

## ARAŞTIRMA MAKALESİ / RESEARCH ARTICLE

MOBİL UYGULAMALARDA HİBRİT SİSTEMLERİN OLUŞTURDUĞU KULLANIM  
KOLAYLIKLARININ İNCELENMESİİsmail GÜLER<sup>1</sup><sup>1</sup>Bilgisayar Mühendisliği Bölümü, İstanbul Aydın Üniversitesi, İstanbul, Türkiye  
ismailguler@stu.aydin.edu.tr, ORCID No: 0000-0002-0169-4038

GELİŞ TARİHİ/RECEIVED DATE:11.06.2021 KABUL TARİHİ/ACCEPTED DATE:26.04.2022

**Özet**

Akıllı telefon uygulamaları, insanları sağlıklı yaşam tarzlarını benimsemelerine destek olmaktadır. Dolayısıyla davranış değişikliklerini kolaylaştıran hibrit mobil uygulamalara dâhil olan ikna edici tasarım stratejilerini anlamakta fayda vardır. Araştırmamızın amacı, Hibrit mobil uygulamaların Web ile Native uygulamalara göre kullanım kolaylıklarını göstermek için yapılmıştır. Hibrit mobil uygulamalar, Web uygulamalarının özelliklerini ve Native mobil uygulamalarını birleştirir. Web uygulamaları gibi HTML ve JavaScript gibi taşınabilir, platformdan bağımsız dillerde de uygulanırlar. Native uygulamalar gibi yerel cihaz kaynaklarına doğrudan erişirler. Hibrit uygulamalara geçiş teknolojinin hızlı ilerleyişiyle birlikte son dönemde büyük bir artış göstermektedir. Native uygulamalara göre performans eksileri olmasına rağmen yazılım geliştiriciler tarafından kullanım kolaylıkları sebebiyle daha çok tercih edilmektedir. Sonuçlardan, hibrit uygulamalar için kullanıcı arayüzünün dinamik yapısıyla geliştirilmesinin daha basit olduğunu ve hibrit uygulamalarda kullanıldığında mobil cihaz özelliklerini performans bakımından daha iyi kullandığı belirlenmiştir.

**Anahtar Kelimeler:** Mobil uygulamalar, Hibrit sistemler, e-Business**EXAMINING THE EASE OF USE OF HYBRID SYSTEMS IN MOBILE APPLICATIONS****Abstract**

Smartphone apps support people to adopt healthy lifestyles. Therefore, it is useful to understand the persuasive design strategies involved in hybrid mobile applications that facilitate behavior changes. The aim of our research is to show the ease of use of hybrid mobile applications compared to Web and Native applications. Hybrid mobile apps combine the features of Web apps and Native mobile apps. Like web applications, they are implemented in portable, platform-independent languages such as HTML and JavaScript. They directly access local device resources such as native apps. Transition to hybrid applications has been increasing in recent years with the rapid progress of technology. Although it has performance disadvantages compared to native applications, it is more preferred by software developers due to its ease of use. From the results, it has been determined that the user interface for hybrid applications is simpler to develop with its dynamic structure and it uses mobile device features better in terms of performance when used in hybrid applications.

**Keywords:** Mobile applications, Hybrid systems, e-Business

## 1. GİRİŞ

Günümüzde insanlar birbirleriyle bağlantı kurabilmek için genel olarak mobil iletişimi kullanmaktadır. Mobil iletişim; kablosuz cihazlarla mekân ve zaman gözetmeksizin yeterli sayıda baz istasyonunun oluşturduğu telsiz ağlar üzerinden kurulan bir haberleşme şeklidir. İlk mobil haberleşme sistemi 1991 yılının sonlarında Fransa, Almanya ve İtalya tarafından Cenevre’de kurulmuştur (Özkan, 2007). Son dönemde teknolojinin hızlı artışıyla birlikte mobil iletişimde de büyük gelişmeler görülmekte olup bunun en çok rağbet gören kısmı mobil yazılımlardır. Akıllı telefonlar ve tablet bilgisayarlar için geliştirilen yazılımlara mobil uygulama denmektedir. Mobil uygulamaları kullandığımız cihazlarda mobil işletim sistemleri üzerinden çalışmaktadır ve en sık kullanılanlar Windows, IOS ve Android tabanlı işletim sistemleridir. IOS işletim sistemi kullanan cihazlar için Apple App Store’dan, Android işletim sistemi kullanan cihazlar için Google Play Store’dan ve Windows içinse Windows Phone Store gibi mağazalardan istediğimiz uygulamayı istediğimiz anda zaman gözetmeksizin mobil cihazlarımıza indirebilmekteyiz. Bu uygulamalar, kullanıcının isteğine bağlı olarak ücretli veya ücretsiz olabilirler.

Web teknolojilerindeki gelişmeler göz önüne alındığında mevcut seçeneklere bakarak insanlar hem mobil web sitelerini hem de uygulamalarını kullanıyorlar ancak mobil uygulamalar her zaman bir adım önde oluyor. Mobil web siteleri HTML, JavaScript ve CSS standardı ile desteklenen tarayıcı temelli sayfalar olup diğer web sitelerine benzer yapıları vardır. Mobil uygulamalar ise daha kullanıcı dostu ve kullanımı kolaydır. İnsanlar sürekli olarak URL yazmak ya da tarayıcıda sayfanın görüntülenmesini beklemek yerine mobil uygulamalara yöneliyorlar. Web standartlarındaki yeni gelişmelerle birlikte mobil uygulamalar Hibrit ve Native olmak üzere iki ana gruba ayrılmaktadır (Charland ve Leroux, 2011). Bu yapılan ayrım, uygulamanın kodlanmaya başlandığı ortam ve yapılışıyla ilgili bir gruplandırma değildir.

Yazılım geliştiriciler hem veri hem de kullanıcılar için mantıklı yaklaşımlar düşünmelidir. Yapılan uygulamanın en iyi şekilde performans, test ve canlı ortamda denenmesi gerekmektedir. Bir yazılım ilk önce kodlamanın uygulanması ile başlar ve bunun hatasız olması gerekmektedir. Hibrit ve Native uygulamalar farklı platformlarda farklı yapılarla uygulanmaktadır.

## 2. HİBRİT UYGULAMALAR

HTML5, CSS ve JavaScript ile yazılan, Native ve Web platformlarını ortak kullanan mobil çözümlere hibrit uygulamalar denmektedir. Daha anlaşılır şekilde açıklamak gerekirse Native uygulamalar içinden ulaşılan mobil web yazılımlardır. Native mobil uygulama özelliklerine ulaşabilmesi için bir ara taşıyıcı kullanmaktadır. Taşıyıcıya en sık kullanılan PhoneGap örnek verilebilir (Yılmaz, 2010). Bunun dışında birçok taşıyıcı bulunmaktadır.

Hibrit mobil uygulamalar, Web uygulamalarının özelliklerini ve Native mobil uygulamalarını birleştirir. Web uygulamaları gibi HTML ve JavaScript gibi taşınabilir, platformdan bağımsız dillerde de uygulanırlar. Native uygulamalar gibi yerel cihaz kaynaklarına (dosya sistemi, konum, kamera, kişi vb.) doğrudan erişirler.

Uygulama geliştiriciler, mobil cihazların donanım özelliklerini derinlemesine kullanmadan da uygulama geliştirebilirler ve hibrit uygulama yöntemi ile rahatlıkla istediği geliştirmeleri yapabilirler. Hibrit

uygulamalar cihazın ve işletim sisteminin tüm özelliklerini, belirli bir platformda minimum performans yükü ile kullanmanıza olanak tanır. Hibrit sistemler ile iş yükü geliştiricinin üzerinden kalkmış olup tek bir kez yazılarak bütün mobil cihazlarda çalıştırılabilir olmasıdır. Bu özelliği ile son dönemde çok fazla tercih edilir olmuştur. Bununla birlikte, bir web uygulaması oluşturmak kodların platformlar arasında taşınmasını sağlar. Bu da geliştirme süresini ve maliyetini önemli ölçüde düşürebilir. Hibrit uygulamalar, Native ve benzeri uygulamaları çeşitli platformlara dağıtmak için ortak bir kod tabanı kullanarak her iki platformun en iyilerini birleştirir.

### 2.1. Hibrit Uygulamalara Talebin Artmasının Sebepleri

Hibrit mobil uygulamalar, geliştiricilerin mevcut becerilerini web uygulamalar geliştirmede yeniden kullanabilmelerinin bir yolunu sağlar. Bu uygulamalar, platform satıcıları tarafından sağlanan programlama dilleri ve SDK'leri içermektedir.

Hibrit mobil uygulama geliştirme, bir kurum için oldukça ilgi çekici gelmektedir. Kurum, iki ya da daha fazla platform için ayrı ayrı yazılımcı işe almak yerine tek bir yazılımcıyla HTML, CSS ve JavaScript kullanarak her platforma hitap eden yazılımlar geliştirmeyi cazip görmektedir. Maliyet açısından ince hesaplar düşünen firmalar bu şekilde hibrit mobil uygulamalara yönelmektedir.

### 2.2. Hibrit Uygulamaların Pozitif Yönleri

- i. Yazılım geliştiriciler mevcut web becerilerini kullanmaktadırlar.
- ii. Birden fazla platform için bir kod tabanı bulunmaktadır.
- iii. Geliştirme süresinin ve maliyetin düşürülmesi artı bir avantajdır.
- iv. Responsive web tasarımını kullanarak çeşitli form faktörleri için kolayca tasarım yapılabilmektedir.
- v. Bazı cihazlara ve işletim sistemi özelliklerine erişim sağlanmaktadır.
- vi. Gelişmiş çevrimdışı yetenekler bulunmaktadır.
- vii. Uygulama, mağazalar aracılığıyla yerel olarak dağıtıldığı için ve arama motorları aracılığıyla mobil tarayıcılara dağıtıldığı için görünürlüğü artmaktadır.

### 2.3. Hibrit Uygulamaların Negatif Yönleri

- i. Karmaşık Native işlemlere veya 3D oyunlar gibi ağır geçişlere sahip olan bazı uygulamalar için performans sorunları yaratabilmektedir.
- ii. Hibrit sistemlerde Native bir kullanıcı arayüzünü taklit etmek ve gerekli alt yapıyı oluşturmak için gereken zaman ve işçilik artar.
- iii. Hibrit uygulamalar, bütün aygıt ve işletim sistemi özelliklerini desteklememektedir.
- iv. Apple, geliştirilen uygulamayı Native olarak değerlendirmeye ise reddedilme riski ortaya çıkmaktadır.

Bu dezavantajlar önemlidir ve göz ardı edilmemelidir. Hibrit yaklaşımın her tür uygulamaya uymadığı görülmektedir. Hedef kullanıcılarınızı, seçtikleri platformları ve uygulamanın gereksinimlerini dikkatle

değerlendirmeniz gerekmektedir. İçerik tabanlı olanlar gibi pek çok uygulamada avantajlar sakıncaları ağırlaştırılmaktadır.

Hibrit uygulamalar, telefonunuzdaki diğer uygulamalar gibi kolayca kurulabilmektedir. Telefonunuzun bulunduğunu platforma uygun mağazalardan ücretli ya da ücretsiz indirebilirsiniz. Diğer uygulamalardan farkı yoktur; mesaj yollayabilir, fotoğraf çekebilir, alışveriş yapabilir ve çok daha fazlasını yapabilirsiniz. Uygulamanızın hibrit olup olmadığı kolayca anlaşılabilir değildir.

Hibrit uygulamalarda iki yaklaşım bulunmaktadır (Charfi ve Mezini, 2005). Bunlar WebView ve Derlenmiş Hibrit uygulamalardır:

#### 2.4. WebView Uygulamaları

HTML, CSS ve JavaScript kod tabanı ile yerel bir uygulamada sarılmış olan dâhili bir tarayıcıda çalışır ve bu WebView uygulamaları olarak anlandırılır. Bazı yerli API'ler bu taşıyıcı aracılığıyla JavaScript'e açıktır. WebView uygulamaları için en iyi örnekler PhoneGap ve Trigger.io'dur.

#### 2.5. Derlenmiş Hibrit Uygulamalar

Hibrit uygulamalar için kodlama bir kez yapılmaktadır ve desteklenen her platform için yerel koda çevrim yapılmaktadır. Genellikle C# ve JavaScript gibi platformlarda yazılır. Sonuç olarak, her platform için hibrit bir uygulama derlenmektedir, ancak geliştirmede daha az özgür kalınmaktadır. Örnek olarak Xamarin, Appcelerator Titanium ve Embarcadero FireMonkey'dir (Duru, 2013).

### 3. NATIVE UYGULAMALAR

Native, bir platformun kültürünü, dilini, sözleşmelerini ve normlarını oluşturan teknolojileri, yani dilleri ve çerçeveleri ifade eder. Kullanıcıların belirli bir platformdaki uygulamalarla iletişim kurmak için kullandığı temel semboller, hareketleri ve etkileşimleri içeren soyutlama temel seviyesidir (Charland ve Leroux, 2011). Aynı zamanda belirli bir platformdaki uygulamaların kullanıcılarla iletişim kurmak için kullandığı kelimeler, cümleler ve kavramlardır. Belli bir platformda bunlar daha kullanışlı ve tutarlı olursa, o platform için yerel uygulamalar oluşturmanın avantajları o kadar çoktur.

Günümüzde IOS ve Android'in Native platformların başını çektiği ve bunlar üzerinden birçok uygulama yapıldığı görülmektedir. Bu platformlar, üreticiler, taşıyıcılar ve kullanıcılar tarafından, farklı telefonlar ve uygulamalar arasında tutarlılık olmasına rağmen çok az olması durumunda özelleştiriliyor.

Yazılım geliştiriciler IOS uygulamalar için Swift ya da Objective-C gibi nesne yönelik programlama dillerini kullanmaktadır. Aynı şekilde Android için ise Java ya da Android Studio kullanılmaktadır. Yazılımcılar tarafından geliştirilmesi en az tercih edilen Windows uygulamalardır ve C# nesne tabanlı programlama dili aracılığı ile geliştirilmektedir (Elgazzar et. al., 2010).

### 4. HİBRİT UYGULAMALARIN GELİŞTİRİLMESİ

Hibrit mobil uygulamalar web siteleri gibi benzer şekilde HTML, CSS ve JavaScript teknolojileri kullanılarak geliştirilmektedir. Hedef olarak bir mobil tarayıcı yerine, yerel bir taşıyıcı da barındırılan bir WebView

hedef alınır. Burada mobil cihazın donanım özelliklerine erişmek gibi birtakım işler yapılmaktadır. İnternetteki web sitelerinde olduğu gibi hibrit mobil uygulamalar HTML, CSS ve JavaScript gibi web teknolojilerinin bir kombinasyonu ile oluşturulmuştur. En önemli fark, hibrit uygulamaların mobil platformun WebView'ini kullanan yerli bir uygulamada barındırılmasıdır. Bu, onların ivmeölçer, kamera, kişiler ve daha fazlası gibi cihaz özelliklerine erişmesini sağlar. Bunlar, çoğu zaman mobil tarayıcıların içinden erişmek için sınırlandırılan yeteneklerdir.

Günümüzde, çoğu hibrit mobil uygulama native kod ile oluşturulmuş eklentiler aracılığıyla cihazın özelliklerine erişmek için uygun bir JavaScript API desteği sağlayan Apache Cordova'yı kullanmaktadır (Jeong et. al., 2008). Apache Cordova başlangıçta PhoneGap adlı bir proje olarak başladı. Bugünlerde ise ek araçlar içeren Apache Cordova'nın bir dağıtım paketi olarak varlığını sürdürmektedir. Bunun yanında yazılım mühendislerinin geliştirdiği birçok eklenti de bulunmaktadır. Eklentiler cihazın sensörlerine, kişilere, kamera ve daha fazlasına erişmek için API'lar içerir. Bu eklentiler, Apache Cordova'nın loglarında tutulmaktadır.

#### 4.1. Apache Cordova

Apache Cordova, Nitobi tarafından orijinal olarak oluşturulmuş bir mobil uygulama geliştirme yapısıdır. Adobe Systems, 2011 yılında Nitobi'yi satın aldı, PhoneGap olarak yeniden markaladı ve daha sonra Apache Cordova adlı yazılımın açık kaynaklı bir sürümünü piyasaya sunuldu. Apache Cordova, yazılım programcılarının Android, iOS veya Windows Phone'daki gibi platforma özgü API'lere güvenmek yerine CSS3, HTML5 ve JavaScript kullanan mobil cihazlar için uygulamalar oluşturmasına olanak tanır (Wargo, 2013). Cihazın platformuna bağlı olarak CSS, HTML ve JavaScript kodunun bir arada çalışmasını sağlar. Cordova uygulaması kendi içinde 3 katmana ayrılmaktadır.

##### 4.1.1. WebView

Cordova özellikli WebView, uygulamanın tüm kullanıcı arabirimini sağlamaktadır. Hibrit uygulamaların geliştirildiği bazı platformlarda, WebView'i yerel uygulama bileşenleri ile birleştiren daha büyük bir bileşen de olabilir.

##### 4.1.2. WebApp

Bu kısımda, uygulama kodunuzun bulunduğu bölümdür. Uygulamanın kendisi bir web sayfası olarak, varsayılan olarak index.html adında yerel bir dosya olarak uygulanır ve CSS, JavaScript, resimler, medya dosyaları ya da diğer kaynaklara başvurulması bunun çalışması için gereklidir. Web APP uygulama mağazalarına dağıttığınız yerli uygulama paketleri içinde WebView'i çalıştırır.

##### 4.1.3. Plugins

Eklentiler, Cordova içerisinde ayrılmaz bir parçadır. Cordova ve yerel bileşenlerin birbirleriyle ve bağlarla standart cihaz API'leriyle iletişim kurmaları için bir arabirim sağlarlar. Bu, JavaScript'ten yerel kodu çağırmanızı sağlamaktadır.

#### 4.2. Ionic Platform

Ionic Framework, css sınıflarının bir kümesidir ve AngularJS ile Cordova'nın üzerine kurulmuş Javascript yönergeleri ve modüllerinin bir kütüphanesidir (Malavolta, Ruverto, Soru ve Terragini, 2015). Ionic, hibrit

mobil uygulamaları oluşturma hedefi taşıyan bir HTML5 mobil uygulama geliştirme yapısıdır. Hibrit uygulamalar aslında, bir uygulamada Native platform katmanına erişimi olan bir tarayıcı kabuğunda çalışan küçük web siteleridir. Hibrit uygulamalar, native uygulamalara göre özellikle platform desteği, geliştirme hızı ve üçüncü parti koda erişim gibi birçok avantaja sahiptir.

Web geliştiriciler Ionic bir uygulamanın yapısını açıkça görmektedirler. Özünde, bu sadece hibrit bir uygulama kabuğunda çalışan bir web sayfasıdır. HTML, CSS ve Javascript'i kullanabileceğiniz anlamına gelmektedir. Web sitelerinden tek farkı, başkalarının bağlayacağı bir web sitesi oluşturmak yerine, kendine yeten bir uygulama geliştiriyor olmasıdır.

Ionic bir uygulamanın büyük kısmı HTML, Javascript ve CSS ile yazılmış olacaktır. Geliştiriciler, özel Cordova eklentileri veya native kodu kullanarak hibrit katmana da inebilirler, ancak mükemmel bir uygulama oluşturma gerekliliği sunmamaktadır.

## 5. SONUÇ

Bu makale, uygulama geliştirme yaklaşımlarının bazı avantaj ve dezavantajlarını açıklamakla birlikte hangisinin yazılım geliştirici için en iyisi olduğunu ana hatlarıyla belirtmiştir. Hibrit sistemler ve Native uygulama arasındaki seçim, bir uygulamanın amaçlanan özellik kümesine ve uygulamanın kapsamına bağlıdır. Native bir uygulamanın her zaman en iyi seçenek olduğunu söylemek kolay bir çıkarım olabilir, ancak bilgilendirici veya içerik dağıtım uygulamaları oluşturuyorsanız ve küçük bir bütçeyle çalışıyorsanız Hibrit platform en iyi çözümü sunmaktadır. Hibrit bir çözümün en büyük avantajı hata düzeltmeleri veya güncellemeler için Google Play, iTunes ve diğer uygulama mağazalarında yeni uygulama sürümleri yayınlamaya hiç ihtiyaç duymamasıdır. Kullanıcı son güncellemelerden, hata düzeltmelerinden veya güncellemelerden yararlanabilmesi için sayfayı yenilemesi veya uygulamayı yeniden açması gerekir. Her tür uygulama geliştirme için uygun yapılar, kütüphaneler, API'ler, geliştirme ve hata ayıklama araçları önemli rol oynamaktadır. Google Analytics'e entegre edilebilir ve kullanıcıların uygulamadaki davranışlarını uygulama içinde araştırılabilir, kitleleri segmentlere ayırabilir ve kullanıcıların mobil web sitesi uygulamalarındaki hareketlerini izleyebilir, hem de diğer analitik işlemleri gerçekleştirmektedir.

## 6. KAYNAKÇA

**Charfi, A. ve Mezini, M.** 2005. Using aspects for security engineering of web service compositions In Web Services, IEEE International Conference, 11-15 Temmuz, 59-66.

**Özkan, H.** 2007. Mobil Sistemler ve Mobil Phone Çalışma Mimarisi. Yayınlanmamış yüksek lisans tezi. Haliç Üniversitesi. Fen Bilimler Enstitüsü: İstanbul.

**Jeong, Y.-J., J.H. Lee ve G.-S. Shin.** 2008. "Development Process of Mobile Application SW Based on Agile Methodology", Advanced Communication Technology, ICACT 2008. 10th International Conference on, vol.1, pp.362-366, 17-20.

**Elgazzar, K., A.E. Hassan ve P. Martin.** 2010. Clustering WSDL Documents to Bootstrap the Discovery of Web Services, IEEE International Conference on Web Services.

**Yılmaz, O.** 2010. Web Uygulama Güvenliğine Hibrid Yaklaşım. Baskı Yeri: WGT E-Dergi. 4, 1-3.Doi: 10.1132/1941127.1941234.

**Charland, A. ve B. Leroux.** 2011. Mobile application development: web vs. native. Baskı Yeri: Communications of the ACM. 5, 49-53.Doi: 10.1145/1941487.1941504.

**Duru, İ.** 2013. Çevik Yöntemlerle Mobil Uygulama Tasarımı ve Gerçekleştirilmesi. Yayınlanmamış yüksek lisans tezi. Yıldız Teknik Üniversitesi Fen Bilimler Enstitüsü: İstanbul.

**Wargo, M.J.** 2013. Apache Cordova 3 Programming. Baskı Yeri: Pearson. 1, 1-18. ISBN-13: 978-0-321-95736-8.

**Malavolta, I., S. Ruberto, T. Soru ve V. Terragni.** 2015. Hybrid mobile apps in the Google play store: an exploratory investigation. Baskı Yeri: MOBILESoft '15 Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems. 5, 56-59.Doi: 10.1109/MobileSoft.2015.15.