörmüştür.

KAYNAKÇA

- [1] www.newgenapps.com/blog/bid/219838/10-steps-to-create-a-successful-mobile-application
- [2] <https://support.mobiroller.com/tr/bilgi-bankasi/uygulama-ic-i-satin-alma/>
- [3] <https://medium.com/kodcular/react-native-nedir-7b333d319597>
- [4] <https://github.com/ecamlioglu/App-Development-for-Mobile-Devices>
- [5] <https://www.mediatick.com.tr/blog/native-mobil-uygulama-nedir>
- [7] <https://www.hukumdar.com.tr/post/kullanisli-mobil-uygulamalar-gelistirmek-Icin-10-onemli-Ipucu>
- [8] <http://devnot.com/2017/cordova-ile-hibrit-mobil-uygulama-ara-giris/>
- [9] <https://journio.com.tr/6-adimda-mobil-uygulama-arayuzu-tasarimi>
- [10] www.behance.net
- [11] <https://uygulamasepetim.com/mobil-uygulama-gelistirmek/>
- [12] <https://bilginc.com.tr/blog/20/xamarin-nedir-xamarin-framework-hakkinda-bilmedikleriniz>
- [13] www.calltoidea.com
- [14] <http://mobilistanbul.org/fikirden-markete-mobil-uygulama-gelistirme-asamalari/>
- [15] www.swift.org
- [17] www.dotnet.microsoft.com
- [18] <https://thebhgroup.com/blog/mobile-app-development-process>
- [19] <https://www.mobiluygulama.com/mobil-uygulama-gelistirmeye-nereden-baslanmali/>
- [20] <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/>
- [21] www.ionicframework.com
- [22] <https://wontlab.com/blog/teknoloji/ionic-nedir>
- [23] <https://developer.android.com/studio/intro>
- [24] <https://ata.com.tr/blog-detay/kotlin-nedir-148>
- [25] <https://netbeans.org/switch/why.html> <https://netbeans.org/>
- [26] www.docs.microsoft.com/
- [27] www.xamarinturkiye.com/xamarin-nedir/
- [28] <https://www.techopedia.com/definition/33631/android-studio>
- [31] www.draw.io
- [32] www.lucidchart.com
- [33] www.shapesapp.com
- [34] <https://www.gliffy.com/examples/flowcharts>
- [35] <https://gomockingbird.com/home>
- [36] www.lumzy.com(kaynak düzenlemesi sırasında sitenin kapandığı belirlenmiştir.)
- [37] <https://www.simplediagrams.com>
- [38] www.xiffe.com(kaynak düzenlemesi sırasında sitenin kapandığı belirlenmiştir.)
- [39] <https://www.mobile-patterns.com>
- [40] <https://searchmobilecomputing.techtarget.com/definition/Android-Studio>
- [41] www.developer.android.com
- [42] <https://www.credosystemz.com/training-in-chennai/best-android-training-in-chennai/>
- [43] www.apple.com
- [44] <https://www.mertdogan.net/tag/objective-c-nedir/>
- [45] <http://www.vepida.com/java-8217nin-avantajlari-dezavantajlari/detail/?ID=XPOQT>
- [46] <https://hasaneksi.net/xamarin-nedir-avantajlari-ve-dezavantajlari-nelerdir/>
- [47] https://web.itu.edu.tr/kamasak/pubs/pdf/Baris_Sarikaya_tez.pdf
- [48] <https://teknogof.com/nedir/java>
- [49] http://www.fibiler.com/Divisions/Ehil/Mecmua/Magazines/Articles/txt/html/article_WhatIsJavaAndHowRun.html
- [50] <https://blog.burakkutbay.com/java-native-interface-nedir-jni.html/>
- [51] <https://marutitech.com/7-trends-of-mobile-application-development/>
- [52] <https://hackernoon.com/top-mobile-application-development-trends-in-2019-5bc1ba19188>

[53] <https://thenextscoop.com/mobile-app-development-trends-2019/>

[54] <https://www.peerbits.com/blog/top-mobile-app-development-trends.html>

[55] <https://www.smartinsights.com/mobile-marketing/app-marketing/top-mobile-app-development-trends-in-2020-infographic/>

[56] <https://www.pixelcrayons.com/blog/top-app-trends-2019/>

[57] <https://towardsdatascience.com/top-14-iot-mobile-app-development-trends-to-expect-in-2020-7fd7718155dc>

[58] <http://www.yeniisfikirleri.net/gyilebilir-teknoloji-nedir-orneklere-nelerdir/>

[59] <https://becominghuman.ai/artificial-intelligence-machine-learning-deep-learning-df6dd0af500e>

[60] <https://mektepp.com/ar-vr-nedir/>

[61] <http://www.yeniisfikirleri.net/nesnelerin-interneti-nedir-orneklere-nelerdir/>

[62] <https://www.shoppimon.com/2018/04/an-introduction-to-ar-and-vr-in-ecommerce/>

[63] <https://www.easetech.com/4-awesome-advancements-in-cloud-computing/>

[64] <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

[65] <https://sensortower.com/blog/top-apps-worldwide-q3-2019>

[66] <https://www.techocrush.com/hybrid-mobile-app/>

[67] <https://www.mediatick.com.tr/blog/5g-nedir-5g-ne-getirecek>