

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

Mobil Cihazlar İçin Uygulama Geliştirme

**Erdem ÇAMLIOĞLU
Mert MAHANOĞLU
Tunç KÜRTÜNCÜ**

GÜZ DÖNEMİ 1. YARIYIL ÖDEVİ

**DANIŞMAN
Dr. Serdar SOLAK**

Kocaeli, 2018

Mobil Cihazlar İçin Uygulama Geliştirme

Application Development For Mobile Devices

Erdem Çamlıoğlu, Mert Mahanoğlu, Tunç Kürtüncü

Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi

secamlioglu@gmail.com, mmahanoglu73@gmail.com, tunckurtuncu@gmail.com

Özet

Mobil cihazlarda kullanılmak üzere tasarlanmış uygulamaların geliştirilme aşamalarından ve ortamların kullanımlarından bahsedildi. Bunun üzerine yapılan araştırmalar sonucu elde edilen bulgulara dayanarak belirlenen geliştirme ortamları; Android Studio, Apache Cordova, Xamarin, Ionic, Swift, React Native olmuştur. Araştırma kapsamında bu ortamlarda uygulamaların nasıl geliştirilebileceği hakkında genel bir bilgi verilmiştir. Mobil uygulama sektöründen elde edilen bütçe göz önüne alındığında yapılan araştırmalarla da desteklenerek gelecekte bu sektörde trendin nereye doğru yöneldiğine karar verilmeye çalışılmıştır. Kullanım verileri ve uygulama verileri göz önüne alındığında yapılan çalışma kapsamında bu ortamlarla geliştirilen uygulamaların sektöre daha çok yer edindiği görülmüştür. Mobil uygulama geliştiricileri tarafından son yıllarda yapılan uygulamaların hangi kategorilere hizmet ettiği araştırılmış ve gelecekte bu kategorilerde neler olacağı ön görülmeye çalışılmıştır.

Anahtar kelimeler: Mobil Uygulama Geliştirme, Android, iOS, Xamarin, React, Swift, Apache Cordova, Ionic

Abstract

The development stages of applications designed for use in mobile devices and the use of media were mentioned. Development environments determined based on the findings of the research; Android Studio has been Apache Cordova, Xamarin, Ionic, Swift, React Native. Within the scope of the research, a general information is given about how the applications can be improved in these environments. Considering the budget derived from the mobile application sector, it has been supported by the researches and it has been tried to decide where the trend is going in this sector in the future. Considering the usage data and application data, it has been seen that the applications developed with these environments have gained more place in the sector. It has been tried to be predicted by the mobile application developers which categories are being presented and investigated and what might happen in these categories.

Keywords: Mobile Application Development, Android, iOS, Xamarin, React, Swift, Apache Cordova, Ionic

1. Giriş

Son yıllarda mobil uygulama sektörünün popülerleşmesi ve büyümesi ile oluşan mobil uygulama ihtiyacı gün geçtikçe artmaktadır. Bu uygulamaları tasarlayan ve yazılımlarını hazırlayan bireyler hangi cihazlara hangi ortamlardan ürün çıkarılabileceği konusunda çelişki yaşamaktadır. Yeni çıkan ortamların kullanılması veya hangi ortamların ne amaçla çıktıkları büyük bir sorun oluşturmaktadır. Bizim çalışmamız bu noktada mobil cihazlar için uygulama geliştiren yazılım geliştiricilerine yol göstermek için hazırlanmıştır. Bu noktada yaptığımız çalışmalar hangi ortamların hangi geliştiriciler için hazırlandığını ve hangi ortamların hangi cihazların yazılımlarının hazırlanmasında daha büyük rol oynadığını tespit etmeye çalışmıştır. Ortamların kimler için hazırlandığını tespit ettikten sonra yeni hazırlanan veya hazırlanmamış ortamların gelecekte nasıl olabileceğini elde ettiğimiz bulgular ile tahmin ederek tespit etmeye çalıştık. Akabinde oluşan uygulamaların son yıllarda hangi kategorilerde ağırlıkla yer aldığını tespit edip gelecekteki popüler uygulama havuzunu tahmin etmeye çalıştık. Daha sonra bu uygulamaların konularını, hedef kitesini ve şimdiki hedef kitleyi araştırma sonucunda bulduğumuz istatistik kaynaklarına göre tarif ettik.

Sonuç olarak bu araştırma kapsamında mobil uygulama geliştirmenin ne olduğundan, nasıl yapılabileceğinden ve gerekli ortamların ne olduğundan bahsedilmiştir. Ardından bu araştırma kapsamında gelecekte uygulamaların yer alacağı kategoriler ve kullanıcıların tercih edeceği uygulamalar tespit edilmeye çalışılmıştır.

2. Uygulama Geliştirme Aşamaları

Uygulama geliştirmeye başlamadan önce belli adımlar izlenmelidir ve bu adımların sırasına göre gidilmelidir. Adımları sırasına göre takip etmek programın iyi, kaliteli ve olabildiğince az maliyetli olmasını ve geliştirilecek olan bu uygulamanın eksiklerinin fark edilmesini sağlayacaktır. Aynı zamanda bu adımları izlemek, planlı ve programlı çalışıldığından, uygulamanın gidişatı hakkında geliştiriciyi bilgilendirecektir. [1], [2]

- **Fikir Üretmek:** Özgün bir fikir ortaya konulur. Bu fikrin geliştirilmiş olan diğer uygulamalardan daha ucuz veya daha iyi olması sağlanır.
- **Gereksinimleri Belirlemek:** Uygulamanın ihtiyacı olan gereksinimler ve çözümler tespit edilir. Geliştirilecek olan uygulamanın hitap edeceği kitle elde edilen verilere dayanarak belirlenir. Gelir hedefleri ve yöntemleri göz önüne alınır.
- **Konsept Çizim Yapmak:** Uygulamaya ait olan bir çerçeve belirlenir ve tasarım üzerinde çalışılır.
- **Konseptten Yararlanılarak Uygulama Geliştirmek:** Bu aşamada önceki aşamalar dikkate alınarak kontrol yapılır. Uygulamanın bir kısmı hazır durumdadır.
- **Gözden Geçirmek:** Geride kalmış olan aşamalar üzerinde değiştirilmesi gereken kısımlar değiştirilir. Uygulamanın market değeri hesaplanır. Hesaplanan bu market değeri geliştirme aşamasında harcanan bedelden düşük ise uygulamanın değiştirilmesi gerekmektedir.
- **Kâğıt Uygulama Yöntemi Kullanmak:** Geçmişteki aşamalarda belirlenmiş olan kitleye uygun 3-5 kullanıcı seçilir. Seçilen bu kullanıcıların oluşturulan kâğıt uygulamayı denemeleri sağlanır. Deneme esnasında kullanıcıların kâğıtlar üzerinde yaptıkları işlemler takip edilir. Son olarak kullanıcılar tarafından elde edilen bu veriler kaydedilir.
- **Tekrar ve Tekrar Gözden Geçirmek:** Kâğıt uygulama sonucunda kullanıcıların geri dönüşleri incelenir. Geri dönüşler bir sorun teşkil etmiyorsa uygulamanın kodlama işlemine başlanabilir.

3. Geliştirme Ortamları

Çalışma kapsamında içeriğe eklenen uygulama geliştirme ortamlarının farklı türde cihazlara hizmet eden ve en çok kullanılan ortamlar olmasına uğraşmıştır. Bu konuda kategorileştirme “Hybrid” ve “Native” ortamlar olarak yapılmıştır. [3]

3.1. Native Geliştirme Ortamları

Native geliştirme ortamı platform bazlı ortamlara denir. Seçilen sistemde çalışması üzere hazırlanan uygulamaların, hazırlanması için o sistemler için bulunan native ortamlar kullanılarak hazırlanması gerekir. Sisteme özel olan geliştirme ortamlarına “Native” geliştirme ortamı denir. [4]

3.1.1. Android Studio

Android Studio ilk olarak 2013 yılında Google’ın I/O etkinliğinde tanıtılmıştır. Yazılım için Java entegre geliştirme ortamı olan IntelliJ, IDEA’ya dayanır ve kod düzenleme, geliştirici araçlarını içinde barındırır. Gradle tabanlı bir derleme sistemine sahip olan Android Studio emülatör, kod şablonları ve Github entegrasyonu kullanarak uygulamamızı rahatça geliştirmemize yardımcı olur. Android Studio’yu tercih etmemizin başlıca sebeplerinden bazıları şunlardır; Esnek Gradle tabanlı bir sisteme sahip olan Android Studio hızlı ve zengin özellikli bir emülatöre sahiptir. Bu özellikleri bizlere uygulamamızı daha hızlı ve rahat bir şekilde geliştirmemize yardımcı olur. Bu özelliklerin yanında GitHub

entegrasyonu kullanarak uygulamamızda rahatça ekleme ya da çıkarma yapmamıza olanak sağlar. Performans, kullanılabilirlik, sürüm uyumluluğu ve diğer sorunları yakalamamız için Lint araçları kullanılır bu araçlar bize daha hızlı düzeltme yapmamıza olanak sağlar.

Bu avantajlarının yanında eksik yanları da vardır bu eksik yanlarından bahsetmek gerekirse onlarda şöyledir; Bir uygulama yazmak için genellikle diğer ortamlara oranla daha fazla satır yazmamız gerekir bu da bizlere zaman kaybı yaşatır. Karmaşık bir uygulama ya da animasyonlarla ilgili bir uygulama yapmaya çalıştığımızda ise diğer ortamlara göre kodlaması zorlaşmaktadır bu durum bizlerin istemediği durumlardan bir diğeridir. En büyük dezavantajlarından bir diğeri ise güvenlik açısından zayıf bir yapıya sahip olmasıdır. Bu güvenlik sorunu yazdığımız uygulamaları başkalarının da basit bir biçimde erişmesine olanak sağlar bu da oldukça kötü bir durumdur. En sinir bozan özelliği ise ekranın etrafında gereksiz çıkan reklamlardır. Bu reklamların dikkatimizi dağıtması bizleri yavaşlatır ve uygulamamızdaki eksikleri gözümüzden kaçırmamıza neden olabilir. [5], [6], [7]

3.1.2. Swift

Swift, güvenlik, performans ve tasarım kalıplarına modern bir yaklaşım kullanılarak dizayn edilmiş genel amaçlı bir programlama dilidir.

Swift iOS platformu için 2009’da gizlice geliştirilmeye başlanmış daha sonra 2011 yılında kullanıcılara açılmış resmi olarak WWDC2014 konferansında duyurulmuş bir yazılım dilidir. iOS için Swift dili açıklanmadan önce kullanılan Objective-C dilinin desteğinin devam edeceğini sunumda açıklayan Apple yöneticileri Swift’i öğrenmeye ve öğretmeye başladıkları andan itibaren yeni bir dil olmasından mütevellit yetenekleri ve yapabildikleri çoğu programcıyı Objective-C’den Swift’e geçirmek zorunda bırakmıştır. Swift’i özel kılan ise iOS platformu için kullanılabilecek en aktif ve en optimum araç olmasıdır. XCode bulunan bir MacOS’de geliştirilmesi gerekse de swift iOS cihazlar için kullanılan en yetenekli dildir.

Swiftin yazılımcıların yazdığı kodların derlenmesi alanında getirdiği yenilikler ve kullanım farklılıkları hata yapma olasılığını en aza indirir. Bellek otomatik olarak yönetiliyor ve yarı yazı tipi yazmanız bile gerekmiyor. Swift ayrıca diğer dillerden borç alır, örneğin Objective-C’den getirilen adlandırılmış parametreler, Swift’de APT’leri okumayı ve sürdürmeyi kolaylaştıran temiz bir sözdiziminde ifade edilir. Örnek verecek olursak Switch-Case yapısını eğer swiftte programlarsak bir Case bitişine break eklememize gerek yoktur. Case eğer devam edecekse bunu ayrıca belirtmemiz gerekir. Bu gibi kısayollar ve yeni geliştirmeler sonucunda Swift programcıların iOS’de uygulama geliştirmesi için tasarlanan en kompakt ve en yetenekli dil olmayı başarmıştır. Apple Swift’in Objective-C’ye göre çok daha hızlı olduğunu Swift’i duyurduğu konferansta belirtmişti fakat kesin olarak kanıtlayan bir bulgu olmamakla birlikte Swift’in ilk sürümüyle birlikte basit hazırlanan testlerde sonuçlar birbirine oldukça yakın. Teorik olarak bu duruma baktığımızda Objective-C’nin mantıksal çalışma şekli nedeniyle Swift’in biraz daha hızlı olması gayet normal. Eğer iOS projelerinizde sıkça Apple kütüphanesi kullanıyorsanız Swift projenizde performans açısından Objective-C’ye göre çok bir fark olmayacaktır. Bunun nedeni ise Apple kütüphane hazırlaması yaparken

genelde C, C++ ve Objective-C kullanıyor (2018). Apple kompleks dizi sıralama algoritmalarında Swift'in bazı yeteneklerinden ötürü Python ve Objective-C den daha hızlı olduğunu sunumda belirtmişti (Keynote 2011). WWDC 2016'da Apple, Swift ile kodlama nasıl öğretilir amacını taşıyan iPad özel uygulaması olan Swift Playgrounds'u tanıttı. Belirli sırada yerleştirilmiş ve çalıştırılan kodlar ile geri bildirim sağlayan bu uygulama 3 boyutlu video oyununu andıran bir arayüz içinde sunuldu. Bu çalışmaların ardından Swift programlamaya başlamak isteyen kullanıcılar kolay bir şekilde dile başlangıç yaptıktan sonra kendilerini geliştirebiliyorlar.

Bunlara rağmen Swift'in iOS ve MacOS in tabanında olması ve yeni bir dil olmasının etkisiyle daha modern çözümler sunması bir süre daha popülaritesini koruyacağı anlamına geliyor. TIOBE indeksine göre 2018 yılında en çok kullanılan 14. Programlama dili olmuştur. [8] Swift Stack Overflow Developer Survey 2015'in En Çok Sevilen Programlama Dili kategorisinde birinciliği, 2016 tarihinde ise ikinciliği kazandı. Apple Swift'in geliştirme aşamasında olduğunu ve kaynak kodlarının gerektiğinde dönüştürüleceğinden bahsetse de iOS platformu için en modern ve en yetenekli dil olmaya devam ediyor. Swift'in son sürümüne (Swift 4.2) Apple'ın GitHub deposundan erişilebilmektedir. [9], [10]

3.2. Hybrid Geliştirme Ortamları

Hybrid geliştirme ortamı yazıldığı dil önemsenmeden ve değişikliğe gerek kalmadan birçok platformda çalışan uygulamalara denir. Bu uygulamaların hazırlandığı ortamlar "Hybrid" geliştirme ortamıdır. [11]

3.2.1. Apache Cordova

Apache Cordova açık kaynaklı bir mobil geliştirme çerçevesidir. Eski adı PhoneGap olmasına karşın Apache Cordova ile aynıdır ancak, Apache Cordova'nın, Webkit'in Chrome'a güç veren bir motor olduğu gibi PhoneGap'i güçlendiren bir motor olduğunu söyleyebiliriz. Apache Cordova, platformlar arası geliştirme için HTML5, CSS3 ve JavaScript gibi standart web teknolojilerini kullanmanıza ve her bir mobil platformun yerel geliştirme dilinden kaçınmanıza olanak tanır... Başka bir deyişle, HTML, CSS, JS ile yazılmış bir web uygulamasını çalıştırmak için bir kapsayıcı görevi görür.

Apache Cordovayı diğer platformlardan ayıran özelliklere bakacak olursak bunların başında öğrenme kolaylığı gelmektedir. Ayrıca Apache Cordova en yaygın tarayıcılarda çalışır. Bu nedenle, işletme sahiplerinin bunu hızlı ve en uygun maliyetli şekilde geliştirmesini beraberinde getirmektedir. Bir diğer avantajı ise geliştiricilerin mobil uygulamaları daha güçlü yapmasına yardımcı olur: Apache Cordova, geliştiricilerin bir cihazın yerel kaynaklarının özelliklerini kullanarak bir uygulama geliştirmelerine yardımcı olur. Daha sonra uygulama, bir telefonun kamerası, ivmeölçer, pusula, mesajlar, kişiler, coğrafi konum, ağ, bildirimler (uyarı, ses, titreşim) ve ortam depolama gibi güçlü yerel özelliklere erişebilir. Apache Cordova, kullanıcı deneyiminden ödün vermeden bir işletme sahibinin hayatını kolaylaştırabilir. Birden çok platformda uyumludur: Apache Cordova, çoklu platform uygulama geliştirmesindeki varyasyon problemini çözüyor. Farklı platformlardaki farklılıkları ortadan kaldırır ve

kullanıcı arayüzü iOS veya Android gibi tüm mobil platformlarla tamamen eşleşir. Apache Cordova, Amazon Fire OS, Android, Blackberry, iOS, Ubuntu, Windows Phone, Windows 8, Tizen ve FirefoxOS platformlarını desteklemektedir. Apache Cordova diğer platformlara karşın daha az maliyetlidir. Apache Cordova, yüksek kaliteli ve birden fazla özellik setinin yanı sıra geliştirme süresini azaltmak için gerekli araçları sağlaması nedeniyle bir mobil uygulama oluşturmak için uygun maliyetli bir teknolojidir. Bir geliştirici Apache Cordova uygulamasını oluşturduktan sonra, her yere dağıtabilir ve tüm platformlara gönderebilir. Bu nedenle, her platform için farklı programlama dili kullanarak yerel uygulamaları oluşturmanız için çok fazla para, çaba ve zaman tasarrufu sağlayabilirsiniz. Apache Cordova'nın çok geniş bir eklenti bölümü vardır. Geliştiricilerin bir uygulamanın daha fazla işlevselliğini genişletmesini sağlayan bir eklenti mimarisi modelini destekler. Kapsamlı mimarisi, yerel özellikleri, telefon cihazının tam potansiyelini kullanmanıza yardımcı olur ve bulut servisine erişim özelliğini sunar. Bu, teknoloji bildirimlerini geliştirmeler, öğrenciler ve örnekler ile zorlayabilirsiniz. Apache Cordova uygulamalarının belirgin faydaları, daha hızlı yinelenen daha az ve daha basit kod kullanmasıdır. Tüm işletme sahipleri için karlı bir uygulamadır, çünkü tüm mobil platformlarda çalışmak için bir uygulama geliştirmek için daha az çaba gerekmektedir. PhoneGap, muazzam özelliklere ve mükemmel çerçeveye sahip, kapsamlı bir mobil uygulama geliştiren en popüler teknolojidir. Apache Cordova'nın en önemli avantajı ise güvenliğe ve kullanıcıya önem vermesidir. Cordova topluluğu, bu bağlamda, dünyanın herhangi bir yerinde yetki alanlarına ve yasal gereksinimlere uymayı taahhüt etmektedir. Topluluk, Beyaz Listeye (harici alanlara erişim), Iframe'lere (üçüncü taraf sitelerin reklam içeriğini güvenli bir şekilde göstermek için) ve Geri Arama Kimliği Mekanizması'na (arayanlarla ilgili bilgileri almaya), Sertifika Sabitleme'ye ilişkin güvenlik yönergelerini güncellemeyi sürdürür. [12], [13], [14]

3.2.2. Xamarin

Xamarin doğal ve çapraz platform kavramlarını oldukça doğru bir yaklaşımla bir araya getiren ve C# diliyle mobil uygulama geliştirmenizi sağlayan tek platform. Geliştiricilere sunduğu Xamarin Forms arabirimi ile hem kullanım şekline ve kurallarına aşina oldukları bir ortam sağlar hem de yazılan kodu ortaklaştırıp, uygulamanızın eş zamanlı olarak Android, iOS ve Windows platformları için hazır durumda olmasını sağlar. Bu yönüyle hem kaynak maliyetlerini ciddi oranda düşürür hem de üretim süresinin kısaltılmasına yardımcı olur. Platformlar arası (iOS, Android, WindowsPhone) uygulamaları üretebilmektedir. C# diliyle kodu hazırlanır ve mobil uygulama geliştirilmeye başlanır. Yazılan kod Xamarin'in, hybrid bir ortam olmasından dolayı diğer platformlarda kullanılmasına olanak sağlar. Kendi kütüphanenizi oluşturmanıza izin verir. Genelde Xamarin ile iOS ve Android uygulaması geliştirildiği yönünde bir algı vardır, ancak bu platformda iOS ve Android uygulamalarına ek olarak Universal Windows Platform (UWP), Apple Watch, Mac, Android Wear uygulamaları da geliştirilebilir. Xamarin'in en önemli sayılabilecek ve onu diğer hybrid kütüphanelerden ayıran özelliği doğal çıktı üretebilmesidir. Xamarin ile geliştirdiğiniz bütün uygulamalar, ilgilendiğiniz platformun ara yüz kontrollerini ve API'larını

kullanabilmektedir. Xamarin, geliştiricilere ortak IDE, ortak kod, ortak dil ve kolay bakım imkanları sağlar. Windows kullanıcıları Xamarin kullanıcıları için ücretsiz olan Visual Studio Developer Edition programını kullanabilirler. [15], [16], [17]

3.2.3. React Native

React Native Sadece Android ve iOS platformları için desteklenebilir olan JavaScript'in yardımıyla gerçek ve heyecan verici mobil uygulamalar oluşturmanıza yardımcı olur. React Native, büyük bir popülerlik buldu ve Facebook tarafından desteklendi, bugün çok büyük bir topluluk desteğine sahip. React Native, uzun süre favori AngularJS'ye (AngularJS Google tarafından desteklenen, dünya genelinde yazılımcılar tarafından katkı sağlanan açık kaynak kodlu web uygulama çatısıdır.) büyük bir rekabet kazandıran ReactJS'nin (ReactJS ise, kullanıcı arayüzleri oluşturmak için açık kaynaklı bir JavaScript kütüphanesidir.) üzerine inşa edilmiştir. Her ne kadar aşağıda belirtilen ReactJS ve React Native arasında bazı benzerlikler ve farklılıklar olsa da React Native, JavaScript kodunu oluşturmak için UI (User Interface) bileşenlerini bir düzene koyan çerçevedir. Yerel görünüm ve izlenim ile bir mobil uygulama oluşturmak için hem iOS hem de Android platformları için bir dizi bileşene sahiptir. Ancak hem React Native hem ReactJS, arayüzler tasarlama dışında, aynı tasarım ilkelerini kullanarak Facebook tarafından geliştirilmiştir. React Native iOS veya React Native Android uygulamaları ve web uygulamaları oluşturmak için aynı kodu kullandığından, yalnızca HTML, CSS ve JavaScript'i bilmeniz gerekmektedir. React Native bileşenlerini mevcut uygulamanın koduna ekleyebilir veya Cordova tabanlı kodu eklentinin yardımıyla yeniden kullanabilirsiniz. Bununla birlikte, mevcut uygulamanız Cordova ve Ionic koduyla oluşturulmalıdır React Native üzerinden geliştirme yapmak nispeten basit, hızlı ve verimlidir ve React Native, Android'e özgü olan Java'yı veya iOS'un Swift'ini öğrenmeye gerek olmadığı için JavaScript konusunda uzman olan geliştiriciler için harika bir seçimdir.

Ayrıca React Native Framework yakın gelecekte Windows veya TVOS gibi diğer platformlarla uyumlu olabilecek açık kaynaklı bir çerçevedir. İçinde bulunan bu bileşenleri hem Android hem de iOS uygulamaları oluşturmak için yeniden kullanabilirsiniz. Ayrıca React Native Framework kullanmak, geniş bir pazara odaklanmak ve kâr elde etmek arasındaki boşluğu doldurmaktadır. iOS kullanıcılarının sayısı, Android kullanıcılarına kıyasla daha az olduğu ve daha fazla kâr sağladığından, birçok endüstri, iOS uygulamaları oluşturarak veya yalnızca Android uygulamalarıyla kullanıcı gücü oluşturmaya daha fazla kar etmeyi önemsenenin ikilemine giriyor ve bu React Native'e popülerlik katıyor. React Native çerçevesinin son yıllarda popülerlik kazanmasının bir başka nedeni de çerçevenin bir seferde hem platformlara hitap etmesine hem de mobil geliştiriciler için iOS veya Android geliştirmeyi seçmeye mecbur kılmasıdır. Bu nedenle Facebook, Instagram, Skype, Airbnb, Tesla, Walmart, Baidu Mobile, Bloomberg, Uber, EATS, Vogue gibi dev şirketler gözlerini bu tarafa çevirdi. React Native Framework'ü neden kullandığımız konusuna gelecek olursak kullanmamızın ilk ve en önemli nedeni, geliştiricinin React Native Mobile Uygulamalarının performansından yararlanmak için bir kod seti, yani JavaScript kodu yazması gerekmesidir. React Native

kusursuz olmamakla birlikte artıları eksileri vardır. Başlangıçta iOS geliştirme için icat edilen React Native Framework, popüleritesi ve başarısından dolayı bundan sonra Facebook, Android platformu için de destek sağlamaya karar verdi. Böylece çapraz platform kullanımı sağladı. Ayrıca diğer karma veya mobil web uygulamalarına karşın hem yazılı sistemlerde hem de hiçbir gecikme yaşamayan platformlarda da aynı işlev görmesini sağlayan yerel olarak yazılmış kodlarla derlenmesi kendisini diğer uygulamalardan üstün kılmıştır. Hızlı geliştirme, bileşenleri yeniden kullanma ve bakımı kolaylaştırmasından dolayı zaman kazanma ve kolay kullanım özelliğini kendisine ilke edinmiştir. Ancak React Native Framework önceden paketlenmiş öğelerdeki tüm teknik hataları saptayamıyor ve bunun için profesyonel bir geliştiriciye ihtiyaç duymanız gerekiyor. Ayrıca, projeleriniz geliştiricinin elinde olduğundan geliştirici istediği projeyi istediği zaman öldürme hakkına sahiptir bu da size ayrı bir olumsuz etki yaratmaktadır.

Sonuç olarak, belli başlı eksileri olmasına karşın, React Native Framework, neredeyse tüm React Native Development Company'nin mobil uygulamaları geliştirmesi için bir numaralı seçenek haline gelmek için tüm testleri ve engelleri aşmaktadır. React yerel geliştiricisi, React Native uygulamalarını geliştirirken sadece çevikliği ve kolaylığı bulmakla kalmaz, aynı zamanda işlem sırasında yüksek verimlilik ve düşük maliyet ve iş yükü ile karşılaşır. Süreci daha heyecanlı kılan, uygulamayı oluşturmak için hem iOS hem de Android platformlarına odaklanmanıza gerek olmamasıdır. Bu nedenle, dünyadaki birçok muazzam şirket, günlük olarak kullandığımız heyecan verici uygulamalar oluşturmak için React Native Framework'e güvenmektedir. [18], [19]

3.2.4. Buildfire

Buildfire işletmeler, organizasyonlar, bireyler, satıcılar ve geliştiriciler için lider mobil uygulama geliştirme platformu seçimidir ve kodlama gerektirmez. Buna ek olarak, "kullanıma hazır" özellikleri geliştirme veya açık geliştirme portalı aracılığıyla tamamen yeni ve özel işlevler ekleyebilme olanağı sunmaktadır. Peki, Buildfire'ı diğer mobil uygulama geliştirme platformlarından ayıran nedir?

Kullanımının diğer platformlardan basit ve kolay olması bu nedenlerden ilkidir. En önemli farkı ise diğer hiçbir platform tıklama ve düzenleme deneyiminin sadeliğini özel geliştirme becerisiyle birleştirmez fakat Buildfire bunu başarıyor. Buildfire sayesinde geliştirme süresi oldukça azalır. Tüm şartlar uygun olduğu durumda haftalar alacak bir proje sadece günler sürebilir. Çok fazla para harcamanıza gerek yoktur, çünkü sadece aylık abonelikle hizmete erişiminiz mümkündür; Kısacası, maliyetler azalır. Bu aracı kullanmak için büyük bir takım insana ihtiyacınız yoktur ve karmaşık altyapıları düşünmeye gerek yoktur.

Ayrıca seçim yapılabilen birçok eklenti ile gereken eklentiler seçilebiliyor ve bu eklentiler kodun içine girmeyi sağlıyor. Buildfire'ın içindeki bu eklentiler oluşturulmuş olan Buildfire uygulamasına entegre edilebilen kapsüllenmiş kod gruplarıdır. Her bir eklenti 3 ana bölümden oluşur. Bunlar, kontrol, widget ve servisler ve kontrol panelinde bulunan genellikle uygulamada nelerin gösterileceğinin içeriğini ve tasarımını

yöneten kısımdır. Eklentide bulunan servisler, verileri yayınlamak veya cihaz üzerindeki bluetooth işlevlerine erişmek gibi işlevleri geliştirmek için Kontrol ve Widget tarafından kullanılan işlevlerdir.

Örnek verecek olursak bir stok kullanılabilirlik eklentisi oluşturuyorsanız, kontrol bölümü, envanterdeki tüm öğelerin bir listesini ve tarafınızdan düzenlenebilir niteliklerini içerir. Widget bölümünüz ilgilendiğiniz öğeleri bulmak ve envantere salt okunur bir görünüm sağlamak için bir arama özelliğine sahip olacaktır. Bu durumda servisler, verilerin kaydedildiği DataStore olacaktır.

Buildfire'in dezavantajları da bulunmaktadır:

Bu uygulama tamamen native bir uygulama değildir ve bu nedenle performans etkilenebilir. Uygulama tasarımı veya işlevselliklerinin toplam kontrolü yoktur. "App Maker" türündeki platformlar çok popüler olmasına rağmen, bunlar örneğin küçük işletmeler, örneğin: restoranlar, gayrimenkuller, giyim mağazaları vb. Üzerinde yoğunlaşmaktadır, bu nedenle bu tür hizmetler tüm şirketler için uygun değildir. Çeşitli API'ları yönetmeyi veya bunlarla ilgilenmeyi ve aynı web uygulamasından büyük miktarlarda veri yönetmeyi gerektirir. Uygulamanın performansı en uygun şekilde olmalıdır. Çok kişiselleştirilmiş bir arayüze ihtiyaç vardır. Bu tip durumlar sağlanmıyor ise kullanımından kaçınılmalıdır. Son olarak ise Buildfire arabirimi tartışmalı bir konu olmasına rağmen, kolayca düzenlenebileceğinden, her zaman onu tamamen kontrol edemezsiniz, bu nedenle karmaşık animasyonlara sahip arayüzler bu araç ile yapılamaz. Aynı zamanda buildfire.js Buildfire Platformunda özel eklentiler oluşturmanıza izin veren bir Javascript Framework'dür. Standart HTML5, CSS3, Javascript ve buildfire.js kullanarak, bu Uygulama ve Kontrol Paneli'nde yaşayan özel kullanıcı arayüzü ve mantığı oluşturabilirsiniz. Buildfire.js, aksi halde sizin için erişilemeyecek uygulama ve yerel cihaz işlevlerine erişmenizi sağlar.

Buildfire hybrid ve native olmak üzere 2 çeşit geliştirme uygulaması bulundurur. Bu hybrid ve native uygulamalar arasında bazı farklar bulunmaktadır:

Native uygulamalar, native geliştirme dilleri ve telefonun frameworkleri kullanılarak geliştirilen uygulamalardır. Bu size en fazla kontrolü verirken, aynı zamanda en pahalı olandır ve zaman alıcıdır. Özellikle, iki takımı dikkate alırsanız iOS için bir tane, Android için bir tane gerekir.

Buildfire'in hybrid uygulamaları ise aynı yerel bileşenlerde oluşturulan uygulamalardır. Ancak hem iOS hem de Android platformlarının geliştirilmesini birleştirmek için HTML5, CSS3 ve Javascript gibi frameworkleri kullanıyoruz. Bu, teknik karmaşıklığa indirgenerek, bulunması zor ve çok pahalı olan çok özel geliştiricilere olan ihtiyacı ortadan kaldırır. Web geliştirmeyi biliyorsanız, Buildfire ile özel uygulamalar oluşturabilirsiniz. [20], [21]

3.2.5. Ionic

Ionic Framework, yazacağınız tek bir kod ile Android, iOS, Windows Mobile gibi tüm platformlarda uygulama yazmanızı sağlayan bir sistemdir. Ionic AngularJS, Cordova, HTML5 ve CSS kullanarak tek bir koddan çapraz platform uygulama geliştirmenizi sağlar. Ionic sayesinde doğal uygulamalar kadar

hızlı ve yetenekli uygulamalar geliştirilebilir. Ionic tasarlanırken doğal uygulamalar kadar hızlı uygulama geliştirebilmek her zaman göz önünde tutulmuştur.

Tek bir kod ile birçok uygulama geliştirmenizi sağlar farklı platformların gerektirdiği farklı programlama dilleri ve farklı SDK'ler ile uğraşmanıza gerek kalmaz. Uygulamanızdaki hataları giderdiğinizde ya da yeni özellikler eklediğinizde bunu tek bir yerde yaparsınız. Ionic sizin için bu değişiklikleri tüm platformlar için oluşturur. Android ve iOS gibi farklı platformlarda uygulama sahibi olmak daha geniş kitleye ulaşmak demektir. HTML/CSS ve AngularJS ile uygulama geliştirebilirsiniz. Açık kaynaklı ve ücretsiz olması birçok kişinin tercih etmesini sağlar. Ionic performansı daima öncelikleri arasında tutuyor. Bir uygulama geliştiricisi için ise farklı platformlarda uygulama geliştirmek ve bu platformlardaki güncellemeleri takip etmek zordur Ionic ile fazla para, zaman ve emek harcamadan tüm platformlarda uygulama geliştirebilirsiniz. Ionic ile beraber gelen araçlar sayesinde istediğiniz uygulamanın hızlı bir şekilde prototipine ulaşırsınız ve doğal uygulamalara göre daha hızlı uygulamanızı geliştirirsiniz. Uygulamanızı Chrome tarayıcısı üzerinden Android ve iOS platformlarının emülatörleri ile uğraşmaksızın test edip görüntüleyebilirsiniz. [22], [23], [24]

4. Dünya Çapında Trend

Kullanıcısı günden güne artan mobil teknoloji sektörü gittikçe büyümekte ve değişmektedir. Mobil cihazların devrimi ile mobil uygulama geliştirmedeki eğilimler zamanla değişti. Şimdi, 3D grafikleri, yapay zekâ (AI) ve hayatımızı etkileyen diğer birçok öğeyi biliyoruz. [25], [26]

4.1. Sanal Gerçeklik ve Arttırılmış Gerçeklik

2017 yılında, sadece Arttırılmış ve Sanal Gerçeklik nedeniyle eğlence ve oyun endüstrisinde büyük bir değişime tanık olduk. AR (Augmented Reality) teknolojisine dayanan ve herkesi bağlayan Pokémon Go'nun çılgınlığını gördük. Diğer taraftan, VR özellikli cihazlar aracılığıyla bir kullanıcının ilgisini çekmiştir: Samsung Gear VR

Aynı yıl, endüstri uzmanları, teknik değişime uyum sağladı ve onu, Perakende, Tekstil, Emlak, Sağlık, Turizm ve benzeri gibi diğer sektörlerde uyguladı. Örneğin, önde gelen ev mobilyası şirketlerinden biri olan IKEA'yı keşfedebilirsiniz. Firma, "IKEA yer uygulamasında" AR teknolojisini kullanarak, odanızı taramanıza ve mobilyalarınızı bulunduğunuz yere konumlandırmanıza olanak tanıyarak, mobilyaların tam görünümünü sergilemek için. Mobilyaların iç mekanlarınızla uyumluluğuna dair varsayımlarınızı açık hale getirir.

4.2. IoT (Internet of Things (Nesnelerin İnterneti))

Akıllı şeyler için artan talep, artan sayıda IoT uygulamasına neden oluyor ve bir mobil telefon, akıllı sistemler ve insanlar arasında köprü görevi görürken, bir uygulama bağlı cihazları yönetmek için ana araçtır.

IoT uygulamaları şunları yapmanızı sağlar:

- Cihaz ve sensör okumalarını izleme,
- Verileri analiz etme, grafikler ve raporlar oluşturma,
- Cihazları yönetme (açma / kapatma, açma / kapatma, alarmları tetikleme ve daha fazlası).

Müşteri odaklı çözümler (akıllı ev aletleri, giyilebilir cihazlar, trafik sensörleri) ve kurumsal düzeyde yönetim (petrol ve gaz

şebekeleri, demiryolu hatları, otonom otomobiller) için eşit derecede faydalıdır. IoT, sağlık ve eğitimden imalat ve ulaştırmaya kadar birçok sektörde yaygınlaşmaktadır. Giyilebilir cihazlar için uygulama geliştirme, şimdiye kadarki en önemli alan olan ana temsilciler, akıllı saatler, fitbitler, akıllı sütunlar ve ev asistanları için uygulamalardır.

4.3. Blockchain ve Mobil Ödeme Uygulamaları

Akıllı cihazların gelişmesiyle birlikte mobil ödeme yöntemleri açığa çıktı. Bununla beraber bu ödemelerin ve hesapların bir güvenliğe ihtiyacı vardı ve burada da blockchain uygulamaları ortaya çıktı. Bu teknoloji, mobil platformunuzun korunmasını geliştirmenize ve müşteri sadakatini kazanmanıza olanak tanıyan güvenlik konusunda özel bir yaklaşım sunar. App Store'da ve Google Play'de, blockchain ile birleştirilmiş pek çok uygulama bulabilirsiniz. Bunlar çoğunlukla e-cüzdanlar, haber uygulamaları ve döviz çeviricilerdir.

4.4. Yapay Zekâ ve Makine Öğrenmesi

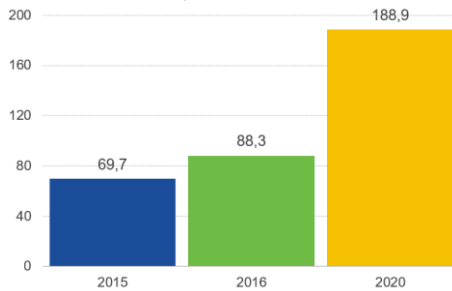
Apple, geliştiricilerin ses asistanı Siri'yi uygulamalarına entegre etmelerine olanak tanıdığından, mobil endüstri trendleri 2017'den bu yana yapay zekaya odaklandı. Bu, yapay zekanın çeşitli alanlara benimsenmesini ve entegrasyonunu teşvik etti. 2018'de, AI mobil uygulama geliştirme süreçlerinde devrim yaratacak.

Daha fazla uygulamanın sesli komutları tanıyacağını, daha akıllı ve karar verebildiğini ve kullanıcı davranışlarını tahmin etmek için daha özelleştirilmiş olacağı öngörülmekle beraber cihazlar mümkün olduğunca çok sayıda AI ögesiyle paketleyecek akıllı telefon geliştiricileri arasında şiddetli bir rekabet göreceğiz.

5. Sonuçlar

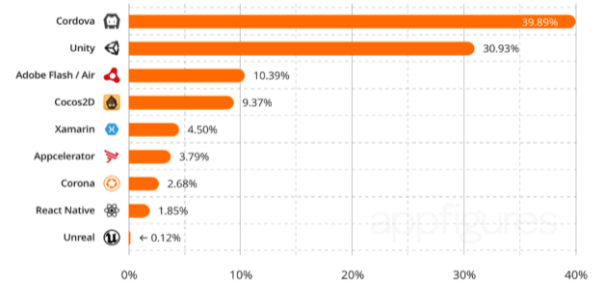
Yapılan araştırmalar sonucunda elde edilen bulgulara göre dünya çapında uygulama geliştirme sektöründe yapılan harcamaların 189 milyar dolara varacağı (Tablo 1) ve oluşturulan ürünlerin mobil sektörde günden güne arttığı gözlemlenmiştir.

Tablo 1: Mobil uygulama sektöründe güncel gelir durumları (milyar dolar) [27]



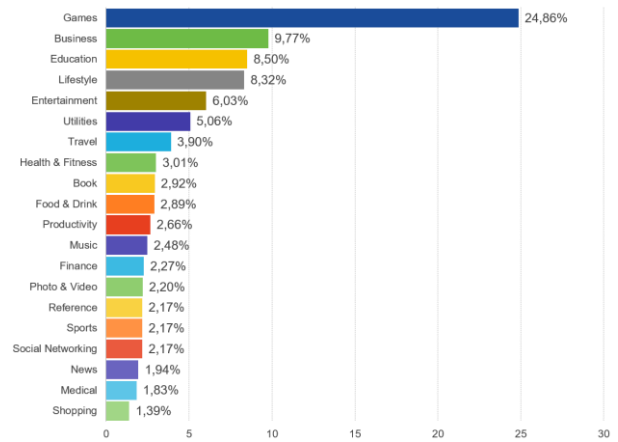
Mobil uygulamaların geliştirilmesi için şu an kullanılan yöntemlere değinilmiş ve yol gösterilmiştir. Hybrid ve native uygulama geliştirme farklarından bahsedildikten sonra hangi tip geliştirme için hangi uygulamanın kullanılacağı hazırlanan içerikte, dünya çapında oluşturulmuş olan resmi araştırma sonuçlarına göre belirlenmiştir. Bunun ardından araştırma sonuçlarında tespit edilen (Tablo 2) native olmayan geliştirme ortamlarına olan ihtiyaç artmıştır.

Tablo 2: Popüler Non-Native (Native olmayan) geliştirme ortamları [28]



Dünya çapında elde edilen grafiklere ve araştırmalara göre kullanılan mobil uygulamaların en çok iOS ve Android tabanlı kullanıcılar tarafından kullanıldığına karar verilmiştir.

Tablo 3: Eylül 2018 App Store en popüler kategoriler [29]



İstatistikler göz önüne alındığında, hybrid geliştirme ortamlarının kullanılma verilerinde artış gözlenmektedir. Araştırma kapsamında en çok uygulamaların kullanıldığı kategoriler Tablo 3'de belirlenmiştir. Bu kategorilerde bulunan uygulamaların geliştirildikleri teknolojiler ve gelecek trendi bu şekilde olacak gibi gözüküyor. Kullanıcıların hizmet ağırlıklı uygulamalara daha fazla değer verdiği ve zaman ayırdığı gözlenmiştir. Akıllı cihazların gelişmesi ile birlikte AI, AR, VR, IOT teknolojilerine entegre olan ve bu teknolojileri kullandıran mobil uygulamalara, tür olarak verilen önem ve harcanan bütçe artışı gözlenmiştir. Bunun sonucunda bu türlere mobil uygulama sektöründe olan ihtiyaç bariz bir şekilde görülmektedir.

6. Kaynaklar

- [1] www.redk.us
- [2] <https://buildfire.com/how-to-create-a-mobile-app/>
- [3] <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/>
- [4] <https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>
- [5] <https://searchmobilecomputing.techtarget.com/definition/Android-Studio>
- [6] <https://developer.android.com/studio/intro/>
- [7] <https://www.credosystemz.com/training-in-chennai/best-android-training-in-chennai/>
- [8] <https://www.tiobe.com/tiobe-index/>
- [9] <https://swift.org/about/#swiftorg-and-open-source>
- [10] [https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))
- [11] <https://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app>
- [12] <https://cordova.apache.org/docs/en/3.5.0/guide/overview/index.html>
- [13] <http://www.vensi.com/benefits-of-phonegap-cross-platform-mobile-app-development/>
- [14] <https://www.rootinfosol.com/top-5-benefits-of-apache-cordova-phonegap-app-development>
- [15] <https://www.quora.com/What-is-Xamarin-What-is-its-need>
- [16] <https://www.techjini.com/blog/xamarin-overview/>
- [17] <https://developer.telerik.com/topics/mobile-development/what-is-xamarin-forms/>
- [18] <https://medium.com/@thinkwik/react-native-what-is-it-and-why-is-it-used-b132c3581df>
- [19] <https://facebook.github.io/react-native/>
- [20] <https://buildfire.com/faq/>
- [21] <https://info.sundevs.com/blog/services-buildfire-mobile-apps>
- [22] <https://hackernoon.com/what-is-ionic-c1da6eab0d8a>
- [23] <https://ionicframework.com/docs/>
- [24] <http://www.how2lab.com/programming/app/what-is-ionic.php>
- [25] <https://www.sam-solutions.com/blog/top-mobile-technology-trends-defining-the-industry-in-2018/>
- [26] <http://www.rginfotech.com/mobile-app-development-trends-2018/>
- [27] <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/>
- [28] <https://appfigures.com/>
- [29] <https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>