



T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



YÜKSEK LİSANS TEZİ

ÇOKLU PLATFORM MOBİL UYGULAMALAR OLUŞTURMAK
İÇİN UYGUN YAKLAŞIM SEÇME ANALİZİ

Çağrı ÇINAR

DANIŞMAN
Prof. Dr. Ahmet SERTBAŞ

II. DANIŞMAN
Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN


Bilgisayar Mühendisliği Anabilim Dalı

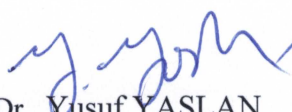
Bilgisayar Mühendisliği Programı

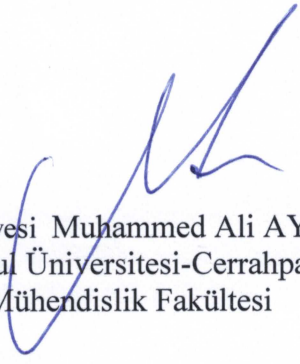
İSTANBUL-2019

Bu çalışma 31.05.2019 Tarihinde ařağıdaki jüri tarafından
Bilgisayar Mühendisliğı Anabilim Dalı, Bilgisayar Mühendisliğı Tezli Yüksek Lisans Programı
Yüksek Lisans Tezi olarak kabul edilmiştir.

TEZ JÜRİSİ


Prof. Dr. Ahmet SERTBAŐ
İstanbul Üniversitesi-Cerrahpaőa
Mühendislik Fakültesi


Doç. Dr. Yusuf YASLAN
İstanbul Teknik Üniversitesi
Bilgisayar ve Biliřim Fakültesi


Dr. Öğr. Üyesi Muhammed Ali AYDIN
İstanbul Üniversitesi-Cerrahpaőa
Mühendislik Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa’nın aboneliği olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Yüksek Lisans çalışmalarım süresince, bana yol gösteren bilgi ve desteğini esirgemeyen değerli hocam ve tez danışmanım Prof.Dr. Ahmet SERTBAŞ'a,

Çalışmalarım süresince yardımlarıyla beni sürekli destekleyen, bilgi ve birikimlerini benimle paylaşan Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN'a

Ayrıca bana desteklerini hiç bir zaman esirgemeyen, sabırları ve destekleriyle tüm öğretim hayatım boyunca her zaman yanımda olan çok sevdiğim aileme,

Eşime,

Sonsuz teşekkürler...

Mayıs 2019

Çağrı ÇINAR

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ.....	viii
SİMGE VE KISALTMA LİSTESİ	ix
ÖZET	x
SUMMARY	xi
1. GİRİŞ	1
2. GENEL KISIMLAR.....	2
2.1. MOBİL İŞLETİM SİSTEMLERİ.....	2
2.1.1. Android.....	3
2.1.2. IOS.....	4
2.2. MOBİL UYGULAMA GELİŞTİRME YAKLAŞIMLARI	6
2.2.1. Native Uygulama Geliştirme Yaklaşımları	6
2.2.2. Web Uygulama Geliştirme Yaklaşımları	7
2.2.3. Çoklu Platform (Hibrit) Uygulama Geliştirme Yaklaşımları	7
3. MALZEME VE YÖNTEM.....	9
3.1. ÇOKLU PLATFORM MOBİL UYGULAMA GELİŞTİRME ARACI SEÇME.....	9
3.2. DEĞERLENDİRME MATRİSİ	10
3.3. ÇOKLU PLATFORM MOBİL UYGULAMA GELİŞTİRME ARAÇLARI	11
3.3.1. PhoneGap	12
3.3.2. Titanium	13
3.3.3. QT.....	14
3.3.4. Xamarin	14
3.3.5. RhoMobile Suite.....	15
4. BULGULAR.....	16
4.1. DEĞERLENDİRME KRİTERLERİ	16
4.1.1. Son Kullanıcı Deneyimi	16
4.1.1.1. Etkileşim Ve Uyum	16

4.1.1.2.	<i>Desteklenen Platformlar</i>	16
4.1.1.3.	<i>Enerji Tüketimi</i>	16
4.1.2.	Geliştirme Deneyimi	17
4.1.2.1.	<i>Paylaşılan Kod</i>	17
4.1.2.2.	<i>Yerel İşlemlere Erişim</i>	17
4.1.2.3.	<i>Yetkinlik Durumu</i>	17
4.1.2.4.	<i>Tasarım</i>	17
4.1.3.	İş Deneyimi	18
4.1.3.1.	<i>Maliyet</i>	18
4.1.3.2.	<i>Fizibilite</i>	18
4.1.3.3.	<i>Yayınlama Ve Dağıtım</i>	18
4.1.3.4.	<i>Güvenlik</i>	18
4.2.	UYGULAMA SEÇİMİ	19
4.2.1.	Hedef Uygulamalar	19
4.2.1.1.	<i>Uygulama A</i>	19
4.2.1.2.	<i>Uygulama B</i>	21
4.2.1.3.	<i>Uygulama C</i>	23
4.2.2.	Değerlendirme Matrisi	25
5.	TARTIŞMA VE SONUÇ	29
	KAYNAKLAR	31
	ÖZGEÇMİŞ	34

ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: Akıllı telefon işletim sistemlerinin 2017-2023 arası pazar payları [1].	2
Şekil 2.2: Android sistem mimarisi [5].	3
Şekil 2.3: IOS işletim sistemi mimarisi [7].	5
Şekil 3.1: Oluşturulan Değerlendirme Matrisi.	10
Şekil 3.2: Research2guidance tarafından yapılan piyasa analizine göre 2013 yılında platformlar arası geliştirme aracı kullanıcılarının kullanım dağılımı [17].	12
Şekil 4.1: Uygulama A'nın iletişim modeli.	20
Şekil 4.2: Uygulama B'nin iletişim modeli.	22
Şekil 4.3: Uygulama C'nin iletişim modeli.	24
Şekil 4.4: Geliştirme yöntemi değerlendirme matrisi	26

TABLO LİSTESİ

	Sayfa No
Tablo 4.1: Uygulama A için kriter ağırlık değerleri.	20
Tablo 4.2: Uygulama B için kriter ağırlık değerleri.	22
Tablo 4.3: Uygulama C için kriter ağırlık değerleri.	24



SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

gb	: Gigabyte
mb	: Megabyte
s	: Saat
sn	: Saniye

Kısaltmalar **Açıklama**

API	: Uygulama Programlama Arayüzü
CSS	: Stil Şablonu
HTML	: Hiper Metin İşaretleme Dili
IDE	: Bütünleşik Geliştirme Ortamı
IOS	: iPhone İşletim Sistemi
MVC	: Model View Controller
OHA	: Open Handset Alliance
SDK	: Yazılım Geliştirme Kiti
UI	: Kullanıcı Arayüzü
VM	: Sanal Makina

ÖZET

YÜKSEK LİSANS TEZİ

ÇOKLU PLATFORM MOBİL UYGULAMALAR OLUŞTURMAK İÇİN UYGUN YAKLAŞIM SEÇME ANALİZİ

Çağrı ÇINAR

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ahmet SERTBAŞ

II. Danışman : Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

Hızla gelişen teknoloji sayesinde mobil uygulama geliştirme yaklaşımları sayısı gün geçtikçe artmaktadır. Mobil dünyada birçok akıllı telefon modeli ve bu cihazların üzerinde çalıştığı çeşitli platformlar vardır. Bu şartlar altında üretilen mobil uygulamaların birden fazla platformda çalışma zorunluluğu doğmuştur. Bu çalışmada, öncelikle mobil cihazlara ait platformlar hakkında bilgi verilerek çoklu platform geliştirme araçları detaylandırılmıştır. Sonrasında, mobil uygulama geliştirirken gereken ihtiyaçlar bir matriste toplanmış bu ihtiyaçlara ağırlık katsayıları verilmiştir. Oluşturulan matriste ağırlıklı ortalama yöntemiyle, geliştirilecek uygulamaya ait bir skor üretilerek geliştirme yaklaşımı seçilmiştir.

Mayıs 2019, 34 sayfa.

Anahtar kelimeler: Mobil uygulama geliştirme, çoklu platform, native, web

SUMMARY

M.Sc. THESIS

ANALYSIS OF CHOOSING THE RIGHT APPROACH FOR CREATE CROSS-PLATFORM MOBILE APPLICATIONS

Çağrı ÇINAR

**Istanbul University-Cerrahpasa
Institute of Graduate Studies
Department of Computer Engineering**

Supervisor : Prof. Dr. Ahmet SERBAŞ

Co-Supervisor : Assist. Prof. Dr. Gülsüm Zeynep GÜRKAŞ AYDIN

With rapidly evolving technology is increasing day by day the number of mobile application development approach. There are many smartphone models in the mobile world and the various platforms on which these devices work. Mobile applications produced under these conditions have to work on more than one platform. In this study, it was firstly provided information about the platforms of mobile devices and cross-platform development tools were detailed. Then, when developing mobile applications, the requirements are gathered in a matrix and weight parameters are given. The weighted average method was used to create a score for the application to be developed and the development approach was chosen.

May 2019, 34 pages.

Keywords: Mobile application development, cross-platform, native, web

1. GİRİŞ

Günümüzde mobil cihazlar hayatımızın her anında kullanılmaktadır. Neredeyse herkesin bir akıllı telefonu var ve akıllı telefonlar hayatımızı gün geçtikçe daha da kolaylaştırıyor.

Mobil dünyada birçok akıllı telefon modeli ve bu cihazların üzerinde çalıştığı çeşitli platformlar vardır. Ayrıca bu platformların birçok etkin sürümü bulunmaktadır. Cihaz üreticileri genellikle kendi özelleştirdikleri platformları kullanıcıya sunarlar. Her işletim sisteminin kendine ait programlama dili, cihaz arayüzü ve geliştirme araçları bulunmaktadır. Ancak her bir platform için uygulama geliştirmek maliyetli ve zaman alıcı bir iştir. Geliştiriciler için bu durum büyük problem teşkil etmektedir.

Bu problemin iki ana çözümü vardır. İlk çözüm her platformda yayınlamak istenen uygulamayı web uygulaması olarak geliştirmektir. Uygulama web sitesi olarak yayınlanıp her cihazın tarayıcısı ile uygulamaya erişilerek bu soruna çözüm sağlanabilir. İkinci çözüm çoklu platform mobil uygulama geliştirme yaklaşımı kullanmaktır. Bu yaklaşım aracılığıyla ortak yazılan kod ile her platforma özgü uygulamalar oluşturabilir. Hem ticari hem de açık kaynak kodlu birçok çoklu platform mobil uygulama geliştirme aracı bulunmaktadır.

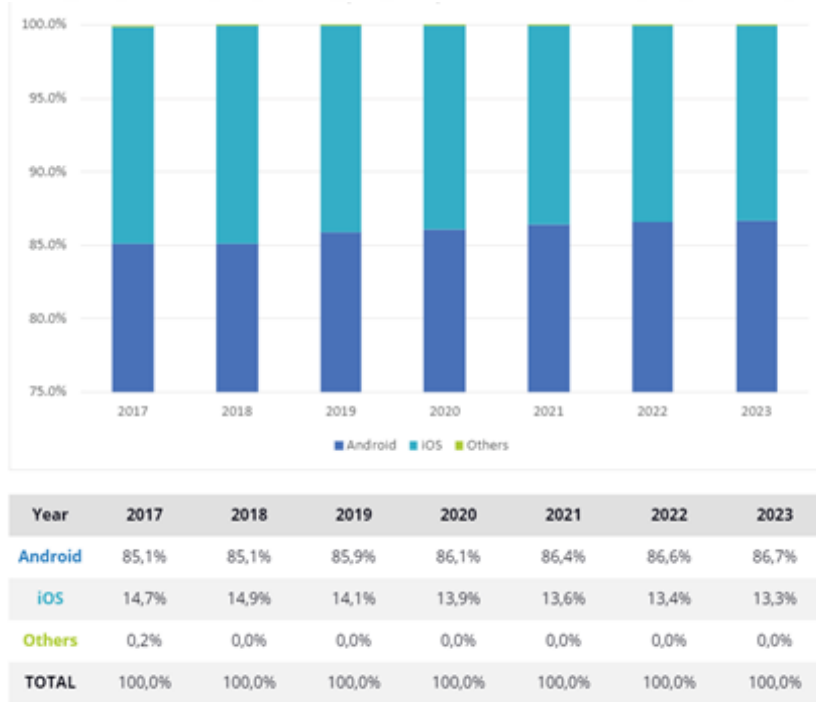
Ancak her platformda uygulama geliştirmek için hangi geliştirme yaklaşımını seçmeliyiz, bizim için en uygun koşulları hangi yaklaşım oluşturmaktadır? Bu tezin amacı çoklu platform uygulama geliştirme yaklaşımlarını tanımak ve hangi yöntemin seçileceğini belirleyen faktörleri bulmaktır.

Yapılan çalışmada öncelikle mobil işletim sistemlerinden ve mobil uygulama geliştirme yaklaşımlarından bahsedilmiştir. Sonrasında çoklu platform geliştirme araçları detaylandırılmıştır. Mobil uygulama geliştirirken son kullanıcı bakış açısı, geliştirme bakış açısı ve iş dünyası bakış açılarıyla çeşitli kriterler belirlenmiştir. Sonrasında, iki tane geliştirmesi tamamlanmış ve bir tane geliştirilmeye başlanmamış uygulama ile belirlenen bu kriterlere uygulamaların ihtiyaçlarına göre ağırlık değerleri verilerek bir puanaj sistemi oluşturulmuştur. Kriterlere verilen cevapların ağırlıklı ortalamaları alınarak farklı geliştirme yaklaşımları için farklı ortalama değerler bulunmuştur.

2. GENEL KISIMLAR

2.1.MOBİL İŞLETİM SİSTEMLERİ

Mobil işletim sistemleri, akıllı telefon ve tabletlerimizi kullanabilmemiz için geliştirilmiş platformlardır. Günümüzde mobil platform pazarı IOS ve Android olarak ikiye bölünmüş durumdadır. Bu iki platform dışında çok küçük oranda kullanılan farklı mobil platformlar bulunmaktadır. Fakat bu platformlar geçmişte pazarda söz sahibi olamamıştır ve Şekil 2.1'deki rapora göre önümüzdeki 5 yıllık süreçte de yine bu pazarda kendilerine yer bulamayacaklardır. Özellikle Microsoft bu pazarda kendine yer bulabilmek için 2010'yılında Windows Phone isimli mobil platformla pazara giriş yapsa da bu alanda başarı sağlayamamıştır. Nokia ortaklığında son olarak 2015 yılında çıkardığı Windows 10 Mobile platformundan da 2019 yılı sonlarında desteğini çekeceğini açıklamıştır. Sektörün lokomotif iki platform, IOS ve Android Şekil 2.1'deki yayınlana rapora göre geçmişte olduğu gibi önümüzdeki yıllarda da pazardaki yerini koruyacaktır.



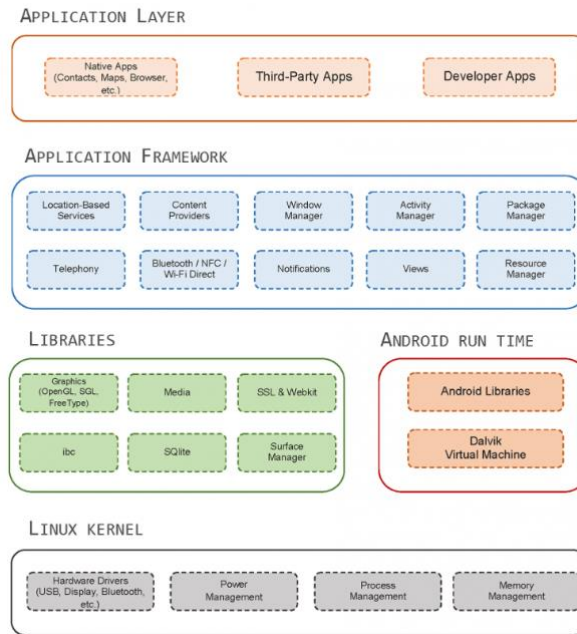
Şekil 2.1: Akıllı telefon işletim sistemlerinin 2017-2023 arası pazar payları [1].

2.1.1. Android

Android 2005 yılında Google tarafından satın alınan Android Inc. Şirketi tarafından kurulmuştur. [1] Google, Android'i satın aldıktan sonra telefon üreticileri ile birlikte Open Handset Alliance (OHA) birliğini kurmuştur. Android işletim sisteminin gelişimi OHA tarafından yönetilmektedir. OHA birliği içerisinde Samsung, LG, T-Mobile, Motorola, Intel, Sony, HTC, Dell ve daha birçok büyük firmalar bulunmaktadır. OHA birliğinin amacı Apple, Nokia gibi firmalara karşı ortak, standart bir platform oluşturmaktır. [2]

Android açık kaynak kodlu bir işletim sistemidir. Samsung, HTC ve LG gibi büyük telefon üretici firmalar genellikle mobil cihazları için Android üzerine geliştirilmiş kendi özel sürümlerini çıkarmaktadır. [3] Bu durum mobil cihaz kullanıcıları için çeşitlilik sunarken Android sürümünün güncel kalmasında tutarsızlıklar oluşturabilir. Çünkü satıcıların Android sürümünde yapılan güncellemeleri, kendi özel sürümlerine entegre etmeleri gerekir. Özellikle uygulama geliştiricilerinin bu durumu dikkate alması ve geliştirdikleri uygulamaların eski sürümlerle uyumluluğunu sağlamaları gerekmektedir. [4]

Android, beş katmana bölünmüş şekilde tasarlanmıştır. Şekil 2.2'de bu katmanlar gösterilmiştir.



Şekil 2.2: Android sistem mimarisi [5].

Android mimarisinde en alt katmanda Linux kernel katmanı bulunmaktadır. Android’de güvenlik, bellek yönetimi, işlem yönetimi, ağ ve sürücü modeli gibi hizmetler Linux 2.6 serisi bir çekirdeğe dayanmaktadır. Linux kernel katman, donanımsal mimari ile üst katmanlar arasında soyutlama sağlar ve ekran, kamera, tuş takımı gibi tüm temel donanım sürücülerini içerir. [5]

Uygulama katmanında, Android’in geliştirilmesine özgü Java tabanlı kütüphaneler bulunmaktadır. Bu kütüphaneler ile kullanıcı arayüzü oluşturma, grafik çizimi, veri tabanı erişimi gibi işlemler yapılmaktadır.[5]

Android runtime katmanında Dalvik Virtual Machine (DVM) yer alır. DVM minimum donanım yapılandırmasında, maksimum performans elde edebilecek Java yazılımlarını çalıştırmaktadır. [6] Application Framework katmanında, geliştiricilere Java sınıfları biçiminde birçok API sağlanır.

En üste seviye olan Application katmanında, tüm Android uygulamalarının bulunduğu katmandır. Mobil cihaz içindeki rehber, tarayıcı, oyunlar vb. yazılımlar bu katman üzerinde çalışmaktadır. [5]

Android uygulamaları genellikle Android Software Development Kit (SDK) ile Java programlama dili kullanılarak yazılır. SDK geliştiricilere sağladığı API’ler, dokümantasyon, örnek kodlar ve test ortamıyla yazılım geliştirme süreçlerini basitleştirir. Google geliştirme için Android Studio’yu önermesine rağmen farklı IDE’ler ile de yazılım geliştirme yapılabilir. [5]

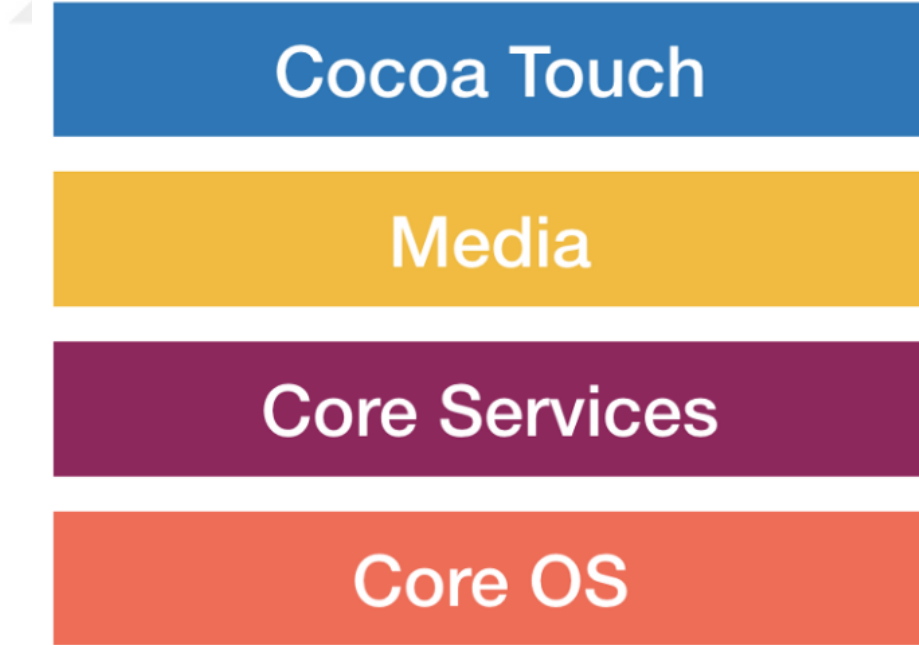
Android uygulamalarının resmi dağıtım kanalı eskiden Android Market olarak bilinen ve geliştiricilerin kayıt olduktan sonra uygulama yayınlatabildikleri Google Play Store’dur. [5] Android uygulamaları, geliştiricilerin kendi web sayfaları ve harici uygulama mağazalarından da ücretsiz olarak dağıtılabilir. [6]

2.1.2. IOS

IOS işletim sistemi ilk kez 2007 yılında iPhone ile birlikte piyasaya sürüldü. iPhone ile birlikte Apple’ın diğer cihazları olan iPad ve iPod için de kullanıma sunuldu. Android işletim sisteminde farklı olarak IOS’un tüm sürümleri Apple tarafından yayınlanır ve daha az sayıda sürümü bulunmaktadır. [5]

IOS sistem mimarisi Şekil 2.3'de gösterilen şekilde 4 katmana ayrılmıştır. Katmanlar uygulamalar ve temel donanım arasındaki farklı soyutlama seviyelerini sağlamaktadır. Temel frameworkleri Objective-C programlama diliyle yazılmıştır. Core OS katmanı, diğer teknolojilerin çoğunun üzerine inşa edilmiş düşük seviyeli özelliklerini içerir. Uygulamalar bu teknolojileri nadiren kullanır. Bununla birlikte bu katman güvenlik, bluetooth desteği ve harici donanımlar ile iletişime geçebilme gibi özelliklere sahiptir. Ayrıca katman, işletimi sisteminin çekirdek ortamını, sürücülerini ve düşük seviyeli UNIX arayüzlerini içerir. [6]

Uygulamalar tarafından kullanılan çeşitli sistem servisleri Core Services katmanında bulunmaktadır. Bu katman iCloud depolama, p2p servisler ve ağ hizmetlerini içeren teknolojileri barındırmaktadır. Bu katmanın hemen üzerinde bulunan Media katmanında uygulamalar tarafından kullanılan grafik, ses ve video teknolojileri bulunmaktadır. Son olarak en üst katmanda Cocoa Layer katmanı yer alır. Bu katman kullanıcı arayüzünü oluşturan katmandır. [7]



Şekil 2.3: IOS işletim sistemi mimarisi [7].

IOS'da uygulama geliřtirmek için Mac bilgisayar üzerinde bulunan Xcode IDE'si kullanılmaktadır. IOS uygulamaları genellikle Objective-C programlama dilinde yazılmıştır. [7]

Apple, IOS için geliřtirilen uygulamaların dağıtımını App Store aracılığıyla yapmaktadır. IOS'da uygulama geliřtirmek için Apple Geliřtici Programı'na kayıt olmak gerekmektedir. Geliřtirilen uygulamalar mağazaya çıkmadan önce onay sürecinden geçmektedir. Bu onay işlemi uygulama geliřtirme sürecini yavaşlatırken mağazadaki uygulama kalitesini arttırmaktadır. [7] Onay süreci çapraz platform yöntemleri ile uygulama geliřtirenler için daha zor olabilir. Örneğin 2010 yılında Apple uygulamaların mağazaya kabul edilebilmesi için orijinal olarak Objective-C, C, C++ ya da JavaScript ile yazılmış olması gerektiğini belirtmiştir.

2.2.MOBİL UYGULAMA GELİŐTİRME YAKLAŐIMLARI

Mobile uygulamalar entegre ve entegre olmayan uygulamalar olarak ikiye ayrılmaktadır. Entegre uygulamalar, cihaza kurulu bir paketten oluşur. Uygulama yüklendiğinde sisteme entegre edilir, uygulamalar listesinde gözükür ve sistem işlevlerine erişir. Entegre uygulamalar, platforma özgü teknolojileri ve kütüphaneleri kullanan native uygulamaları ve hibrit uygulama olarak biline web uygulamalarını içerir. Diğer yandan entegre olmayan uygulamalar doğrudan aygıt üzerinde çalışmaz. Bunun yerine genellikle web tarayıcısı olan başka bir uygulama aracılığıyla çağırılırlar. Web tarayıcısında çalışan uygulamalar web uygulaması olarak adlandırılır. [8]

2.2.1. Native Uygulama Geliřtirme Yaklaşımları

Native uygulama, üzerinde çalıştığı işletim sistemine özgü olarak üretilen uygulamalıdır. Bu tür uygulamalar üzerinde çalıştığı işletim sistemi ya da cihaza özel tasarlandığı için cihazın yazılımsal ve donanımsal kaynaklarına doğrudan erişebilir. Objective-C ya da Swift ile yazılmış IOS uygulamaları, Java ile yazılmış Android uygulamaları birer native uygulama örnekleridir. [9]

Eğer geliřtirici aynı uygulamayı birden fazla platformda yayınlamak istiyorsa her platforma özgü ayrı ayrı geliřtirme yapması gerekmektedir. Native uygulamalar ilgili platforma özgü mağazadan yüklenebilirler ve bu tür uygulamalar cihazın sunduğu tüm native özelliklere erişebilirler. [10]

Native uygulama geliřtirmek için birok framework bulunmaktadır. Bu frameworklerin kullanıcı arayüzü oluşturmak için kendi grafik editörleri bulunmaktadır. Böylece ortak UI kontrolleri kullanılmış olur ve her bir platformdaki kullanıcılar aynı görsel ve işlevsel olarak aynı deneyimi yaşamış olur. [11]

2.2.2. Web Uygulama Geliřtirme Yaklaşımları

Mobil web uygulamaları, standart web teknolojileri ve mobil platformların tarayıcı desteğinden yararlanırlar. Bu tarz uygulamalar aslında web sitelerinin mobil görünüm için optimize edilmiş halidir. Web sitesine herhangi bir mobil cihazdan erişilebilir ve platform bağımsız davranış sergiler. Aslında yazılan kodlar işlevsel olarak bir web sunucusunda saklanır, bu sunucu üzerinde çalışan web sitesi her bir platformda web tarayıcısı yardımıyla mobil uygulama olarak gözükür. Uygulama asıl olarak web sunucu üzerinden yönetildiği için yapılan geliřtirmelerin platform bağımsız sadece bu sunucu da değıştirilmesi bakımı kolaylaştırır. [10]

Bir web uygulaması geliřtirmek, web sitesi geliřtirmeye ok benzerdir. Web uygulamaları genellikle sunum, mantık ve veri katmanını içeren üç katmanlı mimariyi kullanır. Uygulamanın istemci tarafı sunum katmanını, sunucu tarafı ise mantık ve veri katmanını içerir. Son yıllarda her üç katmanın, tek katmanda geliřtirilmesine olanak sağlayan çözümlerde ortaya çıkmıştır. [12]

Web uygulamalarındaki en büyük dezavantaj, cihaz işlevlerine erişimin sınırlı olmasıdır. Web tarayıcısında çalıştıkları için işletim sisteminin sunduğu kamera, bluetooth, depolama gibi native özelliklere erişim sağlayamazlar. [11] HTML'in son sürümü olan HTML5, bu problemlerin çoğunu çözmeyi amaçlamaktadır. Örneğın, HTML5 kullanan web uygulamaları cihaz kamerasına erişim sağlayabilir. [13]

2.2.3. oklu Platform (Hibrit) Uygulama Geliřtirme Yaklaşımları

Daha önce incelenen herhangi bir kategoriye doğrudan girmeyen mobil uygulamalar oluşturmanın başka yöntemleri de vardır. Bunlar, yerel uygulamalar gibi cihazda çalışan, ancak bir platformlar arası geliřtirme aracı kullanarak birden fazla platform için ortak bir kod tabanından oluşturulmuş uygulamalardır. Bu araçlar, uygulama kodu ile yerel işlevsellik arasında farklı soyutlama katmanları sağlayarak geliřtiricinin ortak iş mantığını aygıtta özgü koddan ayırmasına olanak tanır. [14]

Ohrn ve Turau [8], platformlar arası araçlarla oluşturulan aşağıdaki dört tür uygulamayı tanımlamaktadır:

- Sistem API'sine doğrudan erişen yerel uygulamalar
- Yalnızca kütüphane tarafından sağlanan bir soyutlama ile sisteme erişen native uygulamalar
- Uygulama paketinde VM içeren uygulamalar
- VM'nin ayrı bir uygulama olarak kurulmasını gerektiren yorumlanmış uygulamalar.



3. MALZEME VE YÖNTEM

Mobil platformlar ve uygulamalar hakkındaki temel bilgiler literatür taraması yoluyla araştırılmıştır. Google Scholar ve Science Direct gibi arama motorları kullanılarak “mobil uygulamalar”, “mobil uygulama geliştirme”, “çoklu platform mobil uygulama geliştirme” gibi anahtar kelimelerle literatür taranmıştır. Platformlar hakkındaki bilgiler de platformların adlarıyla ayrı ayrı arama yapılarak ve geliştiricilerin web sayfaları incelenerek yapılmıştır. Uygulama türleri ve farklılıkları için “native uygulamalar“, “web uygulamaları” ve “çoklu platform uygulamalar” anahtar kelimeleri kullanılmıştır.

Yapılan araştırmalarda genel olarak iş dünyası bakış açısıyla oluşturulan modeller üzerinden bir çıkarım yapılmıştır. Bu çalışmada ise diğer çalışmalara ek olarak son kullanıcı, geliştirici ve iş dünyası bakış açılarıyla geliştirme sürecinin tüm aktörleri ele alınmıştır.

3.1.ÇOKLU PLATFORM MOBİL UYGULAMA GELİŞTİRME ARACI SEÇME

Literatür taraması sırasında, okunan makalelerde belirtilen çoklu platform uygulama geliştirme araçları Kısım 3.3’de listelenmiştir. Bu çalışmaya katkıda bulunmak için, Google’da "çoklu platform mobil geliştirme araçları" anahtar kelimelerini kullanarak arama yapmak ve herhangi bir ek araç varsa bunlar listeye eklenebilir. Her bir araç daha sonra ilgili aracın geliştiricisinin web sitesini ziyaret ederek ve aracın adıyla genel bir web taraması yaparak kısaca gözden geçirilmelidir. Geliştirme aracı en azından geçerli bir seçenek olarak kabul edilmesi için Android ve IOS platformlarını desteklemelidir. Geliştirme aracının aktif olarak gelişmiş bir durumda olması gerekir. Bu kriterleri yerine getiremeyen geliştirme araçları dikkate alınmamalıdır. Ardından, geliştirme araçları listesi en fazla beş en fazla geliştirme aracına daraltılır. Bu daraltma, aracın geliştiriciler arasındaki popülerliğine, desteklenen platformların ve donanım özelliklerinin miktarına ve aracın ömrüne göre yapılır. Seçilen araçlar farklı teknolojik yaklaşımları da temsil etmelidir. Bu yöntemle belirlenen beş araçtan bu çalışmada bahsedilmektedir.

3.2.DEĞERLENDİRME MATRİSİ

Karar verme işlemi bu çalışmanın önemli bir adımındır. Kararın alınmasından uzun bir süre sonra başkalarının karar vermenizin nedenlerini anlamalarını sağlamak için kararları açık ve yapılandırılmış yollarla belgelendirmek de önemlidir.

Karar verirken göz önünde bulundurulması gereken birçok farklı kriter vardır. Kriterlerin tanımlanması ve bu kriterlere ilişkin kararın mümkün olduğunca kesin olarak verilmesi esastır. Her bir kriter maksimize edilemez ve bir kriteri maksimize etmek, muhtemelen diğer önemli kriterleri yerine getirmeyi sınırlar. Yani, en iyi tasarımı oluşturmak genellikle bir tasarım değişimini gerektirir. Seçilebilecek birçok alternatif olduğunda, kriter daha da ağırlaştırılır. Bu gibi durumlarda, sadece her alternatifin incelenmesi gerekmez, tüm alternatiflerin nihai bir karşılaştırmasının doğru olmasını sağlamak için tüm alternatiflerin tutarlı bir şekilde ele alınması gerekir.

Ağırlıklı bir değerlendirme matrisi karmaşık kararların alınmasında çok yararlı olabilecek bir araçtır. Özellikle de çok sayıda alternatifin olduğu ve dikkate alınması gereken çeşitli önemli kriterlerin olduğu durumlarda bir gereklilik haline gelmektedir. Ağırlıklı değerlendirme matrisleri genellikle tasarım mühendisliğinde alternatifleri değerlendirmek için niteleyici bir araç olarak kullanılır.

Yapılan çalışmada oluşturulan değerlendirme matrisi Şekil 3.1’de detaylandırılmıştır.

Kriter	Geliştirme Araçları			Ağırlık Değerleri
	Metod 1	...	Metod m	Uygulama
Kriter 1	X_{11}	...	X_{1m}	Y_1
Kriter 2	X_{21}	...	X_{2m}	Y_2
....
Kriter n	X_{n1}	...	X_{nm}	Y_n
Toplam	$TOPLA(X_{11}, \dots, X_{n1})$...	$TOPLA(X_{1m}, \dots, X_{nm})$	
Ortalama	$ORTALAMA(X_{11}, \dots, X_{n1})$...	$ORTALAMA(X_{1m}, \dots, X_{nm})$	
Ağırlıklı Ortalama				
Uygulama A	$TOPLA(X_{11} * Y_1, \dots, X_{n1} * Y_n) / TOPLA(Y_1, \dots, Y_n)$...	$TOPLA(X_{1m} * Y_1, \dots, X_{nm} * Y_n) / TOPLA(Y_1, \dots, Y_n)$	

Şekil 3.1 Oluşturulan Değerlendirme Matrisi.

3.3. ÇOKLU PLATFORM MOBİL UYGULAMA GELİŞTİRME ARAÇLARI

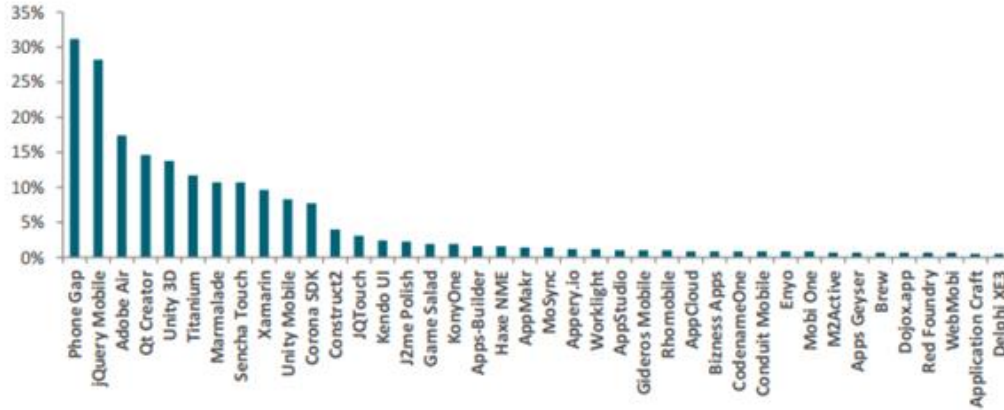
Native uygulama geliştirme araçlarıyla, farklı platformlar için mobil uygulamalar geliştirmek, genellikle uygulama geliştirme döngüsünün her bir platform için tekrarlanması anlamına gelir. Her platformun kendine özel cihazları, programlama dilleri, API'leri ve dağıtım mağazaları vardır ve bu sebeple birçok farklı beceriye ihtiyaç duyulmaktadır. Mobil cihaz kullanıcı dağılımı, farklı platformlar arasında güçlü bir şekilde bölünmüştür ve hangi platformlar için geliştirme yapılacağını seçmek, uygulama için potansiyel müşterilerin sayısını belirler. [15]

Yıllar içinde bu soruna farklı çözümler üretildi. En popüler olanlarından biri, uygulamayı platformun native tarayıcısında çalışan bir web uygulaması olarak tasarlamaktır. Fakat bu tarayıcılar cihaz donanımına sınırlı erişime sahiptir ve native uygulamalara göre daha düşük bir performansa ve kullanıcı arayüzüne sahiptir. Platformlar arası geliştirme araçları, farklı platformlar için uygulamaları ortak bir kod tabanından paketlemeye yönelik bir yol izleyerek bu sorunu çözmeye çalışır. Tasarım ve geliştirme sadece bir kez yapılır, tek bir kod tabanı oluşturur. Ardından geliştirme aracından hedefe özgü özelleştirmeler seçilerek desteklenen herhangi bir platforma dağıtılabilir. Platformlar arası geliştirme araçları genellikle, paylaşılan uygulama kodu ile hedef platformun yerel API'si arasında bir soyutlama katmanına dayanır, ancak teknik uygulamalar ve ayrıntılar farklı araçlar arasında farklılık gösterir. Ayrıca farklı araçlar birçok farklı beceriye ve kaynağa gereksinim duyar. [15]

Platformlar arası geliştirme araçları, kullandıkları işlevsellik düzeyine bağlı olarak üç gruba ayrılabilir: kütüphaneler, frameworkler ve platformlar: [16]

- Kütüphaneler, 3D grafikler veya GUI kontrolleri gibi belirli bir konuda geliştirici işlevselliğini sağlayan küçük bağımsız araçlardır. Tam mobil uygulamayı oluşturmak için genellikle diğer kütüphaneler ve araçlar ile birlikte kullanılır.
- Frameworkler, mobil uygulamaların baştan sona inşa edilmesi için çoklu kütüphaneler, yazılım bileşenleri ve mimari yönergeler içerir.
- Platformlar, geliştiricinin mobil uygulamalar oluşturmasına ve dağıtmasına izin veren bir dizi framework, araç ve hizmetten oluşur. Uygulamanın hatalarını ayıklama, yapılandırma ve paketleme işlevlerini kullanır ve genellikle kendi IDE'leri ile birlikte belgeler, destek ve otomasyon araçlarını sunarlar.

Şekil 3.2, farklı platformlar arası geliştirme araçlarının 2013 yılında pazar paylarını göstermektedir. [17] Kullanılabilecek onlarca araç olmasına rağmen, sadece birkaç aracın geniş kullanım alanı vardır ve bu çalışmanın odak noktası olacaktır. Ayrıca, Unity 3D ve Marmalade gibi popüler araçlardan bazıları, oyun geliştirmek için tasarlanmış olup, endüstriyel uygulamalar için uygun değildir.



Şekil 3.2 Research2guidance tarafından yapılan piyasa analizine göre 2013 yılında platformlar arası geliştirme aracı kullanıcılarının kullanım dağılımı [17].

3.3.1. PhoneGap

PhoneGap, aslen 2011 yılında Adobe Systems tarafından satın alınan Nitobi Software tarafından geliştirilmiştir. Apache License Version 2.0 altında açık kaynaklı bir çapraz platform mobil geliştirme frameworküdür. PhoneGap, ücretsiz, açık kaynaklı veya ticari uygulamalar geliştirmek için kullanılabilir. PhoneGap, Android, iOS, BlackBerry ve Windows Phone dahil olmak üzere çok çeşitli mobil platformları desteklemektedir. [18]

PhoneGap, geliştiricinin, kullanıcı arayüzü için HTML ve CSS kullanmasına, uygulama mantığı için ise JavaScript kullanımına olanak sağlar. Ve böylece, native bir uygulamanın WebView'inde çalışan karma bir uygulama ortaya çıkar. UI, web teknolojileri kullanılarak yazıldığı için, uygulama bir web sayfası görünümü ve havasında olacaktır, native bir görünüme sahip olması için native bir uygulamanın platform özellikli CSS dosyalarını kullanması ve kullanıcı arayüzünü taklit etmesi gerekir. [18] PhoneGap, UI deneyimi geliştirmek için diğer geliştirme frameworkleriyle birlikte de kullanılabilir. PhoneGap, native UI bileşenlerini

desteklemediğinden, UI'yi oluşturmak için JQuery Mobile veya Sencha Touch gibi araçlar kullanılır. [19]

Uygulamanın çalıştığı cihazın donanım fonksiyonlarına, PhoneGap'ın motoru tarafından sağlanan bir JavaScript API'si aracılığıyla erişilir. PhoneGap, Bluetooth gibi ek özelliklere erişim sağlayan eklentilerle de genişletilebilir [20]. PhoneGap açık kaynaklı bir yazılım olduğundan, geliştiriciler de kendi eklentilerini ve özelliklerini geliştirebilirler. [18]

PhoneGap, özel bir IDE ile birlikte gelmez, böylece geliştirici temel kodu yazmak için herhangi bir IDE kullanabilir. Kod, daha sonra oluşturmak için hedef platformun kendi IDE'sine veya Cordova komut satırı arayüzü kullanılarak oluşturulabilir. PhoneGap, uygulamayı oluşturmak için hedef platformun native SDK'sına ihtiyaç duyar. Adobe ayrıca, bulutta uygulamalar oluşturmak için PhoneGap derleme hizmetini de sunar. Bu geliştiricinin native SDK'yı yüklemeyen uygulamalar oluşturmasını sağlar. [21]

3.3.2. Titanium

Titanyum, Appcelerator tarafından geliştirilen ve Apache Public License Version 2.0 altında lisanslanan açık kaynak kodlu bir uygulama frameworküdür. Android, iOS, Windows Phone ve BlackBerry dahil olmak üzere tüm ana mobil platformları destekler. [22]

Titanium, platformlar arası uygulamalara çoğu karma uygulama araçlarından farklı bir yaklaşım sunmaktadır. HTML ve CSS yerine, uygulamalar tamamen JavaScript ile Titanium API kullanılarak yazılmıştır. Kod, çalışma zamanında kodu yorumlayan ve kullanıcı arayüzünü sağlayan Titanium'un motoruyla birlikte gelir. Bu UI-API sayesinde native öğelerden oluşan uygulamalar native uygulamalar gibi görünür. [10]

Appcelerator, Titanium Studio olarak adlandırılan ve uygulama geliştirmek için kullanılan ücretsiz bir IDE sunar. IDE tüm büyük işletim sistemlerinde çalışır: Windows, Mac OS ve Linux. Uygulamaların oluşturulması için native SDK'nın sisteme yüklenmesi gerekir. Bu nedenle IOS uygulamaları oluşturmak için bir Mac OS aygıtı gerekir. [22]

3.3.3. QT

Qt, bir Digia iştiraki olan Qt Company tarafından yürütülen bir uygulama frameworküdür. İlk olarak 1995 yılında Trolltech tarafından piyasaya sürüldü ve daha sonra Nokia tarafından 2008 yılında satın alındı. Nokia, 2011 yılında Qt ticari lisansını Digia'ya sattı ve Digia, 2014'te Qt'nin telif haklarını Qt Company'ye devretti. Qt, geleneksel olarak masaüstü ve gömülü sistemler ile ilişkilendirilmiştir, ancak Qt 5.2 sürümü ile birlikte Android, iOS ve BlackBerry uygulamaları oluşturmak için desteğini eklemiştir. [23]

Uygulamalar, Qt'nun standart C ++ dilinde yazılmış ve Qt'nin makrolarıyla zenginleştirilmiştir. Bu, mevcut masaüstü uygulamalarını Qt kullanarak mobil platformlara taşımayı kolaylaştırır, çünkü masaüstü uygulamaları genellikle C ++ ile yazılır ve bazı iş mantıkları olduğu gibi yeniden kullanılabilir. JavaScript, Python ve QML dâhil olmak üzere diğer programlama dillerini kullanmak için bazı özellikler de mevcuttur, ancak bunların tüm özellikleri desteklenmemektedir. [23]

3.3.4. Xamarin

Xamarin, 2001 yılında Ximian tarafından başlatılan Mono açık kaynaklı projesine dayanan ticari bir uygulama platformudur. Ximian, 2003 yılında Novell ve ardından 2011 yılında Attachmate tarafından satın alındı. Attachmate'in satın alınmasından sonra, Mono üzerinde çalışan orijinal ekibin bir kısmı, Mono projesini devralacak olan Xamarin'i kurdu. Xamarin şu anda iOS, Android ve Windows Phone'u desteklemektedir. [24]

Xamarin'in ana programlama dili C #'tır ve Xamarin 3'ün piyasaya sürülmesiyle F# desteği de eklenmiştir. Kodu çoğu platformlar arasında paylaşılabilir ancak UI kodu genellikle standart native işlevleri kullanan ve platformun kullanıcı arayüzü kurallarına uygun olarak her platform için ayrı ayrı yapılır. UI, program aracılığıyla veya Xamarin'in hedef platform için grafik UI tasarımcısı kullanılarak yapılabilir. Xamarin 3, ayrıca UI tasarımının bir kısmının platformdan bağımsız olarak yapılmasına izin veren Xamarin.Forms'i de tanıttı. Bu şekilde ekrandaki veriler onu sağlayan koddan ayrılır ve yalnızca arayüz işleyiciler platforma göre değişir. [25]

Xamarin, geliştiricilere kendi bağımsız IDE'si Xamarin Studio'yu sunar. Alternatif olarak, Xamarin, Microsoft'un Visual Studio'sına entegre edilebilir. Uygulamaları oluşturmak için native SDK'lara ihtiyaç duyar. [24]

3.3.5. RhoMobile Suite

RhoMobile Suite, Őu anda Motorola Solutions tarafından geliŐtirilen ticari bir uygulama platformudur. Aslen 2009'da RhoMobile tarafından yayınlanan aŐık kaynak kodlu Rodos frameworküne dayanıyor. Rhodes frameworkü MIT lisansı altında mevcuttur. [26] Rodos frameworkünün RhoElements adı verilen geniŐletilmiŐ bir sürümünü iŐeren RhoMobile Suite ticari kullanım amaŐıdır. RhoMobile Suite, ücretsiz olarak veya ek özellikler ve kullanıcı desteĐi iŐeren ücretli bir kurumsal sürüm olarak kullanılabilen bir abonelik sistemine sahiptir. RhoMobile, iOS, Android, Windows Phone ve masaüstü Windows'ları destekler. [27]

RhoMobile, uygulamalar geliŐtirmek iŐin Model View Controller (MVC) modelini kullanır. Veri tanımlarını iŐ mantıĐından arayüzlerden ayırır ve bu üç elemanı birbirine baĐlayan noktaları saĐlar. HTML, CSS ve JavaScript kullanılarak yapılır ve Controllerlar iŐ mantıĐını yöneten sunucu taraflı kod iŐin Ruby kullanır. [28]

RhoMobile Suite, RhoStudio adı verilen kendi IDE'sini iŐerir. RhoStudio Eclipse tabanlıdır ve standart Eclipse eklentilerini kullanabilir. RhoMobile ayrıca Visual Studio'ya da entegre edilebilir. Uygulamalar oluŐturmak iŐin, native SDK'nın cihaza kurulması gerekir, ancak uygulamalar RhoHub kullanılarak uzaktan da derlenebilir ve oluŐturulabilir. [29]

4. BULGULAR

4.1. DEĞERLENDİRME KRİTERLERİ

Bu bölümde, mobil uygulama geliştirmek için farklı yöntemleri karşılaştırma kriterleri incelenecektir. Bu karşılaştırmada kullanılan kriterler, literatürdeki diğer araştırmaları inceleyerek toplanmıştır. [10] Kriterler üç ana kategoride ele alınacaktır; son kullanıcı deneyimi, geliştirme deneyimi ve iş deneyimi.

4.1.1. Son Kullanıcı Deneyimi

Son kullanıcı, bir yazılım veya donanımın son halini kullanan kişilerdir. Başka bir deyişle, bilgisayar yazılımı konusunda çok bilgi sahibi olmayan, uygulama geliştirme yeteneğine sahip olmayan kişiler son kullanıcı olarak nitelendirilirler. Bugün üretilen yazılımların geneli, son kullanıcılar düşünülerek geliştirilir. Son kullanıcının bakış açısına göre, mobil uygulamalar geliştirme yöntemlerinden bağımsız olarak native uygulamalar gibi davranmaları beklenir. Bu nedenle, geliştirme yönteminin yerel kullanıcı deneyimiyle ne kadar iyi eşleşebileceği ve bu amacı gerçekleştirmek için ne kadar ek çalışmanın gerekli olduğu önemli bir kriterdir. Son kullanıcının bakış açısına göre elde edilen kriterler aşağıda belirtilmiştir.

4.1.1.1. Etkileşim Ve Uyum

Uygulamaların genel hızını ve performansını ifade eder. Başka bir deyişle, uygulamanın girdilere ne kadar hızlı yanıt verdiğini ve ekranlar arasındaki geçişi ve dokunmatik tabanlı etkileşim modelini ne kadar iyi kullandığını belirler.

4.1.1.2. Desteklenen Platformlar

Bu kriter, metot için mevcut olan platformların çeşitliliğini ve bu metotla çoklu platformları desteklemenin ne kadar mümkün olduğunu belirler.

4.1.1.3. Enerji Tüketimi

Uygulama verimsizliği ve mobil veri kullanımı, cihazın enerji tüketimini artıran başlıca etkenlerdir.

4.1.2. Geliştirme Deneyimi

Bu kriter, yöntem seçimini, geliştirme hızını ve uzman geliştiricilerin kullanılabilirliğini belirler. Genellikle, native uygulamalar geliştirilirken geliştirme işleminin her hedef platform için ayrı ayrı yapılması gerekir ve ek bir platformun desteklenmesi çalışmanın tekrarlanmasını gerektirebilir. IOS ve ANDROID platformları için ayrı ayrı geliştirme yapmak bu duruma örnek olarak gösterilebilir. Çapraz platform geliştirme çözümü kullanarak, geliştirme maliyetlerinin bir kısmı platformlar arasında paylaşılabilir ve uygulamanın sadece küçük bir kısmının farklı platformlar için ayrı olarak programlanması sağlanabilir. Her platform için ayrı ayrı yapılması gereken en yaygın kısım UI tasarımıdır. Ayrıca, programlama dilleri ve araçlarının ihtiyaç duyduğu ek bilgiler de incelenir, çünkü geliştiricilerin belirli bir aracı kullanmak için eğitilmesi geliştirme süresini uzatır. Bu kriterler aşağıda detaylandırılmıştır.

4.1.2.1. Paylaşılan Kod

Aynı kodu birden fazla platform için tekrar kullanmak geliştirme süresini önemli ölçüde azaltır ve diğer platformlarda geliştirme yapmayı kolaylaştırır. Bu kriter, kodlama mimarisinin ne kadarının farklı platformlar arasında ortak olabileceğini inceler.

4.1.2.2. Yerel İşlemlere Erişim

Platformlar arası geliştirme araçları, geliştiricilerin, cihazın yerel işlevsellik platformuna bağımsız olarak erişmek için özel API'ler kullanmaktadır. Bu nedenle, mevcut yerel işlevlerin çeşitliliği ve kabiliyeti bu API'ye göre değişiklik gösterir.

4.1.2.3. Yetkinlik Durumu

Yaygın olmayan programlama dilleri kullanan veya kapsamlı bir çerçeveye özgü bilgi gerektiren bir geliştirme aracı seçmek, mevcut nitelikli geliştiricilerin sayısını sınırlar. Geliştiricilere verilecek eğitim ek maliyet doğurur ve geliştirme sürecini yavaşlatır.

4.1.2.4. Tasarım

Tasarım ve prototip oluşturma işlemi farklı yöntemler arasında farklılık gösterir. Hızlı prototip oluşturma geliştirme sürecini iyileştirir ve geliştiricilerin uygulama için fikir ve planları kolayca paylaşmalarını sağlar.

4.1.3. İş Deneyimi

Bir geliştirme aracını, birden fazla platform için bir uygulama geliştirmek için kullanmanın finansal sonuçları ve belirli bir süre boyunca uygulamaya bakım maliyeti gibi etkileri olmaktadır. Uygulamayı ve uygulama güncellemelerini son kullanıcı için yüklemenin kolaylığı da önemli bir etkidir. Ayrıca, geliştirme yöntemiyle ilgili tüm güvenlik sorunları ihmal edilmemelidir.

4.1.3.1. Maliyet

Çoklu platform için uygulama geliştirilmesinin toplam maliyeti dikkate alınmalıdır. Bu geliştirmelerin yapılacağı araçların maliyeti ve bu araçlara ait aylık ve yıllık abonelik ücretleri de geliştirme maliyetinde önemli yere sahiptir.

4.1.3.2. Fizibilite

Endüstriyel kullanımdaki bir mobil uygulamanın, mobil uygulamaların ortalama olduğundan daha uzun bir ömre sahip olması gerekmektedir. Uygulamanın belirli bir araçla geliştirilmesi, aynı araçla desteklenmesi ve güncellenmesi gerekmektedir, çünkü başka bir araçla geliştirilmesi genellikle mümkün olmamakta veya ağır miktarda iş yükü gerektirmektedir.

4.1.3.3. Yayınlama Ve Dağıtım

Geliştirilen uygulama belirli bir yayınlanma sürecinden geçmesi gerekmektedir. Öyle ki, bir güncelleme yayınlanıyorsa mevcut çalışan uygulama sürümü bu güncellemeden olumsuz etkilenmemelidir. Geliştirilen uygulamanın son kullanıcılara dağıtılmasının kolaylığı ve süresi de önemli bir faktördür.

4.1.3.4. Güvenlik

Endüstriyel uygulamaların büyük ölçekli sistemlerde kritik bir rol oynaması ve hassas bilgilere erişimi olması dolayısıyla uygulama geliştirme süreçlerinde güvenlik çok önemli bir kriterdir.

4.2. UYGULAMA SEÇİMİ

Önceki bölümde, mobil uygulama geliştirme yöntemlerinin her birinin güçlü ve zayıf yönleri olduğunu ve farklı türdeki uygulamalara uygun oldukları gösterilmiştir. Sayısal değerlendirme, geliştirme yöntemlerinin her biri için çok yakın ortalama puanlar vermiştir, ancak her bir kriterler için puanlar geliştirme yöntemleri arasındaki farklılıkları göstermektedir. Bu arada, her bir uygulamanın gereklilikleri de farklıdır; çünkü farklı uygulamalar farklı özellikler için farklı ihtiyaçlara sahiptir. Örneğin, cihazın kamera, depolama veya kablosuz iletişim gibi donanım işlevlerine olan ihtiyaç tamamen uygulamanın kullanım amacına bağlıdır. Bu nedenle, seçim sürecinin hedef uygulamayı göz önünde bulundurması gerekir. Bu, kriterlerin her biri için ağırlık değerleri belirlenerek elde edilmelidir. Bu ağırlık değerleri 1 ile 3 arasında önem derecesine göre değer almaktadır. Sonrasında ağırlıkları ölçütler için çarpanlar olarak kullanılabilir ve her bir geliştirme yöntemi için ağırlıklı ortalama hesaplanabilir. Sonuçlar, söz konusu uygulama için hangi geliştirme yönteminin en iyi seçim olduğunu belirleyecektir. Bu özelleştirilmiş matris, geliştirme aracı değerlendirme matrisi olarak adlandırılacaktır ve Şekil 4.4'te gösterilmiştir.

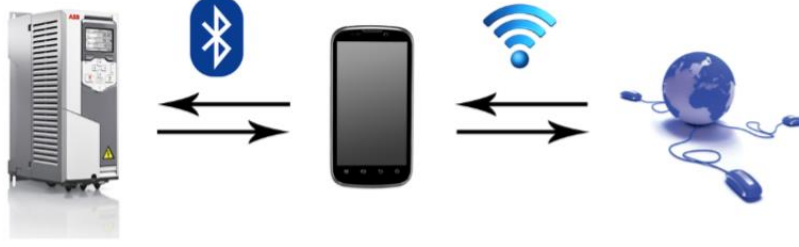
4.2.1. Hedef Uygulamalar

Yapılan çalışmada seçim süreci için üç farklı mobil uygulama belirlenmiştir. Uygulamaların ikisi hali hazırda mevcut ve geliştirmesi tamamlanmış olup, değerlendirme matrisi tarafından önerilen geliştirme yöntemini, gerçekte uygulamanın geliştirilmesinde kullanılan yöntemle karşılaştırarak değerlendirme matrisini doğrulamak için örnek teşkil edecektir. Üçüncü uygulama henüz geliştirme aşamasında değildir ve değerlendirme matrisi bunun için en iyi geliştirme yöntemini bulmak için kullanılacaktır. Uygulamaların özellikleri ve kullanım amaçları, uygulama için farklı kriterlerin önemini belirleyecektir. Uygulamaların asıl adları bu çalışmanın konusu değildir.

4.2.1.1. Uygulama A

Uygulama, hem müşteriler hem de çalışanlar tarafından sürücülerini yönetmek ve kontrol etmek için kullanılır. Uygulama sürücüsüyle iki yönlü bir Bluetooth bağlantısı üzerinden iletişim kurar ve hata analizi için veri toplar ve sürücünün parametrelerini ve durumunu ayarlamak için kullanılır. Ayrıca web kaynakları ve veri tabanlarıyla iletişim kurar. Şekil 4.1, uygulamanın iletişim yöntemlerini göstermektedir. Uygulama, çok çeşitli özelliklere sahip sürücülerini

yönetmek için eksiksiz bir paket olarak tasarlanmıştır ve güncelleştirmelerle daha fazlası eklenecektir. Uygulama Android ve iOS için ayrı geliştirme araçlarıyla native bir uygulama olarak geliştirilmiştir. Uygulama Bluetooth desteği gerektirdiğinden, bir web uygulaması şu anda bunun için olası bir seçenek olamamaktadır. Bu nedenle, A uygulaması için web tabanlı bir çözüm olası değildir.



Şekil 4.1: Uygulama A'nın iletişim modeli.

A uygulaması için kriter ağırlık değerleri ve nedenleri Tablo 4.1'de belirlenmiştir. Ağırlık değerleri uygulamanın sahipleriyle yapılan görüşmelerce toplanmıştır. Görüşülen kişilere her bir kriter için neden ve ne kadar önemli olduğu sorulmuştur.

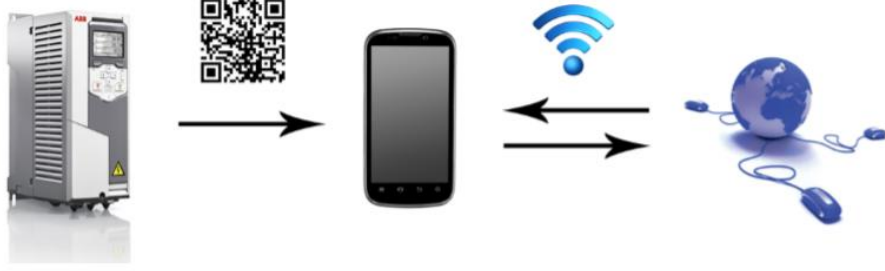
Tablo 4.1: Uygulama A için kriter ağırlık değerleri.

<p>HEDEF PLATFORMLA UI TUTARLILIĞI</p> <p>Uygulama mümkün olan genel platform kurallarına uymalıdır, ancak düzen ve görsel özellikler platformlar arasında aynı olacaktır.</p>	2
<p>ETKİLEŞİM VE DUYARLILIK</p> <p>İyi bir kullanıcı deneyimi ana önceliklerden birisidir.</p>	3
<p>DESTEKLENEN PLATFORMLAR</p> <p>Uygulamanın IOS ve ANDROID platformlarını desteklemesi gerekmektedir. Ayrıca, uygulama hem cep telefonlarını hem de tabletleri desteklemelidir.</p>	2
<p>ENERJİ TÜKETİMİ</p> <p>Enerji tüketimi bir sorun oluşturmamaktadır.</p>	1

PAYLAŞILAN KOD Kod, mümkün olduğu kadar platformlar arasında paylaşılmalıdır.	3
YEREL İŞLEVLERE ERİŞİM Kamera, GPS ve Bluetooth kaynakları kullanılmalıdır.	3
YETKİNLİK Geliştirme, uzmanlar tarafından yapıldığı için herhangi bir ek eğitime gerek duyulmamaktadır. Geliştiriciler, geliştirme araçları kullanımında tecrübeli olmalıdır.	2
TASARIM Uygulama "AGILE" geliştirme yöntemine uyumlu olmalıdır. Uygulama tasarımları hızlı değişeceğinden dolayı hızlı bir şekilde tasarım yapılabilenlidir.	3
MALİYET Diğer kriterler yerine getirilecek şekilde, uygun bir maliyete sahip olmalı.	2
FİZİBİLİTE Uygulama mümkün olduğu kadar uzun süre boyunca desteklenebilecek şekilde olmalıdır.	2
YAYINLAMA VE DAĞITIM Müşteri uygulamaya kolayca erişip yükleyebilmelidir. Uygulamanın düzenli olarak güncellenecektir.	2
GÜVENLİK Uygulama sürücüyü doğrudan kontrol etmek için kullanılabilir. Bu durumun herhangi bir güvenlik zafiyeti oluşturmaması gerekmektedir.	2

4.2.1.2.Uygulama B

Bir müşteri tarafından cihazlarını kaydettirmek ve destek almak için kullanılmaktadır. Kullanıcı, cihazın kamerasını kullanarak sürücünün kimlik numarasını bir barkoddan veya bir QR kodundan tarayabilir ve ardından cihazı kaydedebilir. Uygulama ayrıca kılavuzlar ve müşteri destek iletişim bilgileri dahil olmak üzere çevrimiçi kaynaklara bağlantılar vermektedir. PhoneGap kullanarak Android ve iOS için çoklu platform uygulama olarak geliştirilmiştir. Şekil 4.2, uygulamanın iletişim modelini göstermektedir.



Şekil 4.2: Uygulama B'nin iletişim modeli.

B uygulaması için kriter ağırlık değerleri ve nedenleri Tablo 4.2'de belirlenmiştir. Ağırlık değerleri uygulamanın sahipleriyle yapılan görüşmelerce toplanmıştır. Görüşülen kişilere her bir kriter için neden ve ne kadar önemli olduğu sorulmuştur.

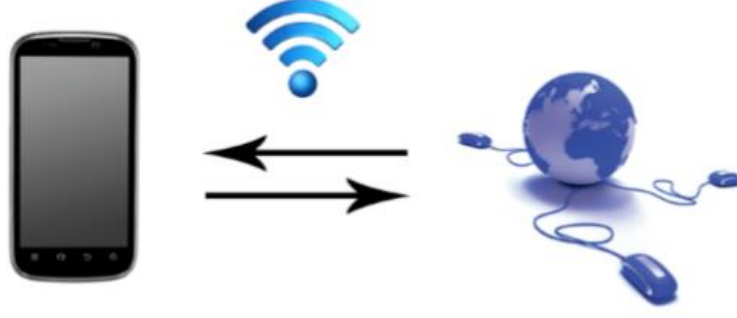
Tablo 4.2: Uygulama B için kriter ağırlık değerleri.

HEDEF PLATFORMLA UI TUTARLILIĞI Uygulamanın basit sade bir arayüzü olmalı.	2
ETKİLEŞİM VE DUYARLILIK İyi bir kullanıcı deneyimi ana önceliklerden birisidir.	3
DESTEKLENEN PLATFORMLAR Uygulamanın IOS ve ANDROID platformlarını desteklemesi gerekmektedir.	2
ENERJİ TÜKETİMİ Enerji tüketimi bir sorun teşkil etmiyor.	1
PAYLAŞILAN KOD Bazı kodlar platformlar arasında paylaşılabilir. Ancak çok da önemli değil.	1
YEREL İŞLEVLERE ERİŞİM Kamera, GPS ve Bluetooth kaynakları kullanılmalıdır.	3
YETKİNLİK Geliştirmeyi bilen uzmanlar zaten yetkin olduğu varsayılır.	1

TASARIM Tasarım ve geliştirme süreci yüksek öncelik değildir.	1
MALİYET Geliştirme maliyeti uygun olmalıdır.	2
FİZİBİLİTE Başvuru birkaç yıl boyunca aktif kalmalıdır.	3
YAYINLAMA VE DAĞITIM Müşteri uygulamaya kolayca erişip yükleyebilmelidir. Uygulamanın düzenli olarak güncellenecektir. Uygulama, kullanım kılavuzları ve yönergeler gibi büyük dosyaları depolamak için harici kaynaklar kullanmaktadır. Bu kaynaklar bir web sunucusunda depolanır ve uygulama paketine dahil edilmez	3
GÜVENLİK Uygulamanın her hesap için kimlik doğrulaması yapması ve kullanıcı bilgilerini gizli tutması gerekmektedir.	3

4.2.1.3.Uygulama C

Yapılan çalışmada geliştirme yöntemi seçmede kullanacağımız uygulamadır. Bu uygulama yeni bir projedir ve geliştirme aracı değerlendirme matrisinin sonucunda, bu uygulama için, bir geliştirme yöntemi önerisi oluşturulacaktır. Uygulama, sunulan mevcut bir web tabanlı hizmetin mobil sürümüdür. Kullanıcı uygun ürün kategorisini seçer ve daha sonra kullanım ihtiyaçlarına göre ilgili bilgileri girer. Uygulama daha sonra kullanım ihtiyacıyla eşleşmesi için gereken ürünleri hesaplar ve bunları listede gösterir. Kullanıcı listeyi ürün temsilcisine gönderebilir veya daha sonra kullanmak üzere cihazına kaydedebilir. Uygulama, Şekil 4.3'te gösterildiği gibi web kaynakları ile iletişim kurmaktadır.



Şekil 4.3: Uygulama C'nin iletişim modeli.

C uygulaması için kriter ağırlık değerleri ve nedenleri Tablo 4.3'de belirlenmiştir. Ağırlık değerleri uygulamanın sahipleriyle yapılan görüşmelerce toplanmıştır. Görüşülen kişilere her bir kriter için neden ve ne kadar önemli olduğu sorulmuştur.

Tablo 4.3: Uygulama C için kriter ağırlık değerleri.

<p>HEDEF PLATFORMLA UI TUTARLILIĞI</p> <p>Uygulama kişisel kullanım için tasarlanmıştır, bu nedenle kullanıcının aşına olduğu platform kurallarına uymalıdır. Uygulama kullanıcı ara yüzünün platformlar arasında aynı olması gerekmez</p>	2
<p>ETKİLEŞİM VE DUYARLILIK</p> <p>Tüm hesaplama yerel cihazda değil bulutta yapılır. Bu nedenle performansın yüksek olması öncelikli değil.</p>	1
<p>DESTEKLENEN PLATFORMLAR</p> <p>Mümkün olduğu kadar çok platform ve sürüm desteklenmelidir.</p>	3
<p>ENERJİ TÜKETİMİ</p> <p>Enerji tüketimi makul olmalı, ancak uygulamanın sürekli kullanılması amaçlanmamıştır, bu nedenle yüksek öncelikli değildir.</p>	1
<p>PAYLAŞILAN KOD</p> <p>Farklı platformlar için hedeflenen uygulamanın farklı sürümlerinin aynı mantığa sahip olması son derece önemlidir. Uygulama bir satış aracıdır ve sunduğu bilgilerin güncel ve tutarlı olması gerekir. Ayrıca, uygulama birden fazla platform için tasarlanmıştır ve kodun bu platformlar arasında paylaşılması geliştirme maliyetini büyük ölçüde azaltır.</p>	3

YEREL İŞLEVLERE ERİŞİM Herhangi bir yerel kaynağa erişim gerekmemektedir.	1
YETKİNLİK İyi bilinen bir teknoloji kullanılmalıdır. Uygulama sürekli bir gelişim veya düzenli bir geliştirme ekibine sahip olmayacak, bunun yerine bir seferde daha fazla güncelleme amaçlanacak.	3
TASARIM Uygulama kararlı olmalı ve nadiren güncellemeler alabilmelidir. Tasarım yapmak açık ve kolay olmalı.	3
MALİYET Uygulama bir satış destek aracıdır ve iş için çok kritik değildir. Bu nedenle geliştirme maliyetinin düşük olması gerekir.	3
FİZİBİLİTE Uygulama gerektiğinde güncellenebilir veya değiştirilebilir.	1
YAYINLAMA VE DAĞITIM Uygulamanın son kullanıcılar için kolayca erişilebilir olması gerekir. Mevcut hizmet yalnızca kullanıcının bir web sitesini ziyaret etmesini ve mobil uygulamayı cihazlarına yüklemesini gerektirir. Kullanıcı daima uygulamanın en son sürümüne sahip olmalıdır.	3
GÜVENLİK Uygulama tarafından sağlanan veriler herkese açık bir bilgidir, bu nedenle güvenlik bir sorun değildir.	1

4.2.2. Değerlendirme Matrisi

Önceki bölümde, farklı yöntemlerin değerlendirme sonuçlarını bir matrise topladık ve sonuçların toplamını ve ortalamasını hesapladık. Şimdi, farklı uygulamaların ağırlık çarpanları eklenerek bir geliştirme aracı değerlendirme matrisi oluşturulacaktır. Her kriter kendi ağırlığı ile çarpılır ve çarpılan kriterlerin toplamı, ağırlıklı ortalamaları almak için ağırlık değerlerinin toplamına bölünür. Bu işlem daha sonra her bir hedef uygulama için tekrar edilebilir. Üç örnek uygulama için eklenen ağırlık değerleri ve son değerleri içeren değerlendirme matrisinin tamamı Şekil 4.4'te gösterilmektedir.

Kriter	Geliştirme Araçları			Ağırlık Değerleri		
	Native	Web	Hibrit	Uygulama A	Uygulama B	Uygulama C
Hedef Platform UI Tutarlılığı	5	3	2	2	2	2
Etkileşim ve Duyarlılık	5	1	4	3	3	1
Desteklenen Platformlar	1	5	4	2	2	3
Enerji Tüketimi	5	2	4	1	1	1
Paylaşılan Kod	2	5	4	3	1	3
Yerel İşlemlere Erişim	5	1	4	3	3	1
Yetkinlik	4	5	3	2	1	3
Tasarım	4	5	3	3	1	3
Maliyet	3	5	4	2	2	3
Fizibilite	4	5	3	2	3	1
Yayınlama ve Dağıtım	3	5	3	2	3	3
Güvenlik	5	2	4	2	3	1
Toplam	46	44	42			
Ortalama	3,8	3,7	3,5			
Ağırlıklı Ortalama						
Uygulama A	3,8	3,6	3,5			
Uygulama B	4,0	3,4	3,5			
Uygulama C	3,4	4,3	3,4			

Şekil 4.4: Geliştirme yöntemi değerlendirme matrisi

Uygulama A incelendiğinde, matristen elde edilen native uygulama için 3,8, web için 3,6 ve hibrit uygulama için 3,5 puan almıştır. Bluetooth desteği zorunluluğu, bir web uygulamasının A uygulaması için uygun bir seçenek olmadığı ve sonuçların dışında tutulduğu anlamına gelir. Derecelendirmeler, bir native ve karma uygulama arasında, native geliştirmenin biraz daha iyi bir seçenek olduğunu göstermektedir. Bu durumda, sadece bir kaç platformun desteklenmesi gerekir, böylece çoklu platform çözümünün temel faydası azalır. Ayrıca, iyi kullanıcı deneyimi

ve etkileşimi, uygulamaların ana odak noktalarıdır ve native bir uygulama, bu alanlarda daha yüksek kaliteye sahiptir. A uygulamasını geliştirmek için platformlar arası bir araç kullanmak geliştirme süresi ve maliyet tasarrufu açısından küçük bir yarar sağlayacaktır ve nihai uygulamanın kalitesinden ödün verme riski doğacaktır. Öyle ki, geliştirme aracı değerlendirme matrisinin sonucunu destekler ve uygulama A özelinde, her platform için ayrı geliştirme yapmak gerekmektedir.

B uygulaması incelendiğinde matrizen, native için 4,0, web için 3,4 ve hibrit için 3,5 değerlerini almaktayız. Native uygulama açıkça büyük bir farkla en yüksek puantaja sahiptir. Bir kez daha, bir web uygulaması donanım işlevselliği gereksinimleri nedeniyle kötü bir seçim olacaktır. HTML teknolojisi son zamanlarda gelişim göstermekte ve cihazın kamerasına sınırlı oranda erişebilmektedir. Bu nedenle, uygulamayı şimdi bir web uygulaması olarak geliştirmek mümkün olabilir, ancak web uygulaması, diğer seçeneklere göre yine de düşük bir değere sahip. Uygulama grafik ve işlevsellik açısından oldukça basittir, performans sorunu olmamalıdır. Uygulama arayüzü farklı platformlar için özelleştirmenin çok fazla olmayacağı kadar basit ve diğer platformlar için aynı bırakılabilir. Uygulama, PhoneGap ile hibrit bir uygulama olarak geliştirilmiştir ve hibrit uygulamanın paylaşılan kod temeli ve daha kısa geliştirme süreleri gibi platformlar arası özelliklerinden yararlanılmıştır. Oluşturduğumuz değerlendirme matrisi sonuçları, uygulamanın bunun yerine native bir uygulama olarak geliştirilmesi gerektiğini göstermektedir. B uygulamasının ana odağı yüksek kaliteli kullanıcı deneyimidir. Native geliştirme yüksek kaliteli bir kullanıcı deneyimi sunduğu için maliyet ve geliştirme süresi gibi kriterler önemsiz kalmıştır.

Matrisin mevcut uygulamalar için çalıştığı doğrulandıktan sonra, yeni bir projede en iyi yöntemi seçmek için kullanılabilir. Üçüncü uygulama, Şekil 4.4'de belirtilen değerlendirme matrisindeki ağırlık değerlerine sahiptir ve sonuçlar en alt satırda gösterilmektedir. Derecelendirmeler native geliştirme için 3,4, web geliştirme için 4,3 ve hibrit geliştirme için 3,4 şeklinde olmuştur.

Kullanıcının olası bir cihazının desteklendiğinden emin olmak için uygulamanın mümkün olduğunca çok platform ve sürümü desteklemesi gerekmektedir. Bunu native uygulamalarla tamamlamak kolay olmazdı, geliştirme maliyeti çok yüksek olurdu ve her bir sürümün aynı ve güncel bir mantığa sahip olmasını sağlamak neredeyse imkânsız olurdu. Uygulama gelişmiş native işlevsellik gerektirmemekte ve kullanıcı deneyimine öncelik vermemektedir, bu nedenle

native uygulamanın ana faydaları öncelikli değildir. Bir web uygulaması, uygulama için gereken özellikleri kapsayacak şekildedir. Platform bazında UI kodunda sadece küçük değişiklikler bulunan, tek bir kod tabanına sahip geniş bir platform yelpazesini destekleyebilir. Ayrıca, kullanıcının kullandığı uygulamanın en yeni sürümüne sahip olmasını ve platformdan bağımsız olarak uygulamanın aynı mantığa sahip olmasını sağlayacaktır. Kullanıcı bakış açısından bakarsak, uygulama kolayca erişilebilir ve kurulum gerektirmez. Bu sonuçlara dayanarak C uygulaması bir web uygulaması olarak geliştirilmelidir sonucunu rahatça çıkarabiliriz.



5. TARTIŞMA VE SONUÇ

Yapılan çalışmada, mobil uygulama geliştirme yaklaşımları ve farklı platformlar için mobil uygulama geliştirme yöntemleri incelenmiştir. Farklı uygulama geliştirme ihtiyaçları için geliştirme yapılacak yaklaşım seçimi analiz edilmiştir. Üç ana geliştirme yöntemi belirlenmiştir.

Native uygulamalar, platformun standart geliştirme araçları ve düzeni ile aynı anda tek bir platforma geliştirme yapmak içindir. Native uygulamalarla birden fazla platformun desteklenmesi, uygulamanın her hedef platform için ayrı ayrı geliştirilmesini gerektirir.

Web uygulamaları web teknolojileri ile geliştirilir ve web sitesi olarak barındırılırlar. Bir web uygulamasının sadece bir kez geliştirilmesi gerekir ve herhangi bir mobil cihaz bir URL üzerinden standart web tarayıcılarıyla onlara erişebilir.

Çoklu platform uygulamaları genel olarak web teknolojilerini kullanırlar, ancak sonuçta ortaya çıkan uygulama, native bir uygulama gibi cihaza yüklenir. Çoklu platform araçları ile geliştirilirler ve uygulamanın ana kısmı sadece bir kez geliştirilir ve bu araç tarafından desteklenen herhangi bir platforma dağıtılabilir.

Yapılan çalışmada yaklaşım değerlendirme işlemi, hedef platform UI tutarlılığı, etkileşim ve duyarlılık, desteklenen platformlar, enerji tüketimi, paylaşılan kod, yerel işlemlere erişim, yetkinlik, tasarım, maliyet, fizibilite, yayınlama ve dağıtım, güvenlik gibi 12 adet kritere göre analiz edildi. Öncelikle her bir kriter 1 (çok zayıf) ile 5 (çok iyi) arasında her bir geliştirme yöntemine göre derecelendirerek değerlendirme matrisi oluşturuldu. Derecelendirmeler, herhangi bir yöntemi genel olarak en iyi ilan etmek için birbirine çok yakındı.

Seçim süreci için üç farklı mobil uygulama belirlenmiştir. Belirlenen uygulamaların ikisi mevcut ve geliştirilmesi tamamlanmış olup, değerlendirme matrisi tarafından önerilen geliştirme yöntemini, gerçekte uygulamanın geliştirilmesinde kullanılan yöntemle karşılaştırarak değerlendirme matrisini doğrulamak için örnek teşkil etmektedir. Üçüncü uygulama henüz geliştirme aşamasında değildir ve değerlendirme matrisi bunun için en iyi geliştirme yöntemini bulmak yönünde kullanılacaktır.

Her uygulama için ağırlık deęerleri belirlenmiştir. Bu ağırlık deęerleri 1 ile 3 arasında önem derecesine göre deęer almaktadır. Yapılan deęerlendirmeler deęerlendirme matrisinde toplandı ve bu matrise uygulamaya özel ağırlık deęerleri eklenerek geliştirme aracı deęerlendirme matrisine genişletildi. Geliştirme araçlarına ait deęerler uygulamalara ait ağırlık deęerleriyle çarpılarak her uygulama için ağırlıklı ortalama deęerleri elde edildi. Böylece bir geliştirme yönteminin belirli bir uygulamanın gereksinimlerini ne kadar iyi karşılayıp karşılamadığını görmüş olduk.

Deęerlendirme matrisi yoluyla üretilen sonuçlar, uygulama A için en iyi çözümün native uygulama, uygulama B için en iyi çözümün native uygulama, uygulama C için en iyi çözümün web uygulama olması gerektiğini göstermiştir.

Geliştirme aracı deęerlendirme matrisi daha da geliştirilebilir. Bu tezde belirtilen süreci takip ederek matrise ek geliştirme yöntemleri eklenebilir. Ayrıca, geliştirme yöntemleri yeni sürümlerle güncellendikçe deęerlendirmeler de gözden geçirilerek güncellenmelidir.

KAYNAKLAR

- [1]. Elgin, B., 2005, Google Buys Android for Its Mobile Arsenal, <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>, [Ziyaret tarihi: 12 Ocak 2019].
- [2]. Klein, K., 2007, Industry Leaders Announce Open Platform for Mobile Devices, http://www.openhandsetalliance.com/press_110507.html, [Ziyaret tarihi: 12 Ocak 2019].
- [3]. Butler, M., 2011, Android: Changing the Mobile Landscape, IEEE Pervasive Computing, 10 (1), 4-7.
- [4]. Meier, R., 2012, Professional Android 4 Application Development, CreateSpace Independent Publishing Platform, İngiltere, ISBN: 978-1118102275.
- [5]. Tracy, K., 2012, Mobile Application Development Experiences on Apple's iOS and Android OS, IEEE Potentials, 31(4), 30-34.
- [6]. Apple, 2012, iOS Technology Overview, [https://developer.apple.com/library/prerelease/iOS/documentation/Miscellaneous/Conceptual/iPhoneOSTech verview/Introduction/Introduction.html](https://developer.apple.com/library/prerelease/iOS/documentation/Miscellaneous/Conceptual/iPhoneOSTech%20vewiew/Introduction/Introduction.html), [Ziyaret tarihi: 8 Mart 2019].
- [7]. Apple, 2016, iOS Dev Center, <https://developer.apple.com/devcenter/ios/index.action>, [Ziyaret tarihi: 8 Mart 2019].
- [8]. Ohrt, J., Turau, V., 2012, Cross-Platform Development Tools for Smartphone Applications, Computer, 45(9), 72-79.
- [9]. Android, 2019, Android Development, <http://developer.android.com/index.html>, [Ziyaret tarihi: 14 Şubat 2019].
- [10]. Heitk, H., Hanschke, S., Majchrzak, T.A., 2012, Comparing Cross-Platform Development Approaches for Mobile Applications, Computer, 32(3), 52-56.
- [11]. Charland, A., Leroux, Brian., 2011, Mobile Application Development: Web vs. Native, Communications of the ACM, 54(5), 0-4.

- [12]. Laine, M., Shestakov, D., Litvinova, E., Vuorimaa, P., 2011, Toward Unified Web Application Development, IT Professional, 13(10), 30-36
- [13]. HTML5TEST, 2019, How well does your browser support HTML5 ? <https://html5test.com/results/mobile.html>, [Ziyaret tarihi: 27 Mart 2019].
- [14]. Wasserman, I, 2010, Software Engineering Issues for Mobile Application Development, FoSER '10 Proceedings of the FSE/SDP workshop on Future of software engineering research, 7-8 November 2010 New Mexico, USA, ACM New York, ISBN: 978-1-4503-0427-6, 397-400.
- [15]. Corral, L., Janes, A., Remencius, T., 2012, Potential Advantages and Disadvantages of Multiplatform Development Frameworks-A Vision on Mobile Environments, Procedia Computer Science, 10(1), 1202-1207.
- [16]. Hartmann, G., Stead, G., DeGani, A., 2011, Cross-platform mobile development, Tribal, 1(4), 1-18.
- [17]. Research 2 Guidance, 2013, Cross-Platform Tool Benchmarking 2013, <https://research2guidance.com/product/cross-platform-tool-benchmarking-2013>, [Ziyaret tarihi: 2 Mayıs 2019].
- [18]. PhoneGap, 2019, PhoneGap, <http://phonegap.com>, [Ziyaret tarihi: 2 Nisan 2019].
- [19]. Dalmaso, I., Datta, S.K., Bonnet, C., Nikaein N., 2013, Survey, comparison and evaluation of cross platform mobile application development tools, Wireless Communications and Mobile Computing Conference (IWCMC), 1-5 July 2013 Cagliari, Cagliari Italy, Mobile Communication Department, EURECOM, 323-328.
- [20]. PhoneGap, 2019, PhoneGap Plugins, <https://build.phonegap.com/plugins>, [Ziyaret tarihi: 2 Nisan 2019].
- [21]. PhoneGap, 2019, PhoneGap Documentation, <http://docs.phonegap.com/en/edge/index.html>, [Ziyaret tarihi: 2 Nisan 2019].
- [22]. Appcelerator, 2019, Titanium, <http://www.appcelerator.com/titanium>, [Ziyaret tarihi: 5 Nisan 2019].

- [23]. Qt, 2019, Qt, <http://www.qt.io>, [Ziyaret tarihi: 6 Nisan 2019].
- [24]. Xamarin, 2019, Xamarin, <http://xamarin.com>, [Ziyaret tarihi: 6 Nisan 2019].
- [25]. ArsTechnica, 2019, Xamarin 3 review: Making crossplatform mobile development painless, <http://arstechnica.com/information-technology/2014/05/xamarin-3-review-making-cross-platform-mobile-development-painless>, [Ziyaret tarihi: 5 Nisan 2019].
- [26]. RhoMobile, 2019, Rhodes, <https://github.com/rhomobile/rhodes>, [Ziyaret tarihi: 5 Nisan 2019].
- [27]. RhoMobile, 2019, RhoMobile, <http://rhomobile.com>, [Ziyaret tarihi: 5 Nisan 2019].
- [28]. Palmieri, M., Singh, I., Cicchetti A., 2012, Comparison of cross-platform mobile development tools, 16th International Conference on Intelligence in Next Generation Networks, 8-11 October Berlin, Berlin, Germany, IEEE, ISBN: 978-1-4673-1525-8, 179-186.
- [29]. RhoMobile, 2019, RhoMobile Suite Documentation, <http://docs.rhomobile.com/en/5.0.0/home>, [Ziyaret tarihi: 5 Nisan 2019].

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Çağrı ÇINAR
Doğum Yeri	Amasya
Doğum Tarihi	02.05.1985
Uyruğu	<input checked="" type="checkbox"/> T.C. <input type="checkbox"/> Diğer:
Telefon	+90 505 488 13 93
E-Posta Adresi	cagricinar@gmail.com
Web Adresi	



Eğitim Bilgileri	
Lisans	
Üniversite	Yıldız Teknik Üniversitesi
Fakülte	Kimya Metalurji Fakültesi
Bölümü	Matematik Mühendisliği
Mezuniyet Yılı	2010

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi-Cerrahpaşa
Enstitü Adı	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı

Makale ve Bildiriler	