

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
(YÜKSEK LİSANS TEZİ)

SERVİS YÖNELİMLİ MİMARİ VE MOBİL HESAPLAMA

Abd-Al-Raheem AL-RADAİİ

Tez Danışmanı: Yrd. Doç. Dr. Özgür GÜMÜŞ

Bilgisayar Mühendisliği Anabilim Dalı

Sunuş Tarihi : 04.12.2015

Bornova-İZMİR

2015

Abd-al-raheem AL-RADAİl tarafından hazırlanan Yüksek Lisans tezi olarak sunulan “Servis Yönelimli Mimari ve Mobil Hesaplama” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 04.12.2015 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

Jüri Başkanı : Yrd. Doç. Dr. Özgür GÜMÜŞ

Raportör Üye: Doç. Dr. Aybars UĞUR

Üye : Yrd. Doç. Dr. Övünç ÖZTÜRK

İmza

.....
.....
.....

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ETİK KURALLARA UYGUNLUK BEYANI

E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Servis Yönelimli Mimari ve Mobil Hesaplama” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

04 / 12 / 2015

İmzası

Adı-Soyadı

Abd-al-raheem AL-RADAİİ

ÖZET**SERVİS YÖNELİMLİ MİMARİ VE MOBİL HESAPLAMA**

Abd-Al-Raheem AL-RADAİİ

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Özgür GÜMÜŞ

Aralık 2015, 125 sayfa

Servis Odaklı Mimari (SOA) farklı uygulamaların etkileşimine; platformları, uygulama dilleri ve konumlarına bakılmaksızın; uygulama yapı taşı olarak kullanılabilir jenerik ve güvenilir servislerini kullanarak olanak sağlayan mimari bir tarzdır. SOA, sofistike uygulamalar ve bilgi sistemleri geliştirmek için izlenecek metodolojileri ve stratejileri içerir.

Son yıllarda mobil cihazlar çok yaygınlaşmıştır. Mobil bilişim sistemleri, kullanım amaçları, tasarlanış ve çalışma biçimleri açısından diğer bilişim sistemlerinden farklılık gösterirler. Mobil bilişim sistemlerinin ayırt edici yönleri arasında yaygın kablosuz ağ bağlantıları, küçük boyutları, güç kaynakları ve özellikle mobil kullanıcılara uygun düşen işlevleri gösterilebilir.

Tez iki amacı izlemeyi hedeflemektedir. Bunlardan birincisi, SOA ve mobil bilişim alanında genel bir bakış sağlamak ve uygulama geliştirmenin farklı yollarını açıklamaktır. Tezin ikinci amacı ise, hemofili hastalarının hastane dışında takibini sağlamak için kullanılacak bir mobil hasta takip sistemi geliştirmektir.

Anahtar sözcükler: Servis Odaklı Mimari, Mobil Hesaplama, Hemofili.

ABSTRACT
SERVICE ORIENTED ARCHITECTURE AND MOBILE
COMPUTING

Abd-Al-Raheem AL-RADAİİ

MSc in Computer Engineering

Supervisor: Asst. Prof. Dr. Özgür GÜMÜŞ

December 2015, 125 pages

Service Oriented Architecture (SOA) is an architectural style which allows interaction of diverse applications regardless of their platform, implementation languages and locations by utilizing generic and reliable services that can be used as application building block. SOA includes methodologies and strategies to follow in order to develop sophisticated applications and information systems.

In recent years, mobile devices have become widespread. Mobile computing systems are different from other computing systems in the way of usage purposes, designs and operating styles. Distinguishing properties of mobile computing systems are wireless connection, small size, power supply and features for mobile users.

This thesis aims at pursuing two goals. First one is to provide a general overview of the field of SOA and mobile computing and to explain different ways of developing applications. Second goal is to develop a mobile patient tracking system to follow hemophilia patients outside of the hospital.

Keywords: Service Oriented Architecture, Mobile Computing, Hemophilia.

TEŞEKKÜR

Öncelikle danışmanım Yrd. Doç. Dr. Özgür GÜMÜŞ' e önerileri, alakası ve yardımları için sonsuz teşekkür etmek istiyorum. Daimi yardım ve destekleri için sevgili arkadaşlarım Ali Naji ve Ghasan çok teşekkürler.

Bu noktada bu işin başarısına önemli ölçüde katkıda bulunan tüm kişilere teşekkür etmek istiyorum.

Ayrıca bu tezi babam Ahmad Nassir, annem Bahja Mohammed, eşim Hassna, çocuklarım Roaa, Rehab, Nora, Ali ve kardeşlerime ithaf ediyorum.

İÇİNDEKİLER

Sayfa

ÖZET	VII
ABSTRACT	IX
TEŞEKKÜR	XI
ŞEKİLLER DİZİNİ	XV
ÇİZELGELER DİZİNİ.....	XVIII
SİMGELER VE KISALTMALAR DİZİNİ	XIX
1. GİRİŞ	1
1.1 Altyapı.....	1
1.2 Gelişme Seviyesi.....	2
1.3 Motivasyonlar	3
1.4 Tezin Amacı ve Yapısı.....	4
2. SERVİS YÖNELİMLİ MİMARİ	6
2. 1 Giriş	6
2.2 Bilgi İşlem Sistemlerinin Evrimi	6
2.3 Nesne Yönelimli Geliştirme	10
2.4 Bileşen Yönelimli Geliştirme	11
2.5 Dağıtılmış Bilgi İşlem	12
2.5.1 Uzaktan yordam çağırısı	14
2.5.2 Dağıtılmış nesneler	15
2.5.3 Mesaj temelli orta katman	16
2.6 Servis Yönelimli Mimarinin Tanımı	17
2.6.1 SOA varlıkları ve özellikleri.....	18
2.6.2 Servis odaklı geliştirme	20
2.6.3 SOA katmanlı mimari.....	22
2.7. Servis Yönelimli Mimari Teknolojileri	25
2.7.1 J2EE.....	25
2.7.2 COM / DCOM	28
2.7.3 CORBA	30
2.7.4 Web servisleri	30
2.8 Bulut Bilişim ve Servis Odaklı Hesaplama	33
2.8.1 Servis olarak yazılım	33

İÇİNDEKİLER (devam)

Sayfa

2.8.2 Servis olarak platform.....	33
2.8.3 Servis olarak altyapı.....	34
3. MOBİL HESAPLAMA.....	35
3.1 Giriş.....	35
3.2 Mobil Donanımlar.....	42
3.3 Mobil Yazılım Platformları ve Uygulamaları.....	51
3.3.1 Symbian	51
3.3.2 Palm	53
3.3.3 Windows Mobile.....	54
3.3.4 Gömülü Linux	54
3.3.5 RIM Blackberry	55
3.3.6 iPhone OS	56
3.3.7 Android	58
3.3.8 Brew	60
3.4 Java Micro Edition (J2ME).....	61
3.5 En Yaygın Cihazlar ve Platformlar Arasındaki Karşılaştırma.....	63
4. MOBİL HEMOFİLİ HASTA TAKİP SİSTEMİ	69
4.1 Hemofili Hastalığı.....	69
4.1.1 Tanım	69
4.1.2 İnsidans	69
4.1.3 Genetik.....	70
4.1.4 Klinik tanı	70
4.1.5 Kanama yerlerine göre klinik özellikler.....	71
4.2 Mobil Hemofili Takip Sistemi Uygulaması.....	72
4.2.1 Sistem mimarisi.....	73
4.2.2 Mobil uygulamanın tanıtımı.....	76
4.3 Mobil Hemofili Takip Sistemi Kullanıcı Kılavuzu.....	78
5. SONUÇ	102
KAYNAKLAR DİZİNİ	103

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Yazılım Mimarisinin Özet Tanımı	6
2.2 Yordamsal Yazılım Geliştirme	7
2.3 Yapısal Yazılım geliştirme	8
2.4 İstemci - Sunucu ve N-Katlı Yazılım Gelişmeler.....	9
2.5 Bilgi İşlem Sistemlerinin Fiziksel Evrimi	10
2.6 Nesne Yönelimli Geliştirme	11
2.7 Bileşen Yönelimli Geliştirme	12
2.8 Yanıtlı Senkron Mesajlaşma.....	13
2.9 Yanıtlı Asenkron Mesajlaşma.....	14
2.10 Uzaktan Yordam Çağrısı Kavramsal Modeli	15
2.11 Nesne Talep Aracısı.....	16
2.12 Mesaj temelli Orta Katman.....	16
2.13 Servis Yönelimli Mimarinin Kavramsal Modeli	18
2.14 Öge boyu.....	19
2.15 Servis Tabanlı Geliştirme	21
2.16 İki katmanlı ve üç katmanlı Mimari Modeller.....	23
2.17 Servis Odaklı Mimarinin Katmanları	24
2.18 Java Mesaj Hizmetinin Kavramsal Modeli.....	27
2.19 RMI Dağıtılmış Uygulama Modeli.....	28
2.20 COM Mimari Modeli.....	29
2.21 DCOM Mimari Modeli.....	29
2.22 CORBA Nesne İsteği Ara Yüzlerinin Yapısı	30
2.23 Web Servisleri Mimarisi.....	31
2.24 Web Servisleri Teknolojisi Yığını	32
3.1 Dijital yakınsama	36
3.2 Mobil Bilgi İşlemin Vizyonu.....	40
3.3 Cortex için ARM işlemci evrimi	48
3.4 Symbian OS mimarisi.....	52
3.5 Android platform mimarisi	59
3.6 konfigürasyonları ve profilleri.....	63
4.1 Sistem mimarisi	73

ŞEKİLLER DİZİNİ (devam)ŞekilSayfa

4.2 Cep telefon masaüstünü arayüzü.....	78
4.3 (Splash) arayüzü.....	78
4.4 Ana kullanıcı arayüzü.	79
4.5 kullanıcının bilgileri girmeyin arayüzü.....	79
4.6 Serverdan kullanıcının bilgileri getirme arayüzü.....	80
4.7 Serverden bilgiler getirme arayüz.....	80
4.8 Şifre değiştirme Arayüzü	81
4.9 Şifre değiştirme Arayüzü.	81
4.10 Şifre sıfırlama Arayüzü.....	82
4.11 Ana arayüzü	82
4.12 Yan arayüzü.	83
4.13 Ana log infusion arayüzü.....	83
4.14 Ana log bleed arayüzü.....	84
4.15 Log Infusion arayüzü	84
4.16 Log Infusion arayüzü.	85
4.17 Log Bleed arayüzü.	86
4.18 Log Bleed arayüzü.	86
4.19 kontrol butonlar listesi.	87
4.20 Ana arayüzünü butonları.....	87
4.21 Ana Uyarılar arayüzü.....	88
4.22 Ana raporlar arayüzü.....	88
4.23 Hesap Yöneticisi arayüzü.....	89
4.24 Uyar Bildirimi arayüzü.	89
4.25 Uyar Bildirimi arayüzü.	90
4.26 Acil Aramalar arayüzü.....	90
4.27 Görüş bildirim arayüzü.	91
4.28 Barkod arayüzü.	91
4.29 Log Bleed bir kayıt silmesi arayüzü.	92
4.30 Ayarlar arayüzü.....	92
4.31 Günlük olarak Rapor göndermesi arayüzü.....	93
4.32 Raporlar gönderme arayüzü	94
4.33: Ana alarm arayüzü ve Alarm ayarları.....	94

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
4.34 Alarm listesi arayüzü.	95
4.35 Alarm arayüzü.	95
4.36 Alarm ayar arayüzü.	96
4.37 Çıkış arayüzü.	96
4.38 Hemofili website arayüzü.	97
4.39 Hemofili websitede kullanıcı arayüzü.	98
4.40 Hemofili websitede doktora ait hastalarının bilgi arayüzü.	98
4.41 Hemofili websitede doktora ait hastalarının bilgileri arayüzü.	99
4.42 Hemofili websitede sohbet bilgileri arayüzü.	100
4.43 Hemofili websitede sohbet bilgileri arayüzü.	100
(4.44 – 4.46) HemoÖzgür programını Sohbet arayüzü.	101

ÇİZELGELER DİZİNİÇizelgeSayfa

2.1 Mimari Geliştirme Modellerinin Karşılaştırılması.	22
3.1 ARM işlemci aileleri ve uygulamaları bölüm 1.....	44
3.2 ARM işlemci aileleri ve uygulamaları bölüm 2.....	45
3.3 ARM işlemci aileleri ve uygulamaları bölüm 3.....	46
3.4 ARM işlemci aileleri ve uygulamaları bölüm 4.....	46
3.5 Akıllı telefonlar arasında karşılaştırma - bölüm 1.	66
3.6 Akıllı telefonlar arasında karşılaştırma - bölüm 2.	67

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
(SOA)	Service oriented architecture
(PDA)	Personal digital assistants
(SMS)	Short messaging service
(GSM)	Global system for mobile
(GSMA)	Global system for mobile association
(CDMA)	Code division multiple access
(3G)	Third generation
(4G)	Fourth generation
(MNO)	Mobile network operators
(HTTP)	Hypertext transfer protocol
(SNA)	Social-national assembly
(TCP)	Transmission control protocol
(IP)	İnternet protocol
(IPX)	İnternetwork packet exchange
(API)	Active pharmaceutical ingredient
(RPC)	Remote procedure call
(RFC)	Remote function call
(ORB)	Object request broker
(MOM)	Message oriented middleware
(WSRP)	Web services for remote portlet
(QoS)	Quality of service
(J2EE)	Java 2 platform enterprise edition
(DCOM)	Distributed component object model
(XML)	Extensible markup language
(WSDL)	Web services description language
(SOAP)	Simple object access protocol
(HCIs)	Challenge on human-computer interfaces
(GUI)	Graphical user interface
(PVPs)	Portable video players
(PMCs)	Portable media centers
(TFT)	Thin-film transistor
(MIDP)	Mobile information device profile

SIMGELER (devam)

(SDK)	Software development kit
(SIM)	Subscriber identity module
(GPL)	General public license
(BREW)	Binary runtime environment for wireless

1. GİRİŞ

1.1 Altyapı

İnsan toplumunun en başında insanlar eşit değerler ile mal alışverişinde bulundu. Daha sonra, para kavramı getirilmiştir, böylece insanlar işlemler için para kullanmaya başladılar. Sanayi devrimi ve küreselleşmeden dolayı ticari işlemler hızla artmıştır. Bununla birlikte, büyük miktarda para alışverişi riskli bir iştir. Çözüm olarak, insanlar finansal işlemler için bankacılık imkânları kullanmaya başladılar. Günümüzde bankalar, etkin para işlemleri için cazip hizmetler sunmaktadır. Ancak, banka işlemleri ile ilgili birçok sorun hala devam etmektedir.

İnsanların sağlık açısından önceliklerinden birisi, fiziksel sağlığa ilişkin mümkün olduğu kadar çok iş sahası mevzusudur.

Gelişmekte olan ülkelerde bu sorunlar daha da kötü hale gelmiştir. Birçok kırsal alanda, tedavisi uzun zaman alan ve özel dikkat isteyen ve sürekli takip edilmesi gereken ciddi vakalar gibi hiçbir gelişmiş sağlık hizmeti yoktur. Bu büyük bir endüstridir ancak sadece büyük işletmeler bu tür işlemlerin üstesinden gelme yeteneğine sahiptirler. Daha da kötüsü bu şirketlerin çoğunun kırsal bölgelerde şubelerinin olmamasıdır. Buna ek olarak, gelişmekte olan ülkelerde IT tabanlı çözümlerin hayata geçirilmesi ve kullanılması, sağlık tesislerinin genişletilmesi için darboğaz yaratan zayıf iletişim ve bilişim altyapısı nedeniyle aynı zamanda büyük bir sorundur.

Öte yandan, kablosuz ve mobil teknolojilerde dramatik bir büyüme yaşanmaktadır. Mobil telefonların fonksiyonel yetenekleri hızla genişlemektedir ve klasik haberleşme uygulamalarının - telefon görüşmeleri ve kısa mesaj - çok ötesinde kullanımlarını arttırmıştır. IT altyapısı genellikle kırsal alanlarda gelişmemiş olmasına rağmen, bu teknolojilerin tüm dünyada olumlu, mali, ekonomik ve sosyal etkilerine ilişkin artan kanıtlar mevcuttur.

Son yirmi yıl boyunca mobil teknolojisi büyük ilerleme kaydetmiştir. Bir taraftan, mobil şebekeler, hücresel ağlardan 3G, artık 4G ağlara gelişim göstermiştir. Öte yandan, mobil cihazlar yirmi yıl öncesinden çok daha güçlü hale gelmiştir.

Bugün, cep telefonları tüm dünyada insanların günlük yaşamlarında zaten en önemli aksesuarlardan biri haline gelmiştir. GSMA'ya göre, bir endüstri grubunun dünyada mobil abone sayısı 2013 yılı itibarıyla 6 milyara ulaşacaktır

(Mobile marvels 2009). Mobil ağların gelişmesiyle birlikte, mobil endüstrisi ses merkezli olmaktan veri merkezli olma değişimi göstermektedir. Tüketicilerin %90 'i konuşmaya zaman harcarken, zamanlarının %80' inde mobil veri hizmetleriyle meşgul oluyorlar (Chetan Sharma Danışmanlık 2011). Bu yüzden, standart iletişim hizmetlerine ek olarak - telefon görüşmeleri ve Kısa Mesaj Servisi (SMS) mesajlarının karşılıklı alınıp verilmesi, mobil bankacılık ve mobil ticaret gibi mobil uygulamalar daha da popüler hale gelmiştir. Mobil teknoloji, özellikle mevcut web tabanlı uygulamaları ve işlemleri tamamlamak ve mevcut, yine de genellikle kâğıt bazlı finansal sistemleri nakitsiz sistemlere dönüştürmek için yenilikçi bir yaklaşım olarak kabul edilmektedir. Finansal kuruluşlar ve mobil ağ operatörleri (MNO), telefon kontörü gibi ödeme işlemleri ve para transferleri veya fatura ödemelerini mobil elektronik forma dönüştürmek için birçok yatırım çalışması yapmaktadır. Müşteriler, her zaman ve her yerde çeşitli servis sağlayıcılar tarafından sağlanan kaynaklar, veritabanı ve hizmetlerine erişmek için bu çeşit mobil uygulamaları ayrıca talep etmektedirler.

Son birkaç yıl içinde, kablosuz iletişim ve mobil bilgi işlem giderek daha popüler ve yaygın hale gelmiştir.

Hücreli teknolojiler veya ad-hoc teknolojileri veya karma yaklaşımlardan oluşan mobil ortamlarda, "Kablosuz bağlantılar" kullanarak iletişim kurulmasını sağlayan mobil heterojen sunucular topluluğu mevcuttur. Bu kablosuz bağlantılar, mobil ağların niteliklerine göre değişebilir, dahası ad-hoc ağ düğümleri, herhangi bir merkez veya yardım almadan iletişim kurmak durumundadır (Hashem et al., 2003). Her mobil düğümü sadece sınırlı işlevsellik sunmaktadır. Ancak, bir bütün olarak cihazlar daha karmaşık görevlerin üstesinden gelebilirler. Bu nedenle, farklı - mobil veya sabit - cihazlar arasında işlevsellik paylaşımına izin veren mekanizmalara ihtiyaç vardır. Servis Odaklı Mimari ve Servis Odaklı Bilgi İşlem, dağıtılmış sistemlere ilişkin uygulamaların tasarım ve gelişimini kolaylaştırmayı amaçlayan en son yaklaşımlardır.

1.2 Gelişme Seviyesi

Gerçi kompakt güçlü cihazların ortaya çıkması, mevcut küresel uygulamalara her zaman ve her yerde erişmelerini olanak sağlamaktadır. Mobil bilgi işlemin 'her zaman her yerde' amacı, bilgi erişimi için ihtiyaç duyulan

tepkisel bir yaklaşım olarak kabul edilebilir, yaygın bilgi işlem, "her yerde her zaman" vizyonu ile proaktif bir yaklaşım olarak algılanmaktadır. 1991 yılında, Mark Weiser (Weiser, 1991) yaygın bilgi işlem vizyonunu tarif etmiştir; o vizyonun özü henüz insan kullanıcılarıyla olağan entegre böylece 'kaybolan teknoloji' halini alan bilgi işlem ve iletişim yeteneği ile doymuş ortamların yaratılmasıydı. Yaygın bilgi işlem, kullanıcının fiziksel ortamında derinden yerleşik yazılım, cihaz ve ağlar kurmayı taahhüt eder. Aslında kullanıcılar, bir bilgi işlem ortamıyla uğraştıklarının dahi farkına varamayabilirler çünkü imkanlar fiziksel ortamla sorunsuz ve doğal olarak entegre edilmiştir. Hareket günlük yaşamın ayrılmaz bir parçası olduğundan, bu tür teknolojilerin hareketliliği desteklemesi gerekir; aksi takdirde, kullanıcı hareket ettiğinde eksikliğiyle birlikte teknolojinin farkına varacaktır. Her yerde bulunan heterojen ve her zaman açık iletişimler ve hem cihazların hem de ortamın akıllı algılaması, evrensel mobil terminaller gibi PDA ve akıllı telefonlar, fiziksel ortamın ayrılmaz bir parçası haline gelmektedir. (Satyanarayanan, 2001) de belirtildiği gibi, yaygın bilgi işlem, 1970 ortalarına kadar uzanan bir çalışma doğrultusunda önemli evrimsel bir adımı sunmaktadır. Bu evrimin iki ayrı ilgili adımı, dağıtık sistemler ve mobil bilgi işlemdir. Dağıtılmış sistemlerin alanı, kişisel bilgisayarlar ve yerel alan ağların kesişme noktasında ortaya çıktı. Dağıtık sistemler, bilgisayarların faaliyetlerini koordine etmelerini ve kullanıcıların işbu bilgisayarları tek bir entegre ve tutarlı sistem olarak algılamaları için sistem kaynaklarını paylaşmalarını sağlayan ağ ve dağıtım ara katmanı üzerinden bağlı otonom bilgisayarlar topluluğundan oluşur. Dağıtık sistemler ağırlıklı olarak açık olma özellikleriyle karakterize edilir. Açıklık, değişen fonksiyonel ihtiyaçlara cevap verebilmek için sistemi kolayca değiştirme ve genişletme olasılığıdır. Her türlü dağıtım sistemi kullanım süresi boyunca gelişecektir.

1.3 Motivasyonlar

Bu dönemde yaşadığımız teknolojik gelişmelerle, iş bu gelişmelerden sağlık açısından yararlanmak mümkündür. Uygun tedavi ve uygun danışmanlık sunulana ve hastanın vücudunda değişiklik izlenene kadar hasta ve doktor tarafından sürekli takip gerektiren bazı hastalıklar vardır.

Mobil insan hastalık gözetim sisteminin geliştirilmesi, Patolojide ileri gözetime ilişkin büyük bir gelişme olarak görülmektedir. Gözetim sistemleri, ülkenin insan sağlığı programının bir parçasını oluşturur ve risk değerlendirme süreçlerine hastalık ölçme ve algılama yeteneği sağlayarak katkıda bulunur (yaygınlık veya olasılık tahmini olarak verilmiştir). Riskleri ölçen, azaltan ve ileten sistemler hastalık gözetim sistemleri için gereklidir. Mobil gözetim sistemine ilişkin motivasyon, doğrudan insanı gözlemlerken hastalığı tanımlama kapasitesidir. Bu, gözetim sistemine sağlanan verilerin kalitesi ve miktarını muhtemelen arttıracaktır.

Mobil veri toplama platformlarının bu bağlamda dikkate alınması gerekir. Elbette, cep telefonları ve PDA' lar birçok sektörde giderek artan oranda kullanılmaktadır, bu artış sadece kişiden kişiye iletişim değil aynı zamanda veri toplama ve diğer uygulamalara yöneliktir. Doktorların cep telefonu teknolojisiyle rahat hissedebileceklerini varsayalım.

Avuç içi uygulaması, insani hastalıklarda veri toplamak üzere bariz bir yaklaşımdır, ancak daha sonra tartışacağımız gibi bu kullanım-durum senaryosu, verilerin toplanması gerektiği çevresel koşullardan kaynaklı birçok ilave zorluk getirmektedir. Sahada avuç içi bilgi işlem cihazlarının kullanımıyla karşılaştırıldığında çevrimiçi web-tabanlı veri gönderme sistemi (merkezi veritabanına gerçek zamanlı erişim sağlayan ve anlık analizi mümkün kılan) kullanılarak elde edilebilecek elektronik veri yakalama olanaklarında büyük bir tezat vardır.

Web-tabanlı sistemler, genel durumlara ilişkin olarak nadiren uygundur. Avuç içi cihazlar hareketlilik sağlarken, çok büyük miktarda veriyle başa çıkma yeteneğine sahip değildirler (Vourc'h et al., 2006; Sprince, 2000). Bu yüzden, avuç içi cihazları yeterince büyük miktarda verilerle çalışmak ve dolayısıyla yararlı olmak için yöntemler ve aynı zamanda kullanıcılara verilen bir durum için uyarlanabilir rehberlik sunmak üzere bir sistem gereklidir.

1.4 Tezin Amacı ve Yapısı

Bu tez iki amacı izlemeyi hedeflemektedir. Bunlardan birincisi, SOA ve mobil bilişim alanında genel bir bakış sağlamak ve uygulama geliştirmenin farklı yollarını açıklamaktır. Tezin ikinci amacı ise, hemofili hastalarının hastane

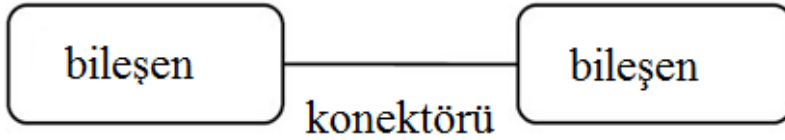
dışında takibini sağlamak için kullanılacak bir mobil hasta takip sistemi geliştirmektir. Çünkü hemofili hastalığı, her an bir kanama meydana gelme olasılığı olan ve hastaların kendi kendine tedavi uygulaması gereken kronik bir hastalıktır.

Tezin 2. Bölümü'nde servis odaklı mimari bileşenleri ve hizmetleri dâhil olmak üzere açıklanacaktır. Bölüm 3'te, mobil bilişim kavramı detaylı olarak açıklanacaktır. 4. Bölümde ise, tezin ana katkısı olan mobil hasta takip sistemi açıklanacaktır. Bölüm 5'te sonuçlar verilecektir.

2. SERVİS YÖNELİMLİ MİMARİ

2.1 Giriş

Yazılım geliştirmenin, gerçek dünya problemlerini çözme yeteneğine sahip karmaşık altyapılara sahip olma hususunda ihtiyaç ve istekler geliştikçe, daha da zorlaştığı ortaya çıkmaktadır. Benzer şekilde, bileşik mimarileri inşa etmeye yönelik yazılım sistemleri geliştirme hususunda birçok eğilim ve alternatif gerekçeler yoluyla teknolojik gelişmeler söz konusudur. Yazılım mimarisi, bileşenlerini ve her biri arasındaki yüksek düzeyde etkileşimleri açıklayarak yazılım sistemi altyapısını araştırır. Bu bileşenler, diğer bileşenleri ile birlikte bir "birim" olarak yapılmış soyut modüllerdir. Bileşenler arasındaki yüksek düzeyde etkileşimlere "bağlayıcılar" denir. Bileşenler ve bağlayıcıların yapılandırılması, sistemin Şekil 2.1 'de gösterildiği üzere oluşturulduğu ve hareket ettiği şekli açıklamaktadır (James et al., 2003).



Şekil 2.1 Yazılım Mimarisinin Özet Tanımı

Bir program ya da bilgi işlem sisteminin yazılım mimarisini, yazılım bileşenlerini, bu bileşenlerin dışarıdan görünen özelliklerini ve bunlar arasındaki ilişkileri ihtiva eden sistemin yapısı veya yapılarıdır (Bass et al., 1997). Mimarinin karmaşıklığını basitleştirmek için, geleneksel olarak, sistem fonksiyonlar, nesneler, bileşenler ve hizmetleri içeren modülleri ile kurulmuştur.

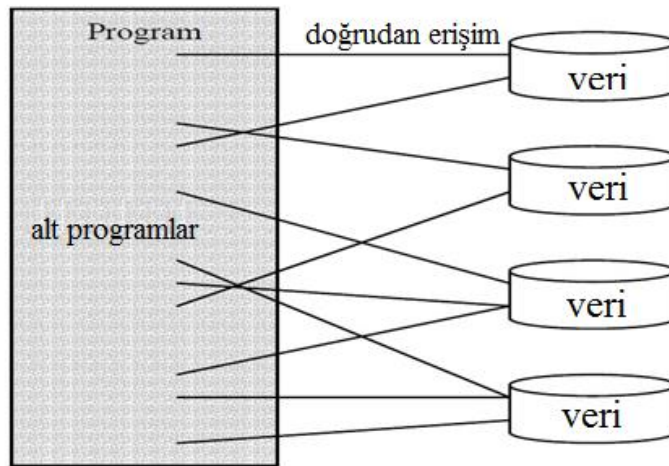
2.2 Bilgi İşlem Sistemlerinin Evrimi

Donanım teknolojilerindeki gelişmeler ve özellikle network (ağ) kurmanın buluşu; yazılım geliştiricilerinin bu geliştirmelerden daha fazla fayda elde etmelerini ve bu teknolojileri destekleyen kompozit yazılım sistemleri kurmalarını zorunlu kılmıştır. Network kurma, birkaç birleşik makineden internet üzerinden birbirine bağlı muazzam bilgi işlem kaynaklarına dönüşmüştür. Bu duruma göre, bilgi işlem sistemlerinin karmaşıklığı, büyüklüğü ve gücü; yekpare programlardan dağıtılmış bilgi işlem altyapılarına gelişim göstermiştir. Bilgi işlem

teknolojilerinde ortaya çıkan bu eğilimlere dayanarak geliştiriciler ve mimarlar; uygulama geliştirmenin eski yaklaşımlarını geçerli teknolojilere yönelik yeni mantıksal modellerin genişlemesine dönüştürmeyi mümkün kılmak için vizyonları geliştirmek durumundaydılar.

Bilgi işlemin önceki yaklaşımları kapalı, yekpare anabilgisayar sistemleriyle başladı. Yekpare uygulamalar, veri işleme ve yönetiminin tamamen merkezileştirildiği tek işlemcili sistemlerin evriminin sonucuydu.

Yordamsal gelişme tek bir makinede yürütülen ve Şekil 2.2 'de gösterildiği gibi doğrudan erişim işlemleri yoluyla verileri yönlendiren süreci içeren uygulama geliştirmenin başlangıçtaki yoluydu. Bu bilgi işlem yönteminin, programın algoritmaları arasında birçok potansiyel bağımlılıkları vardır ve bu yöntem kolayca değişiklik yapmaya izin vermez. Veri sunumu değiştirilirse, program üzerinde birden fazla yerde önemli etkileri olabilir (Raphael et al., 2004).



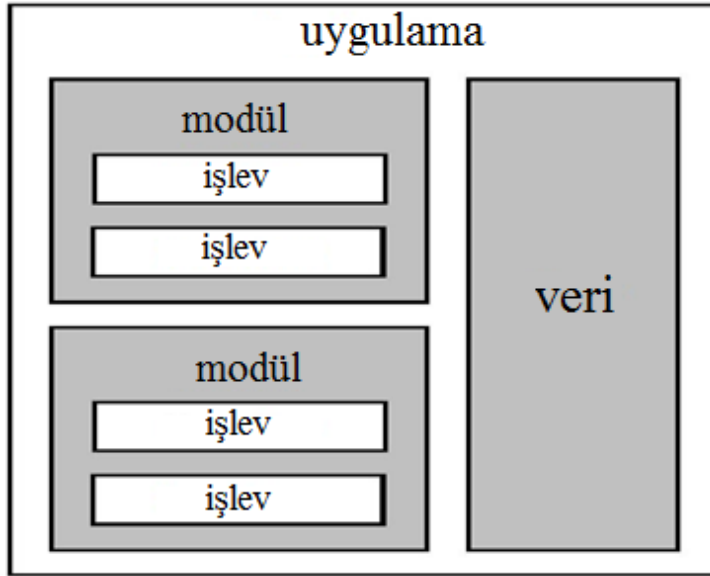
Şekil 2.2 Yordamsal Yazılım Geliştirme

FORTRAN, COBOL, C, Pascal ve BASIC, yordamsal yazılım geliştirmeye yönelik dillerdir. Bu dillerle uygulama geliştirme, verilerin fiziksel depolamasına ilişkin karmaşık bir bilgi gerektirir. Başka bir zorluk, geliştiricilerin kodların paylaşımlı kitaplıklarını hayata geçirmelerine ve uygulama yapısını modüller halinde ayrıştırmalarına rağmen, programın hala kendi içinde birçok bağımlılığının olması ve değişikliği yönetmenin zorluğudur.

Yordamsal yazılım geliştirme ile karşılaşılan zorluklar, aşağıdaki şekilde gösterildiği gibi daha büyük süreçleri küçük olanlara ayrıştırmayı içeren yeni bir yaklaşımın tasarlanmasını doğurmuştur (Yourdon and Constantine, 1975). **Yapısal**

tasarım ve geliştirme programın karmaşıklığını, uygulama içinde yeniden kullanılabilen küçük modülleri tasarlayarak azaltır. Bu şekilde, programın veri yönü ve davranış parçası ayrı ayrı belirtilmiştir.

Yordamsal yazılım geliştirmeden daha esnek ve karmaşık yazılım yapılarının gelişimini sağlayan yapısal tasarımın, bununla birlikte, ayrı modüller arasında hala bağımlılıkları vardır ve veriler etkili bir yazılım sistemi kurmak için yüksektir.

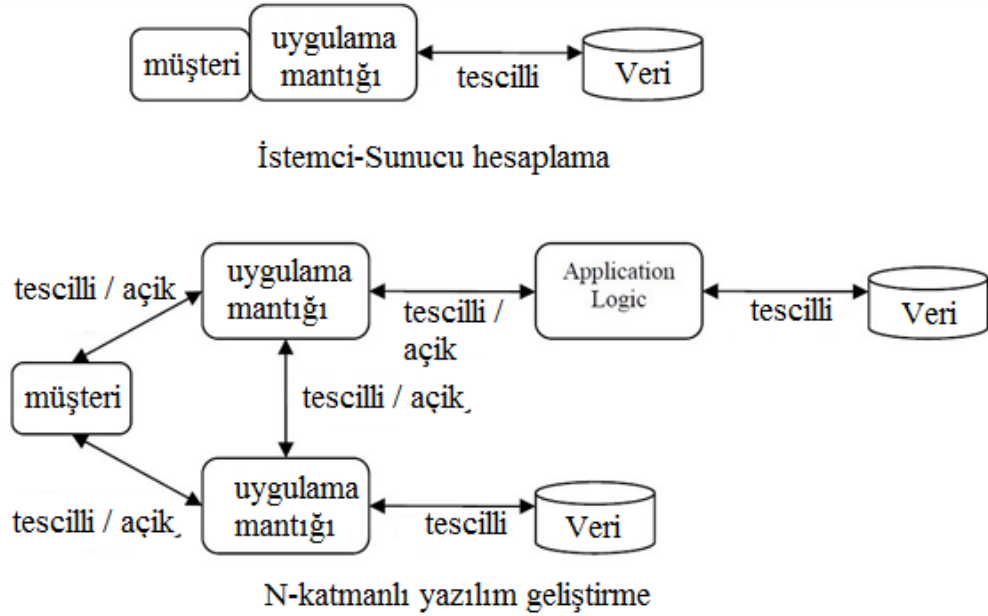


Şekil 2.3 Yapısal Yazılım geliştirme

İstemci-sunucu bilgi işlem, bireysel başvuru sistemleri kurmaya yönelik verilen öneme ilişkin olarak yazılım geliştirme evrimidir. Tasarım, istemci ve iş mantığının veri tutucu sunucudan hem mantıksal hem de fiziksel bir şekilde ayrıştırılmasını ihtiva eder. İstemci-sunucu teknolojilerin kullanıcıların ağ bağlantısı üzerinden veri işlemlerini sağlaması, yazılım geliştirmede önemli bir faktördür. Teknoloji, mevcut küresel dosya sistemlerine HTTP ve veritabanı sunucu teknolojileri gibi destekleyici protokoller üzerinden erişim için hâlihazırda kullanılan dosya paylaşım sisteminin evrimiyle sonuçlanmıştır. Bu gelişmeler, dağıtılan bilgi işlem teknolojilerinin evrimiyle yakından ilişkilidir.

Bu dönemde, hareket-işleme monitörleri, veri bütünlüğü dağıtım sistemleri genelinde tutarlı ve güvenilir bir şekilde sürdürülmesini etkinleştirmek için gelişmiştir. 1980'lerin sonu ve 1990'ların başı arasındaki bir başka gelişme, sohbet odaları ve video konferans gibi etkileşimli uygulamaların daha yüksek biçimleri ve e-postayı mümkün kılan grup yazılımı teknolojileridir.

1990'lı yılların başından itibaren, veri ve uygulama mantığı katmanının sunum katmanından temiz bir şekilde ayrıştırılmasını destekleme gerekliliği, N-katmanlı bileşen tabanlı çözümlerin istemci-sunucu teknolojilerinin yerini almasına sebep olmuştur. İlave tabakalar modüller arasındaki bağlaşımı azaltmıştır ve Şekil 2.4 'ta gösterildiği gibi daha fazla müşterinin, iş mantığı katmanına erişmesini ve görüşme yapmasını mümkün kılmıştır.



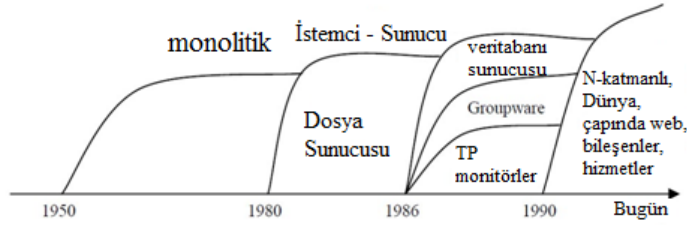
Şekil 2.4 İstemci - Sunucu ve N-Katlı Yazılım Gelişmeler

Yazılım geliştirmeye yönelik ilk yaklaşım, ürün sürümlerinin her aşamasında esas olarak özel satıcılardan kaynaklanır. Bu yazılıma Özel Yazılım denir ve çoğunlukla bu yazılımı üreten satıcıya bağlıdır.

Bugünün sistemleri değişen derecelerde özel yazılıma dayalıdır. Özel yazılımın kalite ve yetenekleri, sürekli hata ayıklama ve son kullanıcının yazılım sahibinin satıcısından beklendikleriyle tutarlı destekleyici özelliklerinden dolayı yüksek olabilir.

Ticari yazılımın diğer büyük kategorisi Açık Yazılımdır. Bu yazılım, birden fazla satıcının özel yazılımlardan bağımsız teknoloji özelliklerini oluşturmak için işbirliği yaptığı yazılım geliştirmeye yönelik bir yaklaşımdır. Açık yazılımın temel yararı, birçok son kullanıcıya uygun standart teknolojiyi kurmak için temel olan yazılım yapısının muntazam bir terminolojisini sunmasıdır. İlave yararı, farklı yazılım uygulamaları arasında sunabileceği birlikte çalışabilirliktir.

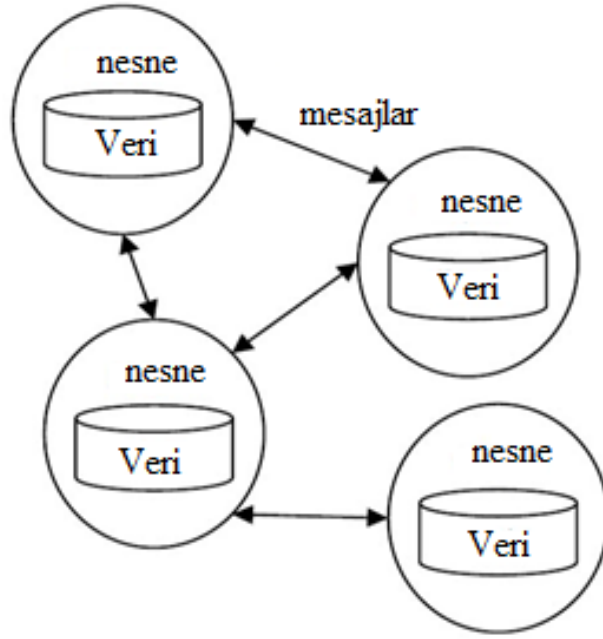
World Wide Web' in sunulmasıyla birlikte, yazılım geliştirme için yeni bir dönem gelişmektedir. İnternet ilk olarak bilimsel makaleleri yayınlamanın bir yolu olarak başlamıştır ve dinamik HTML (Köprü Metni Biçimlendirme Dili) ve sonunda da temel, standart, farklı platformlar arasında veri alışverişi için teknoloji sağlayan bir üst dil olan XML (Genişletilebilir İşaretleme Dili) haline gelmiştir. XML verileri, uygulamadan bağımsız bir şekilde açıklar ve Web hizmetleri, heterojen bilgi işlem ortamları arasında dağıtılan süreçlerinin paylaşımını etkinleştirmek için bu teknolojiyi kullanırlar. Bugünün uygulama geliştirmesi, güvenlik duvarı üzerinden erişilebilir ve bağımsız servislerin oluşturulmasını içerir ve ayırık fonksiyonu destekler. İstemciler, uygulama altyapısını kurmak için oluşturulan servislerle etkileşim halindedir.



Şekil 2.5 Bilgi İşlem Sistemlerinin Fiziksel Evrimi

2.3 Nesne Yönelimli Geliştirme

Nesne Yönelimli Geliştirme, hem veri hem de davranışları **sınıflar** olarak adlandırılan soyut veri türlerinde saklayan yazılımın gelişimini destekler (Booch, 1990). Sınıfların örnekleri küçük modüller halinde oluşturulmaktadır, Şekil 2.6 'da gösterildiği gibi, nesneler olarak adlandırılır. Veri temsilindeki herhangi bir değişiklik, sadece verileri saklayan mevcut nesneyi etkiler. Sınıflar sonsuza dek yaşayabilir, ancak nesnelerin sınırlı bir ömrü vardır.



Şekil 2.6 Nesne Yönelimli Geliştirme

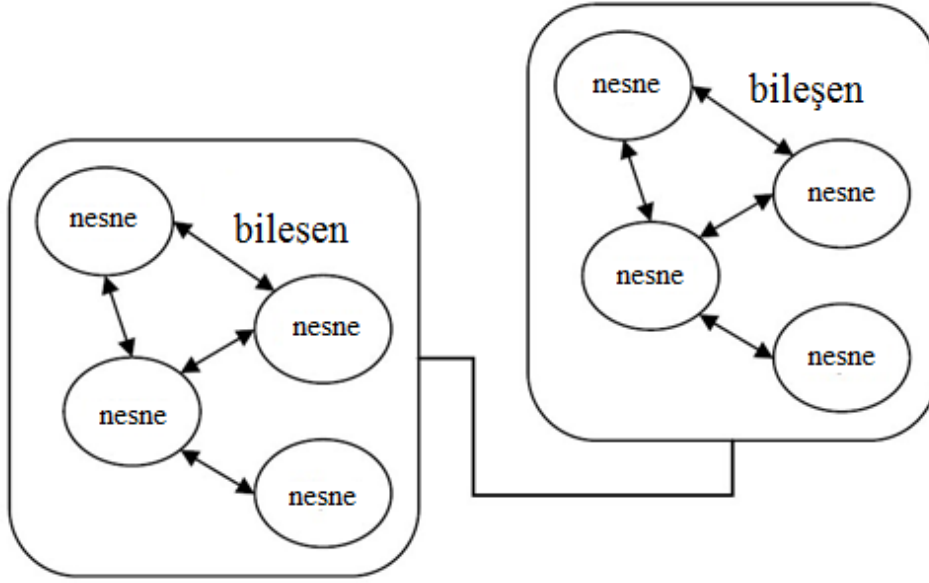
Nesneler mesajlaşma yoluyla birbirleriyle iletişim kurarlar. Nesne tabanlı geliştirme, davranış ve verileri nesneler ve sınıflar aracılığıyla saklamaya yönelik daha fazla destek sağlayarak yazılım tasarımı ileri götürür. Nesneler arasında neredeyse hiçbir bağımlılık yoktur, ancak birbirine bağlı çok sayıda nesne, yönetilmesi zor olabilecek bağımlılıklar oluşturur.

2.4 Bileşen Yönelimli Geliştirme

Bileşenler nesnelerden daha karmaşık yazılım modülleridir ve sistem düşüncesinde, yazılım süreçleri ve teknoloji kullanımında köklü değişiklikler gerektirir.

Bileşen tabanlı geliştirme, geliştiricilerin daha karmaşık, yüksek kalite sistemleri oluşturmalarına olanak sağlar, çünkü uygulama içerisindeki bağımlılıkları ve karmaşıklıkları yönetmenin daha iyi araçlarını sunar.

Yazılım bileşeni sözleşmeyle belirlenmiş ara yüzleri ve açık içerik bağımlılıkları olan bir bileşen birimi olarak tanımlanır. Yazılım bileşeni bağımsız olarak kurulabilir ve üçüncü şahıslar tarafından bileşimine bağlıdır (Szyperski, 1998). Aşağıdaki şekilden görülebileceği gibi belirtilen bir ara yüze sahip, uygulama işlevi sağlamak için birlikte çalışan nesneler grubudur.



Şekil 2.7 Bileşen Yönelimli Geliştirme

Bileşen, tek bir uygulama mantığından bütün fonksiyonel sistem için birçok farklı yazılım yapısı anlamına gelebilir. Tüm durumlarda bileşen, bir veya daha fazla iyi tanımlanmış ara yüze sahip bir yazılım paketidir. Bileşen, işlem yönetimi ve veritabanı bağlantı havuzu gibi gerekli işlevleri sağlayan J2EE kapsayıcısı gibi uygulama sunucusu tarafından sunulan bir bileşeni yürütme ortamında çalıştırılır.

Bileşenler, kalıtım özelliklerini azaltmaları haricinde kapsülleme ve polimorfizm gibi nesne yönelim özellikleriyle örtüşür. Bileşen düşüncesinde kalıtım sıkıca birleştirilmiştir ve ambalaj ve yeniden kullanımın birçok biçimi için uygun değildir.

Bunun yerine, bileşenler diğer nesneleri ve bileşenleri kalıtsal devralma yerine çalıştırarak işlevselliği yeniden kullanırlar. Bileşen terminolojisinde bu yürütmeye yetkilendirme denir (Raphael et al., 2004).

2.5 Dağıtılmış Bilgi İşlem

Servis Yönelimli Mimari (SOA) dağıtılmış bilgi işlemin doğrudan bir çıkarımı olmamasına rağmen, mevcut ara katman teknolojilerini ve dağıtılmış bilgi işlem kavramlarını birleştirmelidir. Başarılı bir SOA, elde edilebilen kavramların ve teknolojilerin göz önüne alınmasıyla uygulama geliştirme ve

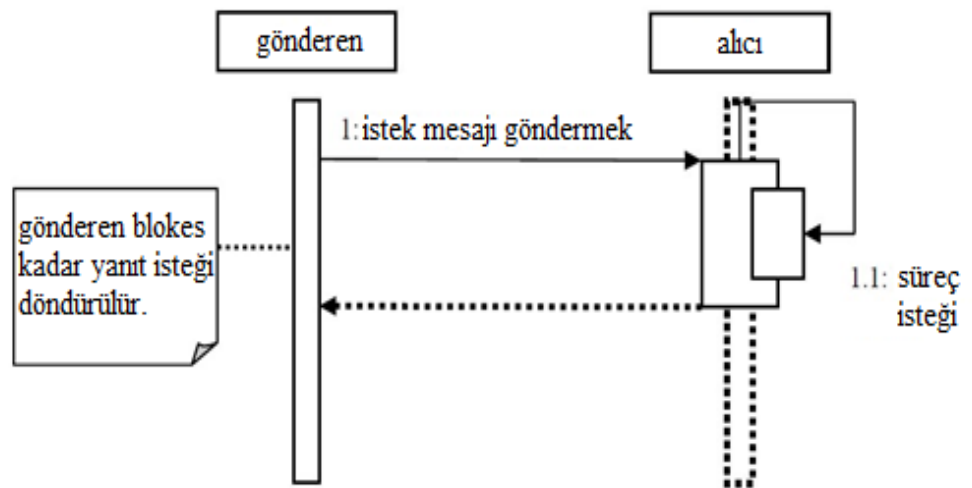
yaklaşmakta olan teknolojilere yönelik etkili bir yaklaşımı destekleyerek mevcut katman teknolojileriyle karşılaşılan zorlukların üstesinden gelmelidir.

Dağıtılmış bilgi işleme yönelik önceki yaklaşım doğrudan ham fiziksel ağ protokolüne göre iki adet dağıtılmış program arasında iletişim kurmaktır (Dirk et al., 2004). SNA, TCP / IP ve IPX gibi daha yüksek seviyeli protokoller, uygulama çabaları ve teknoloji bağımlılıklarının azaltılmasına yardımcı olan API' leri sağlamıştır. Bir sonraki evrimsel adım olarak iletişim katman çatısı; işletim sistemleri, ağ protokolü alt düzey bilgileri ve fiziksel ağ adresi gibi teknik detayları bilmeden uzak bir uygulamaya erişme olanak sağlar.

Uygulamalar, temel olarak iki iletişim modunda birbirleri ile iletişim kurar:

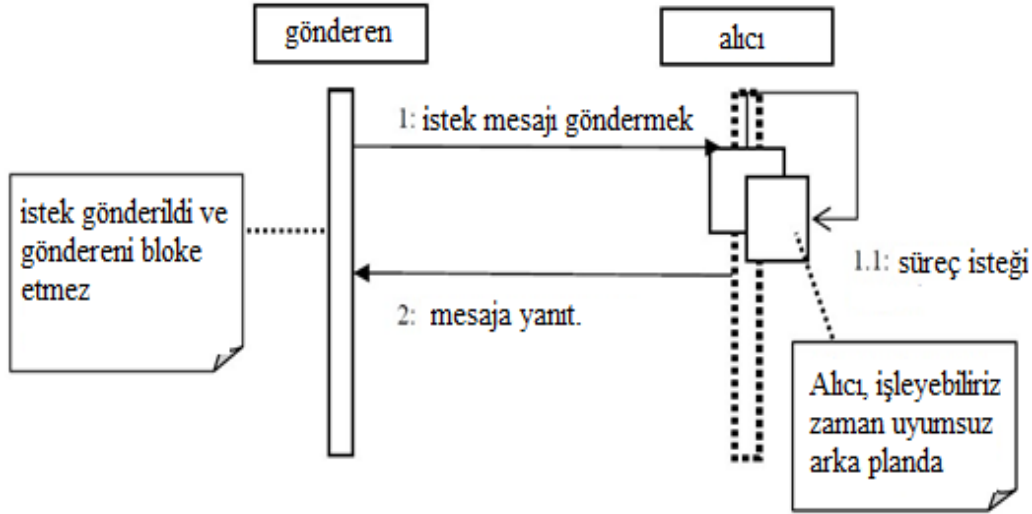
Senkron ve asenkron mekanizması: İletişimin bu temel modlarına ilişkin genellikle sayısız varyasyonlar mevcuttur.

Senkron mesajlaşmada, mesajlarının değiş tokuşu 'Simulta' gerektirir.



Şekil 2.8 Yanıtlı Senkron Mesajlaşma

Asenkron mesajlaşmada, istemci bir mesaj gönderir ve geri dönüş beklemeden işleme devam eder. Önceden tanımlanmış ve üzerinde anlaşılan son nokta gibi kimi öncül anlaşmaya dayanarak, alıcı göndericiye bir yanıt gönderir.

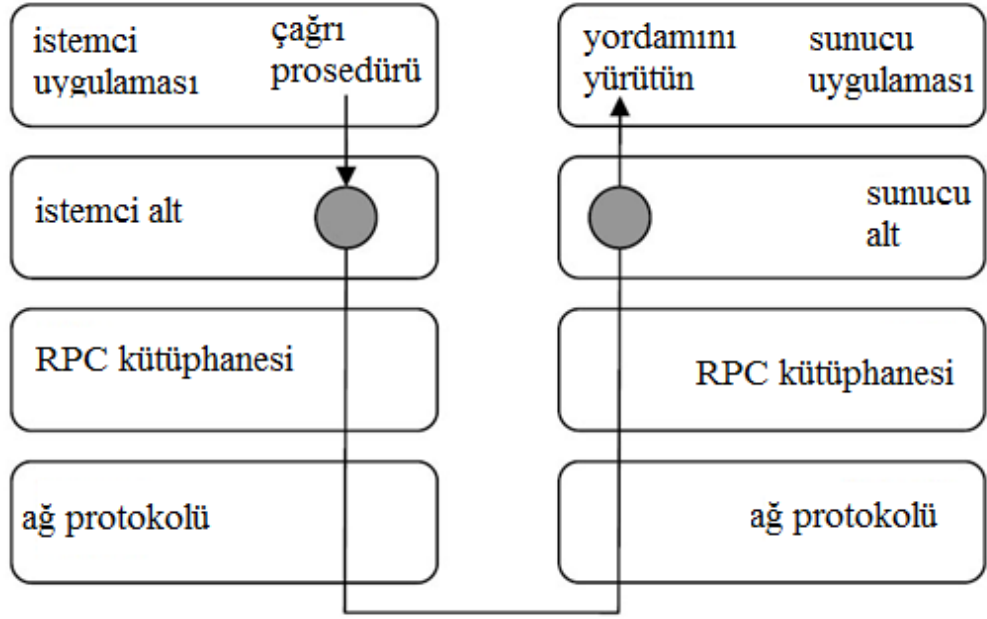


Şekil 2.9 Yanıtlı Asenkron Mesajlaşma

2.5.1 Uzaktan yordam çağırısı

Uzaktan Yordam Çağırısı (RPC) kavramının gelişimi 'Sun Microsystems' tarafından 1980'lerin ortalarında ivede kazandı ve RFC protokolleri 1050, 1057 ve 1831 olarak belirtilmektedir. Bu özelliklere sahip iletişim altyapısına, uygulanması uygun RFC' lere dayalı olmasa dahi RPC tarzı denir.

RPC, kodu içeren ve uzaktan erişim için yeniden kullanılabilir hale getiren dağıtılmış uygulamanın işlev veya yordamının yürütülmesini gerektirir. Uzaktan arama, yürütüldüğü ve sonucun arayan kişiye iade edildiği ağı üzerinden başka bir uygulamaya yönlendirilir. Çoğu RPC uygulaması senkrona dayalıdır, istemcinin sonucu talebine cevap verene kadar engellenmesini içeren cevap protokolünü talep eder.



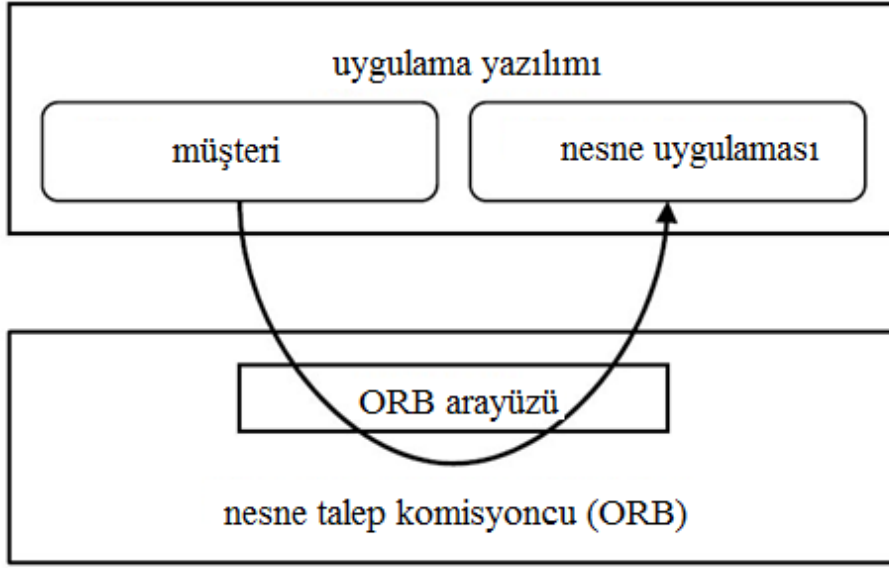
Şekil 2.10 Uzaktan Yordam Çağrısı Kavramsal Modeli

RPC modelinin birçok bileşeni vardır. Ağ protokolü, düğümden düğüme verileri iletmek için sanal devreler arasında geçiş, yönlendirme, paket sıralama, adresleme ve yönlendirme sağlar. **RPC** kitaplığı belirli teknolojiye özgüdür ve ana bilgisayarlar arasında veri aktarımı sağlar. Uçtan uca hata kurtarma ve akış kontrolünden de sorumludur. İstemci ve sunucu taslakları, uygulamalar arasında diyalogları, alışverişleri ve bağlantıları kurar, koordine eder ve sonlandırır. Son olarak uygulama katmanı, iş işlevselliğinin yürütüldüğü son kullanıcı işlemlerini ve gerçek uygulamayı ihtiva eder.

2.5.2 Dağıtılmış nesneler

1990'ların başında ortaya çıkan Dağıtılmış Nesneler kavramı, geleneksel modüler programlama stillerinin yerine geçmek üzere uzaktan yordam veya işlev çağrılar sağlayan nesne yönelimli programlama evrimidir.

Tipik olarak dağıtılmış nesneler, olası uzaktan nesneler ile iletişim ve veri alışverişini yöneten Nesne Talep Aracısı (ORB) tarafından desteklenir. Nesne yönelimli dağıtım platformu, konum şeffaflığı sağlar ve nesnelerin uygulama detaylarını müşterilerden gizlemelerini mümkün kılar.

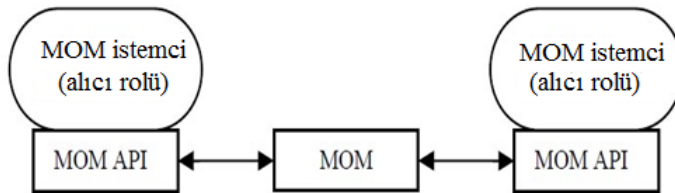


Şekil 2.11 Nesne Talep Aracısı

CORBA, en yaygın RPC tabanlı ORB uygulama teknolojisidir. Veri veya program yürütme hizmetlerine özel ilgi göstermez, çünkü temel amacı, düzgün dağıtılmış nesne çatısının hayata geçirilmesini sağlamaktır.

2.5.3 Mesaj temelli orta katman

Mesaj temelli Orta Katman (MOM), iletişim kuran taraflar arasında yer alan ve çeşitli uygulamaları bağlamak için mekanizma sunan tipik olarak bir yazılım parçasıdır. Aralarındaki işletim sistemi, donanım ve iletişim protokolleri gibi farklı bağımlılıkları işlemekten sorumludur. MOM, dağıtılmış uygulamaların mesajlar üzerinden birbirleri ile iletişim kurmak için arka plandaki MOM' u nasıl kullanmaları gerektiğini tanımlayan bir API kullanarak olanaklarını ortaya koyar. Mesajlar, mesaj alındıktan sonra yürütülmesi gerekli belirli bir fonksiyonla ilgilidir.



Şekil 2.12 Mesaj temelli Orta Katman

MOM tabanlı mesajlaşma çözümleri, MOM' un uygulama bileşenleri arasında bir ileti otobüsü olarak hareket ettiği merkezi topla ve dağıt topolojisi veya kendi MOM' larını kullanan iletişim tarafları arasında ki bir yönlendirici olarak davranan merkezi **MOM'** a sahip **karma MOM** topolojisi olarak tasarlanan merkezi MOM gibi çeşitli topolojilerde tasarlanabilir.

MOM, iki uygulamanın konuşma ve veri alışverişi kurlmalarını mümkün kılan ortamı sunarak önemli bir rol oynar. İletişim ara katman özellikleri aşağıdaki gibi tanımlanabilir:

- Gönderici ve alıcının aynı anda bağlı olmasını gerektirmez
- Katılımcılar arasında istek ve cevaba ilişkin güçlü teslimat garantisi sağlar
- Bazı durumlarda, katılımcılar arasında mesajları çevirerek ve biçimlendirerek işlevsellik sağlar. MOM, mesaj tüketicileri ve mesaj üreticileri arasındaki gevşek bağlantıyı teşvik eder ve dinamik, güvenilir ve yüksek performanslı sistemlerin kurulmasına olanak sağlar. Ancak, MOM-tabanlı sistemlerin arka plandaki karmaşıklığının düzgün ve verimli bir çalışma sisteminin gelişimini sağlamasına ilişkin zorlu durum yabana atılmamalıdır.

2.6 Servis Yönelimli Mimarinin Tanımı

Servis Yönelimli Mimari (SOA), işletmelerin farklı iş ortakları ve platformları arasında dinamik bağlantı kurlmalarını ve yazılım uygulamalarını iletmelerini uygulama yapı blokları olarak kullanılabilen jenerik ve güvenilir hizmetler sunarak sağlayan yöntem ve teknolojileri kullanan mimari bir tarzıdır. Böylece, daha zengin ve daha gelişmiş uygulamalar ve bilgi sistemleri geliştirmek mümkündür.

SOA yeni bir kavram olmamasına rağmen, özellikle Web Hizmetleri icadından sonra, bu alandaki yeni gelişmeler, yazılım uygulaması mimarilerinin oluşturulmasına ilişkin yeni yollar getirmiştir. Mevcut yazılım altyapılarını yeniden oluşturmaya yönelik yeni yaklaşım ve mevcut hizmetlere göre diğer işletmeler ile iletişim imkânı:

Bununla birlikte, SOA oluşturmak hala aşağıdaki nedenlerle zordur:

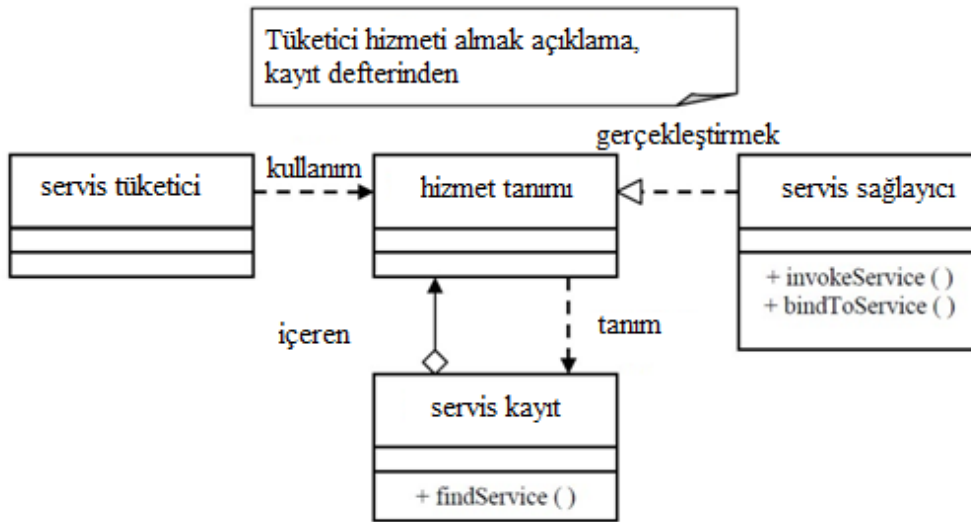
- SOA' nın yazılım kaynaklarına yaklaşım metodu geleneksel mimarilerden farklıdır.
- SOA bir düzeyde mimari düzenlemeye ihtiyaç duyar.

- SOA uygulaması, farklı işletmeler tarafından erişilme yeteneğinde bir ortama ihtiyaç duyar.

Servislerin tanımı ve bu servislerin güvenli ve yönetilen bir ortamda yenileriyle birleşimi, belirli bir yoğunlaşma gerektiren başka bir yönüdür.

2.6.1 SOA varlıkları ve özellikleri

Servis Yönelimli Mimari, hizmet tüketicisinin kayıt defterinde arama yoluyla ihtiyaçlarını eşleştirmek üzere servis sağlayıcı ile etkileşime girdiği üç ana işlevsel birim arasında bir etkileşim modeli tanımlayan mimari bir tarzıdır. Bu etkileşimi açıklayan meta-modeli aşağıdaki şekilde gösterilmiştir.



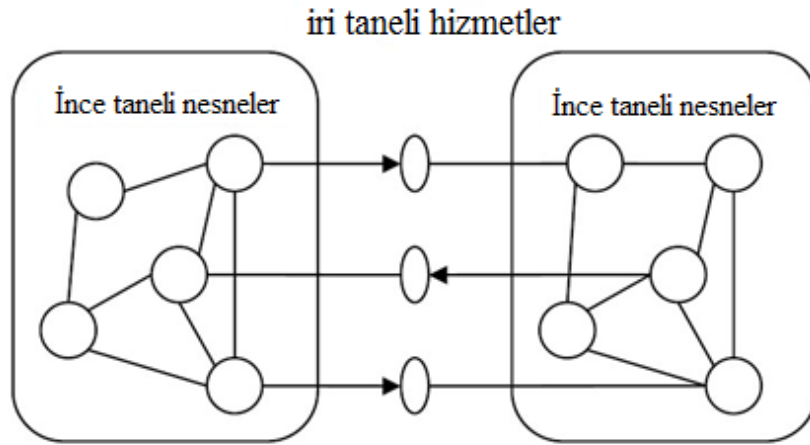
Şekil 2.13 Servis Yönelimli Mimarinin Kavramsal Modeli

Yazılım yapı birimi olarak gevşek eşleşmiş hizmetlerinin geliştirilmesine ilişkin servis yönelimli mimari gerilimi. Hizmet, özerk olarak yürütülecek başka hiçbir belirli sistem gerektirmez. Bununla birlikte, sıkı eşleşmeden ara yüz tanım düzeyinde kaçınılması mümkün değildir veya belirli bir protokole bağlı olamaz.

- Hizmetler ağ adresli bir ara yüzü sahiptir. Servis, servis yönelimli mimari tasarım ilkelerine uymak için ağında kendi ara yüzünü yayınlamak zorundadır böylece tüketici, servisi ağ üzerinden dağıtılmış bir şekilde çalıştırabilmektedir. Böylece servisi, farklı tüketicilerle herhangi bir zamanda kullanmak muhtemeldir. Servise ağın kullanımı olmadan aynı zamanda yerel bir ara yüz üzerinden ulaşılabilir.

Bununla birlikte, SOA kurulumundaki esas amaçlarından biri, servislerin yerden bağımsız bir şekilde tüketilmesini sağlamaktır.

- Servisler kaba taneli ara yüzlere sahiptirler. Öge boyu kavramı, yazılım sistemi içinde ara yüzleri uygulama yoluyla ilgilidir. Ara yüz, tam bir iş mantığı için gerekli bütün işlevleri destekliyorsa, kaba taneli bir ara yüzüdür. Bunun aksine, ara yüzün belirli işlevlerin sadece bir parçasını uygulaması halinde, ince taneli olarak kabul edilir. Öge boyu tüm hizmet uygulamasına aynı zamanda uygulanabilir ve ara yüz yürütmeye ilişkin münferit yöntemlere uygulanabilir. Servis yönelimli yazılım sistemi, doğası gereği farklı öge boyu seviyeleriyle iri taneli ara yüz tasarımını destekler. Hizmeti oluşturan nesneler yine de ince taneli olabilir, bununla birlikte bu nesneler servisin fiziksel yapısı içinde tutulurlar.



Şekil 2.14 Öge boyu

- Hizmetlerin konumu şeffaftır. SOA, servis tüketicinin, kayıt defteri üzerinden çalışma zamanında işbu servisi yürütene kadar servisin konumu hakkında ön bilgisinin bulunmamasını ifade eden konum şeffaflığını teşvik eder. Tüketici ve sağlayıcı arasındaki tek bağımlılık, tüketiciyi etkilemeden bir konumdan diğer konuma kayabilecek servis sözleşmesidir.
- Servisler yeni uygulamalar içinde oluşturulabilir. SOA' nın diğer önemli özelliği, mevcut servislerden türetilen yeni uygulamaların oluşturulmasını sağlamasıdır. Bileşim, hangi uygulamaların gelecekte servisi kullanacağına ilişkin ön bilgi olmadan ağırlıklı olarak hizmet modülerliğine ve servis işlevselliğinin yeniden kullanımına odaklanan etkili bir tasarımıdır.
- SOA, kendini iyileştirmeyi destekler.

Kendini iyileştirme, sistemin yürütülmesi esnasında bir hatanın oluşması durumunda kendini kurtarmasına ilişkin sistem özelliği olarak tarif edilmiştir. Servis yönelimli sistemler, kendini iyileştirmeye diğer mimari stillerden daha fazla önem vermelidir çünkü servisler, bir işi çalışma zamanında yürütmek üzere birleştirilmiştir. Tüketici, azılım sisteminin doğru ve düzenli yürütülmesi amacıyla ağdaki farklı servisleri bulma yeteneğine sahip olmalıdır.

2.6.2 Servis odaklı geliştirme

Servisler, birden çok bileşen ara yüzlerinin, belirli bir işlevi gerçekleştirmek için tek bir ara yüzde meydana geldiği bileşenlerin evrimidir. Servis, bir görevi yerine getirme yeteneğine haiz soyut bir kaynaktır (W3C, 2004). Servisler, iş fonksiyonlarının ve aynı zamanda teknik görev tanımlarının olası yansımasına sahiptirler.

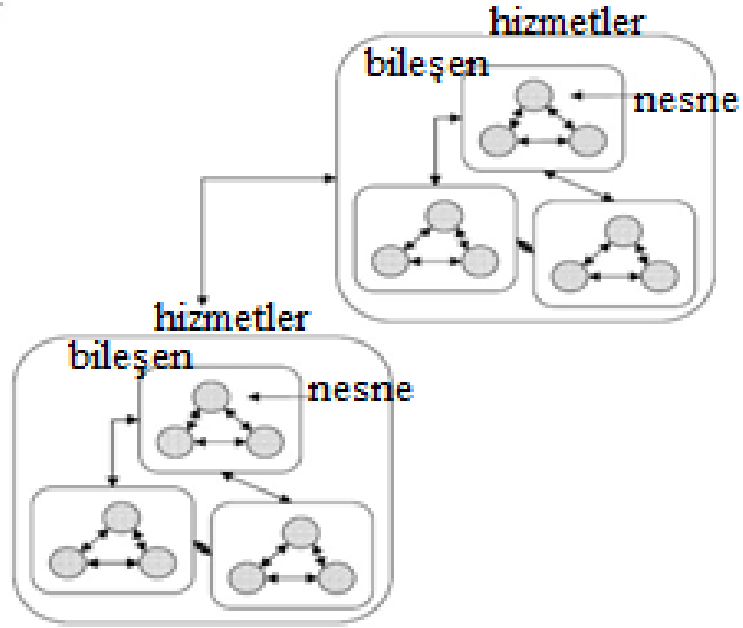
Servis tabanlı geliştirme, bileşen tabanlı geliştirmeyi ilerletir. Bu ilerleme, servislerin farklı programlama dillerinde gerçekleştirilen dağıtılmış ve birlikte çalışan bileşenleri ihtiva etmesi, ama aynı zamanda servis yönlendirmesinin, platform bağımsızlığı, servislerin dinamik keşfi ve yeniden kullanılabilen uygulama modüllerinin geliştirilmiş seviyesi gibi özellikleri desteklemesi şeklindedir.

Servis yönelimi, servislerin internet ağları ve World Wide Web teknolojilerinin doğal semeresi olduğu günümüzün servis tabanlı uygulama geliştirmesinde olduğu gibi evrimleşmiştir. Bu teknolojiler, daha dinamik ve açık yazılım sağlamak ve birlikte çalışabilirlik sorunlarını çözmek üzere birçok yazılım kuruluşları ve satıcıları tarafından kabul gören standart gelişmeleri uygulayarak uygulama yapısını değiştirebilirler.

Servisler, aşağıdaki özellikleri desteklemek üzere tasarlanmış ve geliştirilmiştir:

- Her bir servis belirli bir iş işlevini tanımlar ve gerçek-yaşam aktiviteleriyle eşleşebilir.
- Servisin çeşitli prosedürleri ve işlemleri olabilir.
- Servisler, diğer servisler ve sistem bileşenleriyle, iş hedeflerini gerçekleştirmek için gevşek eşleşmiş, mesaj odaklı bir ortamda etkileşim içindedir.
- Servislerin ara yüzleri açıkça tanımlanmıştır ve birçok farklı diğer servis ve uygulamalarca kullanılabilirler.

- Servislerin dağıtılmış bir ortamda olması gerekmez. Aşağıdaki şekil, servis tabanlı geliştirmeyi, bileşenler ve nesne bağlamında göstermektedir.



Şekil 2.15 Servis Tabanlı Geliştirme

Yazılım geliştirme, platformdan bağımsız yeniden kullanılabilir servislerin geliştirilmesini sağlayana kadar mimari modellerin bazı aşamalarından geçmiştir. Her bir mimari model, gerçek dünya gereksinimlerine uyan gelişmiş yazılım yapıları gerçekleştirmek için zorluklarla önceki yaklaşımlardan daha iyi araçlarla başa çıkmayı sağlar.

Yazılımın önceki mimarisi, güçlü ve gelişmiş uygulama geliştirmeyi etkinleştirmek üzere yazılım yapıları ve sınırlı destek gelişimi hususunda katı kuralları olan yapısal tasarıma dayalıdır. Nesne yönelimli teknolojiler, iş mantığının daha iri taneli fonksiyonları ve sınıfları yoluyla kapsüllenmesini destekleyen esnek yazılım geliştirmeyle sonuçlanır; ancak, sağlam uygulama geliştirmenin somut faydaları, bileşenler ve hizmetlerin ilerlemesi vasıtasıyla elde edilir.

Aşağıdaki tablo, her bir yazılım mimari modelinin yapısı ve özelliklerini ele alır.

Tablo 2.1 Mimari Geliştirme Modellerinin Karşılaştırılması.

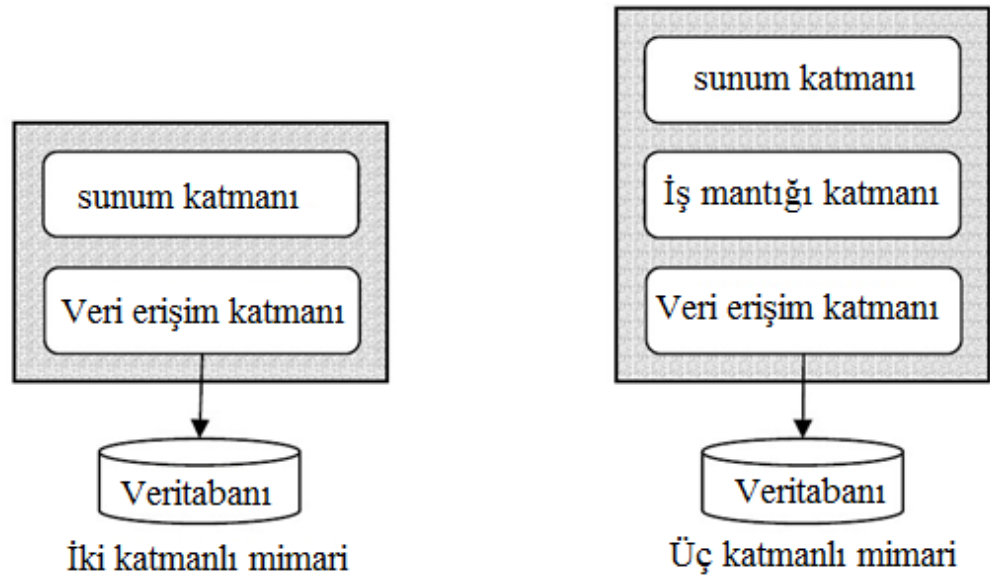
Yapılandırılmış Geliştirme	Nesne Tabanlı Geliştirme	Bileşen Tabanlı Geliştirme	Servis Tabanlı Geliştirme
Çok güzel bir sistemdir yapılanması.	Küçük - tohum sistemi yapılanması	Orta - tohum sistemi yapılanması	Kaba - tohum sistemi yapılanması
Düşük yeniden kullanılabilirliği	Düşük yeniden kullanılabilirliği	Orta yeniden kullanılabilirliği	Yüksek yeniden kullanılabilirliği
Sıkı bağlama	Sıkı bağlama	Gevşek bağlama	Gevşek bağlama
Zamanı bağımlılıkları derleme var.	Zamanı bağımlılıkları derleme var.	Zamanı bağımlılıkları derleme var.	Sadece zamanı bağımlılıkları var.
Intra-uygulama iletişim kapsamı	Yapı taşları tek tek sınıfları vardır	Yapı blokları çeşitli sınıflardan oluşur (bileşenler)	Yapı blokları bileşenlerden oluşur
	Encapsulation, Kalıtım, Polimorfizm	Etkileşim, bağlantı ve Bileşenlerinin exchangeability	Yayınlanmış arayüz tanımı
	İşlevsellik sınıf bildirimleri tarafından açıklanmıştır.	İşlevsellik arayüzü bildirimleri tarafından açıklanmıştır.	Dinamik keşfedilebilir hizmetleri dağıtıldı.
	Nesnenin dinamik bağlı ama çok sayıda.		İşlevsellik ağ adresli bileşen arayüzü bildirimleri tarafından açıklanmıştır.
			Inter-kurumsal iletişim kapsamı.

2.6.3 SOA katmanlı mimari

Daha önceki sistemler, istemcinin aralarında herhangi bir mantıksal model olmadan veritabanı ve ağ API'lerine doğrudan erişebildiği iki katmanlı olarak yapılandırılmıştır. Bu yaklaşım, geliştirmenin küçük ölçekli ve prototip olduğu yazılım sistemlerinde hala kullanılabilir.

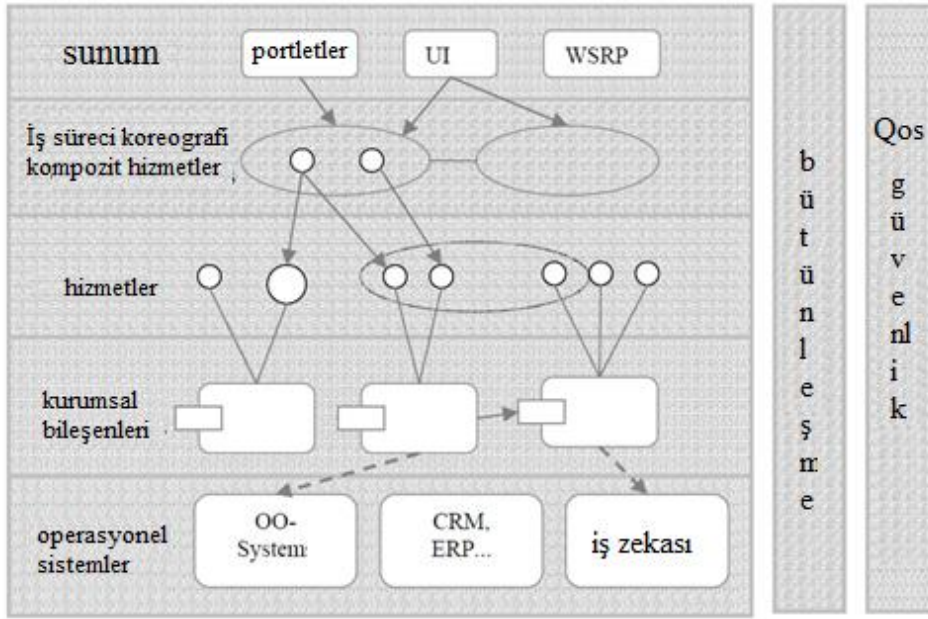
Ayrıca gelişmiş sistemlerdeki bazı modüller, belirli işlevleri sağlamak üzere bu yaklaşıma başvurabilirler. Bununla birlikte iki-katmanlı uygulama geliştirme, kısa yaşam döngüsü sistemleriyle sınırlıdır ve esnek API' leri desteklemez. İstemciden gelen çok sayıda eşzamanlı kullanıcı için mimariyi ölçeklenemez ve katı kılan yeterli uygulama kodu yalıtımı sağlamaz.

Şu anda en sık kullanılan uygulama geliştirme modeli, istemci ve veri depolama katmanları arasında ek bir katmanı destekleyen üç katmanlı mimari yapıya dayanmaktadır. İş mantığı katmanı olarak adlandırılan ek tabaka, istemciden gelen kod izolasyonu ve çeşitli istemci uygulamaları arasında uygulama mantığı paylaşımını sağlar. Verilerin esnek yönetimi ve sistem kaynaklarının kullanımı hususunda yazılım geliştirme için yetkili bir yaklaşımdır.



Şekil 2.16 İki katmanlı ve üç katmanlı Mimari Modeller

SOA, Şekil 2.17'de gösterildiği gibi belirli işlevleri sağlamaktan ve servis (Ali, 2005) gerekliliklerinin kalitesini korumaktan sorumlu bileşenlerin üzerine katmanlandırıldığı tek-katmanlı uygulama geliştirmeye dayanmaktadır.



Şekil 2.17 Servis Odaklı Mimarinin Katmanları

SOA içinde her katman, aşağıda tarif edildiği gibi kendine özgü mimari özelliklere sahiptir:

Operasyonel Sistemlerin Katmanı: Bu katman, CRM ve ERP paket uygulamaları, nesne yönelimli sistemleri ve iş zekâsı uygulamaları da dâhil olmak üzere mevcut uygulamaları içerir. Bu uygulamalar, servisler için arka plan sağlar ve bunların her biri kendilerine özel yapılara ve veritabanlarına ve diğer sistem kaynak erişimine sahiptir.

Kurumsal Bileşenler Katmanı: Bunlar, servislere yönelik belirli işlevleri ve gereksinimleri sağlayan özel bileşenlerdir. Onlar, servis uygulamaları için işletme varlıkları ve yönetim, kullanılabilirlik ve hizmet yük dengelemesi gibi diğer sistem ihtiyaçlarıdır.

Hizmetler Katmanı: Bu katman, keşfedilebilir ve işletmeler için belirli bir iş fonksiyonu sağlamak üzere işletmelere yönelik belirli iş fonksiyonu temin eden diğer uygulamalar tarafından çağrılan gerçek servisleri içerir. Servisler, hizmet tanımları olarak kurumsal bileşen ara yüzleri gerçekleştirir ve dağıtır ve ağ üzerinde yayınlanır.

İş Süreci Bileşim Katmanı: Servisler, özel kullanım durumları ve iş süreçlerini destekleyen hizmet orkestrasyon veya koreografisi yoluyla tek bir uygulama içinde oluşturulabilir.

Sunum Katmanı: Servisler ve kompozit uygulamaları için kullanıcı ara yüzleri sağlar.

Sunum katmanı SOA için doğrudan bir husus olmamasına rağmen, servisleri kullanıcı ara yüzü olarak kullanım hedefinden dolayı portletler ve Uzaktan Portlet özellikleri için Web Servisleri gibi standartların geliştirmesine neden olmuştur.

SOA' ya ilişkin diğer bir endişe; güvenilirlik, doğru yönlendirme ve hizmetlerin koordinasyonu gibi özellikleri destekleyerek ve protokol ve bütünleştirici taraf anlaşmaları dâhil olmak üzere diğer teknik detayları yöneterek kuruluş içerisinde servisler ve kompozit uygulamaların bütünleştirilmesidir. Servis tabanlı uygulama mimarileri tasarlarken açıklığa kavuşturulması gereken QoS (servis kalitesi) gereksinimleri, güvenlik, yönetim ve hizmetlerin izlenmesi de diğer gereksinimlerdir.

2.7. Servis Yönelimli Mimari Teknolojileri

Jini Ağ Teknolojisi olarak adlandırılan ilk servis yönelimli teknoloji 1990'ların sonlarında sun Microsystems tarafından tanıtılmıştır (Sun Microsystems, 2002). Jini, servislerin bir ağ üzerinde dinamik olarak keşfedilmesi ve kullanılması için hafif bir ortamdır. Başlıca amacı, yazıcılar gibi cihazların dinamik olarak ağa bağlanmalarını ve mevcut hizmetleri kaydetmelerini mümkün kılmasıdır.

Sonuç olarak, servisleri yazılım yapı birimi olarak kullanma ilgisi, servis odaklı mimariyi uygulamak için daha fazla teknoloji grubunun gelişmesine yol açar.

Bununla birlikte SOA oluşturma zorlukları, özellikle birlikte çalışabilirlik, entegrasyon karmaşıklığı ve endüstri standartlarına taşıma ile ilgilidir ve servis tabanlı uygulama geliştirme bugünün yaklaşımını etkilemektedir.

2.7.1 J2EE

Java 2 Platform Kurumsal Baskısı (J2EE), bileşen tabanlı dağıtılmış uygulamaların tasarım, geliştirme ve dağıtımını destekleyen kapsayıcı-merkezli bir teknolojidir (Sun Microsystems, 2006), J2EE, uygulama mantığının işlevsel

olarak farklı bileşenlere ayrıştırıldığı çok katmanlı uygulama mimarisini kullanır ve her bir bileşen farklı makineler üzerinde çalışabilecektir.

J2EE bileşenleri Java dilinde yazılmıştır ve J2EE uygulaması içerisinde oluşturulmuştur. Java uygulamaları, J2EE uyumlu uygulama sunucusu tarafından sağlanan altyapı hizmetleri ile harici sistemlere erişim sağlar. J2EE teknik özellikleri, aşağıdaki J2EE bileşenlerini tanımlar:

– **İstemci bileşenleri:** Uygulama istemcileri ve uygulamaları

– **Web bileşenleri:** Java Servlet'ler ve Java Sunucu Sayfaları

– **İş bileşenleri:** Enterprise JavaBeans

J2EE teknik özellikleri, çeşitli servisler sunmaktadır ve bu servislerin her birinin servis odaklı uygulama geliştirme oluşturmada kullanılabilecek farklı işlevleri vardır. İstemciler, sağlanan API'ler tarafından bu servislere erişirler. Bazı temel servisler aşağıdaki gibi tanımlanmıştır:

– **JavaBeans:** JavaBeans, kodlama standartlarına ve kurallarına uyan sıradan Java sınıflarıdır.

– **Enterprise JavaBeans (EJBs):** EJB' ler dağıtılabılır, bir uygulama sunucusu tarafından hayata geçirilen altyapıya dayanan sunucu tarafı bileşenleridir. EJB' ler uygulama iş mantığının gelişimini desteklerler.

– **Java Veritabanı Bağlantısı (JDBC):** İlişkisel veri tabanlarına erişim sağlar.

– **Java Mesaj Servisi (JMS):** JSM Java sınıflardan mesajlaşma kullanmak için bir mimari ve API'dir.

– **J2EE Bağlantı Mimari (JCA):** JCA Java programlarının eski ve diğer Java harici sistemlere erişim sağladığı standart bir ara yüz tanımlayan bir çatı ve API'dir.

– **Java İşlem Servisi (OTS):** JTS, Nesne İşlem Servisinin (OTS) Java uygulamasıdır ve dağıtılmış işlemler için bir endüstri standardıdır.

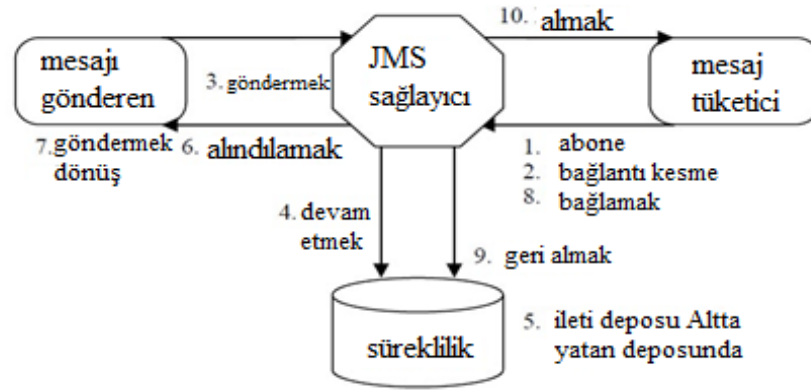
– **Java Posta:** Java sınıflarından e-posta göndermek için bir API sağlar.

2.7.1.1 Java mesaj servisi

Kurumsal düzeyde mesajlaşma sistemleri, çeşitli heterojen bilgisayar ağları ve sistemleri boyunca farklı uygulamalar arasında güvenilir bilgi taşınmasını sağlar.

Bu sistemler, etkileşim programlarının farklı zamanlarda ve gizleme ağ karmaşıklığı ile çalışmasına imkân sağlayan özelliklere sahiptir. **Java Mesaj Servisi (JMS)** standart asenkron mesaj etkileşimini destekleyen Java etkin geliştirme ara yüzüdür ve senkron istek / yanıt iletişim modu simülasyonu yeteneğini sunar.

JMS Java uygulamalarına, MOM sağlayıcı mesajlaşma hizmetlerine standart ve tutarlı bir ara yüz veya mesajlaşma sunucusu sağlar (James et al., 2003). JMS tabanlı uygulamalar, heterojen ortamlarda dağıtılmış eski sistemler ve uygulamalar için bütünleşik yetenekler sağlar. Mimari yapısı mesajların garantili, zamanında teslimat ile sıraya girmesine ve çevrimdışı uygulamaların mesajları alma yeteneğine sahip olduğunda daha sonra işlemlerine olanak sağlar.



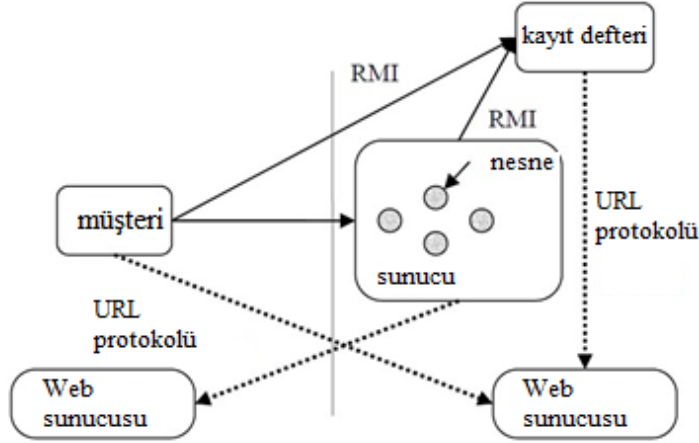
Şekil 2.18 Java Mesaj Hizmetinin Kavramsal Modeli

JMS, bayt-mesajı, nesne-mesajı ve metin-mesajı gibi çeşitli mesaj türlerini destekler. JSM aynı zamanda metin mesajını XML belgesiyle değiştirerek XML mesajlaşması için de kullanılabilir.

2.7.1.2 Uzaktan metot yürütme

Uzaktan Metot Yürütme (RMI) uzak bir nesne üzerinden uzak ara yüze yönelik bir metodun yürütülmesi eylemidir. En önemlisi, uzak bir nesne üzerinden metot yürütme, yerel bir nesne üzerindeki metot yürütme ile aynı sözdizimine sahiptir. RMI istemci ve sunucuların dağıtılmış bir ağında, nesnelerin yönetimini sağlamayla ve her bir makine arasında onların kontrolünü eline geçirme yeteneğiyle metotların yürütülmesine olanak sağlar.

RMI kullanımı kolay yerli Java modeli nedeniyle hafiftir. Küçük boyutlu dağıtılmış nesne tabanlı uygulama geliştirmeyi destekler. Uzak nesne, farklı bir ana bilgisayarda potansiyel olarak bulunan başka bir uygulama tarafından çağrılan metotlara sahiptir. Uzak nesnenin metotlarını belirten ara yüzler, uzak ara yüzler olarak adlandırılır.



Şekil 2.19 RMI Dağıtılmış Uygulama Modeli

RMI, uzaktan yordam çağrısı (RPC) uygulamasına dayanmaktadır. İstemci koçanı, uzak nesne için lokal bir temsilci veya vekil olarak kullanılır. Her uzak nesne, ortak bir ara yüz nesnesinden oluşturulan eşdeğer sunucu temel çatısına sahip olabilir. İstemci koçanı RMI Kaydını bağlamak için kullanılır, oysaki sunucu temel çatısı, uzaktan metot yürütmeyi, güncel uzak nesne uygulanmasına iletir.

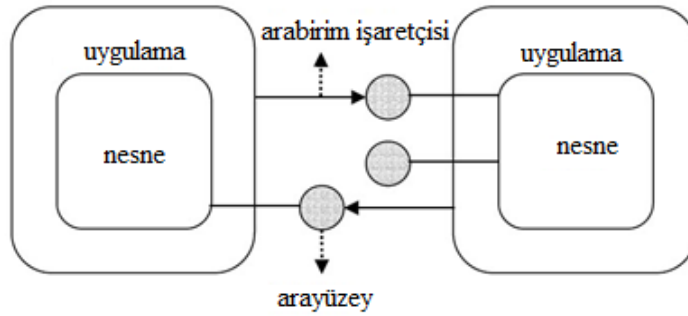
2.7.2 COM / DCOM

Bileşen Nesne Modeli (COM), Microsoft Windows İşletim Sistemleri ailesinin bir parçasıdır ve geliştiriciler tarafından yeniden kullanılabilir yazılım oluşturmak, uygulamaları oluşturmak için bileşenleri birbirine bağlamak ve Windows hizmetlerinden yararlanmak için kullanılır (Microsoft Corporation, 2006). COM, COM +, Dağıtılmış COM (DCOM) ve ActiveX denetimleri de dâhil olmak üzere diğer Microsoft teknolojileri birleşimidir. COM çeşitli uygulamalardan gelen bileşenlerin birleştirilmelerine ve yeni üst düzey yazılım uygulamasında kurulmalarına olanak sağlayan bileşen tabanlı bir yazılım mimarisidir.

COM bileşeni, birlikte çalışabilirliği için bir standart tanımlar. Özellikle programlama dilinden bağımsızdır, çoklu platformlarda kullanılabilir ve genişletilebilir (Sara and Charlie, 1994).

COM nesnesi, belirli bir servisin sağlanması için kullanılan derlenmiş kodların bazı parçaları olarak tanımlanır. Bu nesneler, nesne yönelimli programlamada tanımlanan nesnelerden farklıdır, öyle ki, nesne sistemde kapsamlı bir şartı yerine getirmektedir.

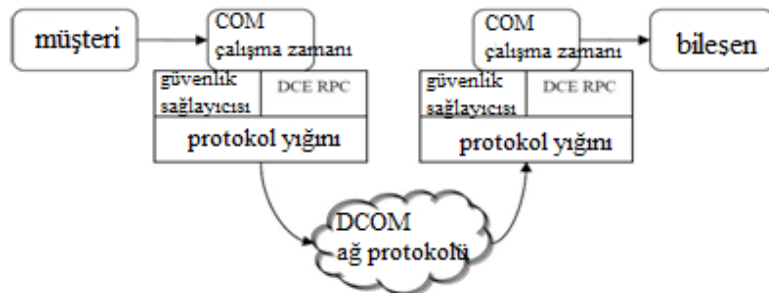
COM bu nesneleri, bileşen nesnesi ya da sadece bir nesne olarak tanımlar.



Şekil 2.20 COM Mimari Modeli

Dağıtılmış Bileşen Nesne Modeli (DCOM), farklı bilgisayarlarda, LAN, WAN ve İnternette ki nesneler arasında iletişimi desteklemek için COM' u genişletir (Microsoft Corporation, 1996). Ağ protokollerine ait alt düzey ayrıntıları ele alarak bileşenlerin dağıtılmış bir ortamda kullanılmalarına olanak sağlar.

Aşağıdaki resimde DCOM mimari örneğidir. Bu model, hep birlikte DCOM kablo protokolü uygulaması için bir arka plan sağlayan COM çalışma zamanı, DCE RPC ve güvenlik sağlayıcıdan oluşmaktadır.



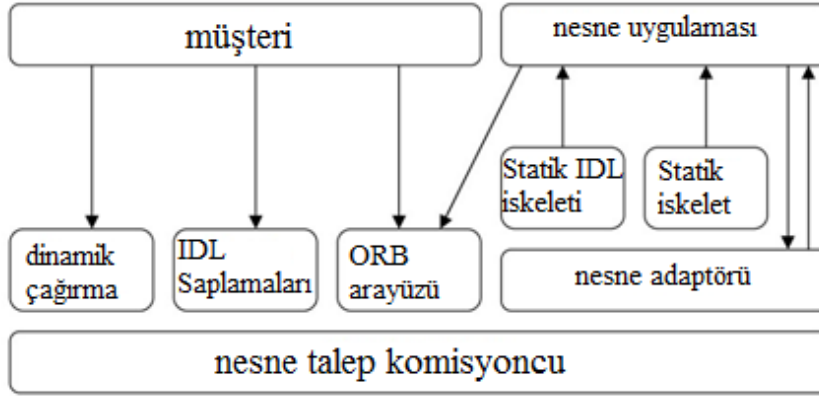
Şekil 2.21 DCOM Mimari Modeli

Mimari açıdan DCOM, platformlar arası geliştirme sağlar. Platformdan bağımsız geliştirme çatılarının bütünleşmesini sağlar ve sanal makine ortamlarının tek bir dağıtılmış uygulama kurmasını mümkün kılar.

2.7.3 CORBA

Genel Nesne İstek Aracı Mimarisi (CORBA) nesnelerin platform ve dil bağımlılıkları olmadan birbirleriyle dağıtılmış bir tarzda iletişim kurabildiği yerlerde çatı sağlayan bir nesne sistemidir. Bir nesne sistemi, servis istemcilerini servis sağlayıcılardan iyi tanımlanmış bir kapsül ara yüzü ile ayrı tutan nesneler topluluğudur (Object Management Group, 2004). Özellikle istemciler, veri temsilleri ve çalıştırılabilir kodlar olarak servis uygulamalarından ayrı tutulur.

Şekil 2.22'de gösterildiği gibi bileşenlerin CORBA mimari yapısında gösterimidir.



Şekil 2.22 CORBA Nesne İsteği Ara Yüzlerinin Yapısı

CORBA mimarisi, istemcinin sunucu nesnelerinden haberdar olmasını ve onları dinamik çağırma yoluyla çalıştırır. İstemciler, sunucu nesnelerini çalıştırmanın yolunu istemciler için tanımlayan IDL taslaklarını ayrıca kullanabilirler. ORB ara yüzleri, CORBA ya özel servisler için API' lerdir. Temel çatılar nesne uygulanması için ortam sağlar. Nesne bağdaştırıcısı, bir nesne ve ORB arasındaki iletişimi yönetir.

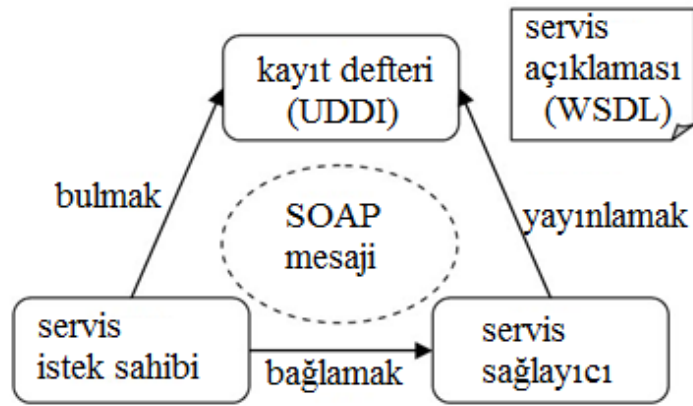
2.7.4 Web servisleri

Web servisleri, farklı uygulamaların iletişimi ve heterojen bilgi işlem ortamlarında birlikte çalışabilirlik için standart bir yöntemi sağlamayı amaçlayan bir dizi XML-tabanlı teknolojidir. Web servisleri, platformdan bağımsız bir ortamda erişilebilir uygulama geliştirmeyi uygun kılan mesajlaşma ve veri

alışverişi için standart internet teknolojilerini kullanır. Web hizmetlerinin resmi tanımını aşağıdaki gibidir (W3C, 2004):

“Web servisi, ağ üzerinden birlikte çalışabilen makineler arası etkileşimi desteklemek için tasarlanmış bir yazılım sistemidir. Makinede işlenebilir formatta tanımlanan bir ara yüze sahiptir (özellikle WSDL). Diğer sistemler Web servisiyle SOAP mesajları kullanarak etkileşime girer ve bu SOAP mesajları açıklamasında öngörülen şekilde diğer Web ile ilgili standartlarla birlikte XML serileştirme ile HTTP kullanarak tipik olarak iletilir.”.

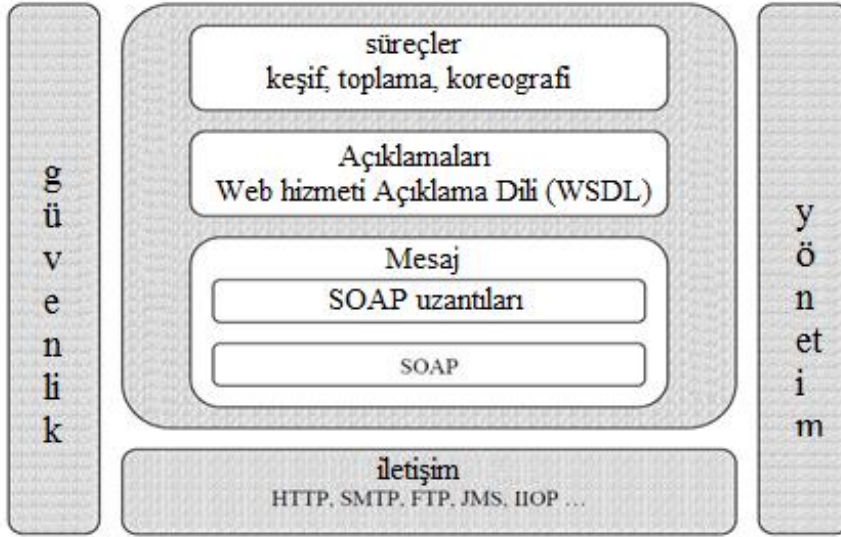
Şekil 2.23'ta gösterildiği gibi Web servisleri mimarisindeki rollerini göstermektedir.



Şekil 2.23 Web Servisleri Mimarisi

Web servisleri mimari modeli, katmanlı teknoloji ailesine dayalıdır. Her bir tabaka birbiriyle ilişki içindedir ve Web servis tabanlı uygulama geliştirme için soyutlama ve işlevsellik düzeyi sağlar.

Aşağıdaki şekilde, bu temel teknolojilerinin bazıları gösterilmektedir.



Şekil 2.24 Web Servisleri Teknolojisi Yığını

XML (Genişletilebilir İşaretleme Dili) (W3C, 2004) Web servislerinin temel dayanağıdır ve veri işleme ve tanımlamaya ilişkin temel bir dildir. Dinamik içerik oluşturma ve yönetimi için XML desteği. Veriyi, sonrasında alt elemanları olabilecek kendini açıklayan elemanlar olarak ifade eder. XML belgelerini oluşturma, tanımlama ve dönüştürmeye yönelik teknolojiler ailesidir.

XML-tabanlı teknolojilerin bazıları şunlardır:

- **XML şema:** Veri türleri, yapılarını ve ilişkili XML belgesinde izin verilen unsurları tanımlayan bir XML belgesidir.
- **XML ad alanları:** XML belgesi elemanları ve uygulamaları için benzersiz nitelikli isimler.
- **Genişletilebilir Stil Sayfası Dili Dönüşümleri (XSLT):** XML belgelerinin diğer XML belge biçimine dönüşümü için kullanılır.

Basit Nesne Erişim Protokolü (SOAP) (W3C, 2003) XML mesajlarının servis tüketicisi ve servis sağlayıcı arasında Web servisleri mimarisinde nasıl okunduğuna ve oluşturulduğuna dair tanımlama özelliğidir. İşletim sisteminden, programlama dilinden veya dağıtılmış bilgi işlem platformundan bağımsız bir mesajlaşma çatısı sağlar. Bir SOAP mesajı üç bölümden oluşur:

- **SOAP zarfı:** XML belgesinin üst ögesidir ve mesajın bir SOAP mesajı olduğunu belirtir.

- **SOAP üstbilgisi:** Mesajın işlenmesinde kullanılan seçmeli özellikleri, uygulama kapsamı bilgilerini ve direktiflerini içerir. Bir ara noktada ya da bitiş noktasında olabilecek mesajın konumunu belirler
- **SOAP gövdesi:** Üretici ve tüketici arasında gönderilen fiili uygulama verilerini ihtiva eder.

2.8 Bulut Bilişim ve Servis Odaklı Hesaplama

Bulut bilişim, farklı kaynakların gerçekleştirim detaylarından soyutlanmış ve ölçeklenebilir bir şekilde kullanıcılara servis olarak sunulması anlamına gelmektedir. Bu şekilde kullanıcılar ihtiyaç duydukları kaynaklara, önceden üzerinde anlaşılan kriterler çerçevesinde belirli bir ücret karşılığında erişebilmektedir. Bulut bilişimde genellikle 3 tür servis bulunmaktadır: servis olarak yazılım (Software as a Service-SaaS), servis olarak platform (Platform as a Service-PaaS), servis olarak altyapı (Infrastructure as a Service-IaaS).

2.8.1 Servis olarak yazılım

Bir hizmet (veya SaaS) gibi yazılım uygulamaları bir satıcı veya servis sağlayıcı tarafından barındırılan ve tipik bir ağ üzerinden internet'e ince istemci kullanan kullanıcılar için erişilebilir yapılan bir yazılım dağıtım modelidir. Bunun yerine yükleme ve yazılım sürdürmenin, son kullanıcıların sadece karmaşık yazılım ve donanım yönetimi kullanıcıların kurtararak, internet üzerinden yazılımı erişim. SaaS uygulamaları on-demand olarak adlandırılan yazılım veya yazılım barındırılan. Onun uygulamalar bir SaaS sağlayıcının sunucularında çalışır. Sağlayıcı güvenlik, kullanılabilirlik ve performans dâhil olmak üzere uygulamalara erişimi yönetir.

2.8.2 Servis olarak platform

PaaS müşteri platformu mevcut uygulamaları çalıştırmak için sanallaştırılmış sunucuları ve ilgili hizmetleri kiralamak için izin veren bir dağıtım hizmeti modeli için bir paradigmadır. Genellikle, işletim sistemleri ve ilgili servisler indirme veya kurulum yapmadan internet üzerinden uzak istemciler olarak görev yapmaktadır. PaaS'ın asıl hedefi, dağıtım platformu içinde bir

entegre geliştirme ortamı için tüm hizmet seçeneklerini sağlamaktır. Böylece yüksek ölçeklenebilirte ve güçlü Web-tabanlı uygulamaları inşa etmek için temel sağlamış olurlar, örneğin, Amazon ve Google servisleri.

2.8.3 Servis olarak altyapı

Hizmet olarak Altyapı bilgisayar altyapısı, tipik bir sanallaştırılmış bir platform ortamı, bir hizmet olarak sunulduğu bir modeldir. Bir hizmet sağlayıcı depolama, donanım, sunucu ve ağ bileşenleri dâhil olmak üzere operasyonlarını desteklemek için kullanılan altyapı sağlar. Servis sağlayıcı ekipman sahibidir ve ekipmanı korumaktan sorumludur. İstemci böyle bir hizmeti (IaaS) kullanıcısı konumundadır ve sunulan altyapı aracılığıyla depolama, bilgisayar ve ağ kapasitesi gibi sanal BT kaynaklarını kontrol edebilmektedir.

3. MOBİL HESAPLAMA

3.1 Giriş

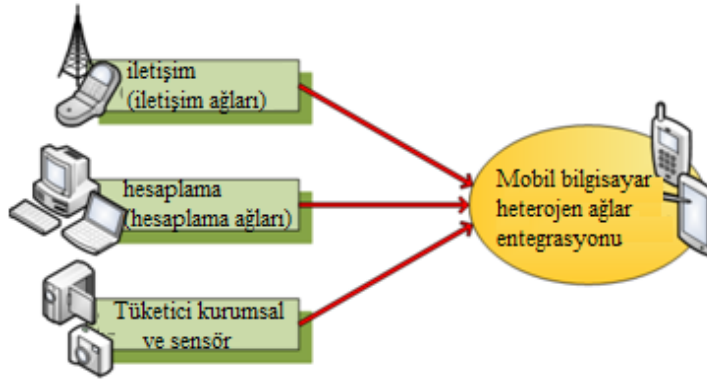
Mobil hesaplama sistemleri fiziksel olarak kolayca taşınabilirler ve bilgi işlem yetenekleri taşınırlarken kullanılabilir. Örnekleri dizüstü bilgisayarlar, kişisel dijital asistanlar ve cep telefonlarıdır. Mobil bilgi işlem sistemleri, gerçekleştirmeleri için tasarlandıkları görevlerde ve tasarlanış ve çalışma biçimlerinde diğer bilgi işlem sistemlerinden farklılık gösterirler. Mobil bilgi işlem sistemlerinin ayırt edici yönleri arasında yaygın kablosuz ağ bağlantılarını, küçük boyutlarını, kullarımlarına ilişkin mobil niteliklerini, güç kaynaklarını ve özellikle mobil kullanıcılara uygun düşen işlevlerini hatırlarız. Bu yönleri nedeniyle mobil bilgi işlem uygulamaları, sabit bilgi işlem sistemlerinde kullanım için yazılmış uygulamalardan kendilerine has şekilde farklıdır. Bugün bilgi işlemleri ağ bağlantısı olmadan düşünmek zordur; ağ oluşturma ve dağıtılmış bilgi işlem, bilgi işleme yönelik mevcut çabaların odak noktası olan en büyük bölümlerden ikisidir. Kablosuz iletişim sistemleri, ağ bağlantısını kolaylaştırmak için genellikle mobil bilgi işlem sistemlerde kullanılır, fakat mutlaka hareketliliği ifade etmezler. Çünkü bazı kablosuz teknolojiler; bir ya da her iki taraf, kısa menzilli Bluetooth veya kızılötesi veya uzun menzilli wi-max gibi kablosuz sinyal iletimi sırasında sabit olduğunda ve zorlukla hareket ettiğinde ancak kullanılabilirler. Kablosuz radyo katmanına ve ağ protokol yığınının bağlantı katmanına önem verir, oysaki mobil bilgi işlem uygulamaları ve hizmetleri oluşturmak için her yerde bulunan kablosuz iletişimden yararlanır. İletişimin geleceği doğal olarak bilgi işlemin yakınsamasını ihtiva eder ve iletişim iletişimin neredeyse her bir ögesindedir, böylece *herhangi bir cihazdan herhangi bir yer ve zamanda* bilgi erişimine olanak sağlamaktadır. Bilgilerin sadece herhangi bir yerden ve zamanda erişilebilir olma zorunluluğu değil aynı zamanda son derece merkezi olmayan, dağıtılmış bir bilgi altyapısında saklanması da gereklidir. Bilgi sunucularının geniş bir yelpazesinin (hem kamu hem de özel) mobil cihazlar için erişilebilir olması gerekir. Bu tür cihazların kendileri veri içerebilir veya veriler taşınabilir bellek akıllı-kartlarında saklanabilir. Bu vizyon, güvenilir ve yüksek performanslı bilgi işleme yönelik olarak sınırlı bilgi işlem kaynakları aracılığıyla heterojen kesintisiz kablosuz erişimine sahip evrensel bir mobil platform oluşturma sorununa yol açar.

Açıkçası herhangi bir mobil bilgi işlem sistemi sabit de olabilir; bu nedenle sabit bilgi işlem altkütmesi dışındaki bu unsurlara bakmak önemlidir.

(Reza, 2005) de, *hareketlilik boyutları*, yazılım uygulamaları ve mobil bilgi işlem sistemlerini oluşturmaya yönelik problemlerin üstesinden gelmeye olanak sağlayan araçlar olarak tanımlanır. Bu boyutların birbirine göre tam olarak dik açılı olmamasına rağmen, dikey değişkenler için ayırt ve tahmin edilmek üzere doğaları gereği yeterince ayırırlar.

Üstelik bu boyutların bazıları, tipik sabit bir uygulama ile uğraşırken genellikle dikkate alınmayan faktörleri sınırlamaktadır. Hareketliliğin bu boyutları şunlardır:

- Konum farkındalığı
- Network servis bağlantısı kalitesi
- Sınırlı Cihaz yetenekleri (depolama ve CPU).



Şekil 3.1 Dijital yakınsama

- Sınırlı güç kaynağı
- Kullanıcı ara yüzlerinin geniş yelpazesine destek
- Platform yayılması.
- Aktif işlemler

Bir mobil cihaz her zaman aynı yerde değildir, konumu sürekli değişiyor olabilir. Hareketlilik tarafından sunulan zorluklar ve fırsatlar iki kategoriye ayrılabilir: *yerelleştirme* ve *konum hassasiyeti*. Yerelleştirme, genellikle *lokal* olarak anılan ve konum bilgisi verilen bir dizi temelinde iş akışının farklı düzeyde seçimini ve iş mantığı ve ara yüzlerini sağlayan mobil uygulama mimarisinin

mantığını sadece tanıtmaya yeteneğidir. Yerleştirme mobil uygulamalara özel değildir ancak mobil uygulamalarda çok daha önemli rol alır. Konum hassasiyeti, mobil uygulamalar için oldukça özel bir şeydir.

Kablolu veya kablosuz bağlantı kullanılsın veya kullanılsın, hareketlilik ağ bağlantısı güvenilirlik kaybı anlamına gelir. Kablosuz ağ bağlantısı durumunda, fiziki koşullar *hizmet kalitesini* (QoS) önemli ölçüde etkileyebilir. QoS araçları ve ürünleri, ağa bağlanabilirliğin güvenilirliğini veya güvenilmezliğini ölçmek ve nitelendirmek için tipik olarak uygulanır ve daha çok şebeke operatörleri tarafından kullanılır. Ağ operatörleri ağın fiziksel katmanını kontrol ederler ve yazılım uygulama bağlantısı için internet protokolü (IP) gibi imkânları sağlarlar. Genellikle şebeke operatörleri tarafından işletilen QoS araçları, yazılım uygulamalarının akıllı bilgi işlem kararları almalarına olanak sağlamak için mevcut bant genişliği, bağlantı kaybı riski ve istatistiksel ölçümler gibi bilgileri sağlarlar. Sabit yazılım uygulamaları, yoğunlukla bağlı olunma veya bağlantısının kopmasıyla sınırlı bağlantının bazı ayrı modlarını tipik olarak kabul ederler.

Bu çoğu uygulamalarda işe yarar çünkü kablolu ağ bağlantılarının yoğunluğu oldukça güvenilirdir. Bunun tersine, mobil uygulamalar ağ bağlantıları kesildiğinde veya kesintili ve sık sık ağa bağlanmaları veya kopmaları halinde dahi nasıl çalışmaya devam edeceklerini bilmek durumundadırlar. Neredeyse tüm mobil uygulamalar, uygulama ağ bağlantısından bir anda koptuğunda çalışmayı nasıl durduracaklarını bilmelidirler ve akabinde tekrar bağlandığında çalışmaya devam etmelidirler. Örneğin, Mevcut gerçek zamanlı bant genişliği, bazı zaman aralıklarında sağlanan ve yenilenen verinin bir parçası olmalıdır. Bu tür veriler, özelliklerini ve işlevselliklerini mevcut bant genişliğine dinamik bir şekilde uyarlayan uygulamaları tasarlamak için kullanılabilir.

Başka bir sorun *cihazların boyut* ile ilgilidir. Mobil cihazlar küçüktür ve bu fiziksel büyüklük sınırlaması, bu cihazlara yönelik uçucu bellek, uçucu olmayan bellek ve CPU sınırlamalarını dayatır. Daha küçük cihazlar taşımak daha kolaydır ve sonuç olarak, daha yaygın olabilirler. Bu yaygınlaşma ayrıca büyük ölçüde cihazların fiyatına bağlıdır. Elektronik cihazları çok küçük yapma maliyeti

normal olarak arttırır çünkü cihazları daha küçük yapmaya giden araştırma ve geliştirme çok pahalıdır.

Ancak teknoloji olgunlaştığında ve bun yapmaya yönelik süreçlerin üretimi çoğunlukla otomatikleştirildiğinde, fiyatları azalmaya başlar. Cihazların daha çok bir emtia olduğu noktada, daha küçük aynı zamanda daha az pahalı anlamına gelir. Mobil uygulamaların veri depolama kullanımını ve aygıtın işlemci gücünü kullanıcı tarafından uygulama kullanımı açısından iyi duruma getirmek için tasarlanmış olması gerekir. Bazı cihazların işletim sistemi, kullanılabilir depolama alanı sunabilir ancak bu garanti edilmemektedir. Bu yüzden, cihazın donanım özellikleri hakkında en az varsayımla veya mobil uygulaması ile desteklenecek tüm cihazlar için geçerli tüm bu varsayımlarla birlikte tasarlamamız gerekir. Mühendislik mobil uygulamalarının büyük bir kısmı, öncelikle çeşitli platform türlerinin teorik bir anlayışını ve mobil cihazlarda kullanılabilir işletim sistemlerini ve daha sonra çeşitli işletim sistem ve platform türlerinin mevcut ticari uygulamalarının bir anlayışını ve her bir platform cihaz kombinasyonu için en uygun uygulamaların türünü gerektirir.

Mobil bilgi işlem insan-bilgisayar ara yüzlerine (HCIs) başka bir düzeyde sorun getirir. Çünkü uygulamalar ve servisler ağırlıklı olarak hareket halindeyken kullanılır, kullanıcı ve mobil terminal ve kullanıcı sunucu uygulama hizmet sistemi arasındaki etkileşimin yeterince basit, kullanışlı, sezgisel, esnek ve verimli olması gerekir. Buna ek olarak, ara yüzü güç tasarrufu için tasarlanmış olması gerekir.

Sabit kullanıcılar, PC ya da benzer bir cihazda çalışırken mobil olmayan uygulamaları kullanırlar. Klavye, fare ve monitörün bu tür uygulamalar için oldukça verimli bir kullanıcı ara yüzleri olduğu kanıtlanmıştır. Bu tüm mobil uygulamalar için geçerli değildir; Bazı alternatif ara yüzlerin örnekleri, genellikle birleşik şekilde kullanılan sesli kullanıcı ara yüzleri, daha küçük ekranlar, stilus ve diğer işaretleme aygıtları, dokunmatik ekranlar ve minyatür klavyelerdir. Mobil uygulamalar tasarımcıları ve uygulayıcılarının geçmesi gereken en büyük paradigma kayması; uygulamaya yönelik en iyi kullanıcı ara yüzlerini bulma, uygun kullanıcı ara yüzlerini uymak üzere sistemi kurma ve bunları uygulama ve yeni kullanıcı ara yüzünün her zaman gerekli

olabileceğini unutmama gerekliliğini anlamaktır. HCI çerçevesinde anahtar konular şunlardır:

Bir mobil cihaz üzerinde kullanıcı ara yüzü tasarımı. Boyut, renk derinliği, ekran çözünürlüğü, pil süresi, ağırlık, bağlantı, vb gibi araya eklenmiş bir dizi faktör, mobil uygulamaya yönelik kullanıcı ara yüzü tasarımını etkiler.

- *Birden fazla mobil cihazlar üzerinden tutarlı kullanıcı ara yüzü tasarımı.* Girdi ve çıktı cihazları standart gösterim tabakasının olmaması nedeniyle, kullanıcı ara yüzü hedefleme düzeneğinin her türüne tasarlanmak ve uygulanmak durumundadır. GUI bileşenleri ve operasyon mantığı düzeni, kolay gezinme ve sezgisel işlemlere olanak sağlamak için tutarlı olmalıdır.
- *Uyarlanabilir kullanıcı ara yüzü tasarımı.* Bir mobil cihaz üzerinde ki uygulama ve destek sistemi yazılımının, kullanıcı ara yüzünü cihazda dinamik olarak mobil cihazın donanım yapılandırması ve çalışan uygulama bağlamı temelinde iyi duruma getirmesi mümkün olmalıdır.

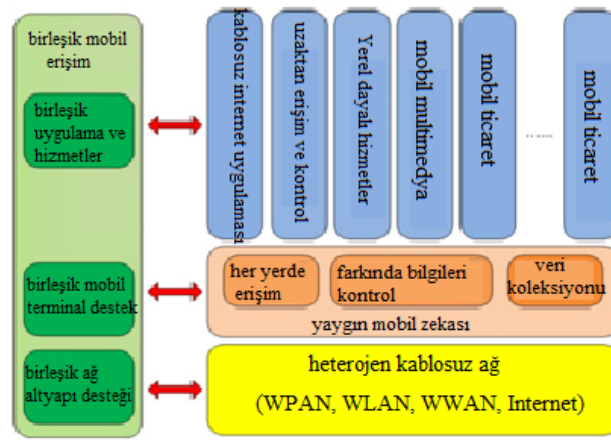
Platform yayılmasının mobil uygulamaların mimarisi, tasarımı ve geliştirmesi üzerinde çok önemli etkileri vardır. Platform yayılması, platformdan bağımsız cihazların tasarım ve geliştirmesinin önemini artırır. Mobil cihaza özgü yerel kod yazılımı, performans gereksinimlerinden ötürü kesinlikle gerekli olmadıkça, cihazların çoğalmasından ötürü önerilen bir uygulama değildir. Platform yapıcılar ve cihazların üreticileri ve bu cihazların işletim sistemleri, platformdan bağımsız uygulamaların yazılımını yasaklamak için geliştiricilere her zaman kısıtlamalar oluşturmayı deneyecektir. Ticari platform yapıcılarının çabalarına bakılmaksızın yazılım mimarları ve geliştiricileri, kullanıcı ihtiyaçlarını karşılamaya yönelik birincil görevlerine odaklanmalıdırlar. Ve bu gereksinimler, mobil bilgi işlem sistemlerinde olduğundan daha sık meydana gelmeyen birden fazla platform desteğini ihtiva etmezse, platform bağımsızlığı uygulama oluşturmak için araçlarını seçerken mimarlar ve geliştiricilerin listesinin tepesinde olmalıdır.

Lionel'a and Zheng'a (2006) göre mobil bilgi işlem vizyonu, *mobil erişim yakınsaması* ve *mobil istihbarat yaygınlaşması* Şekil 3.2'de gösterildiği gibi özetlenebilir. Sistem bileşenleri ve ağ elemanlarının, kablosuz teknolojilerin bir karışımını kullanarak, birlikte çalışabilir ve koordine edilmesi gerektiği gerçeğine

yönelik ilk adresler, sistematik olarak bütünleşik bir dizi donanım ve yazılım bileşenleriyle uygulamaları ve hizmetleri sağlayabilen mobil terminallerin kullanımına sebep olmaktadır.

(Reza, 2005)'de açıklandığı üzere, mobil ve sabit kullanıcı arasındaki fark hareketli bir durumda ifade edilebilir; mobil kullanıcı sabit kullanıcıdan temelde aşağıdaki şekillerde farklıdır:

1. Mobil kullanıcı, bilinen veya bilinmeyen konumlar arasında azından ara sıra hareket etmektedir. Belirli bir zamanda kullanıcının konumu değişkendir. Diğer değişkenler şunlar olabilir: mobil kullanıcının seyahat edebildiği hız,



Şekil 3.2 Mobil Bilgi İşlemin Vizyonu

Hangi ağların kullanıcı için hareketsizlik modlarında mevcut olduğu, ilgili bağlanabilirliğin herhangi bir yer ve zamanda ne kalitede olduğu veya ne kadar süreyle bağlantıda olup olmayacağı. Mobil kullanıcılar sistemden ayrıca iyi bir bağlantı kapsama alanına sahip olmasını beklerler. Mobil kullanıcının değişen konumu; güç, cihazın boyutu, cihazın kabloşuz bağlanabilirliği ve mobil kullanıcının durumunun neredeyse her yönü üzerindeki kısıtlamaları da zorlaştırır.

2. Mobil kullanıcı tipik olarak bilgi işlem görevlerine odaklı değildir. Mobil kullanıcılar genellikle mobildir çünkü hedefe ulaşmanın birincil göreviyle iki nokta arasında hareket ederler. Odaklanma eksikliğinin bir diğer nedeni çoklu görevlendirmedir, mobil kullanıcı doğasındaki bu tür çoklu görevlendirmeden ötürü, ses gibi çeşitli kullanıcı ara yüzü girdi türleri, bir başka görevi ilgilendirmeyen duyulardan yararlanmak için gerekli olabilir. Ayrıca sisteme yönelik kullanıcı ara yüzü çok kullanıcı dostu olmalıdır ve makineyle mümkün

olduğunca verimli bir şekilde iletişim odaklamaya yönelik kullanıcı duyularının bir kaçını gerektirmelidir. Örneğin, sesli kullanıcı ara yüzleri, sistemden hali hazırda her türlü bilgiyi alırken sürüşe odaklanmalarına olanak sağlar.

3. Mobil kullanıcı, sistemden sıklıkla yüksek derecede yakınlık ve duyarlılığa ihtiyaç duyar. Mobil kullanıcılar, sıklıkla iletişim bilgilerini almak, es veya e-posta iletisi göndermek veya bazı uzak süreçleri tetiklemek gibi bir veya daha fazla bilgi işlem görevini hızlı bir şekilde gerçekleştirmek için gerek duydukları durumdadırlar. Uzun önyükleme dizisinden veya uzun uygulama kurulumu sürelerinden geçecek vakitleri yoktur. Mobil kullanıcıların cihazlarından sabit kullanıcılara göre normalde daha yüksek performans beklentileri vardır. Uygulama yanıtlanmasında kısa bir gecikme, yararlılığını son derece azaltabilir.
4. Mobil kullanıcı görevleri sıklıkla ve/veya aniden değişir. Mobil kullanıcının bazı bilgi işlem görevlerini gerçekleştirmeyi aniden durdurması, tamamen ilgisiz olabilecek bir şey yapması ve daha sonra belirsiz bir süre sonrada uygulamaya geri dönmesi mümkün olmalıdır. Uygulamada ne yapmakta olduğunu hatırlamak için çok çaba sarf etmemelidir ve bilgi işlem görevine devam etmelidir. Mobil kullanıcılar sorunsuz akış gösteren uygulamaları beklerler ve eylemlerinin ani yapısına rağmen karmaşık gezintileri istemezler.
5. Mobil kullanıcı veri almanın ve herhangi bir an ve zamanda bilgi işlem yapmanın mümkün olmasını bekler. Çeşitli kullanıcı ara yüzleri içeren çeşitli platforma ilişkin desteklerin mobil uygulama için kritik öneme sahip olmasının tam olarak nedeni budur: Her yerde ve her zaman bir uygulamayı kullanmak için, Birisi, her ne cihaz mevcutsa ve belirtilen yer ve zaman için uygunsa bunu kullanmak zorunda kalabilir. Mobil kullanıcılar bir cihazda bir eylemi başlatmayı ve bitirmeden bırakmayı düşünebilir ve aynı eylemi farklı bir cihazda farklı bir yer ve zamanda bitirir.

Dijital yakınsama nedeniyle mobil sektörü bu yıllarda önemli bir bozulmayla karşı karşıyadır. Çok fonksiyonlu ürünler tüketiciler için ortaya çıkmaktadır ve esneklik, ölçeklenebilirlik ve modülerlik gibi çeşitlendirmeler mimarlar ve platformlar için biz dizi yeni gereksinimler ortaya koymaktadır. Hareketlilik kurumsal iş stratejisinin bir bileşeni olarak kabul edilir ve mobil uygulamaları dağıtmak büyük verimlilik iyileştirmeleri sağlar. Hareketlilik karmaşıktır çünkü bazısı eski ve bazısı yeni dağıtılmış ve mobil işletim

sistemlerinin (BlackBerry OS, Windows Mobile, Symbian OS, Mac OS X, Palm OS, Android ve mobil Linux) giderek artan sayısına sahip bir dizi mobil cihazlar gibi çoklu arka uç sistemleri içerir. CDMA ve GSM standartlarına sahip mevcut hücresel ağlardan Wi-Fi, Wi-Max, ve gelecek nesil 4G ağlara kadar kablosuz teknolojilerin büyük bir çeşitliliği küresel pazarda mevcuttur.

3.2 Mobil Donanımlar

Küçük bir bilgisayar olarak mobil cihaz, bütünleştirilmiş veya birbiriyle bağlantılı donanım bileşenleri ve yazılımlardan oluşur. Bu donanım bileşenleri; mikroişlemci, salt okunur bellek (**ROM**), rastgele erişim belleği (**RAM**), genişletilmiş hafıza, ağ ara yüzleri ve anten, pil ve ekrandan oluşur. Bazı mobil cihazlar da sabit diske sahiptir.

Mobil İşlemciler

Mobil cihaz; küçük, gömülü bilgisayar sistemi tarafından kontrol edilir, gömülü sistem kavramı bir dizi sabit fonksiyonu gerçekleştiren özel bir bilgisayar sistemi anlamına gelir. Mobil cihazlar; cep telefonları, PDA'lar, dizüstü bilgisayarlar, çağrı cihazları ve kablosuz özellikli taşınabilir oyun cihazları gibi gömülü sistemlerin büyük bir bölümünü oluşturmaktadır. Mobil / taşınabilir cihazlarda kullanılan mikroişlemciler (CPU'lar), gömülü işlemci türü olan genelde mobil işlemciler olarak adlandırılır. Mobil işlemcilerin ana özellikleri şunlardır (Koushanfar et al., 2008):

- *Sınırlı programlanabilirlik* - Mobil bir işlemci, güç tüketimi kısıtlamaları nedeniyle bir masaüstü bilgisayara kıyasla sınırlı bilgi işlem yeteneği temin edebilir.
- *Hesaplama oranı için yüksek I / O* - Ağ ve I / O haberleşmesi masaüstü bilgisayarlara göre mobil terminallerde daha siktir.
- *Akış ve veri işleme* - Çoklu ortam işleme pek çok mobil terminallerde vazgeçilmezdir.
- Birkaç mobil işlemci ailesi, mobil pazarda büyük oyuncu olmak için ortaya çıkmıştır.

Gelişmiş Azaltılmış Komut Kümeli Bilgisayar (RISC) Makinesi (ARM)

başlangıçta Acron adında bir İngiliz firması tarafından geliştirilmiş mikroişlemci mimarisidir. Acron' un yan ürünü ARM şirketi, ARM teknolojileri lisansını yarı iletken üreticilere ve elektronik cihaz üreticilerine verir.

NVIDIA'nın Tegra' sı, ST-Ericsson Nomadik, Qualcomm'in Snapdragon' u, ve Texas Instruments OMAP ürün grupları 1.1, 1.2, 1.3, 1.4 tablolarında özetlenmiştir.

ARM tabanlı makinelerde çevresel aygıtlar, fiziksel kayıtlarını ARM bellek ya da ya da yardımcı işlemci alanında eşleştirerek ya da sırasıyla işlemciye takılan başka bir cihaza (veri yolu) bağlanarak genel olarak işlemciye bağlıdır. Yardımcı işlemci erişimlerinin daha düşük gecikme süresi vardır, bu nedenle bazı çevre birimleri (örneğin XScale kesme denetleyicisi) her iki yönden (bellek yoluyla ve yardımcı işlemci yoluyla) erişilebilir olması için tasarlanmıştır. Diğer durumlarda, çip tasarımcıları yardımcı işlemci mekanizmasını kullanarak sadece donanımları bütünleştirir.

Derlenmiş kod yoğunluğunu artırmak için, 'Thumb' modu tanıtıldı. Bu moddayken işlemci 16-bit talimatlarını yürütür. Bu 16-bit çapındaki Thumb talimatların çoğu, doğrudan normal ARM talimatlarıyla eşleştirilir. Yerden tasarruf, işlenen talimatların bazısını örtülü hale getirerek ve tam ARM modu talimatına göre olasılıkların sayısını sınırlayarak ortaya çıkmaktadır.

Thumb-2, ilave 32-bit talimatlarına sahip Thumb' in sınırlı 16-bit komut setine daha geniş talimat kümesi vermek için uzanır (yani bit-alanı manipülasyonunun tanıtılması, masa dalları ve koşullu yürütme). Thumb-2 için belirtilen amaç, 32-bit bellekte ayarlanan ARM talimatına benzer performans ile Thumb ' a benzer kod yoğunluğu elde etmektir. ARMv7' de bu hedefin karşılanmış olduğu söylenebilir. Mimari zamanla gelişmiştir ve V7 mimarisi ile başlayan üç "profil" tanımlanmaktadır: "A" (başvuru), "R" (gerçek zamanlı), ve "M" (mikro denetleyici) profiller. Dijital sinyal işleme ve çoklu ortam uygulamaları için ARM mimarisini geliştirmek üzere, birkaç yeni kullanım talimatı ayarlamalara ilave edilmiştir. Bunların; imzalı çarpın-biriktirme, doymuş eklenti ve çıkarma ve başlangıçtaki sıfırların sayısında çeşitlemeleri vardır.

Tablo 3.1 ARM işlemci aileleri ve uygulamaları bölüm 1

Family	Architecture Version	Core	Features	Cache (I/D)/MMU	Typical MIPS@MHz	Application
Xscale	Armv5TE	PXA26x			default 400MHz, up to 624MHz	Palm Tungsten T3
Xscale	Armv5TE	PXA27x	Applications processor	32KB/32KB, MMU	800MIPS@624 MHz	Gumstix verdex, "Trizeps-Modules" PXA270 COM, HTC Universal, HP hx4700, Zaurus SL-C1000, 3000, 3100, 3200, Dell Axim x30, x50, and x51 series, Motorola Q, Balloon3, Trolltech Greenphone, Palm TX, Motorola Ezx Platform A728, A780, A910, A1200, E680, E680i, E680g, E690, E895, Rokr E2, Rokr E6, Fujitsu Siemens LOOX N560, Toshiba Portégé G500
Xscale	Armv5TE	Monahans		32KB/32KB L1, TCM, MMU	1000MIPS@1.25GHz	Samsung Omnia
Xscale	Armv5TE	PXA900				Blackberry 8700, Blackberry Pearl (8100)

Tablo 3.2 ARM işlemci aileleri ve uygulamaları bölüm 2

Family	Architecture Version	Core	Features	Cache (I/D)/MMU	Typical MIPS @ MHz	Application
ARM11	ARMv6	ARM1136J (F)-S	8-stage pipeline, SIMD, Thumb, Jazelle DBX, (VFP), Enhanced DSP instructions	variable, MMU	740@532-665 MHz (i.MX31 SoC), 400-528MHz	Texas Instruments OMAP2420 (Nokia E90, Nokia N93, Nokia N95, Nokia N82, Nokia N800, Nokia N810, Qualcomm MSM7200 (with integrated ARM926EJ-S Coprocessor@274 MHz, used in Eten Glofish, HTC TyTN II, HTC Nike), Freescale MXC300-30 (Nokia E63, Nokia E71, Nokia 5800, Nokia E51, Nokia E75, Nokia N97, Nokia N81), Qualcomm MSM7201A as seen in the HTC Dream, HTC Magic, Motorola Z6, HTC Hero
ARM11	ARMv6KZ	ARM1176JZ (F)-S	8-stage pipeline, SIMD, Thumb, Jazelle DBX, (VFP), Enhanced DSP instructions	variable, MMU + TrustZone		Apple iPhone, Apple iPod touch, Conexant CX2427X, Motorola RIZR Z8, Motorola RIZR Z10, NVIDIA GoForce 6100[19]; Telechips TCC9101, TCC9201, TCC8900, Fujitsu MB86H60, Samsung S3C6410, S3C6430

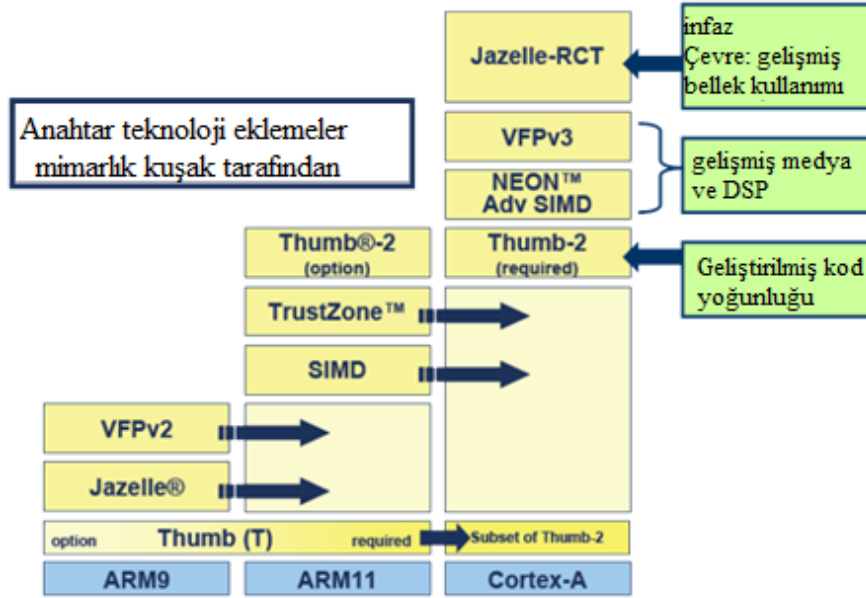
Tablo 3.3 ARM işlemci aileleri ve uygulamaları bölüm 3

Family	Architecture Version	Core	Features	Cache (I/D)/MMU	Typical MIPS @ MHz	Application
ARM11	ARMv6	ARM1136J (F)-S	8-stage pipeline, SIMD, Thumb, Jazelle DBX, (VFP), Enhanced DSP instructions	variable, MMU	740@532-665 MHz (i.MX31 SoC), 400-528MHz	Texas Instruments OMAP2420 (Nokia E90, Nokia N93, Nokia N95, Nokia N82, Nokia N800, Nokia N810, Qualcomm MSM7200 (with integrated ARM926EJ-S Coprocessor@274 MHz, used in Eten Glofish, HTC TyTN II, HTC Nike), Freescale MXC300-30 (Nokia E63, Nokia E71, Nokia 5800, Nokia E51, Nokia E75, Nokia N97, Nokia N81), Qualcomm MSM7201A as seen in the HTC Dream, HTC Magic, Motorola Z6, HTC Hero
ARM11	ARMv6KZ	ARM1176JZ (F)-S	8-stage pipeline, SIMD, Thumb, Jazelle DBX, (VFP), Enhanced DSP instructions	variable, MMU + TrustZone		Apple iPhone, Apple iPod touch, Conexant CX2427X, Motorola RIZR Z8, Motorola RIZR Z10, NVIDIA GoForce 6100[19]; Telechips TCC9101, TCC9201, TCC8900, Fujitsu MB86H60, Samsung S3C6410, S3C6430

Tablo 3.4 ARM işlemci aileleri ve uygulamaları bölüm 4

Family	Architecture Version	Core	Features	Cache (I/D)/MMU	Typical MIPS @ MHz	Application
Cortex	ARMv7-A	Cortex-A8	VFP, NEON, Jazelle RCT, Thumb-2, 13-stage superscalar pipeline	variable (L1+L2), MMU + TrustZone	up to 2000 (2.0 DMIPS/MHz in speed from 600MHz to greater than 1GHz)	Texas Instruments OMAP3xxx series, SBM7000, Oregon State University OS- WALD, Gumstix Overo Earth, Pandora, Apple iPod touch (3rd Generation), Ar- chos 5, FreeScale i.MX51-SOC, Bea- gleBoard, Apple iPhone 3GS, Motorola Droid, Palm Pre, Samsung i8910, Sony Er- icsson Satio, Touch Book, Nokia N900, ZiiLABS ZMS-08 system on a chip
Cortex	ARMv7-A	Cortex-A9 MPCore	Application profile, (VFP), (NEON), Jazelle RCT and DBX, Thumb-2, Out-of-order speculative issue super scalar	, 1-4 core SMP MMU + TrustZone	2.5 DMIPS/MHz (per core)	Texas Instruments OMAP4430/4440, ST- Ericsson U8500

XScale mimarisi, kayan nokta talimatları olmadan ARMv5TE ISA' e dayanmaktadır. XScale yedi aşamalı tamsayı ve sekiz aşamalı hafıza süper ardışık düzen RISC kullanır. Intel' in DEC's Digital' den aldığı mikroişlemci ve mikro denetleyicilerin Intel StrongARM hattının devamı niteliğindedir.



Şekil 3.3 Cortex için ARM işlemci evrimi

İki şirket arasındaki davanın yan etkisi olarak yarıiletken bölümü. Talimatlar reklam veri bellek yönetim birimleri, veri önbelleklerini ve tamponları, güç yönetimi, performans izleme, hata ayıklama ve JTAGS birimleri, ara yüzleri ve çekirdek bellek veri yolu ile birlikte ARM uyumlu yürütme çekirdeğinden oluşur. XScale mimarisi, fiçı tipi çalışma gerilimi ve frekans ölçeklemeye olanak sağlayan Intel dinamik voltaj yönetimi teknolojisini ön plana çıkarır.

XScale mikroişlemciler, popüler RIM Black-Berry avuç içi, Cep Bilgisayarların Dell Axim ailesi, Zire' nin birçoğu, Palm tarafından Treo ve Tungsten avuç içi hatları, Sharp Zaurus' un sonraki sürümleri, Motorola A780, Acer n50, Compaq iPad 3900 serisi ve diğer birçok PDA ' ler gibi ürünlerinde bulunabilir. XScale ayrıca PVP'ler (Taşınabilir Video Oynatıcılar), Yaratıcı Zen Taşınabilir Ortam Oynatıcıları ve Amazon Kindle e-kitap okuyucusu dâhil PMC'ler (Taşınabilir Medya Merkezleri) ve endüstriyel gömülü sistemler gibi cihazlarda kullanılır.

Bellek ve Depolama

Bellek, mobil cihazlar için kısıtlamaların başka bir boyutunu gösterir çünkü hem mobil işletim sistemi hem de uygulamalar küçük bir program kullanım alanı gerektirir. Bu etki alanında üç çeşit bellek kullanılır: RAM, ROM ve flaş bellek. İkincisi, diğer ROM tiplerinin bayt seviyesine karşı verilerin blok düzeyinde okunmasına veya silinmesine olanak veren geçici olmayan elektriksel olarak silinebilir programlanabilir salt okunur bellek (EEPROM) için özeldir. Masaüstü bilgisayarlarda fiziksel bellek, birinci seviye depolama olarak kabul edilir ve sabit diskler (örneğin sabit disk) ikincil depolamadır.

Mobil bilgisayar sistemleri için, sistem kod büyüklüğünden dolayı ve uygulama kodlarının, masaüstü bilgisayar sistemlerindekiyle nazaran daha küçük olması nedeniyle; saf birinci düzey bellek tasarımı iyi performans elde etmek için makul maliyetle uygulanabilir. Sonuç olarak, Bugünün PDA ve akıllı telefonları, belleği sadece kod ve veri depolama için kullanır. Mobil cihazlar, I / O uzantısı arabirimleri sıklıkla destekler böylece gigabaytlık kapasitesine sahip geniş flaş kartlarının kullanımına olanak sağlamaktadır. Daha hızlı erişim sağlarlar, sabit disklerden daha küçük ve hafiftirler. Sessizdirler ve mekanik parçaları yoktur, bu yüzden mobil cihazlar için sabit disklere göre tercih edilirler. Üstelik sabit disk aşırı tüketimi engelleyici biçimde yüksektir ve veri erişim hızı mobil uygulamaların ihtiyaçlarını karşılamaz.

Giriş cihazı ve Görüntü

Giriş cihazı, mobil cihazın kabulü için çok önemlidir; cep telefonlarında kullanılan mevcut giriş cihazları, PDA'lar, akıllı telefonlar ve diğer taşınabilir bilgi işlem cihazları şunlardır:

- Cep telefonu tuş takımı; 0-9 arası tuşlardan oluşan 12 tuşlu klavyedir, her bir tuş aynı zamanda bazı harfleri ve karakterleri ve çağrı, telefonu kapatın, menü, iptal şeklinde dört adet fonksiyon tuşunu gösterir.
- QWERTY klavye. Standart İngilizce bilgisayar klavyeleri veya yazı makinelerin küçük bir versiyonudur.
- Alfabetik klavye. Alfabetik olarak düzenlenmiş tuşlara sahip küçük bir klavye türü.
- Özel kalem-tabanlı sanal klavye ve El yazısı Tanıması: PDA' larda çoğunlukla popülerdirler, kullanıcı ya cihaz ekranında yazmak için özel kalemi kullanabilir ya

da ekranda görüntülenen sanal klavyede ki tuşlara tıklar. Özel kalem-tabanlı giriş yönetimi sıkı metin girişi için çok uygundur, ancak tek elle çalıştırılmaz.

Mobil cihazların ekranı uzun yıllar boyunca hızlı tempoda bir dizi yönde gelişmiştir. Ekran boyutları, esas olarak cihazın amacıyla belirlenerek cihazdan cihaza değişiklik gösterir. Ayrıca akıllı telefonların, örneğin, kablosuz internet uygulamaları ve kişisel bilgi yönetimi (PIM) için sıklıkla yeterince büyük ekranı vardır. Tasarım seçeneklerinin bir dizi faktöre ile ilgili yapılması gerekir:

- Boyut - cep telefonlarının genellikle arka aydınlatmalı 2.2-inç diyagonal LCD ekranları vardır. Akıllı telefonların 3,7-4 inç ekranları varken PDA' lar için ekran boyutu 2 ile 10 inç arasında değişebilir.
- Çözünürlük - düşük çözünürlüklü hala mobil cihazlar arasında yaygındır; ancak; QVGA (320x240) ama aynı zamanda 320x480 ve 480x360 yeni akıllı telefonlar için desteklenir.
- Renk derinliği- 16 bit (65536 renk) veya 18 bit (262144 renk) renk derinliğine sahip tüketici mobil cihazlarını görmek çok yaygındır
- Arka ışık- Daha iyi görüntü etkisine yönelik olarak, cep telefonu ya da PDA gösterim ekranlarının genellikle arka ışığı vardır.
- Güç tüketimi - İnce film transistör (TFT) ekranları, daha önceki pasif matris ekranlara göre daha fazla güç tüketme eğilimindedir.

Aslında, yüksek performanslı akıllı telefonlar dokunmatik ekran teknolojilerini kullanır; Böyle bir teknolojiye yönelik iki ana kategori vardır: kapasitif ve rezistif. Kapasitif dokunmatik ekranlar, ekrandan parmağınıza küçük bir elektrik yükünü aktararak ve yükün ortadan kalktığı bölgeyi tespit ederek çalışırlar. Rezistif ekranlar, camın altında ekrana dokunulduğunda birbirleriyle sıkışan iki adet son derece ince tabaka kullanır. İkisi arasındaki fark; rezistif ekranlara ekran kalemi gibi bir nesne ile dokunulabilirken, kapasitif ekranların genellikle parmaklarınız vasıtasıyla vücudunuzla temas kurmasını ifade eder. Uygulamada iPhone' da kullanıldığı şekliyle kapasitif ekranları, Sony Ericsson Satio gibi yeni dirençli ekranların, iyi tasarlanmış rezistif ekranların neredeyse duyarlı olabileceklerini göstermesine rağmen, daha duyarlı buluyoruz. Dokunmatik ekranlı cihazlar için dikkate alınacak ikinci önemli unsur, üreticileri tarafından kullanılan ara yüzün tasarımıdır. Dokunmatik ekranlar, simgelerin

parmakla basılacak kadar geniş ve de aralardaki simgelere yanlışlıkla basmamak için yeterince aralıklı olmasını talep eder.

3.3 Mobil Yazılım Platformları ve Uygulamaları

Aslında mobil cihazlarda yüksek performansa olan ihtiyaç, zengin işletim sistemleri ve java ve yürütme ortamı desteğine ilaveten, herhangi bir web sitesine göz atma, posta ve farklı türdeki ticari belgeleri işleme, karmaşık kullanıcı ara yüzlerini yönetme; gelecek nesil 3D oyunları destekleme; fotoğrafları ve videoları düzenleme ve herhangi bir formatta video izleme yeteneğini kapsar.

Bu nedenle, tüm mobil sektörünün dikeyden yatay iş modeline derin bir dönüşüm geçmesi gerekir. Açık ara yüzlerin ya standardizasyon yoluyla ya da açık şartnamelere sahip fiili endüstri standartları aracılığıyla benimsenmesi gerekmektedir - mobil platformlara yönelik artan maruziyete sahip açık kaynak topluluğu unutulmadan. Günümüzde mobil cihaz sektöründe ki birçok doğal sürücüler; modüler çözümler, standartlaştırılmış ara yüzler ve özel alan bilgisine sahip üçüncü tarafların kullanımına doğru itilmektedir (Suoranta, 2006). Mobil yazılım platformu, bir dizi ilgili yazılım geliştirme kütüphaneleri, uygulama programlama ara yüzleri (API) ve programlama araçlarına sahip uyumlu mobil cihazların bütününe yönelik bir işletim sistemi kombinasyonu olarak tanımlanır. Mobil yazılım platformları ya özel cihazlar için patentlidir ya da tüm bağımsız yazılım sağlayıcılarına açıktır.

Mobil donanım teknolojileri ve kablosuz teknolojileri ilerlemeye devam ettiğinden, mobil işletim sistemlerinin bu gelişmelerin sunduğu olanaklardan faydalanması ve uygulama geliştiriciler için güçlü bir destek sağlaması gerekir. Aşağıda büyük mobil platformlar sunulmaktadır.

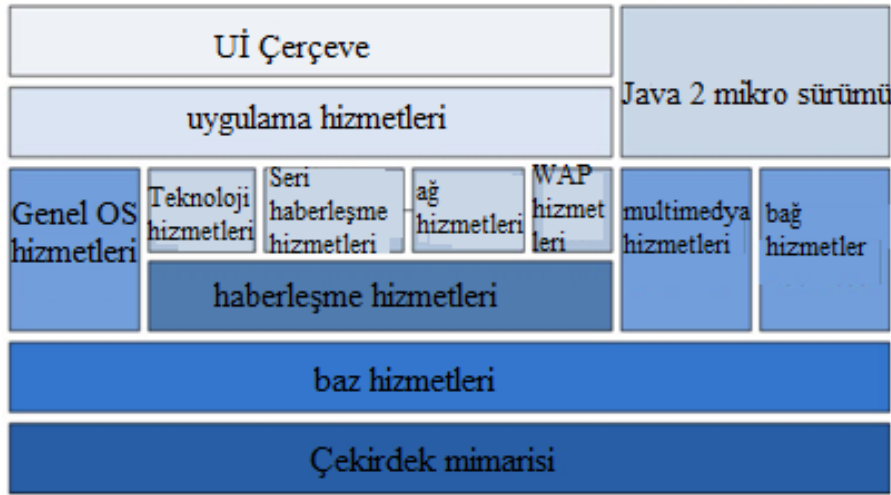
3.3.1 Symbian

Symbian; ilişkili kitaplıklar, kullanıcı ara yüzleri, çatılara ve Symbian Ltd tarafından geliştirilen yaygın araçları referans uygulamalarına sahip Symbian OS isminde açık standart işletim sistemi tedarik eder ve geliştirir özel bağımsız bir şirkettir. Bu, Psion'un EPOC soyundan geliyordu ve sadece ARM işlemcilerinde çalışır. Symbian OS; ses ve veri hizmetlerinin geniş bir yelpazesini 2G, 2.5G ve

3G hücresel sistemlerde desteklemek ve aynı zamanda mltimedya ve veri senkronizasyonuna katkıda bulunmak için tasarlanmıştır. Symbian OS, göz önüne alınan üç sistem tasarımı ilkesiyle oluşturulmuştur: Kullanıcı verisinin btnlğ ve gvenliđi çok önemlidir ve tm kaynaklar kıttır. Bu; iletiřimi, sreçle ve kullanımla alakalı kaynak ynetimini ve donanım soyutlamayı ve hata iřlemeyi ortaya koyan gerçek zamanlı, çoklu mesaj dizili, nleyici bir çekirdektir.

Ayrıca, tm Symbian programlaması olay tabanlıdır ve uygulamaları dođrudan bir olayla ilgilenmezken CPU dřk gç modunda aılır. Bu, aktif nesneler adındaki programlama deyimiyile elde edilir. Benzer řekilde mesaj dizilerine ve sreçlere ynelik Symbian yaklařımı, iřletim yk azaltılarak ynlendirilir. Symbian OS, EPOC C + , salt nesne ynelimli bir dili, hem sistem hizmetleri uygulamaları hem de uygulama programlama ara yzleri iin destekleyici programlama dili olarak kullanır.

Bu, mobil cihazlara ynelik Java uygulamalarının (J2ME apps) kçk bir Java çalıřtırma ortamı zerinde çalıřmalarına olanak verir. Symbian OS son srm; Samsung i8910 Omnia HD, Nokia N97, Nokia 5800 XpressMusic, Nokia 5530 XpressMusic ve Sony Ericsson Satio gibi kullanılmıř cihazlardır.



řekil 3.4 Symbian OS mimarisi

3.3.2 Palm

Palm OS (aynı zamanda Garnet OS olarak bilinen) Başlangıçta Palm, Inc tarafından 1996 yılında kişisel dijital yardımcılar (PDA) için geliştirilen bir mobil işletim sistemidir; Palm Inc, PDA pazarını oluşturan şirkettir.

Palm OS, dokunmatik ekran tabanlı grafik kullanıcı ara yüzüyle kullanım kolaylığına ve verimli işlemler sağlamak için sınırlı bilgi işlem gücünden faydalanmaya yönelik tasarlanmıştır. Kişisel bilgi yönetimine yönelik temel uygulamalar paketiyle donatılmıştır.

Kol saatleri, el oyun konsolları, barkot okuyucular ve GPS cihazları da dâhil olmak üzere mobil cihazların geniş bir dizisinde uygulamaya koyulmuştur. OS (İşletim Sisteminin) sonraki sürümleri akıllı telefonları desteklemek için genişletilmiştir.

Palm OS 6 (Cobalt), Palm OS tarihinin bir dönüm noktasıdır. Daha önceki sürümlerin eksiksiz yeniden yazılmış halidir ve çoklu kullanımları destekleyen ilktir. Ayrıca daha fazla kablosuz özelliği, mültimedya uygulama desteği ve uzatma yuvası çeşitliliği sağlar. Palm OS, sistemin bir parçası olarak kullanılmak üzere üçüncü taraf donanımı sağlar; Palm OS çekirdeği; sonsuz öncelikli çok görevlilik ve koruyucu bellek yönetimi sunan Kodak'tan lisanslı AMX' e dayanmaktadır. Sistem servisleri, iletişim ve giriş yönetimi sağlayan modüler bileşenlerden oluşan bir dizidir. GUI olay işleme ve mültimedya işleme. Üçüncü taraf yazılım kitaplıkları ayrıca sisteme takılabilir, örneğin J2ME profil uygulamaları harici kitaplıklar olarak eklenebilir.

Palm OS dosyaları saklamak için blok-tabanlı dosya modelinden ziyade veritabanı modelini kullanır. Son zamanlarda Palm webOS yayımladı. webOS, özel bileşenleriyle birlikte Linux çekirdeği üzerinde çalışan bir mobil işletim sistemidir. webOS' e ait grafiksel kullanıcı ara yüzü, dokunmatik ekranlara sahip cihazlarda kullanılmak üzere tasarlanmıştır. Kişisel bilgi yönetimine yönelik uygulamalar paketi içerir ve HTML 5, JavaScript ve CSS gibi bir dizi web teknolojilerinden faydalanır. Haziran 2009 tarihinde yayımlanan Palm Pre, bu platformu çalıştırmaya yönelik ilk cihazdır.

3.3.3 Windows Mobile

Windows Mobile, Microsoft tarafından geliştirilen kompakt bir işletim sistemidir ve akıllı telefonlar ve mobil cihazlarda kullanılmak üzere tasarlanmıştır. Windows CE ' ye dayanmaktadır ve diğerleri arasında PDA'lar, cep telefonları, akıllı telefonlar, barkot okuyucuları, el bilgisayarları ve otomobillerde gömülü cihazlar gibi çeşitli düşük kapasiteli mobil cihazları desteklemek için tasarlanmıştır. Çeşitli donanım aygıtlarına yönelik üç adet Windows Mobile sürümü vardır:

- Windows Mobile Professional, dokunmatik ekranlara sahip 'Windows Telefonları' üzerinde çalışır.
- Windows Mobile Standard, düzenli ekranlara sahip 'Windows Telefonları' üzerinde çalışır.
- “Windows Mobile Klasik Cihazları” (Cep Bilgisayarları) üzerinde çalışan Windows Mobile Classic

“Windows Mobile Klasik Cihazı’’, telefon özelliği taşımayan "Windows Mobile kişisel dijital asistan (PDA) 'dir. Eskiden Cep Bilgisayarı olarak biliniyordu. Bu, Windows Mobile işletim sistemi için asıl amaçlanan platform oldu.

.Net Compact Framework, arka plandaki akıllı telefon cihazına ait ROM' a yüklenir. .Net, Microsoft' un İnternet tabanlı bilgi işleme ait genel yazılım altyapısıdır. Özü XML Web servisleridir. Bunlar, heterojen İnternet servisleri ve uygulamalar arasında sorunsuz, sağlam ve güvenli işbirliğini etkinleştirmek için yaygın olarak çözüm olarak kabul edilir. Bu hedefe ulaşmayı mümkün kılan, veri ve yapısını kendini açıklayıcı bir biçimde bütünleştirebilen açık standart XML' dir. Öyle ki, onlar herhangi bir uygulama, hizmetler, web siteleri ve akıllı cihazlar arasında yerleştirilebilir, düzenlenebilir, programlanabilir ve değiştirilebilir. . Net altyapısının kalbi, tablet PC'ler, Cep Bilgisayarları ve Akıllı Telefonlar dahil çoğu Windows tabanlı işletim sistemlerindeki. Net Framework ' dir.

3.3.4 Gömülü Linux

Linux özgür, açık kaynak, UNIX benzeri bir işletim sistemidir. Aslında, iki kısımlı bir kombinasyonu ifade eder: Linux çekirdekleri ve Linux uygulamaları. Birçok sayıda ücretsiz, açık kaynak kodlu uygulamalar geliştirilmiştir ve Linux sistemlerinde kullanılmaktadır. Linux çekirdekleri ve

uygulamaları kaynak kodu Gnu Kamu Lisansı (GPL) kapsamında çoğunlukla mevcuttur. Linux; ağ donanımları, tüketici elektroniği, endüstriyel tesisler ve mobil cihazların geniş bir yelpazesine güç veren gömülü sistem pazarında da yayılmaktadır. Özellikle, PDA'lara yönelik bazı Linux dağıtımlarını internet üzerinden indirmek ücretsizdir ve Monta Vista Linux gibi mobil cihazlara yönelik ticari Linux sistemleri piyasada mevcuttur. Gömülü Linux sistemleri ya sert gerçek zamanlı ya da yumuşak gerçek zamanlı olabilir. Sert gerçek zamanlı, sistemin ilgili olayın meydana geldiği her zaman belirleyici bir şekilde yanıt vermek zorunluluğunu ifade eder. Yumuşak gerçek zamanlı sistemlerde, hızlı yanıt istenir ancak garanti verilmemektedir. Tüketiciler tarafından kullanılan mobil cihazlar genellikle ikinci kategoriye girer. En son Linux çekirdekleri, sonsuz öncelikli çekirdek tasarımı ve süreç programcısı nedeniyle hem sert gerçek zamanlı hem de yumuşak gerçek zamanlı uygulamaları destekler.

Özetle, gömülü Linux sistemlerinin ana özellikleri şunlardır:

- Monolitik çekirdek çoklu görev ve çoklu kullanımı destekler ve farklı uygulamalar senaryosu için uygun hale getirilebilir;
- Açık kaynak topluluğu, Bluetooth, kablosuz LAN ve kablosuz duyar ağı dâhil en son yeni teknolojiler için destek sağlar;
- GPL lisansı kapsamındaki uygulama değiştirilebilir ve uzatılabilir;
- Lisans ücreti düşüktür veya hiç yoktur.

Birçok sanayi grubu, gömülü uygulamalarda Linux kullanımını teşvik etmek için oluşmuştur. Özellikle Gömülü Linux Konsorsiyumu, uygulama taşınabilirliğini destekleyen standart bir platform sağlamak amacıyla hangi işlevselliğin dâhil edilmesi gerektiğine ilişkin geliştiricileri için bir rehber olarak tasarlanan ELCPS (Gömülü Linux Konsorsiyumu Platform Şartnamesini) üretmiştir.

3.3.5 RIM Blackberry

RIM, cihazın özel giriş aygıtlarının (özellikle kaydırma tekerleği veya daha yakın izleme topu) yoğun şekilde faydalanan BlackBerry için özel çok görevli bir işletim sistemini (OS) sağlar. OS, kolay kullanıma odaklanmıştır ve aslında işletme için tasarlanmıştır, Java MIDP 2.0 ve WAP 1.2 desteği sağlar. Üçüncü taraf geliştiriciler yazılımı ve aynı zamanda BlackBerry API'lerini patentli kullanarak yazabilirler. Ancak, Belli kısıtlı işlevsellikten faydalanan herhangi bir

uygulamanın dijital olarak imzalanmış olması gerekir, böylece RIM de ki bir geliştirici hesabıyla ilişkilendirilebilir. Bu imza prosedürü, bir uygulamanın yazarlığını güvence altına alır ancak kodun kalitesi ve güvenliğini garanti etmez. OS; push e-postayı, cep telefonunu, kısa mesajlaşmayı, internet faksını, web tarama ve diğer kablosuz bilgi hizmetleri ve aynı zamanda çoklu dokunmatik ara yüzü destekler.

Eski elde taşınır BlackBerry 950 ve 957, Intel 80386 işlemciler kullanırken, Modern GSM tabanlı elde taşınır BlackBerry ARM 7 veya 9 işlemci içermektedir. En yeni GSM BlackBerry modelleri (8100, 8300 ve 8700 serisi) Intel PXA901 312 MHz işlemci, 64 MB flash bellek ve 16 MB SDRAM donanımına sahiptirler. CDMA BlackBerry akıllı telefonları, ARM 9 tabanlı işlemci ve GSM 900/1800 dolaşımı ve 256MB a kadar flash bellek de içeren Qualcomm MSM6x00 çip setlerine dayanmaktadır. En son BlackBerry 9000 serisi, BlackBerry' i bugüne kadarki en hızlı haline getiren Intel XScale 624 MHz işlemci ile donatılmış. BlackBerry sunucularına bağlanan Black-Berry e-posta istemcisine sahip birçok BlackBerry olmayan cep telefonlu piyasaya sürülmüştür.

3.3.6 iPhone OS

Geçmişinde OS X veya OS X iPhone olarak bilinen iPhone OS, iPhone ve iPod touch için Apple tarafından geliştirilen işletim sistemidir. Türetildiği Mac OS X gibi, Darwin'in temelini kullanır. iPhone OS dört soyutlama katmanına sahiptir:

- Çekirdek OS katmanı;
- Çekirdek Servis katmanı;
- Medya katmanı;
- Cocoa Touch tabakası.

İşletim sistemi, cihazın toplam bellek hafızasının 240 Megabayt alanından azını kullanır. iPhone OS kullanıcı ara yüzü, çoklu dokunmatik jestleri kullanarak doğrudan manipülasyon kavramına dayanmaktadır. Ara yüz kontrol elemanları, kaydırıcılar, anahtarlar ve düğmelerden oluşur. Kullanıcı girişine yanıtın, akışkan bir ara yüz sağlamak için acil olması beklenir.

OS ile etkileşim jestleri, kaydırma, dokunma, kısma ve ters kısma gibi jestleri içerir. Ek olarak, dâhili ivmeölçer kullanan, cihaz yanlarını tutma (ekran yatay

yönde olsun diye) bazı uygulamalarda ekran yönünü değiştirir. Uygulama simgeleri içeren ana ekran ("SpringBoard" tarafından işlenen) ve kullanıcının en çok erişim sağladığı uygulamalara yönelik simgeleri gösteren ekranın alt kısmında ki yuva, cihaz açıldığında veya ana düğmesine basıldığında sunulur. Ekran üstünde; zaman, pil seviyesi ve sinyal gücü gibi verileri görüntülemek için bir durum çubuğu vardır. Ekranın geri kalanı mevcut uygulamaya ayrılmıştır. Uygulamaları başlatma veya bırakma kavramı yoktur, sadece ana ekrandan bir uygulamayı açma ve ana ekrana dönmek için uygulamadan ayrılma söz konusudur.

iPhone ve iPod Touch kullanılan merkezi işlem birimi, Apple'ın Macintosh bilgisayarlarda kullanılan x86 (ve önceki PowerPC veya MC680x0) işlemciler yerine ARM tabanlı bir işlemci (Cortex8) 'dir ve PowerVR 3D grafik donanım hızlandırıcı yardımcı işlemci tarafından oluşturulan OpenGL ES 1.1 kullanır. Mac OS X uygulamaları kopyalanamaz ve bir iPhone OS cihazında çalıştırılmaz. Özellikle iPhone OS ve ARM mimarisi için yazılı ve derlenmiş olmalıdırlar. Yetkili üçüncü taraf yerel uygulamaları, iPhone OS 2.0 sahip cihazlar için ve daha sonra Apple'ın App Store üzerinden kullanılabilir.

Apple, iPhone üzerinde çalışmak üzere Java yı etkileştirmeye yönelik herhangi bir plan açıklamadı. iPhone OS; Apple tarafından desteklenmeyen ancak yasa dışı da olmayan işlevlerin eklenmesi çevresinde toplanan çeşitli nedenlerden dolayı çeşitli hack' lere tabi olmuştur.

SDK yayımlanmadan önce üçüncü tarafların, Safari aracılığıyla çalışan "Web Apps" tasarımlarına izin verilmiştir. İmzasız yerli uygulamalar da mevcuttur. App Store dışında iPhone üzerine yeril uygulamaları yükleme yeteneği, Apple tarafından desteklenmeyecektir. Bu tür doğal uygulamalar herhangi bir yazılım güncelleştirmesi tarafından kırılmış olabilir ancak Apple, yazılım güncelleştirmelerini SIM kilidinin kırılmasını gerçekleştirenler dışında özellikle yerli uygulamaları kırmak üzere tasarlamayacağını ifade etmiştir.

SDK kendisi ücretsiz olarak indirilebilir, ancak yazılımı yayımlamak için biri, ödeme ve Apple'ın onayını gerektiren bir adım olan iPhone Geliştirici Programına kayıt olmalıdır.

İmzalı tuşlar, Apple'ın App Store için uygulamayı yüklemek amacıyla verilir. Uygulamalar üç şekilde dağıtılabilir: sadece şirket çalışanlarına kurumsal dağıtım yoluyla ve 100 iPhone'a kadar "Ad-hoc" temelinde.

Apple Store üzerinden dağıtıldığında geliştirici, geliştirmiş olduğu ticari uygulamanın serbest bir şekilde dağıtılması için kullanılabilecek 50 adet promosyon koduna kadar talepte bulunabilir.

iPhone yazılımına yönelik bu dağıtım modeli, GPLv3 ile lisanslı koda dayalı yazılımı yayımlamayı imkânsız hale getirdiği görülmektedir.

3.3.7 Android

Android; işletim sistemi, katman ve anahtar uygulamaları (3.5) içeren mobil cihazlar için bir yazılım yığınıdır. Android, Java programlama dilini kullanarak uygulamaları Android platformu üzerinden geliştirmeye başlamak için gerekli SDK araçlarını ve API'leri sağlar. Açık bir geliştirme platformu sağlayarak, Android geliştiricilere zengin ve yenilikçi uygulamaları oluşturma yeteneğini sunar. Geliştiriciler, cihaz donanım, erişim konum bilgileri, erişim konum bilgisinden yararlanmakta ve arka plan servislerini çalıştırmakta, alarmları kurmada ve durum çubuğuna bildirimleri eklemeye vs serbesttirler. Geliştiriciler, çekirdek uygulamalar tarafından kullanılan API'lerin aynı çatısına tam erişime sahiptirler. Uygulama mimarisi bileşenlerin yeniden kullanımını kolaylaştırmak için tasarlanmıştır; herhangi bir uygulama yeteneklerini yayımlayabilir ve diğer herhangi bir uygulama daha sonra bu yeteneklerden yararlanabilir (çatı tarafından uygulanan güvenlik kısıtlamalara tabidir). Bu aynı mekanizma bileşenlerin kullanıcı tarafından değiştirilmesine olanak sağlar. Örneğin, bir uygulama; aramalar yapma, metin mesajları gönderme veya kamera kullanma gibi telefonun çekirdek işlevselliği üzerinden çağırılabilir ve bu durum geliştiricilerin, kullanıcılar için daha zengin ve daha tutarlı deneyimler oluşturmalarına imkân sağlar. Android, açık Linux Çekirdeği üzerine kurulmuştur. Üstelik mobil bir ortamda ki bellek ve donanım kaynaklarını optimize etmek için tasarlanmış özel sanal bir makine kullanır. Android açık kaynaktır; yeni kesici kenar teknolojilerini ortaya çıktıklarında birleştirmek için serbestçe genişletilebilir.

Geliştirici topluluğu yenilikçi mobil uygulamalar oluşturmak için birlikte çalıştıkça platform gelişmeye devam edecektir.



Şekil 3.5 Android platform mimarisi

Android, kendi sisteminin çeşitli bileşenleri tarafından kullanılan C / C ++ kitaplıklar setini içerir. Bu yetenekler, Android uygulama çatısı yoluyla geliştiricilere maruz kalmaktadır. Çekirdek kitaplıklardan bazıları aşağıda sıralanmıştır:

- Sistem C kitaplığı - Standart C sistem kitaplığının (libc) BSD' den türetilmiş uygulaması, gömülü Linux tabanlı cihaz için ayarlanmış
- Medya Kitaplıkları - Paket Video'nun Açık ÇEKİRDEĞİNE dayalıdır; kitaplıkları; MPEG4, H.264, MP3, AAC, AMR, JPG ve PNG gibi birçok popüler ses ve video formatları yanı sıra statik görüntü dosyalarının yeniden çalma ve kaydını destekler.
- Yüzey Yöneticisi - görüntü alt sistemine erişimi yönetir ve çoklu uygulamalardan gelen 2D ve 3D grafik katmanlarını sorunsuz bir şekilde kompozitler
- LibWebCore: Android tarayıcısı ve aynı zamanda gömülebilir web görünümünü güçlendiren modern bir web tarayıcısı motoru.

- SGL - arka plandaki 2D grafik motoru
- 3D kitaplıkları - OpenGL ES 1.0 API'lere dayanan bir uygulama; kitaplıklar ya 3D donanım hızlandırmasını (varsa) kullanır ya da son derece optimize 3D yazılım taramasına dâhil edilmiştir.
- FreeType - bitmap ve vektörel font oluşturma
- SQLite - tüm uygulamalar için kullanılabilir güçlü ve hafif bir ilişkisel veritabanı motoru.

Google tarafından geliştirilen Java kütüphaneleri aracılığıyla cihazı kontrol ederek geliştiricilerin Java dilinde yönetilen kodları yazmalarına olanak verir; Android, güvenlik, bellek yönetimi, süreç yönetimi, ağ yığını ve sürücü modeli gibi çekirdek sistem hizmetlerine yönelik olarak Linux 2.6 sürümüne dayanır. Çekirdek aynı zamanda; donanım ve yazılım yığının geri kalanı arasındaki bir soyutlama katmanı olarak hareket eder.

Pazarda Android platformunu kullanan ilk telefon HTC Dream oldu; bunu HTC Hero, HTC MAgic, Motorola Droid, Samsung I7500 ve I5700 izledirler; HTC Google Nexus One Android platformunun son sürümü ile bekleniyor (sürüm 2.1).

3.3.8 Brew

Kablosuz için İkili Çalıştırma Ortamı (BREW), CDMA teknolojisini yaratan şirket olan QUALCOMM tarafından geliştirilen kablosuz bir yazılım platformu çözümüdür. BREW üç bileşenden oluşur: ikili çalışma zamanı ortamı, uygulama geliştirme ortamı ve dağıtım sistemi.

İkili çalışma zamanı ortamı, yerli BREW uygulamalarının cep telefonları tarafından kullanılan hava ara yüzüne bakılmaksızın çalışmasına imkân verir. Uygulama geliştirme ortamı, BREW yazılımı geliştirmeye yönelik olarak bir dizi platform API'lerini ve destekleyici araçları sunar.

Brew dağıtım sistemi (BDS), kablosuz veri hizmet sağlayıcılarının içeriği aboneye sunmalarını sağlar. Mobil cihazda konuşlandırılmak üzere olan tüm mobil uygulamalar, dijital bir anahtar ve sertifika kullanılarak imzalanmalıdır. Brew genellikle, mobil cihazlar için donanımdan bağımsız bir kablosuz çözüm olarak kabul edilir. Bu anlamda J2ME ile yarışmaktadır; Bu teknolojiler tamamen rakip

değildirler, örneğin Qualcomm BREW için, IBM'in J9 Java sanal makinesini seçmiştir, böylece Java uygulamasının yürütülmesine olanak sağlamıştır.

3.4 Java Micro Edition (J2ME)

Java mobil bir işletim sistemi değildir, ancak diğer mobil işletim sistemleri, kod taşınabilirliği ve uygulanan güvenliğini sağlamak için Java platformlarından yararlanabilir. Bu anlamda Java, özellikle J2ME, kablosuz servis sağlayıcılar tarafından sunulan belirli bir mobil işletim sistemi ve uygulamalar arasında ortak bir orta katman olarak performans sergilemektedir.

Tanıma göre Java, SUN Microsystems tarafından geliştirilmiş çapraz platformlu, nesne yönelimli, yorumlayıcı programlama dilidir. Java kalbinde, farklı platformlardan gelen ve Java derlenmiş bayt kodunu yorumlayan bir dizi Java Sanal Makineleri (JVM'ler) vardır. Bir platform üzerinde Java uygulamalarını çalıştırmak için, Java Çalıştırma Ortamı (JRE) gereklidir. Bir JRE, arka plan platformu için JVM' den ve standart paketlerdeki çekirdek sınıfları ve bazı destekleyici dosyalardan oluşur.

Sun Microsystems den edinilebilecek Resmi Java platformları (JRE ve yazılım geliştirme kiti) , bilgi işlem türünü hedefleyen her biri:

- Masaüstü ve sunucu sistemleri için Java 2 Standart Sürüm (J2SE) • kurumsal uygulama sistemleri için Java 2 Kurumsal Sürüm (J2EE).
- Gömülü sistemler için Java 2 Micro Edition (J2ME).
- Akıllı kart uygulamaları için Java Kartı.

J2ME ile sürekli genişleyen gömülü cihazlar pazarı hedeflenmektedir. Bu segmentteki cihazların çeşitliliği nedeniyle, J2ME ayrıca iki farklı yapılandırmaya büründürülmüştür: Mobil cihazlar için *Bağlı Sınırlı Cihaz Konfigürasyonu (CLDC)* ve müşteri ve gömülü cihazlar için *Bağlı Cihaz Konfigürasyonu (CDC)*.

J2ME konfigürasyonu, bir bütün olarak benzer donanım ve ağ yeteneklerine sahip gömülü cihazlar topluluğuna yönelik Java çalışma zamanı ortamını temsil eden JVM'ler ve destekleyici Java sınıfları ve API' ler için bir dizi temel gereksinimden oluşur.

CLDC aygıtlarının özellikleri aşağıdaki gibi özetlenebilir:

- 6 MHz veya daha yüksek bilgisayar çalışma hızına sahip 16-bit veya 32-bit işlemci
- Düşük bellek bütçesi, 160-512 Kilobayt
- Düşük güç tüketimi ve sınırlı güç kaynağı (çoğunlukla pil)
- Basit kullanıcı ara yüzü, ya da hiçbir ara yüzü yoktur
- Düşük bant genişliği ve muhtemelen aralıklı ağ bağlantısı

CLDC cihazlarının örnekleri; cep telefonları, alt uçtaki PDA'lar, çağrı cihazları, kablosuz sensorlar, radyo frekansı tanımlama (RFID) cihazları vs ihtiva eder. CLDC cihazlar, güç, işleme kapasitesi, bellek kapasitesi ve ağ yeteneği açısından kısıtlı kaynaktır; bu yüzden, standart JVM kullanılamaz. Bu yüzden, Sadeleştirilmiş JVM, Kilobayt sanal makine (KVM), CLDC yapılandırması için son derece arzu edilen bir durumdur.

Sun, CLDC yapılandırmasıyla birlikte KVM referans uygulanmasını temin etmiştir.

Qualcomm tarafından BREW platformu için seçilmiş olan IBM'in J9 sanal makinesi gibi diğer KVM uygulamaları da mevcuttur. CLDC cihazlarının özellikleri şunlardır:

- 32-bit bir işlemci
- Geniş bellek bütçesi, ava uygulama ortamında kullanılabilir 2MB RAM ve 2.5 MB ROM
- Kablolu güç kaynağı veya uzun pil süresi
- Kararlı ağ bağlantısı
- Sofistike GUI'ler den UI olmayanlara kadar çeşitli kullanıcı ara yüzleri

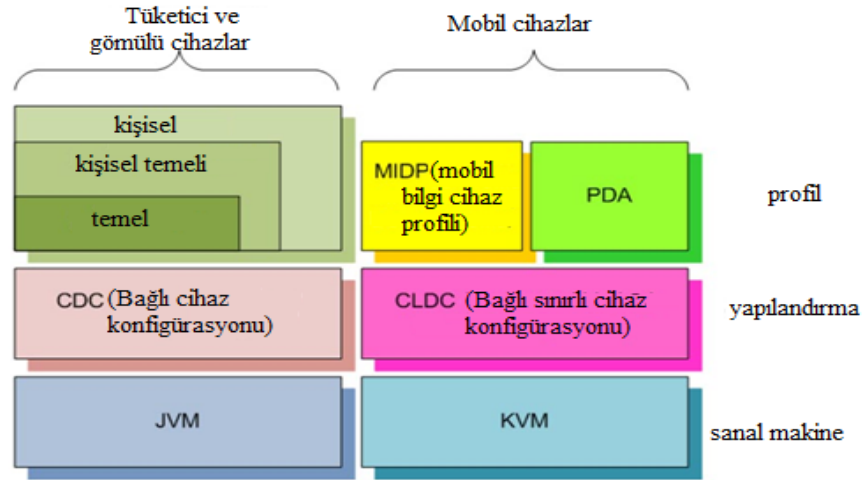
Örnekler, üst uç PDA'lar, televizyon set üstü kutuları ve RFID okuyucuları içerir. CLDC' den farklı olarak, CLDC konfigürasyonu standart JVM kullanır. J2ME yapılandırmaları, çeşitli cihazların uygulama senaryolarına ilişkin dikey sınıflandırılmadan ziyade donanım ve ağ özelliklerine göre tanımlanır. Aynı J2ME yapılandırma cihazları tarafından sağlanan işlevsellik büyük ölçüde değişebilir, arka plandaki ortak zemin J2ME konfigürasyonu tarafından sağlanmayan ilave API'leri gerektirir. Bu nedenle, Bazı J2ME profilleri her konfigürasyon için tanımlanmıştır. Profil, J2ME yapılandırması içerisinde ki cihazların belirli dar bir sınıfına yönelik bir dizi standart Java API'leridir.

CDC yapılandırması için bir kuruluş profili, kişisel basis profili ve kişisel profil tanımlanır; kuruluş profili diğer iki profil için çekirdek gibi hareket eder. CLDC yapılandırması için, Mobil Bilgiler Cihaz Profili (MIDP) ve PDA profili geliştirmiştir. Cep telefonları ve akıllı telefonlar için, CLDC yapılandırılması ve MIDP profili, Şekil 3.6 'da gösterildiği gibi standart bir Java platformundan oluşur.

3.5 En Yaygın Cihazlar ve Platformlar Arasındaki Karşılaştırma

Symbian OS ve Palm OS tarih boyunca sırasıyla cep telefonları ve PDA'lar için özel mobil işletim sistemleridir. Zorluk, küçük biçim katsayısı üzerinde çok sınırlı kaynak sahibi geniş yelpazede mobil telefon hizmetleri ve mobil veri hizmetleri sunmaktır.

Açık kaynak kodlu bir işletim sistemi olarak, Linux, bir dizi mobil cihaz üreticisi tarafından düşük maliyetli ve son derece özelleştirilebilir bir çözüm olarak benimsendi.



Şekil 3.6 konfigürasyonları ve profilleri

J2ME CLDC bağımsız bir platformdur ve J2ME' i destekleyen herhangi bir mobil işletim sisteminde kullanılabilir.

Akıllı Telefonlara yönelik Microsoft Windows Mobile Windows CE işletim sisteminin genel amaçlarından kaynaklanmıştır ve akıllı telefonlar için tasarlanmış olsa bile Win CE ' nin geliştirilmiş mimarisi nedeniyle bilgi işlem kaynakları açısından çok fazlasını gerektirdiği bilinmektedir. Kendisinin yazılım ve genel özellikleri temin ettiği durumlarda Microsoft yaklaşımı, telefonları

tasarlama ve yapma işini telefon üreticilerine bırakmaktır. Tam tersine hem Apple's iPhone hem de Research In Motion's BlackBerry, tepeden tırnağa tek bir şirket tarafından kontrol edilen dikey olarak entegre edilmiş ürünlerdir.

Entegrasyonun avantajları açıktır. PC'lerde daha fazla el takımında kullanıcı arayüzü başarısı, donanımın belirli bir parçası için tasarlanmasına bağlıdır. OS X iPhone sürümünün her biti; cihaz, klavye ve özellikle ekranın özel yeteneklerine uygun görünmektedir. Kullanıcı deneyiminin çok iyi olmasının en büyük nedeni budur. Sonuç olarak, müşteriler paralarını öderler ve seçimlerini yaparlar.

Yukarıdan aşağı iPhone-benzerliği, belki de her zaman en iyi Entegre ürünler elde edecektir.

Kullanıcının ödediği fiyat, Apple'ın tüm seçimlerini yapmak için aldığı tutardır. Apple, üzerinde hangi programların çalışmasına izin verildiğine dair ayrıca kontrol yetkisine sahiptir. Apple, üzerinde çalışan donanım (pil, ekran vb) ve yazılım (işletim sistemi, uygulamalar, vb) dâhil olmak üzere cihazını her yönüyle kontrol etmeye ilişkin stratejiyi desteklemek üzere birkaç tartışma yapmıştır.

Çok daha olgun bir Android'in sunulması (Motorola / Verizon Droid 2.0 sürümü ve beklenen 2.1 HTC Google Nexus One) akıllı telefonların geleceğine yönelik olarak dikey entegrasyon modelleri ve açık platformlar arasında adil bir mücadele göreceğimiz anlamına geliyor. Android gerçek bir şansa sahip çünkü çok daha modern bir tasarım olarak başladı ve Google internet zamanında gelişme yeteneğini göstermiştir. Bir yıldan biraz fazla bir sürede, aralarında iki önemli noktada (1.5 "Cupcake" ve 1.6 "Donut") sürümleri olan iki büyük Android sürümü gördük. Hala Android platformu yaklaşımında riskler vardır. Biri, Google'un donanım üreticilerinin nasıl açık kaynak kodlu Android kullandıklarına dair nispeten daha az kontrole sahip olmasıdır. En büyük risk, yazılım geliştirme işinde kod çatalı olarak bilinen olurdu. Kod çatalı, hepsi aynı uygulamaları çalıştırmayan farklı Android sürümlerini göreceğimizi ifade eder. Bu durumda, Google bunun gerçekleşmesini sağlamak üzere ağırlık ve nüfuzunu kullanacaktır. Hatta fiziksel klavye olan veya olmayan telefon cihazları gibi nispeten basit varyasyonlar, yazılım uzlaşmalarına yol açabilirler ve daha az optimal deneyimler ortaya koyabilirler.

RIM' in daha farklı yeteneklere sahip cihazları desteklemek zorunda olduğu BlackBerry diyarında işler biraz daha pürüzlüdür. Dokunmatik ekranlı Storm2,

özellikle her OS ' ye uygunluk üzerinde büyük bir yük oldu; Klavye donanımlı BlackBerry'ler için tasarlanmış bazı özellikler, büyük dokunmatik ekran için doğru gelmiyor. Yinede RIM çok iyi bir deneyim sunmada da çok iyi iş çıkarmıştır.

Windows Mobile şimdiye kadar lider platform tercihi olmuştur (Symbian' nin rakip olduğu), fakat en yeni sürümü 6.5' in henüz çözüm bulamadığı bazı sorunları vardır; yeni dokunmatik telefonu bir kalemle gelir, bu durum bizi hemen on yıl veya daha eskiye götürür. Daha da kötüsü kalem, hala WinMo' nun gerektirdiği duyarsız rezistif dokunmatik ekranıyla sadece atalarına çekmemiştir. Kullanıcıların gerçekten ona ihtiyaç duyduğu zamanlar vardır. Daha modern kapasitif ekranları destekleyecek ara sürümü bir kaç ay içinde açıklayacaklar ancak telefonlardaki büyük revizyonun, Windows Mobile 7, şundan itibaren bir yıla kadar ortaya çıkması beklenmemektedir. Bugün, iPhone, BlackBerry ve Android ile birlikte, yazılım ve donanım geliştirmeye ilişkin farklı yaklaşımlara sahip üç dünya standartlarında ürün ailesine sahibiz. Black-Berry, e-posta-merkezli dünyasında ağırlıklı olarak yaşamak eğilimindedir ancak iPhone ve Android onunla kafa kafaya gitmektedir. Geliştiriciler Windows Mobile' i terk ettiler ve şimdi bunun yerine Android ve iPhone için uygulamalar yazıyorlar. Bu nedenle, Windows Mobile ' in Uygulama Piyasasında az 300 uygulaması vardır. Windows CE uzun zaman öncesinden eski uygulamalara sahipken, Windows Mobile çok az yeni uygulamaya sahiptir, çünkü hiçbir yeni gelişme olmamaktadır. Microsoft önümüzdeki yıl, çoklu dokunmatik jestleri getirecek ve kapasitif ekranları işletim sistemi (OS) düzeyinde destekleyecek Windows Mobile 7 programını yayınlayacaktır. Bu güncelleştirme, rakiplerin 3 yıl önce oldukları yeri yakalayacaktır. Ama yine de hala birkaç uygulaması olacaktır, Windows Mobile 6.5' in bugünkü durumundan çok az kazanacaktır. Kaçınılmaz sonucu, Windows Mobile' in tüketici akıllı telefonları için durdurulması olacaktır. Ancak, Windows CE türevleri hala yaşamaya devam edecektir ancak yaşamları sadece özel endüstriyel ortamlarda olacaktır.

Aşağıda 2015 yılında özelliklerin mükemmel karışımını, performansı ve paranın karşılığını sunan akıllı telefonları özetleyen bir tablosu 3.5 ve 3.6 mevcuttur.

Tablo 3.5 Akıllı telefonlar arasında karşılaştırma - bölüm 1.

	 HTC One (M8) CDMA	 Samsung Galaxy S5	 Apple iPhone 5s
Battery	Non-removable Li-Po 2600 mAh battery Talk time: Up to 14 h (2G) / Up to 20 h (3G)	Li-Ion 2800 mAh battery Talk time: Up to 21 h	Non-removable Li-Po 1560 mAh battery (5.92 Wh) Talk time: Up to 10 h (2G) / Up to 10 h (3G)
Camera	5 MP, 1080p@30fps, HDR	16 MP, 5312 x 2988 pixels, , LED flash	8 MP, 3264 x 2448 pixels, autofocus, dual-LED (dual tone) flash
Connectivity	Wi-Fi 802.11 a/b/g/n/ac, Wi-Fi Direct, DLNA, Wi-Fi Bluetooth: 4.0 with A2DP	Wi-Fi 802.11 a/b/g/n/ac, dual-band, DLNA, Wi-Fi Direct, Wi-Fi hotspot Bluetooth: v4.0 with A2DP, EDR, LE	Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi hotspot Bluetooth: v4.0 with A2DP
Display	Super LCD3 capacitive touchscreen, 16M colors 1080 x 1920 pixels, 5.0 inches	Super AMOLED capacitive touchscreen, 16M colors 1080 x 1920 pixels, 5.1 inches	LED-backlit IPS LCD, capacitive touchscreen, 16M colors 640 x 1136 pixels, 4.0 inches

Memory	Internal: 16/32 GB, 2 GB RAM microSD, up to 128 GB	Internal: 16/32 GB storage, 2 GB RAM microSD, up to 128 GB	Internal:16/32/64 GB storage, 1 GB RAM DDR3 microSD: NO
OS	Android OS, v4.4.2 (KitKat)	Android OS, v4.4.2 (KitKat)	iOS 7, upgradable to iOS 7.1.1
Input method	Touchscreen	Touchscreen	Touchscreen
Dim. WxDxH	146.4 x 70.6 x 9.4 mm (5.76 x 2.78 x 0.37 in)	142 x 72.5 x 8.1 mm (5.59 x 2.85 x 0.32 in)	123.8 x 58.6 x 7.6 mm (4.87 x 2.31 x 0.30 in)
Video and picture	DivX/XviD/MP4/H.263/H.264/WMV player MP3/eAAC+/WMA/WAV/FLAC player	MP4/DivX/XviD/WMV/H.264/H.263 player MP3/WAV/WMA/eAAC+/FLAC player	-Image viewer/editor - Voice memo/dial/command

Tablo 3.6 Akıllı telefonlar arasında karşılaştırma - bölüm 2.

	 Nokia Lumia 1520	 BlackBerry Z30	 Sony Xperia M2
Battery	Non-removable Li-Ion 3400 mAh battery (BV-4BW) Talk time: Up to 27 h 40 min (2G) / Up to 25 h (3G)	Li-Ion 2880 mAh battery Talk time: Up to 25 h (2G) / Up to 18 h (3G)	Non-removable Li-Ion 2300 mAh battery Talk time: Up to 13 h (2G) / Up to 14 h 36 min (3G)

Camera	20 MP, 4992 x 3744 pixels, Carl Zeiss optics, optical image stabilization, autofocus, dual-LED flash	8 MP, 3264 x 2448 pixels, autofocus, LED flash	8 MP, 3264 x 2448 pixels, autofocus, LED flash
Connectivity	Wi-Fi 802.11 a/b/g/n/ac, dual-band, Wi-Fi hotspot Bluetooth: v4.0 with A2DP	Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, Wi-Fi hotspot Bluetooth: v 4.0 with A2DP	Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi Direct, DLNA, Wi-Fi hotspot Bluetooth: v4.0 with A2DP
Display	IPS LCD capacitive touchscreen, 16M colors 1080 x 1920 pixels, 6.0 inches	Super AMOLED capacitive touchscreen, 16M colors 720 x 1280 pixels, 5.0 inches	TFT capacitive touchscreen, 16M colors 540 x 960 pixels, 4.8 inches
Memory	Internal: 16 GB (RM-940 only)/32 GB, 2 GB RAM microSD, up to 64 GB	Internal:16 GB, 2 GB RAM microSD, up to 64 GB	Internal: 8 GB, 1 GB RAM microSD, up to 32 GB
OS	Microsoft Windows Phone 8 Black	BlackBerry 10.2 OS, upgradable to 10.2.1	Android OS, v4.3 (Jelly Bean), v4.4.2 (KitKat)
Input method	touchscreen	touchscreen	touchscreen
Dim. WxDxH	162.8 x 85.4 x 8.7 mm, 120.3 cc (6.41 x 3.36 x 0.34 in)	140.7 x 72 x 9.4 mm (5.54 x 2.83 x 0.37 in)	139.7 x 71.1 x 8.6 mm (5.5 x 2.80 x 0.34 in)
Video and picture	MP3/WMA/WAV/eAAC+ player MP4/DivX/XviD/H.264/H.263/WMV player	MP3/WMA/WAV/eAAC+/FLAC player DivX/XviD/MP4/WMV/H.263/H.264 player	MP4/H.263/H.264 player MP3/eAAC+/WAV player

4. MOBİL HEMOFİLİ HASTA TAKİP SİSTEMİ

Hemofili hastalığı hastane dışında da izlem gerektiren kronik bir hastalıktır. Bu amaçla hemofili hastalarının hastane dışında takibini sağlamak için kullanılacak bir mobil hasta takip sistemi geliştirilmiştir. Bu bölümde önce hemofili hastalığı tanıtılacak, daha sonra geliştirilen uygulamanın teknik açıklamaları ve kullanım kılavuzu verilecektir.

4.1 Hemofili Hastalığı

Hemofili hastalarında en sık görülen kanama yerleri, eklemler ve ardından kas/yumuşak dokulardır. Kanama yerleri, sıklıkları, ilişkili klinik bulgular tanı bölümünde verilmiştir. Farklı eklemlere göre kanama sıklığı da değişmektedir. Eklem kanamaları öncelikli olurken diğer sistemlerle ilişkili kanamalar morbiditeyi ve mortaliteyi arttırabilmektedir. Kanama tedavisine geçmeden önce, bir hemofili hastasının izlem ve tedavisinin multidisipliner yaklaşımla yapılması gerektiği unutulmamalıdır.

Hematoloji disiplini, Ortopedi, Fizik Tedavi, Genetik, Hepatoloji, Enfeksiyon Hastalıkları, Nükleer Tıp, İmmunoloji ve Diş Hekimliği ile yakın işbirliği içinde çalışmalıdır.

4.1.1 Tanım

Hemofili, faktör VIII veya IX eksikliği sonucunda gelişen nadir bir kalıtsal kanama bozukluğu olup, eklem içi (hemartroz) ve kas içi (hematom) kanamalarla kendini gösteren bir grup hastalıktır.

Faktör VIII eksikliği Hemofili A, faktör IX eksikliği ise Hemofili B olarak adlandırılır.

4.1.2 İnsidans

Hemofili A yaklaşık 5.000-10.000 erkek doğumunda bir olarak görülürken, sıklığı Hemofili B'ye göre 5-6 kat daha fazladır. Tüm hemofiliklerin %85'ini Hemofili A, %15 kadarını ise hemofili B oluşturmaktadır. Hastalık insidansı tüm coğrafyalarda aynı olup ırksal farklılık göstermez. Bu doğumsal kanama bozukluğunun tanısı yaşla birlikte artar. Bebek büyüdükçe, özellikle emeklemeye ve yürümeye başladığında hastalığın klinik bulguları ortaya

çıkılmaktadır. Hastalığın ağırlık derecesi arttıkça, tanı yaşı erkene kaymakta ve klinik bulgular daha şiddetli yaşanmaktadır. Bu nedenle hafif hemofili hastalarındaki kanamaların genellikle daha ciddi travmalar veya cerrahi girişimler sonucunda gelişmesi nedeniyle tanı daha ileri yaşlarda koyulmaktadır.

4.1.3 Genetik

Hemofili, X'e bağlı resesif geçiş gösteren doğumsal bir kanama bozukluğudur. FVIII ve IX geni X. kromozomun uzun kolunda yerleşiktir (sırasıyla Xq27 ve Xq28). Hastalık çoğunlukla taşıyıcı kadınlardan, erkek çocuklarına geçmektedir. Hastalık çok nadir olmakla birlikte kız çocuklarda da görülebilir. Örneğin taşıyıcı bir kadın ile hasta bir erkeğin birlikteliğinde veya Turner Sendromu gibi tek X kromozomu varlığında hastalık kız çocuklarda da ortaya çıkabilir. Olguların yaklaşık 1/3 kadarında aile öyküsü olmaksızın hastalık spontan de-novo mutasyonlar ile ortaya çıkabilir. Hemofiliye neden olan moleküler genetik mutasyonların yelpazesi oldukça geniş olmakla birlikte, yaklaşık %10 olguda mutasyon tam olarak tanımlanamamaktadır. Ağır hemofili A tanılı olguların yaklaşık yarısında veya tüm olguların yaklaşık %30 kadarında “faktör VIII intron 22 inversiyon mutasyonu” bulunmaktadır. Hemofili B hastalarında ise sıklıkla nokta mutasyonları ile mRNA splice-site mutasyonları görülmektedir. FVIII veya FIX'a karşı inhibitör gelişen hastalarda ise büyük delesyonlar ve nonsense mutasyonlar daha yaygın görülmektedir.

4.1.4 Klinik tanı

Erken çocukluk döneminde kolay ekimoz oluşumu, özellikle eklem içi ve kas içi spontan kanamaların varlığı ve girişimler travma sonrası beklenenden uzun süren kanama öyküsünün varlığı hemofiliyi akla getirmelidir. Kanama bulgularının ağırlığı faktör VIII veya IX'un eksiklik derecesiyle doğrudan ilişkilidir. Faktör aktivitesi <%1 olan hastalar “ağır hemofili” kliniği gösterirken, %1-5 arasında olanlar “orta hemofili” ve >%5 olanlar “hafif hemofili” kliniği gösterirler.

Eklem boşluğuna kanama (hemartroz) ve kas içine kanama (hematom) hastalığın en tipik özelliğidir. Hastalığın ağırlık derecesi ve karşılaşılan travmayla ilişkili olarak yenidoğan döneminde vakum uygulamasına bağlı kafa içi, subdural veya periostal kanamalar olabilir. Orta veya ağır hemofili tanılı çocuklarda emekleme

ve yürümeye başladıktan sonra eklem kanamaları, hematomlar ve travmatik ağız içi kanamalar oluşmaya başlamaktadır. Hastaların önemli bir kısmı ise sünnet sonrasında durdurulamayan kanama nedeniyle tanı almaktadırlar.

4.1.5 Kanama yerlerine göre klinik özellikler

4.1.5.1 Eklem kanamaları

Hemofilide en sık kanayan eklemler diz, dirsek ve ayak bileği eklemleridir. Ancak her eklem kanayabilir. Kanayan eklemde ağrı, şişlik, kızarıklık, ısı artışı ve hareket kısıtlılığı olabilir. Ancak “ağrı ve hareket kısıtlılığı” eklem kanamalarının en önemli bulgularıdır.

Eklem boşluğuna kanamalar sinoviyal membranda inflamasyona yol açar (sinovit). Tekrarlayan kanamalar sonucu gelişen kronik sinovit ile sinoviyal membran kalınlaşır. Kan içindeki proteolitik enzimler eklem kıkırdağında dejenerasyona ve sonuçta eklem boşluğunda daralmaya neden olur.

Özellikle hedef eklemlerde (6 aylık süre içinde 3 kez ya da daha fazla sayıda kanayan eklemler) bu olumsuz gelişmeler çok daha hızlı yaşanır ve “kronik hemofilik artropati” denilen ve hemofilinin en sık rastlanan kronik komplikasyonu gelişir. Eğer zamanında gerekli önlemler alınmaz ise eklem hareket açıklığının tamamen kaybolması (ankiloz) ile kalıcı sakatlıklar gelişir.

4.1.5.2 Kas içi kanamalar

Çocuk emekleme veya yürümeye başladıktan sonra travmatik olarak özellikle büyük kas gruplarında hematomlar oluşmaktadır.

Görülme sıklığı yaklaşık %10-20 civarındadır. İlgili kasta şişlik, ısı artışı, ağrı, kızarıklık olması ile tanı konur. Kasın katıldığı eklem hareketleri sınırlandırılmış ve ağrılıdır.

4.1.5.3 İliopsoas kanaması

Kanama miktarının fazla olabilmesi ve atlanma olasılığı bakımından hemofilideki ciddi kanamalardan biridir. Hastada karın ağrısının yanında dik yürüyememe, kalçada fleksiyon varlığı söz konusudur. Kalça ve/veya dizin pasif ekstansiyonu son derece ağrılı olup kısıtlanmıştır. Femoral sinir basısı nedeniyle

bacakta uyuşma, karıncalanma ve duyu kaybı olabilir. En hızlı tanı yöntemi USG olmak ile birlikte BT ve MR ile daha ayrıntılı bilgi alınabilir.

Zamanında tedavi edilmezse yaşamı tehdit eden kanamalardan kalıcı hareket kısıtlılığına varan spektrumda sonuçlar doğurabilir.

Bu nedenle erken, yeterli doz ve süredeki tedavi çok önemlidir.

4.1.5.4 Kompartıman sendromu

El, el bileği, ön-kol, anterior veya posterior tibial kompartmanlar gibi kapalı boşluğa olan kanamalarda sinir ve kan damarları basıya uğrayabilir. Ağrı ve ödem gibi başlangıç belirtileri sonrasında uyuşma, güç kaybı ve distal nabızların kaybolması görülebilir.

Acil yerine koyma tedavisi ile etkin sonuç alınamaz ve bulguların ilerlemesi durdurulamaz ise cerrahi dekompresyon uygulanabilir.

4.1.5.5 Merkezi sinir sistemi kanaması

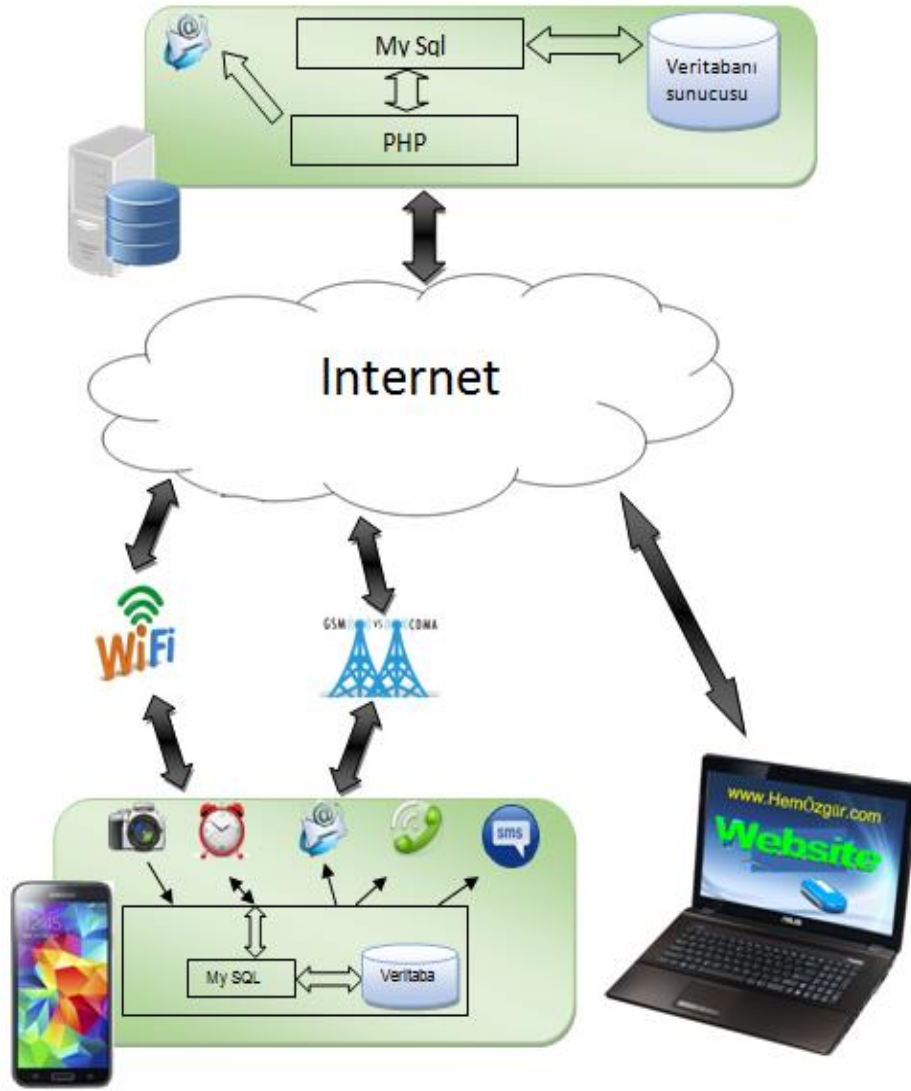
Genellikle travma sonucunda gelişmektedir. Şüpheli travma sonrasında görüntüleme yöntemleriyle aksi kanıtlanıncaya dek, acil bir durum olarak ele alınmalı ve faktör yerine koyma tedavisine başlanmalıdır. Görüntüleme yöntemleri veya nöroloji konsültasyonundan önce “acil olarak faktör tedavisine başlanmalı” ve faktör aktivitesi %80-100 düzeyine yükseltilmelidir. Geç kanama bulguları bazen travmadan 3-4 hafta sonra da ortaya çıkabileceğinden hasta yeterli bir süre dikkatle izlenmeli, şüpheli durumlarda görüntüleme yöntemleri ile ilerleme olup olmadığı takip edilmelidir.

4.2 Mobil Hemofili Takip Sistemi Uygulaması

Hemofili hastalarının hastane dışında yaşadığı kanama ve ilaç infüzyonu olaylarının takip edilmesi, rutin ilaç infüzyonu zamanının hatırlatılması, raporlarmaya, takip eden doktorla iletişime geçilmesi gibi özelliklere sahip bir Mobil Hemofili Takip Sistemi (HemoÖzgür) geliştirilmiştir. Bu sistem Android işletim sistemi üzerinde çalışan bir mobil uygulama, bir web uygulaması ve bir veritabanı sunucusundan oluşmaktadır. Bu sistemin mimarisi Şekil 4.1’de görülmektedir.

4.2.1 Sistem mimarisi

Geliştirilen sistemin mimarisi Şekil 4.1’de görülmektedir.



Şekil 4.1 Sistem mimarisi

Mobil Hemofili Takip Sistemi'nin mimarisini üç bölümde inceleyebiliriz. Birinci bölüm Android işletim sistemi üzerinde çalışan bir mobil uygulamadır ve arayüzleri Şekil 4.2 ile Şekil 4.37 arasında görülebilir. Mobil uygulamada tutulan veritabanı içinde üç tablo vardır:

1. Kullanıcı tablosu
2. Log Bleeding tablosu
3. Log Infusion tablosu

Programı başlattığınızda e-posta ve şifrenizi giriniz, bu bilgiler sunucuyla bağlantısı kullanıcıların kayıtlı olduğundan emin olmak içindir. Kayıtlı kişiler verilerine erişebilir, üç tablolardan veri getirilir ve Mobil veri tabanında saklanır. Kullanıcıların (hastaların) program üzerinde aşağıdaki işlemleri yapma yetkisi vardır:

1. Yeni vakaların tanıtılması.
2. Daha önceki vakaların değişmesi.
3. Daha önceki vakaların silinmesi.
4. Raporlar hazırlayarak e-posta yoluyla veya SMS yoluyla bir uzman doktora veya bu alanda ilgilenen herhangi bir kişiye gönderebilme.
5. Yazılım bir acil listesini oluşturacaktır. Kullanıcı programa cep telefonu numarası, e-posta yoluyla veya SMS yoluyla gerekli kişileri ekleyebilir ve onların iletişim verilerini kullanarak veri gönderebilirsiniz.
6. Ek, değişiklik ya da mevcut verilerin silinmesi durumlarında, sunucuda da gerekli güncelleştirmelerin yapılabilmesi için mobil uygulama ve sunucu arasındaki senkronizasyon vardır.
7. Programdaki kamera sayesinde ilaç barkodunun fotoğrafını çekebilir ve bu barkod numarası ile ilaca ait tüm bilgileri elde edebilirsiniz.

Program bazı durumlarda otomatik uyarı verir:

1. Bir ay içinde vücudun aynı kısmında üç kez tekrarlanan kanama olduğunda.
2. Altı ay içinde eklemlerin üçten fazla kanama tekrarlanması olduğunda.

Mimarinin ikinci bölümü, bir web uygulamasıdır ve bu uygulamanın arayüzleri de Şekil 4.38 ile Şekil 4.46 arasında görülebilir. Hem hemofili hastalığı ile ilgili bilgilerin yer aldığı, hem de doktor ve hastaların kayıtlı veriler üzerinde yetkileri dahilinde çeşitli sorgulamalar yapabildiği bir web sitesi geliştirilmiştir.

Bu web uygulamasının özellikleri şunlardır:

- a. Ana arayüzde bu hastalık ve hastalığın farklı durumlarda bir tanımı yapılmıştır Şekil 4.39'da gösterildiği gibi.
- b. İkinci arayüzü daha önce Android yazılım ve şifre ile kaydedilmiştir. Şifrenizi unuttuğunuz zaman programa e-postanızı girdiğinizde mailinize yeni şifre gönderilir. Bu sayede programa tekrardan giriş yapabilirsiniz. Şekil 4.40'da gösterildiği gibi aşağıdaki veriler görüntülenir ancak değişiklik veya silme yapılamaz:

- 1) log bleeds
- 2) log infusion

Mimarinin üçüncü bölümü ise, PhP ve MySQL ile gerçekleştirilmiş bir veritabanı sunucudur Şekil 4.1'de gösterildiği gibi. Sunucu veritabanını içerir, veritabanında tüm kullanıcıların verileri (hastalıkları), üç tablo şeklinde bulunur.

- 1) Kullanıcı tablosu
- 2) Log Bleeding tablosu
- 3) Log Infusion tablosu

Burada tablolar arasındaki bağlantı ilişkisi (1-n) şeklindedir. Her kullanıcıya ait tabloları her tablodaki verileri sınırsız sayıda ekleyebilirsiniz. Burada sunucuya ait çok önemli bir özellik kullanılmıştır, e-posta ile bağlantı:

- 1) Web uygulaması aracılığıyla veritabanına yeni bir kullanıcı eklerken kullanıcıya bir mesaj gönderir, Kayıt sürecine ilişkin bu mesaj kayıtlanma işleminin başarılı olup olmadığını belirtir.
- 2) Eğer şifrenizi unutursanız, sunucu size e-posta aracılığı ile yeni bir şifre gönderecektir.
- 3) Eğer şifreyi değiştirirseniz, Sunucu E-posta ile şifre değiştirme işleminin başarılı olup olmadığını kullanıcıya yeni bir mesaj ile gönderecektir.

4.2.2 Mobil uygulamanın tanıtımı

Mobil hasta takip uygulaması günlük infüzyon bilgilerini ve kanama verilerini kullanarak hemofili hasta / yakınlarına yardım, genel sağlık ve sağlıklı yaşamını sürdürebilmesine yardımcı olmak için tasarlanmıştır. HemoÖzgür günlük geçmiş görüntüler ve kullanılan cihaz aracılığıyla bakım ekibine rapor gönderebilir.

Uygulama, kür tedavisi, spesifik bir hastalık bozukluğu, ya da herhangi bir özel kanama durumunu tanımlayabilir. Kullanıcıların sağlık durumu kontrolü altında tutması ve tedavi süresinde uygulamadaki verileri kullanabilmesi için tasarlanmıştır.

Mobil hasta takip uygulaması aşağıdaki özelliklere sahiptir:

- Tarih, saat, yer ve her infüzyon nedenini kaydetmek.
- Barkod numarasını (üzerinde bulunduğu ilacın üretim bilgileri ve içeriğini gösteren numara) cihazınızın kamerasını kullanarak çekip gerekli bilgilere ulaşmak.
- Fotoğraf, harita ve günlük kanamaları bilmek.
- Sizin ve bakım ekibi için raporlar oluşturmak.
- Hatırlatmalar fonksiyonu ile doktor randevuları ve faktör ihtiyaçlarını takip etmek.
- HemoÖzgür her ay otomatik olarak inceleme yapar ve aynı ayın içinde üç kanama varsa size uyarı verir ve sizden profilaksi tevasi başlanması ister.
- Bunun yanında her altı ayda el, dirsek, ayak ve diz bölgelerinin herhangi birinde kanama olursa da uyarı verir.
- Bir basitleştirilmiş arayüzü ile günlük infüzyon takibi ve kanamaları belirlemek.
- Vücut Kas haritası geliştirilmiş ve pin açılan etiketler ile gezinmek.
- Faktör kaynağı izlemek.
- Çok çeşitli ihtiyaçlar için birden çok tarih seçmenize olanak sağlar
- Dinamik takvim.
- Elektronik tablo veya Elektronik tablo ile birlikte raporları gösterme yeteneği.
- Bakıcı ve hasta modu arasında geçiş yeteneği.

Uygulama ile ařağıdaki örneklerdeki gibi çeřitli hatırlatma ayarları kaydedilerek zamanı geldiğinde kullanıcıyı uyarması sağlanabilir:

- Kan örnek alma ve iğıne vurma gibi işlemlerin alarmı.
- Günlük, haftalık veya tekrarlayıcı bir şekilde ilaç alma zaman alarmı.
- Rapor gönderme alarmı.
- Doktora danışması için alarm.
- Eklemek istediğiniz herhangi bir amaç için alarm.

Uygulama ile acil durumlarda farklı iletişim yolları ile ařağıdaki örneklerdeki gibi aramalar yapılabilir:

- Kişinin adı, telefon numarası ve e- posta bulunmaktadır ve o kişiye kısa mesaj, arama veya e- posta gönderebilirsiniz.
- Rapor listesinin iki yönlü bulunmaktadır. Birisi iğıne vurmaya diğeri de kanama ile ilgilenir. Her biri ayrı bir şekilde güne göre veya aya göre rapor verir.
- Raporu yapıldıktan sonra oluşturulan raporu doktorunuza, sizin durumun ile ilgilenen derneğıe gönderebilme.

4.3 Mobil Hemofili Takip Sistemi Kullanıcı Kılavuzu

Aşağıdaki gibi hemofili programda kullanılan arabirimleri vardır:

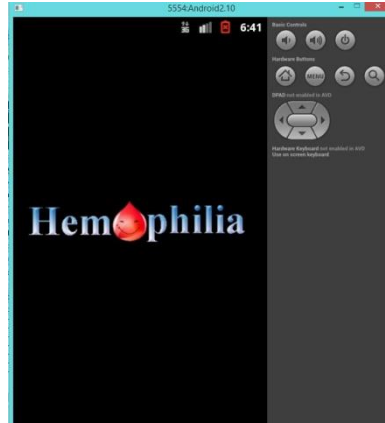
- ❖ Telefon (masaüstünde) ana arayüzü, Programın simgesini gösterir.



Şekil 4.2 Cep telefon masaüstünü arayüzü.

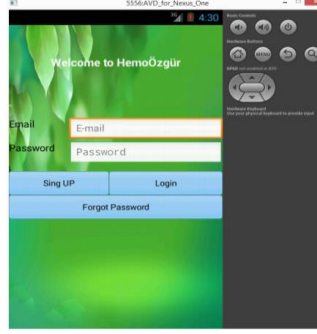
- ❖ (Splash) arayüzü, geçici ekran belirli bir süre için programın uygulanması başında görünecek ve kaybolacaktır

.



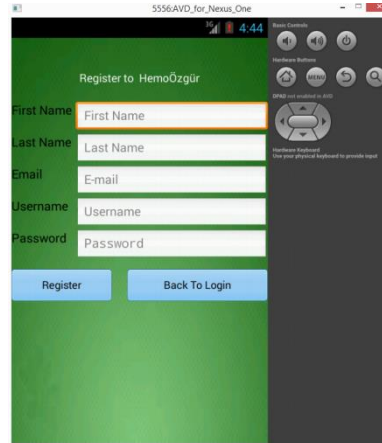
Şekil 4.3 (Splash) arayüzü.

- ❖ Kullanıcı arayüzü Şekil 4.4'de gösterildiği gibi, aşağıdakileri içerir:



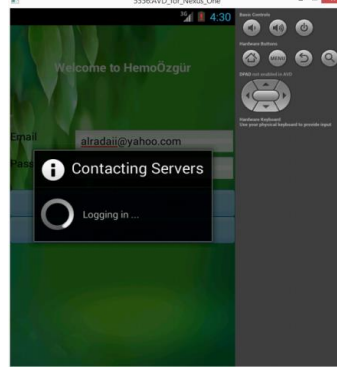
Şekil 4.4 Ana kullanıcı arayüzü.

- ❖ Şekil 4.5'te gösterildiği gibi yeni kullanıcı için program kayıt ve aşağıdakileri içerir:



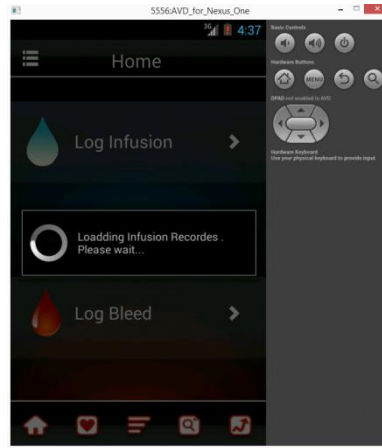
Şekil 4.5 kullanıcının bilgileri girmeyin arayüzü.

- ❖ Giriş düğmesine bastıktan sonra sunucu veritabanına ulaşacak ve Şekil 4.6'da gösterildiği gibi yazılım, girişi onayladıktan sonra verileri cep telefonuna aktaracaktır.



Şekil 4.6 Serverdan kullanıcının bilgileri getirme arayüzü.

- ❖ Uygulama kullanıcı verilerini aldıktan sonra, kullanıcıya ait diğer özel verilerini de getirir. Şekil 4.7 'de gösterildiği gibi, programın resmi internet sitesinde veritabanı tablolarında saklanan veriler.



Şekil 4.7 Serverden bilgiler getirme arayüzü.

- ❖ Kullanıcının şifresini değiştirme arayüzü.



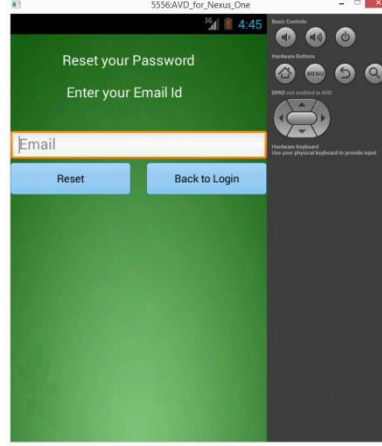
Şekil 4.8 Şifre değiştirme Arayüzü

- ❖ Şifre değiştirme Arayüzü, Yeni bir parola girmek için gereklidir, OK düğmesine düğmesine bastığınızda, Programın sahip olduğu resmi web sitesinden size bir mesaj gönderecektir, Şekil 4.9'da gösterildiği gibi şifre değiştirme işleminin başarılı olup olmadığını size söyler.



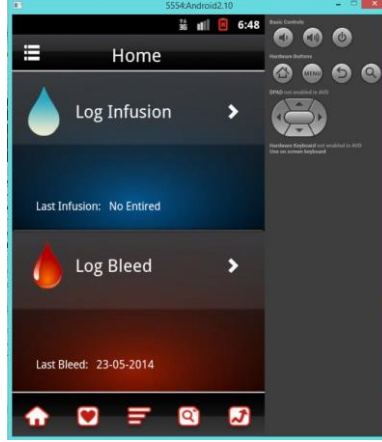
Şekil 4.9 Şifre değiştirme Arayüzü.

- ❖ Eğer şifrenizi unutursanız şifre sıfırlama arayüzü ile tekrardan yeni şifre alma işlemi gerçekleştirebilirsiniz. Programın resmi web sitesi size yeni şifrenizi e-posta aracılığıyla gönderecektir.



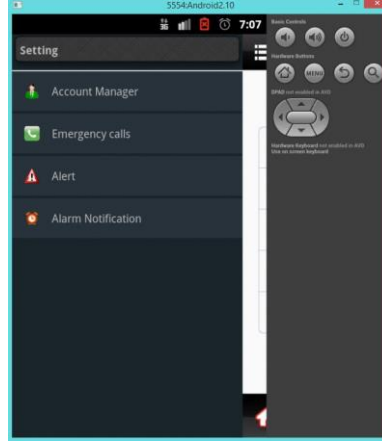
Şekil 4.10 Şifre sıfırlama Arayüzü.

- ❖ Programın ana arayüzü, Şekil 4.11'de gösterildiği gibi aşağıdakileri içerir:



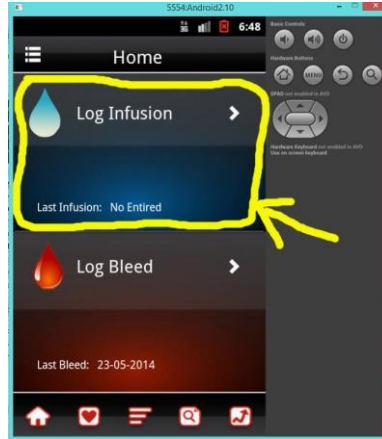
Şekil 4.11 Ana arayüzü

1. Şekil 4.12'de gösterildiği gibi yan menü düğmesi, kontrol düğmelerini gösterir:



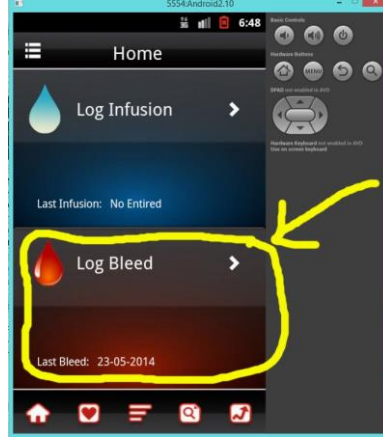
Şekil 4.12 Yan arayüzü.

2. Log İnfasion: Bu kısımda alınan son infüzyon tarihini görebilir ve veri değerlerine ulaşabilirsiniz



Şekil 4.13 Ana log infusion arayüzü.

3. Log Bleed: Bu kısımda alınan son kanama tarihini görebilir ve veri değerlerine ulaşabilirsiniz.



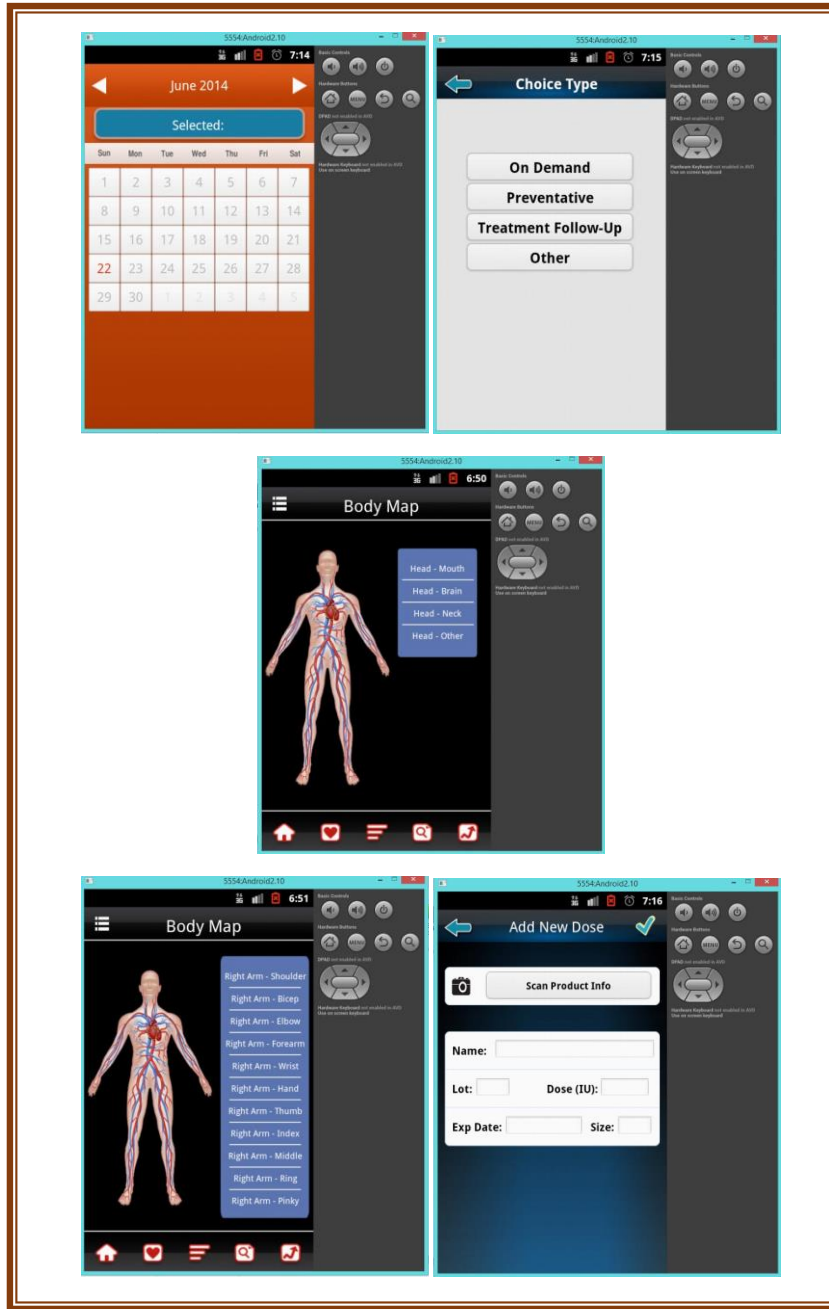
Şekil 4.14 Ana log bleed arayüzü.

❖ Arayüz (Log Infusion), Şekil 4.15'te gösterildiği gibi, aşağıdakileri içerir:

- 1) (+) belirli bir alan içine yeni bir süreç ekleme düğmesi.
- 2) (←) önceki arayüze geri dönme düğmesi.



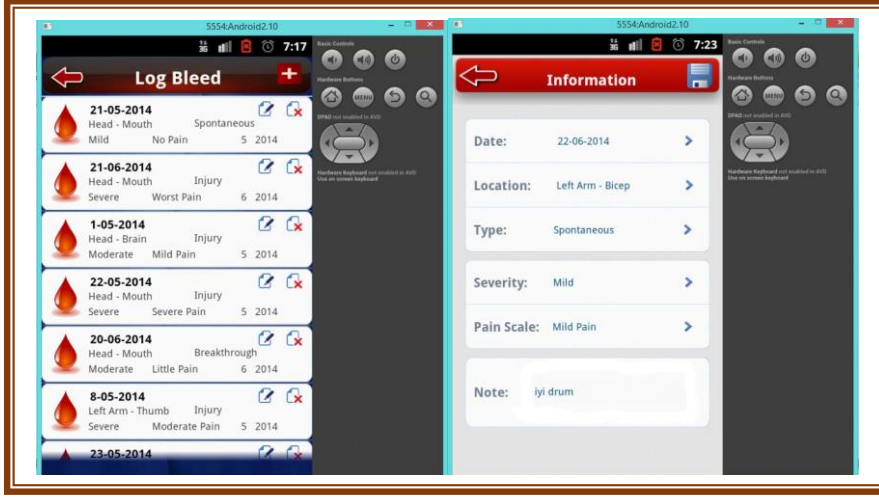
Şekil 4.15 Log Infusion arayüzü



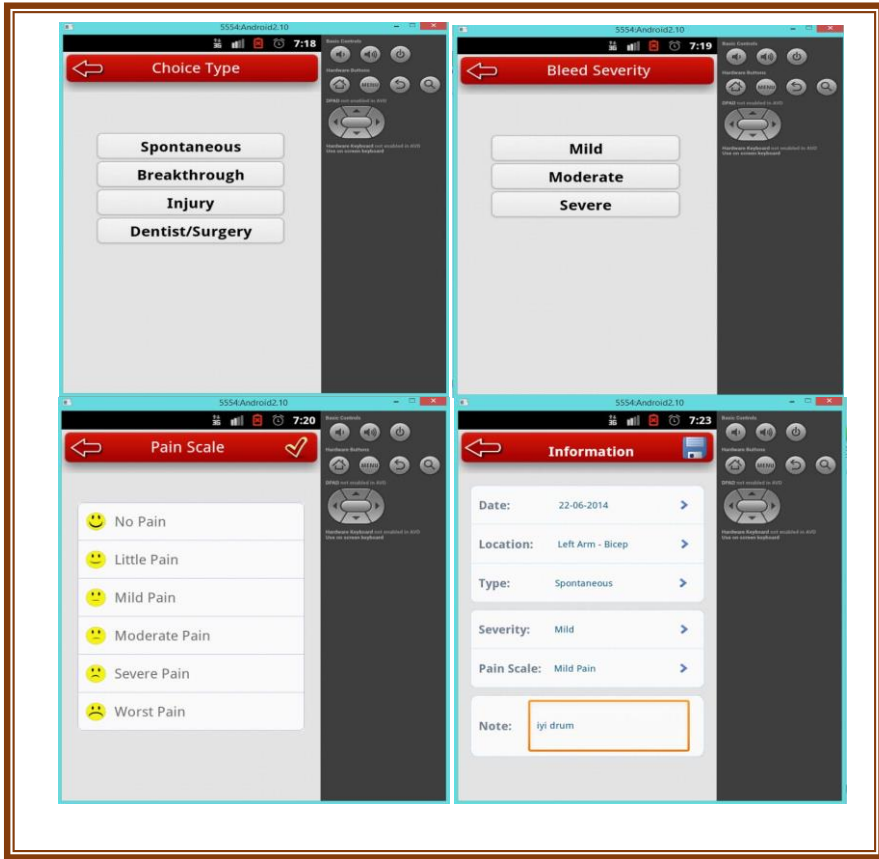
Şekil 4.16 Log Infusion arayüzü.

❖ Log Bleed Arayüzü, aşağıdaki içerir Şekil 4.17'de gösterildiği gibi:

- 1) (+) kan örneği alınması için yeni bir süreç ekleme düğmesi.
- 2) (←) önceki arayüze geri düğmesi.



Şekil 4.17 Log Bleed arayüzü.



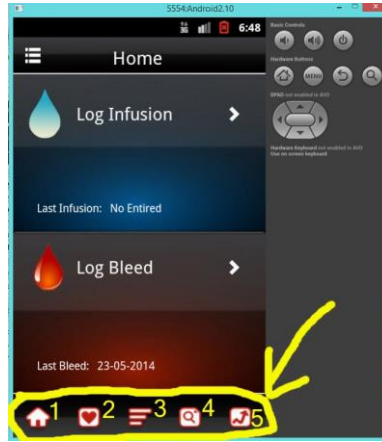
Şekil 4.18 Log Bleed arayüzü.

- ❖ Tablo içindeki iki kısımdan verileri güncelleme veya silme işlem gerçekleştirilebilir, Şekil 4.19 'da gösterildiği gibi.

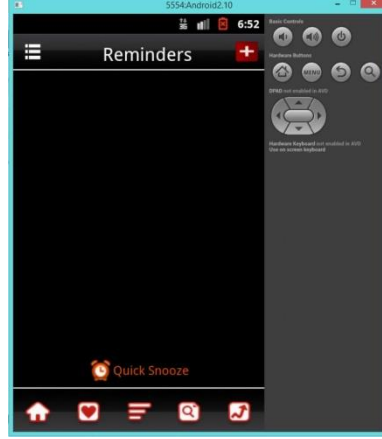


Şekil 4.19 kontrol butonlar listesi.

- ❖ Ana arayüzünde görüldüğü gibi alt kısımdaki 5 tane tuş farklı kısımlara geçme kolaylığı sağlar, Şekil 4.20 'de gösterildiği gibi, her biri belirli bir işleve sahiptir:



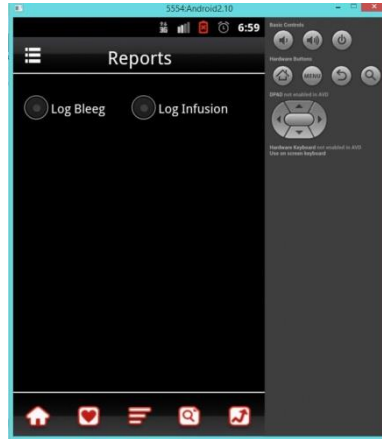
Şekil 4.20 Ana arayüzünü butonları.



Şekil 4.21 Ana Uyarılar arayüzü.

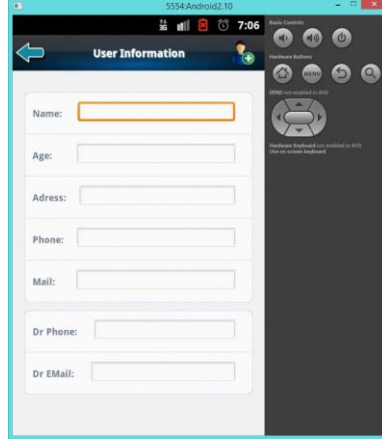
❖ Şekil 4.22 'de gösterildiği gibi, rapor arayüzü:

1. Log Bleed
2. Log Infusion



Şekil 4.22 Ana raporlar arayüzü.

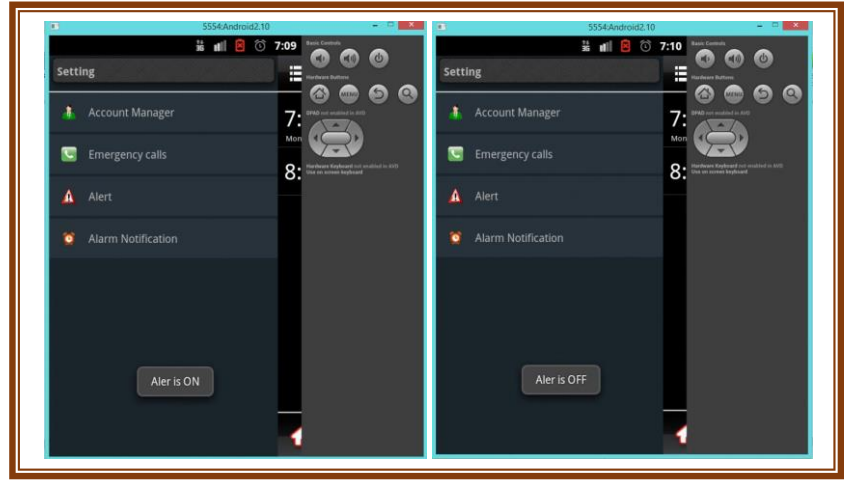
❖ Hesap Yöneticisi arayüzü, aşağıdakileri içerir:



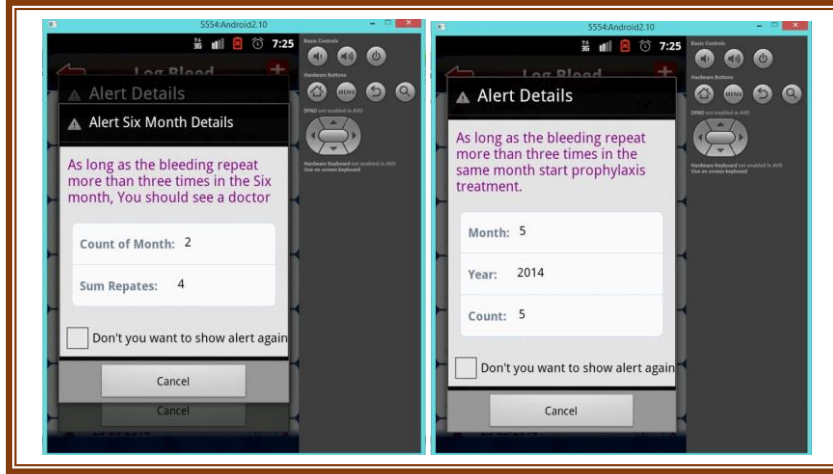
Şekil 4.23 Hesap Yöneticisi arayüzü.

❖ Uyarı Bildirimi arayüzü aşağıdaki içerir:

- Acil uyarı aktivasyonu için uyarı butonunu aktif hale (Alert on) getirmeniz gerekir. Şekil 4.24.1 'de gösterildiği gibi.
- Uyarı aktivasyonunu kapalı hale getirmeniz için (Alert off) yapmanız gerekir. Şekil 4.24.2 'de gösterildiği gibi.

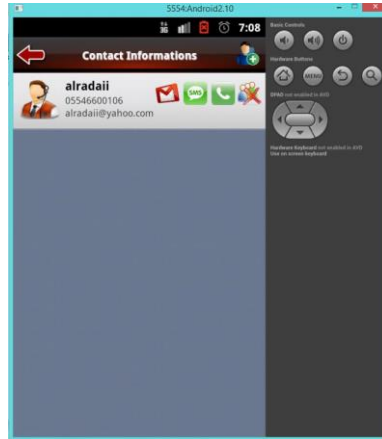


Şekil 4.24 Uyarı Bildirimi arayüzü.



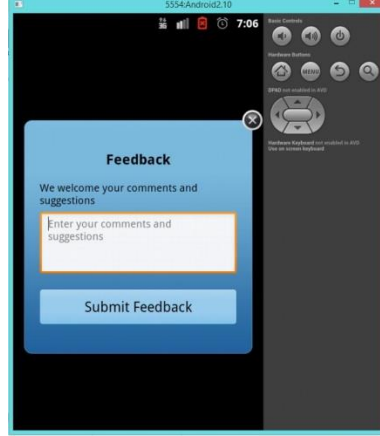
Şekil 4.25 Uyar Bildirimi arayüzü.

❖ Acil Aramalar Arayüzü Şekil 4.26 'da gösterildiği gibi:



Şekil 4.26 Acil Aramalar arayüzü.

- ❖ Feedback arayüzü kullanıcının fark ettiği hataları veya geliştirilmesi gerekli olan alanları bildirebilmesi için eklenen arayüzdür.



Şekil 4.27 Görüş bildirim arayüzü.

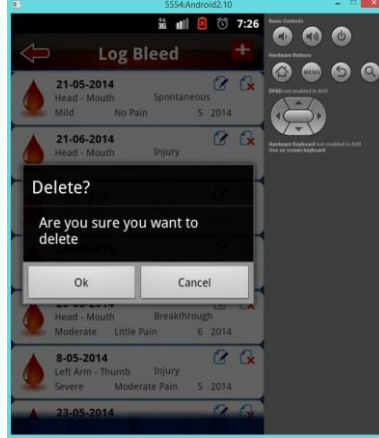
- ❖ Barkod okuma arayüzü, Şekil 4.28 'de gösterildiği gibi:

Bu kısımda Scan Product Info kısmına bastığınız zaman program kameranızı aktif hale getirir. Kamera aracılığıyla okutulan barkod numarası ile ilaca ait tüm bilgilere ulaşabilirsiniz.



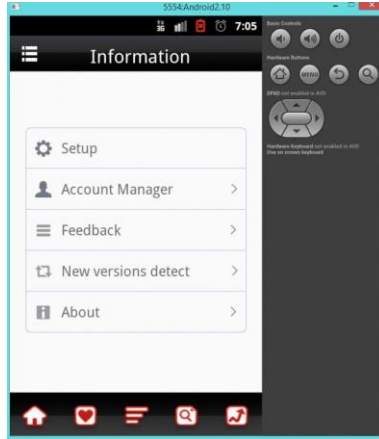
Şekil 4.28 Barkod arayüzü.

- ❖ Kayıt silme arayüzü, Şekil 4.29 'da gösterildiği gibi.



Şekil 4.29 Log Bleed bir kayıt silmesi arayüzü.

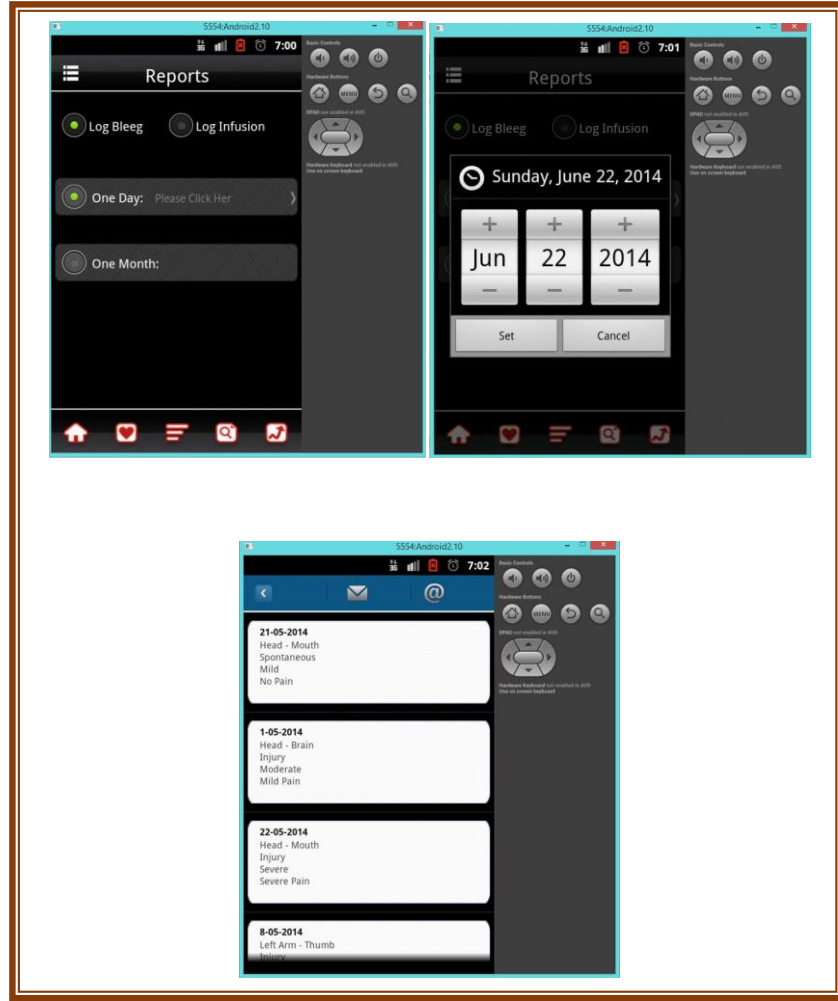
- ❖ Ayarlar arayüzü, Şekil 4.30 'de gösterildiği gibi aşağıdakileri içerir bakın:



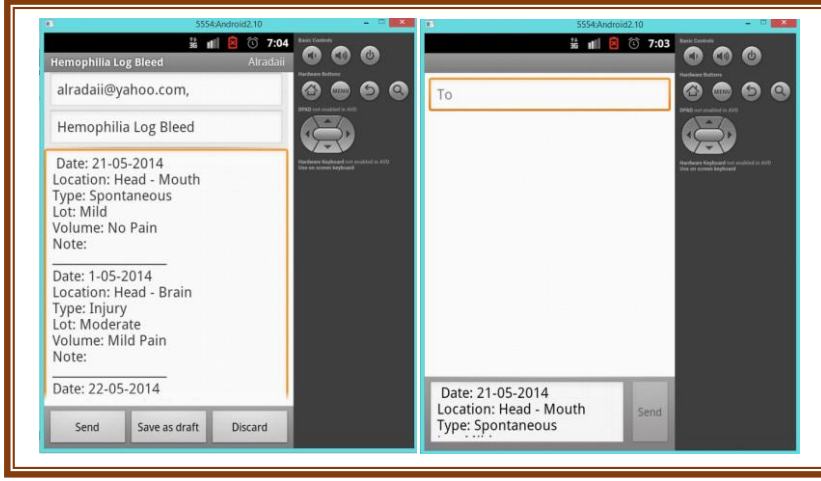
Şekil 4.30 Ayarlar arayüzü.

❖ Raporlar Arayüzü, Şekil 4.31 'de gösterildiği gibi:

1. Gün: One Day kısmına bastığınızda ulaşmak istediğiniz tarihteki verileri sizi yönlendirir.
2. Ay: one month kısmına bastığınız zaman ulaşmak istediğiniz ay içerisindeki tüm verilere sizi yönlendirir.

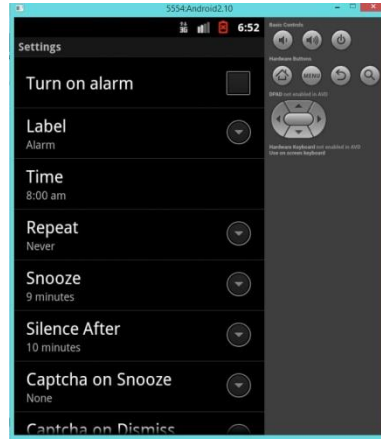


Şekil 4.31 Günlük olarak Rapor göndermesi arayüzü.



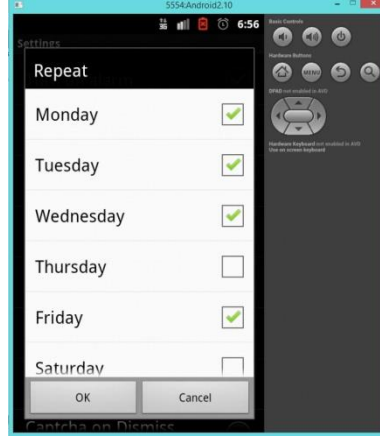
Şekil 4.32 Raporlar gönderme arayüzü

❖ Şekil 4.33 'de gösterildiği gibi, uyarılar arayüzü.



Şekil 4.33: Ana alarm arayüzü ve Alarm ayarları.

- ❖ Alar tekrarının hangi günlerde geçerli olmasını belirleme arayüzü.



Şekil 4.34 Alarm listesi arayüzü.

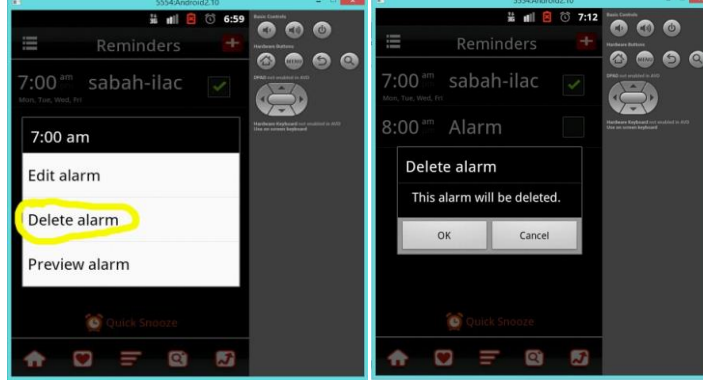
- ❖ Şekil 4.35 'te gösterildiği gibi, seçilen alarm aşağıdakileri içerir:

1. Alarm Adı (Sabah-İlaç).
2. Alarm ve saati, Sabah ya da akşam (07:00) 'de.
3. Alarm saatinin alt kısmında hangi günlerde ilaç alınacağı bilgisi vardır (Pzt, Per, Wen, Cuma).



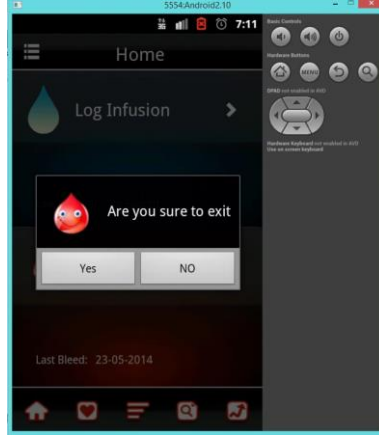
Şekil 4.35 Alarm arayüzü.

- ❖ Şekil 4.36'da gösterildiği gibi, Alarm saatinin üzerine uzun süre basıldığında şekildeki arayüz açılır. Alarmı düzenleme, silme ya da kontrol etme düğmelerine ulaşabilirsiniz.



Şekil 4.36 Alarm ayar arayüzü.

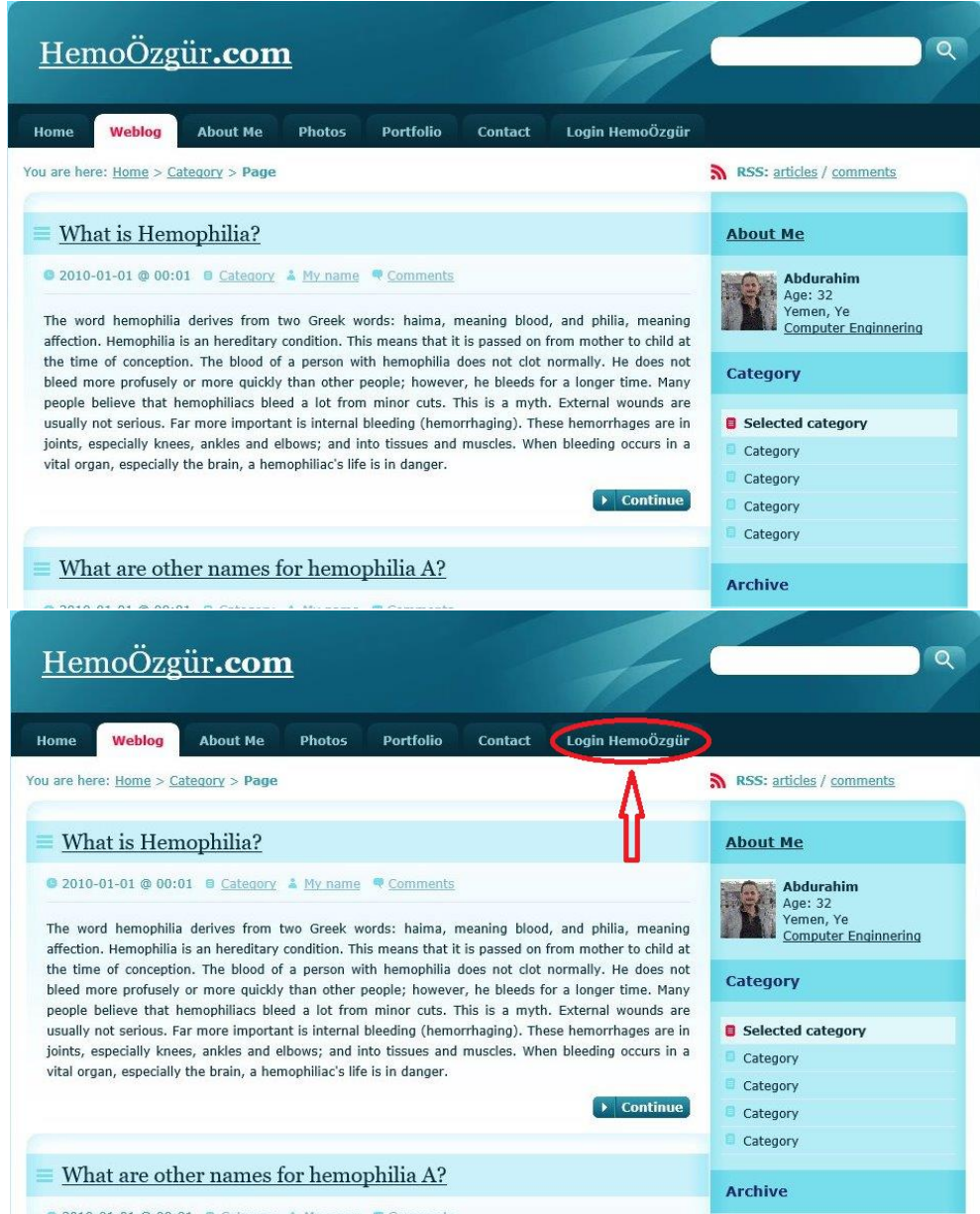
- ❖ Programın kullanıldığı cihazdan geri tuşuna bastığınızda size çıkış işlemi onayı ister.



Şekil 4.37 Çıkış arayüzü.

❖ Hemofili Hastalığı ana web sitesinin arayüzü.

1. Hemofili Hastalığı için gerekli bilgileri içerir, Tanımı, türleri, her tür arasındaki fark ve diğer bilgiler.
2. Hastanın veritabanına erişim arayüzünü içerir, HemoÖzgür programı tarafından alınan kanama verilerini ve örneklerini içerir.



Şekil 4.38 Hemofili website arayüzü.

- ❖ Veritabanına erişimi arayüz, Şekil 4.39 'da gösterildiği gibi.



Please Login

Username: alradaii@yahoo.com

Password:

Login

Şekil 4.39 Hemofili websitede kullanıcı arayüzü.

- ❖ Kullanıcı doktor ise program sayesinde sahip olduğu tüm hastalarının verilerine ulaşabilir.

Message to Patients Logout

Dr: Ozgur Gumus

Patients Details

No.	First Name	Last name	E-mail	Created Date	Details
1	Ali	Naji	ali@yahoo.com	2014-12-31 18:53:50	Go
2	Abdurahim	Alradaii	alradaii@yahoo.com	2015-02-23 18:23:25	Go

Şekil 4.40 Hemofili websitede doktora ait hastalarının bilgi arayüzü.

❖ Hasta verilerinin arayüzü, Şekil 4.41'de gösterildiği gibi: Bu kısımda kırmızı renkli yazılar hastanın bir ay ya da altı ay içerisinde üçten fazla kanaması varsa uyarı şeklide doktoruna gösterir.

1. Kullanıcı adı (hasta).
2. log infüzyon tablo verileri.
3. log bleed tablo verileri.

Ali Naji

Summary

Total Lot	Bleed of Month	Bleed of Six Month
24	Bleeding has repeated more than three times in the same : Month =01 Year =2015	Bleeding has repeated more than three times in the six month : Count of Month =1 Sum Repeats =4

Log Infusion Details

No.	Date	Termenate	Site	Name	Lot	Dose	Exparion
1	2-12-2014	On Demand	Head - Mouth	Abd	2	3	2017
2	14-01-2015	Preventative	Chest/Back-Groin	Trf	10	55	2022
3	18-01-2015	On Demand	Left Arm - Hand	Asf	12	8	2017

Log Bleed Details

No.	Date	Location	Type	Severity	Pain Scale
1	1-01-2015	Left Arm - Wrist	Breakthrough	Severe	Moderate Pain
2	5-01-2015	Left Leg - Ankle	Spontaneous	Mild	Mild Pain
3	9-01-2015	Left Arm - Elbow	Breakthrough	Severe	Moderate Pain
4	13-01-2015	Left Leg - Knee	Breakthrough	Severe	No Pain

Abdurahim Alradaii

Summary

Total Lot	Bleed of Month	Bleed of Six Month
Nothing	Nothing	Nothing

Log Infusion Details

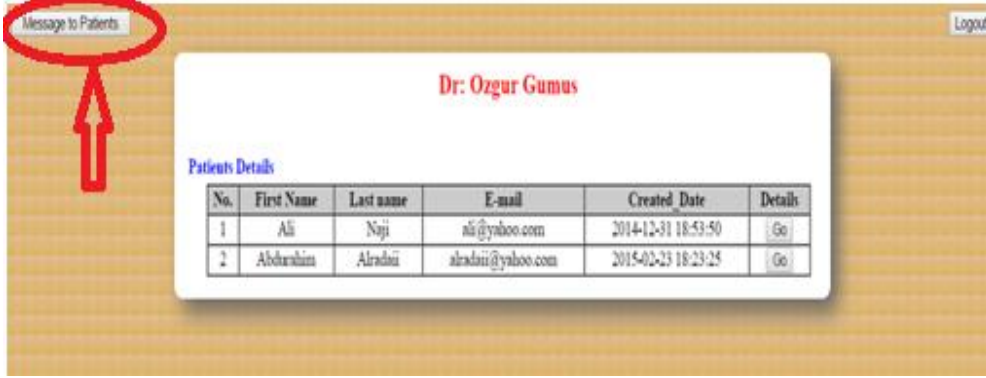
No.	Date	Termenate	Site	Name	Lot	Dose	Exparion
-----	------	-----------	------	------	-----	------	----------

Log Bleed Details

No.	Date	Location	Type	Severity	Pain Scale
-----	------	----------	------	----------	------------

Şekil 4.41 Hemofili websitede doktora ait hastalarının bilgileri arayüzü.

- ❖ Doktor ile hastalar arasındaki iletişim, Şekil 4.42 'de gösterildiği gibi.



Total Devices Registered: 2

Name: Ali

Email: ali@yahoo.com

Type message here

Name: Abdurahim

Email: alradaii@yahoo.com

Type message here

Şekil 4.42 Hemofili websitede sohbet bilgileri arayüzü.

- ❖ HemoÖzgür Programının sohbeti, Şekil 4.43'ta gösterildiği gibi:

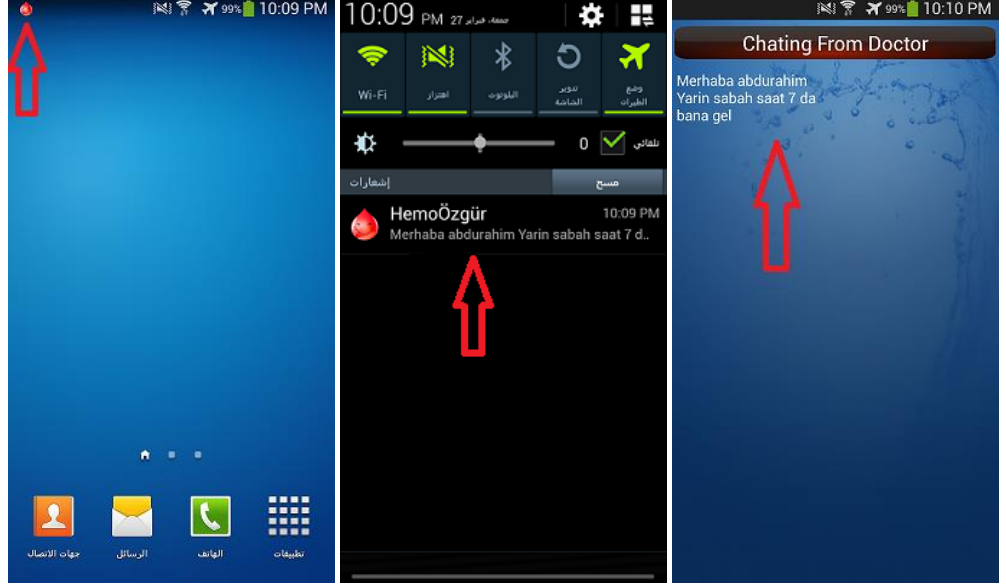
Name: Abdurahim

Email: alradaii@yahoo.com

Merhaba abdurahim
Yarin sabah saat 7 da
bana gel

Şekil 4.43 Hemofili websitede sohbet bilgileri arayüzü.

- ❖ Send butonuna bastığında, Şekil 4.44 ile Şekil 4.46 'da gösterildiği gibi :
Hasta doktorundan gelen bu mesajı programı kullandığı cihaza gelen bildirimle görür. Ve doktoruna bu kısımdan geri dönüş sağlayabilir.



Şekil (4.44 – 4.46) HemoÖzgür programını Sohbet arayüzü.

5. SONUÇ

Servis Odaklı Mimari (SOA) farklı uygulamaların etkileşimine; platformları, uygulama dilleri ve konumlarına bakılmaksızın; uygulama yapı taşı olarak kullanılabilecek jenerik ve güvenilir servislerini kullanarak olanak sağlayan mimari bir tarzdır. SOA, sofistike uygulamalar ve bilgi sistemleri geliştirmek için izlenecek metodolojileri ve stratejileri içerir.

Son yıllarda mobil cihazlar çok yaygınlaşmıştır. Mobil bilişim sistemleri, kullanım amaçları, tasarlanış ve çalışma biçimleri açısından diğer bilişim sistemlerinden farklılık gösterirler. Mobil bilişim sistemlerinin ayırt edici yönleri arasında yaygın kablosuz ağ bağlantıları, küçük boyutları, güç kaynakları ve özellikle mobil kullanıcılara uygun düşen işlevleri gösterilebilir.

Bu tez çalışmasında önce SOA ve mobil bilişim alanında genel bir bakış sağlanmış ve uygulama geliştirmenin farklı yolları açıklanmıştır. Daha sonra kalıtsal ve ömür boyu hastaların yaşam kalitesini etkileyen bir hastalık olan hemofili hastalığı tanıtılmıştır. Hemofili hastalığı hastanın 24 saat yaşadığı ve hastanın başına gelen olayların ve kendi kendine uyguladığı tedavilerin hastane dışında da izlenmesini gerektiren kronik bir hastalıktır. Bu amaçla hemofili hastalarının hastane dışında takibini sağlamak için kullanılacak bir mobil hasta takip sistemi geliştirilmiştir. Bu sistem ile her türlü kanama ve tedavi verisi kayıt altına alınabilmekte, önceden kuralları kaydedilen durumlar için zamanı geldiğinde hasta uyarılabilmekte, acil durumlarda doktor ile iletişime geçilebilmekte ve çeşitli zaman dilimlerini kapsayan raporlar alınabilmektedir.

KAYNAKLAR DİZİNİ

Ali Arsanjani, 2005, How to Identify, Specify and Realize Services for your SOA (Part II and III), IBM,

http://www.ebizq.net/topics/dev_tools/features/5632.html.

Bass L., Clements P., Kazman R., 1997, Software Architecture in Practice, Addison-Wesley.

Booch G., 1990, Object Oriented Design with Applications, Benjamin-Cummings Publishing.

Dirk Slama, Karl Banke, Dirk Krafzig, Dec. 10, 2004, Service Oriented Architecture: Inventory of Distributed Computing Concepts, Prentice Hall PTR.

ED. Yourdon, Constantine L., 1975, Structured Design, Prentice Hall.

James McGovern, Sameer Tyagi, Michael E. Stevens, Sunil Mathew, 2003, Java WebServices Architecture, Morgan Kaufmann Publishers.

Hashem M., Ghoniemy S., Hamdy M., April 22-26, 2003, “A Quality-Of-Service-Aware Genetic Algorithm for the Source Routing in Ad-Hoc Mobile Networks”, ICEIS 2003, Proceedings of the 5th International Conference on Enterprise Information Systems, Angers, France.

Koushanfar F., Prabhu V., Potkonjak M., and J. M. Rabaey., 2008, Processors for mobile applications. In in Proc. 2000 IEEE Int. Conf. Computer Design: VLSI in Computers and Processors, pages 603–608.

Lionel M. Ni Pei Zheng., 2006, Smart Phone Next Generation Mobile Computing, Morgan Kaufmann Publishers, 500 Sansome Street, Suite 400, San Francisco, CA.

KAYNAKLAR DİZİNİ (devam)

Microsoft Corporation, Component Object Model (COM),
<http://www.microsoft.com/com/default.mspx>, 2006.

Microsoft Corporation, November 1996, DCOM Technical Overview, .

Object Management Group, March 2004, Common Object Request Broker
 Architecture: Core Specification, Version 3.0.3,
http://www.omg.org/technology/documents/corba_spec_catalog.htm.

Raphael Malveau, Thomas J., Mowbray, Apr. 16, 2004, Software Architecture:
 Basic Training, Prentice Hall PTR.

Reza B'Far., 2005, Mobile Computing Principles: Designing and Developing
 Mobile Applications with UML and XML. Cambridge University Press.

Sara Williams, Charlie Kindel, October 1994, The Component Object Model: A
 Technical Overview, Microsoft Corporation.

Satyanarayanan M., August 2001, Pervasive Computing: Vision and Challenges.
 IEEE Personal Communications.

Spince N., 6 Sep 2000, "Respiratory symptoms: Associations with pesticides,
 silos, and animal confinement in the Iowa Farm Family Health and Hazard
 Surveillance Project," American Journal of Industrial Medicine, vol. 38, p.
 American Journal of Industrial Medicine.

KAYNAKLAR DİZİNİ (devam)

Sun Microsystems, Jini Network Technology, <http://www.sun.com/software/jini/>, 2002.

Sun Microsystems, Java 2 Platform Enterprise Edition Specification, Version 5.0 <http://java.sun.com/j2ee/>, 2006.

Suoranta R., 2006, New Directions in Mobile Device Architectures. In in Proc. Of the 9th Euromicro Conference on Digital System Design (DSD'06), 2006.

Szyperski C., 1998, Component Software: Beyond Object-Oriented Programming, Addison-Wesley.

Vourc'h G., et al., 12, February 2006, "Detecting Emerging Diseases in Farm Animals through Clinical Observations," Emerg Infect Diseases, vol.

Weiser M., September 1991, The Computer of the 21st Century. Scientific American.

World Wide Web Consortium (W3C), 2003, SOAP Specification, Version 1.2.

World Wide Web Consortium (W3C), February 2004, Extensible Markup Language (XML) Specification, Version 1.0, 04, <http://www.w3.org/TR/REC-xml/>.

World Wide Web Consortium (W3C), 11 February 2004, Web Services Architecture, <http://www.w3.org/TR/ws-arch/>.