

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

KİŞİSELLEŞTİRİLEBİLİR MOBİL ASİSTAN

Ülkü BEYDAĞ İNCE

Tez Danışmanı : Prof. Dr. Ata ÖNAL

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 619.01.00

Sunuş Tarihi : 10.05.2010

Bornova- İZMİR

2010

Ülkü BEYDAĞ İNCE tarafından **Yüksek Lisans Tezi** olarak sunulan **“Kişiselleştirilebilir Mobil Asistan”** başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **10.05.2010** tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı

: Prof. Dr. Ata ÖNAL

.....

Raportör Üye

: Yrd. Doç. Dr. Murat Osman ÜNALIR

Üye

: Yrd. Doç. Dr. Tuğkan TUĞLULAR

.....

ÖZET**KİŞİSELLEŞTİRİLEBİLİR MOBİL ASİSTAN**

BEYDAĞ İNCE, Ülkü

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ata ÖNAL

Mayıs 2010, 89 sayfa

Gerçekleştirilen tez çalışmasının amacı J2ME (Java 2 Micro Edition), web servisleri ve ASP.NET 2.0 ile gelen kişiselleştirme teknikleri kullanılarak Ege Üniversitesi kurumsal web sitesi üzerinde yer alan “Ege Duyuru” ve “Ege Telefon Rehberi” hizmetlerini mobil platforma taşıyan uygulamanın tasarım ve gerçekleştiriminin yapılmasıdır. Tez kapsamında Ege Üniversitesi kurumsal web sitesi üzerinde yer alan bu hizmetlerin, farklı teknolojilerdeki farklı platformlar için entegrasyonunu kolaylaştıran katmanlı bir mimari tasarlanmıştır. Ayrıca kullanılan çeşitli kişiselleştirme teknikleriyle “Ege Duyuru” hizmetinin mobil cihazlar üzerinde kullanıcının tercihleri doğrultusunda kişiye özel farklılaşarak, daha kolay kullanımı hedeflenmiştir.

Anahtar sözcükler: Mobil, Kişiselleştirme, J2ME, Web Servisi

ABSTRACT

PERSONALIZED MOBILE ASISTANT

BEYDAG INCE, Ulku

MSc in Computer Engineering

Supervisor: Prof. Dr. Ata ÖNAL

May 2010, 89 pages

The aim of the thesis developed is to design and implement a mobile application which supports “EgeTel” and “EgeDuyuru” services which are also accessible from Ege Univercity web site; by using technologies J2ME, web services and personalization support that comes with ASP.NET 2.0. In the concept of this thesis, a multi layer structure is designed for the services resides on Ege Univercity web site to be able to integrate these services to different platforms with different technologies. Also by using some personalization techniques, it is supported for “Ege Duyuru” service to make its usage easier on mobile platforms by differantiating it, according to user’s profiles.

Keywords: Mobile, Personalization, J2ME, Web Service

TEŞEKKÜR

Bu çalışma süresince danışmanlığımı yapan ve desteğini esirgemeyen sayın Prof. Dr. Ata ÖNAL'a, tezim esnasında fikir ve tecrübeleriyle değerli katkılarını aldığım Yrd. Doç. Dr. Murat Osman ÜNALIR ve Yrd. Doç. Dr. Tuğkan TUĞLULAR'a, destekleri ve varlıklarıyla her zaman bana güç veren eşim Cenk İNCE'ye ve aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	V
ABSTRACT	VII
TEŞEKKÜR	IX
ŞEKİLLER DİZİNİ	XIII
ÇİZELGELER DİZİNİ	XV
SİMGELER VE KISALTMALAR DİZİNİ	XVIII

1. GİRİŞ	1
-----------------------	----------

2. J2ME	4
----------------------	----------

2.1 Java	4
2.2 J2ME	5
2.2.1 Konfigürasyonlar	8
2.2.1.1 CDC	9
2.2.1.2 CLDC	10
2.2.2 Profiller	11
2.2.2.1 MIDP (Mobile Information Device Profile)	11
2.2.2.2 PDAP (PDA Profile)	13
2.2.2.3 FP(Kuruluş profili - Foundation Profile)	13
2.2.2.4 PP (Kişisel profil - Personal Profile)	13
2.2.2.5 PBP (Kişisel temel profil - Personal Basis Profile)	13
2.2.3 Midlet geliştirme	14

3. WEB SERVİSLERİ	18
--------------------------------	-----------

3.1 Web Servislerinin Kullanım Nedenleri	20
3.2 Web Servislerinin Yararları	21
3.3 XML Web Servislerinin Alt Yapısı	21
3.3.1 XML web servisleri dizinleri	23
3.3.2 XML web servislerin keşfi	23
3.3.3 XML web servis tanımı	23
3.3.4 XML web servisi bağlantı formatları (Wire formats)	23
3.3.5 SOAP	24

İÇİNDEKİLER (Devamı)

3.4 İstemci ve XML Web Servisi Arasında İletişim	25
3.5 .NET XML Web Servis İsim Uzayları (Namespaces)	26
3.6 ASP .NET Web Servisinin Geliştirilmesi	27
4. KİŞİSELLEŞTİRME	32
4.1 Kişiselleştirme Yöntemleri	33
4.1.1 Kullanıcı düzenlemelerine dayalı kişiselleştirme	33
4.1.2 Uyarlanabilir kişiselleştirme	34
4.2 Kişiselleştirmenin Gerçekleştirimi	36
5. UYGULAMANIN GERÇEKLEŞTİRİMİ	41
5.1 Mobil Uygulama Sistemi	42
5.1.1 Sisteme giriş	44
5.1.2 Rehberden kayıt arama	46
5.1.3 Duyuruların listelenmesi	49
5.1.4 Profil Oluşturma	52
5.1.5 Profil Sorgulama	52
5.1.6 Profil Güncelleme	53
5.1.7 Duyuruların etiketlenmesi	54
5.2 Web Uygulama Sistemi	56
5.3 Ege Web Servis Sistemi	59
5.4 Ege Web Data Sistemi	63
5.4.1 EgeTel web servisi	64
5.4.2 EgeDuyuru web servisi	65
5.5 Uygulamada Kullanılan Kişiselleştirme Teknikleri	67
6. UYGULAMANIN MOBİL CİHAZA YÜKLENMESİ	70
7. UYGULAMANIN KULLANIMI	72
8. SONUÇ	86
KAYNAKLAR DİZİNİ	87
ÖZGEÇMİŞ	89

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1	Temel Java sürümlerinin hedefledikleri ortamlar ve cihazlar.....5
Şekil 2.2	Konfigürasyon ve profilin cihazdaki yerleşimi.....8
Şekil 2.3	Konfigürasyon, profiler ve hedef cihaz grupları.....8
Şekil 2.4	J2SE, CDC ve CLDC konfigürasyonlarının karşılaştırması.....9
Şekil 2.5	MIDP 'ın cihaz içerisindeki durumu.....12
Şekil 3.1	XML Web Servisinin Altyapısı 22
Şekil 3.2	Web Servis Bileşenleri.....22
Şekil 3.3	İstemci ve XML Web Servisi arasındaki iletişim 25
Şekil 3.4	Tarayıcı üzerinde web servisi içerisindeki metodların gösterimi 29
Şekil 3.5	Tarayıcı üzerinde bir web metodun gösterimi 30
Şekil 3.6	Tarayıcı üzerinde bir web metodun geri dönüş değerinin gösterimi 30
Şekil 4.1	Asp.Net 2.0'ın kişiselleştirme için oluşturduğu veritabanı yapısı ve tabloları 40
Şekil 5.1	Sistemin genel mimarisi.....41
Şekil 5.2	Mobil uygulama sisteminin kullanım durum (use-case) diyagramı 43
Şekil 5.3	Uygulamanın sıra (sequence) diyagramındaki gösterimi 44
Şekil 5.4	Sisteme giriş senaryosu sıra diyagramı.....45
Şekil 5.5	Rehberden kayıt arama senaryosu sıra diyagramı 47
Şekil 5.6	Duyuruları listeleme senaryosu sıra diyagramı.....49
Şekil 5.7	Duyuruları etiketleme, etiketleri listeleme/filtreleme senaryosu sıra diyagramı.....54
Şekil 5.8	Profil ayarları menüsündeki kişiselleştirme seçenekleri.....68
Şekil 5.9	Etiket girişi, etiket listesi ve duyuru detaylarında etiketlerin gösterimi 69
Şekil 7.1	Kullanıcı bilgileri giriş ekranı.....72
Şekil 7.2	Yanlış kullanıcı adı/şifre girişi uyarı ekranı.....73
Şekil 7.3	Ana menu ekranı 74
Şekil 7.4	Rehberden kayıt arama ekranı 75
Şekil 7.5	Rehberden aranan kişiye ait kayıt ekranı 76
Şekil 7.6	Duyuruların Listelenmesi.....77
Şekil 7.7	Duyuru detay ekranları 78
Şekil 7.8	Duyuruya etiket ekleme ekranı 79
Şekil 7.9	Duyuru ayarları menu ekranı 80

ŞEKİLLER DİZİNİ (Devamı)

<u>Şekil</u>	<u>Sayfa</u>
Şekil 7.10 Profil ayarları – Etkinlik seçimi	81
Şekil 7.11 Profil ayarları – Birim/Tarih aralığı seçimi ve Anahtar kelime girişi.....	82
Şekil 7.12 Etiket Listesi.....	83
Şekil 7.13 Etikete göre filtrelenen duyurular listesi	84
Şekil 7.14 Filtrelenen duyurunun detay ekranı	85

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
Çizelge 2. 1 J2ME konfigürasyonları listesi	6
Çizelge 2. 2 J2ME Profil Listesi	7
Çizelge 2. 3 J2ME seçmeli API'leri listelenmektedir	7
Çizelge 2. 4 MIDlet'in geliştirimi	14
Çizelge 2. 5 MerhabaDunya.java	16
Çizelge 3. 1 Login isimli web metoda ait kod	28
Çizelge 3. 2 Web metoda erişim.....	31
Çizelge 4. 1 ASP.NET 2.0 ile gelen yeni konfigürasyonlar	37
Çizelge 4. 2 Membership API kullanımı	38
Çizelge 4. 3 Membership API ile doğrulama işlemi	38
Çizelge 4. 4 Web.config dosyası içerisine dahil edilmiş örnek Profile özellikleri.....	39
Çizelge 4. 5 Profile sınıfının kod içinde kullanımı.....	39
Çizelge 5. 1 Kullanıcı adı ve şifre sorgusu için web servisinin çağırımı	46
Çizelge 5. 2 Rehberden kayıt aramak için web servisinin çağırımı	48
Çizelge 5. 3 Fakülte bilgilerinin getirilmesi için web servisinin çağırımı.....	48
Çizelge 5. 4 Duyuruların listelenmesi için web servisinin çağırımı	51
Çizelge 5. 5 Profil bilgilerinin web servisinde getirilmesi	53
Çizelge 5. 6 Profilleri güncelleyen web servisinin çağırımı.....	54
Çizelge 5. 7 Etiket ekleme işlemi için Ege Web Servis sistemindeki web metodun çağırımı.....	55
Çizelge 5. 8 Etiket ekleme işlemi için Ege Web Data Sistemindeki web metodun çağırımı.....	56
Çizelge 5. 9 Web uygulamasındaki rehberden kayıt arama sorgusu için web servisinin çağırımı.....	58
Çizelge 5.10 EgeWebpart sistemindeki web servisi üzerinden web metodlarının çağırımı	58
Çizelge 5.11 Kullanıcı adı-şifre doğrulama.....	60
Çizelge 5.12 Rehberden isim bilgisi elde etmek için web servisindeki web metod çağırımı	60
Çizelge 5.13 Rehberden fakülte, bölüm, e-posta gibi bilgileri çekme	61
Çizelge 5.14 Duyuruları getirmek için web servisindeki web metodun kullanımı.....	61

ÇİZELGELER DİZİNİ**Çizelge****Sayfa**

Çizelge 5.15 Duyurulara etiket eklemek için web servisindeki web metodun kullanımı	62
Çizelge 5.16 Kullanıcı profilindeki etkinlik tercihlerini getiren web metod	62
Çizelge 5.17 Kullanıcı profilindeki etkinlik tercihlerine yeni değer atama	63
Çizelge 5.18 Php web servisi üzerinden GetName metodunun çağırımı	65
Çizelge 5.19 Php web servisi üzerindeki GetAllAnnouncements web metodu	66
Çizelge 5.20 Php web servisi üzerindeki SetKeywords web metodu	67

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standart Edition
J2EE	Java 2 Enterprise Edition
JVM	Java Virtual Machine -Java Sanal Makinesinin
CDC	Connected Device Configuration
CLDC	Connected, Limited Device Configuration
CVM	Compact Virtual Machine
KVM	Kilobyte Virtual Machine
MIDP	Mobile Information Device Profile
PDAP	PDA Profile
FP	Foundation Profile
PP	Personal Profile
PBP	Personal Basis Profile
JDK	Java Development Kit
XML	Extensible Markup Language
DTD	Document Type Definition
EJB	Enterprise JavaBeans
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery, and Integration
RPC	Remote Procedure Call - Uzaktan Prosedür Çağırımı

1. GİRİŞ

Günümüzde mobil teknolojilerin kullanım alanı ve bu alanda gerçekleştirilen uygulama sayısı hızla artmaktadır. İnternetin ve mobil teknolojilerin son zamanlarda hızla hayatımıza girişi, iş yapış biçimlerimizde de birçok değişimlere yol açmaktadır. Özellikle 3G teknolojisinin hayatımıza girmesiyle birlikte daha yüksek hızda, her yerden internet erişimi sağlanmış ve bu da mobil uygulamaların kullanımını arttırmıştır.

Mobil uygulamalara olan talebin ve mobil platform çeşitliliğinin artması uygulama geliştiricileri, platform bağımsız hizmet verecek sistemler için arayışa geçirmiştir. Bu arayışlar sonucu ortaya çıkan web servisleri herhangi bir programlama dilinden ve platformdan bağımsız, birlikte çalışabilen dağıtık uygulamalara çözüm sunmaktadır.

Günümüzde artık iş yapılanmaları herkesin kendi işini yapması ve işinin gerektirdiği tamamlayıcı servisler için de üçüncü parti destek alması şeklinde olmaktadır. Yani üretim yapan bir firma dağıtım için lojistik firmasıyla, pazarlama için de reklâm ajanslarıyla çalışmaktadır. Böylece asıl yaptığı işe yoğunlaşmakta ve kaliteli hizmet sunabilmektedir. Bu anlayış internet ortamında hizmet sunan firmalar için de geçerlidir. Artan web portalleri ve internet bankacılığı hizmetlerinin temelinde de bu hizmetleri ortak noktada bütünleştirmek yatmaktadır. E-iş ve e-devlet çalışmaları ise gelecekte bu anlayışın yaygınlığını sağlayacaktır. İşte web servisleri bu noktada devreye girmektedir (Şen, K, 2010).

Web servislerinin popüler olmalarının birçok sebebi vardır (Bhalla and Kazerooni, 2007):

Birlikte çalışabilirlik: Servisler herhangi bir düzenleyici çerçeve (framework) içersinde, herhangi bir programlama modelinden bağımsız, herhangi bir dille hazırlanmış olabilir.

Bakım: Servis tabanlı mimari fonksiyonel parçaları ayırır, böylece bakım çalışmaları merkezileşir.

Yeniden kullanım: Servis tabanlı mimari farklı uygulamalar arasında kodun yeniden kullanımını “servis” kavramı ile teşvik eder.

Akıllı kanal: Sadece tek bir çağrıma değer katan aracı servisler zinciri sağlar.

Web servislerinin sunduğu platform/dil bağımsız çalışabilme olanağı uygulamaların katmanlı mimarilerde tasarımları ve gerçekleştirmeleriyle bu bağımsızlığı daha da arttırmaktadır. Bunun yanı sıra katmanlı mimari ile tasarlanmış uygulamaların farklı platformlara entegrasyonu da çok daha kolay olmaktadır.

Mobil teknoloji dünyasındaki tüm bu gelişmelere ek olarak, mobil teknolojilerin bu denli yaygın kullanımlarında kişiselleştirme kavramının etkisi göz ardı edilemez. Özellikle mobil uygulamalar söz konusu olduğunda mobil cihazlardaki ekran boyutunun bilgisayara göre daha küçük olması, mobil cihazların kullanımının bilgisayara göre daha zor olmasından ve kısıtlı kullanıcı arayüzlerinden dolayı kişiselleştirme, mobil uygulamalarda daha fazla önem taşımaktadır.

Günümüz web uygulamalarında, kullanıcıya dayalı kişiselleştirme oldukça önemli ve popüler bir konudur. Örneğin, bir elektronik ticaret sitesini ziyaret ettiğimizi düşünelim. Bu site bir sonraki ziyaretimizde, daha önceden bakmış olduğumuz ürünler ile ilişkili yeni önerilerde bulunabilir. Böylece siteye giren kullanıcılar başka ürünlere yönlendirilebilir. Yeni önerilerin sunulabilmesi, sitenin bizim alışkanlığımızı takip etmesi ve kendisini buna göre değiştirmesi anlamına gelmektedir. Daha açık bir şekilde söylemek gerekirse bu site, üyelerin özelliklerine göre kişiselleşmektedir (Şenyurt, 2009).

İşte yukarıda değinilen tüm bu teknolojik gelişmeler ve ortaya çıkan gereksinimler göz önüne alındığında tez kapsamında gerçekleştirilen uygulama ve kullanılan teknolojiler bu doğrultuda yapılacak daha sonraki çalışmalar için yol gösterici niteliktedir.

Tez kapsamında, Ege Üniversitesi kurumsal web sitesi üzerinde yer alan “Ege Duyuru” ve “Ege Telefon Rehberi” hizmetlerini mobil platform üzerinde çalıştıran bir uygulama gerçekleştirilmiştir. Tez çalışması, Şen’in 2008 yılında tamamladığı “Web Part ve Ajax Teknolojileri ile Kişiselleştirilebilir Portal Uygulaması” adlı tezi temel alınarak gerçekleştirilmiştir. Şen’in tezinde ASP .NET 2.0 ve AJAX (Asynchronous JavaScript and XML) teknolojileri kullanarak, verilerin kişiselleştirmeye izin verecek bağımsız birimler halinde sunulduğu bir portal tasarımının ve gerçekleştirimi yapılmıştır (Şen, 2008). Bu tez kapsamında ise Şen’in

kurduđu alt yapı kullanılarak Ege Üniversitesi web sitesi üzerinde yer alan “egetel”, “egeduyuru” isimli hizmetler kişiselleştirilerek mobil platforma taşınmıştır.

Sisteme giriş üyelik doğrulama işlemi ASP.NET 2.0’ın desteklediđi Membership API ile gerçekleştirilmiştir. Sisteme başarılı bir şekilde giriş yapılmasının ardından Ege Üniversitesi telefon rehberinden kayıt aranabilir ya da Ege Üniversitesi duyuruları listelenebilir.

Duyuruların listelenmesinde daha çoğunlukla kullanıcıya dayalı kişiselleştirme teknikleri kullanılmıştır. Ege Üniversitesi veritabanından gelen duyurular kullanıcının profilinde saklanan seçimlere göre filtrelenerek ve önceliklendirmelere göre sıralanarak listelenir. Kullanıcının duyuru listesinde görmek istediđi etkinlik türlerini, hangi birimlerin gönderdiđi duyuruların listelenmesini istiyorsa o birimleri ve listelenecek duyuruların zaman aralıđını uygulamadaki arayüzler aracılığıyla seçmesiyle duyuruların kişiye özel filtrelenmesi için gerekli profil bilgisi elde edilir. Duyuruların sıralanması yani kişiye özel önceliklendirilmesi içinse yine arayüzler aracılığıyla girilen anahtar kelimeler ve içsel uzmanlık alanı saptama mekanizması kullanılır. Bu içsel olarak çalışan mekanizma uyarlanabilir kişiselleştirme tekniđini kullanılarak geliştirilmiştir; kullanıcının üyelik işlemi sırasında girdiđi isim bilgisinden fakülte-bölüm bilgisinin elde edilmesine dayanır. Duyuruya ait fakülte-bölüm bilgisi, kullanıcıya ait fakülte-bölüm bilgisiyle aynı olan duyurular listenin en başında olacak şekilde önceliklendirme yapılarak listelenir. Tüm bu profil işlemleri ASP.NET 2.0 ile gelen Profile API ile gerçekleştirilmiştir.

Profil bilgilerine göre kişiselleştirmenin yanı sıra tez kapsamında duyurular için etiketleme mekanizması da gerçekleştirilmiştir. Bu kapsamda kullanıcı yetkisi varsa herhangi bir duyuruya etiket ekleyebilir. Herhangi bir duyuru için eklenen tüm etiketleri listeleyebilir. Sistemdeki duyurulara ait tüm etiketlerin yer aldığı liste üzerinden seçtiđi bir etikete göre o etiketin yer aldığı duyuruları filtreleyebilir. Kullanıcının duyurulara etiket ekleyebilmesiyle kullanıcı da sisteme dahil edilmiş ve bu da uygulamadaki kişiselleştirmeye farklı bir boyut kazandırmıştır.

Bu dokümanda izleyen bölümlerde yukarıda açıklanan uygulamayı tasarlamak ve gerçekleştirmek için kullanılan J2ME, web servisleri ve ASP.NET 2.0 ile gelen kişiselleştirme teknolojileri, sistem mimarisi, geliştirilen uygulamanın gerçekleştirimi, cihaza yüklenmesi ve kullanımı detaylı bir şekilde anlatılmaktadır.

2. J2ME

2.1 Java

Giderek hayatımızda daha çok yer almaya başlayan cep telefonu, kişisel cep bilgisayarları, POS cihazları, televizyon yayın kutuları gibi çeşitli kategorilerden cihazların temelde benzer noktaları olmalarına rağmen yaptıkları işler nedeniyle kendilerine ait özellikleri vardır. Bu cihazlar hafıza, işlem gücü, ekran görüntü özellikleri gibi teknik özellikleri açısından bildiğimiz bilgisayarlara göre daha az güçlüdürler ve daha özel konuları hedeflemektedirler. Çok çeşitli sayıda cihaz için ortak uygulama çalıştırma ortamı sağlamak ancak tüm yapıların ortak olarak kabul ettikleri standartlar üzerinde kurulabilir. Geniş kullanım alanları ve genişletilebilirlik özellikleri ön plana çıkan Java bu amacı oluşturmak için en uygun ortam olarak gözükmektedir (Mobilpro, 2003).

Nesne tabanlı programlama özelliği, yazılan programın değişikliğe uğraması ya da yeniden derlenmesi gerekmeden farklı yapıdaki bilgisayar sistemlerinde kullanılması gibi özellikler Java dilinin popülerliğini sağlayan önemli özelliklerin sadece birkaçıdır. İlk geliştirilme amacı SUN Microsystem şirketinin içinde ortak bir dil kullanılması olan ve o zamanki adı OAK olan Java dili, kısa zamanda dünya çapında yoğun olarak kullanıma girmiştir. Javanın kullanıldığı yerler, basit ev bilgisayarlarından, en karmaşık sistem sunucularına kadar değişen bir yelpazede yer almaktadır.

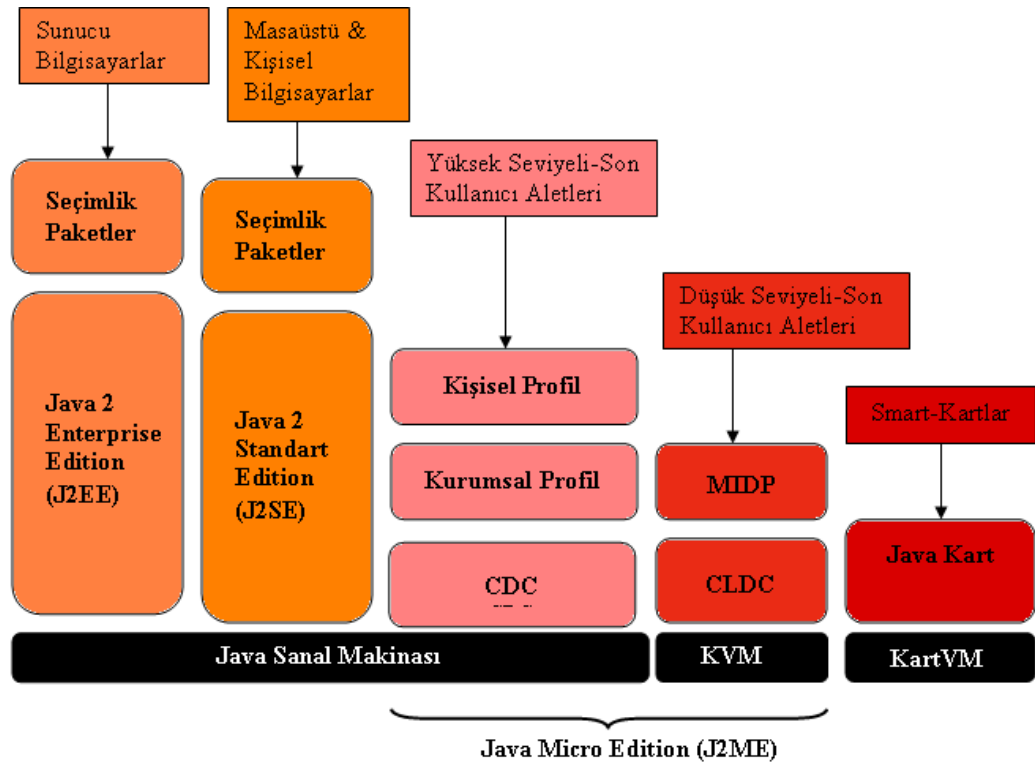
Java dilinin zamanla pek çok özelliğiyle her platformda kullanılan bir programlama diline dönüşmesiyle dil genişlemiş ve zamanla bu büyüklüğü sorun olmaya başlamıştır. Çözüm olarak Java dilini büyük mainframe sistemlerden cep telefonlarına kadar tüm cihazlarda kullanılabilir kılmak için Java dili değişik sürümlere ayrılmıştır.

- J2SE (Java 2 Standart Edition) temel Java ortamıdır. İstemci-sunucu ve web tarayıcı için yazılan uygulamalarda kullanılır.

- J2EE (Java 2 Enterprise Edition) çok katmanlı ve dağıtık yapıdaki genellikle sunucu taraflı uygulamaları geliştirmek için kullanılır.

- J2ME ise elektronik ve gömülü cihazlarda Java uygulamalarını çalıştırmak için geliştirilmiştir.

Şekil 2.1’de, mevcut temel Java sürümlerinin hedefledikleri ortamlar ve cihazlar birlikte gösterilmektedir (Mobilpro, 2003). Burada soldaki yüksek seviyeli kullanıcı platformlarından başlayarak sağdaki düşük seviyeli kullanıcı platformlarına doğru ilerleyen Java 2 Platform’unun sürümleri ve hedef pazarları görülmektedir. Temel olarak, beş hedef pazar tanımlanmıştır. Sunucu bilgisayarlar J2EE, kişisel bilgisayarlar J2SE ile desteklenmektedir. J2ME ise yüksek seviyeli ve düşük seviyeli kullanıcı cihazlarına odaklanan iki kategoriye ayrılmıştır. Son olarak, Java Kart standartları ise smart kart pazarına odaklanmıştır.



Şekil 2. 1 Temel Java sürümlerinin hedefledikleri ortamlar ve cihazlar (Mobilpro, 2003)

2.2 J2ME

Java 2 Platform, J2ME Java’nın kısa tarihinde ikinci bir devrimdir. Java 1995 yılında ilk kez tanıtıldığında, indirilebilen, isteğe bağlı çalıştırılabilen applet’ler ve limitli küçük programlardan ibaretmiş gibi görünüyordu. Ancak sunucular üzerinde çalışan sunucu tabanlı java uygulamaları ile Java, platform olarak değişim geçirdi. Java 2 Platform, J2EE ile sunucu tabanlı yazılım tarafında büyüyerek ilk devrimini gerçekleştirmiştir (Knudsen and Li, 2005).

Java’nın ikinci devrimi halen sürmekte olan küçük cihazlar için yaptığı patlamadır. Küçük cihazlar için market her geçen gün hızla büyümekte ve iki

sebepten dolayı Java önem arz etmektedir. İlk olarak, geliştiriciler Java ile düzinelerce küçük cihaz için kod yazarak hiç bir değişiklik yapmadan bu kodu cihazlarında çalıştırabilmekteler. İkinci olarak da, Java indirilebilen kodlar için önemli derecede güvenlik özelliklerine sahiptir (Knudsen and Li, 2005).

J2ME sadece özel bir yazılım parçası ya da şartnameden ibaret değildir. Tam anlamıyla küçük cihazlar için Java denebilir. Küçük cihazlar boyut açısından pagers, cep telefonları, kişisel dijital asistanlar (PDA) ve hatta uydu alıcıları olarak çeşitlilik gösterirler. J2ME, kütüphaneler ve farklı cihaz aileleri ile ilgili özel bilgi sağlayan konfigürasyonlar, profiller ve seçmeli kütüphaneler olarak gruplanabilir (Knudsen and Li, 2005).

Konfigürasyon, cihazları hafıza kısıtları ve işlemci gücü açısından gruplayabilmek için tasarlanmıştır. Herhangi bir konfigürasyonu destekleyen cihaza aktarılabilecek olan JVM'nin (Java Virtual Machine - Java Sanal Makinesi) kolayca belirlenmesini sağlar. Ayrıca platform üzerinde kullanılan ve ek olarak gerekli olabilecek olan Java 2 Platform, J2SE kütüphanelerinin alt kümesini tam olarak belirler. Cihaz üreticileri belirli bir konfigürasyonun cihazlarına aktarılmasından kendileri sorumludur. Çizelge 2.1'de J2ME konfigürasyonları listelenmektedir (Knudsen and Li, 2005).

Çizelge 2. 1 J2ME konfigürasyonları listesi (Knudsen and Li, 2005).

Configurations		
JSR	Name	URL
30	Connected, Limited Device Configuration (CLDC) 1.0	http://jcp.org/jsr/detail/30.jsp
139	Connected, Limited Device Configuration (CLDC) 1.1	http://jcp.org/jsr/detail/139.jsp
36	Connected Device Configuration 1.0.1	http://jcp.org/jsr/detail/36.jsp
218	Connected Device Configuration 1.1	http://jcp.org/jsr/detail/218.jsp

Profiller, konfigürasyonlardan daha özeldirler. Profil, konfigürasyona bağlıdır ve kullanıcı arayüzü, kalıcı veri depolama ve cihaz üzerinde bir uygulamayı çalıştırmak için geliştirilmesi gerekli olan ne varsa hepsini içeren ekstra kütüphaneleri sağlar. Şekil 2.2'de J2ME profilleri listelenmektedir (Knudsen and Li, 2005).

Çizelge 2. 2 J2ME Profil Listesi (Knudsen and Li, 2005).

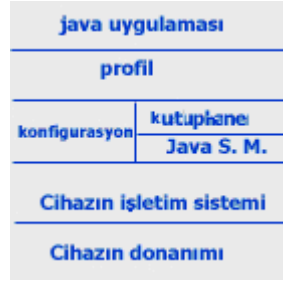
Profiles		
JSR	Name	URL
37	Mobile Information Device Profile 1.0	http://jcp.org/jsr/detail/37.jsp
118	Mobile Information Device Profile 2.0	http://jcp.org/jsr/detail/118.jsp
75	PDA Profile 1.0	http://jcp.org/jsr/detail/75.jsp
46	Foundation Profile 1.0	http://jcp.org/jsr/detail/46.jsp
129	Personal Basis Profile 1.0	http://jcp.org/jsr/detail/129.jsp
62	Personal Profile 1.0	http://jcp.org/jsr/detail/62.jsp
219	Foundation Profile 1.1	http://jcp.org/jsr/detail/219.jsp
217	Personal Basis Profile 1.1	http://jcp.org/jsr/detail/217.jsp

Seçmeli kütüphaneler ise belirli bir konfigürasyon ya da profil tarafından içerilebilen özel ek özellikleri belirler. Şekil 2.3’de J2ME seçmeli kütüphaneleri listelenmektedir (Knudsen and Li, 2005).

Çizelge 2. 3 J2ME seçmeli API’leri listelenmektedir (Knudsen and Li, 2005)

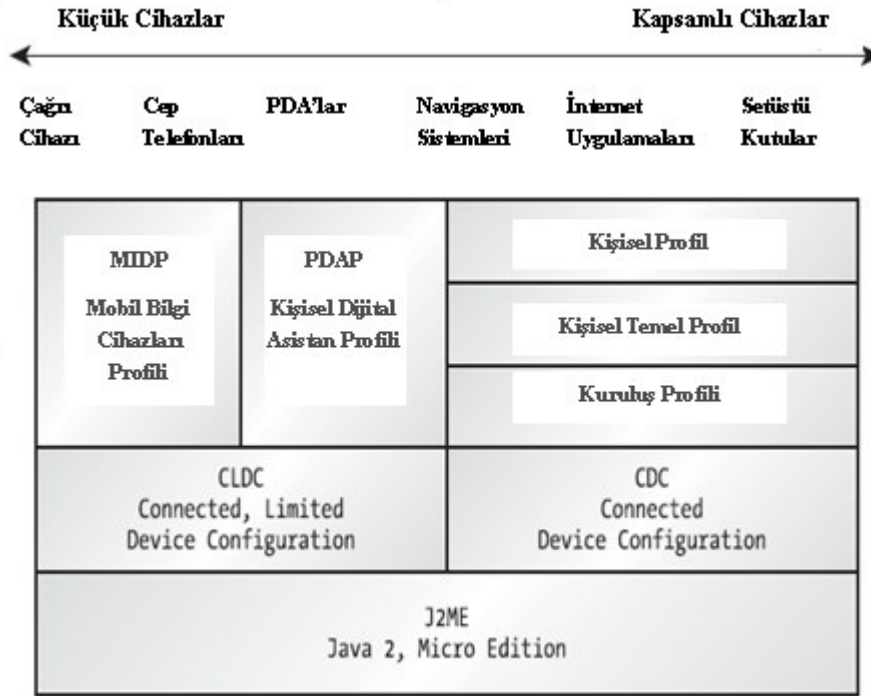
Optional APIs		
JSR	Name	URL
75	PDA Optional Packages for J2ME	http://jcp.org/jsr/detail/75.jsp
82	Java APIs for Bluetooth	http://jcp.org/jsr/detail/82.jsp
135	Mobile Media API 1.1	http://jcp.org/jsr/detail/135.jsp
184	Mobile 3D Graphics	http://jcp.org/jsr/detail/184.jsp
179	Location API for J2ME	http://jcp.org/jsr/detail/179.jsp
120	Wireless Messaging API 1.0	http://jcp.org/jsr/detail/120.jsp
205	Wireless Messaging API 2.0	http://jcp.org/jsr/detail/205.jsp
172	J2ME Web Services APIs	http://jcp.org/jsr/detail/172.jsp
66	RMI Optional Package	http://jcp.org/jsr/detail/66.jsp

J2ME platformunun sanal makinası temel olarak diğer Java sürümleri gibi, çalıştığı cihazın işletim sistemi üzerinde çalışır. Konfigürasyon ve Profil J2ME ortamının temelini oluşturan diğer bileşenleridir. Böylece 3 katmanlı modüler bir yapı (Sanal Makina, Konfigürasyon ve Profil) ile farklı özelliklerdeki cihazlarda uygulama geliştirme imkânı sağlanmış olmaktadır. Şekil 2.2’de konfigürasyon ve profilin bir cihazda nasıl yer aldıkları görülmektedir. Konfigürasyon bir cihazın işletim sisteminin üzerinde, Java sanal makinasını tamamlayıcı özelliklere sahiptir. Profil ise daha üst seviyede işlemlerin yapılmasını sağlar. J2ME uygulamaları temel olarak Profile sınıfları kullanılarak geliştirilir (Mobilpro, 2003).



Şekil 2. 2 Konfigürasyon ve profilin cihazdaki yerleşimi (Mobilpro, 2003)

Şekil 2.3’de şu an en geçerli olan konfigürasyon ve profiller hedef cihaz gruplarıyla birlikte gösterilmektedir (Knudsen and Li, 2005):



Şekil 2. 3 Konfigürasyon, profiller ve hedef cihaz grupları (Knudsen and Li, 2005)

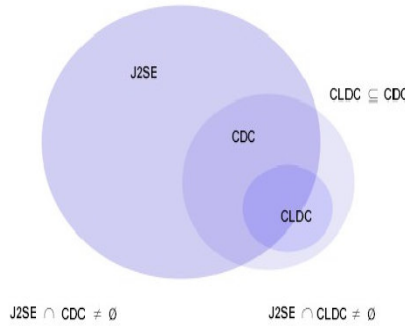
2.2.1 Konfigürasyonlar

J2ME konfigürasyonu küçük cihazlar ailesi için minimum Java platformunu tanımlar. Bu ailenin üyelerinin hepsi yaklaşık olarak aynı bellek ve işlemci gücü gereksinimlerine sahiptir. Konfigürasyon, uygun sistem seviyesindeki özellikleri, Java dili özelliklerini, mevcut sanal makine karakteristik ve özelliklerini ve minimum Java kütüphanelerini tanımlar. Yazılım geliştiriciler belirli bir konfigürasyonu kullanan cihaz ailesi için belirli bir düzeyde sistem desteğinin

olabileceğini varsayarlar. Ayrıca bir konfigürasyon belirli bir cihaz kategorisi için minimum özellikler kümesini belirler.

Konfigürasyon, belirli bir cihaz grubu için gerekli JVM'i ve temel paketler setini belirler. Şu an iki çeşit konfigürasyon bulunmaktadır: CDC (Connected Device Configuration) ve CLDC (Connected, Limited Device Configuration).

Konfigürasyonlar arasında içiçe bir ilişki vardır. J2ME mimarisindeki tüm konfigürasyonlar süper küme altküme sıralanışı içindedirler. Bu kısıtlanmış bir konfigürasyondan daha zengin özellikli bir konfigürasyona doğru ilerlerken taşınabilirliği artırır. Teorik olarak, bir konfigürasyon J2SE platformu kütüphanelerinin benzeri bir destek sağlar. Fakat gerçek dünyada bu mümkün olmaz çünkü J2ME masaüstü bilgisayarlarından daha az güçlü cihazları hedefler. Konfigürasyon spesifikasyonları, J2SE den uyarlanmış tüm Java sınıflarının aynı veya altkümesi şeklinde orjinal J2SE sınıflarını gerektirir. Yani bir sınıf J2SE sürümünde bulunmayan bir metodu ekleyemez. Konfigürasyonlar, kendi spesifikasyonlarına sınıflar ekleyebilirler, yine de konfigürasyonlar mutlaka J2SE'nin bir altkümesi değildirler. Örneğin, şekil 2.4'de her iki konfigürasyonda da cihaz özelliklerine ve kısıtlarına bağlı olarak J2SE' de bulunmayan bazı sınıflar tanımlanmıştır (Dönmez, 2007).



Şekil 2.4 J2SE, CDC ve CLDC konfigürasyonlarının karşılaştırması (Dönmez, 2007)

2.2.1.1 CDC

CDC konfigürasyonlu cihazlar en az 512 KB ROM'a, 256 KB RAM'a ve herhangi bir ağ bağlantısına sahip olmalıdırlar. Bu profilin çekirdeğinde de CVM (Compact Virtual Machine) sanal makinesi bulunmaktadır. CVM, JVM spesifikasyonlarına tamamen uysa da, cihazlar ve ağ uygulamaları için

iyileştirildiğinden gerçekleştirimi J2SE sanal makinesinden tamamen farklıdır. CDC, genellikle CLDC'den daha fazla hafıza ve işlemci gücüne sahip cihazlarda kullanılır. Ayrıca bu cihazlar sürekli ağ bağlantısına sahiptirler.

Bu konfigürasyona ait Java paketleri:

- java.io
- java.lang
- java.lang.ref
- java.lang.reflect
- java.math, java.net
- java.security
- java.security.cert
- java.text, java.util
- java.util.jar, java.util.zip
- javax.microedition.io

2.2.1.2 CLDC

CLDC, CDC'ye göre daha kısıtlı kaynaklara sahip cihazlara yönelik bir J2ME konfigürasyonudur. Bu profilin çekirdeğinde bir Java sanal makinesi olan KVM (Kilobyte Virtual Machine) yatmaktadır. KVM, JVM nin bazı kısımlarının çıkartılarak JVM'nin daha önemli ve gerekli parçalarını içerecek şekilde tasarlanmış halidir.

Bu konfigürasyona ait Java paketleri:

- java.io
- java.lang
- java.lang.ref
- java.util
- javax.microedition.io

CLDC spesifikasyonları hakkında java.sun.com/products/cldc adresinde ayrıntılı bilgi edinilebilir.

KVM olabildiğince Java Sanal Makine Spesifikasyonlarına benzerdir. KVM'nin yetenekleri CLDC spesifikasyonlarıyla tanımlanmıştır. KVM, Java Sanal Makine Spesifikasyonlarından iyileştirme veya kütüphane desteği nedenleriyle CLDC gerektirdiğinde veya izin verdiğinde farklılaşır. KVM cihazlar üzerinde küçük bir yer gerektirir, derleme tercihine ve hedef platforma bağlı olarak 40Kb ve 80Kb arasında yer kaplar. Bu da KVM nin 128Kb toplam belleğe sahip cihazlar üzerinde bile çalışabilmesini sağlar.

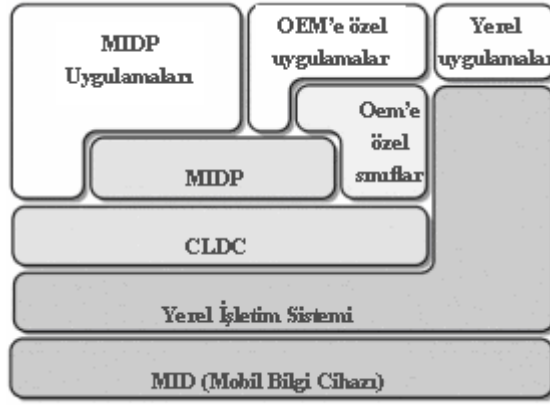
2.2.2 Profiller

Gerek işletim sistemleri gerekse de donanım özellikleri birbirinden farklı olan çok sayıda küçük cihaz için ortak bir uygulama geliştirme alt yapısı profil ile sağlanır. Profiller uygulama geliştirmeye daha yetkin bir ortam sunabilmek için konfigürasyonun üzerine tanımlanan kütüphanelerdir. Konfigürasyon JVM ve temel kütüphaneleri tanımlarken, bir uygulama geliştirilirken ihtiyaç duyulabilecek yetkinlikte kütüphaneleri tanımlamaz. Profiller bu alanda uygulamanın yaşam döngüsü, kullanıcı arabirimi ve kalıcı bellek kütüphanelerini içerirler. Uygulamalar, konfigürasyon ve profilin üzerine yerleşirler ve sadece bu iki düşük seviyeli spesifikasyonların desteklediği sınıf kütüphanelerini kullanırlar. Profillerden biri bir diğerrinin üzerine yerleşebilir. J2ME platformu sadece bir konfigürasyon içerebilir.

Profiller, aşağıdaki bölümlerde detaylı bir şekilde incelenmiştir.

2.2.2.1 MIDP (Mobile Information Device Profile)

MIDP mevcut profiller arasında ilk ve en yaygın olanıdır ve uygulamanın yaşam döngüsü, kullanıcı grafik arabirimleri, iletişim ağı ve kalıcı depolama ile ilgili kütüphaneleri içerir. MIDP CLDC konfigürasyonunun üzerine oturur ve günümüzde sektördeki büyük firmalar tarafından desteklenmektedir.



Şekil 2. 5 MIDP 'ın cihaz içerisindeki duruşu (Mobilpro, 2003)

Şekil 2.5'de MIDP 'ın cihaz içerisindeki duruşu anlatılmaktadır (Mobilpro, 2003). Buradaki bazı kavramlar aşağıda açıklanmaktadır:

OEM'e özel sınıflar ve OEM'e özel uygulamalar: MIDP ve CLDC cep telefonları için standart bir alt yapı oluşturmalarına rağmen, cep telefonu üreticilerinin desteklemek istedikleri bazı özellikleri içermemektedirler. (Ekranın tam olarak kullanımı, cihazın titreşim ve ses özelliklerinin kullanımı gibi) Bu yüzden cihaz üreticileri firmalar, cihazlarını üretirken kendi istediklerini yapabilen Java sınıflarını ekleyerek üretim yapmaktadırlar. (Örneğin NOKIA nın kendi telefonları için kullandığı com.nokia.mid.ui.* paketi gibi) Tabi ki bu sınıfları kullanarak yazılan uygulamalar her cihazda değil sadece o paketi ve sınıfları destekleyen cihazlarda çalışmaktadır. OEM'e özel uygulamalara Nokia'nın kızılotesi kullanarak oynanabilen çok oyunculu yılan oyunu örnek olarak sayılabilir (Mobilpro, 2003).

Yerel Uygulamalar (Native Applications) : Telefonun kendi işletim sistemi özelliklerini kullanarak geliştirilmiş uygulamalardır. Dikkat edilirse uygulama geliştiriciler işletim sisteminden CLDC katmanı ile ayrılmaktadırlar. Yani sadece CLDC nin tanımladığı Java sınıflarının izin verdiği kadarı ile işletim sistemine erişebilirler. Uygulama geliştiriciler ise MIDP sınıflarını kullanarak geliştirirler (Mobilpro, 2003).

MIDP'in içinde bulunan Java sınıfları ile aşağıdaki fonksiyonlar gerçekleştirilebilir:

- Uygulamanın geliştirilmesi (MIDP uygulama kontrol)
- Kullanıcı arayüzü

- Cihaz üzerinde bilgi saklama (Persistent Storage)
- Ağ bağlantısı
- Zamanlayıcılar (Timers)

MIDP'in avantajları (Dönmez, 2007):

Taşınabilirlik: MIDP uygulama geliştirme için diğer araçları kullanma yerine Java'yı kullanmanın avantajı taşınabilirliktir.

Güvenlik: Java; appletler gibi indirilmiş kodu güvenli bir şekilde çalıştırma yeteneğine sahiptir.

2.2.2.2 PDAP (PDA Profile)

Bu profil PDA-stilinde grafiksel kullanıcı arayüzüne ve dokunmatik ekrana sahip kullanıcı arayüzü ve veri depolama yeteneklerine sahip kişisel dijital ajandaların gerekliliklerini karşılamak içindir.

2.2.2.3 FP(Kuruluş profili - Foundation Profile)

Kuruluş profili yeni nesil elektronik cihazlar için geliştirilmiştir. CDC temelli ve grafik arayüzü olmayan cihazlar içindir. Taban profili olarak işlevlerinin yanında yüksek bant genişliği ve fazla bağlantı kuran cihazlar için zengin bir ağ desteği sağlar. Bu profil, kişisel bilgisayarlardan daha küçük cihazlar için diğer profillerle kullanılarak daha zengin bir uygulama ortamı sağlar.

2.2.2.4 PP (Kişisel profil - Personal Profile)

Kişisel profil pek çok kişisel Java kütüphaneleri için yeni bir yurttur. Cep bilgisayarlarını hedef alan kişisel-Java kütüphaneleri J2ME mimarisine uyacak şekilde yeniden oluşturuldu. Kişisel java CDC'nin içinde Kuruluş Profili ve Kişisel Profile ayrıldı. Kişisel profile ek olarak JavaPhone ve JavaTV kütüphaneleri eklendi.

2.2.2.5 PBP (Kişisel temel profil - Personal Basis Profile)

Bu profil CDC konfigürasyonunda ve kuruluş profilinde çalışan cihazların grafiksel yetenekleri için temel bir seviye sağlar. Ayrıca kişisel profilin grafiksel yetenekleri için temel oluşturur.

2.2.3 Midlet geliştirme

Cep telefonları, kişisel dijital ajandalar gibi küçük kapasiteli araçlarda uygulamalar “MIDP” profili kullanılarak, “javax.microedition.midlet.MIDLET” sınıfı genişletilerek oluşturulur. Bu sınıf, cihazda bulunan Uygulama Yönetim Yazılımı ile MIDP uygulamaları arasında bir arayüz gibi davranır. MIDP uygulamaları oluşturulurken, başlangıç noktası bir MIDlet’tir.

Mobil cihazlar için geliştirilen Java uygulamalarına MIDlet denir. J2ME platformuna uygulama geliştirmek için JDK (Java Development Kit) ve J2ME Wireless Toolkit’in bilgisayara kurulması gerekir. J2ME WTK, MIDlet yaratmak için önemli araçları içerir. MIDlet geliştirme ortamını ve 3D ve Mobile Media uygulamaları gibi seçimlik kütüphaneler için gerekli seçimlik paketleri sağlar. Ayrıca MIDlet’leri imzalayarak uzak mobil cihaza kurulmadan önce onların onaylanmasını sağlar (Dönmez, 2007).

MIDlet geliştirme aşamaları:

- Java uygulamasını yazma.
- Uygulamayı derleme.
- Ön-geçerliliğini(pre-verify) yapma.
- JAR dosyasına paketleme.
- Uygulama tanımlayıcısını yaratma.
- MIDlet suitini yayımlama.

MIDlet, MIDP uygulamalarının temel noktasını oluşturmak için yaratılan soyut bir sınıftır. MIDlet sınıfı, “javax.microedition.MIDlet” paketi içinde yer alır. Bu nedenle gerçekleştirilmek istenen bir MIDlet’te çizelge 2.4’de verilen kod parçaları mutlaka yer almalıdır.

Çizelge 2. 4 MIDlet’in geliştirimi

```
import javax.microedition.midlet.MIDlet;

public class MerhabaDunya extends MIDlet {
}
```

MIDlet’ler “startApp()”, “pauseApp()” ve “destroyApp(boolean b)” olmak üzere üç adet önemli metod içerirler.

startApp(): MIDlet’in yaşam döngüsü boyunca defalarca kez çağırılabilir. İlkeme işlemlerinin gerçekleştirildiği bir metod değildir. Uygulamanın çalıştırılacağı cihazda, bir MIDlet’in başlatılacağı mesajı alınınca Uygulama Yönetim Servisi bu metodu çağırır.

pauseApp(): Kullanıcı veya cihaz, çalışan uygulamayı kesip başka bir görev gerçekleştirmek istediğinde cihaz tarafından çağırılan bir metoddur. Bu metod çağırıldığında MIDlet “duraklatılmış” duruma geçer.

destroyApp(boolean b): Kullanıcı uygulamayı kapatmak istediğinde veya sistem (bir nedenden dolayı) uygulamanın kapatılmasını isterse bu metod çağırılır. Bu metod, çalışmakta olan uygulamaya kullandığı kaynak varsa, bunları düzenlemesi için olanak tanır. Parametre olarak “TRUE” değerini alırsa, uygulamanın kaynakları temizlemekten başka çaresi yoktur. Parametre olarak “FALSE” değerini alırsa, uygulama çalışmaya devam etmek için “MIDletStateChangeException” fırlatabilir. Eğer metoda parametre geçirilmezse uygulamanın temizlemesi gereken kaynağı yok demektir.

Basit bir MIDlet için temel olarak yukarıda bahsedilen üç metod yeterlidir. Bu üç metodun kullanıldığı “MerhabaDunya” MIDlet’inin kodu aşağıda çizelge 2.5’de verilmiştir:

Çizelge 2. 5 MerhabaDunya.java

```

import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.*;

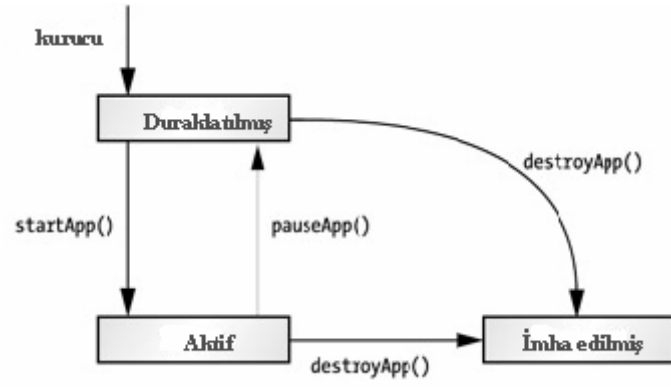
public class MerhabaDunya extends MIDlet {
    private TextBox textbox;
    public MerhabaDunya() {
        textbox = new TextBox("", "Merhaba Dünya!", 20, 0);
    }
    public void startApp() {
        Display.getDisplay(this).setCurrent(textbox);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
}

```

Bir MIDlet yaşam döngüsü boyunca aşağıdaki durumlardan geçer (Şekil 2.6) (Knudsen and Li, 2005):

- MIDlet çalıştırılmak üzereyken, bir örnek yaratılır. MIDlet'in kurucu fonksiyonu (constructor) çalıştırılır ve MIDlet duraklatılmış (paused) durumundadır.
- Sonra, uygulama yöneticisi startApp()'ı çağırdıktan sonra MIDlet aktif durumuna girer.
- MIDlet Aktif iken, uygulama yöneticisi pauseApp()'ı çağırarak çalışmasını durdurabilir. Bu MIDlet'i yeniden duraklatılmış durumuna koyar. notifyPaused() çağrılarak MIDlet kendisini duraklatılmış durumuna geçirebilir.
- destroyApp() çağrılarak, uygulama yöneticisi MIDlet'in çalışmasını sona erdirebilir, bu durumda MIDlet imha edilmiş durumuna geçer. MIDlet yok

edildiğinde çöp toplama mekanizması (garbage collection) beklenir. notifyDestroyed() çağrılarak MIDlet kendini yok edebilir.



Şekil 2. 6 MIDlet Yaşam Döngüsü (Knudsen and Li, 2005)

3. WEB SERVİSLERİ

İnternet ve web alanında son yıllarda meydana gelen hızlı teknolojik ilerlemelerle birlikte yazılımlar daha çok web-tabanlı, yani web ve internet üzerinden etkileşimli, olarak geliştirilmeye başlanmıştır. Son yıllarda bu alanda meydana gelen önemli bir gelişme ise web-tabanlı yazılımların işlevlerini web üzerinden diğer yazılımlara standart protokoller aracılığı ile kullandırabilmeleri olmuştur. Buna kısaca “Web Servisleri” deniyor (Doğdu, 2010).

Bunun için bir yazılım sistemi, yapabildiği işlevleri kendi üzerinde tanımlanan web servisleri aracılığı ile hizmet olarak diğer yazılım uygulamalarına açmaktadır. Örneğin, Google firması geliştirdiği harita hizmetini (Google Maps) web servisi olarak açmıştır. Diğer yazılımlar, mesela bir web-tabanlı seyahat firması, kiraladığı otellerin harita üzerindeki yerlerini Google Maps web servislerini kullanarak harita üzerinde gösterebilir. Bunun için kendi harita hizmetini yazmasına gerek yoktur, mevcut bir web servis hizmetini, belli bir ücret karşılığı kullanarak, bunu sağlayabilir (Doğdu, 2010).

XML Web servisleri, bilginin kullanıcıya sunumundan önce uzak bilgisayarların birbirlerine formatlanır ve analiz edilebilir dinamik bilgi sağladığı servis tabanlı web için tasarlanmışlardır. XML Web servisi, bir bilgisayarın bilgiyi web üzerinden XML formatında diğer bilgisayarlara dağıtmasının kolay yoludur. Teknik olarak, .NET üzerindeki XML Web servisi istemcilere HTML yerine XML döndüren ASP .NET sayfasıdır (Nagel et al., 2008). Web servisleri tamamen XML tabanlıdır ve yazılım geliştiriciler için SOAP ve WSDL tanımlarına dayanan farklı çeşitlerde esneklik ve güvenlik mekanizmaları sağlar (Nayyeri, 2008).

Web servislerinin temelinde XML (Extensible Markup Language) ve web servis geliştirme platformu (.Net, JAVA, J2EE, JAVA Micro vs) vardır (Şen, K, 2010). XML Web Servisleri, heterojen uygulamaların birlikte çalışmasını sağlayabilmek için veri alışverişinde XML tabanlı mesajlaşmayı kullanır (Thangarathinam, 2006). Yıllardır bileşenler sayesinde iletişim sağlayan web siteleri XML standardında daha güvenli ve hızlı iletişim yapabilmektedir (Şen, K, 2010).

XML web üzerinde bilgi sunmak amacıyla hazırlanmış, ortak belirlenmiş standartlar kapsamında veri paketleme ve aktarım için kullanılan bir işaretleme dilidir. XML, bilginin içeriğini tanımlamak ve bilginin farklı sistemler arasında

taşınması için kolay ve standart bir alt yapı sunmak amacıyla ortaya çıkmıştır. Web servislerinde, hizmet alıcının belirlediği formattaki veriler yerine XML formatındaki veriler ile iletişim kurulacaktır. Bu birçok karmaşıklığa çözüm olmakta ve standardizasyonu sağlamaktadır.

XML'in temel özellikleri şöyle sıralanabilir (Küçüksille vd., 2009):

- XML , HTML gibi bir işaretleme dilidir.
- XML, veriyi tanımlamak için tasarlanmıştır.
- Önceden belirtilmiş etiketler (tag) yoktur. Kendi etiketlerinizi kendiniz oluşturmanız gerekir.
- XML de veriyi tanımlamak için doküman tipi tanımlaması DTD (Document Type Definition) veya XML Schema kullanılması gerekmektedir.
- XML DTD veya XML Schema ile birlikte kendini tanımlayıcı bir yapıda olması gerekmektedir.

Microsoft tarafından geliştirilen .NET yazılım geliştirme ortamı, yazılım geliştirme alanında ve web servislerinin hazırlanmasında çığır açmıştır, ancak aslında daha öncelerden IBM, Oracle gibi firmalar web servislerini aktif olarak kullanıyorlardı. .Net platformunun sloganı 'her yerden, her zaman, herşeye erişebilmek' olmuştur. Bu anlayışla artık internet ortamında yapılan tüm işler mobil ortamda da kolayca ve daha az emek harcanarak yapılabilmektedir. Microsoft'un, web servisleri arenasında karşısına çıkan en dişli rakip ise Sun'dır. Sun firmasının geliştirdiği JAVA platformunda EJB (Enterprise JavaBeans) bileşenleri yardımıyla da web servisler geliştirilebilmektedir (Şen, K, 2010).

Web servisleri birlikte-çalışabilir, dağıtık uygulamalar sunan, herhangi bir dilden ve platformdan bağımsız çalışan, web erişimli uygulamalardır (Şen, K, 2010). Web Servisleri HTTP üzerinden herhangi bir istemci tarafından çağrılan nesne ve fonksiyonlardır ve SOAP (Simple Object Access Protocol) üzerine kurulmuşlardır (Thangarathinam, 2006).

XML Web Servisleri kullanıcıya ölçeklenebilir, düşük karmaşıklığa sahip platform bağımsız uygulamalar yazmaya izin veren programlanabilir bileşenlerdir. XML Web Servisleri HTTP, XML, XSD, SOAP ve WSDL (Web Services Description Language) gibi Internet standardındaki protokolleri kullanarak birbirlerinden tamamen farklı uygulamaların mesaj paylaşma ve değiştirmelerine izin verirler (Thangarathinam, 2006).

Farklı programlama dilleri ile yazılan ve farklı platformlar üzerinde çalışan uygulamaların entegrasyonunda XML Web Servisleri kullanılır. Ayrıca XML Web Servisleri internet üzerinde konuşlandırılacağı gibi yerel ağ üzerinde de konuşlandırılabilir (Thangarathinam, 2006).

XML Web Servisleri, birbirleriyle ilgisi olmayan platformlar, işletim sistemleri ve programlama dilleri arasında uyum içinde bilgi paylaşımının bir yoludur. Web Servislerinin önemli bir özelliği istemcinin Web Servisinin yazıldığı dili bilmesine gerek olmayışdır. İstemcinin sadece kullanacağı Web Servisinin yerini ve servis üzerinden çağıracağı fonksiyonun adını bilmesi yeterlidir. İstemci tarafında tek istenen istemcinin gelen XML dökümanını ayrıştırabilmesi ve gelen XML elemanlarını kullanılan platform ve/veya dile özel tiplere dönüştürmesidir (Thangarathinam, 2006).

3.1 Web Servislerinin Kullanım Nedenleri

Güvenli ortamda iletişim sağlamak: Dağıtık ortamlarda, binlerce kullanıcıya hizmet veren uygulamalarda, istemci ile sunucu arası iletişimlerde güvenlik en önemli sorundur. Web servislerinde SSL sertifikaları kullanılabilir. Ayrıca "windows yetkilendirme derecesi, passport ve form yetkilendirmesi" yapılabilir. İstemci arayüzünden girdileri alan orta katman bileşenlerinin web servisleri biçiminde hazırlanıp direk kullanıcı arayüzünden çağırılması birçok güvenlik sorunun çözümü olacaktır.

Uygulama Entegrasyonu: Farklı dillerde yazılmış, farklı sistemlerde çalışan uygulamalarda entegrasyonu sağlamak için uygulamaların bazı işlevlerini ve verilerini web servisleri aracılığıyla sunması gerekir. Böylece entegrasyon için standart yöntemler sunulmuş olur.

B2B Entegrasyonu: Birçok işi ilgilendiren iş süreçlerinin entegrasyonu B2B entegrasyonudur. B2B entegrasyonu için web servislerini kullanmak ucuz, az zamanda, birlikte çalışabilir bir şekilde hayati iş süreçlerinin yetkili tedarikçi ve müşterilere sunulmasını sağlayacaktır.

Yeniden Kullanılabilirlik: Web servisleri yazılımın ihtiyaç duyduğu veriyle birlikte yeniden kullanılabilir özellik kazanır. Örneğin kullanıcıdan alınan bir adres bilgisinin onaylanması için bir adres onaylama web servisi kullanılırsa tüm adres

bilgilerinin uygulama sunucusunda saklanmasına gerek olmayacaktır (Şen, K, 2010).

3.2 Web Servislerinin Yararları

Birlikte Çalışabilirlik: Herhangi bir web servisi platformdan bağımsız diğer web servisleriyle XML kullanarak iletişim sağlayabilir.

Çalışma zamanlı entegrasyon: Çalışma anında yeni web servisleri eklenip değiştirilebilir. Diğer web servisleri bu durumdan hiçbir şekilde etkilenmez.

Kolay dağıtım: Web servisleriyle geliştirme yapmaksızın uygun bileşen, web servisleri kullanarak hızlı ve ucuz servisler sunulabilir.

Etkin uygulama geliştirme: Web servislerinin tekrar kullanılabilirliği etkin bir uygulama geliştirme sunmuştur.

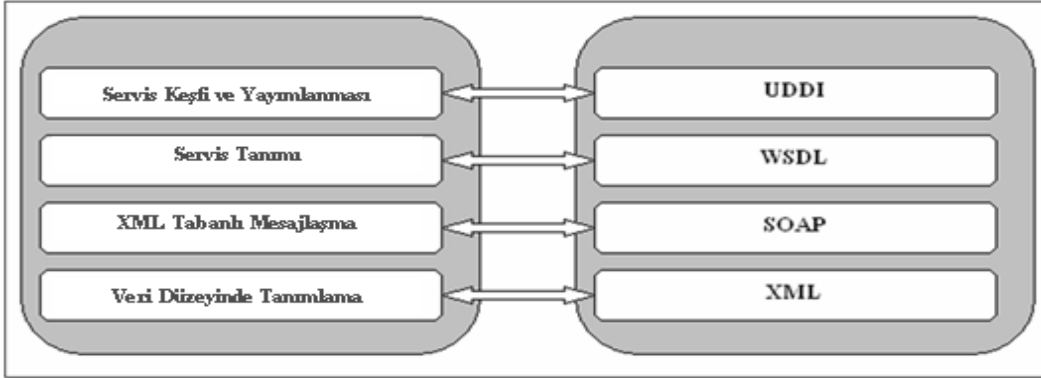
Güçlü kapsama (Encapsulation): Web servisi için servisin sunduğu hizmet, servisin bunu nasıl sunduğundan daha önemlidir. Karmaşıklık web servisinin içinde kalmıştır. Önemli olan isteğin karşılığının doğru ve hızlı bir şekilde alınabilmesidir (Şen, K, 2010).

3.3 XML Web Servislerinin Alt Yapısı

Bir işletmeye ait farklı bölümlerdeki sistemlerin ya da farklı işletmelerdeki sistemlerin entegrasyonunda web servisleri kritik bir çözüm olarak uygulanır. Web servisleri, SOAP, WSDL ve UDDI (Universal Description, Discovery and Integration) gibi açık ve esnek standartları kullanarak, bilginin farklı platformlar ve uygulamalar arasında serbest bir şekilde değiş tokuşuna olanak sağlar (Choi and Han, 2008).

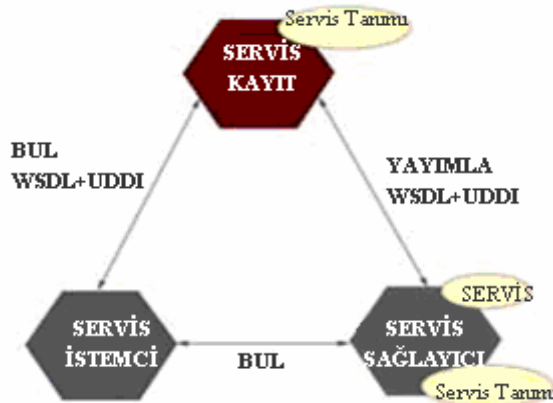
Web servis mimarisi bağlantı, iletişim ve tanımlamalar açısından bazı prensip ve standartları barındırmaktadır. Şekil 3.1' de gösterildiği üzere (Trivedi, 2003); XML, servis sağlayıcıları ve servis istemcilerin bağlanma ve bilgi değiş tokuşu yapabilecekleri ortak bir dili sağlar. SOAP, farklı platformlardaki sistemlerin birbirleri arasında servis taleplerini karşılayabilecekleri ortak bir protokolu sağlar. WSDL, fonksiyon isimlerinin, ihtiyaç duyulan parametrelerin ve sonuçlarının tanımlanmasına imkân veren bir altyapıdır. UDDI, kullanıcılara

ihtiyaç duyulan servislerin belirlenmesini ve erişilebilmesini sağlar (Altınova H. vd, 2008).



Şekil 3. 1 XML Web Servisinin Altyapısı (Trivedi, 2003)

Web servisi mimarisi temel olarak üç rol ve işlemi kapsar. Söz konusu roller servis sağlayıcı, servis istemcisi ve servis kayıt birimidir (Şekil 3.2). Servis Sağlayıcı (Servis Provider), istemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı kendi sitesinde bulunan web servisleri tanımını servis kayıt birimine (service registry) kaydederek bu servisin nasıl çağrılacağını belirtir. Servis istemcisi (Service Requester), servis sağlayıcısında bulunan web servislerini çağırarak bunları kullanan istemci uygulamalardır. Web servisinin nasıl çağrılacağını ve ilgili parametreleri servis kayıt biriminden arayarak bulur ve web servisini çağırır. Servis Kayıt Birimi (Service Registry), servis sağlayıcılarının yayınladıkları web servisi tanımlarını saklar ve aranıp bulunmasını sağlar. Servis sağlayıcıları servis kayıt birimini tarayarak istediği servisler hakkında bilgi alabilir. Servis kayıt birimi her servisin nasıl çağrılacağı konusunda tanım bilgileri içerir (Altınova H. vd, 2008).



Şekil 3. 2 Web Servis Bileşenleri (Altınova H. vd, 2008)

XML Web Servis modelinin en önemli özelliklerinden biri hem istemci tarafının hem de XML Web Servis tarafının birbirlerinin gerçekleştirim detaylarından habersiz olmasıdır. XML Web Servisi alt yapısı, istemci tarafındaki uygulamanın XML Web Servisinin yerini öğrenmesi ve servis hizmetlerinden yararlanabilmesi için birçok bileşen sağlar. Bu bileşenler aşağıdaki bölümlerde açıklanmaktadır (Thangarathinam, 2006).

3.3.1 XML web servisleri dizinleri

Bu dizinler XML Web Servisleri hakkında yayımlanan bilgileri saklamak için merkezi bir yer sağlarlar. UDDI şartnameleri XML Web Servislerinin nasıl yayımlanacağı ile ilgili bir rehber tanımlar. XML Şema UDDI ile ilgili olarak XML Web Servisine ulaşılabilmesi için yayımlanması gerekli dört çeşit bilgi tanımlar. Bu bilgi konu bilgisi, servis bilgisi, bağlantı bilgisi ve servis belirtimlerini içerir. Microsoft'un kendisinin geliştirdiği UDDI belirtimine <http://uddi.microsoft.com> adresinden ulaşılabilir (Thangarathinam, 2006).

3.3.2 XML web servislerin keşfi

İstemciler XML Web Servislerini tanımlayan dökümanlarını WSDL'i kullanarak yerleştirebilirler. Bu keşif süreci, istemcinin herhangi bir XML Web Servisinin varlığından ve belirli bir XML Web Servisinin lokasyonundan haberdar olmasını sağlar (Thangarathinam, 2006).

3.3.3 XML web servis tanımı

Bu bileşen XML Web Servisleri üzerinde hangi operasyonların gerçekleştirilebileceği bilgisini sağlar. XML Web Servis tanımı, Web Servisinin anlayabileceği mesaj formatını belirleyen XML dökümanıdır. Örneğin, tanım dökümanı XML Web Servisi üzerinden fonksiyonlar çağrılırken kullanılan SOAP mesaj şemalarını belirler (Thangarathinam, 2006).

3.3.4 XML web servisi bağlantı formatları (Wire formats)

Birbirlerinden tamamen farklı sistemlerin iletişimde, XML Web Servisleri açık bağlantı formatlarını kullanırlar. Açık bağlantı formatları, HTTP, SOAP gibi genel Web standartlarının herhangi bir sistem tarafından desteklenmesi için anlaşılması gereken protokollerdir. HTTP-GET ve HTTP-POST protokolleri,

parametrelerin isim değ er çiftleri şeklinde iletimine izin veren standart Web protokolleridir. HTTP-GET protokol  URL tarafında şifrelenmiş parametrelerin isim değ er çiftleri şeklinde XML Web Servisine gönderimini sağlar. HTTP-GET protokol  isim değ er çiftleri parametrelerinin XML Web Servisinin URL'ine eklenmesini ister. Aynı şekilde isim değ er çiftleri şeklindeki parametrelerin URL tarafında şifrelenmesi ve XML Web Servisine gönderimi HTTP-POST protokol  tarafından da yapılabilir; ancak bu durumda parametreler XML Web Servisinin URL'ine eklenerek değ il de ger ek mesajın i ersinde g nderilir (Thangarathinam, 2006).

3.3.5 SOAP

Web servisleri ve servisleri isteyen kullanıcı arasındaki XML formatındaki bilgi alışverişinin formatını belirleyen standarta, SOAP denir. SOAP standartları W3C standartlar komitesi tarafından belirlenmiştir. B t n bu standartları uygulamaya ge iren SOAP sayesinde web servisine, platformdan bağımsız çağırımlar yapılabilir. Yani .NET ortamında geliştirilen bir web servisine JAVA ile geliştirilen bir programdan ulaşımak pekala m mk n olabilmektedir ( enyurt, 2010).

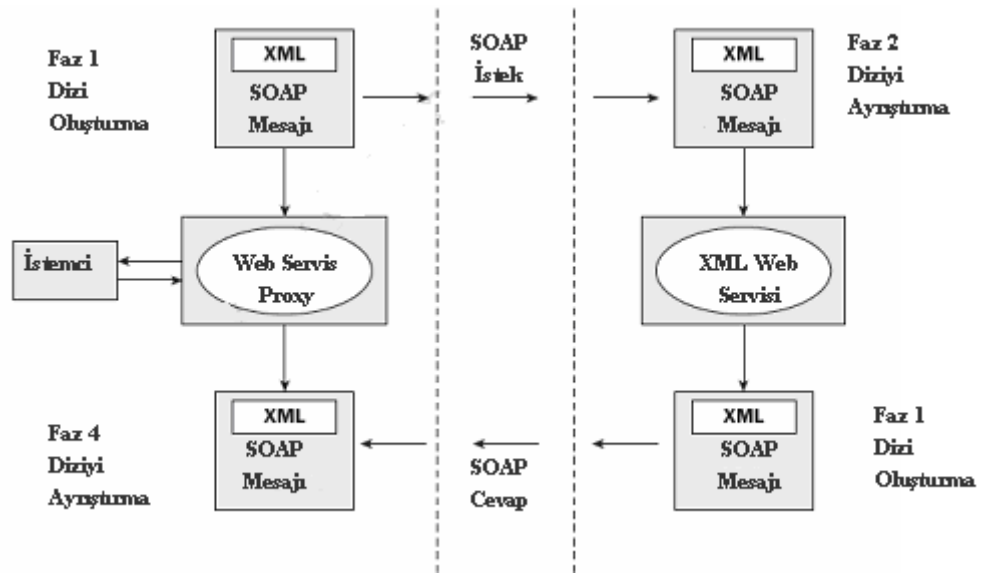
Web servisi uygulamasını çağır an istemcinin kullandığı teknoloji ne olursa olsun SOAP ile HTTP  zerinden bu isteğini iletebilmektedir. Daha  nceden uzaktan veriye erişim i in birtakım standartlar mevcuttu, bunlar k   k ağılarda veya yerel ağılarda uygun sonu lar verebiliyordu, DCOM, CORBA ve J2EE 'nin RMI standartları bunlara  rnek verilebilir. RMI  zerinden ancak JAVA nesnelere ulaşabiliyorduk, bir COM nesnesine RMI ile ulaşabilmek imk nsızdı. Ayrıca bunlar TCP/IP portları  zerinden iletişime ge tikleri i in internet gibi ortamlarda  alıştırıldığında bir takım g venlik sorunları ortaya  ıkabiliyordu, mesela bir ok firewall TCP/IP  zerinden bu t r erişimlere izin vermiyor. SOAP ile artık bu sorun aşılmış durumdadır. SOAP ile veriler HTTP  zerinden aktarıldığı i in hi bir g venlik sorunu  ıkmamaktadır ( enyurt, 2010).

Yapısal ve  eşitli bilgilerin internet  zerindeki uygulamalar arasında değışimine olanak sağlayan SOAP protokolleri d rt par adan oluşur. İlk par a, mesajı i eren zarfı tanımlayan zorunlu kısımdır. İkinci par a, uygulamaya  zel veri tiplerini şifrelemek i in kullanılan, zorunlu olmayan veri şifreleme kurallarını tanımlar.    nc  par a XML Web Servisleri arasında değıştirilen mesajların

istek/cevap desenini tanımlar. Seçmeli olan dördüncü parça ise, SOAP ve HHTP protokolleri arasındaki bağlantıları tanımlar (Thangarathinam, 2006).

3.4 İstemci ve XML Web Servisi Arasında İletişim

İstemci ve XML Web Servisi arasındaki iletişim süreci RPC'ye (Remote Procedure Call - Uzaktan Prosedür Çağırımı) benzer. İstemci, XML Web Servisi üzerinden bir fonksiyonu çağırmak için kendi makinesinden Web Servisinin proxy nesnesini kullanır. Şekil 3.3'de istemci ve XML Web Servisi arasındaki iletişim sürecini gösterilmektedir (Thangarathinam, 2006).



Şekil 3.3 İstemci ve XML Web Servisi arasındaki iletişim (Thangarathinam, 2006)

Şekil 3.3'de görüldüğü üzere, istemci ve XML Web Servisi arasındaki etkileşim farklı aşamalardan oluşmaktadır. Bu aşamalarda gerçekleştirilen iş paketleri şunlardır:

1. İstemci kendi yerel bilgisayarını üzerinde XML Web servis proxy sınıfından bir nesne yaratır.
2. İstemci proxy nesnesi üzerinden fonksiyonu çağırır.
3. İstemci sistemi üzerindeki XML Web servis altyapı sistemi fonksiyon çağırımını ve değişkenlerini SOAP istek mesajı şeklinde dizi haline getirir ve bu mesajı ağ üzerinden XML Web servisine gönderir.
4. XML Web servisinin bulunduğu sunucudaki altyapı sistemi gelen SOAP mesajını birleştirir ve XML Web servis nesnesi yaratır. Daha sonra buradaki

altyapı sistemi Web servis üzerinden değişkenleri de geçirerek gerçek web servis fonksiyonunu çağırır.

5.XML Web servisi fonksiyonu çalıştırır ve oluşan çıktı parametrelerini ve geri dönüş değerlerini altyapı sistemine döndürür.

6.Alt yapı sistemi gelen çıktı parametrelerini ve geri dönüş değerlerini dizi haline getirerek bir SOAP cevap mesajı oluşturur ve bu mesajı ağ üzerinden istemciye gönderir.

7.İstemcinin makinesindeki altyapı sistemi gelen çıktı parametrelerini ve geri dönüş değerlerini içeren SOAP cevap mesajını birleştirir ve bunları proxy nesnesine gönderir. Proxy nesnesi de çıktı parametrelerini ve geri dönüş değerlerini istemciye gönderir.

Yukarıdaki adımlardan anlaşılacağı üzere .NET Framework tarafından sağlanan XML Web Servis altyapı sistemi Web servislerinin oluşturulması, konuşlandırılması ve kullanılmasında önemli bir rol sahibidir (Thangarathinam, 2006).

3.5 .NET XML Web Servis İsim Uzayları (Namespaces)

.NET platformu üzerinde XML Web servisinin yetenekleri beş isim uzayı şeklinde aşağıda açıklanmaktadır (Thangarathinam, 2006).

System.Web.Services : Web servisini oluşturmak için gerekli WebMethodAttribute, WebService ve WebServiceAttribute gibi minimum sınıf setidir.

System.Web.Services.Configuration : ASP.NET XML Web servisinin çalışma zamanındaki davranışlarını düzenlemeye izin veren sınıfları sağlar.

System.Web.Services.Description : Web servisinin tanımlanmasında kullanılan WSDL dökümanı ile etkileşimde bulunmayı izin veren sınıfları içerir.

System.Web.Services.Discovery : Verilen herhangi bir web sunucusu üzerindeki web servisini bulmayı sağlayan sınıfları içerir.

System.Web.Services.Protocols : XML Web servisi ve onun kullanıcısı arasında veri iletiminde kullanılan HTTP_GET, HTTP_POST ve SOAP gibi protokolleri tanımlayan sınıfları içerir (Thangarathinam, 2006).

3.6 ASP .NET Web Servisinin Geliştirilmesi

Web Servisi public ve protected fonksiyonları olan sıradan bir sınıftır. Normalde Web Servis sınıfı .asmx uzantılı olarak kaydedilen kaynak kod içersine yerleştirilir. Web Servis dosyaları ASP.NET'e kodun çalışma zamanındaki davranışıyla ilgili bilgi vermek için aşağıda gösterildiği şekilde “@ WebService” direktifini içermek zorundadır (Thangarathinam, 2006):

```
<%@ WebService Language="C#" Class="MyWebService" %>
```

Yukarıdaki dil (Language) özelliği C#, VB ya da JS olarak değiştirilebilir. Burada sınıf özelliği olarak tanımlanan isim ile ana sınıfın eşleşmesi ve bu sınıfın public olması gereklidir.

Visual Studio .NET ile Web servisi geliştirmek çok basittir. Bir web servisinin nasıl geliştirileceği aşağıda adım adım anlatılmaktadır:

Visual Studio .NET'i çalıştırıp File menüsünden New Project seçeneği seçilir. Açılan pencerede proje tipi Visual C# projects, template olarak da ASP.NET Web Services ya da Web Services kısmı seçilir.

Proje oluşturulurken Visual Studio otomatik olarak birtakım sayfalar ve sınıflar oluşturur. Bu sayfalar Solution Explorer bölümünden görülebilir. Visual Studio otomatik olarak oluşturduğu bu sayfa ve sınıfları aşağıda detaylı bir şekilde açıklanmaktadır:

Global Sınıfı : Bu sınıf System.Web.HttpApplication sınıfından türemiştir. Bu sınıf Global.asax sayfasının içine yerleştirilmiştir. Eski Asp sayfalarında kullanılan global.asa ile eş değerdedir.

Service1 Sınıfı : Bu sınıf System.Web.Services.WebService sınıfından türemiştir. Bizim geliştireceğimiz web servisini temsil etmektedir.

AssemblyInfo.cs Dosyası : Bu dosyada Build işlemi sırasında oluşacak Assembly'lerin sürüm bilgileri saklanır.

Web.config Dosyası : Web Servisimizin çalışma ayarlarını içerir.

CsWS.vdisco Dosyası : Web Servisimizin organizasyonu ile ilgili bilgiler içerir(Klasör adları, isim alanları vs.).

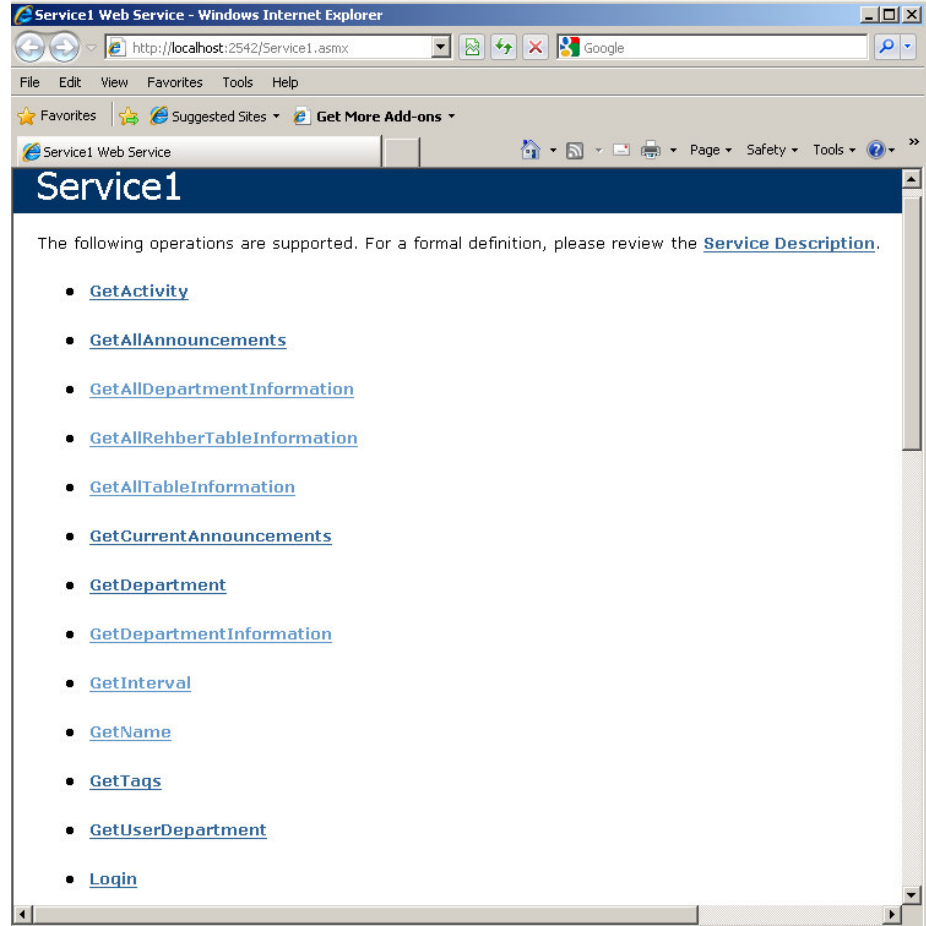
Service1.asmx Dosyası : Web Servisimizin asıl yoludur. Web servisimize ait fonksiyonlar bu sayfadadır.

Çizelge 3.1’de yukarıda oluşturduğumuz web servisi içerisinde çalıştırılan, kullanıcının sisteme girişinin kontrol edildiği Login isimli web metod görülmektedir. Web servisindeki metodların web metodu olarak algılanması için fonksiyonların başında “WebMethod” ifadesi bulunmalıdır. Aksi halde oluşturulan web servisinin arayüzünde bu metod yer almayacaktır.

Çizelge 3. 1 Login isimli web metoda ait kod

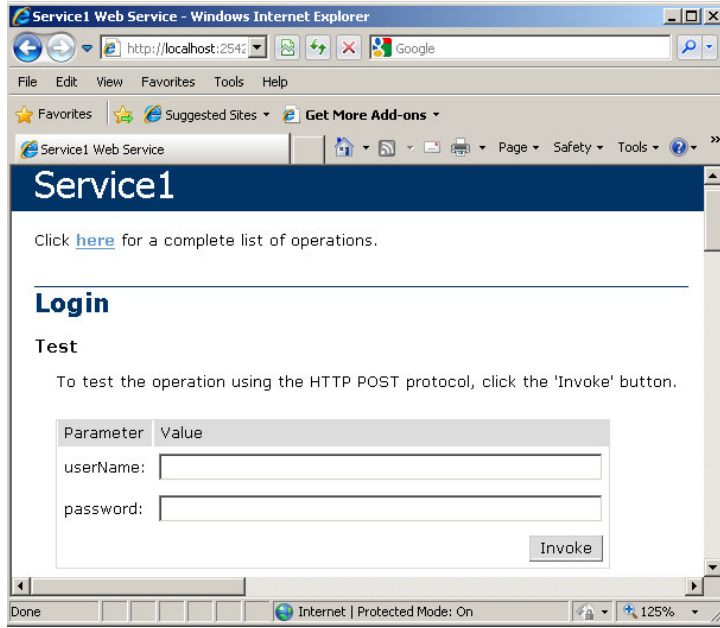
```
[WebMethod]
public bool Login(string userName, string password)
{
    return Membership.ValidateUser(userName, password);
}
```

Oluşturulan proje derlendikten sonra çalıştırılabilir. Web servisi için gerekli dll dosyası proje derlendiğinde otomatik olarak bin klasörü içerisinde oluşturulmaktadır. Oluşturulan web servisini test etmek için projenin çalıştırılması gerekmektedir. Proje çalıştırıldıktan sonra tarayıcı, web servisine ilişkin metodların gösterildiği şekil 3.4’de gösterilen sayfayı görüntüler.



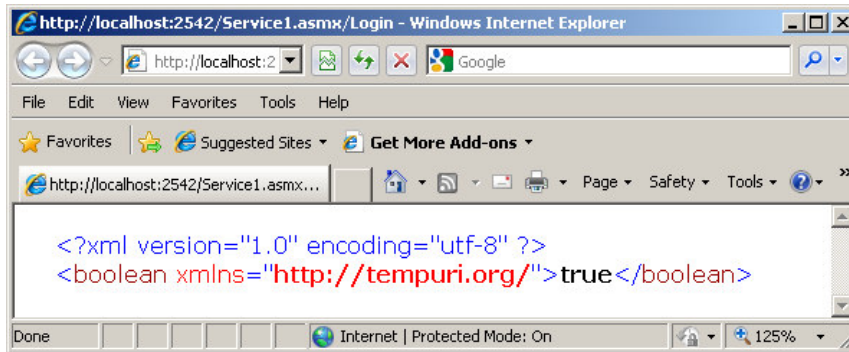
Şekil 3. 4 Tarayıcı üzerinde web servisi içersindeki metodların gösterimi

Web servisinin şekil 3.4’de gösterilen açılan sayfa üzerindeki ilgili metodu, sayfadaki bağlantı üzerine tıklanarak çalıştırılabilir. Metoda tıklandığında açılan sayfadan şekil 3.5’de görüldüğü gibi metodun aldığı parametre değerlerinin girişinin yapılabileceği bir sayfa ekrana gelir.



Şekil 3. 5 Tarayıcı üzerinde bir web metodun gösterimi

Şekil 3.5’de gösterilen sayfa üzerinde kullanıcı adı ve şifre değerleri ilgili kutulara girildikten sonra “Invoke” tıklandığında “Login” metodunun geri dönüş değerini gösteren sayfa XML belgesi olarak şekil 3.6’da gösterildiği şekilde açılmaktadır.



Şekil 3. 6 Tarayıcı üzerinde bir web metodun geri dönüş değerinin gösterimi

Visual Studio.NET ile geliştirilen bir uygulamaya web servisinin nasıl ekleneceği aşağıda adım adım anlatılmaktadır:

- Solution Explorer’da projenin üzerinde sağ tıklanır.
- Açılan menüden “Add Web Reference” seçeneği seçilir.

- Ekrana gelen pencerede “Web services on the local machine” bağlantısına tıklandığında localhost üzerindeki tüm web servislerinin bir listesi görülür.
- Uygulamaya eklemek istenilen web servisi seçilir ve sonra “Add Reference” tıklanarak web servisi uygulamaya eklenir.

Yukarıda açıklandığı şekilde web servisi uygulamaya eklendikten sonra metodu kullanılmak istenilen sınıfa ilişkin bir nesne oluşturularak, ilgili sınıfın metodlarına erişim sağlanır. Aşağıda çizelge 3.2’de buna ilişkin kod gösterilmektedir.

Çizelge 3. 2 Web metoda erişim

```
phpwslocal.EgeDuyuruWS phpwslocal = new phpwslocal.EgeDuyuruWS();  
try  
{  
    str = phpwslocal.GetAllAnnouncements("");  
}  
catch (Exception exc)  
{  
    str = null;  
}
```

4. KİŞİSELLEŞTİRME

Kişiselleştirme, bir uygulamanın sunumunun ve uygulama tarafından sağlanan bilgi veya servislerin uygulamayı kullanan kullanıcının açık veya üstü kapalı tercihleri doğrultusunda değiştirilebilmesidir. Ürünlerdeki kişiselleştirme ve servislerin dağıtımı günümüzün en önemli konularından biridir. Kişiselleştirme, kullanıcıların bireysel tercih ve değerlerini ifade edebilmek ya da bazı durumlarda bir gruba dahil olduklarını göstermek için duyulan ihtiyaca iyi bir çözüm sunmaktadır (Jacques, 2008).

Kablosuz mobil cihazların çıkışıyla 10 yıl içersinde bilgiye erişim ve iş yönetiminde, mobil ticaret (m-commerce) yaygın bir yol haline geldi. Şu an mobil kullanıcılar istedikleri bilgiye, istedikleri yerden erişebiliyorlar. Mobil ilan, satış ve bahis oyunları oldukça popüler, böyle olunca mobil servis sağlayıcılar arasındaki yarış da giderek şiddetleniyor. Bu yüzden bazı servis sağlayıcılar kullanıcılarının içeriği kişiselleştirebilmeleri için kişiselleştirme tekniklerini kullanıyorlar. (Ho, 2009)

Web günümüzün iş ilişkilerinin ayrılmaz bir parçasıdır. Şirketler ve enstitüler web'i işlerini yönetmek için; müşterilerse günlük işlemlerini gerçekleştirmek için kullanmaktadır. Ayrıca birçok kullanıcı kişisel ilgi alanları için de web'i kullanmaktadır. Bilindiği üzere web sayısız kaynaktan elde edilen muazzam bilgi içerir. Web kullanım madeni bilgiyi web sunucularının erişim kayıtlarından elde ettiği geçmişe ait verileri analiz ederek çıkarmaktadır. Bununla kullanıcı hareketlerini modellemek ve bu bilgilere dayanarak gelecekteki hareketlerini tahminlemek mümkündür. Sonuç olarak bu çıkarsaması yapılan bilgi, kullanıcı tercihleri doğrultusunda hazırlanacak olan web sayfalarının içeriklerinin kişiselleştirilmesinde kullanılır. Web kişiselleştirme ya da önerici sistemler bir çeşit “web kullanım madeni” uygulamalarıdır. (Baraglia and Silvestri, 2007).

Web üzerinde karmaşık işleri gerçekleştirmek günlük yaşantımızın önemli bir parçası haline geldi. Mobil servislerin keşfiyle web üzerinden kullanılan servislere yeni büyük bir alan daha eklenmiş oldu. Karmaşık servis önerilerinden küçük istemci cihazları üzerinde bile tam olarak faydalanabilmek ve her geçen gün artan uygulamaları yönetmek için kişiselleştirmenin güçlü kavramları gereklidir (Wagner, 2009).

Kablosuz internet erişimi gerekli olan bilgiye her yerden ve her zaman erişebilmeyi mümkün kılar. Ancak bununla birlikte küçük ekranları ve daha yavaş, daha pahalı ağ bağlantılarıyla mobil cihazlar yeni problemler doğurmaktadır. Mobil cihazlar ile kullanıcının herhangi bir liste üzerinde aradığı gerekli bilgiyi bulabilmesi bir hayli zor olmaktadır. İşte bu sebeple, kullanıcının lokasyonuna, o anki işine ve ilgi alanı doğrultusundaki profiline dayanarak bilgiyi kullanıcıya göre kişiselleştiren servisler, kablosuz internet kullanımını, potansiyelinin üzerine taşıyabilirler (Pazzani, 2009).

Geçen bir kaç yıl içerisinde standardizasyon çalışmalarında ve birbirleriyle etkileşime geçen web teknolojilerinin geliştirilmesinde büyük bir artış oldu. Devam eden standardizasyon çalışmaları bünyesinde ve endüstride son eğilim web üzerinden gerçek kişisel servislerin oluşturulması şeklinde olmaktadır. Uygulamalarda servis seçimi, kullanıcının bireysel tercihlerini içeren kişisel profiline ve bununla birlikte mobil cihaza ait teknik kısıtlamalara göre yapılır (Wagner, 2009).

4.1 Kişiselleştirme Yöntemleri

Yukarıda öneminin açıklandığı kişiselleştirme iki farklı şekilde gerçekleştirilebilir. Bunlar kullanıcı düzenlemelerine dayalı kişiselleştirme (Personalization via User Customization) ve uyarlanabilir kişiselleştirme (Adaptive Personalization). Web sayfaları kişisel ilgi alanlarına göre kişiselleştirilebilirler. Kişiselleştirme, açılan sayfalar, verilen siparişler gibi üstü kapalı verilere dayanan değişiklikler anlamına gelir. Özelleştirme (customization) terimi ise web sayfası üzerinden açıkça yapılan kategorilendirmeler ve tercihler için kullanılır (Choi and Han, 2008).

4.1.1 Kullanıcı düzenlemelerine dayalı kişiselleştirme

Kablosuz internet üzerindeki kişiselleştirme için en yaygın model olan MyYahoo gibi servislerin kullandığı, kullanıcıya ilgilendikleri konular için seçim yapmaya izin veren servislerdir. Bu servislerde, kullanıcı hesabı internet üzerinde yaratılır ve bu esnada kullanıcı ne gibi içeriğin kendisine gönderilmesini istediğine sunulan arayüzler aracılığıyla seçim yaparak karar verir (Pazzani, 2009).

Bu kişiselleştirme çeşitinde bazı problemler bulunmaktadır. İlk olarak, sunulan tercihlerin çok sayıda olmasından dolayı, kullanıcı cep telefonu gibi kablosuz cihaz üzerinde bu seçimlerin gösterimini ve düzenlemesini yapamaz. Hesabı web üzerinden açarak bu hesabı kablosuz cihazla ilişkilendirmenin karmaşıklığı bu çeşit kişiselleştirmenin kullanımını sınırlar. Bazı ülkelerde cep telefonu kullanıcılarının internet kullanıcılarının sayısından fazla olması bu tür servisler için marketi daraltmaktadır. İkinci olarak, kullanıcıya seçim yaptırılan seçenekler istenen bazı seçimleri kapsamayacak şekilde çok yüzeysel olabilir. Örneğin, kullanıcı spor karşılaşmalarının sonucundan haberdar olmak isteyebilir ama herhangi bir spor dalına ya da herhangi bir takıma ait sonuçlardan haberdar olmayı istemenin bir yolu yoktur. Bu problem spor alanına seçim yapılabilecek yeni kategorilerin eklemesiyle çözülebilir ama dünyadan haberler gibi daha geniş kapsamlı bir konuda standart konular belirlemeyi beklemek çok zordur. Üçüncü olarak da, bu tür profiller bilgiyi filtrelemede kullanılabilir ama önceliklendirme yapamaz (Pazzani, 2009).

Kablosuz interneti kullanan bazı servisler alternatif olarak anahtar kelime tabanlı sistemleri kullanırlar. Bu sistemler, kullanıcının almak istediği içeriği belirleyici anahtar kelimeleri kullanıcının profilinde tanımlamasına dayanır. Cep telefonlarındaki karakter girişindeki karmaşıklıktan dolayı bu servisler kullanıcının önce web üzerinde hesap açarak kendi profilinde almak istediği bilgiye ait anahtar kelimeleri buradan girmesini ister. Ve bu karmaşık veri girişini sadece çok az sayıda kullanıcı yapabilir (Pazzani, 2009).

4.1.2 Uyarlanabilir kişiselleştirme

Yukarıda açıklanan kullanıcı düzenlemelerine dayalı kişiselleştirme yönteminde kullanıcı profili oluşturmak için kullanıcıya büyük yük düşmektedir. Bunlara alternatif olarak, yapay zeka ya da istatistiksel bazı teknikler kullanarak kullanıcının ilgi alanları doğrultusunda otomatik bir şekilde profil oluşturulabilir. Uyarlanabilir (Adaptive) sistemler az sayıda örnekle öğrenebilir ve değişen kullanıcı isteklerine hızlı bir şekilde adapte olabilirler. Mobil cihazlar için kullanıcının ilgi alanlarının içsel olarak kullanıcının daha önceden yaptığı seçimlerden çıkartılması, kullanıcıdan kendi ilgi alanlarını tanımlamasını istemekten daha kullanışlı olabilir (Pazzani, 2009).

Kullanıcılara öneriler yapabilen profil filtreleme için iki ana yöntemden biri izlenebilir: Bunlar gruplayarak filtreleme ve içerik tabanlı filtreleme olarak ayrılır.

Gruplayarak filtreleme tüm kullanıcıların her hareketini izler ve benzer tercihleri olanları bulmaya çalışır. Gruplayarak filtreleme web üzerinden lokanta önerisi gibi lokasyon tabanlı servislerin kişiselleştirilmesi için uygundur. Yakındaki lokantalar için öneriler yapan bir uygulama ileri uyarlanabilir kişiselleştirme teknikleri kullanarak sadece en yakındaki lokantayı değil kişinin tercihlerine uygun o civardaki en iyi lokantayı önerecektir. Ancak bu yöntem haberler gibi gruplama yapılamayacak, değişen olaylar için iyi çalışmayacaktır. İçerik tabanlı öneriler kullanıcının tercih ve ilgi alanlarına göre istatistiksel bir profil yaratırlar. Bu yöntem haber, e-posta ve lokanta önerme sistemleri için uygundur (Pazzani, 2009).

İçsel geri beslemelerle çalışan uyarlanabilir kişiselleştirme yaklaşımlarının iki önemli avantajı vardır. Birincisi, kullanıcılar açısından kullanımı daha kolaydır. Kullanıcı sadece sistemi kullanırken profili otomatik olarak oluşur. Kullanıcı profilinde herhangi bir düzenleme yapmak istemedikçe uyarlanabilir bir sunucunun varlığından haberdar olmayacaktır. İkinci olarak, kullanıcıya gönderilebilecek ekstra içerik tabanlı ölçütlerin belirlenmesine izin verir. Kullanıcının istediği hizmetlerin yanı sıra önceki tercihlerinden yola çıkarak ek başka hizmetler de verilebilir (Pazzani, 2009).

Yukarıda açıklanan avantajların yanında uyarlanabilir kişiselleştirme yönteminin bazı dezavantajları da bulunmaktadır. Bir uygulama ile kullanıcı arasında özel ve interaktif bir iletişim kurmanın gerçek yolu, kullanıcıya farklı alternatifler sunmak ve ilgi alanına girenleri seçme şansı tanımak. Çünkü bir kullanıcı, o anki ihtiyaçlarını yine ancak kendisi bilir. Bir uygulamanın kişiselleştirilmesi, bilgisayarın, kullanıcının ihtiyaçlarını belirlemesi esasına dayanır. Bu yapılması güç bir iştir ve eğer karşınızdaki farklı zamanlarda farklı beğeni ve istekleri olan bir kullanıcı ise iş çok daha zordur. Örneğin bu yöntemle, bir kitap sitesinden daha önce aldığınız bir kitaba istinaden, tekrar o siteye girdiğinizde size bu konuya ilişkin diğer kitap alternatifleri sunulur. Aslında amacınız bu kez bambaşka bir kitap satın almak olabilir ama size illa ki daha önce aldığınız kitabın konusuna yakın kitaplar önerilir. Diğer alternatifleri görmek için de epey çaba sarfetmeniz gerekebilir. Bu şekilde kişiselleştirme hizmeti bazı uygulamalarda abartılabiliyor. Çünkü kullanıcının bir ziyaretinde istediği şey, diğer ziyaretinde bambaşka olabilir. Kişiselleştirme bu anlamda, internetin en büyük avantajı seçme özgürlüğü ile de çatışmış olur (Workcube, 2010).

Ayrıca, bir sitenin iyi bir şekilde kişiselleştirilebilmesi için kullanıcı hakkında çok sayıda bilgiye sahip olması gerekir. Bu hem özel hayatın gizliliği hem de aktif kullanıcı kavramıyla çelişkili bir durumdur. İnternet kullanıcısı bir siteyi açtığı anda oradaki bilgileri hemen inceleme veya kullanma eğilimindedir. Bir siteye girerken kendisi hakkında uzun uzun bilgiler soran pencerelerle karşılaşmak kimsenin hoşuna gitmez. Daha ilk kez ziyaret ettiği bir siteye kendisi hakkında pek çok bilgi vermek de kullanıcı için pek hoş bir durum değildir (Workcube, 2010).

Tüm bu yukarıda kişiselleştirmeye dair verilen avantaj ve dezavantajlar göz önüne alındığında kişiselleştirmenin çok dikkatli bir şekilde yapılması gerektiği ortaya çıkıyor. Kullanımı kolaylaştırmak için yapılan bazı düzenlemeler bazen kullanımı olduğundan da daha karmaşık ve zor hale getirebiliyorlar.

4.2 Kişiselleştirmenin Gerçekleştirimi

Bu bölümde yukarıda açıklanan kişiselleştirmenin uygulamalarda nasıl kullanılacağı ve gerçekleştirileceği açıklanmaktadır.

Bir sitenin, üyelerinin isteklerine göre davranış göstermesi için kullanılan çeşitli teknikler vardır. Bunlardan en basit ve ilkel olanı çerez (cookie) kullanımıdır. Çerezler istemci taraflı bilgi saklama amacıyla tercih edilirler. Ancak istemcilerin çerez desteği olmadığı durumlarda kişiselleştirme mümkün olmayacaktır. Alternatif ve daha sağlıklı bir yol olarak kullanıcıya ilişkin bilgilerin kişiselleştirme amacıyla veritabanı sistemlerinde tutulması ve sayfalar arası geçişlerde de oturum (session) nesneleri yardımıyla taşınması çok daha profesyonel bir yaklaşımdır (Şenyurt, 2009).

Asp.Net 2.0 ile gelen doğrulama(Authentication), Rol Yönetimi (Roles Management), profil (Profile) hizmetleri uygulamaların çok daha kolay bir şekilde kişiselleştirilmelerini sağlamaktadır. Özellikle web tabanlı uygulamalarda istemcilerin doğrulanması, rollerine göre ne yapabileceklerinin belirlenmesi, profillerine göre istekte bulundukları sayfaların kişiselleştirilmesi önemlidir. Asp.Net 2.0 Web Site Administration Tool veya kod tarafında Membership API içerisinde yer alan tipler yardımıyla, kullanıcı hesaplarının yönetilmesi ve çeşitli rollere atanması görsel olarak kolayca yapılabilir. Bununla birlikte Profile API içerisinde yer alan tipler yardımıyla da, bir web sitesinin kullanıcıya göre özelleştirilebilmesi son derece kolaylaşmıştır (Şenyurt, 2009).

ASP.NET 2.0 yeni özelliklerini destekleyebilmek için kendisine bir çok yeni konfigürasyon bölümü eklemiştir. Aşağıda çizelge 4.1'deki tabloda bu konfigürasyonların en önemlileri listelenmektedir (Thangarathinam, 2006):

Çizelge 4. 1 ASP.NET 2.0 ile gelen yeni konfigürasyonlar (Thangarathinam, 2006)

Konfigürasyon	Açıklama
<anonymousIdentification>	Anonim kullanıcıya atanan içsel kimlik numarası belirleme mekanizmasını düzenler.
<connectionStrings>	Uygulama tarafından kullanılan bağlantı komutlarını bildirir. Kod tarafından referans vermek için kullanılan her bir bağlantı komutu tek bir isimle tanımlanır.
<healthMonitoring>	Çalışan uygulamaların performanslarını izlemek için tasarlanan yöntemler kümesini sağlayan sağlık izleme kütüphanelerinin düzenlenmesine izin verir.
<membership>	Üyelik bilgilerini getiren ve bunları kaydetmeyi sağlayan sistemin düzenlenmesine izin verir.
<profile>	Kullanıcı profilinin nasıl saklanacağını düzenler. Bu bölüm kullanıcı profilini gösteren sınıfın şemasının tanımlanmasına izin verir.
<roleManager>	Mevcut kullanıcı için rol bilgisinin nasıl saklanacağı ile ilgili düzenlemelere izin verir.
<siteMap>	Site planının saklanması için desteklenen sağlayıcıların kaydedilmesine izin verir
<urlMappings>	Sanal URL'lerin fiziksel URL'lere eşleştirilmesine izin verir.
<webParts>	Web Part'lar için gerekli ayarların düzenlenmesine izin verir

Asp.Net 2.0 ile gelen Profile API yardımıyla, kullanıcıya dayalı kişiselleştirilmiş ayarlar MemberShip sistemine dahil edilebilir. Böylece her üye için farklı kişisel bilgiler tutulabilir ve sayfanın o kullanıcının tercihleri doğrultusunda değiştirilmesi dinamik olarak sağlanabilir. Kullanıcı için geçerli

olan bu kişisel bilgiler veritabanında üyelik sistemine ait tablolarda saklanmalıdır. Profile sisteminde veritabanı tarafında Sql Server 7, 2000 ve 2005 sistemleri ya da Custom Provider'lar kullanılabilirler. Profile API üye ile ilgili verileri sunucu sistem üzerinde tutmaktadır.

Asp.Net 2.0'da Membership API'nin kullanılması için Web.Config dosyasına eklenen "membership" etiketi altındaki "providers" etiketi içerisinde çizelge 4.2'de görüldüğü üzere gerekli tanımlamaların yapılması gerekmektedir.

Çizelge 4. 2 Membership API kullanımı

```
<membership defaultProvider="SqlProvider" userIsOnlineTimeWindow="15">
  <providers>
    <add name="SqlProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="profiller"
  </providers>
</membership>
```

Çizelge 4.2'de gösterilen Web.Config dosyasında yapılan değişiklikten sonra doğrulama işlemi kod içerisinde çizelge 4.3'de gösterildiği şekilde kullanılabilir. Kod içerisinde membership sınıfı üzerinden doğrulamanın kullanımı sağlanır.

Çizelge 4. 3 Membership API ile doğrulama işlemi

```
if (Membership.ValidateUser(Login1.UserName, Login1.Password))
{
    if (Request.QueryString["ReturnUrl"] != null)
    {
        FormsAuthentication.RedirectFromLoginPage(Login1.UserName, false);
    }
    else
    {
        FormsAuthentication.SetAuthCookie(Login1.UserName, false);
        Response.Redirect("UsersPage.aspx");
    }
}
else
{
    //Başarısız giriş. Bilgilerinizi kontrol edip tekrar deneyin.
    if (Membership.GetUser(Login1.UserName) != null)
    {
        Login1.FailureText = "Your login attempt was not successful. Please try again.";
        if (Membership.GetUser(Login1.UserName).IsLockedOut == true)
        {
            Login1.FailureText = Login1.UserName + " account is locked.Contact the administrator.";
        }
    }
}
```

Profile API kullanılırken, kişiselleştirilecek bilgiler birer özellik olarak Web.Config dosyasına eklenirler. Asp.Net 2.0'da tüm kullanıcılara ait ortak profil bilgileri Web.Config dosyasına eklenen Profile etiketi altında tanımlanır. Profile etiketi içerisindeki Properties etiketi altında ise herhangi bir üye için kişisel

ayarların tutulduğu özellik tanımlamaları yer alır. Çizelge 4.4’de Web.config dosyası içerisine dahil edilmiş örnek Profile özelliklerini görebilirsiniz.

Çizelge 4. 4 Web.config dosyası içerisine dahil edilmiş örnek Profile özellikleri

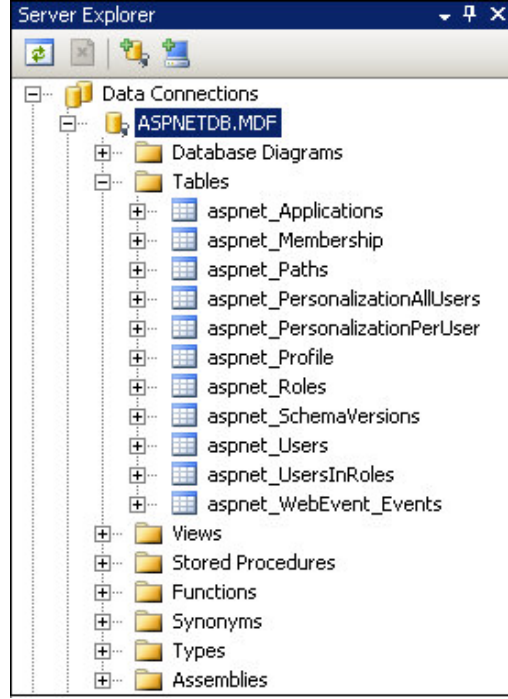
```
<profile>
  <properties>
    <add name="MyTabs" />
    <add name="MyTabIds" />
    <add name="MaxTabId" />
    <add name="isAdmin" type="boolean" />
    <add name="Activity" type="System.String" />
    <add name="Department" type="System.String" />
    <add name="Interval" type="System.String" />
    <add name="MyTags" type="System.String" />
  </properties>
</profile>
```

Çizelge 4.4’deki profil bilgisine göre, sisteme giriş yapan her kullanıcı için MyTabs, MyTabIds, MaxTabId, isAdmin vb gibi özelliklerde kişinin tercihlerine göre farklı değerler tutulabilmektedir. Profil içerisinde belirtilen özelliklerin kullanılabilmesi için sitenin kullanıcıya özel çalışacak şekilde tasarlanması gerekir. Profil için saklanacak özellikler konfigürasyon dosyasına eklendikten sonra kod içerisinde Profile sınıfı üzerinden çeşitli işlemler gerçekleştirilebilir. Web.Config dosyasında, Profile etiketi içerisindeki Properties etiketi altına eklenen her bir elemana Profile sınıfı üzerinden erişilebilir. Buna ilişkin kod çizelge 4.5’de gösterilmektedir.

Çizelge 4. 5 Profile sınıfının kod içinde kullanımı

```
//Profil içerisinde MyTabs belirtilmediğinde, "Home Page"
// isimli bir sekme varsayılan olarak eklenir.
if (Profile.MyTabs == "")
{
    Profile.MyTabs = "Home Page;";
}
```

Kişiselleştirme için kullanılacak veritabanlarını yaratmak ve çalışma mantığını oluşturmak Asp.Net 2.0’da otomatik olarak yapılmaktadır. Şekil 4.1’de Asp.Net 2.0’ın kişiselleştirme için oluşturduğu veritabanı yapısı ve tabloları görülmektedir.

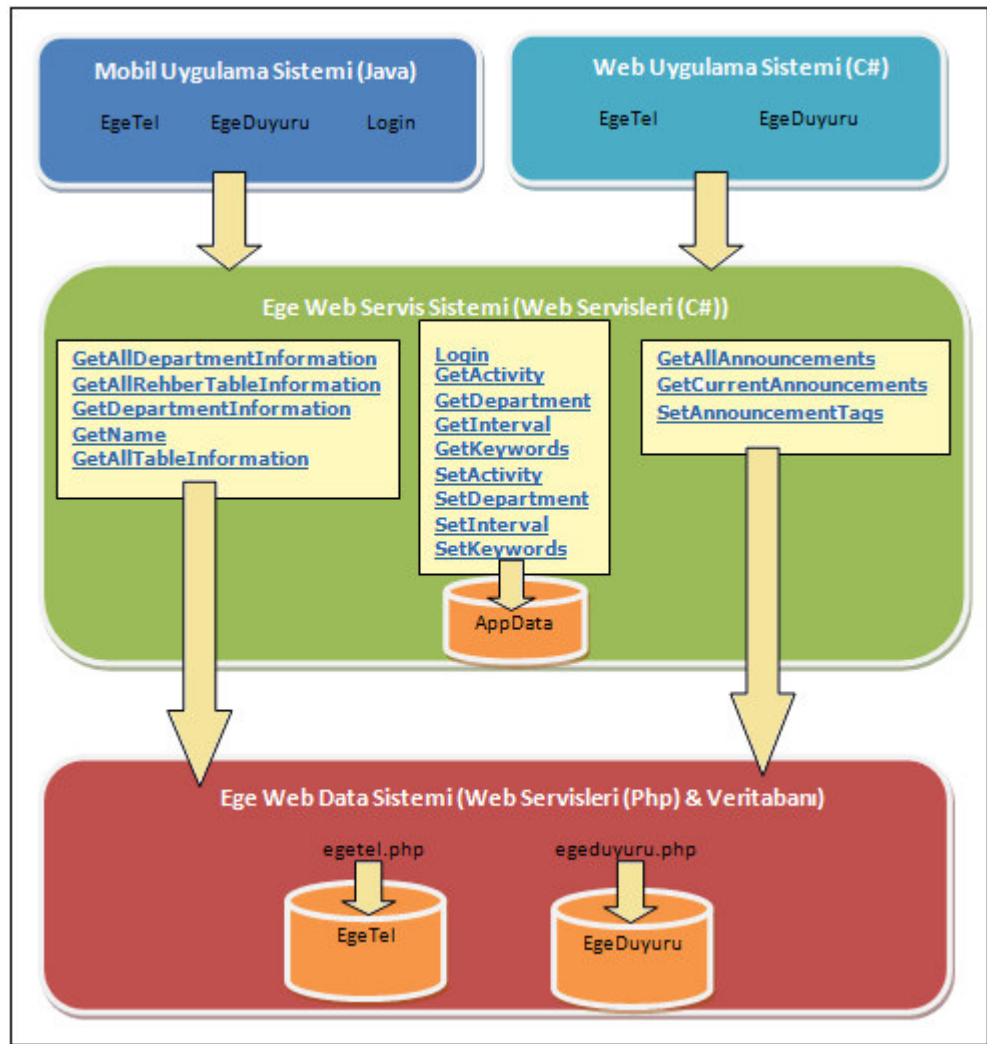


Şekil 4. 1 Asp.Net 2.0'ın kişiselleştirme için oluşturduğu veritabanı yapısı ve tabloları

5. UYGULAMANIN GERÇEKLEŞTİRİMİ

Tez çalışması kapsamında önceki bölümlerde anlatılan teknolojiler kullanılarak Ege Üniversitesi kurumsal web sitesi üzerinde yer alan ‘Ege telefon rehberi’ ve ‘Ege duyuru’ hizmetlerine mobil cihazlar ile erişim sağlanmış, ‘Ege duyuru’ hizmeti kullanıcıya özel hizmet verecek şekilde kişiselleştirilerek geliştirilmiştir.

Gerçekleştirilen sistem, farklı alt sistemlerden oluşan katmanlı bir yapıya sahiptir. Şekil 5.1’de sistemin genel mimarisi ve veri akışı gösterilmektedir. Sistem, şekil 5.1’de görüldüğü üzere üç katmanlı bir yapıda tasarlanmıştır.



Şekil 5. 1 Sistemin genel mimarisi

En üst katmanda uygulamanın mobil cihazlar üzerinde çalışmasını sağlayan “Mobil Uygulama Sistemi” ve uygulamanın web üzerinde çalışmasını sağlayan

“Web Uygulama Sistemi” bulunmaktadır. Bu katmandaki her iki sistemde de kullanıcı etkileşimleri gerçekleştirilerek kullanıcının istediği hizmetlerin bir alt katman ile veri alışverişi ile elde edilip kullanıcıya sunumu sağlanmaktadır.

Ege Web Servis Sisteminin bulunduğu orta katman, en üst katman ile en alt katman arasında bilgi alışverişini sağlayan C# web servislerinden ve ASP .NET 2.0’ın kişiselleştirme üzerine sağladığı destekle birlikte kullanıcıya ait verileri saklayan veritabanından oluşmaktadır.

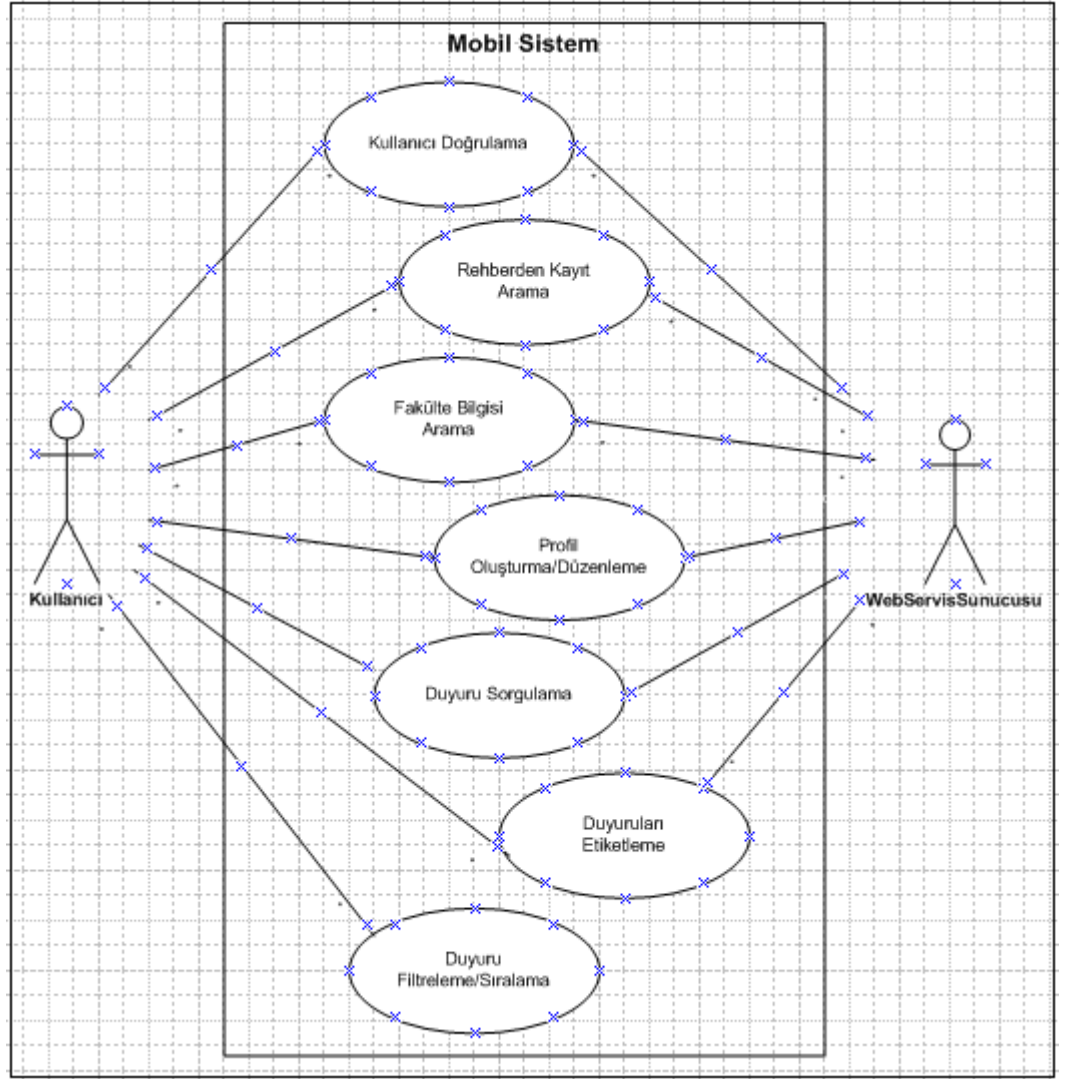
Ege Web Data Sisteminin bulunduğu en alt katman ise orta katmandaki web servislerinin iletişim kurduğu php web servislerinden ve bu web servislerinin veri alışverişini sağlayan Ege Üniversitesi kurumsal web sitesi üzerinden de erişilen ‘egetel’ ve ‘db_egeduyuru’ isimli veritabanlarından oluşmaktadır.

Aşağıdaki bölümlerde bu sistemler ve çalışma mekanizmaları detaylı olarak açıklanmaktadır.

5.1 Mobil Uygulama Sistemi

Mimarının en üst katmanı olan uygulama katmanında yer alan Mobil Uygulama Sistemi uygulamanın mobil cihazlar üzerinde çalışmasını sağlar. Burada mobil cihaz ile kullanıcı arasındaki etkileşimler gerçekleştirilir. Sistem, kullanıcının arayüzler ile istediği hizmetin seçimini sağlar ve bu seçilen hizmetlere göre orta katmandaki Ege Web Servis Sistemindeki ilgili web servislerini kullanarak ve gerekli veri alışverişi yaparak kullanıcının istediği hizmetin sunumunu gerçekleştirir.

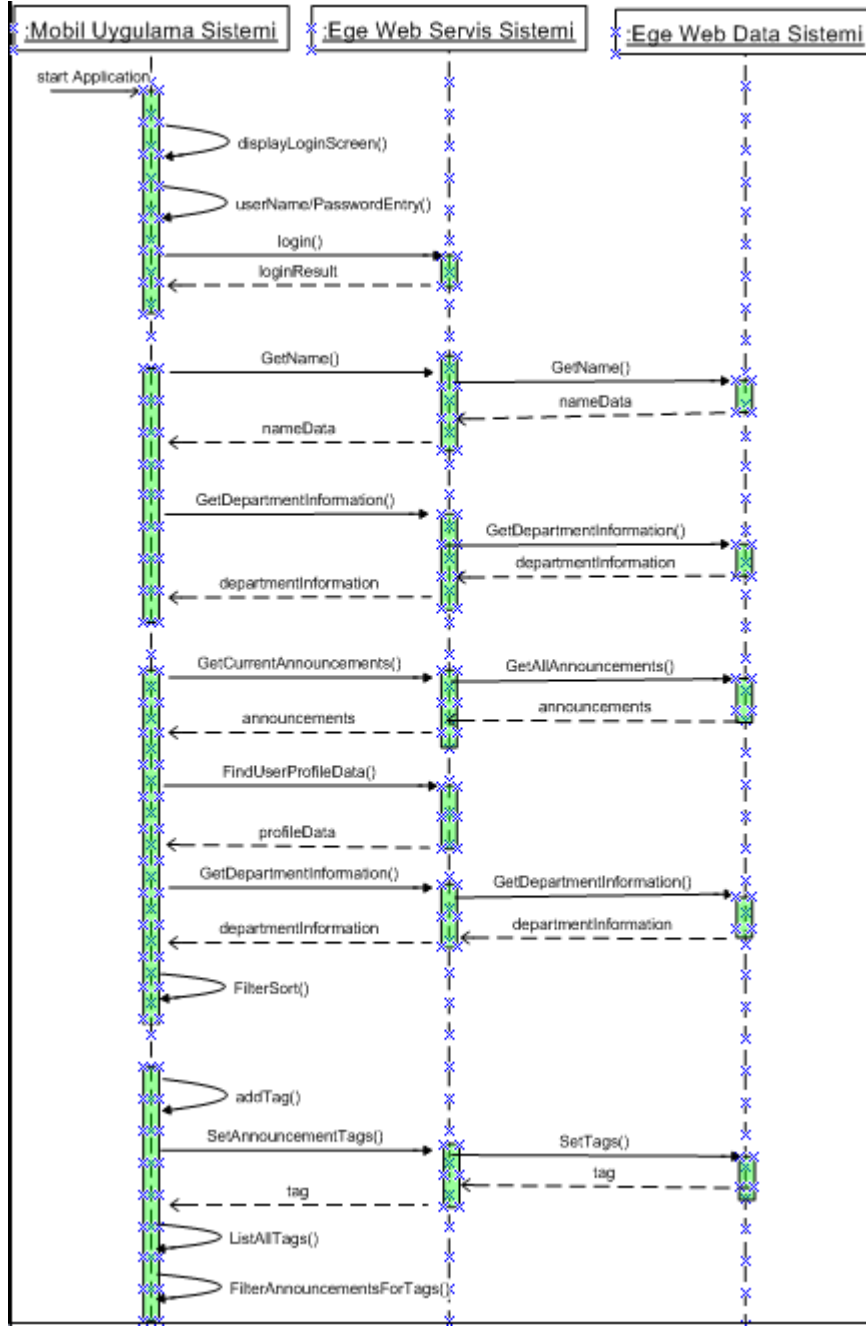
Mobil uygulama sistemi, J2ME teknolojisi kullanılarak, Netbeans IDE üzerinde gerçekleştirilmiştir. Uygulama, Java destekli mobil cihazlarda çalışacak şekilde tasarlanmıştır.



Şekil 5. 2 Mobil uygulama sisteminin kullanım durum (use-case) diyagramı

Mobil uygulama sistemine ait kullanım durum (use-case) diyagramı şekil 5.2’de gösterilmiştir. Mobil uygulama sisteminde gerçekleştirilen hizmetler kullanıcı doğrulama, rehberden kayıt arama, fakülte bilgisi arama, profil bilgisi oluşturma ve düzenleme, duyuruları sorgulama, duyuruları etiketleme, duyuruları filtreleme ve sıralamadır.

Uygulamadaki sistemlerin iş paketleri için etkileşimlerinin gösterildiği sıra (sequence) diyagramı şekil 5.3’de verilmiştir.



Şekil 5.3 Uygulamanın sıra (sequence) diyagramındaki gösterimi

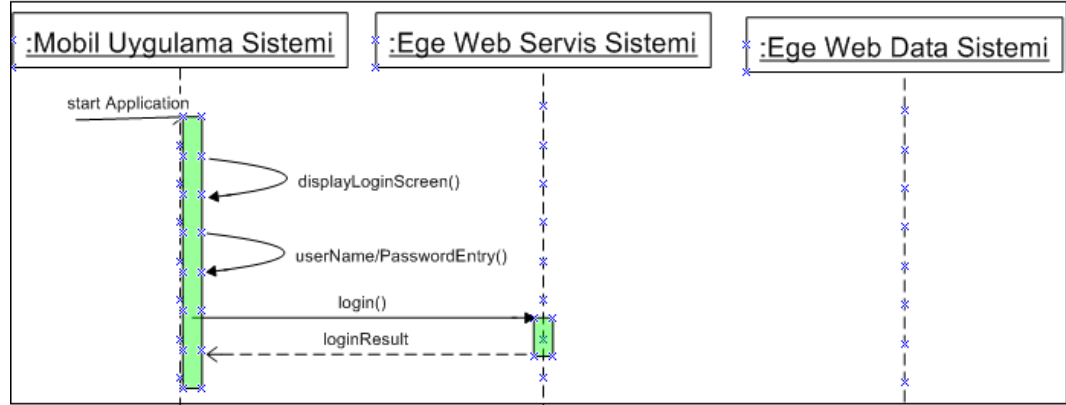
Şekil 5.3'deki sıra diyagramı ilerleyen bölümlerde tek tek detaylı olarak açıklanmaktadır.

5.1.1 Sisteme giriş

Mobil sistemde yer alan uygulama ilk olarak sisteme giriş yaparak başlatılır. Uygulama ilk başlatıldığında ekrana kullanıcının girmesi için kullanıcı adı ve şifre sorgusu getirilir. Kullanıcının sisteme giriş yapabilmesi için öncelikle web

arayüzü ile sisteme üyeliğini gerçekleştirmiş olması gerekmektedir. Web arayüzünde üyelik tanımlama aşamasında kullanıcının belirlediği kullanıcı adı ve şifresi kullanılarak mobil sisteme giriş yapılır.

Şekil 5.4’de sisteme giriş senaryosuna ilişkin sıra diyagramı gösterilmektedir.



Şekil 5. 4 Sisteme giriş senaryosu sıra diyagramı

Şekil 5.4’de gösterildiği üzere sisteme giriş senaryosu orta katmandaki Ege Web Servis Sistemi ile üst katmandaki Mobil Uygulama Sistemi arasında geçmektedir, en alt katmandaki Ege Web Data Sistemi ile bir etkileşimde bulunulmamaktadır. Şekil 5.4’de de görüleceği üzere uygulamanın başlatılmasıyla sisteme giriş senaryosu başlar. Ardından mobil uygulama kullanıcı adı ve şifrenin girileceği menüyü ekrana getirir. Daha sonra kullanıcı mobil uygulama üzerinden kullanıcı adı ve şifreyi girer. En son olarak girilen kullanıcı adı ve şifre, doğruluğu kontrol edilmek üzere orta katmandaki Ege Web Servis Sistemine gönderilir. Sonuç, uygulamada değerlendirilmesi için orta katmandaki Ege Web Servis sisteminden üst katmandaki Mobil Uygulama Sistemine gönderilir.

Sisteme giriş senaryosu aşağıdaki gibidir:

1. Kullanıcı mobil cihazından uygulamayı açar.
2. Mobil uygulama sistemi kullanıcı giriş ekranını çıkarır.
3. Kullanıcı, kullanıcı adı ve şifresini girer.
4. Mobil uygulama sistemi, girilen kullanıcı adı ve şifresini sunucu sisteme (Ege Web Servis Sistemine) gönderir.

5. Sunucu sistem kullanıcıyı sorgular ve sonucu mobil uygulama sistemine gönderir.
- 6.a. Kullanıcı adı ve şifresi doğruysa mobil uygulama sistemine giriş tamamlanmıştır, ana menu ekrana gelir.
- 6.b. Kullanıcı adı ya da şifresi yanlışsa uyarı mesajı ekrana getirilir.

Sisteme giriş senaryosundaki girilen kullanıcı adı ve şifrenin sorgulanması Ege Web Servis sistemindeki web servisi üzerinden kullanıcı adı ve şifre parametreleriyle birlikte 'login' adlı web metodun çağrımıyla gerçekleştirilir. Kullanıcı adı ve şifre sorgusu için web servisinin çağrımına ilişkin kod, çizelge 5.1'de gösterilmiştir.

Çizelge 5. 1 Kullanıcı adı ve şifre sorgusu için web servisinin çağrımı

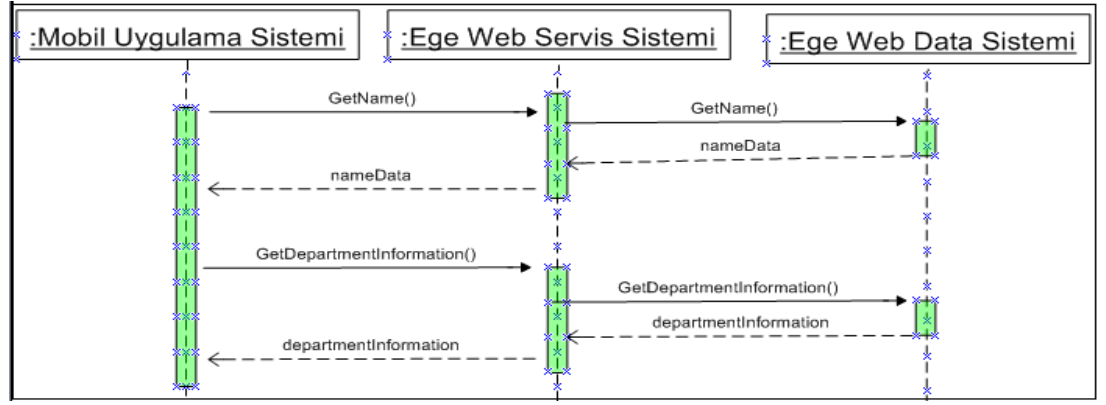
```
boolean login_result = false;
EgeWebService.Service1_Stub s2 = new EgeWebService.Service1_Stub();
ticker1.setString("Lütfen Bekleyiniz...");
try {
    login_result = s2.login(loginScreen.getUsername(), loginScreen.getPassword());
} catch (Exception ex) {
    ex.toString();
}
```

Sisteme giriş aşamasında girilen kullanıcı adı ve şifre doğruysa sisteme başarılı bir şekilde giriş yapılmıştır, ana menü ekrana getirilir. Kullanıcı adı ya da şifre yanlışsa, ekrana bir uyarı mesajı çıkartılarak kullanıcı adı ve şifrenin tekrar girilmesi istenir.

5.1.2 Rehberden kayıt arama

Sisteme başarılı bir şekilde giriş yapılmasının ardından Ege Üniversitesi Telefon Rehberinden kayıt aramak için ana menüden 'Ege Tel' seçilir. Bu seçim sonrası ekrana gelen arayüzde aramak istenilen kişiye ait isim ve soyisim bilgileri girilerek arama yapılır.

Şekil 5.5'de rehberden kayıt arama senaryosuna ilişkin sıra diyagramı gösterilmektedir.



Şekil 5. 5 Rehberden kayıt arama senaryosu sıra diyagramı

Rehberden kayıt arama senaryosunda şekil 5.5’de gösterildiği üzere ilk olarak aranılan kişiye ait kişisel bilgiler, orta katmandaki Ege Web Servis Sistemi üzerinden alt katmandaki Ege Web Data Sisteminde sorgulanır. İkinci adımda ise ilk sorgu sonucu dönen kişiye ait fakülte kodu kullanılarak yine orta katmandaki Ege Web Servis Sistemi üzerinden alt katmandaki Ege Web Data Sisteminde aranılan kişinin fakülte bilgileri sorgulanır.

Rehberden kayıt arama senaryosu aşağıdaki gibidir:

- 1.Kullanıcı mobil uygulama sistemi üzerindeki arayüzden aramak istediği kişinin ismini ve/veya soyismini girer.
- 2.Mobil uygulama sistemi girilen ismi sunucu sisteme (Ege Web Servis Sistemine) gönderir.
- 3.Sunucu sistem sorgunun sonucunu mobil sisteme gönderir.
- 4.a.Sorgu sonucu uygun kayıt dönmüşse Mobil uygulama sistemi ilgili kişiye ait fakülte bilgilerini sorgulamak üzere sunucu sisteme gerekli verileri gönderir.
- 4.b.Sorgu sonucu uygun kayıt dönmemişse Mobil uygulama sistemi kayıt bulunamadığını kullanıcıya bildirir ve süreç tamamlanır.
- 5.Sunucu sistem fakülte sorgusunun sonucunu mobil uygulama sistemine gönderir.
- 6.Mobil uygulama sistemi bulunan kayıt ya da kayıtları kullanıcıya bildirir ve süreç tamamlanır.

Rehberden kayıt arama senaryosundaki girilen isim ve/veya soyismin sorgulanması Ege Web Servis sistemindeki web servisi üzerinden isim ve/veya soyisim parametresiyle birlikte ‘getName’ adlı web metodun çağrımıyla gerçekleştirilir. Rehberden kayıt arama sorgusu için web servisin çağrımına ilişkin kod, çizelge 5.2’de gösterilmiştir.

Çizelge 5. 2 Rehberden kayıt aramak için web servisinin çağrımı

```
EgeWebService.Service1_Stub s = new Service1_Stub();
try {
    return_value = s.getName(name);
} catch (Exception ex) {
    ex.toString();
}
```

Rehberden kayıt aramak için girilen isim ve/veya soyisme ait herhangi bir kayıt bulunamamışsa ekrana uyarı mesajı çıkarılarak kaydın bulunamadığı kullanıcıya bildirilir.

Rehberden kayıt aramak için girilen isim ve/veya soyisme ait herhangi bir kayıt bulunmuşsa, kaydı bulunan kişinin fakülte bilgileri sorgulanır. Fakülte bilgilerinin sorgulanmasında ‘getName’ adlı web metodun çağrılmasıyla gerçekleştirilen isim sorgusundan dönen bilgiler parametre olarak kullanılır. Fakülte bilgilerinin sorgusu, Ege Web Servis sistemindeki web servisi üzerinden kod parametreleriyle birlikte ‘getDepartmentInformation’ adlı web metodun çağrımıyla gerçekleştirilir. Fakülte bilgilerinin getirilmesi için web servisinin çağrımına ilişkin kod, çizelge 5.3 ‘de gösterilmiştir.

Çizelge 5. 3 Fakülte bilgilerinin getirilmesi için web servisinin çağrımı

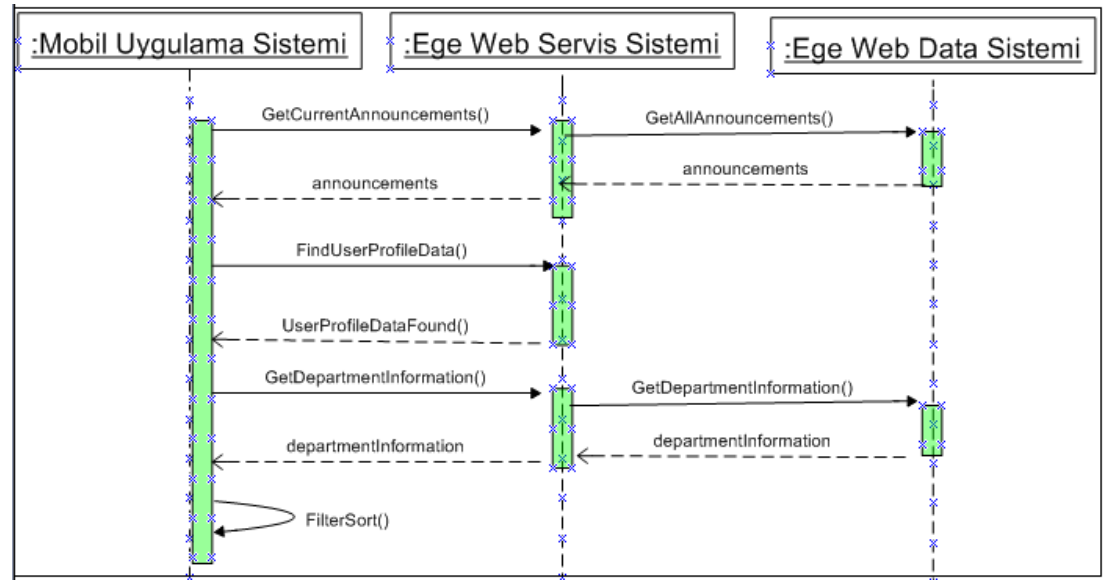
```
try {
    dept_value = s.getDepartmentInformation(kkodu, bkodu, birimkod1, birimkod2, birimkod3);
} catch (Exception ex) {
    ex.toString();
}
```

Yukarıda detaylı olarak anlatılan sorgulardan dönen kayıt ya da kayıtlar aranan kişiye ait fakülte, bölüm, e-posta adresi, telefon gibi bilgiler kullanıcıya gösterilmek üzere ekranda listelenir.

5.1.3 Duyuruların listelenmesi

Sisteme başarılı bir şekilde giriş yapılmasının ardından Ege Üniversitesi duyurularının listelenmesi için ana menüden ‘Ege Duyuru’ seçilir. Bu seçim sonrası duyurular, kullanıcının profilinde saklanan seçimlere göre filtrelenerek ve yine kullanıcının profilinde saklanan önceliklendirmelere göre sıralanarak listelenir.

Şekil 5.6’da duyuruları listeleme senaryosuna ilişkin sıra diyagramı gösterilmektedir.



Şekil 5. 6 Duyuruları listeleme senaryosu sıra diyagramı

Şekil 5.6’daki sıra diyagramında gösterildiği üzere duyuruların listelenmesi için ilk olarak en alt katman olan Ege Web Data katmanından duyurular önce orta katmana, sonra da üst katmana getirilir. İkinci adım olarak gerekli kişiselleştirmeleri yapabilmek için duyuruların filtrelenmesi ve önceliklendirilmesinde kullanılmak üzere orta katmanda tutulan kullanıcıya ait profil bilgileri elde edilir. Üçüncü adımda yine duyuruların kişiselleştirilmesinde üstü kapalı önceliklendirme olarak kullanılması için kullanıcının fakülte bilgileri orta katman üzerinden en alt katman olan Ege Web Data katmanından sorgulanır. Son olarak yukarıda elde edilen tüm bu profil bilgileri ve fakülte bilgileri ile duyurular kişiye özel kişinin tercihleri doğrultusunda filtrelenir ve önceliklere göre sıralanır.

Filtreleme ve sıralama işlemleri şekil 5.6’da da görüldüğü üzere en üst katmanda yani mobil uygulama sisteminde yer almaktadır. Sistemin daha hızlı çalışmasını sağlayacağı düşünülerek filtreleme ve sıralama işlemlerinin gerçekleştirimleri orta katmana alınmış ve uygulamanın bu değişiklik sonucunda elde edilen performansı hız açısından değerlendirilmiştir. Filtreleme ve sıralama işlemleri üst katmanda gerçekleştirildiğinde tüm duyurular sistemin ilk açılışında bir kere bu katmana alınır, ayrıştırıldıktan sonra bir vektörde tutularak tüm filtreleme sıralama işlemleri bundan sonra bu vektör üzerinden gerçekleştirilir. Filtreleme ve sıralama işlemleri orta katmana alındığında bu işlemler tek başlarına mobil uygulamadaki hızlarına göre daha hızlı çalışmışlardır. Ancak buna rağmen her bir filtreleme ve sıralama sonrası filtrelenen ve sıralanan duyuruların orta katmandan üst katmana iletilmesi ve üst katmanda duyuruların ayrıştırılma işlemleri ek zaman harcamaktadır. Bu sebeple toplamda sistemin çalışma hızı performansı açısından filtreleme ve sıralama işlemlerinin orta katmana taşınmasının bir iyileşme sağlamadığı görülmüş ve bu işlemlerin gerçekleştirimleri tekrar üst katmana alınmıştır.

Duyuruların listelenmesi senaryosu aşağıdaki gibidir:

- 1.Kullanıcı mobil uygulama sistemi üzerindeki arayüzden ‘Ege Duyuru’yu seçerek duyuru uygulamasını başlatır.
- 2.Mobil uygulama sistemi duyuruların getirilmesi isteğini sunucu sisteme (Ege Web Servis Sistemine) gönderir.
- 3.Sunucu sistem mobil sisteme tüm duyuruları döndürür.
- 4.Mobil sistem sunucu sistemden kullanıcıya ait kayıtlı profil bilgisini ister.
- 5.Sunucu sistem mobil sisteme kullanıcıya ait profil bilgisini döndürür.
- 6.Mobil sistem sunucu sistemden kullanıcıya ait fakülte-bölüm bilgisini ister.
- 7.Sunucu sistem mobil sisteme kullanıcıya ait fakülte-bölüm bilgisi mevcutsa döndürür.
- 8.Mobil sistem gelen profil bilgisine göre duyuruları filtreler, profil bilgisindeki anahtar kelimelere ve mevcutsa fakülte-bölüm bilgisine göre önceliklendirme yapar. Ve duyuruları kullanıcıya gösterir.

Duyuruların listelenmesi için Ege Üniversitesindeki duyuruların sorgulanması Ege Web Servis sistemindeki web servisi üzerinden ‘getCurrentAnnouncements’ adlı web metodun çağrımıyla gerçekleştirilir.

Duyuruların listelenmesi için web servisinin çağrımına ilişkin kod, çizelge 5.4’de gösterilmiştir.

Çizelge 5. 4 Duyuruların listelenmesi için web servisinin çağrımı

```
EgeWebService.Service1_Stub s = new Service1_Stub();
return_value = s.GetCurrentAnnouncements(index_str);
```

Çizelge 5.4’de gösterilen ‘GetCurrentAnnouncements’ adlı web metodun geri dönüş değeri olarak gelen string tipindeki veri ayrıştırılarak duyuru detayları elde edilir. Her bir duyuru için yeni bir duyuru nesnesi oluşturulur ve bu nesne, duyuru kayıtlarının tutulduğu vektöre eklenir. Duyurulara yapılacak daha sonraki erişimlerde bu vektör kullanılır.

Sisteme giriş aşamasında web servisinden elde edilen profil bilgileriyle oluşturulan kullanıcı nesnesindeki kullanıcının kayıtlı seçimleri, duyuruların kişiselleştirilerek listelenmesinde filtreleme ve sıralama yapmak için kullanılır. Duyurular kullanıcının tercihleri doğrultusunda akıllı bir şekilde görünüm kazanır.

Burada tüm duyurular taranarak mevcut listedeki tüm etkinlik çeşitleri ve duyurunun ait olduğu fakülte/bölüm bilgileri belirlenir. Etkinliklere, duyurunun ait olduğu fakülte/bölgümlere ve zaman aralığına göre filtreleme seçenekleri tanınır. Ayrıca kullanıcının profil bilgisinden gelen, sisteme daha önceki girişlerinde belirlediği anahtar kelimelere göre duyurular önceliklendirilir. Böylece kullanıcı düzenlemelerine dayalı olarak duyuruların sunumu kişiselleştirilmiş olur.

Anahtar kelimelere ek olarak bir de kullanıcının bağlı olduğu fakülte-bölüm bilgisine göre içsel bir önceliklendirme gerçekleştirilir. Kullanıcı Ege Üniversitesi personeli ise bağlı olduğu fakülte ve bölüme ait duyuruları listenin en başında görmek isteyeceği varsayımından yola çıkılarak, kullanıcının bağlı olduğu fakülte ve bölüme ait herhangi bir duyuru mevcutsa, duyuru listesinde bu duyurular tarihleri ne olursa olsun, listenin en başında yer alacaklardır. Burada uyarlanabilir kişiselleştirme tekniği kullanılarak kullanıcının bilgisi dışında sistemin kullanıcıya özel tanımladığı, üstü kapalı bir önceliklendirme gerçekleştirilir.

Kullanıcı profilinde herhangi bir anahtar kelime yoksa ve kullanıcıya ait fakülte-bölüm bilgisi bulunamamışsa duyurular tarihe göre sıralanır. Kullanıcı

profilinde herhangi bir anahtar kelime veya kullanıcıya ait fakülte-bölüm bilgisi bulunmuşsa, bu anahtar kelimeleri içeren duyurular veya kullanıcıya ait fakülte-bölüm bilgisiyle duyuruya ait fakülte-bölüm bilgisi aynı olan duyurular listenin en başında olacak şekilde duyurular sıralanarak listelenir.

Duyuruların listelenmesinde profil bilgisi doğrultusundaki filtreleme sonucunda herhangi bir duyuru bulunamamışsa ekrana uyarı mesajı çıkarılarak uygun duyuruların bulunamadığı kullanıcıya bildirilir.

5.1.4 Profil Oluşturma

Kullanıcının sisteme ilk girişinde varsayılan değerlerle bir kullanıcı profili, kullanıcı adına sistem tarafından oluşturulur. Böylelikle kullanıcı henüz kendine özgü profil tanımlamasını yapmadan da sistemdeki tüm duyuruları listeleyebilir. Bu varsayılan değerler, kullanıcının tüm duyuruları listeleyebilmesi için sistemdeki tüm etkinlik türleri ve tüm gönderici birimler seçilmiş; tarih aralığı bir ay ve anahtar kelime sahası boş olacak şekilde düzenlenmiştir.

5.1.5 Profil Sorgulama

Kullanıcının sisteme girişi ile birlikte, kullanıcı profilinin belirlenmesi için kullanıcının bir önceki sisteme girişinde yaptığı seçimler sorgulanır. Kullanıcıya ait profil bilgileri orta katmandaki Ege Web Servis sistemindeki veritabanında saklanmaktadır. Bu profil bilgilerine erişmek için profil sorgulama Ege Web Servis sistemindeki web servisi kullanılarak yapılır.

Çizelge 5.5’de profil sorgusu için profil bilgilerinin web servisinden getirilmesine ilişkin kod parçası görülmektedir.

Çizelge 5. 5 Profil bilgilerinin web servisinden getirilmesi

```

EgeWebService.Service1_Stub s = new Service1_Stub();
try {
    activities = s.getActivity(getTxtUserName().getString());
    departments = s.getDepartment(getTxtUserName().getString());
    keywords = s.getKeywords(getTxtUserName().getString());
    user_department = s.getUserDepartment(getTxtUserName().getString());
    interval = Integer.valueOf(s.getInterval(getTxtUserName().getString()).intValue());
} catch (Exception ex) {
    ex.toString();
}

```

Çizelge 5.5’de de görüldüğü üzere Ege Web Servis sistemindeki web servisi üzerinden kullanıcının bir önceki tercihleri; seçtiği etkinlik türleri ‘getActivity’, seçtiği birimler ‘getDepartment’, seçtiği tarih aralığı ‘getInterval’ ve girmiş olduğu anahtar kelimeler ise ‘getKeywords’ web metotları ile elde edilir.

Yukarıda gösterildiği şekilde elde edilen kullanıcı profil bilgilerini tutmak ve programın gerekli yerlerinde kullanmak, duyuru listesi üzerinde kullanıcının belirlediği kişiselleştirmeleri gerçekleştirmek üzere mobil uygulama sistemindeki ‘UserData’ sınıfından yeni bir nesne yaratılır ve bu nesne elde edilen profil bilgileri ile doldurulur.

5.1.6 Profil Güncelleme

Duyuruların listelenmesini kişiselleştiren tercihler ‘Duyuru Ayarları’ menüsünden güncellenebilir. Bu menü aracılığıyla kullanıcının duyuruların filtrelenmesi için seçtiği etkinlikler, birimler ve tarih aralığıyla birlikte duyuruların listelenmesinde önceliklendirme yapılabilmesi için anahtar kelimelerin girişi yapılır.

Yapılan seçimlerin aktif olması duyuru listesine yansıtılması için menüden çıkışta ‘Kaydet’ tuşunun seçilmesi gerekmektedir. Menü aracılığıyla yapılan tercihler kullanıcı profili olarak saklanması için orta katmandaki veritabanına yazılır. Değiştirilen yeni profil bilgilerinin kaydedilmesi işlemi Ege Web Servis sistemindeki web servisi üzerinden ilgili web metotların çağırımıyla gerçekleştirilir. Kullanıcı profil bilgilerinin kaydedilmesi için web servisinin çağırımına ilişkin kod, çizelge 5.6’da gösterilmiştir.

Çizelge 5. 6 Profilleri güncelleyen web servisinin çağırımı

```

EgeWebService.Service1_Stub s = new Service1_Stub();
try {
    s.setActivity(getTxtUserName().getString(), etkinlik_str);
    s.setDepartment(getTxtUserName().getString(), birim_str);
    s.setInterval(getTxtUserName().getString(), interval);
    s.setKeywords(getTxtUserName().getString(), etiket_str);
} catch (Exception ex) {
    ex.toString();
}

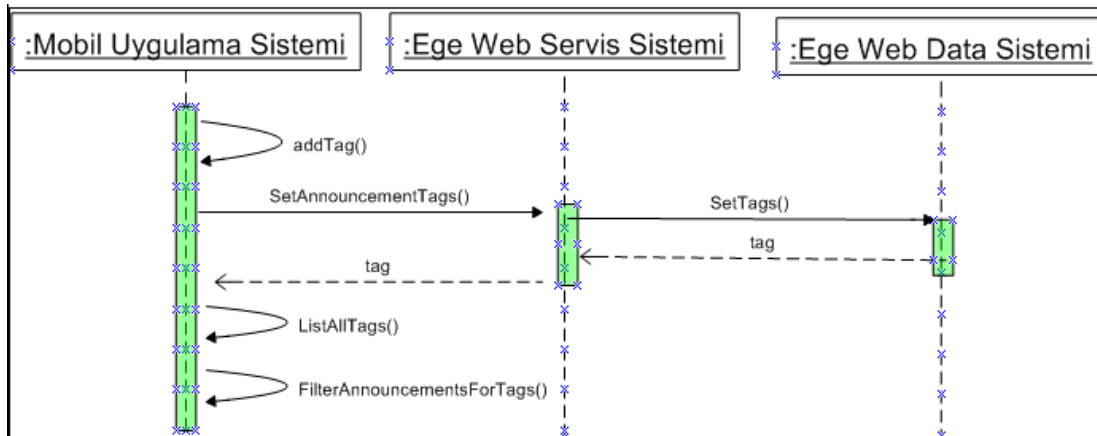
```

Çizelge 5.6’da da görüldüğü üzere Ege Web Servis sistemindeki web servisi üzerinden seçilen etkinliklerin değiştirilmesi için ‘setActivity’, seçilen birimlerin değiştirilmesi için ‘setDepartment’, seçilen tarih aralığının değiştirilmesi için ‘setInterval’ ve girilen anahtar kelimelerin değiştirilmesi içinse ‘setKeywords’ web metotları kullanılır.

5.1.7 Duyuruların etiketlenmesi

Gerçekleştirilen mobil uygulamada, uygulanan kişiselleştirmeler kapsamında duyurulara etiket ekleme, duyurulara eklenen etiketleri gösterme ve girilmiş olan tüm etiketlere göre duyuruları filtreleme destekleri mevcuttur.

Şekil 5.7’de duyurulara etiket ekleme, etiketleri listeleme ve etiketlere göre duyuruları filtreleme senaryosuna ilişkin sıra diyagramı gösterilmektedir.



Şekil 5. 7 Duyuruları etiketleme, etiketleri listeleme/filtreleme senaryosu sıra diyagramı

Şekil 5.7'deki sıra diyagramında gösterildiği üzere duyurulara etiket eklemek için ilk olarak en üst katmandaki Mobil Uygulama Sistemi aracılığıyla kullanıcı duyuruya eklemek istediği etiketlerin girişini yapar. İkinci adım olarak girilen etiketlerin en alt katmandaki Ege Web Data Sistemi üzerindeki duyuru bilgilerinin tutulduğu veritabanına kaydedilmesi gerçekleştirilmiştir. Bunun için girilen etiketler önce orta katmandaki Ege Web Servis Sistemine, ardından da alt katmandaki Ege Web Data Sistemine iletilir. Üçüncü adımda sistemdeki tüm duyurulara ait etiketlerin listelenmesi gösterilmiştir. Bu işlem, üst katmandaki Mobil Uygulama Sisteminde gerçekleştirilmektedir. Dördüncü adımda ise duyuruların kullanıcının seçtiği bir etikete göre filtrelenmesi yer alır. Bu işlem de, üst katmandaki Mobil Uygulama Sisteminde gerçekleştirilmektedir.

Güvenlik gerekçeleriyle etiket ekleme yetkisi sadece Ege Üniversitesi öğretim üyelerine verilmiştir. Kullanıcı sisteme giriş yaptığında girdiği kullanıcı adı ile fakülte bilgilerinin sorgusu için Ege Web Servis sistemindeki web servisi üzerinden 'getDepartmentInformation' adlı web metodun çağrımı gerçekleştirilir. Web servisinden dönen değere göre kullanıcının yetkileri belirlenir.

Çizelge 5. 7 Etiket ekleme işlemi için EgeWeb Servis sistemindeki web metodun çağrımı

```
EgeWebService.Service1_Stub s = new Service1_Stub();
try {
    s.setAnnouncementTags(duyuru.getId(), tags);
} catch (Exception ex) {
    ex.toString();
}
```

Etiket ekleme işlemi Ege Web Servis sistemindeki web servisi üzerinden etiketlerin ekleneceği duyuruya ait indeks ile birlikte 'setAnnouncementTags' adlı web metodun çağrımı ile gerçekleştirilir (Çizelge 5.7). Bu şekilde orta katmana iletilen etiketler, duyuru indeksi ve etiketlerle birlikte 'setKeywords' adlı en alt katmandaki web servisi çağrılarak alt katmandaki duyuru veritabanına yazılır (Çizelge 5.8).

Çizelge 5. 8 Etiket ekleme işlemi için Ege Web Data Sistemindeki web metodun çağırımı

```
[WebMethod]
public string SetAnnouncementTags(string id, string tags)
{
    ArrangeComingCharacters(tags, false);

    phpws3.EgeDuyuruWS phpws3 = new phpws3.EgeDuyuruWS();
    return phpws3.SetKeywords(id, tags);
}
```

Mevcut etiketlerin duyuru detaylarındaki gösterimi ve etiketlere göre filtreleme işlemleri, uygulamaya girişte vektöre aktarılan duyuruların etiket sahaları üzerinden en üst katman olan mobil uygulama sisteminde gerçekleştirilir.

5.2 Web Uygulama Sistemi

Mimarinin en üst katmanı olan uygulama katmanında yer alan Web Uygulama Sistemi, uygulamanın web üzerinde çalışmasını sağlar. Burada uygulama ile kullanıcı arasındaki etkileşimler internet üzerinden gerçekleştirilir. Sistem, kullanıcının arayüzler ile istediği hizmetin seçimini ve istediği hizmetlere göre orta katmandaki Ege Web Servis Sistemindeki ilgili web servislerini kullanarak veri alışverişi yaparak kullanıcının istediği hizmetin sunumunu gerçekleştirir.

Web uygulama sistemi, C# ve ASP .NET 2.0 teknolojileri kullanılarak, Microsoft Visual Studio 2008 üzerinde gerçekleştirilmiştir.

Web uygulama sisteminde, ASP .NET 2.0 ve AJAX (Asynchronous JavaScript and XML) teknolojileri kullanılarak verilerin kişiselleştirmeye izin verecek bağımsız birimler halinde sunulduğu bir portal tasarlanmıştır. Uygulama kapsamında Ege Üniversitesi web sitesi üzerinde yer alan “egetel”, “egeduyuru” isimli hizmetler web servislerine dönüştürülerek portal üzerinde kişiselleştirilebilir bağımsız birimler olarak sunulmuştur (Şen, 2008).

Web uygulama sistemi Şen’in 2008 yılında tamamladığı “Web Part ve Ajax Teknolojileri ile Kişiselleştirilebilir Portal Uygulaması” adlı tezi kapsamında gerçekleştirilmiş, bu tez çalışmasında ise mevcut sistemin yeni kurulan alt yapıya uyum sağlaması sağlanmıştır.

Web uygulama sisteminde gerçekleştirilen hizmetler mobil uygulama sistemi üzerinde gerçekleştirilen hizmetlerle birkaç farklılık dışında genel olarak aynıdır. Web uygulama sisteminde mobil uygulama sisteminde de yer alan sisteme giriş yapma, rehberden kayıt arama ve duyuruların listelenmesi hizmetleri mevcuttur. Bunların yanı sıra web sisteminde mobil sistemde olmayan sisteme kayıt olma, mobil uygulamayı indirme, bu hizmetlerin kişisel portal oluşturmaya izin veren web part'lar olarak sunulması ve uygulamaya internet üzerinde bulunan GoogleMaps gibi hazır webpart'ların eklenmesi gibi hizmetler de bulunmaktadır.

Mobil uygulamayla web uygulaması arasındaki bir fark da duyuruların listelenmesinde mevcuttur. Mobil uygulamadaki anahtar kelimeler ve içsel olarak bağlı olunan fakülte-bölüm bilgisine göre yapılan önceliklendirme ve etiket mekanizması web uygulamasında bulunmamaktadır. Bu tarz kişiselleştirme uygulamaları, mobil cihazların ekran boyutu ve kullanım zorluğu açısından mobil cihazlarda çalışan uygulamalarda uygulamanın kullanımını kolaylaştırırken, web uygulamalarında gereksiz hatta bazen kullanıcıyı kısıtlayıcı olabilmektedir.

Aynı doğrultuda mobil uygulamayla web uygulaması arasındaki bir diğer fark da Ege Üniversitesi rehberinin kullanımında mevcuttur. Mobil uygulamada az önce belirtilen kısıtlardan dolayı sunulan bilginin kısa ve net olması, uygulamanın daha az kullanıcı etkileşimiyle yönetilmesi gerekmektedir. Bu sebeple mobil uygulamada rehberden girilen isim bilgisine göre arama yapıp sonuçlar teker teker kullanıcıya listelenirken, web uygulamasında tüm rehber kayıtlarının ve arama sonuçlarının ağaç yapısında sunumu bulunmaktadır.

Web uygulama sistemiyle mobil uygulama sisteminin farklı teknolojilerle, farklı programlama dilleriyle, farklı platformlar için geliştirilmiş olmalarına rağmen önceki bölümde anlatılan mimarinin katmanlı bir şekilde tasarlanması ve web servislerinin kullanımı sayesinde her iki uygulamada da aynı yapı kullanılabilir. Hatta istenildiği takdirde sisteme, farklı teknolojilerle farklı platformlar üzerinde çalışacak yeni uygulamaların entegre edilmesi kolaylıkla gerçekleştirilebilir.

Bu sebeple Web uygulamasındaki senaryoların gerçekleştirimi farklı gereksinimler dışında genel anlamda mobil uygulamayla aynı doğrultudadır. Aşağıda bu durum bir örnekle gösterilmektedir.

Rehberden kayıt arama senaryosundaki girilen isim ve/veya soyisim sorgulanması mobil uygulamada olduğu gibi web uygulamasında da Ege Web Servis sistemindeki web servisi üzerinden isim ve/veya soyisim parametresiyle birlikte 'getName' adlı web metodun çağrımıyla gerçekleştirilir. Web uygulamasındaki rehberden kayıt arama sorgusu için web servisinin çağrımına ilişkin kod, çizelge 5.9 'da gösterilmiştir.

Çizelge 5. 9 Web uygulamasındaki rehberden kayıt arama sorgusu için web servisinin çağrımı

```
phpws1.EgetelWS phpws1 = getphpws();
try
{
    str = phpws1.GetName(keyword);
}
catch (Exception exc)
{
    str = null;
}
```

Çizelge 5.9'da görüldüğü üzere mobil uygulamayla web uygulaması arasında farklı teknolojilerin kullanımı dışında bir fark bulunmamaktadır. İki tarafta da aynı web servisi kullanılmaktadır.

Web uygulamasında mobil uygulamadan farklı olarak farklı gereksinimler doğrultusunda Ege Web Servis sistemindeki web servisi üzerinden çizelge 5.10'daki web metotlarının çağrımı gerçekleştirilir:

Çizelge 5. 10 EgeWebpart sistemindeki web servisi üzerinden web metotlarının çağrımı

```
phpws1.EgetelWS phpws1 = getphpws();
string str = phpws1.GetAllDepartmentInformation("");
```

```
phpws1.EgetelWS phpws1 = getphpws();
try
{
    str = phpws1.GetAllTableInformation("");
}
catch { }
```

Çizelge 5.10'da kullanımı gösterilen web metotları ağaç yapısında Ege Üniversitesi rehberinin gösteriminde kullanılmaktadır.

5.3 Ege Web Servis Sistemi

Ege Web Servis Sisteminin bulunduğu orta katman, en üst katmandaki uygulamalarla en alt katman arasında bilgi alışverişini sağlayan C# web servislerinden ve ASP .NET 2.0'ın kişiselleştirmeye izin veren yapısıyla birlikte kullanıcıya ait profil verilerini saklayan veritabanından oluşmaktadır.

Ege Web Servis Sistemi, C# ve ASP .NET 2.0 teknolojileri kullanılarak, Microsoft Visual Studio 2008 üzerinde gerçekleştirilmiştir.

Ege Web Servis Sisteminde, en üstteki uygulama katmanıyla en alttaki veri katmanı arasında soyutlama yapılmaktadır. Aradaki orta katman sayesinde üst katmana farklı teknolojilerle, farklı platformlar üzerinde çalışacak yeni uygulamaların entegrasyonu kolaylıkla gerçekleştirilebilir. Ya da aynı şekilde en alt veri katmanında yapılacak veritabanının yerinin, teknolojisinin değiştirilmesi gibi değişiklikler diğer sistemlerde hiçbir değişiklik gerektirmeden bağımsız bir şekilde gerçekleştirilebilir. Kısaca bu katman iki katman arasında arayüz görevi üstlenir.

Ayrıca Ege Web Servis Sisteminde, ASP .NET 2.0 ile gelen kişiselleştirme desteği ile kullanıcı yetkilendirme ve kullanıcılara ait profil bilgilerini saklama gereksinimleri gerçekleştirilmektedir. Kullanıcı bilgileri burada oluşturulan veritabanında saklanmaktadır. Kullanıcı bilgilerinin orta katmandaki Ege Web Servis Sisteminde saklanmasıyla verilerin uygulama bağımsız yönetilmesi sağlanmaktadır. Böylece üst katmandaki bir uygulama kullanıcı bilgileri üzerinde bir değişiklik yaptığında diğer uygulama hiçbir ekstra iş gerçekleştirmeden değişiklikten haberdar olacaktır.

Aşağıda Ege Web Servis Sisteminde kullanılan web servisleri detaylı bir şekilde açıklanmaktadır.

Sisteme giriş aşamasında girilen kullanıcı adı ve şifrenin sorgulanması Ege Web Servis sistemindeki web servisinde 'Login' adlı web metodun çağırımıyla gerçekleştirilir. Login fonksiyonu içerisinde ASP .NET 2.0'ın sağladığı hazır kullanıcı doğrulama fonksiyonu kullanılmaktadır. Kullanıcı adı ve şifre sorgusu için web servisindeki web metoda ilişkin kod, çizelge 5.11'de gösterilmiştir.

Çizelge 5. 11 Kullanıcı adı-şifre doğrulama

```
[WebMethod]
public bool Login(string userName, string password)
{
    return Membership.ValidateUser(userName, password);
}
```

Uygulama tarafında rehberden kayıt arama işlemi Ege Web Servis sistemindeki web servisinde ‘GetName’ adlı web metodun çağrımıyla gerçekleştirilir. GetName fonksiyonu içersinden en alt katman olan veri katmanındaki Ege Üniversitesi rehberine erişme yetkisi olan php web servisi çağrılarak rehberden isme göre sorgu yapılabilir. Rehberden isim bilgisi elde etmek için web servisindeki web metoda ilişkin kod, çizelge 5.12’de gösterilmiştir.

Çizelge 5. 12 Rehberden isim bilgisi elde etmek için web servisindeki web metod çağrımı

```
[WebMethod]
public string GetName(string keyword)
{
    keyword = ArrangeComingCharacters(keyword, false);
    try
    {
        phpws1.EgetelWS phpws1 = new phpws1.EgetelWS();
        string s = "";
        s = phpws1.GetName(keyword);
        return s;
    }
    catch (Exception ex)
    {
        throw;
    }
}
```

Uygulama tarafında rehberden kayıt arama işlemi sırasında isim ile birlikte kişinin fakülte-bölüm bilgilerini göstermek için isim sorgusuyla web servisten dönen kişinin bağlı olduğu birim kodu parametreleriyle Ege Web Servis sistemindeki web servisinde ‘GetDepartmentInformation’ adlı web metodun çağrımı gerçekleştirilir. GetDepartmentInformation fonksiyonu içersinden en alt katman olan veri katmanındaki Ege Üniversitesi rehberine erişme yetkisi olan php web servisi çağrılarak rehberden fakülte-bölüm bilgisi sorgulanır. Rehberden fakülte, bölüm, e-posta gibi bilgileri çekmek için web servisindeki web metoda ilişkin kod, çizelge 5.13’de gösterilmiştir.

Çizelge 5. 13 Rehberden fakülte, bölüm, e-posta gibi bilgileri çekme

```

[WebMethod]
public string GetDepartmentInformation(string kkodu, string bkodu, string abirimkod1, string abirimkod2, string abirimkod3)
{
    phpws1.EgetelWS phpws1 = new phpws1.EgetelWS();
    return phpws1.GetDepartmentInformation(kkodu, bkodu, abirimkod1, abirimkod2, abirimkod3);
}

```

Uygulama tarafında duyuru hizmeti seçildiğinde tüm duyuruları Ege Üniversitesi veritabanından alma işlemi Ege Web Servis sistemindeki web servisinde ‘GetAllAnnouncements’ adlı web metodun çağırımıyla gerçekleştirilir. GetAllAnnouncements fonksiyonu içersinden en alt katman olan veri katmanındaki Ege Üniversitesi duyurularına erişme yetkisi olan php web servisi çağrılarak tüm duyurular elde edilir. Duyuruları getirmek için web servisindeki web metoda ilişkin kod, çizelge 5.14’de gösterilmiştir.

Çizelge 5. 14 Duyuruları getirmek için web servisindeki web metodun kullanımı

```

[WebMethod]
public string GetAllAnnouncements(string keyword)
{
    phpws3.EgeDuyuruWS phpws3 = new phpws3.EgeDuyuruWS();
    return phpws3.GetAllAnnouncements(keyword);
}

```

Uygulama tarafında duyuruya etiket eklenmesi istendiğinde etiketlerin Ege Üniversitesi veritabanına kaydedilme işlemi Ege Web Servis sistemindeki ‘SetAnnouncementTags’ adlı web metodun çağırımıyla gerçekleştirilir. SetAnnouncementTags fonksiyonu içersinden en alt katmandaki Ege Web Data Sisteminde Ege Üniversitesi duyurularına erişme yetkisi olan php web servisi çağrılarak ilgili duyuruya girilen etiket kaydedilir. Duyurulara etiket eklemek için web servisindeki web metoda ilişkin kod, çizelge 5.15’de gösterilmektedir.

Çizelge 5. 15 Duyurulara etiket eklemek için web servisindeki web metodun kullanımı

```
[WebMethod]
public string SetAnnouncementTags(string id, string tags)
{
    ArrangeComingCharacters(tags, false);

    phpws3.EgeDuyuruWS phpws3 = new phpws3.EgeDuyuruWS();
    return phpws3.SetKeywords(id, tags);
}
```

Uygulamada kullanıcıya ait ASP .NET 2.0'ın sağladığı profil bilgileri orta katmandaki yine ASP .NET 2.0'ın sağladığı veritabanında tutulur. Bu profil bilgilerine de Ege Web Servis sistemindeki web servisleriyle erişilir ve değeri değiştirilebilir. Profil bilgisi içerisinde saklanan, duyuruların filtrelenmesinde kullanılan, kullanıcının görmek istediği duyuruların türlerini belirleyen etkinlik tercihlerine 'GetActivity' adlı web metodun çağrımıyla erişilir. Aynı şekilde birimlere göre filtrelemeyi sağlayan GetDepartment, duyuruların tarih aralığına göre filtrelenmesini sağlayan GetInterval ve duyuruların kullanıcının girdiği anahtar kelimelere göre önceliklendirilmesini sağlayan GetKeywords adlı web metodlarıyla kullanıcı profilinde saklanan diğer profil bilgilerine erişilebilir. Kullanıcı profilindeki etkinlik tercihlerini elde etmek için web servisindeki web metoda ilişkin kod, çizelge 5.16'da gösterilmiştir.

Çizelge 5. 16 Kullanıcı profilindeki etkinlik tercihlerini getiren web metod

```
[WebMethod]
public string GetActivity(string userName)
{
    ProfileBase userProfile = ProfileBase.Create(userName);
    return userProfile.GetPropertyValue("Activity").ToString();
}
```

Yukarıda açıklanan profil bilgisi içerisinde saklanan kullanıcının etkinlik, birim, tarih aralığı ve anahtar kelime tercihleri Ege Web Servis sistemindeki web servisleriyle 'SetActivity', 'SetDepartment', 'SetInterval', 'setKeywords' adlı web metodları ile Ege Web Servis sistemindeki veritabanına kaydedilir. Kullanıcı profilindeki etkinlik tercihlerine yeni değer atamak için web servisindeki web metoda ilişkin kod, çizelge 5.17'de gösterilmiştir.

Çizelge 5. 17 Kullanıcı profilindeki etkinlik tercihlerine yeni değer atama

```

[WebMethod]
public void SetActivity(string userName, string activity_val)
{
    activity_val = ArrangeComingCharacters(activity_val, true);

    ProfileBase userProfile = ProfileBase.Create(userName);
    userProfile.SetPropertyValue("Activity", activity_val);
    userProfile.Save();
}

```

Yukarıda detaylı bir şekilde açıklandığı üzere uygulamanın gerek duyduğu hizmetler Ege Web Servis Sisteminde karşılanır. Kimi hizmetler doğrudan bu sistemde gerçekleştirilirken, kimi hizmetler için Ege Web Servis Sistemi aracı sistem görevini üstlenip bir alt katmandaki php ile yazılan web servislerini kullanmaktadır.

5.4 Ege Web Data Sistemi

Mimarinin en alt katmanındaki Ege Web Data Sistemi, orta katmandaki web servislerinin iletişim kurduğu php web servislerinden ve bu web servislerinin veri alışverişini sağlayan Ege Üniversitesi kurumsal web sitesi üzerinden de erişilen ‘egetel’ ve ‘db_egeduyuru’ isimli veritabanlarından oluşmaktadır.

Ege Web Data Sistemi, web servisleri için php, NuSOAP, veritabanları içinse mySql teknolojileri kullanılarak gerçekleştirilmiştir.

NuSOAP bileşen tabanlı Web Servisi araç kitidir. Web servisi etkileşimi soapclient olarak adlandırılan üst seviye bir istemci sınıfı ile sağlanmaktadır. Bu üst seviyeli sınıf kullanıcılara SOAP mesajının kendisinin gönderilmesi ve alınmasını yönetmenin yanısıra HTTP yetkilendirme kimliği, HTTP proxy bilgileri gibi seçenekleri belirleme olanağını sunmaktadır (Şen, 2008).

Ege Web Data Sisteminde, ‘egeTel’ ve ‘egeDuyuru’ isimli iki web servisi bulunmaktadır. ‘EgeTel’ web servisi kendisiyle aynı ismi taşıyan ‘egetel’ isimli veritabanından, ‘egeDuyuru’ web servisi is ‘db_egeduyuru’ isimli veritabanından bilgi alışverişi yapmakla sorumludur. Bu web servisleri ve görevleri izleyen bölümlerde detaylı bir şekilde açıklanmaktadır.

5.4.1 EgeTel web servisi

Ege Web Data sistemindeki EgeTel web servisi, uygulamanın Ege Üniversitesi telefon rehberi üzerinden gerçekleştirmek istediği tüm işlemlerde uygulamaya veriyi sağlayan en temeldeki yapıdır. Ege Üniversitesi kurumsal web sitesi üzerinden de erişilen rehbera ait verileri tutan veritabanına erişerek gerekli sorguları yapmaktadır.

EgeTel web servisi NuSoap kullanılarak geliştirilmiştir. Bu sebeple web servisi içinde ilk olarak NuSoap sınıfı dahil edilir. Daha sonra web servisi içinde yer alan tüm fonksiyonlar parametre ve geri dönüş değerleriyle sunucuya tanıtılır. Ve ardından fonksiyonların gerçekleştirimleri yer alır.

EgeTel web servisi içinde uygulamadaki telefon rehberinden yapılan farklı sorgular için ‘egeTel’ veritabanına ulaşım ve ilgili alanların geri döndürülmesi yer alır. Uygulamanın gereksinim duyduğu farklı verilerin uygulamaya döndürülmesi için farklı fonksiyonlar kullanılmaktadır. Rehberden kayıt arama işlemi için isim ve/veya soyisim sorgulanması ‘GetName’ fonksiyonuyla gerçekleştirilir. Fakülte bilgilerinin sorgusu içinse ‘GetName’ fonksiyonunun çağrımından dönen birim kodu parametreleriyle birlikte ‘GetDepartmentInformation’ adlı fonksiyonun çağrımı gerçekleştirilir. Ege Üniversitesine bağlı tüm fakülteleri yani hiyerarşinin en üst basamağını uygulamaya döndüren sorgu ‘GetAllDepartmentInformation’ adlı fonksiyonda yer alır. Burada dönen fakültelerin bölümleri gibi alt birimlerini sorgulayıp ağaç yapısında sunumunu sağlayan fonksiyonlar ise ‘GetAllTableInformation’ ve ‘GetAllRehberTableInformation’dır.

GetName fonksiyonu için veritabanına bağlantı, ilgili tablodan sorgu ve dataset oluşturma işlemleri çizelge 5.18’de gösterildiği şekilde yapılmaktadır.

Çizelge 5. 18 Php web servisi üzerinden GetName metodunun çağırımı

```

Function GetName($keyword)
{
    if($keyword == "") return "";
    $connection = mysql_connect ("egeweb.ege.edu.tr", "usr_egupart", "3g3.p#rt");

    //veritabanı ismi: egetel
    mysql_select_db ("egetel", $connection);

    //İsim bilgisini içeren kayıtları döndürecek fonksiyon.
    $query = "SELECT * FROM rehber WHERE isim LIKE '" . $keyword . "' . "ORDER BY isim DESC";
    $result = mysql_query ($query, $connection);

    while ($record = mysql_fetch_array ($result))
    {
        $kkodu = $record["kkodu"];
        $bkodu = $record["bkodu"];
        $abirimkod1 = $record["abirimkod1"];
        $abirimkod2 = $record["abirimkod2"];
        $abirimkod3 = $record["abirimkod3"];
        $isim = $record["isim"];
        $dtelefon = $record["dtelefon"];
        $htelefon = $record["htelefon"];
        $ctelefon = $record["ctelefon"];
        $faks = $record["faks"];
        $email1 = $record["email1"];
        $email2 = $record["email2"];
        $weburl1 = $record["weburl1"];
        $weburl2 = $record["weburl2"];
        $sira = $record["sira"];
        $temp = $record["temp"];

        $dataset = $dataset.$kkodu."£".$bkodu."£".$abirimkod1."£".$abirimkod2."£".$abirimkod3
            .£".$isim."£".$dtelefon."£".$htelefon."£".$ctelefon."£".$faks."£".$email1
            .£".$email2."£".$weburl1."£".$weburl2."£".$sira."£".$temp."£";
    }

    mysql_free_result ($result);
    mysql_close($connection);

    return $dataset;
}

```

Ege Web Data sistemindeki EgeTel web servisi içerisindeki diğer fonksiyonların da kodu çizelge 5.18'deki 'GetName' adlı fonksiyonun gerçekleştirimiyle berzerdir. Hepsinde bağlantı cümlesi, seçilen veritabanı aynıdır. Aralarındaki fark, bazı fonksiyonlar 'rehber' tablosundan gerekli sorguları yaparlarken bazıları fakülte bilgilerini almak için 'kurulus' tablosundan gerekli sorguları yapmaktadırlar.

5.4.2 EgeDuyuru web servisi

Ege Web Data sistemindeki EgeDuyuru web servisi, uygulamanın Ege Üniversitesi duyurularına ulaşma gereksiniminde uygulamaya veriyi sağlayan en temeldeki yapıdır. EgeDuyuru web servisi, Ege Üniversitesi kurumsal web sitesinde de yayımlanan duyuruları tutan db_egeduyuru veritabanına erişerek uygulamanın istediği sorguları yapmaktadır.

EgeDuyuru web servisi de EgeTel gibi NuSoap kullanılarak geliştirilmiştir. Bu sebeple iki web servisi çok benzer yapıdadır. Gerçekleştirim aşamasında aynı adımları içerirler. Web servisi içinde ilk olarak NuSoap sınıfı dahil edilir. Daha sonra web servisi içinde yer alan fonksiyonlar parametre ve geri dönüş değerleriyle birlikte sunucuya tanıtılır. Ve ardından fonksiyonların gerçekleştirmeleri yer alır.

EgeDuyuru web servisi içinde yer alan ‘GetAllAnnouncements’ adlı fonksiyon ‘db_egeduyuru’ veritabanında mevcut olan geçerli tüm duyuruları döndürmekten sorumludur. GetAllAnnouncements fonksiyonu için veritabanına bağlantı, ilgili tablodan sorgu ve dataset oluşturma işlemleri çizelge 5.19’da gösterildiği şekilde yapılmaktadır.

Çizelge 5. 19 Php web servisi üzerindeki GetAllAnnouncements web metodu

```
function GetAllAnnouncements($keyword)
{
    $connection = mysql_connect ("egeweb.ege.edu.tr","usr_egепart","3g3.p4rt");

    //Veritabanı ismi : db_egeduyuru
    mysql_select_db ("db_egeduyuru",$connection);
    mysql_query("SET NAMES 'latin1'");

    $query = "SELECT * FROM duyurular2 where bastarih > (CURDATE()-1) ORDER BY bastarih DESC";
    $result = mysql_query ($query,$connection);

    while ($record = mysql_fetch_array ($result))
    {
        $gonderen = $record["gonderen"];
        $konu = $record["konu"];
        $tur = $record["tur"];
        $icerik = $record["icerik"];
        $tarih = $record["tarih"];
        $dyer = $record["dyer"];
        $dsene = $record["dsene"];
        $day = $record["day"];
        $dgun = $record["dgun"];
        $yenidsene = $record["yenidsene"];
        $yeniday = $record["yeniday"];
        $yenidgun = $record["yenidgun"];
        $bastarih = $record["bastarih"];
        $bittarih = $record["bittarih"];
        $ic0dis1 = $record["ic0dis1"];
        $turkceOyabanci1 = $record["turkceOyabanci1"];
        $keywords = $record["keywords"];
        $id = $record["id"];

        $dataset = $dataset.$id."$".$gonderen."$".$konu."$".$tur."$".$icerik."$".$tarih."$".
        $dyer."$".$dsene."$".$day."$".$dgun."$".$yenidsene."$".$yeniday."$".$yenidgun."$".
        $bastarih."$".$bittarih."$".$ic0dis1."$".$turkceOyabanci1."$".$keywords."$";
    }

    mysql_free_result ($result);
    mysql_close($connection);

    return $dataset;
}
```

EgeDuyuru web servisi içersinde duyuruların etiket sahalarını güncellemek içinse ‘SetKeywords’ adlı fonksiyon kullanılır. Bu fonksiyon için veritabanına bağlantı, ilgili tabloda kayıt güncelleme işlemleri çizelge 5.20’de gösterildiği şekilde yapılmaktadır.

Çizelge 5. 20 Php web servisi üzerindeki SetKeywords web metodu

```

function SetKeywords($duyuru_id, $duyuru_keywords)
{
    $connection = mysql_connect ("egeweb.ege.edu.tr","usr_egeweb","3g3.p4rt");

    //Veritabanı ismi : db_egeduyuru
    mysql_select_db ("db_egeduyuru",$connection);
    mysql_query("SET NAMES 'latin1'");

    $query = "UPDATE duyurular2 SET keywords='".$duyuru_keywords.'" where id = ".$duyuru_id;
    $result = mysql_query ($query , $connection);

    mysql_free_result ($result);
    mysql_close($connection);

    return $dataset;
}

```

5.5 Uygulamada Kullanılan Kişiselleştirme Teknikleri

Uygulama kapsamında kullanıcı ihtiyaçlarına daha etkin cevap verebilmek için farklı kişiselleştirme teknikleri kullanılmıştır. Dördüncü bölümde detaylı bir şekilde anlatıldığı üzere kişiselleştirme yöntemleri kullanıcı düzenlemelerine dayalı kişiselleştirme ve uyarlanabilir kişiselleştirme olarak iki grupta toplanmaktadır.

Tez kapsamında gerçekleştirilen mobil uygulamada her iki kişiselleştirme tekniğinden de yararlanılmıştır. Yalnız kullanıcı düzenlemelerine dayalı kişiselleştirme tekniklerinin bu uygulamanın kullanımında daha fazla yararlı olacağı düşünülerek bu tekniğe ait yöntemlerden daha çok yararlanılmıştır.

Kullanıcı düzenlemelerine dayalı kişiselleştirmeye özelleştirme de denebilir. Burada yapılan kişiselleştirme doğrudan kullanıcının kontrolünde gider. Kullanıcı girdiği portal veya internet sitesi ile bundan sonraki ilişkisi konusunda belli alternatifler arasından seçim yapar. Bu seçim tamamen kullanıcının kontrolündedir ve gelecekte tercihlerini değiştirme şansına sahiptir (Workcube, 2010).

Dördüncü bölümde detaylı bir şekilde anlatıldığı üzere kullanıcı düzenlemelerine dayalı kişiselleştirme tekniğinde mobil uygulamalar için öncelikle ilgili bir web sitesi aracılığıyla üyelik işlemlerinin gerçekleştirilmesi gerekmektedir. Tez kapsamında gerçekleştirilen mobil uygulamada da kullanıcının sisteme giriş yapabilmesi için öncelikle <http://www.egewebpart.ege.edu.tr/egewebpart2/> adresindeki portaldaki web arayüzü ile sisteme üyeliğini gerçekleştirmiş olması gerekmektedir. Web arayüzünde üyelik tanımlama aşamasında kullanıcının belirlediği kullanıcı adı ve şifresi ile mobil uygulamaya da giriş yapılabilir.

Uygulamada kullanıcı düzenlemelerine dayalı kişiselleştirme tekniklerinden sistem tarafından sunulan seçeneklere göre filtreleme, kullanıcının belirlediği anahtar kelimeye göre önceliklendirme, etiketleme ve etikete göre filtreleme yöntemleri kullanılmıştır.

Profil ayarları bölümünde sistemdeki tüm duyurulardan dinamik olarak elde edilen etkinlik türleri arasından kullanıcıya seçim yaptırılarak duyuruların etkinlik türlerine göre filtrelenebilir. Aynı şekilde yine profil ayarları bölümünde sistemdeki tüm duyurulardan dinamik olarak elde edilen gönderici birimler arasından kullanıcıya seçim yaptırılarak duyurular gönderen birimlere göre de filtrelenebilir. Bunların yanı sıra yine profil ayarları bölümünde duyurular tarih aralığına göre de filtrelendirilebilir. Burada tarih aralığına göre filtreleme seçenekleri sabit olup bugün, yarın, bir hafta, bir ay, iki ay ve üç ay şeklindedir. Yine profil ayarları menüsünde kullanıcının anahtar kelime girmesiyle bu girilen anahtar kelimeye göre duyuru listesindeki duyurular önceliklendirilebilir. Şekil 5.8’de profil ayarları menüsündeki yukarıda anlatılan kişiselleştirme seçenekleri gösterilmektedir.



Şekil 5. 8 Profil ayarları menüsündeki kişiselleştirme seçenekleri

Ayrıca kullanıcı duyuruları etiketlerine göre filtreleyebilir, herhangi bir duyuru için girilmiş etiketleri görebilir ve yetkisi dahilinde herhangi bir duyuruya etiket ekleyebilir. Bu teknik kullanıcıya daha özgür hareket edebilme olanağı sağlamaktadır. Ancak güvenlik gerekçelerinden dolayı duyurulara etiket ekleme yetkisi sadece Ege Üniversitesi öğretim üyelerine verilmiştir. Şekil 5.9’da uygulamadaki etiket girişi, etiket listesi ve duyuru detayları arasında etiketlerin gösterimleri ile ilgili ekran görüntüleri yer almaktadır.

Etiket Giriş Formu	Etiket Listesi	Ege Duyuru Detay
Duyuruya eklemek istediğiniz etiketi giriniz:	bach	Gonderen: Rektörlük-Bilgi İşlem
Qwertv	bahar	Daire Başkanlığı
Geri Menu	başvuru	Konu: İnsan Kaynakları Uzmanlık
	bir ada	Sertifika Programı - Egesem
	bir tenor aranıyor	Tur: Eğitim
	carmina burana-carl orff	Yer: EÜ Atatürk Kültür Merkezi
	carmina burana-carl orff	Tarih: 27 / 4 / 2010 18:30:00
	konseri	Etiket: İnsan Kaynakları Uzmanlık
	dans	Sertifika Programı, Egesem
	dans gösterisi	
	dans çalışma grubu	
	deneyim	
	eczacılık fakültesi	
Geri Menu	Geri Listele	Geri

Şekil 5. 9 Etiket girişi, etiket listesi ve duyuru detaylarında etiketlerin gösterimi

Yukarıda açıklanan tüm bu kişiselleştirme uygulamaları kullanıcı düzenlemelerine dayalı kişiselleştirme yöntemleri kullanılarak gerçekleştirilmiştir. Buradaki filtreleme tercihleri ve girilen anahtar kelime bilgisi kullanıcı tarafından belirlenir, kullanıcıya özeldir ve her bir kullanıcı için ayrı ayrı tutulmaktadır. Böylelikle uygulama her bir kullanıcı için farklılaşmakta, o kullanıcının belirlediği şekilde çalışmaktadır.

Uyarlanabilir kişiselleştirme yönteminde ise kişiselleştirme işlemi bilgisayarın kontrolü altındadır. Kullanıcı hakkında edinilen her türlü bilgi, bilgisayar tarafından kullanıcının ilgi alanına özel sayfaların sunulması amacıyla yorumlanır ve kullanılır. Hakkında bilgi edinilen kullanıcının ihtiyaçları ve ilgi alanları bilgisayar tarafından belirlenir ve buna göre bilgisayar bazı tavsiyelerde bulunur (Workcube, 2010).

Gerçekleştirilen mobil uygulamada uyarlanabilir kişiselleştirme tekniğinden de yararlanılmıştır. Uygulamada anahtar kelimelere ek olarak bir de kullanıcının bağlı olduğu fakülte-bölüm bilgisine göre üstü kapalı bir önceliklendirme gerçekleştirilir. Kullanıcı Ege Üniversitesi personeli ise bağlı olduğu fakülte ve bölüme ait duyuruları listenin en başında görmek isteyeceği varsayımıyla, kullanıcının bağlı olduğu fakülte ve bölüme ait herhangi bir duyuru mevcutsa, duyuru listesinde bu duyurular tarihlerinden bağımsız olarak önceliklendirilerek, listenin en başında yer alacak şekilde tasarlanmıştır. Burada uyarlanabilir kişiselleştirme tekniği kullanılarak kullanıcının bilgisi dışında otomatik olarak sistemin kullanıcıya özel tanımlandığı, içsel bir önceliklendirme gerçekleştirilir.

6. UYGULAMANIN MOBİL CİHAZA YÜKLENMESİ

Bir önceki bölümde gerçekleştirimi anlatılan mobil uygulama, cihaz konfigürasyonu CLDC-1.1 ve cihaz profili MIDP-2.0 olan Java destekli mobil cihazlar üzerinde çalıştırılabilir.

Uygulamanın mobil cihaza yüklenebilmesi için ilk olarak web uygulamasının bulunduğu <http://www.egewebpart.ege.edu.tr/egewebpart2/> adresinden portala giriş yapılarak uygulamanın buradan indirilmesi gerekmektedir.

J2ME mobil uygulamaları cihaza iki dosya halinde yüklenir.

jar dosyası: Uygulamada kullanılan dosyalar ve derlenmiş haldeki sınıf dosyaları jar dosyasının içinde yer alır.

jad dosyası: Uygulamanın mobil cihaza düzgün yüklenip kullanılabilmesi için mutlaka jad dosyası çalıştırılarak yüklenmelidir. Jad dosyası uygulamanın yüklenme adresini, uygulamanın çalışması için gerekli izinleri içerir.

Uygulamayı cihaza yüklemek için cihazın tipine göre birçok yöntem vardır. Bunlar:

- Uygulama adresini içeren SMS, MMS, WAP Push mesajları ile yükleme
- MMS, E-posta ile dosyayı doğrudan gönderme
- Cihazı bilgisayara bağlayıp cihazın kendi yardımcı programı aracılığıyla yükleme yapmak. USB veri kablosuyla yükleme yapmak için USB kablo bilgisayara ve cihaza takılır. Bilgisayarın cihazı görüp görmediği kontrol edilir. Bilgisayar cihazı görmüşse indirilen jar ve jad uzantılı dosyalar mobil cihaza gönderilir. Gönderim tamamlandıktan sonra jad uzantılı dosya açılarak uygulamanın cihaza yüklenmesi gerçekleştirilir.
- Uygulama dosyalarını cihazın hafızasına bilgisayar, bluetooth, ir-da aracılığıyla kopyalayıp jad veya jar dosyasını çalıştırmak. Bluetooth ile yükleme yapmak için mobil cihazın bluetooth menüsünden bluetooth açılır, bilgisayardaki bluetooth programı da açıldıktan sonra mobil cihaz ile bilgisayar eşlenir. Bilgisayar mobil cihazı bluetooth üzerinden görebilir

hale geldikten sonra jar ve jad uzantılı dosyaların üzerinde sağ tıklayıp bluetooth cihazına gönder seçeneği seçilir. Bu seçim sonrasında bilgisayarın bluetooth üzerinden gördüğü cihazların listesi çıkar. Bu listeden göndermek istenilen cihaz seçilir. Seçim sonrası mobil cihaza gelen bir mesajla bilgisayar ile eşlemenin yapıldığı bildirilir ve gönderim işlemi için kullanıcı onayı istenir. Bu mesaj onaylanırsa bluetooth ile gönderim başlar.

Jar ve jad uzantılı dosyaların gönderimi tamamlanınca jad uzantılı dosya açılarak uygulama cihaza yüklenir. Jad dosyasını desteklemeyen cihazlarda jar dosyası yüklenmelidir. Yükleme tamamlandıktan sonra uygulama çalıştırılabilir.

7. UYGULAMANIN KULLANIMI

Bir önceki bölümde anlatıldığı şekilde özellikleri uygun bir mobil cihaza yüklenen uygulama, ilk çalıştırıldığında şekil 7.1’de gösterilen kullanıcı bilgileri giriş ekranı çıkar.



Şekil 7. 1 Kullanıcı bilgileri giriş ekranı

Kullanıcının sisteme giriş yapabilmesi için öncelikle <http://www.egewebpart.ege.edu.tr/egewebpart2/> adresindeki portaldaki web arayüzü ile sisteme üyeliğini gerçekleştirmiş olması gerekmektedir. Web arayüzünde üyelik tanımlama aşamasında kullanıcının belirlediği kullanıcı adı ve şifre girilerek uygulamaya giriş yapılır.

Girilen kullanıcı adı ya da şifre yanlışsa şekil 7.2’de gösterildiği üzere ekrana bir uyarı mesajı çıkartılarak kullanıcı adı ve şifrenin tekrar girilmesi istenir.



Şekil 7. 2 Yanlış kullanıcı adı/şifre girişi uyarı ekranı

Sisteme giriş aşamasında girilen kullanıcı adı ve şifre doğruysa sisteme başarılı bir şekilde giriş yapılmıştır, bu durumda şekil 7.3’de gösterilen ana menü ekrana getirilir.



Şekil 7. 3 Ana menu ekranı

Ekrana gelen ana menüden Ege Tel ya da Ege Duyuru seçeneklerinden biri seçilir. Ege Üniversitesi telefon rehberinden kayıt aramak için ‘Ege Tel’, Ege Üniversitesi duyurularını listelemek içinse ‘Ege Duyuru’ seçilir.

‘Ege Tel’ seçildiğinde şekil 7.4’de gösterildiği üzere ekrana gelen arayüzden bilgilerini aramak istenilen kişiye ait isim ve/veya soyismin tamamı ya da bir kısmı girilerek arama başlatılır.



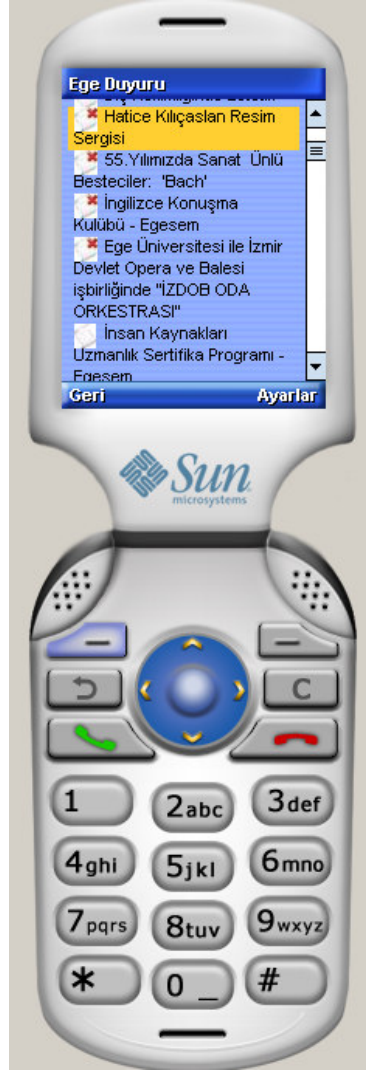
Şekil 7. 4 Rehberden kayıt arama ekranı

Arayüzden girilen isim bilgisini içeren tüm kayıtlar şekil 7.5’de gösterildiği şekilde kullanıcıya sunulur. Burada aranan kişinin ünvanı, fakülte, bölüm bilgisi ile birlikte kişi eğer Ege Üniversitesi resmi sitesi üzerinden kendi kişisel bilgilerini girmişse dahili telefon, harici telefon, e-posta ve web adresi bilgileri ekrana getirilir. Aranılan isim bilgisini içeren birden fazla kayıt bulunmuşsa bir sonraki bulunan kayda ‘Sonraki’ seçeneği ile ulaşılır.



Şekil 7. 5 Rehberden aranan kişiye ait kayıt ekranı

Ana menüden ‘Ege Duyuru’ seçeneği seçildiğinde Ege Üniversitesi duyurularının listelendiği şekil 7.6’da gösterilen duyuru listesi ekrana gelir.



Şekil 7. 6 Duyuruların Listelenmesi

Herhangi bir duyuruya ait detaylı bilgileri görmek için o duyuru seçilir. Seçim sonrası şekil 7.7’de gösterilen duyuruya ait gönderen birim, konu, tür, yer, tarih ve etiket bilgileri listelenir.



Şekil 7. 7 Duyuru detay ekranları

Şekil 7.7’de gösterilen duyuru detay ekranında kullanıcının etiket ekleme yetkisi varsa şekil 7.7’de soldaki ekran görüntüsünde olduğu gibi ekranın sağ alt köşesinde “Etiket Ekle” seçeneği görülmektedir. Kullanıcının etiket ekleme yetkisi yoksa şekil 7.7’de sağdaki ekran görüntüsünde olduğu gibi bu seçenek gösterilmemektedir. Etiket ekleme yetkisi sadece Ege Üniversitesinde görevli öğretim üyelerine verilmiştir.

Etiket ekleme yetkisi olan kullanıcı, ekranın sağ alt köşesindeki “Etiket Ekle” seçeneğini seçtiğinde şekil 7.8’de gösterilen ekrandan etiket girişi yapabilmektedir. Kullanıcı bu menü aracılığıyla duyuruya eklemek istediği etiketi girerek “Kaydet”i seçtiğinde girdiği etiketi o duyuru için eklemiş olur. Birden fazla etiket eklemek için etiketler arasına virgül koyulması gerekmektedir.



Şekil 7. 8 Duyuruya etiket ekleme ekranı

Duyuru listesini kişiselleştirmek için duyurular listesi ekranda iken ‘Ayarlar’ tuşu seçilir. Burada ekrana gelen şekil 7.9’da gösterilen menü aracılığıyla duyuruları kişiselleştirmek yani kişisel tercihlere göre filtrelemek bazı duyuruları listenin başında görmek üzere önceliklendirmek için menüdeki “Profil Ayarları” seçeneği, ya da sistemdeki tüm duyurulara ait girilmiş olan etiket listesini görmek, etiketlere göre filtreleme yapmak içinse menüdeki “Etiketler” seçeneği seçilir.



Şekil 7. 9 Duyuru ayarları menu ekranı

Ayarlar menüsünde “Profil Ayarları” seçeneği seçildiğinde şekil 7.10 ve şekil 7.11’de gösterilen profil ayarları menüsü ekrana gelir.



Şekil 7. 10 Profil ayarları – Etkinlik seçimi

Şekil 7.10’da gösterilen profil ayarları menüsünden duyuruları kişiselleştiren, listelenmesini şekillendiren tercihler belirlenir. Kullanıcı duyuru listesinde görmek istediği etkinlik türlerini (Şekil 7.10), hangi birimlerin gönderdiği duyuruların listelenmesini istiyorsa o birimleri (Şekil 7.11) ve listelenecek duyuruların zaman aralığını (Şekil 7.11) seçerek kendi kişisel tercihlerine göre filtrelemeyi etkin hale getirir. Filtrelemenin yanı sıra kullanıcı yine bu menüden duyuru listesinin sıralandırılmasında önceliklendirme yapabilir. Bu menüden gireceği anahtar kelimelerle (Şekil 7.11), o anahtar kelimeleri içeren duyuruların listenin en başında yer almasını sağlar. Kullanıcının bu menüde yaptığı seçimleri aktif hale getirmesi için menüden ‘Kaydet’ opsiyonu ile çıkış yapması gerekmektedir.



Şekil 7. 11 Profil ayarları – Birim/Tarih aralığı seçimi ve Anahtar kelime girişi

Duyuru listesinin kişiselleştirilmesini sağlayan ayarlar menüsünden “Etiketler” seçeneği seçildiğinde, şekil 7.12’de gösterildiği üzere sistemdeki tüm duyurulara ait girilmiş olan etiket listesi ekrana getirilir.



Şekil 7. 12 Etiket Listesi

Etiket listesi ekranda iken herhangi bir etikete göre filtreleme yapmak için, o etiket üzerinde iken “Listele” seçeneği seçilmelidir. Bu durumda şekil 7.13’de gösterildiği üzere o etiketin tanımlandığı duyuru ya da duyurular listelenmektedir.



Şekil 7. 13 Etikete göre filtrelenen duyurular listesi

Şekil 7.13’de gösterilen etiket filtre listesi üzerinden duyurulara ait detay bilgilerinin görüntülenmesi de mümkündür. Şekil 7.14’de duyuru detay bilgilerinin görüntülediği ekran gösterilmektedir.



Şekil 7. 14 Filtrelenen duyurunun detay ekranı

8. SONUÇ

Günümüzde yaygın bir şekilde kullanılan mevcut uygulamalar, gelecekte daha da fazla sayıdaki kablosuz servislerle zenginleşecektir. Bu servislerin etkin bir şekilde kullanımları uygun kişiselleştirmeler ve aktif servis davranışlarıyla başarılabacaktır (Wagner, 2009).

Farklı platformlar üzerinde çalışan uygulamalara olan talebin ve mobil platform çeşitliliğinin her geçen gün artması yazılım dünyasında yeni bir arayış doğurdu. Bunun sonucunda platform bağımsız hizmet verebilen, platformlar arası geçişi kolaylaştıran web servisleri ortaya çıktı.

Web servisler işletim sistemi ve programlama dilinden bağımsızdırlar ve bu sayede çok esnek bir yapıya sahiptirler. Farklı platform ve sistemlerde geliştirilen uygulamalar birbirlerinden kolaylıkla çağırılabilirler. Teknolojisi halen gelişmekte olan bu sistemin sağladığı imkânlar sayesinde kullanım alanları da yaygınlaşmaktadır (Altınova H. vd, 2008).

İnternet ve mobil cihazların kullanımlarının artmasıyla birlikte, web üzerinde karmaşık işleri gerçekleştirmek günlük yaşantımızın önemli bir parçası haline geldi. Bununla birlikte mobil platformların kullanılabilirliğini artırmak için uygulamaların kullanıcıyı tanıyıp ona özel hizmet sunması kaçınılmaz olmaktadır. Bu sebeple karmaşık servis önerilerinden küçük istemci cihazları üzerinde bile tam olarak faydalanabilmek ve her geçen gün artan uygulamaları yönetmek için kişiselleştirmenin güçlü kavramları gereklidir. Ancak kişiselleştirmenin çok dikkatli bir şekilde yapılması gerekmektedir. Kullanımı kolaylaştırmak için yapılan bazı düzenlemeler bazen kullanımı olduğundan da daha karmaşık ve zor hale getirebilir.

Bugün bu çok konuşulan ve takip edilen konulara paralel olarak hazırlanan tez çalışması daha sonra yapılacak yeni çalışmalara yol gösterici olacaktır. Yine bu çalışmadan yola çıkılarak semantik, yapay zekâ ve istatistiksel teknikler kullanarak kullanıcı isteklerinin daha etkin ve daha doğru belirlendiği kişiselleştirme uygulamaları gerçekleştirilebilir.

KAYNAKLAR DİZİNİ

- Altınova, H., Yalçın, Y. ve Aslan, S.**, 2008, “Web Servis Tabanlı Geliştirilen Mobil Uygulamalar: Odtü Mobil Öğrenci İşleri Bilgi Sistemi (MOİBS)”
- Baraglia, R. and Silvestri, F.**, 2007, “Dynamic Personalization of Web Sites Without User Intervention”
- Bhalla, N. and Kazerooni, S.**, 2007, “Web Services Vulnerabilities: A white paper outlining the application-level threats to web services”
- Choi, O. and Han, S.**, 2008, “Personalization of Rule-based Web Services”
- Doğdu, E.**, “Web Servisleri ve Servis-Tabanlı Yazılım Mimarisi”, <http://www.bil.etu.edu.tr/web-servisleri-ve-servis-tabanlı-yazılım-mimarisi>, (Erişim tarihi: 20 Ocak 2010)
- Dönmez, O., Uğur, A. ve Önal, A.**, 2007, “CLDC Konfigürasyonuna sahip mobil cihazlarda 3B grafiklerin görselleştirilmesi”
- Ho, S. Y.**, 2009, “Opportunities and Challenges of Mobile Personalization: An Exploratory Study”
- Jaques, D.**, 2008, “Up Close and Personal with Your Customer”
- Knudsen, J. and Li, S.**, 2005 “Beginning J2ME: From Novice to Professional, Third Edition”, Apress, United States by Springer - Verlag New York, 456p
- Küçüksille, E., Kutlu, A. ve Taşdelen, K.**, “XML Web Servisleri ve Visual Studio .NET ile Geliştirilen Bir Mobil Uygulamada Kullanımı” ab.org.tr/ab05/tammetin/92.doc (Erişim tarihi: 18 Aralık 2009)
- Mobilpro**, “J2ME Temelleri”, http://www.mobilpro.com/makaleler/arsiv/J2ME_temelleri/J2ME_temelleri.htm (2003), (Erişim tarihi: 28 Aralık 2009)
- Nagel, C., Evjen, B., Glynn, J., Skinner, M., and Watson, K.**, 2008, “Professional C# 2008”, Wiley Publishing, Inc., Indianapolis, Indiana, 1782p
- Nayyeri, K.**, 2008, “Professional Visual Studio® 2008 Extensibility”, Wiley Publishing, Inc., Indianapolis, Indiana, 520p
- Pazzani, M. J.**, “Personalization for the Mobile Web: A Position Paper”, www9.org/w2-mobileweb/PazzaniMobileWeb.html, (Erişim tarihi: 25 Aralık 2009)

KAYNAKLAR DİZİNİ (Devamı)

- Şen, K.**, “Web Servisleri”, www.emo.org.tr/ekler/9cd21a0e0b7d5f0_ek.doc?dergi=428 (Erişim tarihi: 20 Ocak 2010)
- Şen, Ş.**, 2008, “Web Part ve AJAX Teknolojileri ile Kişiselleştirilebilir Portal Uygulaması”
- Şenyurt, B. S.**, “Asp.Net 2.0 Profile API ile Personalization (Kişiselleştirme)”, <http://www.csharpnedir.com/articles/read/?id=644> (2006), (Erişim tarihi: 31 Aralık 2009)
- Şenyurt, B. S.**, 2008, “Client Application Services (İstemci Uygulama Servisleri)”, [http://www.csharpnedir.com/articles/read/?filter=edited&id=887&title=Client Application Services \(İstemci Uygulama Servisleri\)](http://www.csharpnedir.com/articles/read/?filter=edited&id=887&title=Client Application Services (İstemci Uygulama Servisleri)), (Erişim tarihi: 27 Aralık 2009)
- Şenyurt, B. S.**, “C# Web Servislerine Giriş”, <http://csharpnedir.com/articles/read/?filter=&author=&cat=xml&id=49&title=C>, (Erişim tarihi: 2 Ocak 2010)
- Thangarathinam, T.**, 2006 “Professional ASP.NET 2.0 XML”, Wiley Publishing, Inc., Indianapolis, 566p
- Trivedi, R.**, 2003, “Web Services Tutorial: Understanding XML and XML Schema - Part 1”
- Wagner, M., Balke, W., T., Hirschfeld, R. and Kellerer, W.**, “A Roadmap to Advanced Personalization of Mobile Services”, www.l3s.de/apis/paper/coopis02i.pdf, (Erişim tarihi: 21 Aralık 2009)
- Workcube**, “Personalization (Kişiselleştirme) Abartılıyor mu?”, <http://www.workcube.com/erp-crm-programlarında-personalization-kisisellestirme-customization-ozellestirme-nedir>, (Erişim tarihi: 13 Ocak 2010)

ÖZGEÇMİŞ**Kişisel Bilgiler**

Adı Soyadı	Ülkü Beydağ İnce
Doğum Tarihi	03.01.1982
Uyruğu	T.C

Eğitim

Yüksek Lisans	2004-... Ege Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Anabilim Dalı
Lisans	2000–2004 Ege Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
Lise	1993–2000 Hulusi Uçaçelik Anadolu Lisesi

İş Deneyimi

Yazılım Mühendisi	Aralık 2008 – ... Vestel Elektronik A.Ş. - Manisa, Türkiye
Yazılım Mühendisi	Ağustos 2004 – Aralık 2008 Cabot Communications – Urla, Türkiye