

**GPS KULLANAN WEB SERVİS TABANLI UYARLANABİLİR
MOBİL UYGULAMA**

Hanife Hilal KAYA

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

OCAK 2010

ANKARA

Hanife Hilal Kaya tarafından hazırlanan GPS KULLANAN WEB SERVİS TABANLI UYARLANABİLİR MOBİL UYGULAMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. M. Ali AKCAYOL

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. M. Sezai DİNÇER

Elektrik Elektronik Mühendisliği, Gazi Üniversitesi

Prof. Dr. Hadi GÖKÇEN

Endüstri Mühendisliği, Gazi Üniversitesi

Doç. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği, Gazi Üniversitesi

Tarih: 26/02/2010

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans / Doktora derecesini onamıştır.

Prof. Dr. Bilal TOKLU

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Hanife Hilal KAYA

GPS KULLANAN WEB SERVİS TABANLI UYARLANABİLİR MOBİL UYGULAMA

(Yüksek Lisans Tezi)

Hanife Hilal KAYA

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ocak 2010

ÖZET

Bu tezde, GPS kullanan ve bağlam farkındalığı olan, kullanıcının ilgilendiği servisleri bağlam bilgisine ve profil bilgisine göre üreten web servis tabanlı bir sistem geliştirilmiştir. Sistem aynı zamanda kullanıcının yaptığı tercihler göre profili güncellemektedir. Bu sistemin mobil ara yüzü kullanıcıdan bilgi girişi istemeden bağlam bilgilerini merkezi uygulamaya göndermektedir. Merkezi uygulama kayıtlı etkinlikler arasından bir değerlendirme yaptıktan sonra mobil uygulamaya yanıtı gönderir. Mobil uygulama aracılığıyla tercihler yapılır. Yapılan tercihler kullanıcı profilini güncellemek üzere kaydedilmektedir.

Güncellenen profil sayesinde kullanıcının ilgilenebileceği etkinlikler otomatik olarak belirlenmiş olur. Zaman harcayan, birçok servis arasından yapılacak seçimler, konum bilgisine göre gönderilen seçenekler arasından yapılır. Böylece mobil cihaza mobil kullanıcının ilgisini çekmeyecek bilgi iletilmemiş olur. Zaman tüketimi, mobil cihaz kaynaklarının tüketimi ve maliyetler azalacaktır. Bunların yanı sıra, profilde bulunan sağlıkla ilgili etiketler aracılığıyla hastalık gibi acil durumlarda hızlı sonuç üretilmesi faydalı olacaktır.

Bilim Kodu : 902.1.014
Anahtar Kelimeler : Bağlam bilinçli servisler, profil uyarlama, mobil servisler
Sayfa Adedi : 75
Tez Yöneticisi : Doç. Dr. M. Ali AKCAYOL

**WEB SERVICE BASED ADAPTABLE CONTEXT-AWARE MOBILE
APPLICATION USING GPS**

(M.Sc. Thesis)

Hanife Hilal KAYA

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

January 2010

ABSTRACT

In this thesis, a context-aware, web service and GPS based mobile system generating information services responses according to profile and context data, has been developed. System updates profile of users according to their service preferences. This system sends context data to the server application with no entry from the user. The server application replies using some evaluation results of recorded actions. Selected actions have been saved to update profile of the user.

By using this application, information services results are generated according to profile and context data without any effort from the user. The actions which user may be interested are automatically updated using profile of the user. Time consuming selections of list of services, becomes easier by the usage of this system. The mobile user receives only required and interested information about the services. The application saves time, resources of mobile machines and costs. In addition, generating service results according to attributes about health is an important issue for emergency situations.

Science Code : 902.1.014

Key Words : Context-aware services, profile adaptation, mobile services

Page Number : 75

Adviser : Assoc. Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Çalışmam süresince bana yol gösteren hocam Doç. Dr. M. Ali AKCAYOL' a değerli fikirleri ve yardımlarından dolayı teşekkür ederim.

Ayrıca, desteğini hiçbir zaman eksik etmeyen aileme, beni cesaretlendiren, her zaman anlayışlı ve yardımcı olan eşim Ahmet DEMİRTAŞ' a, tüm destekleri için Hilal ve Ülkü KALBAN' a ve deneyimleri, bilgisi ve yardımlarıyla beni yalnız bırakmayan arkadaşım Ezgi MERT' e katkılarından dolayı çok teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ŞEKİLLERİN LİSTESİ	ix
RESİMLERİN LİSTESİ	x
1. GİRİŞ	1
2. WEB.....	3
2.1. Statik Web.....	4
2.2. Dinamik Web	6
3. WEB SERVİSLERİ	9
3.1. Web Servis Altyapısında Kullanılan Teknolojiler	12
3.1.1. Genişletilebilir işaretleme dili (XML).....	13
3.1.2. Uzaktan yordam çağırma (XML - RPC)	13
3.1.3. Basit nesne erişim protokolü (SOAP)	15
3.1.4. Web servis tanımlama dili (WSDL).....	17
3.1.5. Evrensel kayıtçı (UDDI)	20
3.2. Servis Tabanlı Mimari (SOA).....	21
4. DAĞITIK SİSTEMLER	24
4.1. Dağıtık Sistem ve Dağıtık Programlama	25
4.2. Dağıtık Sistemlerin Özellikleri	27
4.3. Dağıtık Sistemlerin Mimarileri	28

Sayfa

5. BAĞLAM BİLİNÇLİ UYGULAMALAR	30
5.1. Bağlam	31
5.2. Bağlam Farkındalığı.....	33
5.3 Mobil Uygulamalarda Bağlam Farkındalığı	34
6. PROFİL UYARLANABİLİR UYGULAMALAR.....	36
7. GPS KULLANAN WEB SERVİS TABANLI UYARLANABİLİR MOBİL UYGULAMA	39
7.1. Sistem Mimarisi	41
7.1.1. Web istemci mimarisi.....	42
7.1.2. Mobil istemci mimarisi	42
7.1.3. Sunucu mimarisi.....	43
7.2. Sistemin Modülleri.....	44
7.2.1. Sunucu modülü.....	44
7.2.2. Mobil istemci modülü	48
7.2.3. Web istemci modülü.....	62
7.2.4. Veritabanı modülü	67
8. SONUÇ VE ÖNERİLER	70
KAYNAKLAR	72
ÖZGEÇMİŞ	75

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Web sitelerinin artış grafiği	4
Şekil 2.2. Statik web’ de akış	5
Şekil 2.3. Dinamik web’ in çalışma döngüsü.....	8
Şekil 3.1. Web servislerinin genel gösterimi	10
Şekil 3.2. Web servislerinin mimarisindeki rollerin birbiriyle ilişkisi.....	11
Şekil 3.3. Web servis protokol yığını.....	12
Şekil 3.4. Web servis teknolojileri ve ilişkileri	21
Şekil 3.5. SOA tanımındaki web servis birimleri	22
Şekil 4.5. Dağıtık sistemlerin genel görünümü.....	25
Şekil 5.1. Geleneksel bilgisayar uygulamaları veri akışı	30
Şekil 5.2. Bağlam bilinçli uygulamalar.....	31
Şekil 5.3. Bağlam bilinçli uygulamalarda kullanılabilecek parametreler	35
Şekil 7.1. Sistem mimarisinin genel şekli	41
Şekil 7.2. Web istemci mimarisi	42
Şekil 7.3. Mobil istemci mimarisi	43
Şekil 7.4. Sunucu mimarisi	43

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 7.1. Eclipse, sunucu modülünün geliştirildiği ortam	45
Resim 7.2. EclipseME, mobil istemci modülünün geliştirildiği ortam.....	49
Resim 7.3. J2ME emülatör ayarları	53
Resim 7.4. ACMIS uygulaması karşılama ekranı	55
Resim 7.5. ACMIS uygulaması başlangıç ekranı	56
Resim 7.6. GPS cihaz ayarları ve profil ekranı	57
Resim 7.7. GPS cihazının bulunması ve ayarlanması ekranı	57
Resim 7.8. Bilgi servisleri hizmetine giriş	58
Resim 7.9. Ağa bağlanma uyarı ekranı	59
Resim 7.10. Bağlam bilinçli etkinlik sonuçları	60
Resim 7.11. Etkinlik detay ekranı	61
Resim 7.12. Güncellenen profile göre yeniden istenen bilgi servisinin sonucu	62
Resim 7.13. Web istemci modülü kullanıcı ara yüzleri giriş ekranı	64
Resim 7.14. Web istemci modülünün yönetici ekranı	65
Resim 7.15. Web istemci modülünün kullanıcı ekranı	66
Resim 7.16. Web istemci kullanıcı ekranı, profil düzenleme sayfası	67
Resim 7.17. MySQL erişim arabirimi.....	68

1. GİRİŞ

Web, 1991 yılında internet üzerinde bir servis olarak çalışmaya başladığından beri hızla gelişmiş, kişiler ve kurumlar için önemi giderek artmıştır. Web teknolojilerinin gelişmesi beraberinde web’ de sunulan uygulama sayısının artmasını sağlamış, bu aşamada ise uygulamaların birlikte çalışması ihtiyacını doğurmuştur. Fakat platform çeşitliliği sebebiyle platformdan bağımsız veri alış verişi yapabilecek bir dil ihtiyacı doğmuştur. Bu aşamada XML’ in [1] duyurulması ile birlikte farklı sistemlerin haberleşmesi için yapılan çalışmalar hızlanmıştır.

XML tabanlı mesajlaşma protokolü olan SOAP’ un [2] geliştirilmesiyle birlikte web ortamındaki uygulamaların haberleşmesi ihtiyacını giderecek bir çözüm üretilmiştir. Ardından servis özelliklerini belirten bir dil olan WSDL ve evrensel kayıtçı olan UDDI’ nin [1] tanımlanmasıyla birlikte web servisleri geliştirilmeye başlanmış ve nihayet web üzerindeki uygulamalar birer servis olarak birlikte çalışabilir hale gelmişlerdir.

Web teknolojilerinin gelişmesi ve mobil cihazların internet erişimine açık hale gelmesi, bu cihazlarda çalışabilecek web uygulamaları geliştirilmesini de sağlamıştır. Mobil cihazların gün geçtikçe daha yüksek teknolojiye üretilmesi ve çeşitlenmesi de aynı zamanda yeni marka ve modellerde mobil cihazların çıkmasına yol açıp, web teknolojilerinin mobil ortamlar için de gelişmesine ve sayı olarak artmasına yardımcı olmuştur. Web’ in mobil cihazdan kullanılması durumunda kişisel bilgisayarlardan kullanılmasından farklı olarak değişken çevresel şartlar mevcuttur. Bunların en başında mobil cihazın bulunduğu yer, çevresel etkenler, ve ayrıca zaman, kişisel bilgiler gibi dinamik veriler yer almaktadır.

Yer bilgisi öncelikle gezi uygulamaları, turistik amaçlı uygulamalarda yaygın olarak kullanılmıştır, fakat günümüzde farklı amaçlara hizmet eden birçok uygulamada kullanılmaktadır. Bu uygulamaların başında coğrafi bilgi sistemleri, turistik amaçlı sistemler, lojistikte kullanılan sistemler gelmektedir [3]. Bu tür uygulamalarda yer bilgisini elde edebilmek için iki farklı yöntemden faydalanılabilir. Bunlar; “Küresel

Konumlama Sistemi” (Global Positioning System veya kısaca GPS) ve “GSM Hücre Bilgisi ile Yer Belirleme Sistemidir”. Bir mobil cihazda GSM Hücre Bilgisi ile yer belirlendiğinde, cihazın hali hazırda bağlı bulunduğu hücresel sistemlerin özelliklerinden faydalanılır. Ancak küresel konumlama sistemi ile yer belirleme sisteminde dahili veya harici GPS cihazından faydalanılır.

GPS verisini kullanarak mobil kullanıcılar için geliştirilmiş bağlam bilincine sahip uygulamalardan biri Karademirci tarafından geliştirilen MARIS isimli uygulamadır. Ancak uygulamada adaptif bir profil bulunmayıp, sunulan servislerde sadece konumla ilgili bağlam bilgileri kullanılmıştır [4].

Bu çalışmada mobil cihaz üzerinden çalışabilecek bağlam bilinçli ve GPS tabanlı, kullanıcı profiline adapte edilebilen bir uygulama geliştirilmiştir. Uygulamaya kısaca ACMIS (Adaptable Context-Aware Mobile Information Services) adı verilmiştir. Uygulamanın temel akışları şu şekildedir; mobil cihaz sunucudan istemde bulunmadan önce küresel konum bilgisini harici bir cihazdan edinir. Bu bilgi ayrıştırılarak sunucu sistemine profil bilgisiyle birlikte gönderilir. Sunucu sistem ise yer bilgilerine ve profil bilgilerine göre var olan etkinlikler üzerinden bir sorgu yapıp mobil cihaz arabirimine bir sonuç döndürür. Mobil uygulama aracılığıyla oluşturulan kullanıcı profili, kullanıcının tercihlerini, hangi servisle ilgili ne kadar bilgi almak istediğini içerecek şekilde yapılandırılmıştır. Ayrıca kullanıcının yaptığı tercihlere göre kendini güncelleyip, kullanıcının tercihlerini sistemdeki davranışlarına göre anlamlandıracak bir mekanizma oluşturulmuştur.

Anlatımda, ikinci bölümde web teknolojisinin gelişimi incelenmiştir. Web servislerinin temel prensipleri, gelişimi ve bu teknolojinin temelleri olan kavramlardan üçüncü bölümde bahsedilmiştir. Dördüncü bölümde dağıtık sistemlerin özellikleri ve kullanmanın faydalarından bahsedilmiştir. Beşinci bölümde ise bağlam, bağlam farkındalığı kavramları incelenmiş ve mobil uygulamalardaki faydalarından bahsedilmiştir. Altıncı bölümde profil uyarlama ve uygulamalarından bahsedilmiştir. Yedinci bölümde ise tez çalışmasında geliştirilen uygulamadan bahsedilmiştir.

2. WEB

Dünya Çapında Ağ (World Wide Web ya da kısaca w3, www veya web) [5] internet üzerinde çalışan ve www ile başlayan birbirine bağlı sayfaların görüntülenmesini sağlayan bir servistir. İnternet adı verilen; dünya üzerindeki ağların ve bilgisayar sistemlerinin birbirine bağlanması donanımsal ve yazılımsal bileşenler aracılığıyla olurken, web bu geniş ağın en üst katmanında çalışan soyutlanmış genel servisler [5] kümesidir.

Web ve internet kavramları günlük hayatta karıştırılmasına rağmen birbiriyle ilişkili ancak farklı kavramlardır. İnternet dünya çapındaki ağları ve kurumsal bilgisayar sistemlerinin birbirine bağlayan bir elektronik iletişim ağı iken web internet üzerinde çalışan bir servistir.

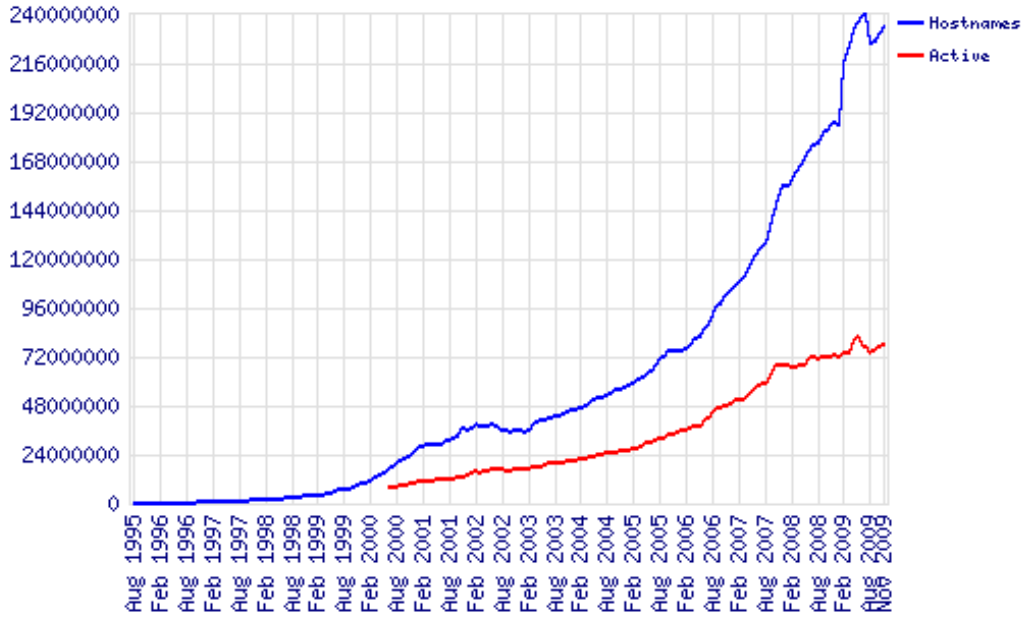
1989 yılında Tim Berners-Lee tarafından öne sürülen web' in gelişimi hızla devam etmiştir. 1990 yılında ilk web sunucusu ve ilk tarayıcı uygulamayı, projenin kendisini anlattığı web sayfalarından oluşacak şekilde geliştirmiştir. 6 Ağustos 1991' de ise projenin bir özetini bir haber grubuyla paylaşmıştır. Bu tarih web' in internet üzerinde bir servis olarak hizmet vermeye başlamasının başlangıcı olarak bilinir [5]. Ekim 1994'de yine Tim Berners-Lee başkanlığında kurulan Dünya Çapında Ağ Birliği (World Wide Web Consortium ya da kısaca W3C) [6] ile gelişmesi hızlanmıştır. W3C web' in standartlarını belirleyen organizasyondur.

İlk üretilen web sayfası Zengin Metin İşaretleme Dili (Hyper Text Markup Language veya kısaca HTML) [5] ile üretilmiştir. HTML bir belgenin internet tarayıcılar üzerinde yorumlanabilmesi için geliştirilmiştir. HTML aslında bir programlama dili değildir, HTML kullanarak tek başına çalıştırılabilecek yazılımlar üretilemez. Bu dil çeşitli biçimlerdeki verileri birbirine bağlar ve ancak internet tarayıcı yazılımlar tarafından yorumlanabilir.

1991 yılından beri hızlı bir şekilde büyümüş ve dünyada kullanımı yaygınlaşmıştır. 2009 yılı Kasım ayı itibarıyla internet istatistikleriyle ilgili çalışan site Netcraft.com'

un verdiği istatistiklere göre 233 milyona yakın web sayfası bulunmaktadır [7]. Şekil 2.1’ de verilen grafik Ağustos 1995 ve Kasım 2009’ u kapsamaktadır. Bu sayfalardan yaklaşık 50 milyonunu ise Kasım 2008 ve Kasım 2009 tarihleri arasında aktif hale gelmiştir. Ekim 2009 ve Kasım 2009 tarihleri arasındaki bir aylık periyotta ise 3 milyon web sayfası kullanılır hale gelmiştir.

Web’ in ilk ortaya çıkış amacı bilgi paylaşımıyken bugün günlük hayatın birçok aşamasında faydalı ve gerekli bir hale gelmiştir. 1991 yılından bugüne devam eden bu süreçte kullanım artışı ile birlikte teknolojik yeniliklerin geliştirilmesi web’ in gelişimini daha ivmeli hale getirmiştir. Diğer açıdan bakıldığında, günlük hayata faydalı projeler geliştirilmesi gerektiğçe yeni teknolojilere ihtiyaç da artmaktadır. Bu amaçla web kullanılmaya başlandığından beri kaydedilen gelişmeyi statik ve dinamik web olarak ikiye ayırmak mümkündür.



Şekil 2.1. Web sitelerinin artış grafiği

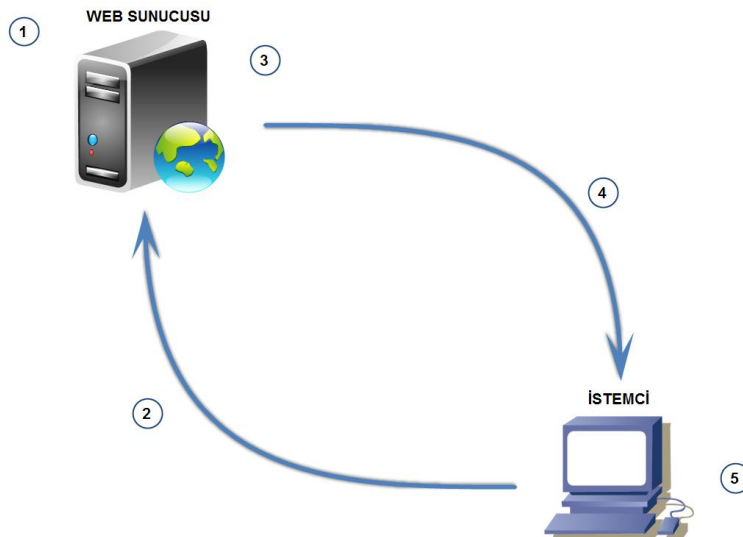
2.1. Statik Web

Web’ in HTML kullanarak kodlanmış sayfalardan oluşan ilk versiyonuna Statik Web adı verilir. Tim Berners-Lee tarafından üretilen ilk web örnekleri gibi HTML

kullanarak üretilen web sayfaları ilk zamanlar ihtiyaçları karşılamış ve birçok kişi ve kurumun bilgi paylaşım aracı olmuştur.

Statik Web' in çalışma prensipleri istemci mimari yapısına dayanır. İstemci tarafında web tarayıcısı adı verilen araçlar çalışır ve sunucudan gelen sayfaları görüntüler. Sunucu tarafında ise web sunucusu adı verilen yazılımlardan oluşan makineler mevcuttur. Bunlar; HTML dilinde kodlanmış dosyaları barındıran cihazlardır. Statik web' de bir web sayfası görüntüleme isteminin aşamaları şekilde de numaralarla ifade edildiği gibi şöyledir:

1. Statik yazılan web sayfası web sunucu üzerinden herkesin kullanımına açılır.
2. Kullanıcı web tarayıcısına görüntülemek istediği sayfanın adresini girer.
3. Web tarayıcı ilgili sunucudan ilgili sayfa için bir HTTP isteminde bulunur.
4. Sunucu isteği inceler ve istemde bulunulan sayfayı HTTP aracılığıyla geçirmek için hazırlar.
5. HTML ile kodlanmış dosya web tarayıcısına ulaştığı zaman tarayıcı dosyanın içeriğini çözerek kullanıcıya gösterir.



Şekil 2.2. Statik web' de akış

Statik web’ de bulunan bir sayfa her defasında farklı kullanıcılara, farklı zamanlarda ve farklı durumlara aynı sonucu döndürmektedir. Her kullanıcıdan gelen her istek sunucudan sayfanın aynı versiyonunu indirmiş olur. Sayfalar web sunucusuna koyulduktan sonra kullanıma hazır olarak bulunurlar ve bir istek geldiği zaman aynı sayfa bir dosya transferine benzer şekilde kullanıcıya gönderilir.

Statik yapıda üretilen sayfaların oluşturulması kolaydır. Bir sitenin modelini oluşturmak, nasıl olduğunu görmek amacıyla kullanılabilir. Aynı zamanda da önbellekte saklanarak aynı sayfa yeniden yüklenmeden birçok kişi tarafından erişilebilir. İçeriği değişmeyen veya uzun süreçlerde değişen sayfaların bu şekilde saklanması uygun olabilir, ancak güncelleme gerektiği zaman kolayca yapılabilir olması gerekir.

Üretilen sayfalar otomatik güncellenemediği için sayfayı ziyaret eden bir kişinin tekrar ziyaret etmesi ihtiyacı kalmaz, ayrıca ziyaretçilerin herhangi bir değişiklik yapma veya katkıda bulunma şansı, sayfayı kişiselleştirme ihtimali bulunmamaktadır. Sayfa içeriği ise herhangi bir müdahale olmadan değiştirilememektedir ve site büyüdükçe sayfaların içeriğinin güncellenmesi amacıyla müdahale edilmesi oldukça zaman kaybettiren bir iştir. Aynı zamanda bu türde yaratılan sayfaların içeriğinin güncel tutulması da takip gerektirir. Bu gibi zorlukları çözebilmek için dinamik web fikri ortaya çıkmıştır.

2.2. Dinamik Web

Statik web’ in kullanımının yaygınlaşması sonucunda web teknolojilerinde değişiklik ihtiyacı doğmuştur. Statik web’ de yaşanan sorunlardan dolayı içeriği dinamik olarak güncellenebilen sayfalar oluşturma amacıyla dinamik web fikri ortaya çıkmıştır. Statik web sayfalarındaki dosya içeriğinin web sunucuda saklanması ve her gelen istekte aynı sonucun döndürülmesi dinamik web sayfalarında bulunmamaktadır. Dinamik web sayfalarındaki içerik değişkendir. İçerik her bir bireye göre veya duruma göre güncellenebilir.

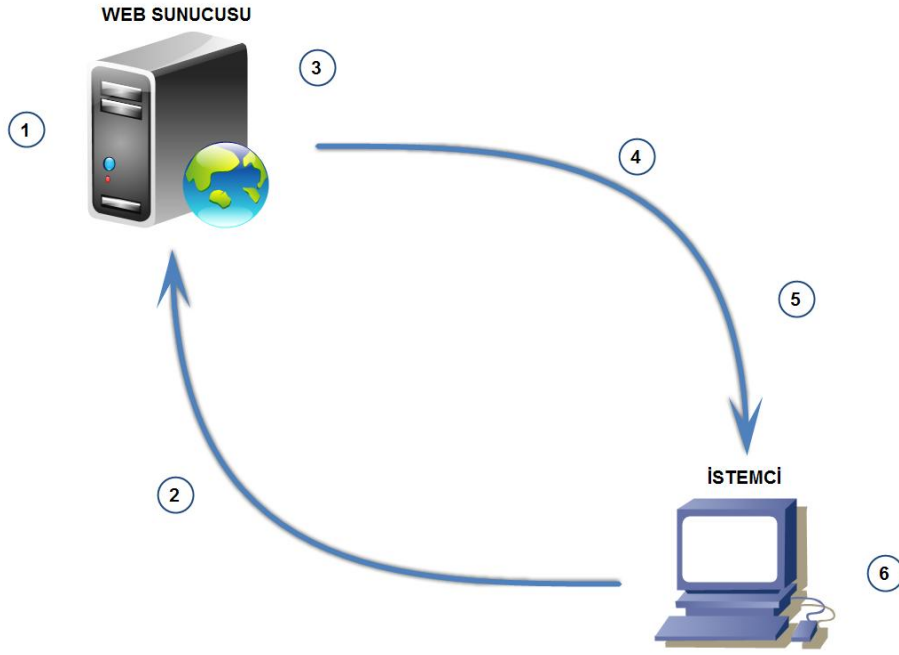
Resimler, form sahaları, bilgiler vb. farklı durumlarda veya bağlamlarda farklı sayfalar üretebilir. Böylece web sayfasını ziyaret eden bir kişiye güncel içerik sunulmuş olur. Bu sunuşu yapabilmenin yolları kullanıcı tarafında ve sunucu tarafında yazılan betikler aracılığıyla olabilir [8]. Kullanıcı tarafında yazılan betikler, kullanıcının sayfadaki klavye veya fare ile yaptığı seçimlere göre veya zamanlanmış işlemlere göre çalışabilir. Bu çalışma işlemi sayfanın kendisinde olur ve dinamik işlem sayfanın görüntüsünde meydana gelir. Sunucu tarafı betikleri de sayfaların yeniden yüklenmesi ve içeriğinin tarayıcı tarafından yenilenmesi ile yapılır. Sunucudan gelen yanıt HTML formunda gönderilen parametrelere göre, URL aracılığıyla gönderilen parametrelere göre, kullanılan tarayıcıya, zamana, veritabanı veya sunucu makinelerin durumuna göre değişebilir.

Dinamik web sayfalarında içerik web sunucuda hazır bulunmamaktadır. İçerik web sunucusundan bağımsız oluşturulmakta ve ardından HTML dosyası yeniden yapılandırılarak istemcilere gönderilmektedir. Bu amaçla kullanılan en yaygın sunucu tarafında kullanılan programlama dillerine PHP, PERL, ASP, JSP, ASP.NET, ColdFusion örnek verilebilir [8].

Dinamik web sayfasının istekten yanıtı kadarki süreci Şekil 2.3.' de numaralarla ifade edildiği gibi şöyledir:

1. Öncelikle sunucu tarafında kullanılan dillerden biri kullanılarak web sayfası üretilir ve kullanıma açılır.
2. Kullanıcı tarayıcı penceresinden istekte bulunulacak sayfanın adresini girer ve istemci sunucudan istekte bulunur.
3. Web sunucusu sayfayı inceler ve dinamik içeriğin oluşturulması gereken uygulamaya gönderir. Bu günümüz web uygulamalarında çoğunlukla uygulama sunucusudur.
4. Uygulama sonucu üretmek üzere işlemleri gerçekleştirir.
5. Sonuç HTML olarak kodlanmış şekilde web sunucusu aracılığıyla istemciye döndürülür.
6. Tarayıcı HTML sayfasını çözerek sayfayı görüntüler.

Dinamik web’ de gereken durumlar için statik sayfalar da kodlanmaktadır. Bu durumda 3 numaralı aşamada web sunucusu uygulama sunucusuna isteği yönlendirmeden önce sayfanın statik bir sayfa olduğunu algılayıp dosyayı doğrudan istemciye döndürür.



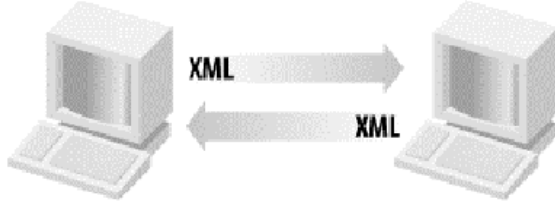
Şekil 2.3. Dinamik web’ in çalışma döngüsü

3. WEB SERVİSLERİ

Web uygulamalarının yeni geliştirilmeye başladığı yıllardan itibaren web servis teknolojisi kullanılabildiği kadarı zamanda web ortamında fazla sayıda web sayfası bulunmaktaydı. Ancak bu uygulamalar farklı platformlarda çalışıp, farklı diller kullanılarak geliştirilmiş olduğu için birbirine entegre olup bir arada çalışmaları mümkün olmuyordu.

Web' in yaygınlaşması ile birlikte insan hayatını kolaylaştıran çözümler sunan hizmetler de artmıştır. Aynı zamanda elektronik uygulamaların artması sayesinde birçok rutin işlemi daha kısa sürede ve daha zahmetsiz halledebilmek mümkündür. Elektronik ortamda verilen fazla sayıdaki hizmetin birlikte daha tutarlı ve yardımlaşarak çalışması amacıyla verilen hizmetlerin birbiriyle entegrasyonu ihtiyacı doğmuştur. Bu sayede gerçek hayatta kurumlar arasında bulunan bilgi alışverişleri ve çalışan işlevler web ortamındaki uygulamalar aracılığıyla hizmet verir hale gelmiştir. Web servisleri bu aşamada eski teknolojilere göre daha hızlı ve verimli bir şekilde farklı programlar arasında veri alış verişine izin vererek, uygulamaların birbirine bağlanmasını sağlamıştır.

Web servis kısaca internet üzerinden erişilebilen, standartları belirlenmiş bir XML mesajlaşma sistemine sahip olan ve herhangi bir işletim sistemi veya programlama diline bağlı olmayan servistir [9]. Daha detaylı bir tanım ise şu şekildedir; web servis kendi tanımına sahip olan, diğer uygulamalar tarafından aranıp bulunabilen ve kolayca çağrılabilen bir tür yazılımdır. Geleneksel web uygulamaları ile arasındaki temel fark web sayfalarının kullanıcı kitlesi olarak insanları hedeflerken, web servislerinin kullanıcı kitlesi olarak diğer uygulamaları hedeflemesidir. Web sayfasının kullanımında insanla web sayfası arasında bir bilgi aktarımı oluyorken, bu ilişki web servislerinde uygulamadan uygulamaya olmaktadır [10].



Şekil 3.1. Web servislerinin genel gösterimi

Web servis yazılımlarının birer servis olarak nitelendirilmesinin sebebi de makinelerin web servisinin özelliklerini ve ne gibi işlemlere sahip olduğunu kişilerin müdahalesine gerek kalmadan bilebilmesi ve diğer uygulamalarla birlikte çalışabilmesidir.

Web servisleri geleneksel yazılım teknolojilerinden ayıran farklar aşağıdaki gibidir [1].

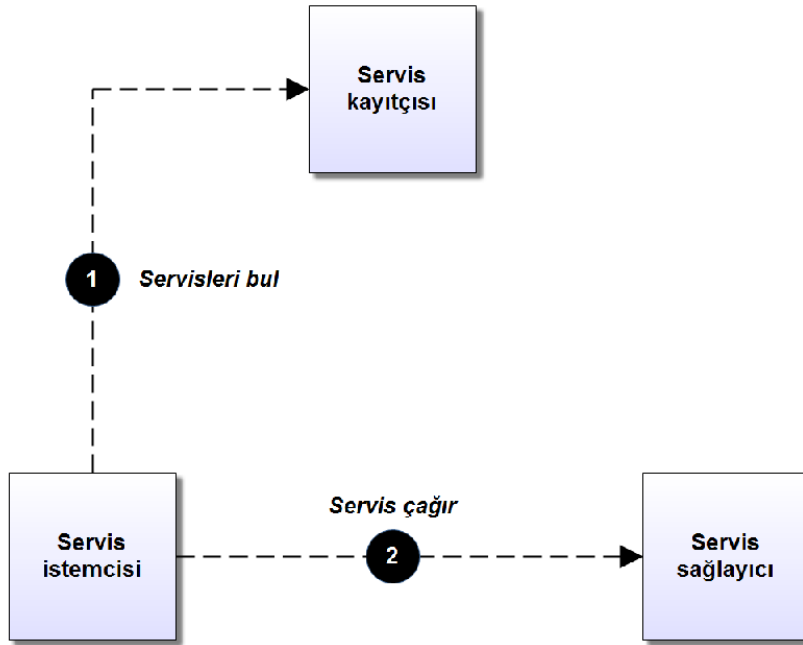
1. Web servisleri geliştirmede kullanılan programlama dili, işletim sistemi, ağ iletişimi ve veri saklama metotlarından bağımsız, esnek bir şekilde kodlanabilen ve anlaşılabilen veri biçimi içermektedir.
2. Verinin karşı taraftaki yazılım uygulamasına nasıl geçirileceği ile ilgili ara yüzde bilgi içermemektedir. Web servisleri daha çok adaptör vazifesi görmektedir. Yani, üretilen verinin karşı taraftaki yazılıma entegre edilmesi kısmıyla ilgilenmez. Standart biçimdeki veriyi sunar ve daha sonra o verinin kullanılması XML işleyen kod parçacıklarının işidir.
3. Web servisleri kendilerini çağırarak yazılımlarla ilgili herhangi bir kural koymadan, bütün yazılımlarla uyum içerisinde çalışabilir.

Web servislerinin yaygınlaşmasında XML tabanlı protokollerin önemli bir payı vardır. Web servisleri web ortamının standartlarını destekleyerek hem işlevsellikleri hem de erişim şekliyle ilgili XML tabanlı bilgi içermektedirler.

Web servislerin mimarisini iki farklı grupta toplanan öğelere göre sınıflandırmak mümkündür. Bu gruplar serviste kullanılan roller ve teknolojilerdir [9].

Web servis mimarisindeki roller:

1. Servis sağlayıcı: Servisin gerçekleştirilmesine sahip olan ve onu internet üzerinden erişime açan taraftır.
2. Servis istemci: Web servisini kullanan bir öğedir. Web servisine bir ağ bağlantısı açıp ardından da bir XML istemi göndererek servisi kullanır.
3. Servis kayıtçısı: Servis kayıtçısı bütün web servislerin mantıksal olarak kaydının tutulduğu ve istenildiğinde aranıp bulunabildiği bir yapıya sahiptir.

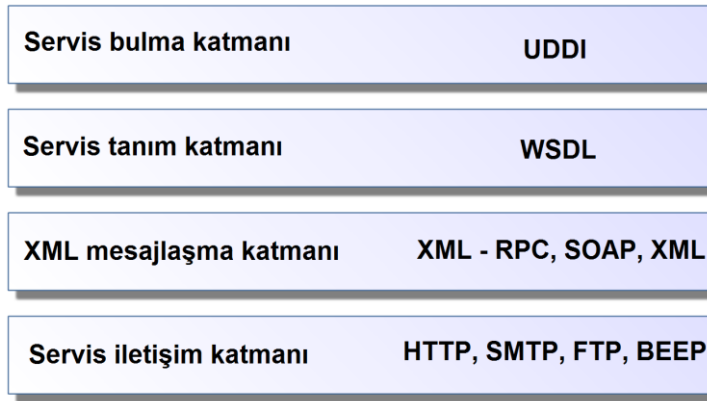


Şekil 3.2. Web servislerinin mimarisindeki rollerin birbiriyle ilişkisi

Web servis mimarisinde kullanılan teknolojilerin tamamına web servis protokol yığını adı da verilir. Servis protokol yığını geliştirmekte olan eklentilerle birlikte güncellenmektedir, ancak bu kısımda temel 4 parçasından bahsedilecektir.

1. Servis İletişim Katmanı: Bu katmanın görevi uygulamalar arasında mesaj geçişini sağlamaktır. HTTP, SMTP, FTP ve benzer protokollerden oluşur.

2. XML ile Mesajlaşma Katmanı: Bu katmanda mesajların istemci ve sunucu taraflarında kolay anlaşılmasını sağlamak için XML ile kodlanması yapılır. Şu anda bu katman XML-RPC ve SOAP teknolojilerini içermektedir.
3. Servis Tanım Katmanı: Bu katman servisin diğer servisler tarafından anlaşılabilir şekilde tanımlanmasını sağlar. Servis tanımı WSDL ile yapılmaktadır.
4. Servis Bulma Katmanı: Bu katmanın görevi servislerin merkezi bir kayıtçıda saklanmasını, bu kayıtçidan tanıtılmasını ve arandığında bulunmasını sağlar. Bu katmanın işlevliğini sağlayan teknoloji UDDI' dir.



Şekil 3.3. Web servis protokol yığını

3.1. Web Servis Altyapısında Kullanılan Teknolojiler

Web servis teknolojisine temel olan gelişme Genişletilebilir İşaretleme Dili' nin (Extensible Markup Language kısaca XML) [11] geliştirilmesi olmuştur. XML' den sonra geliştirilen teknolojiler XML tabanlı olup, avantajlarından faydalanmaktadır. Web servislerin geliştirilmesi sürecinde bir diğer teknoloji olan Web Servisleri Tanımlama Dili' nin (Web Service Definition Language veya kısaca WSDL) tanımlanması yoluyla web servislerinin kendi işlevselliklerini ve metotlarının çağrılma şekliyle ilgili bilgi barındırması mümkün olmuştur. Diğer uygulamaların web servisini kullanabilmesi açısından web servisinde bulunan WSDL [10] dosyasının yayınlandığı yer ise UDDI [10] (Universal Description Discovery and Integration) kayıtlarının bulunduğu kayıtçidir.

3.1.1. Geniřletilebilir iřaretleme dili (XML)

Web'in doęal olarak daęıtık ve farklı platformlardan oluřan yapısı sonucu güvenli, platformdan baęımsız, tüm dnyada sorunsuz alıřabilecek ve hantal olmayan iletiřim protokollerine ihtiyaı vardır [12].

1998 yılında W3C tarafından duyurulan Geniřletilebilir İřaretleme Dili (Extensible Markup Language veya kısaca XML) veriyi saklamanın yanında, farklı platformlarda kullanılabilir ve veri alışveriři yapabilecek bir biime sahiptir. XML, HTML' in eksiklerini tamamlamak için dinamik ierik retimi ve ynetimini desteklemek için geliřtirilmiř bir dildir. rneęin; XML' de bir řema dosyası aracılıęıyla dosyanın doęrulanması mmkndr. XML birok trdeki uygulamada kullanılmasına raęmen, web servisleri aısından bakıldığında XML' in sadece bir kodlama dili olmadıęı, aslında web servisleri tanımlamanın bir yolu olduęu sylenebilir [1].

XML etiketleri aynı zamanda barındırdıkları veri hakkında fikir vermektedir, xml belgelerinde herhangi bir programlama dili veya platform bilgisi yer almamaktadır. Bu da XML' in bařlıca avantajı olan veri tipleri ve yapılarının gerekleřtirim detaylarını barındırmaması zellięinden kaynaklanmaktadır.

3.1.2. Uzaktan yordam aęırma (XML - RPC)

Web servislerinin birbirine entegre olması için zm sunan teknolojilerden biri de Uzaktan Yordam aęırma (Remote Procedure Call veya kısaca RPC)' dir. XML-RPC 1998 yılının bařlarında ortaya ıkmıř bir teknolojidir [9] . XML-RPC belirli programlama trlerle kısıtlanmıř bir řekilde HTTP zerinden XML dosya biiminde sahip mesaj geirmeye yarar.

XML-RPC' nin programcılar tarafından tercih edilmesini iki sebebe baęlamak mmkndr. Bu sebeplerden birincisi; daęıtık sistemlerde birbirinden farklı

ortamlara sahip makineleri bir araya getirebilme amacıdır. Diğer de sadece ara yüz tanımlayarak arka planda o ara yüzü istenen platformda geliştirme isteğidir.

XML-RPC 3 bölümden oluşur. Bunlar; XML-RPC veri modeli, XML-RPC istem yapıları, XML-RPC yanıt yapılarıdır [9]. XML-RPC veri modeli 6 tipte veri tanımlama türü içermektedir. Bunlar tamsayılar için “int”, kesirli sayılar için “double”, mantıksal doğru veya yanlış için “Boolean”, karakter dizeleri için “String”, tarih belirtmek için “dateTime.iso8601” ve yalnızca 0 ve 1’lerden oluşan bilginin “Base 64” biçiminde kodlanmış olarak saklandığı base64 türleridir.

XML-RPC istemleri ise XML içeriğine ve HTTP başlık bilgisine sahip yapılardır. Şekilde görüldüğü gibi HTTP başlık bilgisi ve bir XML belgesi içermektedir. XML belgesinde belirtildiği gibi örnekte istemde bulunan metodun adı “circleArea” ve aldığı tek parametre “double” türündeki “2.41” değeridir.

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169

<?xml version="1.0"?>
<methodCall>
  <methodName>circleArea</methodName>
  <params>
    <param>
      <value><double>2.41</double></value>
    </param>
  </params>
</methodCall>
```

XML-RPC yanıtları da aynı şekilde istemlere benzer. Görüntüsel olarak “<methodName>” etiketi “<methodResponse>” ile değişmiş olur ve “<param>” etiketleri ile döndürülen yanıt ifade edilir.


```

HTTP/1.1 200 OK
Date: Sat, 06 Oct 2001 23:20:04 GMT
Server: Apache/1.3.12 (Unix)
Connection: close
Content-Type: text/xml
Content-Length: 124

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><double>18.24668429131</double></value>
    </param>
  </params>
</methodResponse>

```

XML-RPC kullanarak mesajlaşabilen bir uygulama kodlayabilmek için kodlanan platformda kullanılabilecek kütüphanelere ihtiyaç vardır. Bu kütüphaneler sayesinde kodu kolayca oluşturup çalıştırmak mümkündür. Ancak XML-RPC' nin bazen kısıt olarak gösterilen basitliği, bazen avantaja dönüşebilmektedir. Basit bir kod parçasıyla karmaşık bir işlemi kolayca ve sorunsuz gerçekleştirmek mümkün olmaktadır [9].

3.1.3. Basit nesne erişim protokolü (SOAP)

Web servislerin kullanımı yaygınlaştıkça web uygulamalarının birlikte çalışabilmesi ihtiyacı daha da önem kazanmıştır. Bir web servis geliştirilip yayınlandıktan sonra başka bir uygulama servisin veya kendisinin hangi ortamda geliştirildiğine bakmaksızın kolayca bu web servisi bulup kullanabilmelidir. Bu amaçla Basit Nesne Erişim Protokolü adındaki (Simple Object Access Protocol veya kısaca SOAP) XML ile mesajlaşmanın kurallarını içeren protokol 2000 yılının Eylül ayında tanımlanmıştır. Platformdan bağımsız bir şekilde verileri kodlamak için XML'in oldukça uygun olması, mesajlaşma protokolünün XML ile geliştirilmesini sağlamıştır. SOAP protokolü, web üzerindeki servis fonksiyonlarını kullanmak için geliştirilen kurallar bütünüdür. Web servisi isteklerini paketleme ve işleme sürecinde kullanılır. SOAP protokolü iletişim protokolleri ile, sıklıkla HTML ile, XML' in esnek ve geliştirilebilir özelliklerini birleştirerek var olan program ve platformların birlikte işlerliğini sağlar [10].

SOAP iletişim protokollerinden herhangi birine bağlı değildir. Bir SOAP istek veya yanıt mesajı farklı protokoller ile taşınabilir. Ancak SOAP tanımında HTTP taşıma protokolünün detayları mevcut olduğundan HTTP, SOAP mesajlarını taşımak amacı ile en popüler kullanılan protokoldür [9].

SOAP kurallar bütününe göre üretilmiş bir mesaj 3 ana parçadan oluşur

1. Zarf; mesaj içeriğinin paketlenmesine yarayan altyapıdır.
2. Kodlama kuralları; mesajların nasıl işleneceğine dair bulunur.
3. Prosedür çağırma tanımı; prosedürlerin nasıl çağrılacağını ve yanıtlarını içerir.

SOAP mesajlarının yapısı basittir. Bir mesajda iki adet alt eleman bulunur, bunlardan biri başlık, diğeri gövde bilgisidir [12].

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Header>
    <!-- Başlık bilgisinin içeriği buraya yazılır, zorunlu değildir -->
  </SOAP:Header>
  <SOAP:Body>
    <!-- Gövde bilgisinin içeriği buraya yazılır, zorunludur -->
  </SOAP:Body>
</SOAP:Envelope>
```

Örnek bir SOAP mesajı aşağıdaki gibidir [12]. Mesajın başındaki satır numaraları anlatımı kolaylaştırmak amacıyla verilmiştir ve gerçek bir SOAP mesajında bulunmaz.

```
1 POST /travelservice HTTP/1.1
2 SOAPAction: "http://www.acme-travel.com/checkin"
3 Content-Type: text/xml; charset="utf-8"
4 Content-Length: nnnn
5 <SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
6 <SOAP:Body>
```

```

7 <et:eTicket xmlns:et="http://www.acme-travel.com/eticket/schema">
8 <et:passengerName first="Joe" last="Smith"/>
9 <et:flightInfo airlineName="AA"
    flightNumber="1111"
    departureDate="2002-01-01"
    departureTime="1905"/>
10 </et:eTicket>
11 </SOAP:Body>
12 </SOAP:Envelope>

```

1.satırdan 4. satıra kadar bu mesajın anlattığı; mesajın bir HTTP POST mesajı olduğunu ve iletişim protokolü olarak HTTP kullanıldığıdır. Ayrıca servisin web adresi yer almaktadır. Son olarak da servis tarafından üretilen yanıtın kodlama karakter kodlama bilgileri ve mesaj uzunluğu belirtilmektedir.

5. ve 12. satırlar arası ise SOAP mesajının temel bileşenlerinden “Zarf” bilgisini içermektedir. Kısaca biletteki yolcu ismini ve uçuş bilgilerini XML etiketleri şeklinde bu bölümde belirtilmektedir.

3.1.4. Web servis tanımlama dili (WSDL)

SOAP ve XML-RPC gibi tanımlar web servisleri için basit bir mesajlaşma protokolü olarak tanımlansa da bir web servisi ile iletişim kurabilmek için gerekli bilgileri içermezler. Bu amaçla 2000 yılında web servis hizmetinin, onu kullanmak isteyen diğer servislere ifade edilmesi ve bunun bir kurallar bütününe bağlı olması amacıyla XML tabanlı olan Web Servis Tanımlama Dili (Web Service Definition Language veya kısaca WSDL) kabul görmüştür. WSDL’ i bir web servisini kullanmak için belirtilen ara yüz veya web servisinin irtibat noktası olarak tanımlamak mümkündür. Diğer bir deyişle de WSDL aslında web servisi ve kullanan yazılımlar arasındaki sözleşmeye benzer [10].

WSDL, web servisteki verilerin bir veya birden fazla XML şema belgesi kullanarak tanımını yapar ve böylece hem servis hem de servisi kullanan taraf gidip gelen veri hakkında bilgi sahibi olur. Ayrıca bir WSDL belgesi web servis tarafından sağlanan metotları, girdi ve çıktı parametrelerini ve genelde http olarak kullanılan iletişim protokolünü tanımlar.

Örnek bir WSDL dosyası aşağıdaki gibidir [9]. Her satırın başındaki sayılar anlatımı kolaylaştırmak amacıyla koyulmuştur, dosyanın gerçek biçiminde bulunmamaktadır.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <definitions name="HelloService"
3      targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
4      xmlns="http://schemas.xmlsoap.org/wsdl/"
5      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
6      xmlns:tns="http://www.ecerami.com/wsdl/HelloService.wsdl"
7      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
8      <message name="SayHelloRequest">
9          <part name="firstName" type="xsd:string"/>
10     </message>
11     <message name="SayHelloResponse">
12         <part name="greeting" type="xsd:string"/>
13     </message>
14     <portType name="Hello_PortType">
15         <operation name="sayHello">
16             <input message="tns:SayHelloRequest"/>
17             <output message="tns:SayHelloResponse"/>
18         </operation>
19     </portType>
20     <binding name="Hello_Binding" type="tns:Hello_PortType">
21         <soap:binding style="rpc"
22             transport="http://schemas.xmlsoap.org/soap/http"/>
23         <operation name="sayHello">
24             <soap:operation soapAction="sayHello"/>
25             <input>
26                 <soap:body
27                     encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
28                     namespace="urn:examples:helloservice" use="encoded"/>
29             </input>
30             <output>
31                 <soap:body
32                     encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
33                     namespace="urn:examples:helloservice" use="encoded"/>
34             </output>
35         </operation>
36     </binding>
37     <service name="Hello_Service">
38         <documentation>WSDL File for HelloService</documentation>
39         <port binding="tns:Hello_Binding" name="Hello_Port">
40             <soap:address
41                 location="http://localhost:8080/soap/servlet/rpcrouter"/>
42         </port>
43     </service>
44 </definitions>

```

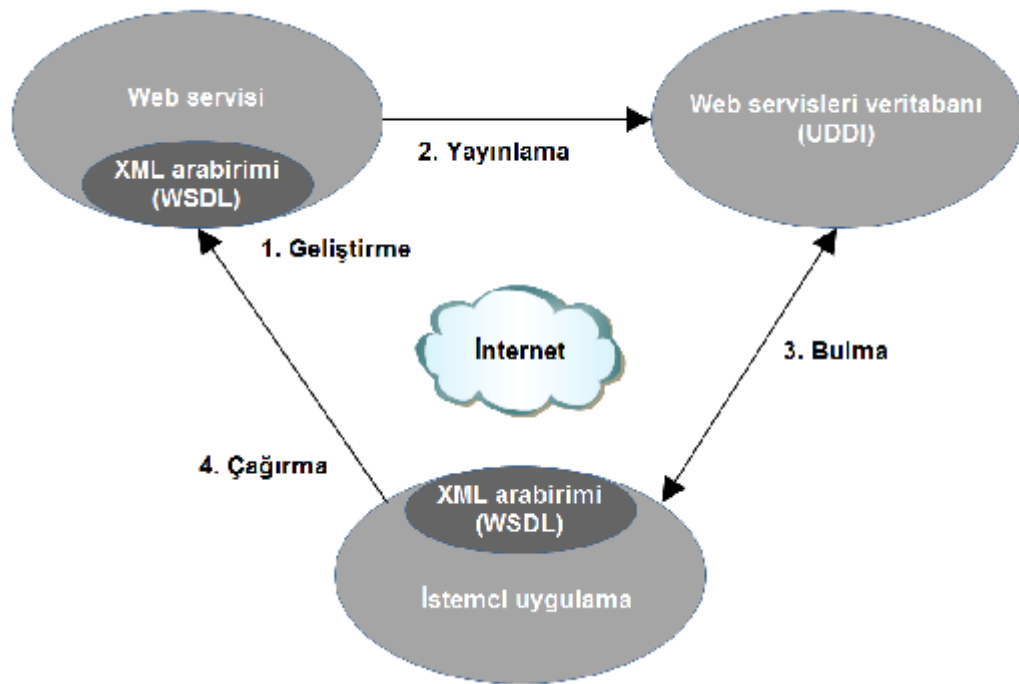
2 numaralı satırdan 7 numaralı satıra kadar olan kısım web servis ile ilgili tanım bilgilerinin yer aldığı kısımdır. 2 numaralı satır bu servisin bir “HelloService” olduğunu belirtir. Ayrıca 3-7 satırları arasında da dosyanın kalan kısmında kullanılacak isim alanları belirtilir. 8 numaralı satırdan 10 numaralı satıra kadar olan bölüm bu mesajın bir istek mesajı olduğunu ve 9 numaralı satır ile “firstName” adında bir parametre aldığını belirtir. 11-13 numaralı satırlar arası bu mesajın bir yanıt mesajı olduğunu ve “greeting” adında bir yanıt değeri döndüreceğini belirtir. 14 ve 19 numaralı satırlar arasındaki kısım bu serviste “sayHello” adındaki bir işlemin sağlandığı ve bunun “SayHelloRequest” istek mesajı ve “SayHelloResponse” istek mesajından oluştuğunu gösterir. 20 ve 33 numaralı satırlar “sayHello” işlevinin internet hattından nasıl taşınacağı ile ilgili detaylı bilgi içermektedir. 34 numaralı satırdan başlayan ve 39 numaralı satırda biten “<service>” etiketi ile ilgili kısım ise servisin bulunduğu yeri işaret etmektedir. 35 numaralı “<documentation>” etiketi, okunabilir bilgi içermektedir.

WSDL dosyası elle oluşturulması zor ve hataya açık bir yapıda olduğundan, yazılım geliştirme araçları üzerinde WSDL dosyasını otomatik üretilecek işlevler mevcuttur.

3.1.5. Evrensel kayıtçı (UDDI)

Bir web servisi yayınlandıktan sonra diğer uygulamaların bu servisten faydalanabilmek için bu servise ulaşabilecekleri bir dizine ihtiyaçları vardır. Bu amaçla Evrensel Kayıtçı (Universal Description Discovery and Integration veya kısaca UDDI) Ağustos 2000’de tanıtılmıştır.

Web servisi kodlandıktan sonra SOAP mesajları aracılığıyla UDDI kayıt veritabanına kaydedilir. Böylece de o web servisi kullanmak isteyen uygulamalar aradıklarında WSDL dosyası ile birlikte servis bilgilerini bulabilirler. Böylece dünya çapında üretilen web servislerinin birbirini UDDI ile bulması ve entegre olması sağlanır.



Şekil 3.4. Web servis teknolojileri ve ilişkileri

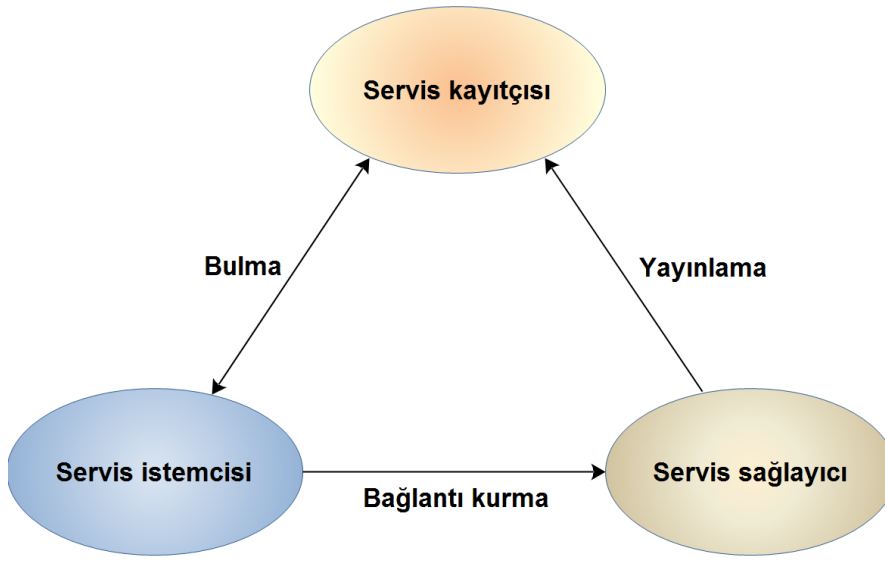
Yukarıdaki şekilde web servis gerçekleştiriminde kullanılan teknolojilerin özetlendiği gibi WSDL ara yüzüne sahip bir web servisi öncelikle UDDI kayıtçısında yayınlanmalıdır. UDDI kayıtçısı bir servis çağırma istemi geldiği takdirde bu servisin bulunmasını sağlar ve servise bağlanabilmek için gerekli bilgileri istemciye sunar. Ve bu bilgilere sahip olan istemci servisi kullanabilmek için WSDL’ de belirtildiği şekilde çağırma işlemini gerçekleştirir.

3.2. Servis Tabanlı Mimari (SOA)

Bir yazılım olarak servis, nasıl gerçekleştirildiği hakkında fikre sahip olmayan ve bu servisi sadece kullanmak isteyen yazılım etmenlerinin kolayca arayıp bulabildiği ve çağırabildiği; bir iş sürecinin gerçekleştirimi ve kurallara uygun biçimde yayınlanan ara yüzünden oluşan yapıdır [13].

Servis tabanlı mimari (Service Oriented Architecture veya kısaca SOA) standart ara yüzler ve mesajlaşma protokolleri aracılığıyla erişilecek şekilde yazılımın

servislerden oluşturulmasıdır [13]. Bu mimari yaklaşım daha çok farklı platformlar ve işletim sistemlerinde çalışan yazılımların birlikte çalışması gerektiği zaman kullanılmaktadır. Böylece minimum programlama yapılarak uygulamaların entegre edilmesi sağlanabilir. SOA yalnızca servislerle ilgili bir mimari olarak tanımlanamaz. SOA Şekil 3.5’ de görüldüğü gibi servis istemci, servis sağlayıcı ve servis keşfi için çalışan servis kayıt birimi arasındaki etkileşimi de tanımlar.



Şekil 3.5. SOA tanımındaki web servis birimleri

Servis sağlayıcı servis kayıtcısına servisi kaydederek nasıl çağırılacağına ait detayı belirtir, böylece servis yayınlanmış olur. Servis istemcisi ise servis kayıt biriminden servisi ve ona nasıl ulaşabileceğini arayarak bulur, gerekli parametreleri öğrenir ve servisi ona göre çağırır. Servis kayıtcısı ise web servis sağlayıcılarının yayınladığı web servisi ve tanımlarını bulundurur, istemcilerden gelen taleplere göre de servisin bulunmasını sağlar. Ve bu tanımları servis istemcisine ileterek servisin nasıl çağırılacağıyla ilgili istemciyi bilgilendirmiş olur.

Servis tabanlı mimarideki bir servisin ne yaptığını anlamak için detayına bakmaya gerek yoktur. Aynı zamanda servisin de kendisini kullanan yazılımların gerçekleştirim detayı veya iş mantığı ile ilgili bilgiye ihtiyacı yoktur. Böylece servis

tabanlı yaklaşım ile istemciye bağımlılık ortadan kalkmıştır. Sisteme eklenecek yeni bir istemcinin az bir programlama çabası ile entegre olması kolaylaşmıştır.

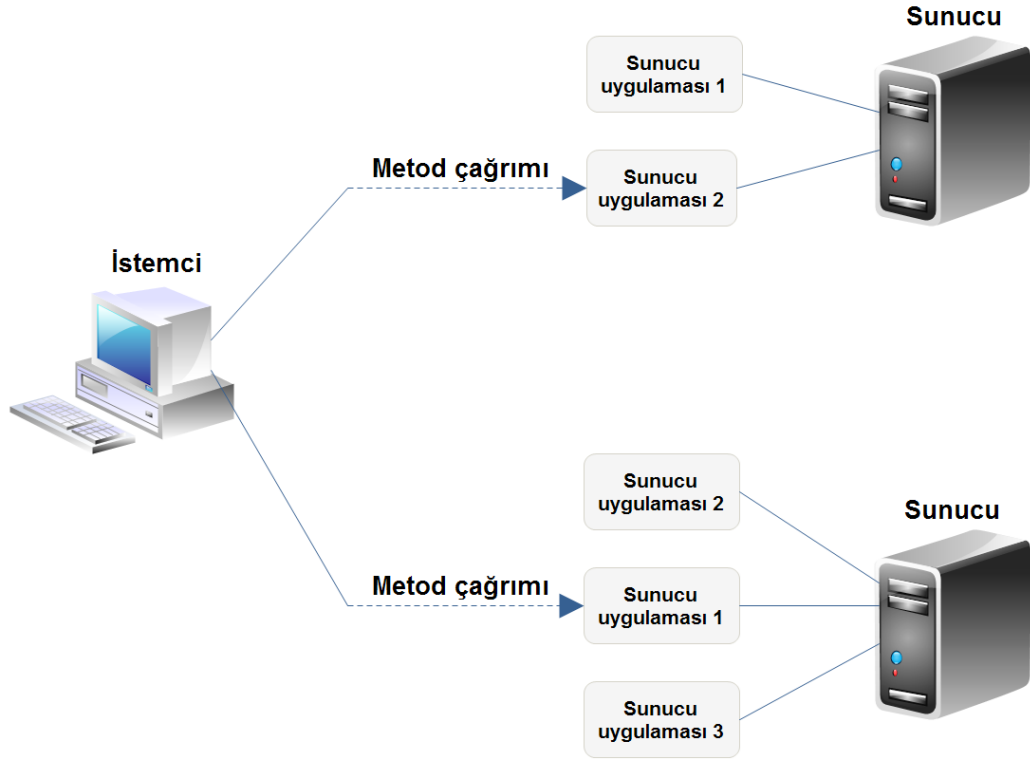
Servis tabanlı mimarinin özelliklerini aşağıdaki özellikleriyle tanımlamak mümkündür [14].

- Mantıksal görünüm: Servis bir iş sürecine çözüm bulabilmek için var olan, programların, veritabanının ve iş süreçlerinin soyutlanmış halidir.
- Mesaja yönelik: Bir web serviste istemci ve sunucu etmenlerin özelliklerinden ziyade birbiriyle mesaj iletişimi o servisi tanımlar. SOA’ da istemci veya sunucu etmenlerin iç yapısını tanımlayan gerçekleştirim dili, işletim yapısı veya veritabanı yapısı soyutlanmıştır. SOA modelini kullanan bir etmen web servisin gerçekleştirim detayını bilemez.
- Tanıma yönelik: Bir servis makineler tarafından da okunabilen veri hakkında veriden oluşur. Servislerin bu özelliği XML tabanlı olmalarından kaynaklanır. Servisin tanım dosyasında ancak dış dünyayla paylaşılmasına sakınca olmayan ve diğer servislerin kullanımı için gerekli olan bilgiler yer almalıdır. Servisin hakkındaki bilgiler servis tanımı yoluyla sunulmalıdır.
- Küçük parçalardan oluşma: Büyük işlemleri bile küçük parçalar halinde işlemeye yöneliktirler.
- Bilgisayar ağına yönelik: Web servisler bilgisayar ağı üzerinden kullanılabilirler.
- Platform belirlemez: Mesajlar platformdan bağımsız bir şekilde ve belirli bir biçimde ara yüzlerden geçirilirler. Bu XML bu şekilde mesaj geçirme kısıdının çözümünü oluşturmaktadır.

4. DAĞITIK SİSTEMLER

Web' in ve bilgisayar sistemlerinin gelişimi, beraberinde sistemlerin sağladığı işlevlerin artmasına yol açmış ve bu ihtiyaçlar karşısında kaynak problemi oluşmaya başlamıştır. Büyütülemeyen bilgisayar sistemleri talep karşısında yetersiz kalmaktadır. Tek merkezden yönetilen sistemler büyümek zorunda kaldığında yapılacak değişiklikler için gereken esnekliği sağlayamaz duruma gelebilir, bu sistemlerde performans iyileştirilmesi zorlaşabilir. Tek merkezli sistemlerde örneğin bir veritabanı cihazı kapasitesi yetmediği zaman ancak yerine yenisi temin edildiğinde değiştirilebilir. Ancak dağıtık sistemlerde bu ve benzer problemler sisteme yeni bir makine eklenmesi yoluyla çözülebilir. Tek merkezli sistemlerdeki problemlere çözüm bulmak için farklı yöntemler geliştirilmeye ihtiyaç duyulmuştur.

Dağıtık programlar sayesinde ağ üzerinde birden fazla makineden faydalanmak mümkün hale gelmiştir. Dağıtık sistemler tek bir sistemmiş gibi çalışan ancak arka planda birçok bilgisayarın birlikte çalışmasıyla bir araya gelmiş sistemlerdir. Daha iyi başarımlar ve kaynakları daha verimli kullanabilmek için dağıtık sistemlerde değişik görevler paralel olarak çalıştırılabilmektedir.



Şekil 4.5. Dağıtık sistemlerin genel görünümü

4.1. Dağıtık Sistem ve Dağıtık Programlama

Dağıtık programlar ve dağıtık sistemler; tek başına çalışabilen bilgisayarların fiziksel olarak dağıtık şekilde bulunduğu bilgisayar ağlarıyla ilgili çalışmaktadır. Bu sistemlerde birbirine uzak olan birçok fiziksel bileşen tek bir sistem gibi çalışmaktadır [15]. Burada fiziksel bileşeni bir bilgisayar olarak nitelendirmek mümkündür.

Dağıtık sistemler bilgisayar ağları aracılığıyla birbiriyle haberleşebilen ve tek başına çalışabilen birçok bilgisayardan oluşurlar. Dağıtık sistemde çalışan bilgisayarların ortak bir hedef için birlikte çalıştığını söylemek mümkündür, bu hedef de çoğunlukla karmaşık problemleri çözmektir. Çalışan her bir bilgisayarın kullanıcı kitlesi vardır ve dağıtık bir sistemin amacı paylaşılan kaynakları koordine etmek veya kullanıcılara iletişim servisleri sunmaktır [16].

Dağıtık sistemlerde çeşitli ve birbirinden farklı yazılım etmeni, çeşitli işlemleri gerçekleştirebilmek için birlikte çalışır. Çalışan yazılım etmenleri aynı işletim ortamında çalışmadığından birbiriyle yazılım veya donanım protokolleri ile haberleşmek durumundadır.

Dağıtık sistemlerde çalışan bilgisayar programlarına dağıtık program, bu programların geliştirilmesi için yapılan çalışmaya dağıtık programlama denir [16]. Dağıtık sistemleri farklı özellikleriyle tanımlamak mümkündür ancak en çok kullanılan tanımlayıcı iki özellik şunlardır:

- Sistemde birden fazla kendi hafızasına sahip olan bireysel olarak çalışabilen bilgisayar vardır.
- Bu bilgisayarlar birbirine mesaj geçirme yoluyla haberleşirler.

Dağıtık sistemlerin kullanılmasının bazı sebepleri mevcuttur. Bunlardan biri; bilgisayarda çalışan uygulamaların, birden fazla bilgisayarın iletişime geçebildiği ağı kullanma ihtiyacıdır. Örneğin; bir veri bir bilgisayarda kayıtlıyken, diğerinin ona ihtiyacı olabilir. Diğer bir sebep ise dağıtık sistemleri kullanmanın daha faydalı olmasıdır. Örneğin; bir adet yüksek özellikli bilgisayar kullanmak yerine birden fazla ortalama özelliklere sahip makineyi kullanmak maliyet açısından karlı olabilir [16]. Böylece kapasitesi daha az makinelerden da ağ ortamında birlikte işlerlik sayesinde faydalanılmış olur. Dağıtık yapısı olmayan, merkezi sistemlerde hata merkezde meydana geldiği anda tüm sistemde kesinti yaşanabilir, bu durum hatanın tek odakta olmasından kaynaklanır. Hatanın yaşandığı makine probleminden dolayı kullanılmaz hale gelirse bu sorundan tüm sistem etkilenmiş olur. Oysa dağıtık sistemlerde bir makinede problem olduğu anda ona alternatif olarak işlemleri sürdürecekt diğer makinelerin oluşu sistemi daha güvenilir yapmaktadır.

Avantajlarından dolayı dağıtık sistemleri kullanan birçok aktif uygulama mevcuttur. Bu uygulamalar; telekomünikasyon ağlarında, paralel çalışmaların yapıldığı sistemlerde, ağ uygulamalarında ve uçak kontrol sistemi gibi gerçek zamanlı sistemlerde kullanılmaktadır.

4.2. Dağıtık Sistemlerin Özellikleri

Bir dağıtık sistemin sahip olması gereken özellikler şu şekilde sıralanabilir [17].

- **Hataya toleransı:** Dağıtık sistemdeki bileşenlerden birinde hata yaşanması durumunda, bu hata düzeltilmelidir.
- **İyi çalışabilirlik:** Bazı bileşenlerde problem olması durumunda, hizmetin devamlılığı sağlanıp işlemler onarılmalıdır.
- **Düzeltililebilirlik:** Problem yaşayan bileşenler problemin çözümünden sonra sisteme tekrar dahil olmalıdır.
- **Tutarlılık:** Eş zamanlı kullanım ve hata durumlarında bileşenler sistem tarafından koordine edilmelidir. Bu özelliği, bir dağıtık sistemin dağıtık olmayan bir sistemmiş gibi davranması şeklinde de tanımlanabilir.
- **Ölçeklenebilirlik:** Sistemde kullanılan kaynakların artırılması durumunda probleme yol açmadan sistem tutarlı bir şekilde çalışmaya devam edebilmelidir.
- **Tahmin edilebilir performans:** Tahmin edilebilir ve makul yanıt sürelerine sahip olmalıdır.
- **Güvenilirlik:** Sistem veri ve servislere erişimi doğrulayabilmelidir.

Burada bahsedilen özelliklerin gerçekleştirilmesi zordur. Fakat bu standartlardan sağlanması en zor olanı, bileşenlerin bazılarında sistem hataları olmasına rağmen sistemin doğru bir şekilde çalışmaya devam etmesidir.

Hata yönetimini yazılım ve donanımsal hatalar diye ayırmak mümkündür. Donanımsal hatalar geçtiğimiz yıllarda donanım teknolojilerindeki gelişmeler sayesinde büyük bir zorluk olmaktan çıkmıştır. Oysa yazılımsal hatalar dağıtık sistemlerde zorluk oluşturmaya devam etmektedir.

Yazılımsal hatalar daha ayrıntılı olarak şu şekilde tanımlanabilir [17].

- Sistemin duraklama hataları: Burada bahsedilen hatalar bir bilgisayarın işletim sisteminin donmasına benzeyen hatalardır. Bu hatalarda bir bileşenin sistemi duraklar ve bu duraklamayı ancak zaman aşımı süresi geçtiğinde öğrenmek mümkündür.
- Başarısız bileşen hataları: Bu şekilde olan hatalar bir bileşende karşılaşılan sorundan kaynaklanır ve diğer bileşenlere ağ üzerinden bildirim şeklinde bir mesaj gönderilir.
- Bileşen atlama hataları: Çoğunlukla ağ yönlendiricilerindeki yoğunluktan kaynaklanan mesaj gönderen ve mesaja yanıt bekleyen bileşene bildirimde bulunulmamasıdır.
- Ağ hataları: Ağdan kaynaklanan hatalardır.
- Alt ağ hataları: Bir ağda bulunan alt ağların arasındaki iletişimden kaynaklanan ve ağ hatalarına dayanan hatalardır.
- Zamanlama hataları: Ağdaki bileşenlerin senkron çalışmamasından kaynaklanan hatalar.
- Beklenmedik yapısal hatalar: Çeşitli programlardan kaynaklanan ve veri kaybına yol açan hatalardır.

4.3. Dağıtık Sistemlerin Mimarileri

Dağıtık sistemlerin temel çalışma prensibi birçok uygulamanın aynı işlem için birlikte koordinasyon içinde çalışmasıdır. Başka bir yaklaşım ise uygulamanın dağıtılabilir parçalara ayrılmasıdır [15].

Birçok uygulama temel olarak 3 katmana ayrılır.

- Sunum katmanı: Bu katman son kullanıcının kullandığı, uygulama ile veri alış verişini yapabildiği katmandır. Bu katmanda çalışan kod parçası son kullanıcının makinesiyle etkileşim içindedir.

- İş katmanı: Uygulamanın karar mekanizmalarının yer aldığı katmandır. Yazılımda iş kurallarını barındıran ve sağlanan hizmetin mantığının barındırıldığı katmandır.
- Veri katmanı: Bu katman sistem verilerinin saklandığı ve yönetildiği katmandır. Katmanların gerçekleştiriminin birbirinden düzgün bir şekilde ayrılması gerekir.

Dağıtık sistemleri geliştirirken çeşitli yazılımsal ve donanımsal mimariler kullanmak mümkündür. Bunlar istemci-sunucu, 3 katlı mimari, N katlı mimari, kümelenmiş mimari, noktadan noktaya mimaridir.

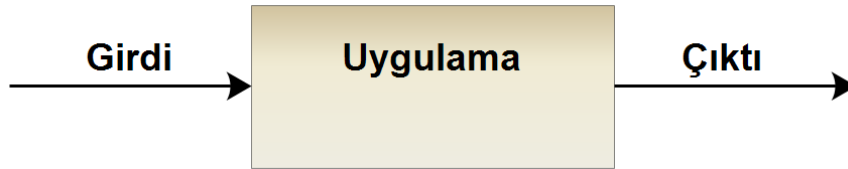
İstemci sunucu mimarisinde işlem yükünü sunucu taşımaktadır. Bu mimariye 2 katmanlı istemci sunucu mimarisi de denmektedir. İstemci sunucuyla bağlantı kurup gerekli veriyi alıp uygun biçimde kullanıcıya sunar. Sunucu ise iş kurallarına göre veriyi işleme ve istemciye iletmekle yükümlüdür. 3 katlı mimaride sunucuda iş katmanı tamamen ayrılmış durumdadır. Bu mimaride iş mantığı, sunum ve veri yönetimi katmanları bulunur ve iş mantığı katmanı birincil öneme sahiptir. Bazen uygulama sunucusu denilen ayrı bir cihaz iş mantığı katmanını işletir. N katmanlı mimari web uygulamalarının istemleri diğer iş servislerine yönlendirdiği uygulamalardır. Aslında 3 katmanlı veya daha fazla katmana sahip mimari tasarımlar N katmanlı mimari olarak adlandırılmaktadır [15]. Uygulama sunucularının kullanıldığı ve başarımın en yüksek olduğu ve aynı zamanda çoğu web uygulamasında kullanılan mimariler N katmanlı mimarilerdir. Kümelenmiş mimariler ise birden fazla makine tarafından paralel çalıştırılan işlemleri gerçekleştiren bir makine kümesinden oluşan mimaridir. Her cihazın hem sunucu hem istemci olabildiği tüm gereksinimlerin makinelere eşit paylaştırıldığı mimariye ise noktadan noktaya mimari denir.

5. BAĞLAM BİLİNÇLİ UYGULAMALAR

İnsanların birbiriyle iletişimde özel durumlarıyla ilgili bilgi vermeleri sık rastlanan bir durumdur. Bu da iletişimi güçlendirir, iletilen bilginin daha anlaşılır olmasını sağlar. Ancak böyle bir durum insanların bilgisayarlarla etkileşiminde söz konusu değildir. Dolayısıyla da bilgisayarlar insanların insanlarla iletişimde olan bağlam bilincine erişemeyeceği için iş süreçlerinde bağlam bilgilerini kullanma şansı olmaz. Oysa bilgisayarların insanlarla ilgili durum bilgilerine ulaşması, etkileşimi kuvvetlendirecek ve daha kullanışlı sonuçlar üretilmesine sebep olacaktır [18].

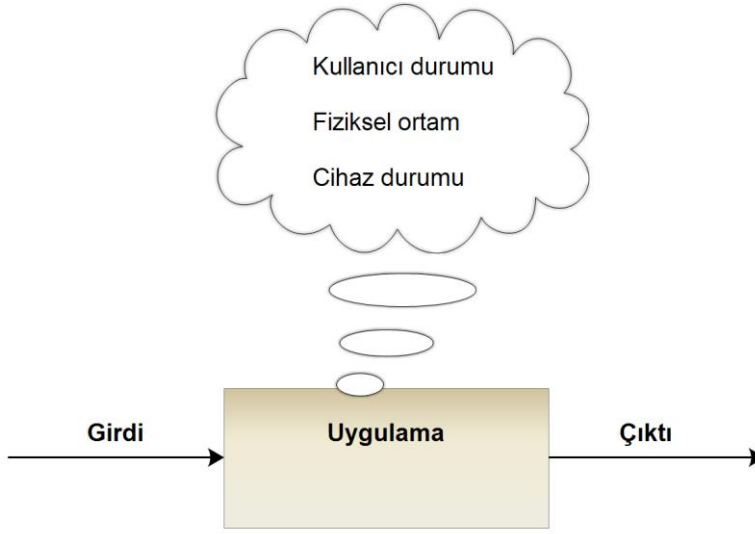
Mobil cihazlar yaygınlaşmaya başladığından beri, mobil cihazların dinamik özelliğinden dolayı kişilerin bulunduğu ortam bilgileri çabuk değişmektedir. Bu değişim mobil uygulamaların bu ortam bilgisinden faydalanabilmesi ihtimalini de beraberinde getirmiştir [19].

Geleneksel uygulamalar Şekil 5.1’ deki gibi herhangi bir bağlamsal bilgiyi kullanmadan süreçlerini tamamlarlar.



Şekil 5.1. Geleneksel bilgisayar uygulamaları veri akışı

Oysa bağlam bilinçli uygulamalar daha gelişmiş servisler sunabilmek için bağlam bilgisini edinmeye çalışırlar. Bu tür uygulamalar değişkenliğe çözülmesi gereken bir problem gibi bakmaktan çok bunu avantaja dönüştürmeye çalışır. Bu sebepten de kişiyle uygulama arasındaki etkileşimin iyileştirilmesi için uygulama bağlamsal bilgileri edinmeyi amaçlar [19]. Şekil 5.2’ de görüldüğü gibi bağlamsal bilgiler uygulamanın ortamından edinilir ve dışarıdan alınan girdi bilgilerine ek olarak bu bilgiler değerlendirilir.



Şekil 5.2. Bağlam bilinçli uygulamalar

Bağlam bilgisini uygulamanın algılamasını sağlamak için ilk akla gelen yöntem bağlam bilgisini kullanıcıdan elde etmektir. Ancak bağlamı bu şekilde elde etmek için kullanıcıdan belirli bir duruma karşı tüm bilgileri iletmesi beklenirdi. Ancak bu da bağlam bilinçli uygulamaların amacı olan bilgisayar insan arasındaki etkileşimi kolaylaştırmak amacına ters düşerdi. Ayrıca kullanıcıların bağlam bilgilerinin neler olduğunu algılaması ve bunları bilgisayara girmesi hem zahmetli hem de zor olurdu. Bunun yerine Dey ve Abowd [18] bağlam farkındalığı olan uygulama geliştirirken bağlam bilgisinin otomatikleştirilmiş bir şekilde toplanması, bu bilginin uygulamanın çalışma ortamında kullanılabilir durumda olması ve uygulama geliştirme aşamasında gereken bilgilerin kullanılması yaklaşımını önermişlerdir.

5.1. Bağlam

Birçok kişi bağlam kavramını anlamasına rağmen bu kavramı açıklamakta zorlanırlar. Bunun yerine bundan önceki örneklerle açıklar veya bağlamı tanımlamak yerine eş anlamlı kelimeleri kullanırlar.

Bağlam bilinçli kavramının kullanıldığı ilk çalışma Shilit ve Theimer [20] tarafından 1994 yılında yapılmıştır. Bağlam kavramını konum, çevredeki insanlar ve nesnelerin

kimlikleri ve nesnelerdeki deęişimler olarak tanımlamışlardır [18]. Brown ve arkadaşları [21], bağlamı yer, kullanıcının etrafındaki insanlar, saat, mevsim ve ısı gibi niteliklerle tanımlamıştır. Ryan, Pascoe ve Morse [22] önceki bağlam tanımlarına ek olarak kimlik özelliğini de bağlam çeşidi olarak belirtmiştir. Dey [3] ise bağlamı bir kişinin fiziksel, duygusal sosyal ve zihinsel ortamı olarak tanımlamıştır. Ancak bu tanımların uygulanması zordur.

Bağlam tanımlarına farklı bir bakış açısı da bağlamın uygulama ortamıyla ilgili olduğudur. Bazı tanımlar uygulamayı kullanan kişi referans alınarak yapılmıştır.

Shilit tarafından yapılan bir tanıma göre bağlam kişilerin nerede, kiminle olduğu ve yakınındaki kaynakları tanımlar. Bağlamı devamlı deęişen bir yürütme ortamı olarak tanımlarlar. Bağlamı şu ortamlara indirgeyerek tanımlarlar [18]:

- Bilgi işleme bağlamı; kullanıcının kullanımına uygun olan görüntü ve girdi aygıtları, kullanılabilir durumdaki işlemciler, ağ kapasitesi, bağlantı bilgileri ve hesap maliyetleridir.
- Kullanıcı bağlamı; kişinin bulunduğu konum, etrafındaki insanlar ve sosyal durumudur.
- Fiziksel bağlam; aydınlatma ve ortamdaki gürültü seviyesidir.

Zaman bağlamı da birçok çalışmada bağlam bilgilerinden biri olarak varsayılmıştır [23]. Zaman bağlamını dördüncü bir kategoride tanımlamak gerekir.

- Zaman bağlamı; o andaki zaman, hafta, ay ve yıla ait parametrelerdir [24].

Pascoe[24], bağlamı belirli bir varlıkla ilgili fiziksel ve kavramsal alt kümeler olarak tanımlamıştır. Bu tanım fazla spesifiktir. Bağlam, bir uygulamanın ve uygulamanın bütün kullanıcılarının bütün durumlarıyla ilgilidir. Bir durumun önemli özelliklerini sıralamak doğru olmayacaktır, çünkü önem arz eden parametreler duruma göre

değişebilir. Bazen fiziksel ortam önemliyken, bazense tamamen gereksiz olmaktadır [18].

Bu yüzden Dey [18] bağlamı aşağıdaki gibi tanımlamıştır:

“Bağlam, bir varlığın durumunu tanımlamak için kullanılan herhangi bir bilgidir. Bu varlık; bir insan, bir yer veya uygulama ve kullanıcı dâhil olmak üzere uygulama ile kullanıcı etkileşimiyle ilgili olan herhangi bir nesne olarak düşünülebilir.”

Bu tanım sayesinde bir uygulama geliştirilirken bağlamı ifade etmek daha kolay olacaktır. Eğer bir bilgi etkileşimde bulunan nesnelerden birinin durumunu karakterize etmede kullanılıyorsa, o zaman bu bilgiye bağlam adı verilir [18]. Örneğin; amacı bir hesap tablosuna çeşitli ağırlıktaki kalemleri yazmak olan bir uygulama var olsun. Bu uygulamadaki varlıklar kullanıcı ve uygulamanın kendisidir. Bağlam olup olmadığını anlamak için diğer insanlar ve bulunulan yer bilgileri yukarıdaki tanıma göre değerlendirilirse; odadaki diğer kişiler uygulamayı veya kullanıcıyı bu görevle ilgili etkilemeyecektir. Dolayısıyla diğer kişiler bağlam değildir. Ancak bulunulan yer kullanıcının durumunu karakterize etmekte kullanılabilir, bulunulan ülkeye göre ağırlık birimi değişebilir. Yani kişinin bulunduğu yer bir bağlam bilgisi oluşturmaktadır.

5.2. Bağlam Farkındalığı

Bağlam farkında programlar Shilit ve Theimer [20] tarafından “kullanıldığı yere adapte olan, yakınındaki kişi ve nesnelerin farkında olan ve zaman içerisinde o nesnelere göre değişim gösterebilen” yazılım olarak tanımlanmıştır. Bir sistem gerçekleştirmek durumunda olduğu işle ilgili olan bilgiyi ve/veya servisleri kullanıcıya sağlarsa o sistem bağlam farkındadır [18].

Bir bağlam bilinçli uygulamayı kullanıcıdan bilgi almasına veya komut çalıştırmasına, işin otomatik veya elle çalıştırılmasına göre sınıflandırmak mümkündür [25].

- Yaklaşık seçme: Bağlam bilgileriyle ilgili kullanıcıdan bilgi alan sistemler bu yaklaşımı kullanır. Bu tür sistemlerde kullanıcıya uygun nesneler listelenir, kendisine daha yakın olan vurgulanır.
- Otomatik yeniden yapılandırma: Var olan bağlama göre otomatik olarak bilgi çeken uygulamalar bu yöntemi kullanır. Bağlam bilgisine göre uygun olan kaynağın seçimine dayanan sistemdir.
- Bağlamsal komut çalıştırma: Bağlamın değişimine göre değişebilen ve elle çalıştırılabilir servisler içerir. Kullanıcının bağlamına göre çeşitli komutlar çalıştırılmış olur.
- Bağlam tetikleme: Çeşitli koşul komutlarına göre ve var olan bağlama göre otomatik komutlar çalıştıran sistemlerdir.

5.3. Mobil Uygulamalarda Bağlam Farkındalığı

Mobil teknolojinin gelişmesi ve mobil cihazların yaygınlaşması ile birlikte uç kullanıcılara sunulan mobil uygulamalarda da artış ve gelişme yaşanmıştır. Gelişmiş görsel ara yüzleri ve kapasiteleri ile mobil uygulamalar görsel açıdan etkili ve daha kolay kullanıcı ara yüzlere sahip uygulamalar sunmaktadır. Bağlam farkındalığı olan uygulamalar geliştirilirse kullanıcının bilgi ihtiyacının azalacağı ve daha verimli sonuçlar üreten uygulamalar kullanabileceği fikrinden yola çıkılmıştır. Dolayısıyla mobil ortamda geliştirilen uygulamaların bir kısmı bağlam farkındalığı olan uygulamalardır. Bağlam farkındalığına sahip olabilmek için uygulamaların bağlamı ortamdaki edinmesi gerekir, böylece bağlam bilinçli uygulamalar geliştirilebilir. Ancak bağlam farkındalığı olan uygulamaların geliştirilmesi aşamasında bağlam bilgisinin yüksek kalitede elde edilmesi, dağıtık mimarilerde bu bilginin düzgün işlenmesi gereklilikleri ve gizlilik seçeneklerinin uç kullanıcı tarafından belirlenmesi gibi başa çıkılması gereken problemler mevcuttur.

Sinderen ve arkadaşları [26] mobil bir uygulamada bağlam için kullanıcı aktiviteleri, bulunduğu konum, GSM hücre bilgisi, hız ve yön, batarya gücü, boş zaman, uygun ağlar, favori yerler ve ulaşım şekli gibi parametreleri tanımlamışlardır. Şekil 5.3' teki

gibi bağlamı mobil uygulama için tanımladığımızda birçok parametreyi değerlendirmek mümkündür ancak burada önemli olan geliştirilecek uygulamaya göre, uygulamadaki karar aşamalarına faydası olabilecek bağlam parametrelerini kullanmaktır.



Şekil 5.3. Bağlam bilinçli uygulamalarda kullanılabilecek parametreler

Bir mobil uygulamanın bağlam farkında olması, mobil uygulama tarafından sunulan servislerin bağlam farkındalığına sahip olması anlamına gelmektedir.

6. PROFİL UYARLANABİLİR UYGULAMALAR

İnternet ortamındaki bilgi kaynaklarının artmasının etkisiyle İnternet kullanıcılarının aradığını bulması zorlaşmıştır. İstenen bilgiye ulaşmak için fazla sayıda sayfayı taramak gerekebilir. Bu durum masaüstü bilgisayar kullanıcıları için arama motorlarının gelişmiş özellikleri sayesinde kısmen aşılmaktadır. Ancak mobil cihazlar için büyük bir problem teşkil etmektedir. Daha kısıtlı kaynaklara sahip mobil cihaz kullanıcısı kişisel bilgisayar kullanıcısına göre bilgiyi taramak için daha fazla çaba sarf etmek zorundadır. İnternette dönen fazla sayıdaki sayfa, cihazın kısıtlı görsel ara yüzü, kısıtlı gücü ve kısıtlı internet hızı dolayısıyla hem zaman kaybına, hem de kaynakların erken tüketimine yol açmaktadır. Bu yüzden mobil cihazlarda çalışan uygulamaların profile uyarlanabilmesi ihtiyaç duyulan ve işlevselliği yüksek bir kavramdır.

İlk geliştirilen profil uyarlanabilir uygulamalar masaüstü bilgisayarlarda çalışacak şekilde modellenmiştir. Bilgisayardan web tarayıcıda yapılan http istemlerine dönen yanıtların kişiselleştirilmesi konusunda çalışmışlardır. Kişinin web’ de gezinti davranışlarını izleyerek veya doğrudan beğendiği kategorileri ve sayfaları belirtmesi yoluyla kişinin tercihlerini elde etmiş ve geribildirim yöntemiyle profil uyarlamayı ve profile uygun veri aktarmayı gerçekleştirmişlerdir. Bazı uygulamalarda sonuçlar profile uygunluk açısından sıralanarak iletirse de bazı uygulamalarda sadece profile uygun olan sonuçlar gönderilmektedir.

Profile uyarlanabilir uygulama, sistemin kişinin profiline uygun davranması, farklı profillere farklı sonuçlar döndürmesi demektir. Bunu sağlamak için sistemde kişinin profilinin saklanması gerekir. Daha sonra gerçekleştirilen adım ise her veri isteminde profile uygun verinin gönderilmesidir. Profil uyarlanabilir uygulamalarda, profili saklamak için yapılan çalışmaların bir kısmında vektör uzay modeli [27] kullanılmıştır. Bu modelde her doküman bir vektöre karşılık gelir, vektörün her terimi bir kelimenin ağırlığına denk gelir. Bu ağırlık da terim frekansı, doküman frekansı hesaplama yöntemi TF-IDF [27] ile hesaplanmaktadır. Profil bu vektörlerden oluşan, tek veya birden fazla vektör içerir. Profildeki vektör sayısı

çalışmanın tercihidir, bazı çalışmalar tek vektör içeren bir model tercih ediyorken, bazıları belirli sayıdaki vektörü profil olarak saklar. Profile uygun dökümanların birbirine benzerliği ise vektörlerin benzerliği aracılığıyla hesaplanır.

Chen ve Sycara'nın geliştirdikleri “Webmate” [28] uygulaması bir tarayıcıdan yapılan http istemine karşılık sunucudan dönen yanıtları uyarlanmış olarak döndürmektedir. Bu uygulamada geliştirilen yazılım bir ara birim gibi tarayıcı ve web sunucusu arasında çalışmaktadır. “Webmate” [28], sadece kişinin olumlu tercihlerini kullanarak profil uyarlamayı gerçekleştirir. Profile uygun veriyi ise sistem yoğunluğunun az olduğu bir saatte toplu bir şekilde gerçekleştirir. Böylece kişiye sunulacak profile uygun sayfalar önceden belirlenmiş olur. Bir kişi bir web sayfasını beğendiği zaman, o sayfayı profile ekler. Aslında profil, kişinin önceden tercih ettiği sayfaların bir uzantısıdır.

Olumlu ve olumsuz geri bildirimlerin her ikisinden de profili öğrenebilecek şekilde yapılan çalışmalar da mevcuttur. Widyantoro ve arkadaşlarının geliştirdiği Alipes [29], kişilerin adına ilgilenebilecekleri sayfalar için web’ i filtreleyen bir servistir. Aynı zamanda olumlu geri bildirimin dışında olumsuz geri bildirimlerden de profil uyarlamayı gerçekleştirmektedir.

Balabanovic’ in “Fab” [30] çalışmasında da vektör uzay modeli kullanılmıştır. “Webmate” [28] çalışmasından farkı birden fazla vektörden oluşan profil yerine, beğenilen sayfaların bileşimi tek bir profil tutulmaktadır.

Mobil cihazlarda profil uyarlama için yapılan çalışmalardan bir kısmı mobil turizm rehber cihazları için tasarlanmıştır. COMPASS [31] ve CRUMPET [32] bu uygulamalara örnek gösterilebilir. Bu uygulamalarda profil için kullanılan ilk veri mobil cihazı kullanan kişiden elde edilir. Kişiden alından geri bildirim göre profil uyarlaması gerçekleştirilir.

Mobil cihazlarda geliştirilen profil uyarlamalı uygulamalarda verilerin bağlama göre değişmesi, “Webmate” [28] uygulamasında olduğu gibi önceden hazırlanacak bir

benzerlik listesi oluřturmayı engellemektedir. Mobil cihaza gnderilecek verinin baēlama gre deēiřmesi, benzerlik hesabının btn veri kmesi ierisinde deēil, sadece baēlam bilinli veri zerinde yapılmasını gerektirmektedir.

7. GPS KULLANAN WEB SERVİS TABANLI UYARLANABİLİR MOBİL UYGULAMA

Gerçekleştirilen uygulama mobil cihazlar üzerinde çalışabilen, bağlam bilincine sahip, uyarlanabilir bir mobil uygulamadır. Uygulamaya kısaca ACMIS adı verilmiştir, ACMIS; “Adaptable Context-Aware Mobile Information Services” adının kısaltması olarak kullanılmıştır.

ACMIS platformu başlıca dört modülden oluşmaktadır; Sunucu Modülü, Mobil İstemci Modülü, Web İstemci Modülü, Veritabanı Modülüdür.

Sunucu Modülü; ACMIS platformunda gerçekleştirilen iş süreçlerinin iş mantığı kısmının çalıştığı yapıdır. Sunucu modülünde internet üzerinden, mobil veya web istemci uygulamalarından gelen istemleri karşılayan web servis gerçekleştirimi bulunmaktadır. Aynı zamanda Veritabanı ile iletişimin kurulduğu, gerekli sorgulama ve müdahale işlemlerinin yapıldığı modüldür. Sunucu modülü dağıtık olarak ağ üzerinde birden fazla sunucu ve veritabanında kullanıma açılabilir. Kaynak gerekliliğinde ve fiziksel olarak dağıtık bir yapı kurmak gerektiğinde sisteme yeni sunucu modülleri eklense bile dağıtık yapının esnekliği sayesinde uygulama tutarlılığı devam edecektir.

Mobil İstemci Modülü; ACMIS platformunun mobil cihaz kullanıcıları tarafından kullanılabilmesini sağlayan bir ara yüzdür. Aynı zamanda GPS üzerinden edinilen yer bilgilerini sunucu tarafına iletmekle yükümlüdür. Dolayısıyla mobil istemci, ACMIS’ in bağlam bilinçli olması için gereken girdi değerlerini sağlayan bir modüldür.

Web İstemci Modülü; web tarayıcılar tarafından çağırılıp çalıştırılabilen ve ACMIS uygulamasına kayıt olmak ve hesap yönetimi için kullanıcı ara yüzü, uygulamanın diğer yönetim işlemleri için ise yönetici ara yüzüne sahip bir modüldür.

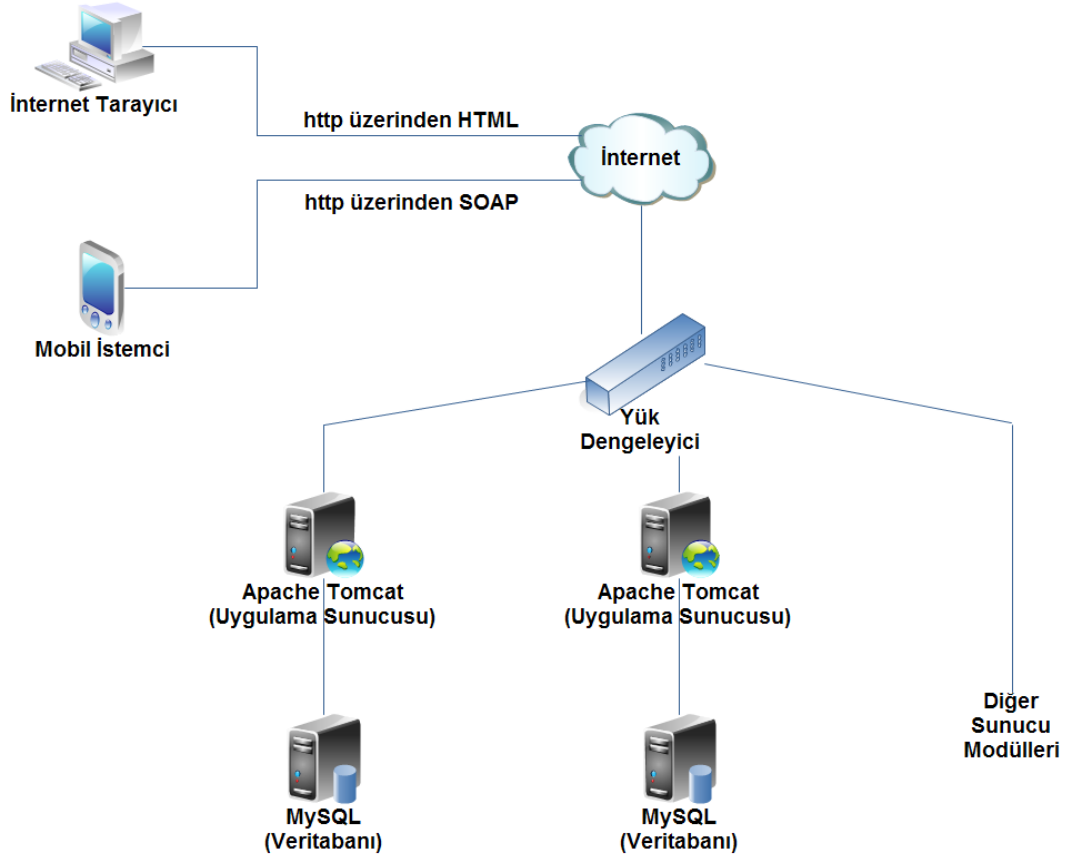
Veritabanı Modülü; ise kullanıcı profil bilgileri, etkinlik bilgileri, kullanıcı tercihleri ve uyarılama parametrelerini içeren ve veri yönetimi yapabilen kısımdır. Bu kısım sayesinde sistemde ilişkili veritabanı tablo yapıları oluşturulmuştur. Sistem yöneticisi tarafından müdahale edilebilir durumdadır.

Uygulamanın çalışması sırasında modüller birbiriyle anlık olarak haberleşebilmektedir. ACMIS platformunun kullanılması sırasındaki operasyonu aşağıdaki adımlarda tanımlamak mümkündür.

1. Web istemci modülünün kullanıcı ara yüzü kullanılarak hesap bilgileri oluşturulur. Hesap oluşturulurken kullanıcı için öncelikli anahtar kelimeler sisteme kaydedilir.
2. Mobil cihazdaki uygulamada GPS ve profil ayarları yapılır. GPS ayarları GPS cihazının mobil cihaza tanıtılması
3. Mobil cihazdaki istemci uygulaması çalıştırılarak o anki bağlam bilgilerine göre bir istem talebinde bulunulur.
4. Sunucu modülü bu talebi alır, veritabanı modülü ile birlikte çalışarak kullanıcının bağlam bilgileri, kullanıcı profil bilgileri ve kullanıcı tercihlerine göre verileri değerlendirir ve sonucu üretir .
5. Servis sonucu mobil istemci modülüne gönderilir ve sonuçlar kullanıcıya sunulur.
6. Mobil istemci uygulaması kullanıcıdan gelen bilgiyi değerlendirmesini bekler. Kullanıcının servisi kabul etmesi sonucunda bu bilgi sunucu modülüne aktarılır.
7. Sunucu modülü kullanıcı profilinde uyarılama işlemini gerçekleştirir, profil bilgilerini günceller, güncel profilin bir özetini mobil istemciye iletir.

ACMIS platformunun düzgün çalışabilmesi için, yukarıdaki adımlardan bağımsız bir şekilde veri girişinin sistemde yetkili kullanıcılar aracılığıyla yapılması gereklidir. Akabinde de sistemin mobil kullanıcılara faydalı sonuçlar üretebilmesi için, mobil istemci modülünden veya web istemci modülünden bilgi servislerine ait tercihlerle ilgili girişler yapılmalıdır veya bilgi servisi verilerinin güncellemesi yapılabilir.

7.1. Sistem Mimarisi

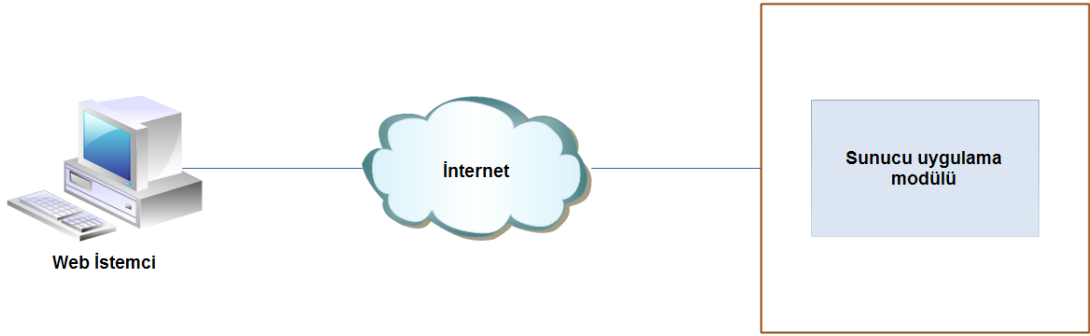


Şekil 7.1. Sistem mimarisinin genel şekli

Sistem mimarisi istemciler ve merkezi sunuculardan oluşmaktadır. Mimarideki her bir bileşen sistemin en az bir modülüyle ilgili işlevi yerine getirmektedir. Bileşenler hem bireysel olarak hem de birbiriyle problemsiz çalışabilir. Sunucu bileşenlerde hata kaydını tutmak ve ileriye yönelik performans analizi yapabilmek maksadıyla kayıt alt yapısı kurulmuştur. Ayrıca sunucularının dağıtık konumlanması sayesinde artan talep ve ihtiyaçlara göre yeniden düzenlenebilen bir sistem oluşturulmuştur. Böylece ölçeklenir bir sistem oluşturulmuş olur. Dağıtık bulunan sunucular arasındaki iş paylaşımı yük dengeleyici cihazlar tarafından yapılabilir.

7.1.1. Web istemci mimarisi

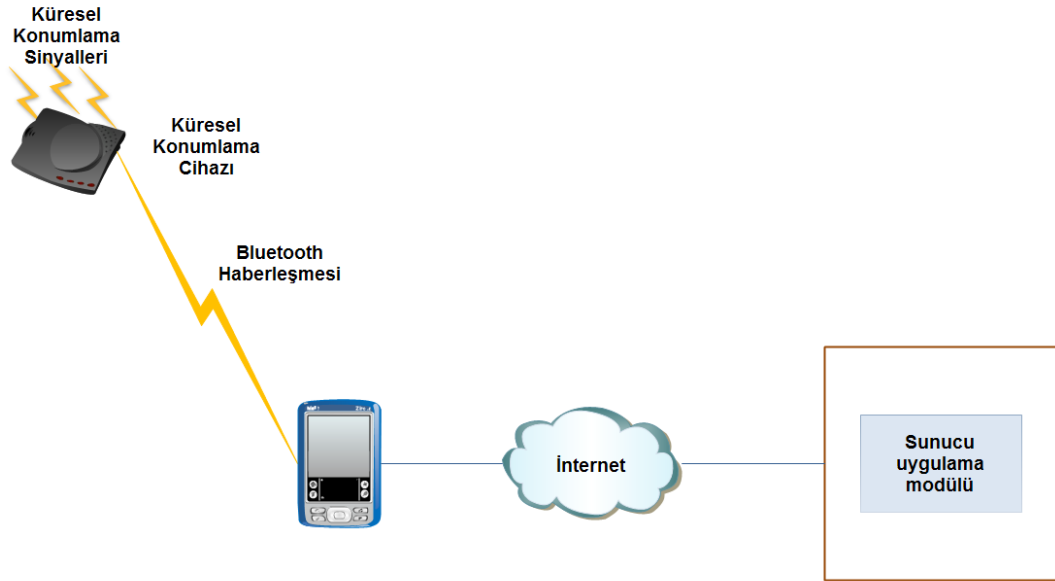
Web istemci modülü tarayıcı üzerinde çalışan, sunucu modülüyle HTTP protokolü üzerinden haberleşen bir yapıdır. Web istemci mimarisinde de görüldüğü gibi, web istemci uygulamasının kullanılabilmesi için internet bağlantısı ve tarayıcısı olan bir bilgisayarın kullanılması gerekmektedir.



Şekil 7.2. Web istemci mimarisi

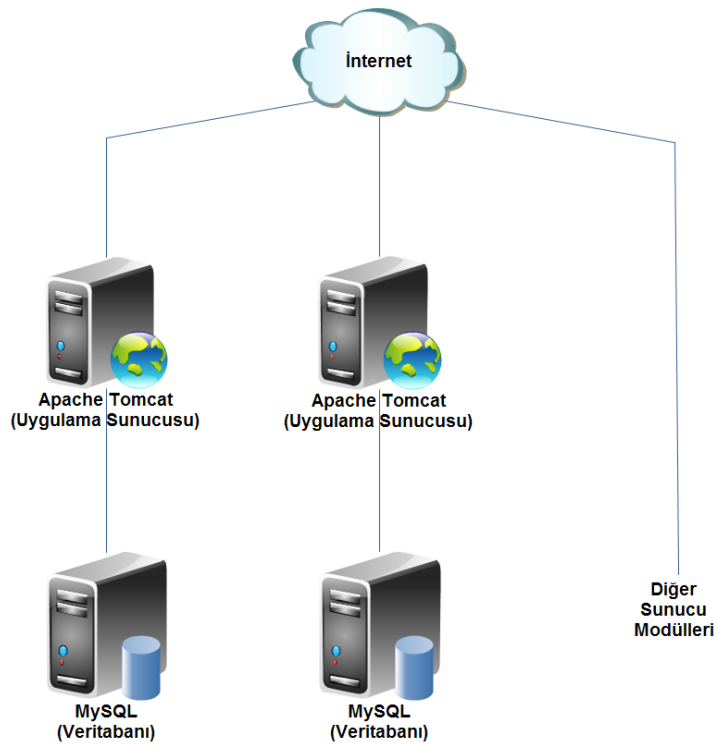
7.1.2. Mobil istemci mimarisi

Konum bilgilerini bluetooth aracılığıyla tanımlı GPS cihazından alan mobil istemci uygulaması, bu bilgiyi ayrıştırdıktan sonra cihazda bulunan profil bilgileriyle birlikte sunucu modülüne aktarır. Mobil istemci ile sunucu modülü arasındaki haberleşme HTTP protokolü üzerinden SOAP kurallarına göre gerçekleşmektedir. Sunucu modülü değerlendirme işlemini yaptıktan sonra aynı şekilde HTTP protokolü kullanarak mobil istemci uygulamasıyla haberleşir.



Şekil 7.3. Mobil istemci mimarisi

7.1.3. Sunucu mimarisi



Şekil 7.4. Sunucu mimarisi

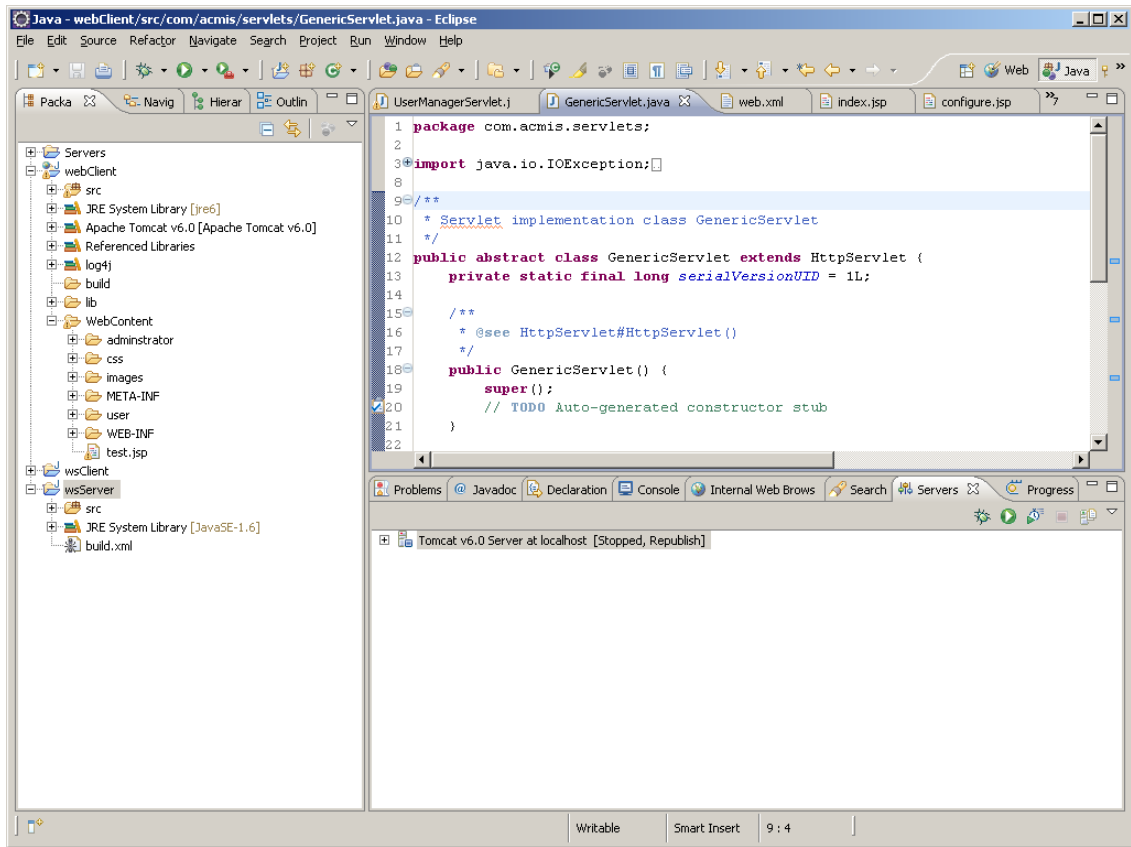
Sunucu mimarisi dağıtık yapıdadır. Fiziksel olarak ayrı yerlerde olma ihtiyacından veya aynı yerdeki fazla kaynak ihtiyacından dolayı dağıtık yapı kurulabilir. Dağıtık sistemdeki her bir sunucu modülü bir Apache Tomcat [33] uygulama sunucusu ve sunucunun haberleşebildiği bir MySQL [34] veritabanı sunucusundan oluşmaktadır.

7.2. Sistemin Modülleri

7.2.1. Sunucu modülü

Sunucu modülü bir web servis uygulamasından ve servlet [35] uygulamasından meydana gelmektedir. Modülün yaptığı başlıca iş; istemcilerden gelen istek taleplerini değerlendirip, uygun modüllerle iletişime geçerek bir yanıt oluşturmak ve oluşturulmuş yanıtı dönmektir.

Sunucu modülü Eclipse [36] ortamında Java programlama diliyle geliştirilmiştir. Eclipse aracının web uygulamaları geliştirmek için kullanılan özel bir versiyonu kullanarak mobil istemciyi karşılayan sununu kısmı ile web istemciyi karşılayan sunucu kısımları gerçekleştirilmiştir.



Resim 7.1. Eclipse, sunucu modülünün geliştirildiği ortam

Web servis teknolojilerinde, önceden de belirtildiği gibi, servisi kullanacak mobil istemciler açısından servisin hangi dille geliştirildiği önem arz etmemektedir. Web servisi karşılayan kullanıcı uygulaması mobil bir uygulamadır. Mobil uygulamaların marka ve model bağımlılığı mevcuttur. Bu sebeple farklı kişiler tarafından, farklı şekillerde geliştirilen ve sunucu modülünün, mobil istemci için yazılan web servisini kurallara uygun olarak çağıran yazılımlar, sisteme kolayca entegre edilebilir. Dolayısıyla sisteme gerektiği takdirde sonradan yaratılacak farklı platformlarda çalışan mobil istemci uygulamaları kolayca entegre olabilir. Böyle bir durumda yapılması gereken web servisine, servis tanımında belirtildiği şekilde bağlantı kurulmasıdır.

Sunucu modülü iki kısımdan oluşmaktadır. Bu kısımların her ikisi de Java platformunda geliştirilmiştir. Bu kısımlar mobil istemci için web servis uygulaması ve web istemci için servlet uygulamasıdır.

Web servisi

Web servis uygulaması mobil istemciden yapılan HTTP protokolü ile gelen istekleri karşılar ve bağlam bilinçli, kullanıcı profiline uyarlanmış yanıtlar döner. ACMIS sisteminde geliştirilen web servis Java platformunda web servis teknolojisi standartlarına uygun şekilde geliştirilmiştir. Web servisin veritabanı modülü ve mobil istemci modülü ile arasında etkileşim bulunmaktadır.

ACMIS platformu kullanılmaya başlandığında mobil istemciden gelen ilk isteği web servis uygulaması karşılar. İstemciden gelen profil bilgisine ve bağlam bilgilerine göre bilgi servislerini tarar. Bağlama göre elde edilen sonuçlar mobil istemciye gönderilmeden önce profile uygunluğu kontrol edilir. Profilde henüz eski tercih bulunmuyorsa, sadece profil etiketlerine göre bir sıralama yapılır ve sonuç mobil istemciye iletilir.

Sunucu modülünün profil adaptasyonu kısmında, Webmate [28] uygulamasındakine benzer bir yaklaşım izlenmiştir. ACMIS uygulamasında sunucudan mobil istemciye iletilen bilgi bağlam bilinçli etkinlik bilgisi olduğu için, profil kümesinde bağlam bilinçli etkinlik kayıtları tutulacaktır. Ancak benzerlik metodu Webmate [28] çalışmasında olduğu gibi kosinüs benzerliği ile hesaplanmıştır.

ACMIS uygulamasında profil uyarlama işlemi şu şekilde çalışmaktadır: Kişinin profilini P , profilindeki beğendiği N adet etkinlik olduğunu, incelemekte olduğu ve beğendiği etkinliğin ise P_i olduğunu varsayılırsa;

1. Eğer $|P| < N$ ise ($|P|$ profilde beğenilen kategori sayısıdır), $P \leftarrow P \cup P_i$
2. $|P| = N$ ise, $P \leftarrow P \cup P_i$
 - 2.1. Profil kümesi P' deki her iki etkinliği birbiriyle kıyasla.
 - 2.2. Birbirine en çok benzeyen iki etkinliği bul. Bu iki etkinliği tek etkinlik altında birleştir.

Yukarıdaki algoritma kullanıcı mobil istemci ara yüzünden “Kabul” adımını tercih ettiği anda sunucu tarafına bir istemle birlikte gitmekte ve çalışmaktadır. Çalıştıktan sonra yeni üretilen profil dosyası XML [11] biçiminde mobil cihaza iletilmektedir. Profil XML dosyasının bir örneği aşağıda görülmektedir.

```
<profile distanceTolerance="0,5" maxCategoryCount="4"
profile_id="1">
<Item>
<id>11</id>
<summary>konser pop muzik</summary>
<name>Funda Arar Konseri</name>
<detail>Funda Arar,zamanin eli albumu ile 13. ifa yilin en iyi
albumu odulunu aldigi Bostanci Gosteri Merkezi sahnesinde bu kez
sevenleriyle bulusuyor</detail>
</Item>
<Item>
<id>34</id>
<summary>imza kitap roman</summary>
<name>Can Dünder, Imza Günü</name>
<detail>Can Dünderin Mustafa kitabı imza günü</detail>
</Item>
<Item>
<id>20</id>
<summary>sanat dans muzikal</summary>
<name>Troia </name>
<detail>Bati uygarligina kaynaklik etmis, dünya edebiyatinin odagi
olan TROYA ilk kez anayurdundan dünyaya atiliyor. Efsanevi Dans
Topluluğu Anadolu Atesinden ynzylın dans projesi... Bir Anadolu
Efsanesi Anayurdunda Canlanıyor. </detail>
</Item>
<Item>
<id>13</id>
<summary>esya taki promosyon alisveris</summary>
<name>IKEA Firsat Günüleri </name>
<detail>21.Ocak.2010 tarihine özel IKEA geçerli kampanya firsati
Ikea Borje Sandalye IKEA Umraniye mağazasında 99TLlik etiket fiyatı
yerine süre ve stoklarla sinirli kampanya kapsamında KDV Dahil
39TLden satısa sunuluyor. </detail>
</Item>
</profile>
```

Profil uyarlamadaki benzetim algoritması yukarıdaki XML dosyasında görülen <summary> ad alanı içerisinde bulunan etiketlere göre çalışmaktadır. Ortak etiket sayısı en çok olan iki etkinlik birbirine en çok benzeyen etkinlikler olarak değerlendirilir.

Mobil istemcinin sunucu tarafının web serviste gerekleřtirimi, mobil cihaz uygulamalarının karakteristięi aısından faydalıdır. Web servisin servis szleřmesi olan WSDL dosyasındaki kurallara uyarak, web servise baęlantı kurulabilir. Bu sayede yeni zelliklere sahip cihazlara, var olan sistemin iřlerlięini etkilemeden ve kesinti yařatmadan, kullanıcı ara yzleri yazılıp sisteme kolayca entegre olması imkanı saęlanmıřtır.

Servlet

Servlet [35] Java ortamında geliřtirilen zel program paralarına verilen isimdir. Servlet yapılarını Java nesneleri gibi deęerlendirmek de mmkndr. Sunucu tarafında Java kabında alıřır ve dinamik ierik retirler. Web istemcileriyle HTTP protokol zerinden haberleřir, rettikleri sonucu da aynı řekilde HTTP zerinden geri gnderirler. Servlet nesneleri kullandıkları uygulama ktphanesi sebebiyle hantal olmayan uygulamalar olarak deęerlendirilir ve Java programlama dilinin bir ok avantajından faydalanırlar.

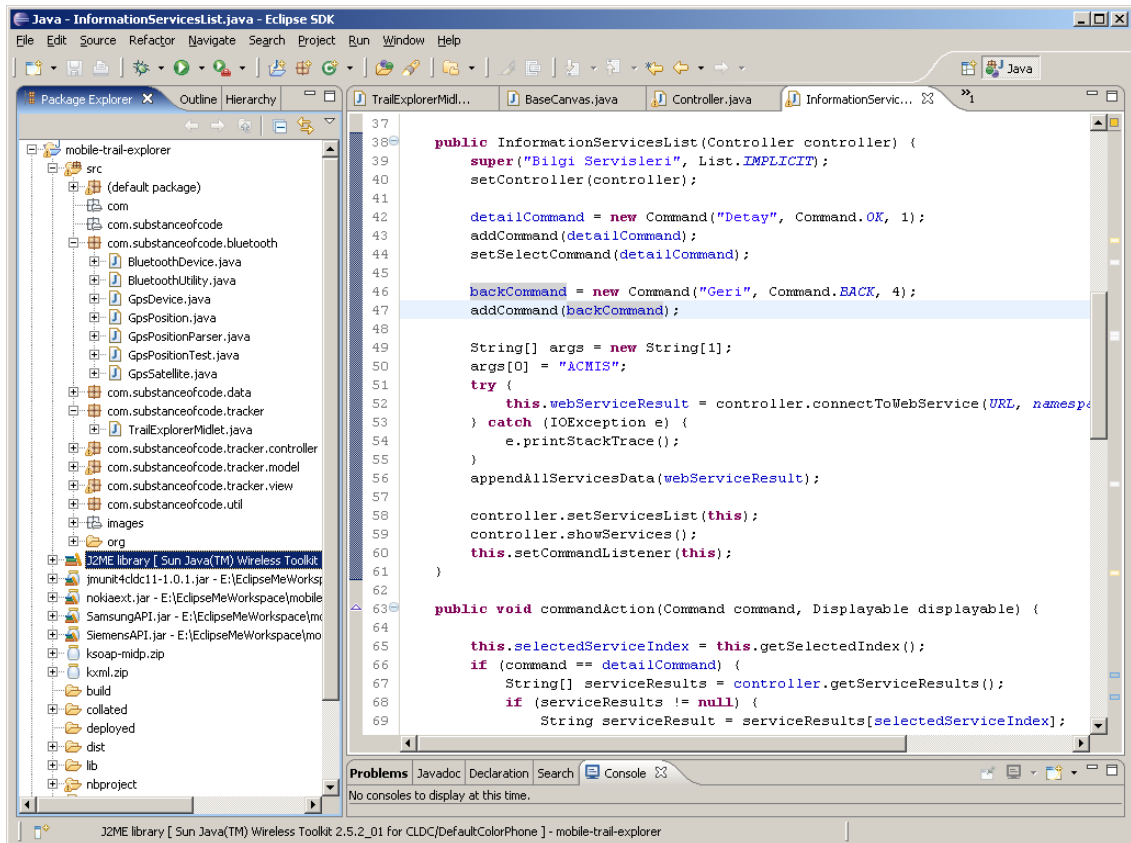
ACMIS uygulamasında gerekleřtirilen servlet, web istemci modlnden gelen isteklere yanıt verebilecek řekilde tasarlanmıřtır. Bylece mobil istemciden gelecek isteklerle web istemciden gelecek isteklerin gerekleřtirimi farklı tutulur.

Web istemci uygulamasının sunucu tarafı Servlet olduęu iin ACMIS sistemini ilk kullanmaya bařlayan kiřinin sisteme kayıt ařamasının iř akıřları da Servlet yapısında gerekleřtirilmiřtir.

7.2.2. Mobil istemci modl

Mobil istemci modlnn temel iřlevi dinamik yer bilgisini elde edip sunucu modlnde gerekleřtirimi bulunan web servis uygulamasına eriřerek, mobil cihaz kullanıcısının ACMIS platformunu kullanmasını saęlamaktır. Mobil istemci modl aynı zamanda profil bilgisinin bir kısmını bulundurmaktadır.

Mobil istemci modülü mobil cihazlarda uygulama geliştirmek için tasarlanmış J2ME(Java Platform Micro Edition) platformunda Eclipse ortamında geliştirilmiştir. J2ME Java ürünlerinden taşınabilir cihazlar için üretilmiş, sistem gereksinimleri kişisel bilgisayarlarda kullanılan Java çalıştırma ortamına göre daha az, başarımı yüksek olan bir ortamdır.



Resim 7.2. EclipseME, mobil istemci modülünün geliştirildiği ortam

Mobil istemci GPS ile haberleşerek global konum bilgilerini elde edip bunları istem talebiyle birlikte sunucu modülüne ulaştırır. GPS bilgileri ve profil parametreleri kullanılarak ACMIS platformunda bağlam bilinçli sonuçlar üretilir. Mobil istemci modülü sunucu modülünden gelen yanıtları görsel ara yüzü sayesinde mobil cihaz kullanıcısına gösterir.

GPS verisinin alınması

Harici küresel yer belirleme cihazı bluetooth bağlantısı üzerinden aranarak bulunur ve bağlantı kurulur. Bluetooth ile bağlantı kuran Java sınıfında öncelikle başlangıç parametreleri ilklenir. Böylece bluetooth servisi kullanılmak üzere hazır hale gelmiş olur. Bu akışı özetleyen kod parçası aşağıdaki gibidir. Akışta kullanılan Java metotlarının detayları da belirtilmiştir.

```
public void searchDevices() {
    try {
        BluetoothUtility bt = new BluetoothUtility();
        logger.log("Initializing bluetooth utility", Logger.DEBUG);
        bt.initialize();
        bt.findDevices();
        while (bt.searchComplete() == false) {
            Thread.sleep(100);
        }
        devices = bt.getDevices();
    } catch (Exception ex) {
        System.err.println("Error in Controller.searchDevices: "
            + ex.toString());
        ex.printStackTrace();
    }
}

public void initialize() throws BluetoothStateException {
    localDevice = null;
    discoveryAgent = null;
    // Retrieve the local device to get to the Bluetooth Manager
    localDevice = LocalDevice.getLocalDevice();
    // Servers set the discoverable mode to GIAC
    localDevice.setDiscoverable(DiscoveryAgent.GIAC);
    // Clients retrieve the discovery agent
    discoveryAgent = localDevice.getDiscoveryAgent();
}
```

GPS verisini alabilmek için bluetooth ile ulaşılabilir durumdaki cihazlar aranır.

```
public void findDevices() {
    try {
        discoveryAgent.startInquiry(DiscoveryAgent.GIAC, this);
        searchComplete = false;
    } catch (BluetoothStateException ex) {
        Logger.getLogger().log("Error in
BluetoothUtility.findDevices: " + ex.toString(), Logger.ERROR);
        ex.printStackTrace();
    }
}
```

Bir cihaz bulunduğu aşağıdaki metot çağrılır. Bir cihazın birden fazla bulunma olasılığından dolayı, bulunan cihazın önceden listeye eklenip eklenmediği kontrol edilir.

```
public void deviceDiscovered(RemoteDevice remoteDevice, DeviceClass
deviceClass) {

    // same device may found several times during single search
    if (devices.indexOf(deviceClass) == -1) {
        String address = remoteDevice.getBluetoothAddress();
        String name = null;
        try {
            name = remoteDevice.getFriendlyName(false);
        } catch (IOException ioe) {
        }
        if (name == null || name.trim().length() == 0) {
            try {
                name = remoteDevice.getFriendlyName(true);
            } catch (IOException ioe) {
            }
            if (name == null || name.trim().length() == 0) {
                name = address;
            }
        }
        Logger.getLogger().log("Device found: " + name + " (" +
address + ")", Logger.INFO);
        BluetoothDevice dev = new BluetoothDevice(address, name);
        devices.addElement(dev);
    }
}
```

Cihazın bulunmasından sonraki aşama uygun erişim protokolünü ve Java sınıflarını kullanarak bağlantıyı sağlamaktır. Cihaza bir bağlantı kurulduktan sonra cihazın ürettiği GPS verisi uygulamaya aktırılmış olur.

```
public synchronized void connect() throws IOException {
    connection = (StreamConnection) Connector.open("btspp://" +
this.getAddress() + ":1",
Connector.READ);
    reader = new InputStreamReader(connection.openInputStream());
    thread = new Thread(this);
    thread.start();
}
```

Cihazdan GPS verisi alındıktan sonra mobil istemci uygulamasının bu veriyi kullanılmak üzere sunucu modülüne ulaştırması gerekir. Ancak ulaştırmadan önce veri üzerinde ayrıştırma işleminin yapılması gerekir. GPS cihazından veri geldikçe tekrarlı olarak çalışan bu metot, gelen ham verinin ayrıştırma işlemini gerçekleştirmektedir.

```

private synchronized void parseGPGSV(String record) {
    String[] values = StringUtil.split(record, DELIMETER);

    final short cyclePos = Short.parseShort(values[6]);
    if (cyclePos == 1) {
        copyLastCycleSatellitesAndClear();
    }

    int index = 4;
    while(index+3 < values.length){
        short satelliteNumber = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short elevation = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short azimuth = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short satelliteSnr = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        if(satelliteNumber != GpsSatellite.UNKNOWN){
            final GpsSatellite sat = new
GpsSatellite(satelliteNumber, satelliteSnr, elevation,
azimuth);
            this.tempSatellites.addElement(sat);
        }
    }
}

```

Sunucu modülüne erişim

Mobil istemcinin web servise erişmesi normal bir web uygulamasının web servis erişiminden farklıdır. Web uygulamaları SOAP aracılığıyla web servis istek ve yanıt süreçlerini gerçekleştirebilir. Oysa J2ME ortamında geliştirilen uygulamaların SOAP mesajlarını otomatik oluşturma ve bölme yetenekleri kısıtlı kaynaklar sebebiyle bulunmamaktadır. SOAP mesajlaşmayı yönetmek mobil uygulamalar için karmaşıklığı ve işlem yükü fazla olan bir iştir. Bu yüzden web servislerin mobil cihazlar aracılığıyla çağrılması için var olan çözümlerden biri kSoap2 [37] kütüphanesini kullanmaktır.

```

public String connectToWebService(String url, String nameSpace,
String method, String urn, String[] args) throws IOException {
    SoapObject soap = new SoapObject(nameSpace, method);
    for (int i = 0; i < args.length; i++) {
        soap.addProperty("arg" + i, args[i]);
    }
    SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.bodyOut = soap;

    HttpTransport ht = new HttpTransport(url);

```

```

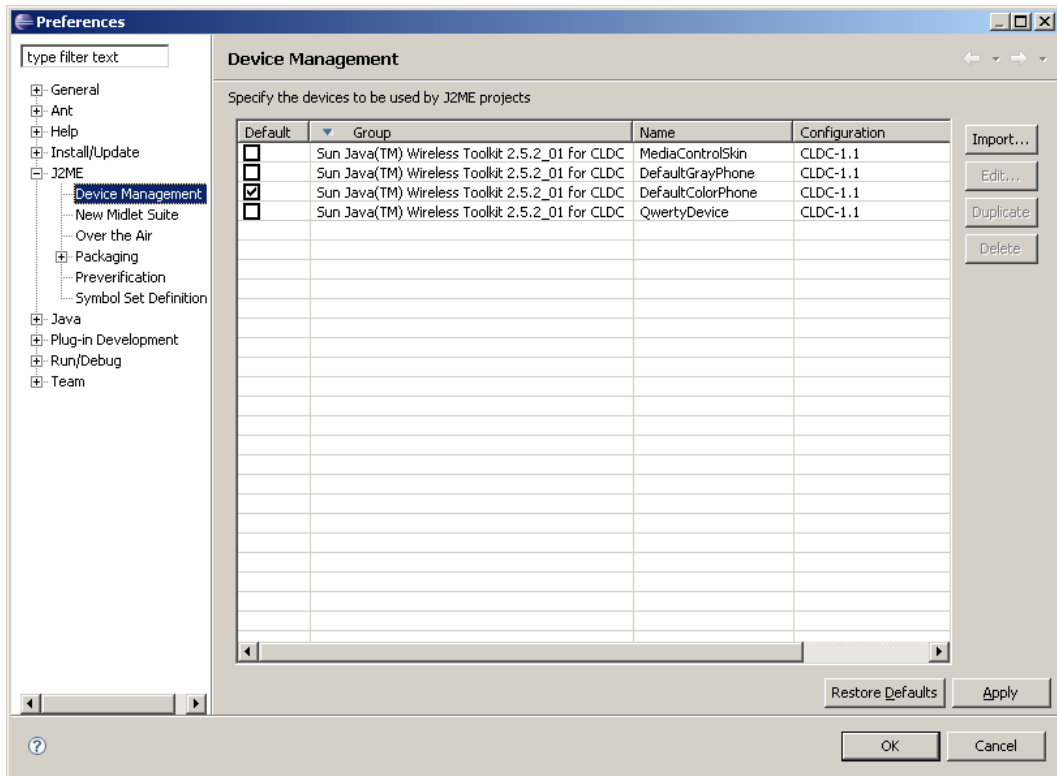
try {
    ht.call("", envelope);
} catch (Exception e) {
    e.printStackTrace();
}
Object soapIn = envelope.getResponse();
return soapIn.toString();
}

```

Mobil istemci ara yüzü

Mobil istemci ara yüzü J2ME kütüphaneleri kullanılarak üretilmiştir. J2ME’ de tanımlanmış mobil cihazlar için grafik ara yüzü geliştirmeye yarayan MIDP (Mobile Information Device Profile veya kısaca MIDP) profili kullanılmıştır. Bu profili kullanarak üretilen ve mobil cihazda çalışan uygulamalara “MIDlet” denir.

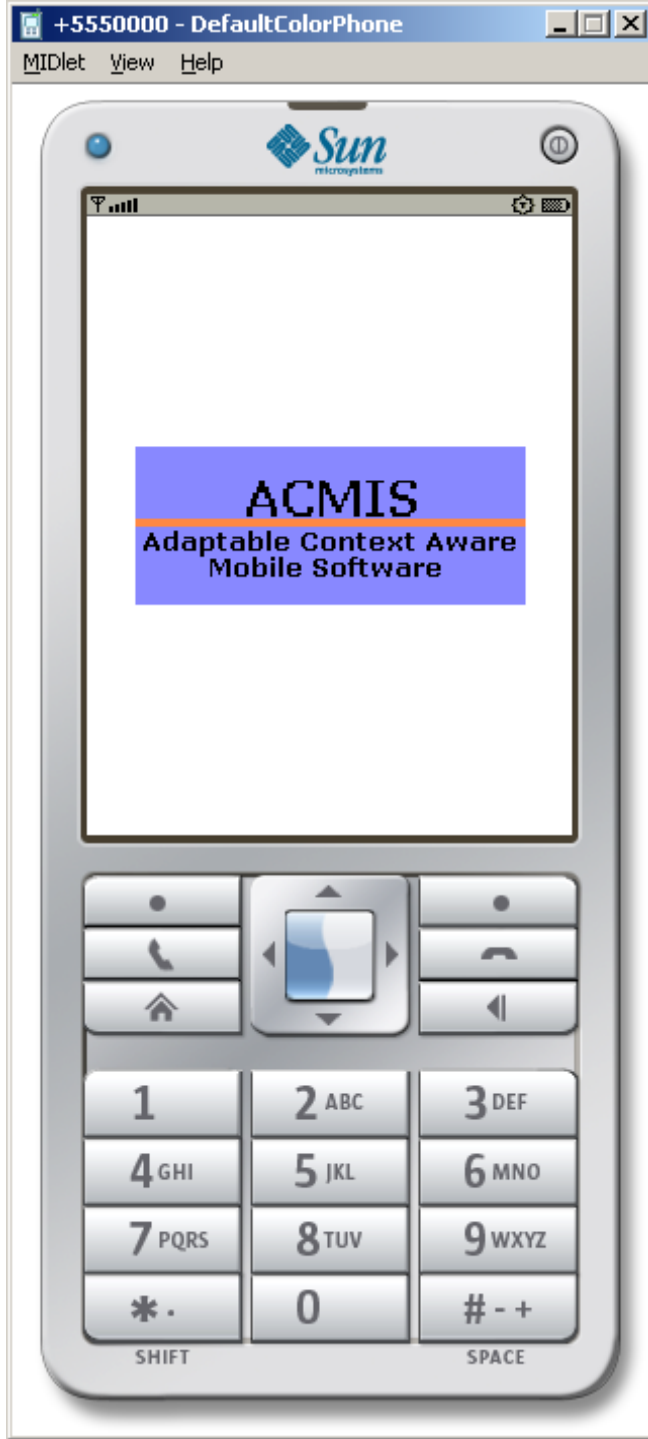
Verilen örnekler J2ME kablosuz platformunun içerisinde bulunan bir emülatör aracılığıyla alınmıştır. J2ME altyapısında dört tipte emülatör bulunmaktadır, ancak örneklerde Eclipse ortamı aracılığıyla kullanılan “DefaultColorPhone” adındaki emülatör kullanılmıştır.



Resim 7.3. J2ME emülatör ayarları

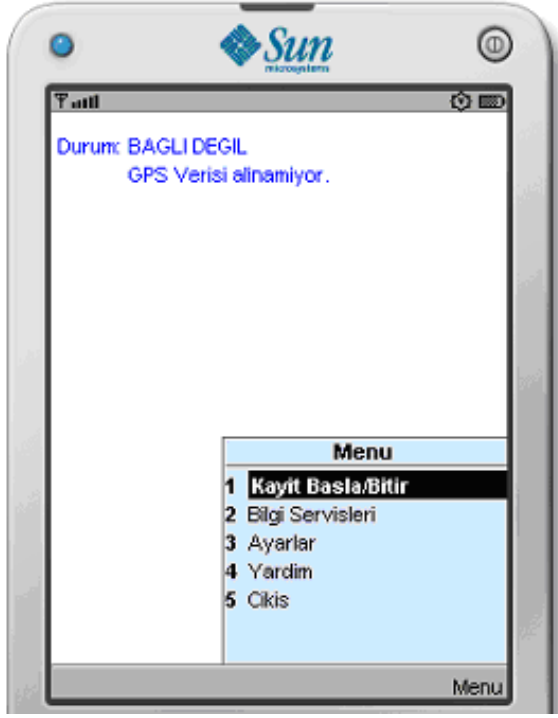
Mobil istemci arabirimi GPS cihazından bağlam parametrelerini, uygulamadan da gömülü profil bilgilerini edindiğinden herhangi bir veri girişine gerek kalmadan kullanıcı tarafından kullanılmaktadır. Kullanıcı mobil cihazının uygulamalar bölümünden ACMIS uygulamasına erişebilir.

Uygulama ilk başlatıldığı zaman uygulamanın başlangıç amblemi olan bir ekranla karşılaşılacaktır. Bu ekran herhangi bir tuşa basılmazsa altı saniye bekler, ancak bir tuşa basılması durumunda uygulama doğrudan karşılama ekranını açar. Karşılama ekranındaysa beş seçenekten oluşan bir menü bulunur.



Resim 7.4. ACMIS uygulaması karşılama ekranı

ACMIS mobil istemci uygulaması başlatıldığı zaman GPS cihazı ile bağlantı yoktur. Dolayısıyla uygulama açıldıktan sonraki ilk adım GPS cihazını başlatarak konum bilgilerini mobil istemci uygulamasının almasını sağlamaktır.



Resim 7.5. ACMIS uygulaması başlangıç ekranı

Ancak uygulamada tanıtılmış bir GPS cihazı bulunmuyorsa uygulama kullanılmaya başlandığı zaman öncelikle GPS cihazının tanıtılması ve profil ayarlarının yapılması gerekir. Bu ayarların yapılabilmesi için ana menüden “Ayarlar” menü adımı seçilmelidir. Ayarlar menü adımı hem GPS cihazı ile ilgili ayarlar, hem de profil ile ilgili ayarlar bulunur.



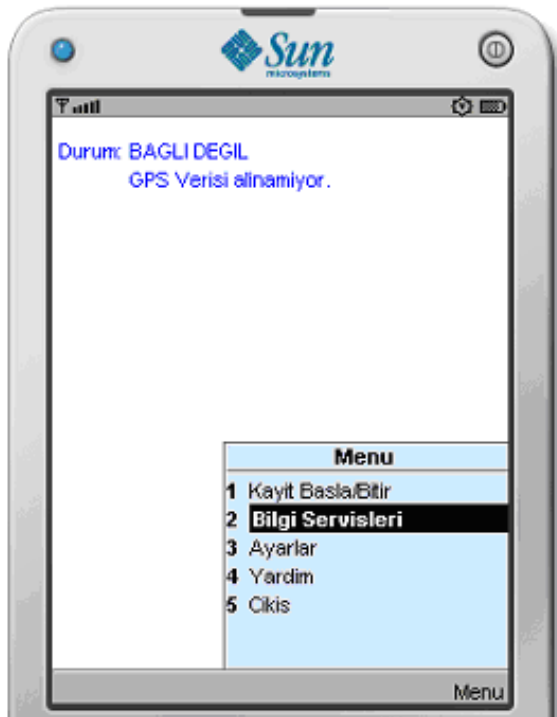
Resim 7.6. GPS cihaz ayarları ve profil ekranı



Resim 7.7. GPS cihazının bulunması ve ayarlanması ekranı

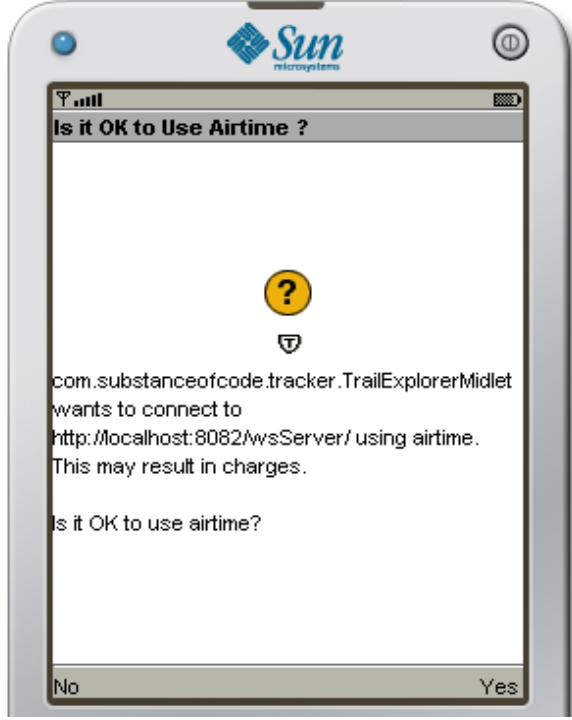
GPS adımı seçildiği zaman bluetooth bağlantısı aracılığıyla ulaşılabilir durumda olan cihazlar görüntülenir. Ardından ekrana gelerek kullanıcıdan seçmesi beklenir. Bu seçim yapıldıktan sonra uygulamada başka bir değişiklik yapılana kadar bu GPS cihazından konum bilgisi alınacaktır.

Uygulama GPS cihazına başarılı bir şekilde bağlandıktan sonra bilgi servisleri adımını kullanabilir. GPS cihazına bağlı değilse de sistemde tanımlı konum bilgisine göre bağlam parametreleri ilklenir. Uygulama ara yüzünden “Bilgi Servisleri” adımının tıklanmasıyla mobil ara yüz konum bilgileri ve profil bilgisini kullanarak, internet üzerinden sunucu modülünün web servis kısmıyla iletişim kurmak için işleme başlar.



Resim 7.8. Bilgi servisleri hizmetine giriş

Ancak uygulama mobil cihaz üzerinde kullanılıyorsa cihaz ayarlarıyla bağlantılı olarak, emülatörde kullanılıyorsa MIDP ve emülatör ayarlarıyla ilgili olarak bir uyarı ekranı çıkabilir. Bu ekran bir hata ekranı değil, ağa bağlanmak için uyarı ekranıdır.



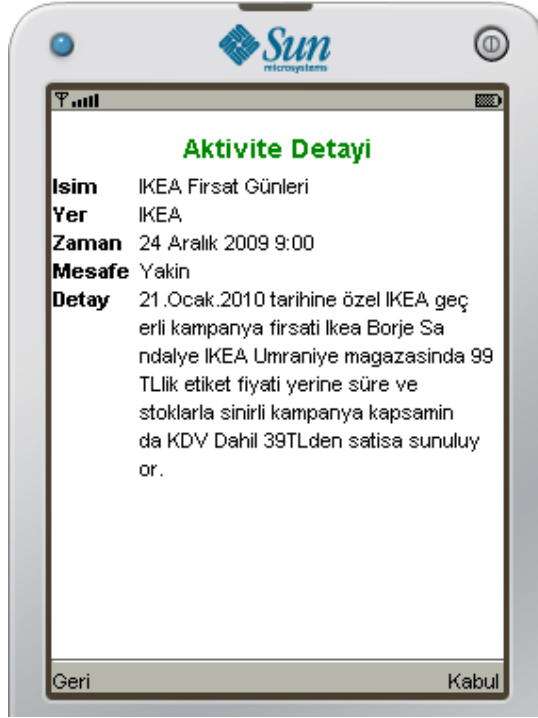
Resim 7.9. Ağa bağlanma uyarı ekranı

İnternet erişimine izin verildikten sonra sunucu uygulamasından gelen cevap ekrana yansıtılır. Yanıtlar, kullanıcının bağlam parametrelerine ve kişinin profiline göre sunucu tarafında hesaplanmış ve gönderilmiştir. Profil henüz yeni yaratılmışken, gelen bilgiler herhangi bir sıralamaya sokulmadan gönderilmiştir.



Resim 7.10. Bağlam bilinçli etkinlik sonuçları

Örnekte görüldüğü gibi mobil cihaz kullanıcısına bağlam bilgisine göre çeşitli etkinlik bilgileri verilmiştir. Verilen bu bilgilere göre kullanıcı bir alışveriş kategorisiyle ilgili bir etkinliği seçer ve “Detay” tuşuna basarsa, sunucu tarafına bu yaptığı seçim kaydedilir. Ve profil XML’ i istemciye tekrar iletilir. İstemcinin bir sonraki bilgi servisi isteminde ilk yaptığı tercih göz önünde bulundurularak bir sonuç döndürülecektir.



Resim 7.11. Etkinlik detay ekranı

Mobil cihaz kullanıcısı bu etkinliği kabul ettiği zaman, bu etkinliğin sahip olduğu etiketlerdeki diğer etkinlikler kullanıcının sonraki ACMIS yanıtlarında üst sıralarda gelir. Sunucu uygulamasının yanıtlarının kullanıcı tercihlerine göre adapte edilebilmesi için kullanıcının kabul veya tercihi veritabanına kaydedilir.

Kullanıcının bir sonraki bilgi servisi sorgulamasında ise bağlam bilinçli olarak, bundan önceki tercihindен etkilenen bir sonuç üretilir. Bu sonuca göre, gelen etkinlikler arasında önceden tercih ettiği alışveriş kategorisindeki başka bir etkinlik daha üst sıraya yerleşmiş olur. Örneğin; Resim 7.9’ da daha alt sıralarda görünen “Axe ve Magnum promosyonu” isimli etkinlik en üst sıraya yerleşmiştir.



Resim 7.12. Güncellenen profile göre yeniden istenen bilgi servisinin sonucu

7.2.3. Web istemci modülü

Web İstemci Modülü; web sayfalarından oluşan ve sunucu uygulama modülünün servlet kısmı tarafından karşılanan yapıdır. ACMIS platformunu kullanmak isteyen kişiler Web İstemci Modülü üzerinden uygulamaya kayıt olabilir ve profil bilgilerini ilgi alanlarına göre doldurmaları gerekmektedir. Web istemci modülü aracılığıyla sisteme kaydedilen profil bilgileri, sonradan mobil istemci modülü üzerinden yapılan profil ayarlarındaki bilgilerle örtüşmelidir. Web istemci modülü, web tarayıcı üzerinde çalışabilmesi sebebiyle hesap yönetimiyle ilgili mobil istemciye göre esnek bir arabirim sunmaktadır. Bu sebeple mobil istemci üzerinden mobil cihaz kısıtları sebebiyle kullanılamayan işlemleri veya mobil istemci tarafında olmayan işlevleri web istemci üzerinden yapmak mümkündür.

Web istemci modülü de sunucu uygulama modülü gibi web uygulamaları geliştirmek için özelleştirilmiş bir Eclipse ortamında geliştirilmiştir. Web istemci modülünü oluşturan web sayfaları sunucu modülünde Servlet tarafından karşılanmaktadır.

Modüldeki web sayfaları JSP (Java Server Pages) kullanılarak geliştirilmiştir. JSP sayfalarında dinamik ve statik içerik bir arada bulunmaktadır.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-9"
    pageEncoding="ISO-8859-9"%>

<!-- ACMIS User Login Page -->
<html>
<head>
<link rel="stylesheet" href="../css/style.css"
type="text/css"></link>
</head>
<body class="othersBody">

<form name="loginForm" action="organizeAccount" method="post">
<table width="%100">
<tr>
    <td colspan="2" align="center">
        <br>
        <img src='../images/logo_ACMIS.png'>
        <br><br>
    </td>
</tr>
</table>
<table width="%100">
    <COLGROUP>
        <COL width="%30">
        <COL width="%70">
    </COLGROUP>

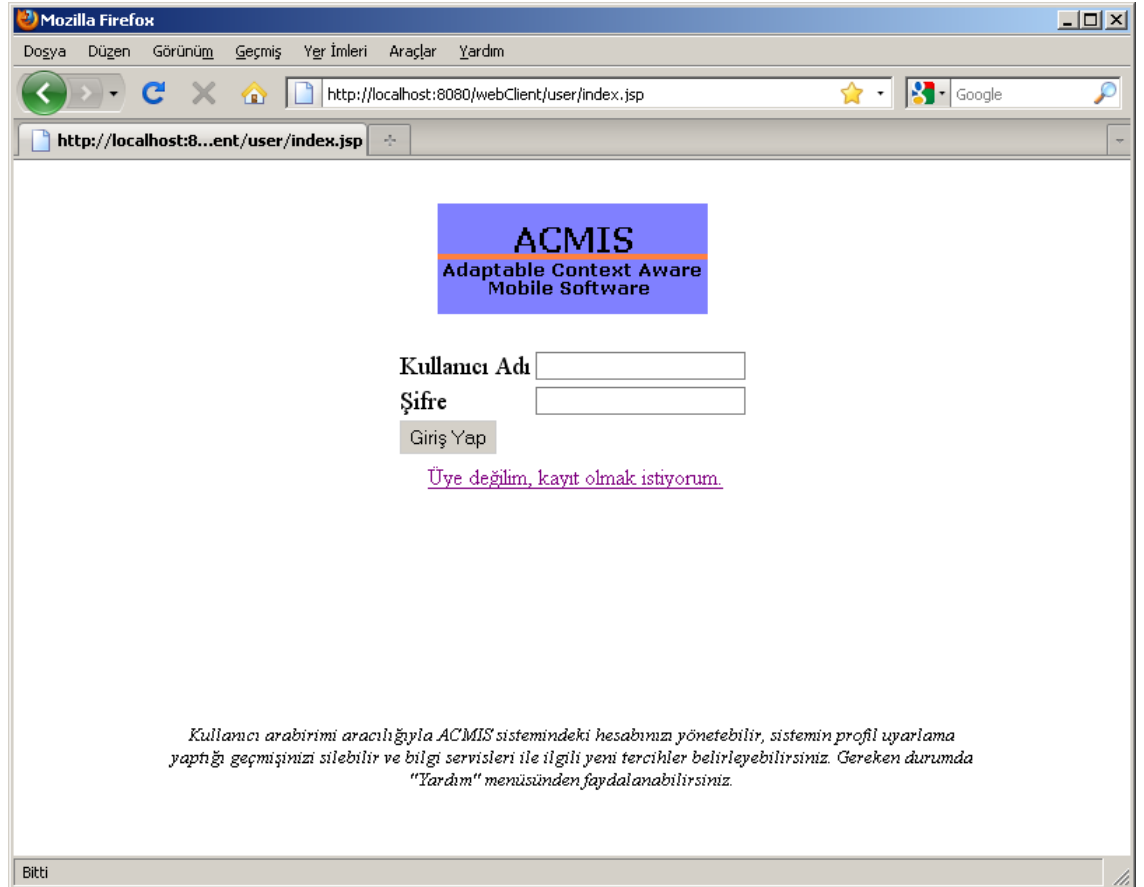
    <table class="tableStyleX" align="center">
<tr>
    <td>Kullanıcı Adı</td>
    <td><input type="text" name ="user"
class="inputbox"/></td>
</tr>
<tr>
    <td>Şifre </td>
    <td><input type="password" name ="pass"
class="inputbox"/></td>
</tr>
<tr>
    <td colspan="2"><input type="submit" name="submit"
value="Giriş Yap" class="otherSubmitButton"/></td>
</tr>
</table>

</form>
</body>
</html>
```

Web istemci ara yüzü

Web sitemci modülünü oluşturan iki ara yüzü özetlemek gerekirse; ACMIS uygulamasına kayıt olmak ve hesap yönetimi yapmak için kullanılan kullanıcı ara yüzü, uygulamanın diğer yönetim işlemleri için ise yönetici ara yüzü bulunmaktadır.

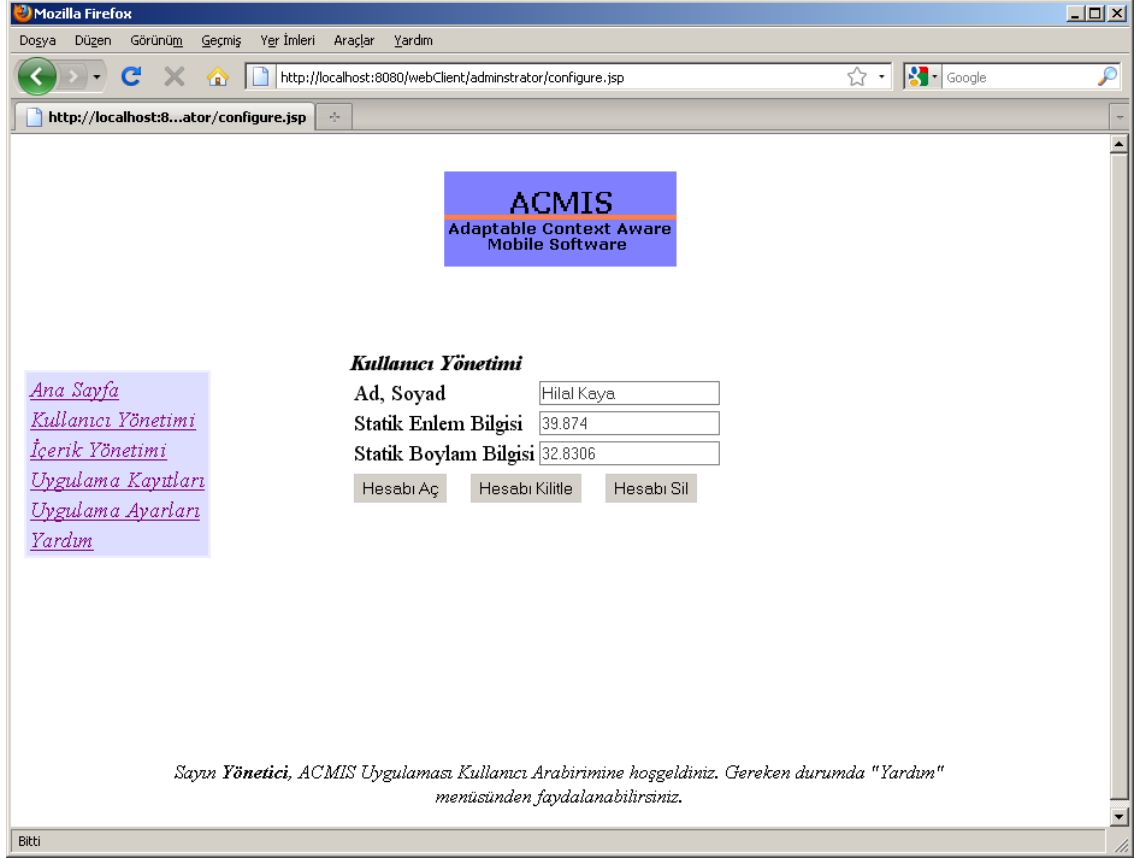
Kullanıcı ara yüzlerine ulaşmak için kullanılan giriş sayfası ortaktır. Kullanıcı önceden kayıtlı bir kişi ise direkt giriş yapabilir. Giriş yaptıktan sonraki işlevler kullanıcının sistemdeki tipine göre özelleşmektedir.



Resim 7.13. Web istemci modülü kullanıcı ara yüzleri giriş ekranı

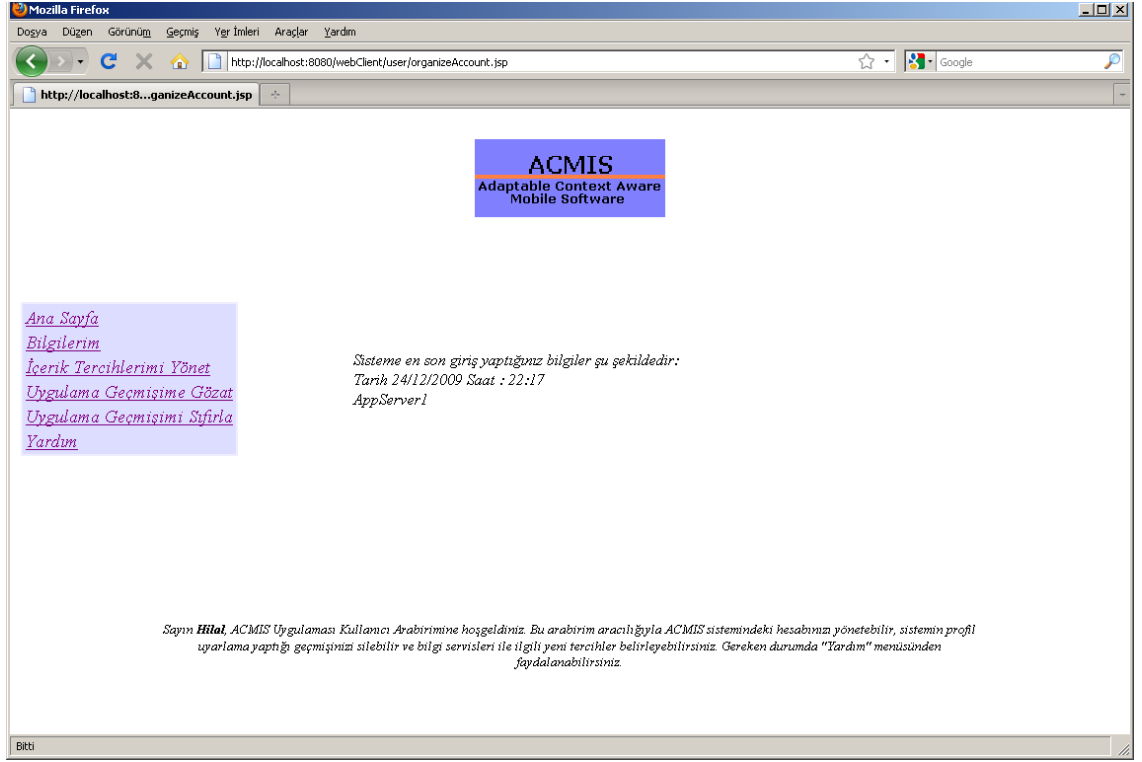
Sisteme giriş yaptıktan sonra yönetici hesabına sahip bir kullanıcının yapabileceği işlevler kullanıcılarla ilgili yönetimsel işlevler, ACMIS uygulamasında kullanılan içeriğe yönelik işlevler, uygulama kayıtlarını inceleme ve uygulama ayarlarını

düzenleme işlevleri şeklinde toplanmıştır. Her bir işlev, kendi sayfasında detay işlevleri içermektedir.



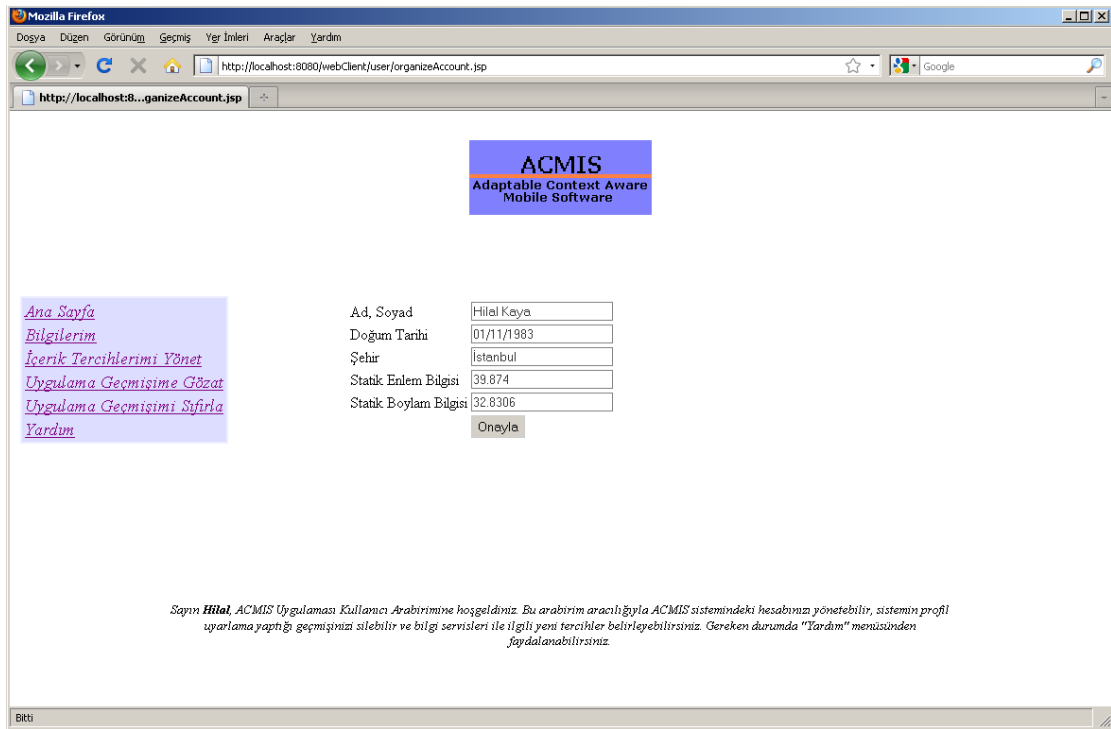
Resim 7.14. Web istemci modülünün yönetici ekranı

Bir kullanıcı sisteme giriş yaptıktan sonra hesabıyla ilgili genel yönetsel değişiklik ve incelemeler yapabileceği sayfaya yönlendirilir.



Resim 7.15. Web istemci modülünün kullanıcı ekranı

Web istemci modülünü kullanan kişiler web istemci ara yüzü sayesinde profil bilgilerini inceleyip, güncelleyebilir. Profilinde tutulan statik enlem ve boylam bilgileri GPS verisine erişilemediği zaman kullanılmak üzere bulunur ve kişinin evi, iş yeri gibi, vaktinin çoğunu geçirdiği konum bilgilerini belirtirler.



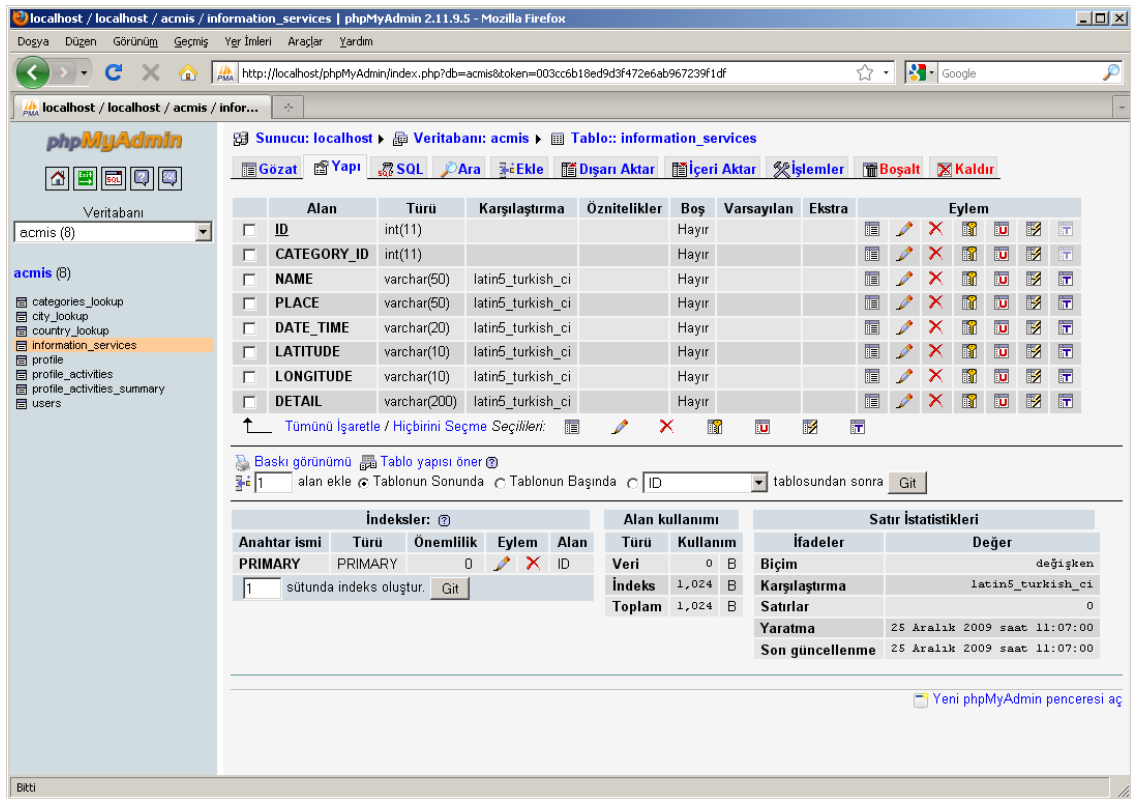
Resim 7.16. Web istemci kullanıcı ekranı, profil düzenleme sayfası

Kullanıcı içerik tercihiyle ilgili değişiklikler de yapabilir. Uygulamayı kullandığı sürece bilgilendirildiği etkinliklerden veya onayladığı, onaylamadığı etkinlikleri inceleyebilir. Ayrıca uygulamaya yeni kayıt olmuş bir kullanıcı gibi, eski tercihlerini silebilir.

7.2.4. Veritabanı modülü

Veritabanı Modülü; ACMIS platformunda bulunan verilerin saklandığı ve yönetildiği modüldür. Bu modül; kullanıcı bilgilerini, bilgi servislerini, kullanıcıların tercihlerini, kullanıcıların bilgi almak istediği servisleri, uyarlama parametrelerini tutan tablolardan oluşmaktadır.

Veritabanı MySQL üzerinde kurulmuştur. MySQL üzerinde veritabanı modülünü oluşturmak ve veritabanı tasarımının yapılabilmesi için phpMyAdmin isimli arabirim kullanılmıştır.



Resim 7.17. MySQL erişim arabirimi

Geliştirilen uygulama modülü Mysql'e erişebilmek için Java ortamında veritabanı erişim arabirimleri (Java Database Connectivity Application interface veya kısaca JDBC) spesifikasyonuna uyan bir kütüphaneden faydalanır. Kütüphanenin adı MySQL Connector/J [38] versiyonu ise 5.1.9' dur. Veritabanı modülünde bir işlem gerçekleştirmek için öncelikle bağlantı kurulur.

```
public static Connection getConnection() throws SQLException,
InstantiationException, IllegalAccessException,
ClassNotFoundException {
    Class.forName("com.mysql.jdbc.Driver");
    Connection connection =
    DriverManager.getConnection("jdbc:mysql://" + PATH_TO_DB, USERNAME,
    PASS);
    connection.setAutoCommit(false);
    return connection;
}
```

Bağlantı başarılı bir şekilde kurulduktan sonra ise uygulamada, bağlantı kurduğu kullanıcı tanımının yetkisine göre işlemler gerçekleştirir. İşlem tamamlandıktan

sonra veritabanı erişiminde kaynak problemi yaşanmaması açısından kurulan bağlantının sonlandırılması gerekir.

```
public static void releaseConnection(Connection conn) {  
    if (conn != null) {  
        try {  
            conn.commit();  
            conn.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Veritabanındaki veri tutarlılığının sağlanması açısından veritabanına bağlantı kurulduktan sonra herhangi bir hata yaşanması durumunda yapılan işlemler geri sarılır, ancak akışın normal işlemesi sonucunda bağlantı sonlandırılırken yapılan işlemlerin onaylanması adımı gerçekleştirilir.

8. SONUÇ VE ÖNERİLER

Yapılan çalışmada GPS kullanan ve bağlam farkındalığı olan, kullanıcının ilgilendiği etkinlikleri mobil cihaza herhangi bir bilgi girişine gerek kalmadan profiline uygun olarak gönderen bir sistem geliştirilmiştir. Sistem istemci sunucu mimarisine sahip olup, web istemci, mobil istemci ve sunucu modülünden oluşmaktadır. Mobil istemci mobil cihaz üzerinde çalışan, ayrıca bağlam bilgisinin GPS cihazından alınıp ayrıştırılması ve sunucu modülüne iletilmesi, ardından da sunucudan gelen yanıt kullanıcıya görüntülemekten sorumlu olan modüldür. Web istemci ise profil bilgilerinin oluşturulması ve yönetilmesi, içerikle ilgili tercihlerin yapılması ve yönetilmesinden sorumludur. Sunucu modülü ise mobil cihazdaki istemci uygulamasından gelen bağlam bilgilerini, veritabanındaki profil bilgilerini değerlendirerek bilgi servislerinden kullanıcı profiline uygun veriyi çeker ve mobil istemciye ulaştırır. Sunucu modülü aynı zamanda, mobil istemci arabiriminden yapılan servis tercihlerini kaydeder, kullanıcı profili yaptığı tercihlere göre yeniden düzenlenir. Böylece yapılan sorgularda adapte edilen profil bilgisi sonraki sorgularda kullanılmış olur.

Gelişen teknoloji ve yaygınlaşan internet sayesinde bilgiye erişim daha kolay hale gelmiştir. Fakat bilgiye erişmek gittikçe genişleyen bir ağda kullanıcıların zaman kaybı, ortama bağımlılığı anlamına da gelmektedir. Oysa mobil teknolojiler sağladıkları kolaylıklarla bu dezavantajları ortadan kaldırarak, insanların günlük hayattaki yaşam standartlarını yükseltirler. Mobil cihazlar bilgiye hızlı, her an ve kolay erişimi sunarlar. Ayrıca mobil cihazlar aracılığıyla bilgi mekandan bağımsız bir şekilde kablosuz ortamdan herkese duyurulabilir. Fakat burada yaşanan sıkıntı, çok geniş bir ağ olan internette mobil cihaz kullanıcılarının istedikleri bilgiye ulaşabilmek için yüksek miktarda bilgi taramalarının gerekmesidir. Bu aşamada mobil cihaz uygulamalarının kullanıcıya göre kişiselleşmiş bilgi sunması kullanılabilecek çözümlerden biridir. Geliştirilen uygulama sayesinde kullanıcıdan bilgi almadan konum bilgilerine ve profil bilgilerine göre sonuçlar üretilmiş olur. Normalde fazla zaman harcanarak, bir çok servis arasından yapılacak seçimler, konum bilgisine göre gönderilen seçenekler arasından yapılır. Aynı zamanda mobil

cihazdan elde edilen kullanıcı profilinden, mobil kullanıcının ilgilendiği konularla ilgili bilgi alınarak, bu doğrultuda mobil cihazın bulunduğu alandaki etkinlikler iletilir. Böylece mobil cihaza gereksiz, mobil kullanıcının ilgilenmediği etkinlikler ile ilgili bilgi iletilmemiş olur. Kullanıcının arama yapmasını gerektirmeyen otomatik çalışan bir bilgilendirme yöntemi izlenmiş olur.

Bilgi paylaşımının bu şekilde kullanıcı profiline ve ilgi alanına göre gerçekleştirilmesi sayesinde tanıtıcı reklam giderlerini ve maliyetleri indirgeyen bir yapı kurulmuş olur. Reklamın sadece ilgilenen mobil kullanıcılara duyurulması sayesinde reklamı iletmek için yapılan masraf azaltılacak, kağıt vb. gibi masraflar yapılmayacak ve reklamın faydası ile ilgili daha gerçekçi sonuçlar elde edilecektir.

Bilgi paylaşımının profile göre yapılması sayesinde kullanıcının öncelikli ihtiyacı olan çevresel bilgiler kendisine iletilmiş olur. Profilinde sağlıkla ilgili etiketler olan kişiye eczane veya hastane bilgilerinin herhangi bir aramaya gerek kalmadan sunulması, özellikle kalp krizi gibi acil durumlarda kullanıcıya zaman kazandıracak ve faydalı olacaktır.

Bundan sonra yapılacak çalışmalarda sunucu uygulamasının profile göre sorgulama aşamasında kullanılan değerlendirme parametrelerine, mobil cihaz kullanıcısının profil bilgisine yakın profil bilgileri de eklenerek daha kullanışlı önerilerde bulunan sistemler geliştirilebilir. Ayrıca kullanılan bağlam parametreleri çeşitlendirilebilir. Böylece kullanıcılara alternatif etkinlik önerilerinde bulunulmuş olur ve sistemin tek bir kullanıcı için benzer sonuçları üretme olasılığı azalmış olur.

KAYNAKLAR

1. Newcomer, E., “Understanding Web Services, XML, SOAP, WSDL and UDDI”, **Addison Wesley**, Boston, 1 – 29 (2002).
2. İnternet : World Wide Web Consortium (W3C) “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)”, <http://www.w3.org/TR/soap12-part1/> (2007).
3. Anind, K. D., “Understanding and using context”, **Personal and Ubiquitous Computing**, 5 (1): 4-7 (2001).
4. Karademirci, O., “Anlamsal web teknikleri kullanarak GPS tabanlı bağlam bilinçli mobil uygulama”, Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 49- 56 (2008).
5. İnternet : Wikipedia “World Wide Web” <http://en.wikipedia.org/wiki/www> (2009).
6. İnternet : World Wide Web Consortium (W3C) <http://www.w3.org/Consortium/> (2009).
7. İnternet : Netcraft Company, “November 2009 Web Server Survey” http://news.netcraft.com/archives/2009/11/10/november_2009_web_server_survey.html (2009).
8. İnternet : Wikipedia “Dynamic Web Page” http://en.wikipedia.org/wiki/Dynamic_web_page (2009).
9. Cerami, E., “Web services essentials”, **O’Reilly & Associates Inc., California**, 3 – 62 (2002).
10. Papazoglou, M. P., Dubray J., “A survey of web service technologies”, **University of Trento DIT-04-058, Ingegneria e Scienza dell’Informazione, Italy**, 6 - 29 (2004).
11. İnternet : Wikipedia “XML” <http://en.wikipedia.org/wiki/Xml> (2009).
12. Curbera, F.; Duftler, M.; Khalaf, R.; Nagy, W. Mukhi, N.; Weerawarana, S., “Unraveling the web services: An Introduction to SOAP, WSDL, and UDDI”, **Internet Computing, IEEE**, 6 (2): 86 – 93 (2002).
13. Papazoglou, M. P., “Service-oriented computing: Concepts, characteristics and directions”, **Proceedings of the Fourth International Conference on Web Information Systems Engineering**, 3 - 13 (2003).

14. İnternet : Wikipedia “Service Oriented Architecture”
http://en.wikipedia.org/wiki/Service-oriented_architecture (2009).
15. Ustundag, S., Kara, S., “Dağıtık programlama”, **Ege Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü**, İzmir, (2002).
16. İnternet : Wikipedia “Distributed Computing”
http://en.wikipedia.org/wiki/Distributed_system (2009).
17. İnternet : Google Code University “Introduction to Distributed System Design”
<http://code.google.com/intl/tr-TR/edu/parallel/dsd-tutorial.html> (2009).
18. Abowd, G. D., Dey A. K., Brown, P. J., Davies, N., Smith, M., Steggles, P., “Towards a better understanding of context and context-awareness”, **Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing**, Germany, 304 – 307 (1999).
19. Costa., P. D., “Towards a services platform for context-aware applications”, Yüksek Lisans Tezi, **University of Twente Telematics, Enschede**, Enschede, 10-35 (2003).
20. Schilit, B.N. And Theimer, M.M., “Disseminating active map information to mobile hosts”, **IEEE Network Magazine**, 8 (5): 22-32 (1994).
21. Brown, P.J., Bovey, J.D. Chen, X. “Context-aware applications: From the laboratory to the marketplace”, **IEEE Personal Communications Magazine**, 4 (5): 58-64 (1997).
22. Ryan, N., Pascoe, J., Morse, D., “Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant”, **British Archaeological Reports, Oxford**, (1997)
23. Chen, G. And Kotz, D., “A survey of context-aware mobile computing research”, **Dartmouth College Computer Science TR2000-381, Hannover**, (2000).
24. Pascoe, J., “Adding generic contextual capabilities to wearable computers”, **Proceedings of the 2nd IEEE International Symposium on Wearable Computers**, Pittsburgh, 19 (20): 92 – 99 (1998).
25. Schilit, B., Adams, N. And Want, R., “Context-aware computing applications”, **1st International Workshop on Mobile Computing Systems and Applications**, Santa Cruz, CA, 85–90 (1994).
26. Van Sinderen, M.J., Van Halteren, A.T. Wegdam, M., Meeuwissen, H.B., Eertink, E.H., “Supporting context-aware mobile applications: An infrastructure approach”, **IEEE Communications Magazine**, 44 (9): 96 – 104 (2006).

27. Salton G. and McGill M.J., An Introduction to Modern Information Retrieval, **McGraw-Hill**, New York, 152-160 (1983).
28. Chen, L., Syacara, K., “Webmate: A personal agent for browsing and searching”, ***Proceedings of the Second International Conference on Autonomous Agents***, Minnesota, 132 – 139 (1998).
29. Widyanoro, D. H., Ioerger, T. R., Yen, J., “An adaptive algorithm for learning changes in user interests”, ***Proceedings of the Eighth International Conference on Information and Knowledge Management***, Missouri, 405 – 412 (1999).
30. Balabanovic, M., “An adaptive web page recommendation service”, ***Proceedings of the Second International Conference on Autonomous Agents***, Minnesota, 378 – 385 (1998).
31. Setten, M. V., “Context aware recommendations in the mobile tourist application COMPASS”, ***Third International Conference on Adaptive HyperMedia and Adaptive Web Based Systems***, Eindhoven, 235 – 244 (2004).
32. Poslad, S., Laamanen, H. Malaka, R. Nick, A. Buckle, P. Zipl, A. , “CRUMPET: Creation of user-friendly mobile services personelised for tourism”, ***Second International Conference on 3G Mobile Communication Technologies***, London, 28 – 32 (2001)
33. İnternet : The Apache Software Foundation “Apache Tomcat” <http://tomcat.apache.org/index.html> (2009).
34. İnternet : Sun Microsystems “MySQL” <http://www.mysql.com/about/> (2009).
35. İnternet : Sun Microsystems Developer Network “Java Servlet Technology” <http://java.sun.com/products/servlet/> (2009).
36. İnternet : Eclipse “About Eclipse Foundation” <http://www.eclipse.org/org/> (2009).
37. İnternet : SOAP Engine for J2ME Project Site “kSOAP2” <http://ksoap2.sourceforge.net/> (2009).
38. İnternet : Sun Microsystems “MySQL Connector/J” <http://dev.mysql.com/doc/refman/5.0/en/connector-j.html> (2009).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KAYA, Hanife Hilal
Uyruğu : T.C.
Doğum tarihi ve yeri : 01.11.1983 Balıkesir
Medeni hali : Evli
e-mail : hilalkaya@gmail.com

Eğitim

Derece

Lisans

Eğitim Birimi

Dokuz Eylül Üniversitesi/

Bilgisayar Mühendisliği

Mezuniyet tarihi

2005

Lise

Susurluk Anadolu Lisesi

2001

İş Deneyimi

Yıl

2005-2008

2008-

Yer

Dış Ticaret Müsteşarlığı

Türk Hava Yolları

Görev

Yazılım Mühendisi

Analist Programcı

Yabancı Dil

İngilizce