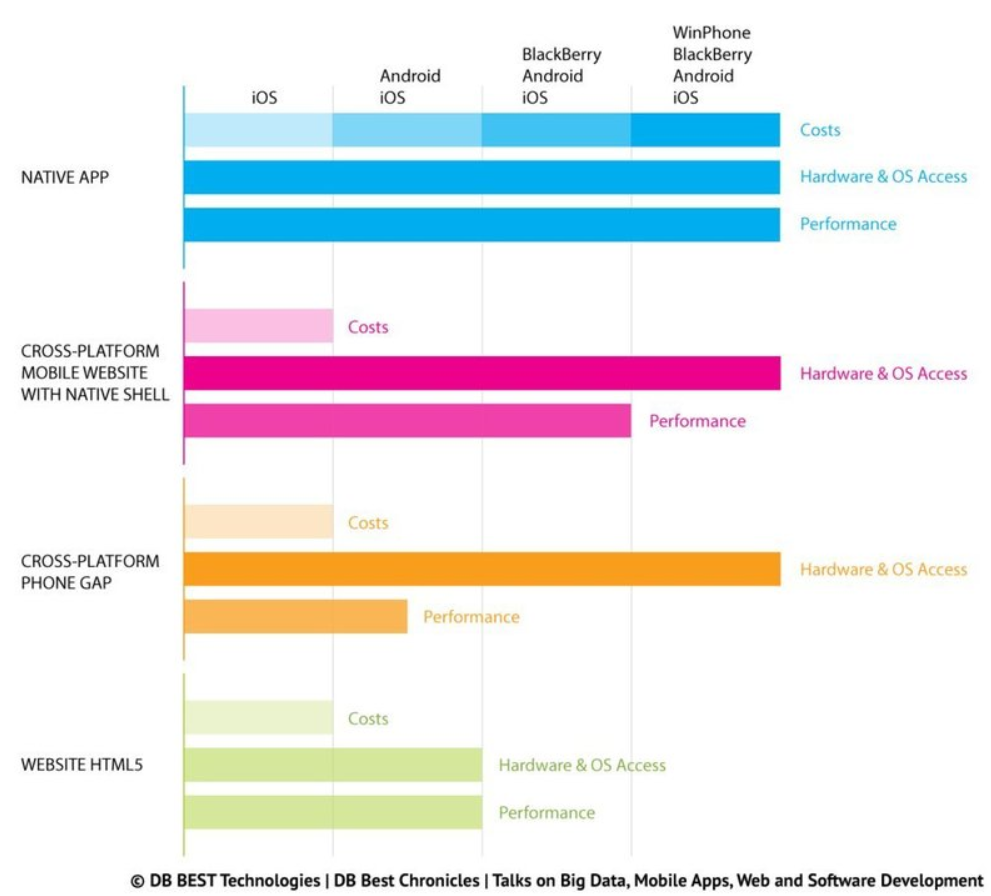
1. Native uygulamaların farklılıkları;

* **Programlama Dili ve IDE:** Android ile uygulama yazabilmek için Java ya da Kotlin dillerinden birini bilmek gerekir ve IDE olarak Android Studio kullanılmalıdır.Burada dilin kolaylığı konusunda Swift Javadan daha sade bir syntax a sahiptir. Ancak Javanın popülerliği ise oldukça ortadadır. IOS için geliştirme yapılacaksa Objective-C ya da Swift dillerinden birini bilmek gerekir ve IDE olarak XCode kullanılabilir.
* **Test:** Android ve IOS da uygulamanın çalışıp çalışmamasını test etmek developer için oldukça önemlidir. Android uygulamaları için Android emülatorler kullanılır ve oldukça başarılıdırlar. Bu sanal cihazların içerisinde sanal CPU, GPU ları ayrıca Harici, Dahili depolama alanları vardır ve gerçek cihazlara çok yakındır. Ancak IOS simulatorleri IOS cihazların etkili ya da doğru birer örnekleri değillerdir ve çok başarılı sonuçlar vermezler. Başarılı bir test için gerçek bir cihazda test edilmesi şarttır.
* **Tasarım:** Adroid de tasarım XML dosyaları ile IOS da ise XIB dosyları ile yapılmaktadır. Buradaki yapı genel olarak benzerdir. İster sürekle bırak(drag-drop) yöntemiyle ister tek tek yazarak sayfa tasarımı yapılabilir. IOS da animasyonlar ve tasarım kalitesi Android e göre daha gelişmiştir. Google ise bu sorunu Material Design olarak bilinen bir uygulama tasarım trendi oluşturarak çözmeye çalıştı ancak yine de IOS un gerisinde kalmıştır. IOS kullanıcıya daha görsel bir deneyime odaklanırken, Android ise daha çok donanıma odaklanıyor.
  + **Ekran boyutu ve çözünürlük:** IOS cihazlarda genel olarak cihazların boyutunda farklılık yoktur. IPhone ve IPad boyutları cihazlarda çeşitlilik göstermemektedir. Responsive tasarım yaparken IOS da çok fazla cihaz tipi olmadığı için bu oldukça kolay ve uğraştırıcı değildir. Ancak Android e farklı markalar cihaz ürettiği için çok sayıda cihaz boyutu ve çözünürlüğü çeşidi vardır. Responsive tasarım yaparken bunlar göz önünde bulundurularak geniş düşünmek gereklidir.
  + **Navigation:** Anroid navigation için drawer pattern kullanmaktadır. (Sol taraftan açılan menu). Bu ekranın sol tarafını görüntü olarak biraz kaplamaktadır. IOS ise navigation için tab bar pattern kullanmaktadır.(Ekranın altında bulunan ve genellikle saga sola dürükleyerek hızlıca geçiş yapılan menü)
  + **Geri Tuşu:** Android de varsayılan olarak bir geri tuşu bulunmaktadır. Ancak IOS da böyle bir geliştirme yoktur. Ancak geliştiricinin uygula içinde bir önceki menüye ya da bir önceki işleme dönmek için geri tuşu ya da bu işlevi yerine getiren herhangi bir tasarımı düşünüp uygulama içinde uygun bir yere yerleştirmesi gerekmektedir. Bu önemli bir ayırt edici özelliktir.
* **Tercihler ve Erişim İzinleri:** Mobil uygulamada en önemli özelliklerden biri de uygulama izinleridir.Uygulama içerisinde kamerayı, konumu ya da cihaza ait herhangi bir özelliği kullanabilmek için kullanıcıdan izin istenmesi gerekmektedir. IOS da izinleri yönetmek daha kolay ve derli topludur. IOS da izinlerden bir üst erişime izin verince alttakilere de otomatik izin verilir. (parent-child ilişkisi). iOS'un uygulama izin sistemi, kullanıcıya izinlerin tüm teknik bilgilerinin ayrıntılı bir şekilde anlaşılmasını sağladığı için iyi organize edilmiştir. Android de ise izinler tek tek verilir. Android de bir uygulamanın sadece bir kısmında erişilmek istenen izin uygulama yüklendiğinde ya da başında kullanıcıya sorulur. Kullanıcı bu izni vermek istemediği zaman uygulama bileşenleri tamamen yüklemeyebilir bu da uygulamayı kullanamamaya sebep olabilir ya da uygulama kullanılmak istenirse bu izni onaylamaya kullanıcı zorlanmış olur. Ancak IOS da bu daha farklıdır. Uygulama yüklenir ve kullanıcı o izine erişmek isteyen bileşeni kullanmak istediğinde kullanıcıya sorulur. Onaylanmazsa uygulama o izine ihtiyaç duyan bileşen harici kullanılabilir. Bu oldukça önemli bir avantajdır. Uygulama izinlerine erişirken Android ‘Uygulama kameraya erişmek istiyor.’ şeklinde uyarı verirken IOS ise ‘Uygulama fotoğraf çekmek için kameraya erişmek istiyor.’ şeklinde uyarı vererek sebebini de kullanıcıya açıklıyor.
* **Harita:** Android için Google Maps, IOS da ise Apple Maps vardır. Google Maps çok daha popular olduğu için bir çok IOS uygulaması da Apple Maps yerine Google Maps kullanmayı tercih etmektedir. Google Maps daha fazla gelişmiş olduğu için kullanıcıların tercihi olmuştur.(Daha detaylı zoom, çok az sayıda etiket gösterimi, ses ile yönledirme vs.) IOS geliştiricileri bunu göz önünde bulundurmalıdır.
* **Maliyet ve Para Kazanma:** IOS da uygulama yayınlamak Android e göre daha pahalıdır. IOS ya da Android de uygulama yayınlamak için geliştirici lisansı almak gerekirve fiyatlandırmalar ise IOS tarafında Android de göre daha fazladır. Ayrıca IOS pahalı olmasına rağmen lisans süresi Android e göre daha kısadır. Ancak uygulamadan para kazanmak istenirse IOS dan daha fazla gelir elde edilir.Android de genellikle uygulama içine gömülü reklamlardan para kazanılır dolayısıyla bir çok Android uygulaması ücretsizdir. IOS da ise uygulamanın kendisinden para kazanmaya odaklandığı için uygulama içi satın almalardan para kazanılır.
* **Uygulama Yayınlama:** Android de uygulama yayınlamak oldukça kolaydır. Android uygulamayı Google Play Store a koymak için herhangi bir kısıtlaması yoktur. Ancak IOS Apple Store a uygulama koymak için belirli kısıtlamaları vardır. Uygulamanın sahte olup olmaması, güvenlik politikasına uyup uymaması gibi. Dolayısıyla IOS a uygulama yükledikten sonra 1-2 hafta alabilir. Ancak Android de çok fazla sahte uygulama vardır, bu durum IOS da yoktur.

Genel olarak IOS ve Android uygulamaların benzerlikleri yukarıda siyah belirtilen başlıklar sayılabilir. Her iki mobil uygulaması da programlanmalı, test edilmeli, uygulama tasarımı yapılmalı, mobil izinler verilmeli, user-friendly olmalı, her ikisi de kendi pazarlarına uygun olmalı ve yayınlanabilmelidir. Her ne kadar Teknik kısımları farklı olsa da temelde aynı işlevleri yapmalıdırlar.

1. Cross platformlar son dönemde oldukça popular olmaya başladı. Bunun artılarını şöyle sıralayabiliriz. Tekrar kullanılabilir kod mantığı devreye girer. Native uygulamalarda aynı uygulamayı Android, IOS ve Windows işletim sistemleri için ayrı ayrı yasmak gerekir. Ancak burada tek seferde kodu yaz hepsinde çalıştır. Ayrıca bu da daha hızlı geliştirme yapmaya yarar ve zaman açısından maaliyet oldukça azalır. Bu bir geliştirici için oldukça avantajlı bir durumdur. Ayrıca kısa sürede daha fazla pazarda uygulama yer alır. HTML5 ve CSS3 desteklediği yerel dillerde kaynak bulmak oldukça kolaydır, geliştirmeyi de hızlandırır. Tasarımda tutarlılık vardır ve her cihazda aynı tasarım görüntülenir. Native dillerde birbirinde olmayan özellikler için alternatiflere bakılır bu da tasarım da farklı iştetim sistemlerinde farklılıklara sebep olabilir. Ancak eksi yönleri de bulunmaktadır. Özellikle farklı işletim sistemlerine entegrasyonu geliştirici ayarlamalıdır. Native dillerde geliştirmelerde geliştrici buna vakit ayırmamaktadır. Kullanılan framework, işletim sistemi tarafından desteklenen tüm özelliklere sahip olmayabilir. Google a, Android'e ya da Windos a bir özellik eklerse, frameworkün bunları güncelleyebilmesi gerekir. Her platform, ekran düzeni ve işlevsellikleri birbirinden farklıdır, yani tüm cihazlarda ve platformlarda mükemmel kullanıcı deneyimi sunan bir uygulama oluşturmak kolay değildir. Cross platformda yazılan uygulamalarda bu çok da sağlanamamaktadır. Örneğin, Facebook öceleri React native ile geliştirme yapmanın kullanıcı deneyimi açısından iyi olmadığını söyleyip, native uygulamaları destekledi. Daha sonra kullanıcı deneyimi için cross-platform mantığından native uygulama mantığına geçmiştir. <https://venturebeat.com/2012/09/11/facebooks-zuckerberg-the-biggest-mistake-weve-made-as-a-company-is-betting-on-html5-over-native/> Ayrıca oyun ya da grafiklerin çok fazla olduğu uygulamalarda grafikler ya da 3D teknolojisi native lar kadar iyi iş çıkaramaz. Bu yüzden bir çok oyun firması oyunlarını native olarak geliştirmektedirler. Platformun ve işletim sisteminin kendince üstün olan özellikerini daha fazla kullanabilmeleri için.

Fox Sports, Alaska Airlines, American Cancer Society, BBC Good Food gibi uygulamalar cross platfom (Xamarin) de yazılmıştır. Burada kullanıcıya hızlı bir şekilde ulaşmak ve çok fazla da platform işlevsellikleri ön planda olmadığı için tercih edilmiştir.



<https://medium.com/simbirsoft/creating-a-mobile-app-in-2020-native-vs-cross-platform-development-d90f25cef188>