

## Native/Cross Platform Uygulama Geliştirme

Mobil uygulama geliştirmek için birden fazla tür vardır. Native, Hibrit, Cross Platform bu türlerden bazılarıdır. Bir seçim yapmadan önce native ve cross platform'dan biraz bahsedeceğim. Her ikisinin de hem avantajları hem dezavantajları var.

### Native Geliştirme

Native mobil uygulamalar belirli bir işletim sistemi için yapılan uygulamalardır. Andorid için yapılan uygulamalar sadece android işletim sistemlerinde, ios için yapılan uygulamalar da sadece ios işletim sistemlerinde kullanılabilir. Yani bir uygulama fikrini her iki platformda da kullanılmasını istiyorsak android geliştiriciler android için, ios geliştiriciler ios için kod yazmalıdır.

- **Native uygulama geliştirmek istiyorsak**

- 

**Andorid için:**

Java - Kotlin

**IOS için:**

Object-C veya Swift

- **Native uygulama geliştirmenin avantajları ve dezavantajları nelerdir?**

**Avantajları:**

1. Uygulamaları tek bir platform için gerçekleştirdiğimiz için performans konusunda çok iyi ve hızlıdır.
2. Tek bir platforma ait olduğu için geliştiriciler ui-ux tasarımını o platformda en iyi çalışacak şekilde yaparlar ve bu da tasarımın harika olmasını dolayısıyla kullanıcının memnuniyetini garanti eder.
3. Uygulamalar cihazın işlevselliği ile uyum sağlar ve cihazın donanımıyla kolayca entegre olur. (gps, kamera, dokunmatik ekran vb.)
4. Uygulamayı google play market ya da app store gibi platformlara yüklerken onların yönergelerine uyumluluk sağlar bu da yükleme işini kolaylaştırır.
5. Uygulamayı kullanmak genellikle belirli işlevler dışında internet bağlantısı gerektirmez.

**Dezavantajları:**

1. Uygulama tek bir platforma özgü yapılır bu da aynı uygulamayı farklı platformlarda farklı kod tabanları gerektirir.
2. Her platforma özel uygulama yapmak ise daha fazla geliştiriciye ihtiyaç duyar.
3. Daha fazla geliştirici daha fazla maliyet demektir. Bu yüzden native uygulama yapmak cross platforma göre bazı şartlar dışında daha maliyetlidir.

**Cross Platform Geliştirme**

Cross platformda uygulama geliştirmek istediğimizde farklı kod tabanlarına ihtiyaç kalmadan birden fazla işletim sistemine uygun bir uygulama yapmış oluruz.

- **Uygulama geliştirmek için kullanılabilecek diller:**

Dart - Flutter

C# - Xamarin

JavaScript - React Native, Ionic, Cordova

- **Cross Platform uygulama geliştirmenin avantajları ve dezavantajları nelerdir?**

**Avantajları:**

1. Aynı anda birden fazla işletim sisteminde olması daha fazla kullanıcıya erişmeyi sağlar. Bu aynı zamanda uygulamanın gelirinin de artmasını sağlar.
2. Birden fazla geliştirici olmadan tek bir kişi ile yapılabildiğinden maliyeti de çok yüksek olmaz.
3. Native uygulama geliştirmeye göre (belirli durumlar dışında, bu durumları karşılaştırma yaptığım kısımda belirteceğim.) daha kısa sürede yapılabilir.

### **Dezavantajları:**

1. Performans açısından native uygulamalara göre düşüklük yaşanır daha yavaştır.
2. Kitaplık ve SDK'ların sınırlı desteğinden dolayı gerekli özelliklerin entegrasyonu çok zor olabilir.
3. Uygulamayı birden fazla platforma uyarlamak zorunda olmak çeşitli cihazlarda tasarım konusunda sorun yaratabilir. Bu da geliştirici için ekstra bir iş yaratır.

### **• Native / Cross Platform Karşılaştırma**

1. Performans konusunda native uygulamalar daha iyi performans gösterir.
2. Native uygulamalar cihazın işlevleri ile uyumludur ama cross platform uygulamalarının api'ye erişimi sınırlıdır. Bu yüzden üçüncü taraf kitaplıklarını uygulamaya entegre etmek gerekir ama bu kolay değildir çünkü diğer kitaplık ve SDK desteği sınırlıdır.
3. Daha çok kullanıcıya ulaşma konusunda Cross Platform daha iyidir çünkü birden fazla platformdadır ve bu da kullanıcı sayısını arttırır. Aynı zamanda daha fazla kullanıcı daha fazla gelir demektir.
4. Zaman konusuna gelecek olursak native uygulamalar daha fazla zaman alır ama bu proje gereksinimlerine göre değişebilir bir durumdur. Çünkü cross platform uygulamalarına özellikler entegre edilmesi gerekiyorsa bu bazen native uygulamaların bitirilmesinden bile daha uzun sürebilir. Ayrıca performans sorunları vb. nedenler ile hata düzeltme de daha fazla zaman alabilir. Yani zaman tamamen projenin gereksinimlerine bağlıdır.
5. Native uygulamalar tasarım ve performans konusunda daha iyi olduğundan ve cross platform uygulamaları ise tüm platformların taleplerini aynı anda karşılayamadığı için istenilen kullanıcı deneyimi sağlanamayabilir. Bu yüzden tasarım konusunda native uygulamalar daha iyidir.
6. Native uygulamalar tek bir platforma özel ve cihazın işlevselliği ile tam uyumluluk göstereceğinden kullanıcılarını daha doğru bir şekilde korurken cross platform uygulamaları birden fazla platformun tüm yönlerini kapsamak zorunda olduğu için veri koruması yeterli olmaz. Yani verilerin güvenirliliği ve güvenlik açısından native uygulamalar daha iyidir.

- **Native uygulamaları geliştirebileceğimiz IDE (Android/IOS)?**

**Andorid:**

- Android geliştirmede resmi ve en yaygın olarak kullanılan ide android studio'dur.
- Performans konusunda iyidir.
- Hata ayıklama konusunda çok iyidir.
- Açık kaynak kodlu yazılımdır.
- Ücretsizdir.
- Açılması biraz geç olur.
- İlk derlemede biraz yavaş olsa da ikinci derlemede daha hızlı olur.
- Büyük android geliştirici toplulukları tarafından desteklenir.
- Java'dan Kotlin'e otomatik çeviri sağlar.
- Esnek arayüze sahiptir.
- Hedef cihazın özelliklerine tam erişim verir.
- Gradle'a dayalı esnek bir yapı sistemi vardır.
- (Gradle: Derleme, test etme, uygulanabilir ve yayınlanabilecek bir apk haline getirmeyi sağlar.)
- Google bulut platformu ve birçok işletim sisteminde native uygulama oluşturulmasına izin verir. Tek şart jdk yüklü olmasıdır.
- Sadece android platformunu destekler.

## **IOS:**

- XCode Android Studio'nun doğrudan eşdeğeri.
  - Apple'ın resmi idesidir.
  - IOS uygulamalar için ihtiyaç olan (derleyici, ayıklayıcı vs.) sahiptir.
  - Kapsamlı dil desteği vardır. (C, C++, Objective-C, Java, Python vb.)
  - Kodsuz kullanıcı arayüzü tasarımı oluşturmayı kolaylaştıran gui aracı vardır.
  - Simülatör Iphone'u taklit eder ve test etmeye olanak ve kolaylık sağlar.
  - Çok iyi bir kod tamamlama özelliği vardır.
  - Hedef cihazın özelliklerine tam erişim verir.
  - Yalnızca apple işletim sistemlerine özeldir.
  - Uygulamayı doğrudan cihaza aktarmak karmaşıktır.
  - Yalnızca IOS platformu için ayrılmış kod tabanına sahiptir.
- 
- **Cross Platform uygulamaları geliştirebileceğimiz IDE**

## **Visual Studio Code:**

Microsoft tarafından Windows, Linux ve MacOS için geliştirilen bir kaynak kodu düzenleyicidir. Araştırmalara göre en çok kullanılan idedir.

Hata ayıklama, kod tamamlama, git kontrolü desteklerini içerir.

Hızlı, ücretsizdir ve açık kaynak kodlu yazılımdır.

Çok geniş bir eklenti desteği sağlar.

Geliştiricilerin uzantılar oluşturulmasına ve kullanmasına, zengin bir şekilde özelleştirmesine olanak tanır.

**Hem IOS hem de Android için mobil uygulama geliştirileceğini düşünelim.**

**Ayrıca Web versiyonu ve yönetim paneli olacağını da düşünelim.**

**1. Böyle bir projeyi geliştirirken hangi programlama dillerini tercih edersiniz?**

Hem IOS hem de Android için mobil uygulama geliştireceğim için ve bütçemin de yeterli olmadığını varsayarak araştırmalarım dahilince Cross Platform uygulama geliştirmeyi seçerim.

Dil seçimimden bahsetmeden önde React Native ve Flutter'den bahsetmek istiyorum.

**Flutter:**

Bir UI araç kitidir ve içerisinde birçok eklenti bulundurur.

Dart programlama dili ile oluşturulmuş , açık kaynak kodlu ve ücretsiz bir teknolojidir.

Flutter uygulamasını test etmek kolaydır.

Hızlıdır.

Düzenli destek gerektirir yani build, test vs. için özel scriptler geliştirilmesi gerekir. Bu uygulamanın geliştirilmesinde zaman kaybı yaratır.

Google Flutter teknolojisini desteklemektedir ve eğer desteklemeyi bırakırsa Flutter yok olabilir. Riskli bir teknolojidir diyebiliriz.

Dosya boyutu büyüktür bu yüzden geliştiriciler daha küçük uygulama boyutu oluşturmak için farklı numaralar kullanırlar. Animasyon kullanılmaması, görüntülerin küçültülmesi dolayısı ile kalitenin düşmesi vb.

**Flutter kullanan bazı markalar:**

Alibaba Group, Google Ads, Abbey Road Studios...

### React Native:

JavaScript kullanan açık kaynaklı bir mobil uygulama platformudur.

Hızlı güncellenir ve olgun bir platformdur.

Geliştirici topluluğu çok büyüktür.

Öğrenmesi kolaydır.

Native uygulamalara göre daha büyük dosya boyutuna sahiptir. Çünkü Android JavaScript ile yazılan kodu direct derleyemez bunun için gerekli kütüphanelerin dahil edilmesi lazımdır. Bu kütüphaneler dahil edildiğinde ise dosya boyutu büyür.

### React Native kullanan bazı markalar:

Facebook, Walmart, Bloomberg, Instagram...

Yukarıda yazdıklarıma dayanarak seçtiğim dil React Native olurdu. Çünkü Flutter'a göre daha olgun ve desteklenmesi daha güçlüdür.

Zaman tasarrufu sağlıyor ve buna ek olarak maliyeti düşürüyor.

Şu anda durma belirtisi durmadan hızla büyümeye devam ediyor.

## 2. Hangi geliştirme ortamında (IDE) geliştirmeyi tercih edersiniz?

Kesinlikle Visual Studio Code ortamı derim.

Neden peki?

### Visual Studio Code:

Microsoft tarafından Windows, Linux ve MacOS için geliştirilen bir kaynak kodu düzenleyicidir. Araştırmalara göre en çok kullanılan idedir.

Hata ayıklama, kod tamamlama, git kontrolü desteklerini içerir.

Hızlı, ücretsizdir ve açık kaynak kodlu yazılımdır.

Çok geniş bir eklenti desteği sağlar.

Geliştiricilerin uzantılar oluşturulmasına ve kullanmasına, zengin bir şekilde özelleştirmesine olanak tanır.

Tüm bu sebepler Visual Studio Code seçmemi sağlar. Ayrıca araştırmalara göre bugün yaklaşık 14 milyon kullanıcıya sahiptir. Microsoft'a göre ise dünya çapında yaklaşık 24 milyon geliştirici vardır yani geliştiricilerin yarısından fazlası visual studio code kullanımını tercih etmektedir.

### 3. Hangi veritabanı yapısını kullanırsınız?

2 veritabanı yapısı arasından birini seçmeden önce PostgreSQL ve MySQL'den biraz bahsetmek istiyorum.

#### PostgreSQL:

Kendisini “dünyanın en gelişmiş açık kaynaklı ilişkisel veritabanı” olarak tanıtıyor.

Nesne ilişkisel bir veritabanıdır.

Karmaşık sorguları ve büyük veritabanlarını işler. Bu yüzden çok fazla güzel özelliklere sahiptir.

Zengin özelliklere sahiptir ve genişletilebilirdir.

Eskiden okumaları MySQL'e göre daha yavaş ama büyük miktarda veriyi daha verimli bir şekilde yazabiliyordu. Fakat son sürümlerde bu aradaki fark neredeyse silinmiştir diyebiliriz.

Birçok bulut sağlayıcısı, her iki veritabanının da yönetilen ölçeklenebilir sürümlerini sunar.

İşlem düzeyinde veri bütünlüğünü korur ve bu da verilerin bozulmasına karşı korur.

Genişletilebilirlik, standartlara uygunluk, ölçeklenebilirlik ve veri

Bütünlüğü göz önünde bulundurularak oluşturulmasından dolayı basit okuma ağırlıklı işler için PostgreSQL yerine MySQL daha iyi bir seçimdir.

Sık güncellemelere sahiptir.

#### PostgreSQL'in Desteklediği Diller:

C/C++ , Delphi, Erlang, Go, Java, JavaScript, Lisp, .Net, Python, Tcl R ve diğer diller



## MySQL:

Tamamen ilişkisel veritabanıdır.

Açık kaynak kodlu olmasına rağmen bazı lisans sorunları vardır.

Kurulumu ve yönetimi nispeten daha kolay bir veritabanıdır.

Hız ve güvenilirliğe odaklanabilmesi için daha hafif özelliklidir.

## MySQL'in Desteklediği Diller:

C/C++ , Delphi, Erlang, Go, Java, Lisp, Node.js, Perl, PHP, R

Bu araştırmalarım sonucunda yapılacak uygulamanın büyük olması, önemli ve büyük verileri tutacak olması ve bu verilerin bozulmaması için güvenilir bir veritabanı kullanmak, ek olarak kullanacağım framework olan React Native için kullanılan JavaScript dilini destekliyor olmasından dolayı PostgreSQL veritabanı yapısını seçtim.

## 4. Proje yöneticisi olsanız kaç kişi ile çalışırsınız ve nasıl görevlendirme yaparsınız?

Hangi geliştiricilere ihtiyaç var?

### Mobile Developer:

Kodlama ustalığını alan ve uygulama geliştiren yazılım geliştiricisidir. Akıllı telefon, tabletler vs. için yeni teknolojiler geliştirir.

### Backend Developer:

Uygulamanın veritabanı ve işletim sisteminin düzgün ve verimli entegrasyonundan sorumludur. Tüm ön uç dinamiklerinin arka uç yazılımına anında yanıt vermesini sağlamaktan sorumludur.

### UI – UX Designer:

UI arayüz tasarımını ifade ederken, UX deneyim üzerine kurulmuş bir tasarım sürecini anlatan geliştiricilerdir. UI Uygulamanın tasarımının işleyişinden, hızından, estetiğinden yani tüm grafiklerden sorumludur. Yazılan koda bir dış görünümüm sağlar. UX ise tasarım yapmaktan çok kullanıcıların site üzerindeki etkileşimin nasıl olacağıyla ilgilenir.

### Quality Assurance Specialist:

Uygulamanın test aşaması ile ilgilenir ve projenin başından sonuna kadar ayrılmaz bir parçasıdır. Projenin son aşamasına kadar hazır olmasını sağlamak için hataları ve eksiklikleri tespit eder.

Daha az geliştirici ile hatta tek bir kişi ile bile yapılabilir ama zaman, kalite vb. konularda düşüklük görülür. Bu yüzden yukarıdaki geliştiricilerin olması gerekir. Proje erken bitirilmek istenirse sayılar arttırılabilir.

Proje sürecini ve görevleri belirlerken agile metodlarından biri olan Scrum metodunu kullanırım. Bu metodu kullanma sebepime değinecek olursam, belirlediğim projemin birçok insana hitap etmesi ve büyük bir proje olması, değişebilecek yani esnek, belki de birçok kez üstünden geçilmesi ve tekrar edilmesi gereken son olarak da müşterinin isteklerinin ve geri dönüşlerinin bu proje için çok önemli olmasından dolayı Scrum metodu kullanarak görevlendirme ve projeyi yönetme yoluna giderim.

- 1- Öncelikle projenin gereksinimlerini tam olarak belirlerim.
- 2- Gereksinimleri belirledikten sonra bir öncelik sıralaması yaparım ve onları listelerim.
- 3- Öncelikleri belirledikten sonra projeyi küçük kısımlara ayırırım. Böyle yapmak projede aşama aşama ilerlenmesine ve projede bir değişiklik gerektiğinde bunu proje üzerinde uygulanmasına kolaylık sağlar.
- 4- Daha sonra bu projeyi yapacak ekibim ile parçalara ayırdığım her kısım için de ayrı gereksinimleri belirlerim.
- 5- Öncelik sırasına göre ilk kısımdan başlarım. Seçtiğim kısım için projenin ne zamana kadar yetişmesi gerekiyorsa ve büyüklüğüne göre o kısma ortalama bir süre veririm 2-3 haftalık veya daha fazla. Daha sonra ekip arkadaşlarımla projeye başlarız ve sprint boyunca günlük toplantılar yaparak (bu toplantılara müşteri de katılır çünkü ilerleyiş onun da görmesi ve eğer değişiklik istiyorsa belirtmesi önemlidir.) projemizin ne durumda olduğunu, değişiklik yapmak isteyip istemediğimizi yani genel olarak 10-15 dakikalık toplantılar ile genel ilerleyiş hakkında değerlendirmeler yapılır. Böyle toplantıların yapılması projenin doğru ve düzenli ilerlenmesinde, eğer bir değişiklik yapılacaksa çok ilerlenmeden hemen değişikliğin uygulanmasına ve projenin aksamamasını en önemlisi de zaman kaybının yaşanmamasını sağlar.

- 6- Sprint bittikten sonra test aşamasına geçilir ve o kısım test edilmeye başlanır. Bir sorun çıktığında ise o sorun çözülür ve yeniden test edilir. Test sonucunda sorunsuz olduğu belirlendikten sonra o kısım için sprint biter ve diğer kısım için sprint başlar. Böyle kısımlara ayırmamız bir sorun çıktığında tüm projeyi değil de sadece o kısmı etkiler ve kısımlar da küçük olduğu için sorunu düzeltmek projenin tamamını düzeltmeye göre daha kolay olur.
- 7- Diğer sprintler için de aynı aşamalar uygulanır ve en son bittiğinde yeniden bir test aşamasına geçilir.
- 8- Test aşaması da sorunsuz bir şekilde bittikten sonra projemiz tamamlanmış ve hazır hale gelir.

## 5. Bir ekiple birlikte projeyi yapıyor olsanız sizce ne kadar sürede biter?

Her mobil uygulama gereksinimleri birbirinden farklı olduğu için geliştirilmesinin gerektirdiği süre de birbirinden farklıdır.

3 faz vardır.

Analiz (ortalama 1-2 hafta arası),

Tasarım(ortalama 4-6 hafta arası ama ekran sayısı arttıkça süre de o kadar uzar),

Geliştirme(bu kısım yazılım geliştiricilere göre değişir)

Geliştirme kısmındaki süre ise insan kaynağının artması ile azalır ürün özellikleri karmaşılaştıkça uzar.

Bu proje 5 kişi ile birlikte çalışacağı için ona göre düşünecek olursam projenin bitmesi yaklaşık 4-5 ay sürer. Projenin gereksinimleri arttıkça, test aşamasında karşılan hataların ve eksikliklerin düzeltilmesi karmaşılaştıysa bu süreyi biraz daha uzatabilir.

Gizem Coşkun