

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AJAX TEKNOLOJİSİ KULLANARAK ÇEVİRİMİÇİ
KATALOG TASARIMI**

Zeynep ÖZER AYDOĞMUŞ

Tez Yöneticisi:
Prof. Dr. Asaf VAROL

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2009

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AJAX TEKNOLOJİSİ KULLANARAK ÇEVİRİMİÇİ
KATALOG TASARIMI**

Zeynep ÖZER AYDOĞMUŞ

Tez Yöneticisi:
Prof. Dr. Asaf VAROL

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2009

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AJAX TEKNOLOJİSİ KULLANARAK ÇEVİRİMİÇİ
KATALOG TASARIMI**

Zeynep ÖZER AYDOĞMUŞ

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

Bu tez,tarihinde aşağıda belirtilen jüri tarafından oybirliği/oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

Danışman: Prof. Dr. Asaf VAROL

Üye:

Üye:

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez çalışması süresince desteklerini esirgemeyen Danışmanım Sayın Prof. Dr. Asaf VAROL'a, Prof. Dr. Z. Hakan AKPOLAT'a, çalışmamın her aşamasında sevgi ve güvenlerini her zaman hissettiğim aileme, eşim Fatih AYDOĞMUŞ'a ve özellikle kardeşim Yağmur ÖZER'e teşekkürlerimi ve şükranlarımı sunarım.

Zeynep ÖZER AYDOĞMUŞ

İÇİNDEKİLER

TEŞEKKÜR

İÇİNDEKİLER..... I

ŞEKİLLER LİSTESİIV

TABLO LİSTESİ VI

KISALTMALAR LİSTESİ VII

ÖZETIX

ABSTRACT X

1 GİRİŞ..... 1

1.1 Tez Çalışmasının Amacı 1

1.2 Tez Çalışmasının Kapsamı..... 1

2 WEB 2.0 AKIMI 2

2.1 Web 2.0 Platformunda Öne Çıkan Popüler Uygulamalar 4

2.1.1 Double Click - Google AdSense 4

2.1.2 Flickr 5

2.1.3 BitTorrent..... 5

2.1.4 Sosyal Ağ Siteleri..... 8

2.1.4.1 Facebook 8

2.1.5 Web Günlükler 11

2.1.6 Wiki..... 11

2.1.7 İçerik Yönetim Sistemleri 13

2.1.8 Etiketleme (Folksonomi)..... 14

2.2 Web 2.0’da Site Tasarımında Dikkat Çeken Özellikler 14

2.2.1 Logolar 14

2.2.2 Butonlar 15

2.2.3 Arka Planlar..... 15

2.2.4 Çerçeveler..... 16

2.2.5 Menüler 16

3 AJAX..... 17

3.1 Google Suggest 18

3.2 Google Maps 18

3.3 Neden AJAX..... 19

3.4 AJAX Metodolojisi 23

3.4.1 HTTP Requests Nesnesi 24

3.4.2	HTTP Request Nesnesinin Özellikleri ve Yöntemleri	25
3.4.3	HTTP Response Nesnesi	26
3.5	AJAX İletişim Teknikleri.....	27
3.5.1	XMLHttpRequests (XHR) Nesnesi.....	27
3.6	AJAX'ı Oluşturan Bileşenler	34
3.6.1	Eşzamanlı Olmayan İletişim	34
3.6.2	Eşzamanlı İletişim	35
3.6.3	Javascript.....	36
3.7	Basit Bir AJAX Örneği	49
3.8	XML.....	51
3.8.1	Örnek XML Belgesi Oluşturma	53
3.8.2	XML Dokümanının Yapısı.....	55
3.8.3	XML'nin Söz Dizim Kuralları	67
3.8.4	XML Dokümanlarının İşleyişi ve Sunumu	69
3.9	AJAX Prensipleri	69
3.10	AJAX Kütüphaneleri ve Araçları.....	70
3.10.1	Prototype	70
3.10.2	AJAX Gear Toolkit	71
3.10.3	Asp.Net Ajax.....	71
3.10.4	Dojo Toolkit	71
3.10.5	MochiKit	71
3.10.6	PHPLiveX	71
3.10.7	Sardalya (Sarmal Dinamik Altyapısı)	72
3.10.8	Sajax	72
3.10.9	Solvent.....	72
3.10.10	AjaxAC.....	72
3.10.11	Xajax	72
3.10.12	Scriptaculous	73
3.11	AJAX Uygulamalarında Kullanılan Diğer Teknolojiler	73
3.11.1	HTML/XHTML	73
3.11.2	CSS.....	73
3.11.3	XSLT.....	74
4	FLEX.....	75
4.1	Adobe® Flex® 3 Kurulumu İçin Gereken Minimum Sistem Gereksinimleri.....	76
4.1.1	Windows İçin Flex Builder 3 (Standart ve Profesyonel).....	76

4.1.2	Macintosh İçin Flex Builder 3 (Standart ve Profesyonel)	77
5	ÇEVİRİMİÇİ KATALOG TASARIMI: Fırat Üniversitesi Örneği.....	78
6	SONUÇ	88
6.1	Öneriler.....	89
	KAYNAKLAR.....	90
	ÖZGEÇMİŞ.....	95
EK		96

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1 Web 2.0 kavramı	2
Şekil 2.2 Web 2.0 akımında öne çıkan uygulamalar	3
Şekil 2.3 Doubleclick ve Google Adsense sitelerinin logoları	4
Şekil 2.4 Flickr sitesinin logosu	5
Şekil 2.5 İçerik dağıtım sitelerinde bittorrent site logosu	5
Şekil 2.6 µTorrent programının anasayfası	6
Şekil 2.7 µTorrent programının arayüzü	7
Şekil 2.8 Facebook kullanıcı sayısının aylara göre dağılım grafiği	9
Şekil 2.9 Aylara göre facebook ve myspace site analizi	10
Şekil 2.10 Vikipedi özgür ansiklopedi sitesinin logosu	12
Şekil 2.11 Örnek etiket bulutu	14
Şekil 2.12 Yansımali logo örneği	15
Şekil 2.13 Web 2.0’da buton örnekleri	15
Şekil 2.14 Web 2.0’da arka plan örnekleri	15
Şekil 2.15 Web 2.0’da çerçeve yapısına örnek	16
Şekil 2.16 Web 2.0’da menü örneği	16
Şekil 3.1 Jesse James Garrett’in makale görüntüsü	17
Şekil 3.2 Google Suggest ekran görüntüsü	18
Şekil 3.3 Google Maps ekran görüntüsü	18
Şekil 3.4 Geleneksel sunucularda istek/cevap modeli	20
Şekil 3.5 AJAX’ın asenkron istek/cevap modeli	20
Şekil 3.6 Klasik Web uygulamalarında tıkla-bekle döngüsü	21
Şekil 3.7 Tıkla-Tıkla interaktif AJAX uygulaması	21
Şekil 3.8 Geleneksel Web iletişim modeli ile AJAX iletişim modeli	21
Şekil 3.9 Web uygulamasındaki cevap akışı	22
Şekil 3.10 AJAX uygulamalarındaki istek akışı	22
Şekil 3.11 Normal Web ile AJAX uygulamalarının karşılaştırması	23
Şekil 3.12 Geleneksel istek/cevap (request/response) yöntemi	23
Şekil 3.13 Standart AJAX etkileşimi	32
Şekil 3.14 Eşzamanlı olmayan iletişim modeli	35
Şekil 3.15 Eşzamanlı iletişim modeli	36
Şekil 3.16 Javascript nesneleri	44

Şekil 3.17 DOM ağaç yapısı	46
Şekil 3.18 AJAX örneği	49
Şekil 3.19 sonuc.html sayfasının ekran çıktısı	51
Şekil 3.20 deneme.xml dosyasının ağaç yapısı	53
Şekil 3.21 XML belgesi	55
Şekil 3.22 Örnek bir XML doküman yapısı	56
Şekil 3.23 deneme.xsd'nin ekran görüntüsü	65
Şekil 4.1 Flex uygulamalarının işlem akış şeması	75
Şekil 4.2 Flex arayüzü.....	76
Şekil 5.1 Kodlama algoritması	79
Şekil 5.2 Kataloğun ilk sayfasının ekran görüntüsü.....	81
Şekil 5.3 Kataloğun 2/3 sayfa görüntüsü	82
Şekil 5.4 Kataloğun sayfa çevirme efektli görüntüsü.....	83
Şekil 5.5 Kataloğun navigasyon çubuğu	84
Şekil 5.6 Tüm Sayfalar düğmesi tıklandığındaki ekran görüntüsü	85
Şekil 5.7 Zoom özelliği	86

TABLO LİSTESİ

Tablo 2.1 Facebook ve myspace'in büyüme tablosu	10
Tablo 3.1 Standart XHR nesnesinin özellikleri	30
Tablo 3.2 Standart XHR nesnesinin fonksiyonları	31
Tablo 3.3 Değişken türleri	41
Tablo 3.4 İşleçler ve görevleri	41
Tablo 3.5 İkili işleçler.....	42
Tablo 3.6 DTD karakterleri ve açıklamaları	61
Tablo 3.7 DTD içindeki niteliklerin değer ve açıklamaları	62
Tablo 3.8 Varsayılan değerler ve açıklamaları	62
Tablo 3.9 XML Belgelerinde kullanılan varlık bildirimleri	67

KISALTMALAR LİSTESİ

AJAX	:	Asenkron JavaScript ve XML
API	:	Application Programming Interface (Uygulama Program Arayüzü)
ASP	:	Active Server Pages (Aktif Sunucu Sayfaları)
CDNs	:	Content Distributed Networks (İçerik Dağıtıcı Ağlar)
CMS	:	Content Management System (İçerik Yönetim Sistemi)
CGI	:	Common Gateway Interface (Genel Geçiş Arayüzü)
CSS	:	Cascading Style Sheets (Basamaklı Stil Sayfaları)
DFA	:	DART for Advertisers
DOM	:	Document Object Model (Döküman Nesne Modeli)
DTD	:	The Document Type Definition (Doküman Tipi Tanımlama)
DWR	:	Direct Web Remoting (Direkt Web Erişimi)
ECMA	:	European Computer Manufacturers Association (Avrupa Bilgisayar Üreticileri Birliği)
GÖBL	:	GNU Özgür Belgeleme Lisansı
HTML	:	Hyper Text Markup Language (Yardımlı Metin İşaretleme Dili)
HTTP	:	Hypertext Transfer Protocol (Yardımlı Metin Aktarım Protokolü)
IDE	:	Integrated Development Environment (Tümleşik Geliştirme Ortamı)
JSP	:	Java Server Pages (Java sunucu Sayfalar)
LAN	:	Local Area Network (Yerel Alan Ağı)
MIYS	:	E-ticaret ve Müşteri İlişkileri Yönetim Sistemleri
MSN	:	Microsoft Network
OOP	:	Object Oriented Programming (Nesne Yönelimli Programlama)
PHP	:	Hypertext Preprocessor (Yardımlı Metin Önileme)
PIs	:	Processing Instructions (İşlem Açıklamaları)
RIA	:	Rich Internet Applications (Zengin İnternet Uygulamaları)

RPC	:	Remote Procedure Call (Uzaktan Yordam Çağırma)
SAX	:	Simple API for XML (XML için Basit API)
WIKI	:	What I Know is (Ne Biliyorsam)
W3C	:	World Wide Web Consortium (Dünya Çapında Ağ Birliği)
XML	:	eXtensible Markup Language (Geliştirilebilir İşaretleme Dili)
XHR	:	XMLHttpRequest
XHTML	:	Extensible HyperText Markup Language (Geliştirilebilir Yardımlı Metin İşaretleme Dili)
XSD	:	eXtensible Schema Document(Genişletilebilir Şema Dokümanı)
XSL	:	Extensible Stylesheet Language (Geliştirilebilir Biçimlendirme Dili)
XSLT	:	Extensible Stylesheet Language Transformations (Geliştirilebilir Biçimlendirme Dil Dönüşümleri)

ÖZET

Yüksek Lisans Tezi

AJAX TEKNOLOJİSİ KULLANARAK ÇEVİRİMİÇİ KATALOG TASARIMI

Zeynep ÖZER AYDOĞMUŞ

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik-Bilgisayar Eğitimi Anabilim Dalı

2009, Sayfa: 107

Gerçekleştirilmiş olan bu uygulama kapsamında Flex programı ve Web 2.0'ın çekirdeğini oluşturan AJAX (Asynchronous JavaScript and XML) teknolojisi kullanılarak Fırat Üniversitesini tanıtan, kolaylıkla güncellenebilen Zengin İnternet Uygulamalarına (RIA- Rich Internet Applications) örnek bir çevrimiçi katalog tasarımı ve gerçekleştirimi yapılmıştır. Projenin gerçekleştirim aşamalarında kullanılan teknolojilerin ve izlenen yöntemlerin anlatımlarını içeren bir yol haritası oluşturulmuştur.

Anahtar sözcükler: AJAX, RIA, Web 2.0, çevrimiçi katalog, e-dergi.

ABSTRACT

Master Thesis

ONLINE CATALOG DESIGN USING AJAX TECHNOLOGY

Zeynep ÖZER AYDOĞMUŞ

Fırat University

Graduate School of Natural and Applied Sciences

Department of Electronic and Computer Education

2009, Page: 107

Embracing in this comparison, Fırat University is introduced by using Flex program and AJAX (Asynchronous JavaScript and XML) technology which forms Web 2.0. Also, RIA (Rich Internet Applications) is easily renowned which is instance for online catalog design. A road map including realization of the project and tracing methods are illustrated.

Keywords: AJAX, RIA, Web 2.0, online catalog, e-journal.

1 GİRİŞ

Web'in gelişim sürecinde çeşitli yöntemler kullanılır. Bu yöntemlerin farklı amaçlara hizmet etmeye başlamasıyla Web'te akımlar meydana gelmiştir. Bu akımlardan ilki Web 1.0 akımı olarak adlandırılır. Bu akımda kullanıcıları bilgilendirmeye yönelik içerik sunumu yapılmaktadır [1].

Web 2.0 akımı ise, internet sitelerinin bilgi barınağı olmasından işleyen platformlara taşınması olarak tanımlanabilir. Web 2.0, RIA (Rich Internet Applications- Zengin İnternet Uygulamaları) işlemleri, internet sunucu mimarisi temeline dayanmaktadır. RIA da beraberinde kullandığı AJAX, Flex gibi teknolojiler üzerine kuruludur [1].

Bazı yeni teknolojilerle değişen World Wide Web dünyası Web 2.0, RIA, AJAX (Asynchronous JavaScript and XML- Eş zamanlı olmayan JavaScript ve XML) terimleriyle izah edilmektedir.

1.1 Tez Çalışmasının Amacı

Bu tez çalışmasıyla Web 2.0'in çekirdeğini oluşturan AJAX (Asynchronous JavaScript and XML) teknolojisi ve Flex programı kullanılarak Fırat Üniversitesini tanıtan, kolaylıkla güncellenebilen Zengin İnternet Uygulamalarına (RIA-Rich Internet Applications) örnek bir çevrimiçi (online) katalog tasarımı yapılması hedeflenmiştir. Projenin gerçekleştirim aşamalarında kullanılan teknolojilerin ve izlenen yöntemlerin anlatımlarını içeren bir yol haritası oluşturulması amaçlanmıştır. Bu tez çalışması sonucunda geliştirilen yazılımla kolay güncellenebilen katalog hazırlanmasına katkı sağlanması amaçlanmıştır.

1.2 Tez Çalışmasının Kapsamı

Bu tez çalışması 6 bölümden oluşmuştur. 1. Bölüm girişi oluşturmaktadır. 2.Bölümde, Web 2.0 akımı incelenecektir. 3. Bölümde, AJAX teknolojisi açıklanacaktır. 4. Bölümde, yazılım gerçekleştirme aracı olarak kullanılan Flex tanıtılacaktır. 5. Bölümde gerçekleştirilmiş çevrimiçi katalog hakkında bilgi verilecektir. 6. Bölümde, çalışmanın sonuçları ve geleceğe yönelik öneriler bulunmaktadır.

2 WEB 2.0 AKIMI

Çoğu teknolojilerde olduğu gibi, yenilik arayışı ihtiyaçtan doğar. Yeni Web akımlarının doğuşu da 2001 yılındaki tüm dünyayı etkileyen kriz sonrasında iflas eden şirketlerin yeniden ayağa kalkma çabalarının bir sonucu olarak ortaya çıkmıştır [2, 3]. Web 2.0 akımı, 2005'te ilk kez O'Reilly ve MediaLive International arasında gerçekleşen bir beyin fırtınası sırasında üretilmiş bir kavramdır [3]. Web 2.0 kavramı World Wide Web teknolojisinde kullanılan değişen ve gelişen eğilimlere verilen bir ad olarak da tanımlanabilmektedir.



Şekil 2.1 Web 2.0 kavramı [4].

Web 2.0 kavramı Şekil 2.1'de gösterildiği üzere Web bir uygulama platformu ve sosyal etkileşimli olma özelliklerini barındırır. Çevrimiçi hayatın başlangıcı ve devamını temsil eder. Sayısal ortamda kendini ifade etme imkânı sunar. Kullanıcılar (oku-yaz özelliği ile) Web'te katılımcı kimlik kazanır.

Aslında Web 2.0 ortamını 1964 yılında McLuhan "Global Köy" terimini kullanarak ortaya atmıştır. McLuhan, Global Köyü gelecekte ileri teknolojiye geçilmesiyle ve toplumlar arasında sınırların kalkmasıyla, dünyanın her köşesinde teknolojiyi kullanan herkesin, coğrafi sınırlar ve sınırlamalar olmaksızın diledikleri kişiyle iletişim kurabilecekleri, haberleşebilecekleri ve paylaşımında bulunabilecekleri bir ortam olarak tanımlamıştır [5].

Mc Luhan'nın tanımından günümüze kadar geçen süreç içinde kullanıcı ihtiyaçları da değişim ve gelişim göstermiştir. Bu ihtiyaçları karşılamak için genel olarak internet uygulamalarının daha zengin bir arayüz ile sunulmasını sağlayan uygulamalar Zengin İnternet Uygulamaları adı altında kısa zaman önce hayatımıza girmiştir.

RIA uygulamaları sıradan HTML web uygulamaları ile standart masaüstü uygulamaları arasındaki boşluğu dolduran bir geliştirme platformudur. Web arayüzünde basit HTML ile

sınırlı kalmadan masaüstündeki zengin görsel öğeleri program niteliğinde kullanabilecek her türlü araç RIA konseptini uygulamaktadır [6].

Kullanıcılar artık içerik üreticilerin sınırlarını kaldırarak kullanıcının yanı sıra katılımcı olarak ta nitelendirilerek etkileşimli bir rol üstlenmişlerdir. İnternetin, bu yeni gereksinimlere cevap vermesiyle ortaya çıkan Web 2.0 dönüşümü hem kullanıcı sayısında artışa neden olmuştur hem de kullanıcıların içerikleri yani mesajları oluşturduğu ve paylaştığı büyük bir güce dönüşmüştür. Bu gücün önemini çabuk anlayan kuruluşlar mimarilerine AJAX'ı da dâhil ederek önce arama motorlarını; Google'ı, Yahoo'yu daha sonra MSN'yi, Hotmail'i, blogları (Myspace, Blogger.com, Blogcu.com gibi), Youtube'u, Wikileri ve sosyal ağ sitelerini (Facebook, Twitter, Orkut, vb.) keşfetmişlerdir. Bu keşiflerin sonucunda McLuhan'ın Global Köy'ünün 21. Yüzyılda tam olarak gerçekleştiği söylenebilir. Bu interaktif siteler sayesinde aynı anda milyarlarca kişinin ırk, dil, din, yaşam tarzı, eğitim, yaş, cinsiyet, statü gözetmeksizin eş zamanlı paylaşımında bulunup kendi çevreleri ve başka çevrelerle iletişime geçebilmektedirler. Şekil.2.2'de gösterilen şirketler Web 2.0 akımında öne çıkan sitelerin özellikle çevrimiçi faaliyet gösteren reklâm şirketleri, pazarlama ve ticari şirketler perspektifinden bakıldığında da, önemli bir potansiyele sahip kitle iletişim araçları olarak görülür [7].



Şekil 2.2 Web 2.0 akımında öne çıkan uygulamalar [8].

Kitle iletişim aracı olarak kullanılabilen bu uygulamalar için birçok geliştirme ortamı mevcuttur. Adobe Flex, AJAX, Java Applet, Adobe Flash ve Microsoft Silverlight uygulamaları en çok bilinen RIA uygulamalarıdır. Aşağıda da bu uygulamalarla oluşturulmuş Web 2.0 platformunda öne çıkan popüler uygulama örnekleri incelenecektir [7].

2.1 Web 2.0 Platformunda Öne Çıkan Popüler Uygulamalar

İkinci nesil internet hizmeti olarak literatüre giren Web 2.0 akımıyla internet sitelerinin çehresi değişmiştir. İnternet siteleri artık masaüstü uygulamaları ile internet uygulamaları arasında bir köprü görevi üstlenmiştir. Aşağıda bu akımın popüler uygulamalarından örnekler bulunmaktadır.

2.1.1 Double Click - Google AdSense

Google firmasına ait iki uygulama da reklâm verenlerin ve yayıncıların web üzerinden reklâm hizmeti sunmasını sağlayan sitelerdir. Her reklâmcının hedefi tüketicilerin isteklerine uygun çözümler sunmaktır. Bu çözüm ise Web 2.0'la reklâmların profilini değiştirmiştir. Artık istenmeyen, kendiliğinden açılan reklâm pencerelerinin yerini, kullanıcının bilgisayarına yerleştirilmiş çerezlerden çektiği bilgilere dayanarak kullanıcının ilgisini çekebileceğini tahmin ettiği rahatsız edici olmayan metin ya da resim tabanlı reklâmlar almıştır.

DoubleClick'in diğer reklâm firmalarından ayrılan yönü, reklâmcılar için DART önerisiyle kısaca DFA (DART for Advertisers) ile reklâm yönetimi ve servis çözümlerinde kullanıcılarına dijital pazarlama programlarının tümünü kapsayan yönetimi kolay, daha iyi bir arayüze sahip, hızlı, geniş bir arama sistemine sahip, değeri artan sitelerin trafik akışını belirleyen bir çözüm olmuştur. DoubleClick daha çok grafiksel reklâmlar yayınlamaktadır. Önceleri Web 1.0 uygulamaları arasında sayılan bu site, Web 2.0 dönüşümünü başarıyla gerçekleştirmiştir [9].

Google AdSense ise, web sitesi yayıncılarının site içeriğiyle alakalı, dinamik ve göze batmayan Google reklâmlarını web sitelerinde göstererek kullanıcıların bu reklâmlara tıklamalarıyla para kazanmalarını sağlamanın hızlı ve kolay bir yoludur. Ücretsiz olan bu program sayesinde her tıklama ve gösterim için site sahiplerine ödeme yapılır [10]. Bu uygulamalar sayesinde Web'de daha akıllı bir sisteme doğru yönelim başlamıştır.



Şekil 2.3 Doubleclick ve Google AdSense sitelerinin logoları [11, 12].

2.1.2 Flickr

Yahoo'nun sunduğu bir hizmet olan ücretsiz fotoğraf arama ve paylaşım sitesi Flickr, aynı zamanda web servisi ve çevrimiçi topluluk platformudur. Web 2.0'ın ilk uygulamalarından biridir. Fotoğraf paylaşımında en popüler sitelerin başında gelmektedir. İçeriğinde yaklaşık olarak 3 milyarı aşan fotoğraf bulunmaktadır. Yenilikçi çevrimiçi topluluk araçları fotoğrafları etiketlemeye ve sınıflandırmaya yardımcı olur. Bu da popüleritesindeki ciddi artışın nedenlerinden biridir.

Flickr'a imaj yükleyebilmek için Yahoo'dan bir e-mail adresiyle sisteme giriş yapılmalıdır. Paylaşılacak istenen fotoğraflar ücretli üyelik olmadan her imaj için 5 Mb'lık sınırlamayla ayda toplam 100 Mb'a kadar yükleme yapılabilir. Sadece fotoğraf eklemenin dışında imajlar üzerinde düzenlemeler yapılabilir. Belirlenen alanlara notlar eklenebilir. Beğenilen fotoğraflara yorumlar eklenebilir. Çeşitli gruplara üye olunup iletişim kurulabilir. Ayrıca API'leri sayesinde XML-RPC (Remote Procedure Call- Uzaktan Yordam Çağırma) arabirimi kullanarak flickr sitesindeki fotoğraflarla ilgili bilgileri prosedürler sayesinde XML formatında alıp ayrıştırarak başka bir programda kullanılabilir. Flickr'daki blog, grup ve etiketleme alt yapısı bu API'lere dayanmaktadır [13].



Şekil 2.4 Flickr sitesinin logosu [14].

2.1.3 BitTorrent

BitTorrent, internet üzerinden yüksek hızlarla içeriğin dağıtıldığı dünya çapında bir standarttır. Ayrıca 150 milyondan fazla kullanıcıya sahip en son p2p (peer to peer) dosya paylaşım yöntemlerinden biridir. Diğer p2p paylaşım programlarından farkı: sabit olmayan bağımsız sunucu tanımlama dosyaları sayesinde sabit bir sunucuya ihtiyaç olmaksızın belirli bir protokol çerçevesinde indirme (download) yaparken yükleme (upload) yapılmasına da izin vererek paylaşımına devam edebilmesidir [15].

Python programlama diliyle yazılmış bir uygulamadır [15].

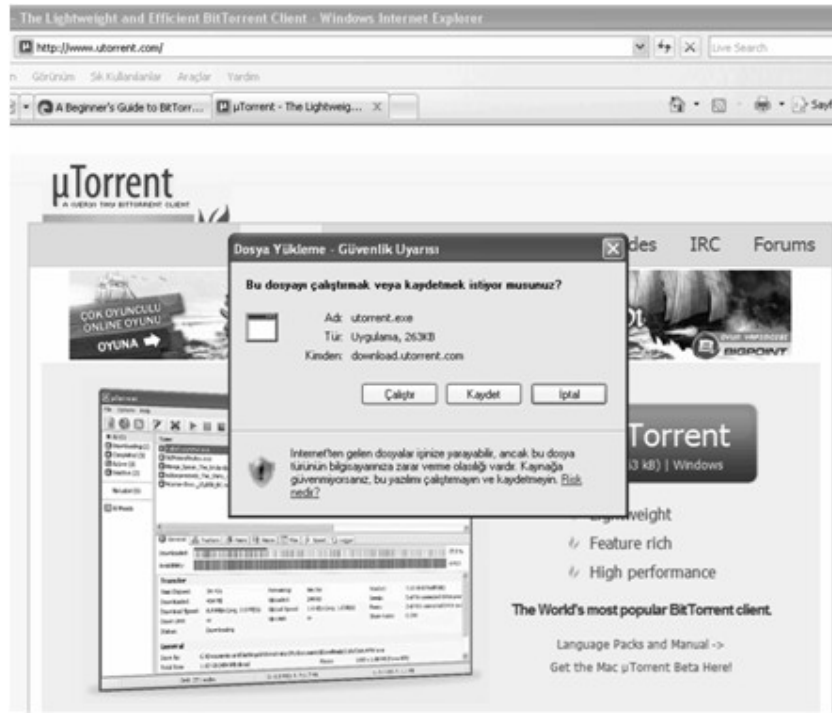


Şekil 2.5 İçerik dağıtım sitelerinde bittorrent site logosu [16].

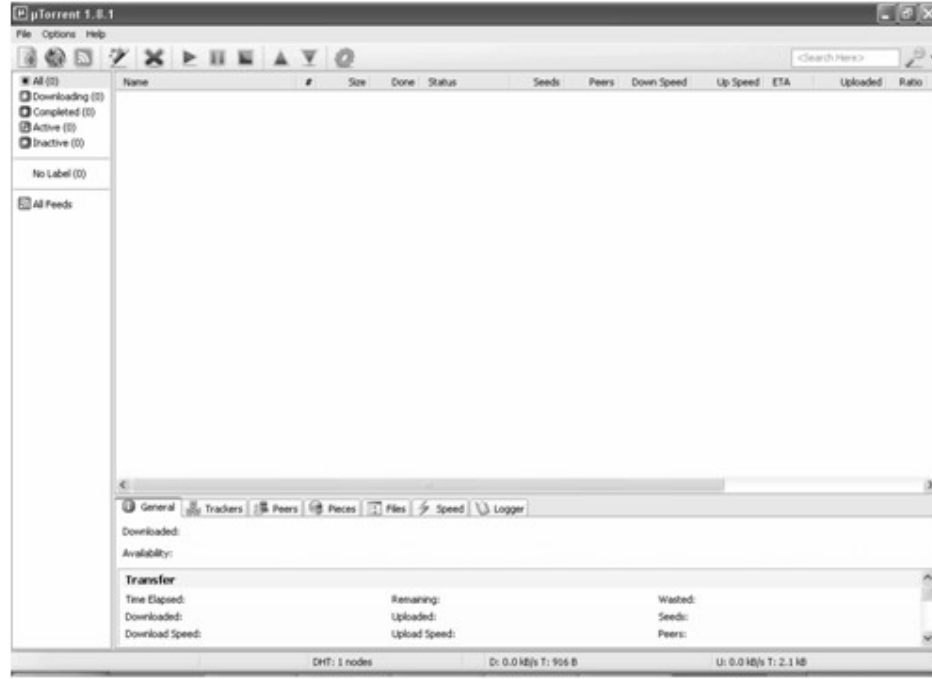
“Bit Seli” olarak da çevirebileceğimiz çok hızlı bir sistemdir. BitTorrent sayesinde oldukça büyük boyutlu videolara, ses ya da yazılım dosyalarına daha hızlı, daha ucuz ve daha kolay bir şekilde ulaşılabilmeyi sağlar.

Günümüzde İçerik Dağıtıcı Ağlar (Content Distributed Networks-CDNs) gün geçtikçe popülaritesini artırmaktadır. Bittorrent kullanımının 2004’te internet’te teknik standartlar ve yönetim konusunda yetkili organ olan W3C (World Wide Web Consortium - Dünya Çapında Ağ Birliği) tarafından yapılan bir çalışmaya göre P2P bağlantıların toplamda %60’ını, tüm internet trafiğinin ise %30’unu açacağı tahmin edilmektedir.

BitTorrent’le dosya transferi yapılırken, öncelikle bilgisayarda BitTorrent İstemci (Client) yüklü olmalıdır. İstemci yazılımlara örnek Flashget, Azureus, BitTorrent, µTorrent, ABC, LimeWire vb yazılımlar olabilir. Şekil 2.6 ve Şekil 2.7’de en popüler uygulamalardan µTorrent yazılımının kurulum ve arayüz örneği görülmektedir.



Şekil 2.6 µTorrent programının anasayfası.



Şekil 2.7 µTorrent programının arayüzü.

BitTorrent istemciler BitTorrent Protokol uygulamalarını yapar. Her istemci bu protokolü kullanarak dosyalarını kullanıcıların BitTorrent'in kullandığı protokole sahip dünyadaki tüm içeriklere hızlı ve güvenilir bir şekilde erişmesine izin verir. İstenilen içeriğin indirilebilmesi için gerekli tanımlama bilgilerinin barındığı “.torrent” uzantılı bir dosyaya ihtiyaç vardır. Bu torrent dosyalarını çeşitli duyuru sitelerinden ya da Web’den temin edildikten sonra, istemciler yukarıda bahsedilen herhangi bir torrent yazılımı kullanarak bu dosyayı açarlar. İndirme başladığı anda indirilmek istenen dosyayı paylaşan kullanıcıların oluşturduğu küçük ağa (swarm) dâhil olurlar. Bu küçük ağdaki her kullanıcı dosyanın indirdiği bölümünü diğer kullanıcılarla paylaşması gerekir. Tüm bu işlemleri de takipçi (tracker) isimli bir sunucu kontrol eder. Yaklaşık olarak 256 Kb büyüklüğündeki bu küçük boyutlu dosyalara indeksleyici (indexer) de denmektedir. Küçük bir ağ kurulması için tüm dosyaya sahip en az bir dağıtıcının (seeder) olması gerekir. Dağıtıcılar herhangi bir dosyayı yükleyen kaynaklardır. O dosyanın tamamını kendi bilgisayarına indiren her kullanıcı artık bir dağıtıcı olur. Fakat dağıtıcı olmak kullanıcıların insiyatifindedir. İndirme işlemi bittikten sonra o dosya paylaşımına açık tutulmazsa diğer kullanıcılar dağıtıcıdan bu dosyayı çekerek indirme işlemini hızlandıramazlar. Dağıtıcı sayısı ne kadar artarsa indirme işlemi o kadar hızlı olur. İndirme yapan kullanıcılara da grup üyeleri (peers) denilmektedir. Torrent dosyasını indiren kullanıcılar izleyicilere (tracker) dosya hakkında bilgi gönderirler, diğer kullanıcılar (peers) ise izleyicilere bağlanarak kimde hangi dosyanın hangi parçalarının olduğunu öğrenirler. İzleyiciler üzerinden kesinlikle dosya transferi

gerçekleşmez sadece kaynak paylaşılır. Hangi kullanıcı ne kadar dosya çekti ve ne kadar yükledi gibi istatistiksel bilgileri de barındırabilir [16, 17].

Sadece indirme işlemi yapıp kaynak olmayı istemeyen kişilere leecher (sülük) denir. Bu anlayışa sahip kişiler gruplar içinde hoş karşılanmazlar. Bir torrent dosyası paylaşımında kaynak (seed) sayısı sıfır ise yani dosyanın tamamına sahip kişiler paylaşımdan çekilmişlerse, eksik dosyaya sahip bütün üyeler (peer) birbirleri arasında veri alışverişi yaparak, eksik parçaları indirebilecekleri bir kaynak olmadığından, belli bir yüzdesi tamamlanmış dosyaya sahip olarak kalırlar. Bu durumda daha önce dosyayı indirmiş ve dosyanın tamamına sahip olan bir kullanıcı eksik parçaları göndermek için dosyayı yeniden paylaşımına açabilir (seed eder). Bu olaya reseed etme denir [18].

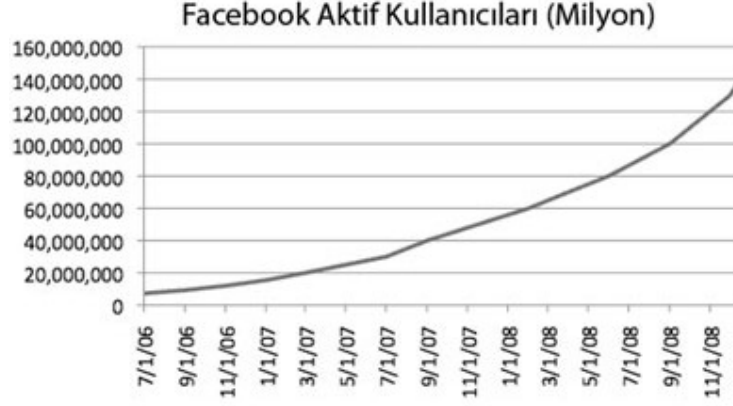
2.1.4 Sosyal Ağ Siteleri

Sosyal ağlar, bireyler arasındaki kişisel veya profesyonel ilişkilerin oluşturulduğu, düzenlendiği ağıdır [19]. Son yıllardaki bilgi teknolojilerindeki gelişmeler, sosyal ağların gerçek yaşamdakinden daha farklı bir şekilde yeniden yapılanmasını sağlamıştır. Herhangi bir şirketteki yerel ağ içinde personelin e-posta alıp-göndermesiyle oluşan etkinlik ve e-posta grupları ilk çevrimiçi sosyal ağ örneklerinden sayılabilir. Daha sonra hızla hayatımıza giren bloglar, wikiler ve sosyal ağ siteleri gibi sosyal işbirliği sağlayan teknolojilerin gelişmesi insanların iletişim kurdukları, bilgi, resim, müzik, video paylaştıkları ve birbirleriyle sürekli haberleştikleri çevrimiçi sanal toplulukların oluşmasını sağlamıştır. [20]. Bu toplulukların hızla yayılmasının temelinde kullanıcıların bireysel aktivitesine bağlı olarak duygu ve düşüncelerini, bilgiyi, güncel haberleri, tavsiyelerini, beğenilerini paylaşma fikri vardır. Aşağıda sosyal ağ sitelerinden en popüler olanlar içinde yer alan Facebook incelenecektir [7].

2.1.4.1 Facebook

En yaygın sosyal ağ sitesi Facebook Amerika Birleşik Devletleri'nde 4 Şubat 2004 tarihinde Harvard Üniversitesinde öğrenci Mark Zuckerberg tarafından kurulmuştur. Facebook, öncelikle Harvard öğrencileri için kurulmuştu. Daha sonra hızla tüm dünyaya yayılmıştır. Kullanıcılar siteyi ücretsiz üye olarak kullanabilmektedir. Site gelirini banner reklamlarından ve sponsor gruplarından almaktadır. Dünyanın en popüler Web 2.0 blogu TechCrunch'a göre; Amerika Birleşik Devletleri'ndeki üniversitelerdeki öğrencilerin %85'inin facebook'da bir hesabı bulunmakta ve bunların %60'ı her gün bağlanmaktadır. %85 her hafta, %93 her ay bağlananlar arasındadır [21]. Facebook sitesinden bağımsız Web 2.0 akımını yakından takip

eden bir blog sitesi olan insidefacebook 2009 Ocak ayının başında yayınladığı habere göre, facebook kullanıcı sayısı aşağıdaki grafikte görüldüğü gibi yaklaşık 140 milyonu bulmuştur. Milyonları aşkın aktif üye sayısı ile günde ortalama 150 bin aktif kullanıcıyla sanal dünyanın en popüler sitesi olmuştur [22].

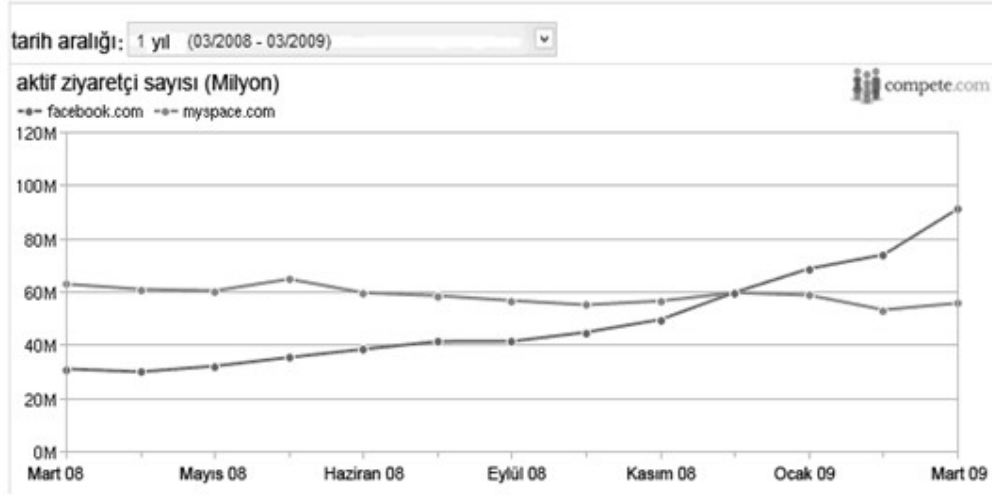


Şekil 2.8 Facebook kullanıcı sayısının aylara göre dağılım grafiği [23].

Aktif kullanıcıların içeriğe katkısı aşağıdaki istatistik sonuçlarını doğurmuştur [22].

- 13 milyon kullanıcı durumlarını en azından günde bir kez güncellenmektedir.
- 2.5 milyon kullanıcı her gün Sayfa hayranı (fan of Pages) olmaktadır.
- Siteye her ay 700 milyon fotoğraf yükleniyor ki toplam fotoğraf rakamı yaklaşık 10 milyarı bulmaktadır.
- Her ay 4 milyon video yüklenmektedir.
- Her ay 15 milyon içerik parçası (web linki, yeni hikâyeler, blog yazıları, notlar, fotoğraflar vb.) paylaşımına sunulmaktadır.
- Her ay 2 milyon etkinlik oluşturulmaktadır.
- Sitede 19 milyon adet aktif grup varlığını sürdürmektedir.

Yukarıda belirtilen bu rakamlar sosyal iletişim ağlarının ne kadar ciddi boyutlarda bir kapasiteye sahip olduğunu göstermektedir.



Şekil 2.9 Aylara göre facebook ve myspace site analizi [22]

Facebook ve ona en yakın rakibi sosyal iletişim web sitesi olan MySpace compete'de karşılaştırıldığında yukarıdaki grafik elde edilmektedir.

Tablo 2.1 Facebook ve myspace'in büyüme tablosu

Tarih: 03/2008–03/2009	Kullanıcı	Aylık	Yıllık
facebook.com	91,054,535	23,4%↑	195,1%↑
myspace.com	55,594,761	4,8%↑	-11,4%↓

Tablo 2.1'de Mart 2008 ile Mart 2009 arasındaki aylık aktif ziyaretçi sayısını göstermektedir. Compete verilerine göre Aralık 2008'de facebook, myspace'in kullanıcı sayısını yakalamıştır. Tablo 2.1'deki verilere göre bir önceki yıla göre Facebook'taki kullanıcı sayısında aylık %23,4'lük bir artış varken, MySpace %4,8'lik bir büyümeye sahiptir. Yaklaşık olarak aylık MySpace, Facebook'u 5 kat geriden takip etmektedir. Facebook'taki bir önceki yıla göre %195,1'lik büyümeye karşın, myspace'de %11,4'lük küçülme görülmektedir. Elde edilen veriler ileriki yıllarda bu rakamlar arasındaki uçurumun artacağını işaret etmektedir. Facebooktaki bu hızlı büyüme, ekonomi ve ticari sektörünün de ilgisini çekmektedir. Önemli bir yatırım alanı olarak girişimcileri cezbetmektedir.

Facebook yöneticilerinden Peter Thiel ise; Facebook'un iç değerinin 8 Milyar dolar olduğunu ve proje gelirlerinin 2015 yılı için 1 milyar dolar olduğunu söylemektedir [24]. Dünyaca tanınmış Microsoft şirketi 24 Ekim 2007'de Facebook'u %1,6'lık hissesini 240 milyon dolar'a satın almıştır. Google ve diğer şirketler de gittikçe değer kazanan Facebook hisselerine yönelmektedir [24].

2.1.5 Web Günlükler

İngilizcedeki “Web” ve “log” kelimelerinin birleşiminden oluşan Blog kelimesi internet üzerinden kişilerin istedikleri konu hakkında fikir beyan edebilecekleri, kendilerini kolayca ifade etmelerini sağlayacak bir ortamdır.

Bloglar, genellikle en yeni yorum ve yazılardan eski tarihli yorum ve yazılara doğru sıralanmış, web tabanlı yayınları belirtir. Blogların içeriği sürekli güncellendiği için sadece bloglar için kurulmuş özel indeksleme mekanizmaları ve arama motorları bulunmaktadır [25].

Dünya çapında indeksleme mekanizmalarının en önde geleni Technorati firması milyonlarca blog okuyucusu için her geçen gün 1,5 milyondan fazla yeni blogu indekslemektedir [26]. Forrester’in araştırma şirketlerinden internet ile ilgili ciddi bir araştırma kurumu olan JupiterResearch’ün 2008 Ağustos’unda bloglar üzerine gerçekleştirdiği araştırmada, internet kullanıcılarına 40 soruluk kapalı uçlu soru yöneltilmiştir. Bu araştırma sonucunda geçtiğimiz dört yıl içerisinde blogların okunma oranlarında %300 oranında bir artış olduğu ortaya çıkmıştır. "Belki ilgimi çekecek bir bloga ulaşabilirim" düşüncesi ile kullanıcılar okudukları bloga eklenmiş olan diğer linkleri de takip edebiliyorlar. “Sıkı blog okuruyum” diyenlerin %38’i yeni bir blogu okuduğu blogun linklerinden keşfettiğini belirtmiştir. Araştırmaya katılan blog okurlarının %34’ü Web’te dolaşırken yeni bir blogla tanıştığını, %11 i blog arama motorlarını kullandığını belirtmiş [27].

Blogların bu kadar yaygınlaşmasının nedenlerinde biri de, internet gezginler sanal âlemdeki tek düze, yorumsuz, hatta bazen oldukça bilimsel bir dille yazılmış yazılardan sıkıldıkları için blogları tercih etmektedir. Bloglarda kişiler gittikleri bir seminer hakkındaki yorumlarını, kullandıkları bir ürün hakkındaki fikirlerini, kısaca hayata dair kendi yorumlarını katarak başka kişilerle paylaşabilmektedirler.

2.1.6 Wiki

Bilgiye ulaşmak için eskiden olduğu gibi kitap ya da ansiklopediler kullanılmamaktadır. Günümüzde küreselleşmenin ve gelişen teknolojinin etkisiyle tüm kaynaklara bilgisayar üzerinden ulaşılması, elde edilen bilgilerin kısa sürede derlenebilmesine imkân vermiştir.

Wiki’ler, kullanıcıların üzerinde istediği gibi düzenlemeler yapmasına izin veren bilgi sayfaları topluluğudur. İngilizce “What I Know is?” sözcüklerinin kısaltması olan Wiki “Bildğim kadarıyla” anlamına gelir [28].

Wikilerin çalışma prensibi şu şekilde açıklanabilir: Bir web programı, script ya da script grubu sayesinde veritabanına yazılmış olan, ya da sadece dosya halindeki verileri okur, bu verileri HTML kodları ile değiştirir ve Web tarayıcısına yansıtır.



Şekil 2.10 Vikipedi özgür ansiklopedi sitesinin logosu

Wiki dilini kullanan ve internetin en geniş içeriğine sahip çevrimiçi ansiklopedisi Vikipedi, 22 Ocak 2007'de Altın Örümcek 2006 "En İyi İçerik" ödülünün sahibi oldu. Wikipedia.org tarafından geniş kitlelere tanıtılan Wiki dili pek çok farklı programlama platformunda "Javascript, ASP, PHP, C++ v.b." kullanılabilmektedir.

2.1.6.1 Wiki Kullanmanın Avantajları

Web proje grupları projeleri oluştururken ekiplere ayrıldığı için; programcılar kendi görevi dışındaki çalışmalarda neler yapıldığını tam olarak bilemez. İçeriğin yapılandırılmasında gelişmelerden sadece içerik koordinatörleri haberdardır. Anında yap göster özelliğiyle Wikiler bu sorun için pratik bir çözüm teşkil eder.

İndekslenme yapan sitelerde bile arama yapılırken defalarca aynı siteye girilebilir yahut aranılan bilgiyi bulmak çok zaman alabilir. Wiki gereksiz tekrarlardan kurtarır. İsteyen herkes wiki içeriğindeki bilgileri eksik ya da hatalı ise değiştirilebilir veya kaldırabilir. Bu işlemleri yapan wiki yöneticilerine wikipedist denir. Wikipedistler sayfa düzenlemelerine ve içerik zenginleştirmeye yardımcı olurlar.

Wiki'de değişiklik yapıldığında değişiklik yapan kişinin bir hesabı yoksa IP adresiyle varsa, kullanıcı adıyla kaydedilerek hemen sonuç gözlenebilir. Yapılan içeriksel herhangi bir hata başkaları tarafından fark edildiğinde düzeltilir. Geçmiş sekmesiyle düzeltilen sayfaların önceki sürümleri kronolojik sıralamayla takip edilebilir, hatta sayfalar arasında karşılaştırma bile yapılabilir. Sayfa içerikleri hakkında "Tartışma" sekmesiyle fikir beyanında bulunulabilir. Tartışmalara yorum eklenebilir. Denetleme mekanizması sayesinde telif haklarının ihlali gibi durumlarda ihlali yapan kullanıcılar engellenebilmektedir.

Wiki, diğer ansiklopedilerden farklı olarak, Vikipedi GNU Özgür Belgeleme Lisansı (GÖBL) kullanmaktadır. Kullanıcılarına özgür içerik katılımı, eşit, tarafsız ve ücretsiz katılım hakkı verir. Ayrıca dil/ülke/konuya göre gelişmiş kategorilendirme sistemine sahiptir. Diff özelliği sayesinde sayfalar arasındaki fark izlenebilir [29].

2.1.7 İçerik Yönetim Sistemleri

İngilizce “Content Management System” ya da kısaca “CMS” olarak tanınan “İçerik Yönetim Sistemleri - İYS”, bilgiyi oluşturmak, arşivlemek, araştırmak, denetlemek ve yayınlama olanağı sağlayan büyük bir veritabanına sahip modüllerden oluşan bütünleşik bir ortamdır. İYS’lerde içerik iki ayrı bölümden oluşur. Bunlardan birincisi üzerinde işlem yapılmamış olan ham bilgiyi barındıran veritabanı diğeri ise bu bilginin yerleşimini gösteren şablondur. Şablonla bilginin gösterileceği sayfanın görsel bölümünü oluşturan yazı tipleri, renkleri gibi tüm biçim bilgileri saklanır. Genellikle PHP benzeri dillerle kodlanır. Böylece daha esnek hale gelebilirler [30].

İçerik Yönetim Sistemleri, uluslararası platformda birçok organizasyon için zorunluluk haline gelen modern bir veri yönetimi tekniğidir. Günümüzde içerik yönetim sistemleri kendi içerisinde bölünmüş bir yazılım endüstrisi haline gelmiştir. Kendi içerisinde portal sistemleri, Wiki sistemleri, web-based grupware vb. alt dallara ayrılan endüstri günümüzde halen gelişimini sürdürmektedir [31].

Web yayınlama sistemi kullanıcıları, aradıkları içeriğe göre veritabanı içinde anahtar kelimeler, yazar, oluşturulma tarihi gibi çeşitli ölçütlere göre arama yapabilirler. Web içerik yönetim sistemleri bilgi portalları oluşturmak için bilgi yönetimi hizmeti sunar. İYS, bünyesinde barındırdığı yazılım modülleri sayesinde, herhangi bir kullanıcının web sitesi üzerindeki işlemleri yapılabilmesi için grafiksel kullanıcı arayüzü sunar.

İYS'nin temeli, içeriğin oluşturulmasından son hali olan yayınlanma evresine kadar tüm evrelerde içeriği yönetme esasına dayanır. Ayrıca İYS'nin içeriği, E-ticaret ve Müşteri İlişkileri Yönetim Sistemleri (MIYS) (ing. CRM - Customer Relationship Management System) ile de paylaşılabilir. İYS sayesinde, içerik yayını teknik bilgisi olmayan kullanıcılar tarafından bile yapılabilir. Kullanıcılarına sade, basit ama güçlü, web tarayıcısı destekli bir uygulama arayüzü sunar [32].

2.1.8 Etiketleme (Folksonomi)

Web 2.0 sitelerinde göze çarpan etiket bulutları ile kullanıcılara ilgilendikleri konulara ilişkin kolay ve hızlı bağlantılar sağlanmaktadır. Site içeriğiyle ilgili konular çeşitli etiketler adı altında sınıflandırılır. Her etiket içinde barındırdığı makale, yazı, başlık ile farklı boyutlarda olabilmektedir. Yani etiketin yazı boyutu ne kadar büyükse, içinde barındırdığı içerik o kadar fazladır. Şekil 2.11’de bir etiket bulutu görülmektedir.



Şekil 2.11 Örnek etiket bulutu

2.2 Web 2.0’da Site Tasarımında Dikkat Çeken Özellikler

Yukarıdaki popüler uygulamalar oluşturulurken, dikkat çeken ortak özellikler bu başlık altında incelenecektir. Genel olarak Web sitelerindeki değişim ve gelişim, tasarım ve arayüz açısından değerlendirilecektir.

2.2.1 Logolar

Logolara daha parlak bir görüntü verilirken, yansıma efekti bolca kullanılmaktadır. Yansıma efekti dinamik olmayan tüm objelere kullanılarak siteye derinlik, hareket ve modern bir görünüş kazandırılır. Kullanıcıların dikkatini çekmek için Şekil 2.12’de görüldüğü gibi logolara rozet şeklinde yenilik bildiren kelimeler eklenebilmektedir [33]. Bu yenilik bildiren kelimelerden “Beta” ifadesiyle sık sık karşılaşılır. Bunun nedeni Web 2.0 akımına göre kullanıcılar birer yardımcı-geliştiriciler olarak benimsenir. Sonuç olarak ortaya çıkan site sürekli gelişmekte ve değişmektedir. Logolara yerleşen Beta, bu gelişimin bir simgesi haline gelmiştir [34].



Şekil 2.12 Yansımalı logo örneği [34]

2.2.2 Butonlar

Butonlarda ise klasik butonlar yerine resim olarak hazırlanan butonlarda özellikle cam efekti ve renk geçiş (gradient) efekti hâkimdir. Gradient efekti butonlarda, logolarda ya da arka planlardaki renk geçişlerini daha yumuşatır. Bir renkten farklı bir renge keskin geçişlerin önüne geçilmiş olur [33]. Çeşitli buton örnekleri Şekil 2.13’de görülmektedir.



Şekil 2.13 Web 2.0’da buton örnekleri

2.2.3 Arka Planlar

Web 2.0 ile sitelerde arka planlar oluşturulurken yatay şeritlerden ya da düz renk geçişlerinden faydalanılabilir. Her iki şekilde de hedeflenen ortak amaç Şekil 2.14’te olduğu gibi gözü yormayan tasarımlar gerçekleştirmektir [33].



Şekil 2.14 Web 2.0’da arka plan örnekleri

2.2.4 Çerçeveler

Site içinde görünmez iframe'ler yerine Şekil 2.15'deki gibi çerçevelerin yuvarlak köşeli kutular olarak tasarlanması ön plana çıkmıştır. Köşeleri keskin düz, kare veya dikdörtgen biçimde kutular artık kullanılmamaktadır. Yuvarlak köşeli kutu yapımı için farklı tekniklerle oluşturulabilecek programlar mevcuttur. Resim, css ve html teknikleri kullanılacaksa Spiffy Corners, RoundedCornr ve CSS Rounded Box Generator gibi araçlar kullanılabilirken, Nifty Corners Cube ve curvyCorners gibi araçları da JavaScript tekniği ile kullanılabilir [33].



Şekil 2.15 Web 2.0'da çerçeve yapısına örnek

2.2.5 Menüler

Site içeriğinden etkin bir şekilde faydalanabilmek için menü ve sekmeler sade, kolay anlaşılabilir olmalıdır. Hem görsel hem kullanışlı menüler CSS kodlarla oluşturulabilir [33].



Şekil 2.16 Web 2.0'da menü örneği

Yukarıda bahsedilen konu Web 2.0 platformunda oluşturulan sitelerin genel özellikleri olarak algılanabilir. Bu özellikler dışında da tasarımlar olabilmektedir.

3 AJAX

AJAX kavramının ilk çıkış noktası Şubat 2005’de Jesse James Garrett tarafından, the Adaptive Path Web sitesinde “Ajax: A New Approach to Web Applications” başlığıyla yayınlanan makaledir [35].



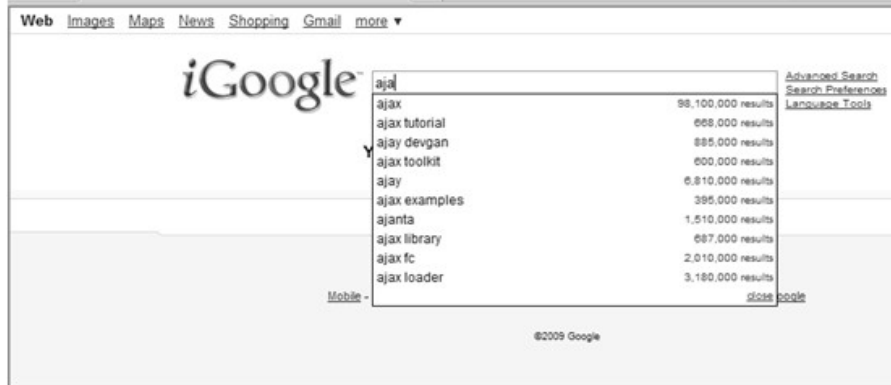
Şekil 3.1 Jesse James Garrett’in makale görüntüsü [36].

Bu makalede Web uygulamaları oluşturulurken tasarımlardaki büyüleyici gelişmeler yenilikçi çevrimiçi projelerle desteklenirken bunun yanında hızlı cevap verebilir, daha zengin görsellere sahip masaüstü uygulamaları arasındaki boşluğa dikkat çekilmiştir. Bu boşluğu dolduran Google Suggest ve Google Maps uygulamaları Adaptive Path sitesinde AJAX olarak adlandırılan Web uygulamalarına yeni yaklaşımın en iyi örnekleri olduğu belirtilmiştir. Klasik Web uygulamalarının çalışma modeliyle AJAX’lı uygulamaların çalışma modeli karşılaştırılmıştır. AJAX’ı diğer uygulama yöntemlerinden ayıran özellikleri belirtilmiştir. AJAX yaklaşımına büyük yatırımlar yapan Google’ın neden AJAX’ı tercih ettiği açıklanmıştır. Google Groups, Google Suggest, Gmail, Google Maps ve ardından Flickr, Amazon’nun A9.com servisleriyle kullanıcıların odağı haline geldiğine değinilmiştir [35].

AJAX’ın çekirdeği olan XMLHttpRequest (XHR) nesnesi sayesinde de iyi hazırlanmış, sağlam ve kolay anlaşılabilir-kullanılabilir sitelerin oluşumuna dikkat çekilmiştir [35]. Aşağıda

ise AJAX tekniği kullanan en popüler servislerden Google Suggest ve Google Maps hakkında kısa bilgi verilecektir.

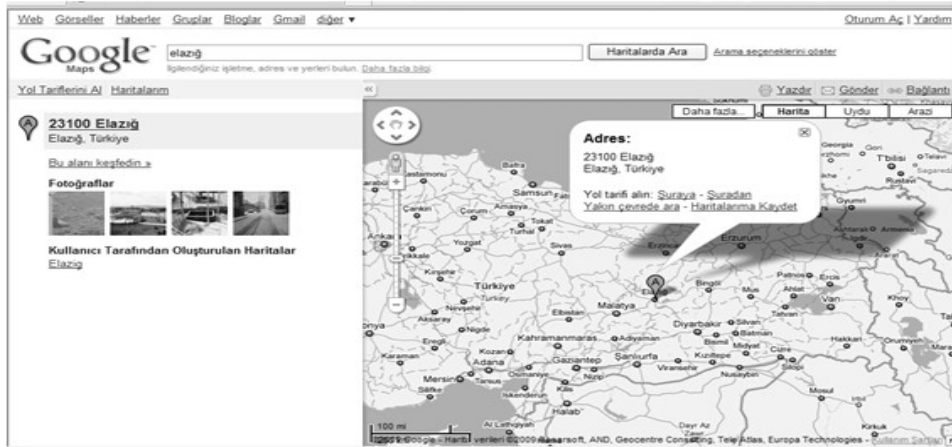
3.1 Google Suggest



Şekil 3.2 Google Suggest ekran görüntüsü

Google Suggest'a <http://www.google.com/ig?complete=1&hl=en> adresine girildiğinde Şekil 3.2'de görünen ana sayfaya ulaşılmaktadır. Google Suggest servisinin özelliği arama motorunda aranacak kelimenin harfleri girilmeye başlandığı andan itibaren filtreleme yaparak text alanının altına açılan bölmede öneriler sunulmaktadır. Sadece öneri sunmakla da kalmayıp önerileri popülerite sırasına göre sıralar. Kelime tamamlama özelliği sayesinde aranan kelime tam yazılmadan gelen önerilerden biri tıklanarak istenen sonuca ulaşılabilir. Ayrıca önerilerin yanında o kelimeye ait kaç sonuç olduğu da belirtilir ve bu işlem sadece 1-2 sn içinde gerçekleşir.

3.2 Google Maps



Şekil 3.3 Google Maps ekran görüntüsü

Google Maps sitesine <http://maps.google.com> adresinden ulaşılabilir. Google'ın etkileşimli harita servisi ile arama motorunda görüntülenmek istenen konum yazılarak “Haritalarda Ara” butonu ile o konuma odaklanılır. Bir yerden bir yere gitmek için “Yol Tariflerini Al” butonu kullanılır. Konumun isteğe göre uydu, arazi ve sokak haritası görünümüne ulaşılabilir. Daha fazla seçeneği ile resim, video ve wikipedia katmanları da görüntüye eklenir. Hatta yurtdışındaki bağlantılarda yoldaki trafik akışının hızı hakkında bile bilgi katmanı mevcuttur.

Diğer harita web uygulamalarından farklı olarak, Google Maps'te harita üzerinde çeşitli yönlerle ulaşmak için sürekli bırak yöntemi kullanılır. Harita bir imaj serisi halinde ayrılarak bölünür. Bitişik imajların birlikte görüntüsünün oluşturulabilmesi için bu serilerden faydalanılır. Görüntülenen haritada kullanılan imaj sayısı sınırlıdır. Bu da her seferinde yeni imajların oluşturulması gerektiğini gösterir. Kullanıcının harita üzerindeki hızlı hareketleri hafıza problemlerine yol açabilir. Bunun yerine aynı imajlar haritanın farklı segmentlerinde tekrar tekrar görüntülenir. [37].

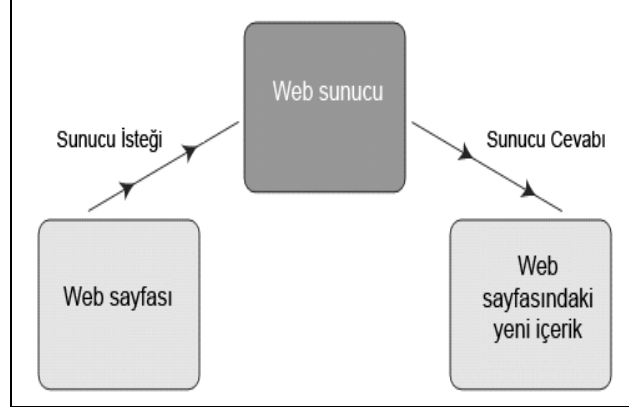
İstemci-sunucu iletişimi gizli bir IFRAME üzerinden yapılır. Bir arama yapıldığında veya yeni bir yol sorulduğunda, bu bilgiler gönderilir ve IFRAME içinde döndürülür. Veri XML formatında çevrilir ve bir Javascript fonksiyonunda (AJAX Motorunda) kullanılmak için aktarılır. Bu XML verisi sonra farklı çeşitli şekillerde kullanılır: Bazen hatasız harita imajlarının çağrılmasında bazen de XML dokümanlarının XSLT (Extensible Stylesheet Language Transformations, Genişletilebilir Biçimlendirme Dili Dönüşümleri) sayesinde HTML kodlarına dönüştürülerek, ana pencerede görüntülenmesinde kullanılır. [37].

Bunların dışındaki daha birçok açık kaynak projeler ve ticari ürünler bu yeni Web uygulama modelinin avantajlarından faydalananarak geliştirilmeye başlanmıştır [37].

3.3 Neden AJAX

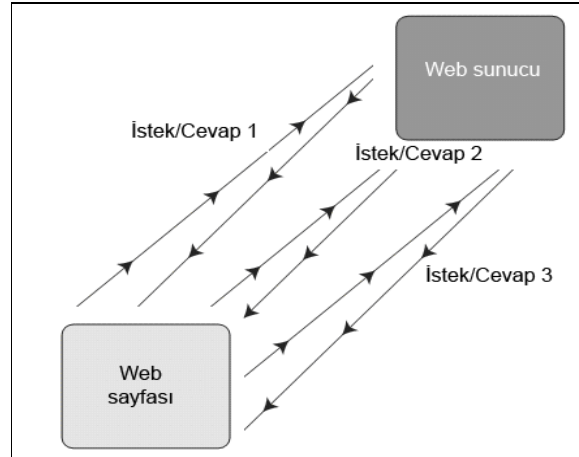
AJAX terimi oldukça güçlü bir kelimedir. Gücünü daha zengin geniş endüstri trendlerinin oluşumunda kullanılabilmesinden ve özgür Web uygulamalarının kurulumunda odak noktası olmasından alır. Ayrıca programcılar yabancı oldukları yeni programlama dilleri öğrenmelerine gerek kalmadan, geliştirdikleri Web uygulamalarına istedikleri etkiyi katabilmektedirler. Bunu yaparken de sunucu tarafı teknolojileri kullanırken var olan yatırımlarının dışında başka masrafları olmaz. AJAX güçlü bir teknik olmasının yanı sıra diğer teknolojilerle birlikte kullanıldığında daha etkili sonuçlar alınabilmektedir [38]. AJAX, Java™ 2 Enterprise Edition (JAVA EE), .Net, PHP, Flex, CGI scriptler ve diğer birçok uygulama geliştirme ortamında çalışabilir.

AJAX uygulamalarının gittikçe yaygınlaşmasının bir nedeni de geleneksel uygulamalara göre istek/cevap modelleri arasındaki farktır. Şekil 3.4'te gösterildiği üzere geleneksel Web uygulamalarında sunucuya yapılan her istekte oluşturulan cevap, Web sayfasının tümünü etkileyerek yeniden yüklenmesine neden olmaktadır.



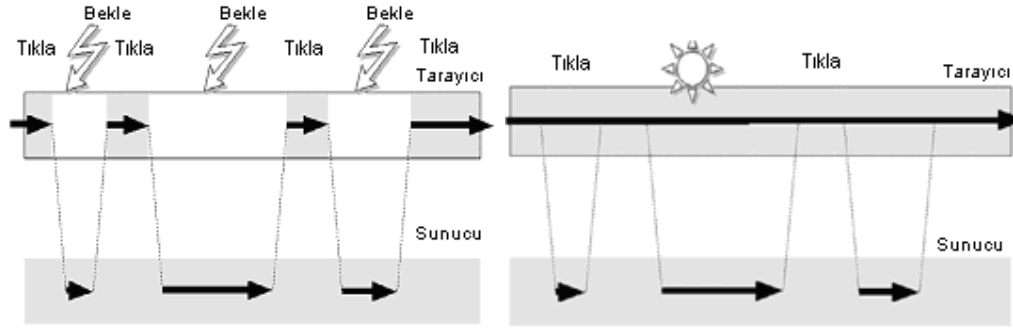
Şekil 3.4 Geleneksel sunucularda istek/cevap modeli [39].

AJAX'ın asenkron metodolojisiyle klasik Web mimarilerindeki gibi sunucuya gidip parametreleri işlemek yerine, Şekil 3.5'te görüldüğü üzere birçok sunucu isteği tüm sayfanın yenilenmesine ihtiyaç duymadan cevaplanabilmektedir.



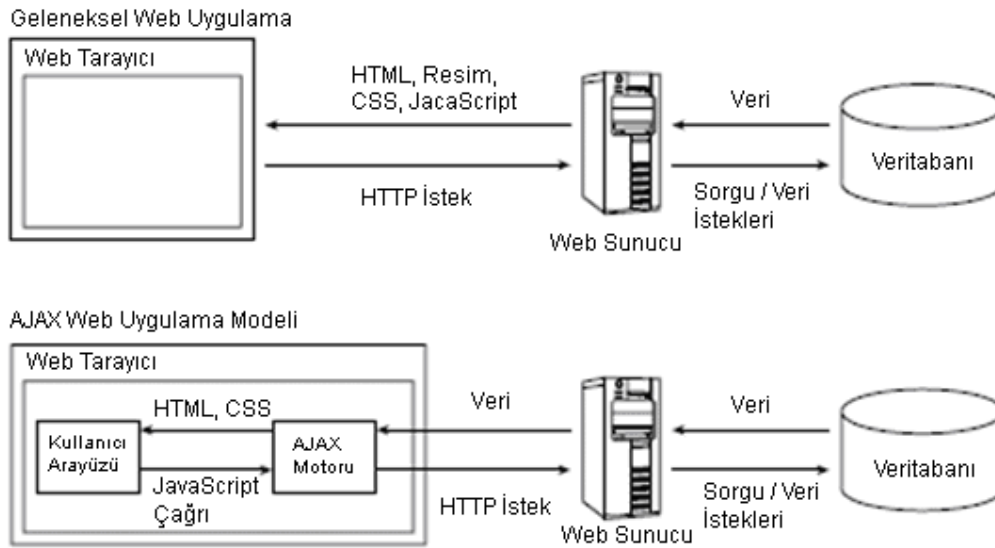
Şekil 3.5 AJAX'ın asenkron istek/cevap modeli [39].

Yukarıda anlatılan durum kullanıcıları Şekil 3.6'da görülen tıkla-bekle döngüsünden kurtarır. AJAX kullanıcılara Şekil 3.7'deki gibi kesintisiz gezinti (sörf) imkânı sunar.



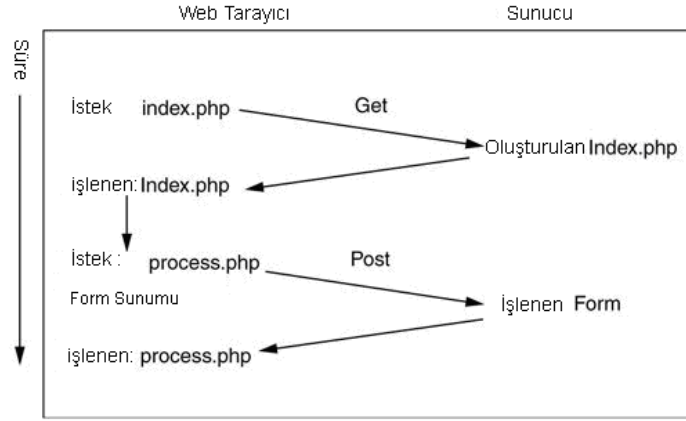
Şekil 3.6 Klasik Web uygulamalarında tıkla-bekle döngüsü [40]. **Şekil 3.7** Tıkla-Tıkla interaktif AJAX uygulaması [40].

İstek işlem basamakları Şekil 3.8’te gösterildiği gibi AJAX motoru sayesinde JavaScript kod betikleri kullanılarak, sunucuya HTTP tabanlı bağlantı sağlanır. Bu bağlantı XMLHttpRequest protokolünü kullanarak mesaj transferi yapılmasını sağlar. Bu sayede hem hız artar hem de bant genişliği verimli bir şekilde kullanılmış olur.



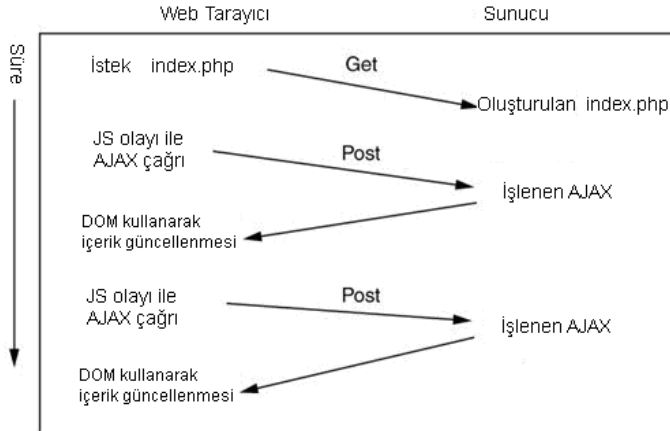
Şekil 3.8 Geleneksel Web iletişim modeli ile AJAX iletişim modeli [37]

AJAX, sunucuya kullanıcının Web tarayıcısı üzerinden Javascript ortamında ikinci bir iletişim yoludur. AJAX’ın kullanımı normal Web uygulamalarında görülen tipik sayfa akışında değişikliklere neden olur. AJAX ile istekler daha sık ve daha küçük cevaplar halinde HTML olmayan veri formunda cevaplanır. Daha iyi anlayabilmek için normal bir Web uygulamasının iletişim akışı Şekil 3.9’da görülmektedir [41].

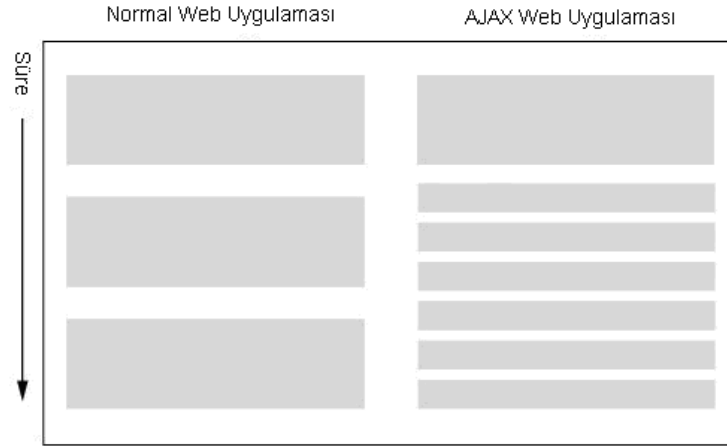


Şekil 3.9 Web uygulamasındaki cevap akışı

Bir AJAX Web uygulaması bir Web uygulamasının normal iletişim akışını alır ve yeni bir istek tipi ekler. Bu durum, sunucuya normal sayfa isteği gibi görünür ama Web tarayıcı için farklı formatta bir girişi vardır. Kullanıcı tarafından doğrudan tetiklenmeyen ve sayfanın tamamının yeniden yüklenmesini gerektirmeyen bir istektir. Çoğu durumda bu AJAX istekleri bazı verilerin ön yüklenmesi ya da bir formdaki bazı alanların doğrulama istekleri gibi küçük değişiklikler olabilir. Bazen de JavaScript ile bir form gönderimi veya HTML sayfalarının bir parçası üzerindeki içeriği değiştirmek için kullanılır. Örnek bir AJAX iletişim akışı Şekil 3.10'da görülmektedir. AJAX uygulamalarındaki istek bitiş zamanları ve bunun normal Web uygulamalarıyla karşılaştırması Şekil 3.11'de görülebilir. Böylece ikisinin arasındaki istek frekanslarının farkına varılabilir [41].



Şekil 3.10 AJAX uygulamalarındaki istek akışı

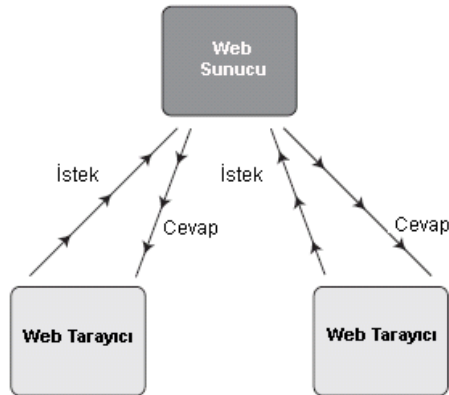


Şekil 3.11 Normal Web ile AJAX uygulamalarının karşılaştırması

AJAX Web uygulamalarıyla frekanstaki bu değişim hostlar için büyük etki yapabilir. Böylece sunucular aynı kullanıcı sayısı ile yüksek hızlarda veri alış-verişini yapabilecek duruma gelir. Bu özellikle daha fazla işlem gerektiren normal sayfa yüklemelerinde oldukça zor bir işlem idi [41].

3.4 AJAX Metodolojisi

World Wide Web istemci-sunucu ağ mimarisine uygun çalışır [42]. Ağ aktiviteleri iki tiptir. Bunlardan birincisi kullanıcı tarafından tetiklenen HyperText Transfer Protocol (HTTP) istekler (URL yazarak ya da bir link tıklanarak oluşturulur) diğeri ise sunucu cevaplarıdır. HTTP protokolü soru ve cevaplar serisinden oluşan bir iletişim yoludur [41].



Şekil 3.12 Geleneksel istek/cevap (request/response) yöntemi [39].

Web tarayıcıdan bir istekte bulunulduğunda HTTP kullanılarak cevap gönderilir. Web tarayıcı bir istemci, internet sayfasının sunucusu ise HTTP sunucudur. AJAX sunucu cevaplarını HTTP kullanarak gönderir. Sunucu tarafından geri döndürülen farklı türlerdeki HTTP istek ve cevaplarının tanınması önemlidir. AJAX uygulamaların sunucudan gelecek olan HTTP cevapların içeriğinin ne olduğu hakkında karar verebilmesi için sunucunu sorgulanmasında HTTPRequest nesnesine ihtiyaç duyar [42].

3.4.1 HTTP Requests Nesnesi

HTTP request nesnesinin kullanım formatı aşağıdaki şekildedir:

```
<request-line>
<headers>
<blank line>
[<request-body>]
```

HTTP request nesnesinin ilk satırı genellikle isteğin tipini belirten üç bölümden oluşur. Yöntemin adı, gerekli olan sunucu kaynağının yolu, HTTP sürümüdür. Birinci istek satırı (açılış satırı) olmalıdır. İkinci bölüm sunucunun kullanabileceği başlık satırı, eklenebilen bilgilerin gösterildiği bölümdür. Son bölüm olan boş satır ise isteğe bağlı olarak BODY bölümünden çağrılacak ek verilerin takip edilebileceği satırdır.

HTTP içinde çeşitli istek türlerinin tanımlanabileceği çok sayıda yöntem vardır. Fakat AJAX sadece GET ve POST yöntemleriyle ilgilenir.

3.4.1.1 GET Yöntemi

HTTP GET yöntemi, çağırılmak istenen URL adresi ardına eklenen anahtar-değer çiftleri (key-value pair) ile çalışır. Bir Web tarayıcıdan talep edilen bir URL çağırısı, sunucuya GET yönetimi ile bir istek gönderir. Basitçe sunucu kaynağı getirir ve tarayıcıya cevabı gönderir. Aşağıda Get yönetimi kullanılarak Fırat üniversitesinin Web sitesine yapılan bir çağrı satırı gösterilmektedir.

```
GET / HTTP/1.1 → <request-line>
Host: www.firat.edu.tr
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)
Connection: Keep-Alive } <headers>
```

Örnek kod parçasında da görüldüğü üzere ilk satır GET istek satırıdır. Diğer bölüm ise istek hakkında bilgi gönderilen başlık (header) kısmıdır. Bu bölüme bir parametre ve her satır başına bir değer çifti gelebilir. Host başlığı isteğin hedef sunucu adresini içerir. User-Agent başlığı erişilebilir her iki sunucu ve istemci taraflı scriptler ve tarayıcıların temel taşıdır. Bu bilgi ile her istekte otomatik olarak gönderilir ve kullanılan tarayıcı tanımlanır. Connection başlığı ise tarayıcı işlemleri için kullanılan tipik bağlantı ayarlarında kullanılır.

3.4.1.2 POST Yöntemi

POST isteği ise sunucuya request body içinde ek bilgi sağlar. Genellikle çevrimiçi formlarda doldurulmayan satır bilgileri POST yöntemi ile sunucuya bildirilir. POST istekleri aşağıdaki yapıdadır:

POST / HTTP/1.1

Host: www.firat.edu.tr

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)

Content-Type: application/x-www-form-urlencoded

Content-Length: 40

Connection: Keep-Alive

Yukarıdaki kod parçasında gri ile vurgulanan satırlar GET yönteminden farklı olan satırları göstermektedir. İlk satırda kullanılan yöntemin adı belirtilmiştir. Content-Type satırı request-body'nin nasıl kodlandığını gösterir. Content-Length satırında request-body'nin kaç byte uzunluğunda olduğu gösterilir [37].

3.4.2 HTTP Request Nesnesinin Özellikleri ve Yöntemleri

Request nesnesi beş tane temel fonksiyon bulundurmaktadır [43]. Bunlar;

- **Querystring:** URL içerisinde bulunan dosya ismi ile birlikte gönderilen bilginin veya form içerisinde “GET” yöntemi ile gönderilen bilgilerin alınmasında kullanılır.
- **Form:** “POST” yöntemi genellikle form içerisinde bilgi alınması için kullanılır.
- **Servervariables:** Web sunucu request ile ilgili bilgilere (Http ServerVariables) ulaşmayı sağlayan bir yöntemidir.
- **Cookies(Çerezler):** Kullanıcının tarayıcısı sunucudan gelen çerezleri kabul ediyor ise bu bilgilere ulaşılmasını sağlar.

- **ClientCertificate:** Dijital bir sertifikadır. Sunucu ve istemcinin birbirini tanımlaması için kullanılır.

3.4.3 HTTP Response Nesnesi

HTTP response nesnesi HTTP request nesnesine benzer şekilde gösterilir:

```
<status-line>  
<headers>  
<blank line>  
[<response-body>]
```

HTTP request nesnesinden tek farkı ilk satırda istek bilgisinin yerine, içeriğin durum bilgisi vardır. Durum satırı istek kaynağı hakkında temin edilen bir durum kodunu bildirir. Aşağıda basit bir HTTP response kod parçası gösterilmektedir:

```
HTTP/1.1 200 OK  
Date: Sat, 31 March 2009 16:59:59 GMT  
Content-Type: text/html;charset=ISO-8859-1  
Content-Length: 122  
<html>  
  <head>  
    <title>Firat Universitesi</title>  
  </head>  
  <body>  
    <!-- body buraya gelir! -->  
  </body>  
</html>
```

Bu örnekte, durum satırında 200 şeklinde bir HTTP durum kodu ve “Tamam (OK)” diye bir mesaj verilmektedir. Durum satırı her zaman durum kodu ve her hangi bir karışıklığı önlemek için kısa bir mesaj içerir. Aşağıda en yaygın kullanılan durum kodları verilmektedir:

200 (OK): Sunucu başarılı istekler için bu kodu kullanır.

304 (NOT MODIFIED): Sunucu gönderilen istekle ilgili verilerin daha işlenmediğini istemciye bildirmek için bu kod kullanılır. Tarayıcıların ön bellek mekanizmaları için en sık kullanılan koddur.

401 (UNAUTHORIZED): “İstemcinin kaynak erişim yetkisi yoktur” anlamında kullanılır. Genellikle bu tarayıcının sunucudan kullanıcı adı ve şifre gibi bilgi girişi istemesiyle oluşan hatalarda meydana gelir.

403 (FORBIDDEN): “Yetki elde etmek için istemci başarısız oldu” hata kodudur. 401 hatasından sonra eğer doğru bir kullanıcı adı ve şifre ile oturum açılmaması halinde, genelde bu hata kodu oluşur.

404 (NOT FOUND): Kaynak belirtilen yerde bulunamadığında bu hata kodu üretilir.

Kod parçasının ikinci satırında üretilen cevabın tarihi ve saati Date başlığında sunucuya gönderilir. Content-Type ve Content-Length başlıkları POST yönteminde olduğu gibi kullanılır. BODY cevabı basitçe istemci kaynağı (içerik düz metin, ikili veri ya da farklı türden bilgiler olabilir) HTML kaynağını içerir [42].

3.5 AJAX İletişim Teknikleri

Web uygulamalarında çoğunlukla kullanıcı istekleri bir form üzerinden yapılır ve sunucu cevabı hızlı bir şekilde Java ya da Php gibi bir programlama dili ile oluşturulur. Normal Web uygulama modelinde kullanıcı daima istek üretir, böylece kullanıcıların hızlıca tıklamalarıyla oluşan yüksek hızlardaki sayfa istekleri genellikle rastgele aralıklarla oluşur ve her bir istek hızının cevabı düşüktür [41].

AJAX uygulamalarında yukarıda bahsedilen eksikliğin önüne geçebilmek için çeşitli iletişim teknikleri geliştirilmiştir. Bu teknikler:

- Gizli Çerçeve (Hidden Frame) Tekniği: İstemci-sunucu iletişiminde kullanılan gizli çerçeve yapısı, yüksekliği ve genişliği 0 pikselden oluşan ekranda görünmeyen bir frameset yapısına dayanır.
- XMLHttpRequests (XHR): AJAX teknolojisinin temel nesnesidir. Aşağıdaki başlıkta ayrıntılı bir biçimde incelenecektir.

3.5.1 XMLHttpRequests (XHR) Nesnesi

AJAX’la doğrudan ilgili teknolojilerin çoğu asenkron ve XML tekniği ile yakından ilgilidir. Çünkü veri kodlama yolları ve iletişim katmanlarının tedarik edilmesi bu teknikler ve kütüphaneler üzerinden hareket eder. Çoğu durumda bu XMLHttpRequest nesnesi kullanılır ancak IFrame’lerde popülerdir. Çerezler veya ActiveX / Java gömülü başka bir alternatif de kullanılabilir [41].

XHR Nesnesi, Internet Explorer 5.0 (IE 5) içindeki ActiveX control ile ortaya çıkmıştır. XHR Javascript ile oluşturulur [44]. Sayfaların tamamen yeniden yüklenmeden sağlanan parçalı güncelleme özelliği bu nesne sayesinde olmaktadır. Arka planda çalışan bu sistem sayesinde asenkron bir iletişim gerçekleşir. Yenilenme sırasında tarayıcıyla bağlantı kesilmeksizin yeni bir istek yapılabilir.

World Wide Web Consortium (W3C) standart organizasyonu, standartlaşmamış XMLHttpRequest nesnenin tüm tarayıcılarda kullanılabilen yöntem ve özelliklerinin genel bir alt kümesini oluşturma çalışmaları sürmektedir. Şu anda kullanılan kural kümesi W3C DOM Level 3 yükleme ve kaydetme özellikleri içerir. İkinci çalışma taslağı www.w3.org/TR/XMLHttpRequest adresinde bulunabilir. Henüz bir standart haline gelememesinin nedeni tarayıcıdan tarayıcıya göre farklı davranışlar göstermesidir.

Çalışma taslağına göre XHR nesnesinin iki görevi vardır. Birincisi HTTP yolu ile sunucuya programlanabilir bağlantı kuran betiklerin çalışmasına izin verir. İkinci ifade şekli ise normal HTTP istek/cevap döngüsünün dışında sunucuyla iletişime izin verir.

XMLHttpRequest, Microsoft tarayıcı çeşitlerinde, Mozilla, Firefox, Konqueror, Opera ve Safari gibi neredeyse tüm modern tarayıcılar tarafından desteklenmektedir [42].

3.5.1.1 XMLHttpRequest Nesnesi Oluşturma

Nesneler oluşturulurken farklı tarayıcılar için farklı kurallar vardır. Örneğin Microsoft Internet Explorer XMLHttpRequest nesnesini bir ActiveX nesnesi olarak, Safari, Mozilla Firefox ve Opera ise sadece normal bir JavaScript nesnesi olarak yorumlar.

Windows'un güncelleme sisteminin temelini oluşturan ActiveX Web sayfalarındaki aktif nesneleri etkinleştiren özel bir Microsoft teknolojisidir. ActiveX'i mevcut Web tarayıcıları arasında, şu anda sadece Microsoft Internet Explorer desteklemektedir. Microsoft IE'nin diğer sürümleri için MSXML geliştirerek daha aktif ve kararlı bir yapı oluşturulmuştur. IE'nin farklı sürümleri için aşağıdaki bileşenler kullanılır [45]:

- Microsoft.XMLHttp
- MSXML2.XMLHttp
- MSXML2.XMLHttp.3.0
- MSXML2.XMLHttp.4.0
- MSXML2.XMLHttp.5.0

IE 6.0+ tarayıcılar için aşağıdaki tanımlama yapılır.

```
var nesne = new ActiveXObject("Msxml2.XMLHTTP");
```

IE 5.5 sürümleri için ise aşağıdaki tanımlama yapılır.

```
var nesne = new ActiveXObject("Microsoft.XMLHTTP");
```

Mozilla Firefox , Opera, Safari gibi tarayıcılar için ise aşağıdaki tanımlama yapılır.

```
var nesne = new XMLHttpRequest ();
```

Aynı anda birden fazla internet tarayıcısı (ie, firefox, opera, vb) ile uyumlu Web uygulamaları için çapraz tarayıcı desteği ifadesi kullanılmaktadır. Çapraz tarayıcı desteği, bir arayüz kodlamasında kodlayıcıların en çok zaman harcadığı konudur. Günümüzde çapraz tarayıcı desteği bir zorunluluk haline gelmiştir [46]. AJAX uygulamalarının sorunsuz tüm tarayıcılarla uyumlu bir şekilde çalışabilmesi için JavaScript içerisinde try-catch blokları yazarak tarayıcıya göre yeni bir nesne oluşturulur.

```
function NesneOlustur()
{
    var nesne;
    try
    {
        nesne = new ActiveXObject("Msxml2.XMLHTTP"); // IE 6.0+ tarayıcılar için
    }
    catch(e)
    {
        try
        {
            nesne = new ActiveXObject("Microsoft.XMLHTTP"); // IE 5.5 tarayıcıları için
        }
        catch(e)
        {
            try
            {
                nesne = new XMLHttpRequest (); // Firefox, Opera 8.0+ ve Safari tarayıcıları için
            }
        }
    }
}
```

```

        catch(e)
        {
            alert(" Maalesef tarayıcınız AJAX'ı desteklemiyor!");
            return null;
        }
    }
    return nesne;
}

```

Yukarıda tanımlanan fonksiyon sayesinde dinamik ve çapraz tarayıcı desteğine sahip bir XHR nesnesi oluşturulmuştur.

3.5.1.2 XHR Nesnesinin Özellik ve Yöntemleri

XMLHttpRequest nesnesine eklenebilen özellikler Tablo 3.1’de gösterilmektedir [42].

Tablo 3.1 Standart XHR nesnesinin özellikleri

Özellikler	Tanımlar
onreadystatechange	Nesnenin readyState özelliği değiştiğinde hangi olay tutucunun (event handler) çağrılacağını belirler.
readyState	İsteğin durumunu raporlayan tamsayı değeridir: 0 = uninitialized (İstek hazırlanmadı. Henüz open yöntemi kullanılmadı) 1 = loading (İstek hazırlandı, gönderiliyor) 2 = loaded (İstek gönderildi) 3 = interactive (İstek değerlendiriliyor ve sonuç bekleniyor) 4 = completed (İstek tamamlandı ve sonuç geldi)
responseText	Form içindeki metin dizesi sunucu tarafından veriye dönüştürülür.
responseXML	Sunucu tarafından bir belge nesnesi gibi ifade edilen veri geri döndürülür.
Status	Sunucu tarafından geri döndürülen HTTP durum kodudur. Örneğin 404 kodu “Not Found” yani istenilen sayfa bulunamadı anlamına gelir. 200 ise “OK” değeri gönderir ve işlemin tamamlandığını belirtir.

statusText	Status özelliğinin metin karşılığını verir.
------------	---

XMLHttpRequest nesnesinin fonksiyonunu arttıran ek yöntemler Tablo 3.2’de gösterilmektedir. Bu yöntemler sayesinde sunucuyla asenkron iletişim kurulur.

Tablo 3.2 Standart XHR nesnesinin fonksiyonları [42]

Yöntemler	Tanımlar
abort()	Geçerli isteği iptal eder.
getAllResponseHeaders()	Gelen cevabın başlık bilgilerini string olarak getirir.
getResponseHeader(x)	x gibi bir parametre olarak verilen başlık bilgisini getirir.
setRequestHeader('x','y')	İstekle gönderilen başlık bilgisine istenilen etiketi ya da değeri atar.

Tablodaki yöntemler dışında open ve send yöntemleri de sıkça kullanılmaktadır.

Open yönteminin parametreleri aşağıda gösterilmektedir.

```
nesne.open( "POST | GET" , "URL", asynch, "username", "password");
```

- 1. Parametre:** Web sayfasına yapılacak isteğin hangi yöntemle yapılacağını belirtir.
- 2. Parametre:** İstek yapılacak Web sayfasının URL adresi. Eğer GET yöntemi kullanılıyorsa querystring’lerin değerleri URL’e birleştirilerek belirtilmelidir.
- 3. Parametre:** True değeriyle yapılacak isteğin eşzamansız olacağı, false değeri ile eşzamanlı olacağı belirlenir. Eğer false değeri verilirse işlem bitene kadar tarayıcı diğer kullanıcı isteklerine cevap veremeyecektir. True yapılırsa istek ayrı bir işlem olarak ele alınacaktır.
- 4. ve 5. Parametre:** Opsiyonel parametreler olup eğer istek yapılan Web sayfası şifre ve kullanıcı adı istiyorsa bu şekilde gönderilebilir [47].

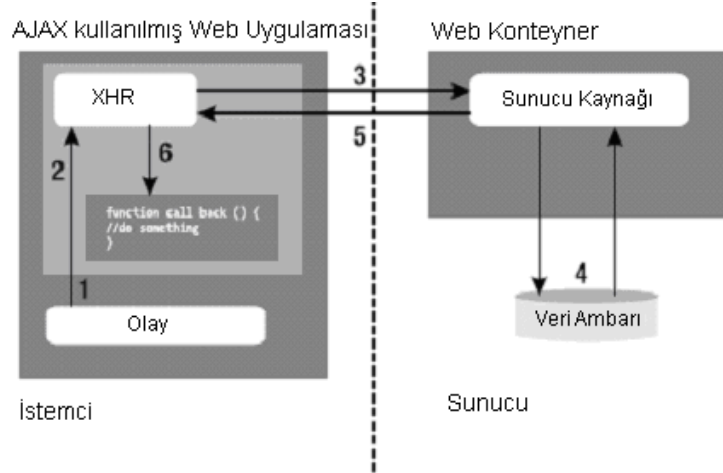
Open ile hazırlanan istek sunucuya gönderilir. Eğer open ile açılan istek post metodunu kullanıyorsa, send metodu parametre olarak gönderilecek veriyi taşıyabilir.

Send metodu parametresiz veya null parametresi ile de kullanılabilir. Send metodunda cevabın gelip gelmediğinin anlaşılabilmesi için yani eşzamanlı haberleşmenin hangi safhada olduğunun belirlenebilmesi için Tablo 3.1’deki özellikler eklenir. Send metodunun kullanım şekli aşağıda verilmiştir [42, 47, 48, 49].

```
nesne.send(); veya nesne.send(null);
```

3.5.1.3 XHR Çalışma Mantığı

Şekil 3.13 bir AJAX uygulamasında standart etkileşimli bir paradigmayı gösterir [49]. İşlem adımları aşağıda sıralanmaktadır.



Şekil 3.13 Standart AJAX etkileşimi [49].

1. İstemci taraflı bir olay AJAX olayı (event) tetikler. İstemcinin aşağıdaki gibi her hangi bir hareketi onchange olayını başlatır.

```
<input type="text" id="eposta" name="eposta" onblur="validateEposta()";>
```

2. Örnek bir XHR nesnesi oluşturmak için open() yöntemi kullanılır ve çağrılar için ayarlanır. URL ayarları isteğe göre kullanılacak HTTP yöntemlerinden GET ya da POST'a göre belirlenir. Aşağıdaki kod örneğinde olduğu gibi, istek send() yöntemi ile bir çağrıyı tetikler.

```
var xmlhttp;  
function validateEposta() {  
    var eposta= document.getElementById("eposta");  
    var url = "validate?eposta=" + escape(eposta.value);  
    if (window.ActiveXObject) {  
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    else if (window.XMLHttpRequest) {  
        xmlhttp = new XMLHttpRequest();  
    }  
    xmlhttp.open("GET", url);  
    xmlhttp.onreadystatechange = gericagir;
```

```
xmlHttp.send(null);  
}
```

3. Sunucuya bir istek yapılır. Bu bir servlet, CGI script, PHP ya da ASP.NET ile yazılmış herhangi bir sunucu taraflı (server-side) script olabilir.
4. Sunucu herhangi bir veri ambarına erişebilir.
5. İstek tarayıcıya geri döndürülür. XHR nesnesi Content-Type ayarı “text/xml”dir. XHR nesnesi sadece “text/html” tipindeki sonuçları işleyebilir. Daha karmaşık örneklerde cevap JavaScript, DOM manipülasyon ya da başka benzer teknolojiler içerebilir.

```
response.setHeader("Cache-Control", "no-cache");  
response.setHeader("Pragma", "no-cache");
```

Tarayıcı lokal sonuçları ön belleğinde depolamaz. Bunun için yukarıdaki kod satırlarına ihtiyaç duyar.

6. Bu örnekte işlemler geri döndürüldüğünde gericagir() adındaki XHR fonksiyonu ayarlanır. Bu fonksiyon sunucudan gönderilen durum kodlarına bakar ve XHR nesnesinin readyState özelliğini kontrol eder. gericagir() fonksiyonu beklendiği gibi işlenmişse, aşağıdaki gibi bir kod parçası görülür.

```
function gericagir() {  
    if (xmlHttp.readyState == 4) {  
        if (xmlHttp.status == 200) {  
            //İstek Başarılı}  
        }  
    }  
}
```

Yukarıdaki anlatımla XHR nesnesinin çalışma mantığına değinilmiştir. Genellikle kullanılan standart çağrılar örneklenmiştir. Bu çağrılar yapılırken bir kütüphaneden, yapılan bir uygulamadan da faydalanılabilir. Yeni bir çalışma alanı olmasına rağmen açık kaynak topluluklarında hatırı sayılır etkinlikler yapılmaktadır [49].

Uygulamalar oluşturulurken genellikle çeşitli frameworkler ve toolkitlerden faydalanılır. Bunlardan bazıları sadece istemci taraflı çalışır, bazıları ise sunucu üzerinde çalışmaya ihtiyaç duyar.

Bu frameworkler sürekli değişen ve gelişen bir ortam olduğu için yeni kütüphaneler ve sürümleri takip edilmelidir. En yaygın kullanılan kütüphanelerin başında Prototype, script.aculo.us, Dojo Toolkit, Direct Web Remoting (DWR), Taconite ve Rico gelmektedir [49].

3.6 AJAX'ı Oluşturan Bileşenler

Asynchronous JavaScript and XML kelimelerinin baş harflerinden oluşan AJAX kavramı kullanışlı, zengin Web uygulamaları geliştirebilmek için pratik bir araçtır [50].

AJAX'ın "A" parçası veri iletişim tekniğini, "JA" parçası iletişim yöntemini, "X" parçası tekniğin veri kodlama kısmını oluşturur. AJAX'ı tam manasıyla anlayabilmek için onu oluşturan bileşenlerin her birini de anlamak gerekmektedir [41].

3.6.1 Eşzamanlı Olmayan İletişim

AJAX tekniğinin çekirdeğini oluşturan XMLHttpRequest nesnesi hem eşzamanlı hem de eş zamansız olarak çalışabilen bir yapı üzerine kurulmuştur.

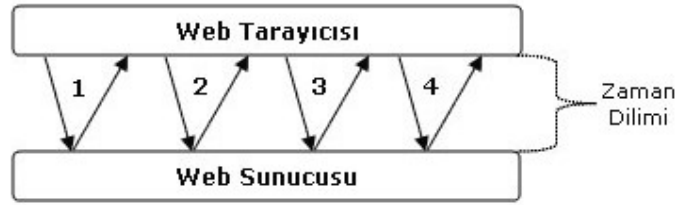
Çoğu AJAX uygulaması XHR nesnesini asenkron olarak kullanır. Asenkron istek tarayıcıya gönderilir ama sunucudan bir cevap gelmesini beklemektense sonraki istek kodları işletilmeye devam eder. İşlenen cevap gönderildiğinde XHR nesnesinin onreadystatechange özelliği bir fonksiyona bağlanır. Bu fonksiyonun işletilmesinden sonra readystate koduyla isteğin durumu test edilir. readystate özelliği 4 değerini aldığı anda, sunucuya "istek tamamlandı" ve "sonuç geldi" mesajı gider [51].

Aşağıdaki kod parçasında asenkron bir iletişim gösterilmektedir [51]:

```
// get recipe using Ajax
function getRecipe(evt) {
    if (!xmlhttp) xmlhttp = aaGetXmlHttpRequest( );
    if (!xmlhttp) return;
    var drink = encodeURIComponent(document.getElementById('drink').value);
    var qry = "drink=" + drink;
    var url = 'recipe.php?' + qry;
    xmlhttp.open('GET', url, true);
    xmlhttp.onreadystatechange = printRecipe;
    xmlhttp.send(null);
    if (evt && evt.preventDefault( ))
        evt.preventDefault( );
}
```

```
return false;  
}
```

Eşzamansız iletişimde kullanıcı aynı zaman diliminde birden fazla işlem yapabilir. Örneğin Web sayfasında bir form doldurulduğunda, formdaki bilgiler AJAX ile gönderilirken işlemin bitmesini beklemek gerekmemektedir. O zaman diliminde Web sayfasının başka bir yerinde bulunan anket cevaplandırılabilir. Çünkü eşzamanlı çalışılırken bloke edilen JavaScript motoru, eşzamansız çalışırken serbesttir ve başka isteklerin gönderilmesine müsaade eder. Yani aynı zaman dilimi içerisinde birden fazla isteği kabul edebilir [52].



Şekil 3.14 Eşzamanlı olmayan iletişim modeli

Şekil 3.14’te görüldüğü gibi aynı zaman dilimi içerisinde 4 adet istek yapılmıştır. Kullanıcı bu isteklerin tamamlanmasını beklemek zorunda değildir. Bu sebeple istemci ve sunucu arasındaki tıkla-bekle-tıkla-bekle etkileşimi ortadan kalkmaktadır.

3.6.2 Eşzamanlı İletişim

Senkron isteklerde tarayıcı cevap için beklerken kilitlenir.

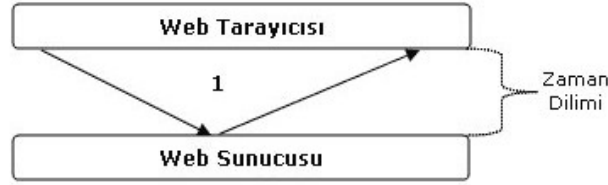
```
var state = document.forms[0].elements[0].value;  
  
var url = 'ajax.php?state=' + state;  
  
xmlhttp.open('GET', url, false);  
  
xmlhttp.send(null);  
  
alert(xmlhttp.getAllResponseHeaders( ));
```

Yukarıdaki kod dizesiyle bir form alanına Web servisi üzerindeki HTTP GET istek yapısı kullanılarak erişilir. Üçüncü satırdaki open yönteminin üçüncü parametresinin false olmasının anlamı, isteğin eş zamanlı (senkron) olduğunu gösterir. Herhangi bir parametre almadan null değerli istek gönderilir. İstek gönderildikten hemen sonra uyarı mesaj kutusu

açılarak cevap başlığı yazılır [51]. Kullanılan Web servisi hızlıysa cevap hemen gelir. Eğer servis yavaşsa istek bitinceye kadar diğer sayfa aktiviteleri gerçekleşemez [51].

Eşzamanlı iletişimde aynı zaman diliminde yalnızca bir işlem yapılabilir. Web tarayıcımızdan uzaktaki Web sunucusuna gönderilen istek, Web sunucusunda işlenir ve tekrar geldiği yoldan Web tarayıcısına geri döner. Bu işlem belli bir zaman alır. Eşzamanlı çalışırken bu zaman diliminde başka bir istek gönderilemez [52].

Eşzamanlı çalışma, özellikle Web sunucusu aynı bilgisayarda ise veya yakın olan bir LAN üzerindeyse sağlıklı çalışabilir. Fakat Web sunucusu ağır bir yük altındaysa veya Web tarayıcısı Web sunucusuyla yavaş bir bağlantı üzerinden etkileşime geçtiyse, kötü bir performans verecektir. Çünkü Web tarayıcısı içindeki JavaScript motoru, gönderdiği istek tamamlanincaya kadar bloke edilir ve Web tarayıcısı donmuş bir şekilde isteğin tamamlanmasını bekler. Bu zaman diliminde kullanıcı, gönderilen isteği iptal de edemez. Bu yüzden Web sayfasındaki başka bir noktaya tıklayamaz veya başka bir sekmeye geçiş yapamaz [52].



Şekil 3.15 Eşzamanlı iletişim modeli

Şekil 3.15’de görüldüğü gibi Web tarayıcısından gönderilen 1 nolu istek, Web sunucusuna iletilmektedir. Alınan istek işlendikten sonra Web tarayıcısına geri gönderilmektedir. Bu zaman diliminde başka bir istek yapılamaz.

3.6.3 Javascript

1995 yılında Brendan Eich adındaki yazılım geliştiricisi tarafından Netscape Navigator 2.0 ile birlikte geliştirilen JavaScript dili istemci taraflı (client-side) bir betik (script) dilidir. Web sayfalarında dinamik içerik sağlamak ya da kullanıcıyla iletişim kurmak için kullanılır [53]. Bu iletişimin sağlanabilmesi için ortak kurallar dünyada uluslararası standart kurumu olan Avrupa Bilgisayar Üreticileri Birliği (European Computer Manufacturers Association-ECMA) tarafından belirlenir. Temel kuralları ECMA Script (ECMA-262) adı verilen bir standartta toplanmıştır [54].

Javascript'ın söz dizim kuralları C diline benzerlik gösterir. Derleme işlemi yapılmayan ayrıştırılmalı (parsed language) bir dildir. Nesne yönelimli programlama (Object Oriented Programming - OOP) yeteneğine sahiptir [55].

AJAX'ın JavaScript'le olan ilişkisi kısaca şu şekilde özetlenir: Herkes tarafından bilinen gönder (submit)-cevap(response) döngüsünün önüne geçerek kullanıcıların Web sunucusuyla direk etkileşim içinde bulunmasını sağlar [56]. AJAX gönder düğmesi olmaksızın sunucu etkileşimini sağlar. Veriye ihtiyaç duyulduğunda Web sayfası içindeki Javascript kodları ile bir istek yapar. Sunucu cevabı tam bir HTML sayfası olarak değil, veri olarak geri döndürür ve istemciyi cevaplar. Cevabın bu şekilde oluşturulması Web uygulamalarının, masaüstü uygulamalar gibi çalışabilmesini sağlar [56].

Sonuç olarak Javascript HTML kodları arasına gömülebilen kod parçalarıdır. Bunun yanı sıra tek başına .js uzantılı JavaScript kodları da yazılabilir. Bu kodlar Internet Explorer, Firefox, Opera, vb tüm tarayıcılar tarafından desteklendiği için sorunsuz çalışır [57].

3.6.3.1 JavaScript Çalışma Prensipleri

JavaScript kodları için herhangi bir metin editörü kullanılabilir ya da Notepad gibi bir düz yazı programıyla da kodlar oluşturulabilir. Tarayıcıların yorumlayabilmesi için <SCRIPT> etiketiyle başlayan </SCRIPT> etiket ile biten JavaScript kodları HTML belgelerinin çeşitli yerlerine eklenebilir. Eklenen kodlar kullanıcıların HTML belgesindeki bir bağlantıya tıklaması veya klavyede bir tuşa basmasıyla tetiklenir ya da HTML sayfası kullanıcı Browser'ında görüntüldüğü anda otomatik olarak çalışmaya başlar. Otomatik çalışan Javascript kodu ise iki ayrı yöntemle çalıştırılabilir: İlk yöntem HTML kodları işletilmeden önce (yani Web sayfası kullanıcının Web Browser'ında görüntülenmeden önce) çalışması, ikinci yöntem sayfa görüntüldükten sonra script kodlarının çalışmasıdır [58]. Bu yöntemler aşağıda sırasıyla açıklanmaktadır.

- HEAD bölümünde
- BODY bölümünde

HEAD Bölümündeki Scriptler: HEAD etiketleri içine <SCRIPT>, </SCRIPT> etiketi kullanarak istenilen JavaScript kodları eklenebilir. Bu kodlar HTML sayfasının BODY bölümü yüklenmeden işlenir.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript1.2">
alert("Merhaba Dünya!") </SCRIPT>
</HEAD>
<BODY>
<B>Deneme Sayfasına Hoşgeldiniz!</B><br>
</BODY>
</HTML>
```

Yukarıdaki örnekte kaynak kodlar yukarıdan aşağıya doğru işletilirken sayfada ilk görüntülenecek olan alert() komutuyla ekrana yazılan “Merhaba Dünya!” iletişim kutusu olacaktır. Gelen iletişim kutusunda “Tamam” butonuna basıldığında body bölümündeki kodların işlenmesine sıra gelecektir. Ekrana “Deneme Sayfasına Hoş geldiniz!” mesajı yazacaktır.

BODY Bölümündeki Scriptler: HTML belgesinin BODY bölümünde JavaScript kullanmak için iki yol vardır:

Doğrudan çalışacak kodlar ve kullanıcı tarafından çalışacak kodlar.

Doğrudan çalışacak JavaScript kodları <BODY>..</BODY> etiketleri arasına <SCRIPT> etiketi kullanılarak HTML belgesinin istenilen yerine yazılır. Bu cümleler tarayıcı tarafından okunduğu anda yerine getirilir:

```
<HTML>
  <HEAD>
    İkinci Ornek
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="Javascript1.2">
      alert("Javascript dünyasına hoş geldiniz!")
    </SCRIPT>
  </BODY>
</HTML>
```

Yukarıdaki örnekte ise kod işleme sırasına uygun olarak işletilerek önce HEAD etiketleri yorumlanarak ekrana “İkinci Ornek” yazıp hemen ardından iletişim kutusu çıkarak

“Javascript dünyasına hoş geldiniz!” yazacaktır. Kullanılan diğer bir yöntem ise kullanıcı tarafından bir olayın başlatılmasıyla bir JavaScript kodunun çalıştırılmasıdır.

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="Javascript1.2">
// genel değişken tanımları
var n = "Merhaba Dünya!"
//bu fonksiyon uyarı kutusu açar
function merhaba()
{
  alert(n)
}
    </SCRIPT>
  </HEAD>
  <BODY>
<br><br><br><br><B>Butona Tıkladığınızda "Merhaba Dünya!" yazan iletişim kutusu ekrana
gelecektir.</B><br><br>
    <FORM>
<INPUT TYPE="SUBMIT" NAME="dugme1" VALUE="TIKLAYINIZ!"
onclick="merhaba()">
    </FORM>
  </BODY>
</HTML>
```

Yukarıdaki örnekte ise HEAD etiketleri arasında değişken tanımlama ve fonksiyon oluşturma işlemleri yapılmıştır. BODY komutu işletildikten sonra FORM ögesiyle oluşturulan düğme kullanıcı tarafından tıklandığında script kodu tetiklenerek fonksiyon çağrılır ve mesaj kutusu ekrana gelir.

Ayrıca <BODY> içinde URL kullanılarak da aşağıdaki örnek kodla gösterildiği gibi JavaScript kodu çağrılabilir: mümkündür:

```
<A HREF="javascript: alert('Çıkmak istediğinizden emin misiniz?')">Siteden Çık</A>
```

3.6.3.2 Javascript Dilinin Özellikleri

JavaScript dilinin temel söz dizim yapısına aşağıda incelenecektir.

3.6.3.2.1 Değişkenler

Bir değeri içeren bellek adlarını ifade ederler. Değişkenler verileri saklar ve onlara erişimi sağlarlar. Değişkenlerin bir adı olur, bir de değeri olur. Program boyunca değişkenin adı değişmez; fakat içeriği değişebilir.

Bir JavaScript değişkeni belli kurallara uymak zorundadır:

- Değişkenler harfle ya da alt çizgi (_) ile başlamalıdır. Rakam veya diğer işaretler, birinci karakter olarak kullanılamaz, fakat değişken adının diğer karakterlerinde kullanılabilir.
- Javascript, büyük küçük harfe karşı duyarlıdır (case-sensitive özelliği). Aynı kelimenin büyük harfle yazılması ile küçük harfle yazılması farklı isimler olarak algılanmasına neden olur.
- Değişken isimleri içinde boşluk olmamalıdır.
- Kullanıcıların bilgisayarında kullanılan dil setinin ayarları her bir kullanıcı için farklılık gösterebileceği için İngilizce alfabede bulunmayan, Türkçe ve diğer dillerdeki high-ASCII karakterleri (ı, İ, ğ, Ğ, Ş, ş ile ü, Ü, ö, Ö, ç ve Ç) kullanmamakta fayda vardır [58].

3.6.3.2.2 Kapsam

Bir değişkene bir değer atandığında bu değer geçerli olacağı süreyi belirlemek için kapsam (scope) kavramı kullanılır. Bir değişkenin kapsamı global ya da lokal olabilir. Bir global değişkene sayfa üzerinde herhangi bir JavaScript'ten erişilebilir. Lokal değişken ise değer atandığı yordam içinde geçerlidir.

Aşağıdaki örnekte bir değişkenin kendisine atama yaparak değer verilir:

```
adi= "Zeynep";
```

Bir fonksiyon yazılırken, yerel bir değişken oluşturmak için aşağıdaki örnekte olduğu gibi "var" kelimesiyle tanımlama yapılır:

```
function toplama()
{ var sayi=1;
toplama=0;
...deyimler...
}
```

Yukarıdaki örnekte sayi adlı değişken toplama fonksiyonu için yerel, toplam değişkeni ise tüm sayfa için genel (global) değişken olacaktır. Değişkenlere atanan değerler Tablo 3.3'deki değişken türlerinden herhangi biri olabilir. Verinin türüne bağlı olarak değer üzerinde belirli işlemler yapılabilir ya da yapılamaz. Örneğin iki string değer üzerinde aritmetik işlem yapılamaz.

Tablo 3.3 Değişken türleri

Değişken Türleri	
Numbers	Tam ve ondalıklı sayılar içerir. Pozitif, 0 ya da negatif olabilir.
Booleans	Mantıksal veri türüdür. True (Doğru) ya da False (Yanlış) değerini alır. Sorgulama sonunda geriye True için 1, False için 0 değerini döndürür.
Strings	Karakter bilgi içeren bir değişkendir. Tırnak içinde yazılan ifadeler değişkene atanır.
Objects	Tarayıcıya ait nesnelerdir. nesne= new Object(); şeklinde kodlamada kullanılabilir.
Null	Özel bir değişkendir. İçi boş ya da değersiz olarak ifade edilir.
Undefined	Bir değişken yaratıldıktan sonra içerdiği tanımsız değeri simgeler.

3.6.3.2.3 İşleçler (Operators)

İşleçler, iki ya da daha fazla değer üzerinde işlem yapılmasını sağlar. Değerler ya da değişkenler arasında toplama, çıkarma, mod gibi işlemler yapılabilir.

Tablo 3.4 İşleçler ve görevleri

İşleçler ve Görevleri	
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme

%	MOD. Bölmede kalanı verir. (10 % 3 işlemi 1 değerini verir.)
++	Tek artırım. Değer 1 artırılır.
--	Tekli azaltma.
-	Negatifleştirme: İşlenenin negatif değerini döndürür.

3.6.3.2.4 İkili İşlemler

İkili işlemler birden çok karşılaştırma işlemini tek bir koşul ifadesi olarak birleştirirler.

Örneğin:

sayi =1 AND dongu > 1000 ise gibi.

Tablo 3.5 İkili işlemler

İkili işlemler	
&&	Mantıksal Ve: "And" İki ifade de doğruysa True değerini döndürür.
	Mantıksal Veya: "Or" İfadelerden birisi doğruysa True değerini döndürür.
!	Mantıksal Değil: "Not" Negatifleştirme işlemi doğruysa True değerini döndürür.

3.6.3.2.5 Atama Deyimi

Atama deyimi (=), bir değişkene sağdan bir değer atanmasını sağlar. Değişkenlere türlerine ve tanımlamalarına uygun olan herhangi bir değer atanabilir.

Örnek: sayi=100;

Örnek: toplam=(fiyat+kdv)

Sağdaki değeri soldaki değişkene atayan +=, -=, *=, /= ifadeleri de kullanılabilir.

Örneğin: toplam+=fiyat (diğer bir değişkenden gelen fiyat değerini toplam değerine ekler ve sonucu toplam değişkenine atar.)

3.6.3.2.6 İfadeler

Değişkenlerin, işlemlerin ve deyimlerin bir araya gelerek oluşturdukları yapıları ifade eder.

Örneğin: "toplam = sayi 1* sayi2" formülü bir ifadedir.

3.6.3.2.7 Deyimler

Deyimler belli bir komut ya da söz dizimi bileşenlerine verilen addır. Örneğin bir komut ya da bir IF yapısı deyim olabilir:

Örnek: `if (basari_notu>50) {deyimler;} else {deyimler;}`

3.6.3.2.8 Nesneler

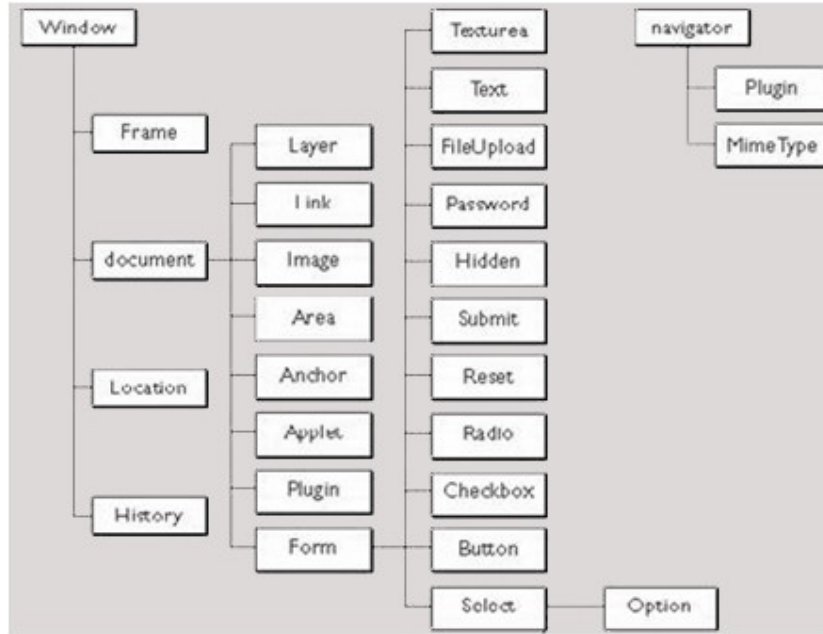
Kendi değerleri, özellikleri ve işlemleri olan bileşenlere nesne (object) denir. JavaScript dilinde çok sayıda yerleşik nesne vardır.

JavaScript, bütün Web sayfasını ve tarayıcıyı nesneler topluluğu olarak görür. Bu sisteme DOM (Document Object Model) denir [59].

JavaScript kullanarak dinamik ve etkileşimli görünüm elde edilir. DOM'a JavaScript ve JScript gibi ECMAScript olan kullanıcı tarafındaki bir scripting dili yardımıyla erişilebilir. DOM'la veri alış-verişi yeni bir sayfada kullanıcı ekranına yansıtılmaz. Parçalı güncelleme olayının büyük kısmı bu modelle gerçekleştirilir.

Başlangıçta, Javascript geliştiricileri kullanıcılarına daha zengin uygulama deneyimleri yaşatabilmek ve sayfalardaki etiketlerde dinamik değişimler sağlanabilmesi için çaba sarf ediyordu. Bu durum sayfaların bir nesne gibi işlenebilmesini gerekli kıldı. Böylece DOM ortaya çıkmıştır. İlk önceleri Javascript ve DOM sıkı bir ilişki içindeydi, sonuçta ise içerik ayrılarak geliştirilmiştir. DOM, sayfaların JavaScript veya VBScript gibi bir betik dili ile değiştirilebilir sayfaların tamamen nesne yönelimli gösterimidir [60].

DOM, Şekil 3.16'da gösterilen JavaScript içindeki bütün nesneleri ifade etmektedir. Hiyerarşik olarak çok sayıda nesneyi içerir. NetScape ve Microsoft tarayıcıları için DOM nesneleri farklı olduğu için yazılan kodlar uyumlu değildir [59].



Şekil 3.16 Javascript nesneleri [59].

Window nesnesi tarayıcının kendisidir. Window nesnesi birçok yönetime, özelliğe ve olaya sahiptir. Form nesnesi DOM nesnesinin ana nesnesidir. Form üzerindeki alanlar bu nesne aracılığıyla kullanılır [59].

Verilerin yüklenmesinde kullanılan XML ile üretilen Javascript istemci kodları ve veri sunucu üzerindeki kodlar arasından ayrılarak gösterilir. Bu istemcinin etkilenmeksizin sunucu üzerinde değişiklik yapılmasını sağlar. Benzer API içeren diğer istemciler için bilgileri açığa çıkarır [41].

3.6.3.2.8.1 DOM (Document Object Model)

Document Object Model (DOM), XML ve HTML/XHTML için World Wide Web Consortium (W3C) tarafından geliştirilen bir belgenin ağaç yapısını (tree-based) temsil eden bir standarttır. DOM bir takım nesneler için belgenin yapısını temsil etmesinin yanı sıra bu nesnelere erişim için yöntemler de sağlar [44].

DOM bir belgenin stilini, kaynak dosyasını, yapısını ve içeriğini günceller ve program ile scriptlerin belgeye erişimine izin veren bir platformdur. Aynı zamanda bir API (application programming interface)'dir [62].

Nesneleri içeren belgeler için standart model olarak DOM üç bölüme ayrılmıştır:

- Core DOM – XML
- HTMLXML DOM - XML

- HTML DOM – HTML

DOM’da her internet sayfası bir belge ve bu belge içerisinde bulunan tüm elemanlar ise nesne olarak kabul edilir. DOM, sayfa üzerindeki her bir XML elemanı için nesnelerin ağaç yapısının resmini görselleştirmenin kolay bir yoludur. Aşağıdaki örnek basit bir XML belgesinin dört düğümden oluşan DOM’u göstermektedir. Kök Doküman düğümü, kök etiketi için bir Element düğümü, alt etiketler için bir Element düğümü ve alt etiketin sahip olacağı herhangi bir bilgi girilebilen bir metin düğümüdür [41].

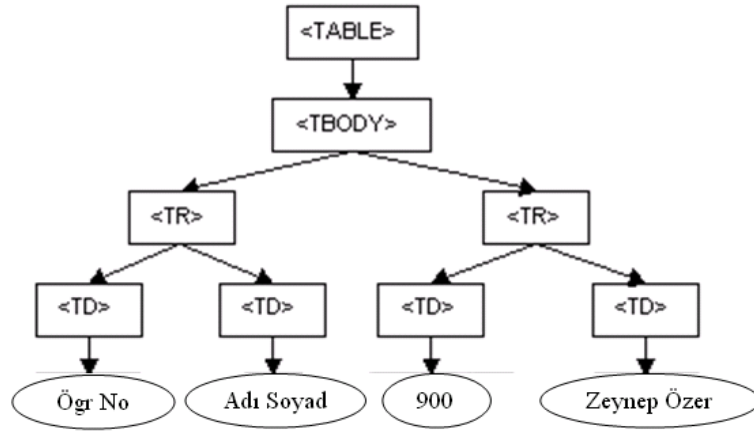
```
<rootTag>
  <childTag>Some Text</childTag>
</rootTag>
```

Bir ağaç yapısının görsel bir temsili aşağıda gösterilmektedir:

```
Doküman
Element (kökEtiket)
Element (altEtiket)
Metin
```

Aşağıdaki kod kümesinin [63] DOM gösterimi Şekil 3.17’deki gibidir.

```
<TABLE border=1>
<TBODY>
<TR>
<TD>Öğr No</TD>
<TD>Adı Soyadı</TD>
</TR>
<TR>
<TD>900</TD>
<TD>Zeynep ÖZER</TD>
</TR>
</TBODY>
</TABLE>
```



Şekil 3.17 DOM ağaç yapısı [63].

Geçerli her HTML ve XML belgesindeki element, nitelik, metin düğümü (text node) hatta boşluk ve sekme karakterleri(whitespace) bile W3C DOM tarafından düğüm (node) olarak kabul edilir. Şekil 3.17'deki gibi belgedeki düğümlerin kullanım hiyerarşisine göre sıralanmasına da düğüm ağacı (node tree) denir. Belge nesnesi yeni düğümler oluşturmak, bu düğümlerin özelliklerini değiştirmek ve yönetmek için çok sayıda yönteme sahiptir.

3.6.3.2.8.2 DOM Elementinin Sık Kullanılan Genel Özellikleri ve Yöntemleri

Aşağıda düğümlere erişmek ve yönetmek için sık kullanılan JavaScript yöntemleri vardır [64]:

- **getElementById()**

Sözdizimi:

```
document.getElementById("herhangiBirID");
```

Örnek kullanımı:

```
element = document.getElementById(kimlik)
```

getElementById() yöntemi, id özelliği belirtilen elementlerin referansının alınmasını sağlar. Aynı id kimliğine sahip başka bir element olması durumunda sadece ilk elementi verir. Diğerlerine erişim sağlamaz. “kimlik” doküman üzerinde elementi bulmak için kullanılacak eşsiz bir özelliğini ifade eder.

- **getElementsByTagName()**

Sözdizimi:

```
document.getElementsByTagName("herhangiBirHTMLElement");
```

Örnek kullanımı:

```
document.getElementsByTagName("etiketAdi");
```

Doküman üzerinde belirtilen aynı etiket ismine ait bütün elementlerin bir listesini verir.

```
document.getElementById("herhangiBirID").document.getElementsByTagName("*")
```

“*” parametresiyle id niteliğinin değeri "herhangiBirID" olan elementin düğümleri içindeki tüm elementler listelenir.

- **getElementsByName()**

Sözdizimi:

```
document.getElementsByName("herhangiBirIsim");
```

Örnek Kullanım:

```
document.getElementsByName("baslik");
```

Bu yöntem, doküman üzerinde belirtilen name özelliğine sahip elementlerin bir listesini verir. Örnek kullanımda name niteliğinin değeri "baslik" olan elementleri verir.

- **firstChild**

Bu özellik, elementin ilk alt elementini verir.

Sözdizimi:

```
document.getElementById("herhangiBirID").firstChild;
```

- **lastChild**

Bu özellik, elementin son alt elemanını verir.

Sözdizimi:

```
document.getElemementById("herhangiBirID").lastChild;
```

- **nodeValue**

nodeValue özelliği, text, comment ve CDATA düğümlerinin içerdikleri değeri verir.

Sözdizimi:

```
m= document.getElementById("herhangiBirID").lastChild.nodeValue;
```

“m” eğer varsa düğümün içediği değerin string olarak değeridir. document nesnesinin ve element'lerin nodeValue değeri null döner.

- **removeAttribute()**

removeAttribute fonksiyonu, elementin belirtilen bir özelliğinin silinmesini sağlar.

Sözdizimi:

```
element.removeAttribute(ozellik)
```

“ozellik” silinecek özelliğin adıdır.

- **nodeName**

Düğüm bilgilerini almak için kullanılır. Geçerli düğümün adını döndürür. Geçerli düğüm element ise, elementin tipini döndürür. Nitelik ise, nitelik adını döndürür.

- **nodeType**

Geçerli düğümün türünü sayısal olarak döndürür.

- **nextSibling**

Düğümmler arası gezinti yapılabilmesini sağlar. Geçerli düğümden bir sonrakine referans verir.

- **previousSibling**

Geçerli düğümün bir öncekine referans verir.

- **setAttributeNode**

Bir düğüme nitelik atamada kullanılır.

Sözdizimi:

```
element.setAttributeNode(nitelik);
```

3.6.3.2.9 Fonksiyonlar

Fonksiyonlar birden fazla adımdan oluşan işlemlerin kolayca tek seferde yapılmasını sağlar. Fonksiyonlar bir dizi işlemi yerine getirir ve bir sonuç döndürür. Ayrıca fonksiyonlar, parametre olarak kabul edilen birçok değer alırlar.

Fonksiyonların HTML dosyasının HEAD bölümüne koyulması, yani gerekli bütün fonksiyonları, alt alta, HTML'in HEAD bölümünde yer alacak bir SCRIPT etiketi içinde oluşturulması tarayıcının Javascript yorumlayıcısı bakımından hız kazanmasını sağlar. Sonra ihtiyaç duyulduğunda yeni bir SCRIPT etiketi koyarak bu fonksiyon göreve çağrılır [58].

```
function fonksiyonun_adi (argüman1, argüman2, ... argümanN){  
işlemler  
    deyim 1  
    deyim 2  
    ...  
    deyim N  
}
```

3.6.3.2.10 Açıklamalar

Programların anlaşılır olmasını sağlayan açıklama satırları JavaScript'te bu <!-- ve --> işaretlerinin arasında ifade edilir [58].

```
<!-- Açıklama satırıdır -->.
```

3.7 Basit Bir AJAX Örneği

Ekran çıktısı Şekil 3.18'de görülen basit bir örnek üzerinde işleyiş anlatılacaktır.



Şekil 3.18 AJAX örneği

index.html dosyası açıldığında “Metnin AJAX’la getirilmesi için tıklayınız” bağlantısına tıklandığında tarayıcı kullanılan AJAX yöntem sayesinde tüm sayfa yeniden yüklenmeden, içeriği sunucu’da depolanan sonuc.htm dosyasından alarak ekrana yansıtır.

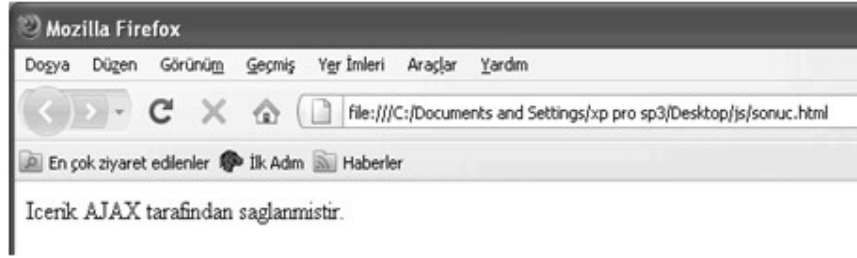
Yukarıdaki index.html dosyasının kodları aşağıda gösterilmektedir.

```
<html>
  <head>
    <title>AJAXla örneği</title>
    <script language="JavaScript" type="text/javascript">
      function getXmlHttpRequestObject() {
        if (window.XMLHttpRequest) {
          return new XMLHttpRequest(); //IE olmayan
        } else if(window.ActiveXObject) {
          return new ActiveXObject("Microsoft.XMLHTTP"); //IE
        } else
          alert("Maalesef tarayıcınız XmlHttpRequest nesnesini desteklememektedir. AJAXı
destekleyen bir tarayıcıda tekrar deneyiniz!");
        }
      }
      var receiveReq = getXmlHttpRequestObject();
      function cevap() {
        if (receiveReq.readyState == 4 || receiveReq.readyState == 0)
          receiveReq.open("GET", 'sonuc.html', true);
          receiveReq.onreadystatechange = handlecevap;
          receiveReq.send(null);
        }
      }
      function handlecevap() {
        if (receiveReq.readyState == 4)
{
document.getElementById('span_result').innerHTML = receiveReq.responseText;
        }
      }
    </script>
  </head>
  <a href="javascript:cevap();">Metnin AJAX'la getirilmesi için tıklayınız!</a><br />
```

```
<span id="span_result"></span>
</html>
```

Örnekte ilk olarak tüm tarayıcılarda sorunsuz çalışabilmesi için çapraz tarayıcı desteği sağlanmıştır. “receiveReq” değişkeni getXmlHttpRequestObject() ile tanımlanmıştır. Daha sonra cevap fonksiyonu oluşturulmuştur. Sunucudan gelen cevabın durumu if...else yapısıyla kontrol edilmiştir. Sunucudan gelen cevaba göre döndürülen değerlerle sonuc.html sayfası çağrılarak uygulama sonlanır.

AJAX yöntemi ile çekilen bilginin tutulduğu sonuc.html dosyasında ise sadece Şekil 3.19’de yazan veri bulunmaktadır.



Şekil 3.19 sonuc.html sayfasının ekran çıktısı

3.8 XML

XML (eXtensible Markup Language), internet’te teknik standartlar ve yönetim konusunda yetkili organ olan W3C (World Wide Web Consortium) organizasyonu tarafından geliştirilmiştir. XML, programcıların kendi sistemlerini oluşturabilecekleri, kendi etiketlerini tanımlayarak çok daha rahat ve etkin programlama yapabilecekleri ve bu belirlenen etiketleri kendi yapıları içerisinde standartlaştırabilecekleri esnek, geliştirilebilir ve kolay uygulanabilir bir meta-dil (bir dili tanımlamak için kullanılan sözcükler, dilbilim dili)’dir [65].

XML ile ilgili ilk ve en önemli nokta başlı başına bir dil olmamasıdır. Daha ziyade, diğer dilleri ya da bir dili oluşturan sözcükleri oluşturmak (vocabularies) için kullanılan bir meta-dildir. XML bu sözlüklerin oluşturulması için gerekli kuralları tanımlar [66].

XML bu kural yapısına uygun olarak etiketleri içerik bilgisiyle tanımlayabilir. XML’nin esnek yapısı sayesinde aşağıdaki örnekte olduğu gibi, kullanılan dilde çok farklı tiplerde etiket oluşturulmasına müsaittir. XML sözlüğündeki tek kısıtlaması etiketlerin iyi biçimlendirilmiş olması gerektiğidir [66].

```
<not>
<kime> Melek </ kime>
<kimden> Zeynep </ kimden>
    <baslik> Hatirlatici </baslik>
    <body> Bana proje teslim tarihini hatirlat! </ body>
</not>
```

Yukarıdaki not örneği kendi kendini açıklayıcı bir örnektir. Gönderici ve alıcı bilgisinin yanında mesaj metni ve başlığı da mevcuttur. XML'nin HTML'de olduğu gibi önceden tanımlanmış etiketleri yoktur. XML yazarı kendi etiketlerini ve kendi belge yapısını örnekteki gibi oluşturabilir.

Her dil farklıdır, ama hepsinde içeriği işaretlemek için etiketler kullanılır. Etiket adlarının seçimi ve yapısı esnektir ve genellikle gruplarda bilginin paylaşılabilmesi için standart XML sözlükleri kararlaştırılmıştır [66]

XML, 1998 yılında SGML alt yapısı üzerine kurulmuştur. SGML (Standard Generalized Markup Language), bir etkinleştirme teknolojisidir. HTML gibi uygulamalarda kullanılır. Genişletilebilir İşaretleme Dili (Extensible Markup Language-XML) 'nin oldukça basit bir diyalektidir [66].

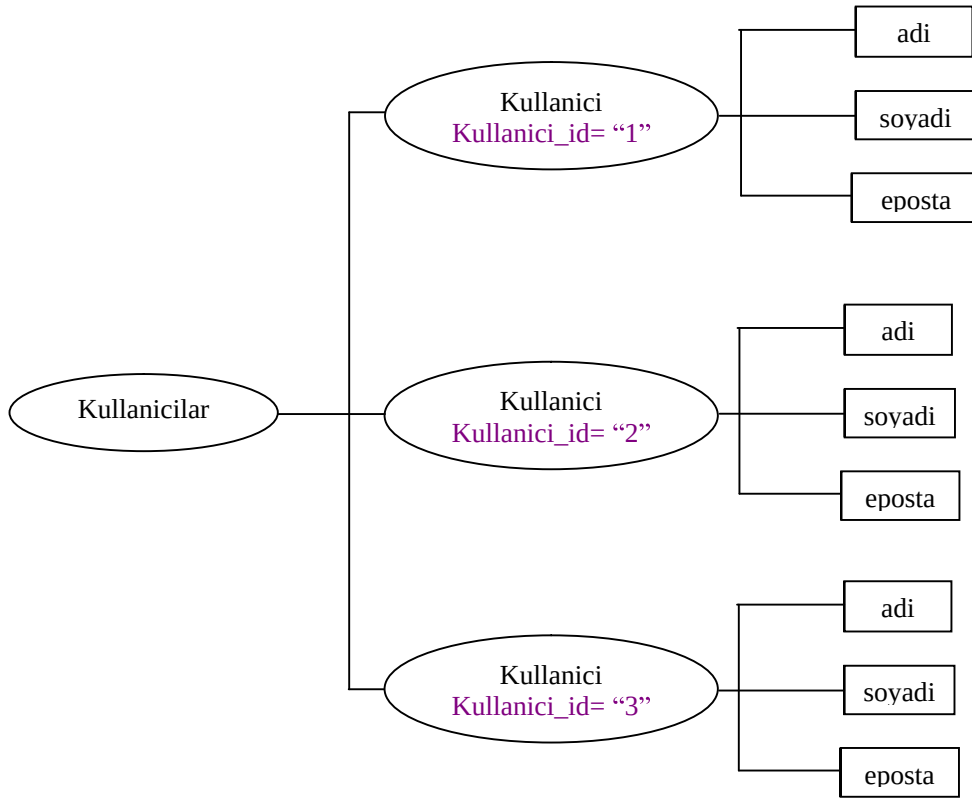
XML, SGML ve HTML sistemlerinin her ikisinin de uygulanmasını kolaylaştırmak için tasarlanmıştır. XML aynı zamanda son derece popüler bir veri formatıdır. Çünkü çoğu programlama dili tarafından desteklenir. Bu dil sayesinde gelişigüzel veri tipleri tanımlamak kolaydır. Çoğu kez XML, Uzak Yordam Çağrısı - Remote Procedure Call (RPC) mekanizması sayesinde istemci-sunucu iletişimini kolaylaştırmak için kullanılmıştır [66]. XML, elde edilmek istenen alan bilgileri, konfigürasyon parametreleri, teknik çizim, hesap çizelgesi (spreadsheet) gibi öğeleri içeren dosyaların yazılımında kullanılır [67].

XML verileri görüntülemek için değil verilerin platformdan bağımsız bir şekilde depolanması ve transferi için tasarlanmıştır. XML tüm bilgiler düz metin formatında saklar. Bu da tüm platformlarda sorunsuz bir şekilde kullanılabilir olmasını sağlar. Bu dosyaları diskte düz metin olarak tutmanın bir avantajı da, oluşturulduğu programda açılması gerekli değildir. Bu yazı dosyaları programcının istediği bir yazı işleme programında açılıp düzenlenebilir. Ayrıca XML, veriyi düz metin (plain text) formatında saklamasından dolayı, veriyi paylaşırma konusunda hem yazılım hem de donanımdan bağımsız hareket edebilme imkânını sunmuştur.

3.8.1 Örnek XML Belgesi Oluşturma

XML belgelerinin yapısının daha iyi kavranabilmesi için basit bir örnekten yola çıkılır. Bir dizi karakter mantıksal bir yapıya çevrilmelidir. Bu çevrim ağaç yapısı kullanılarak yapılabilir. Örnek belgede kök dizin ya da element ve alt bölümler için de dallar belirlenir. deneme.xml olarak kaydedilecek belgenin içeriğinde sisteme kayıtlı kullanıcıların bilgileri vardır.

Hazırlanan deneme.xml belgesinin ağaç yapısı aşağıdaki gibi oluşturulmuştur. Kök element ve alt dalları ve alt dala ait nitelik belirtilmiştir.



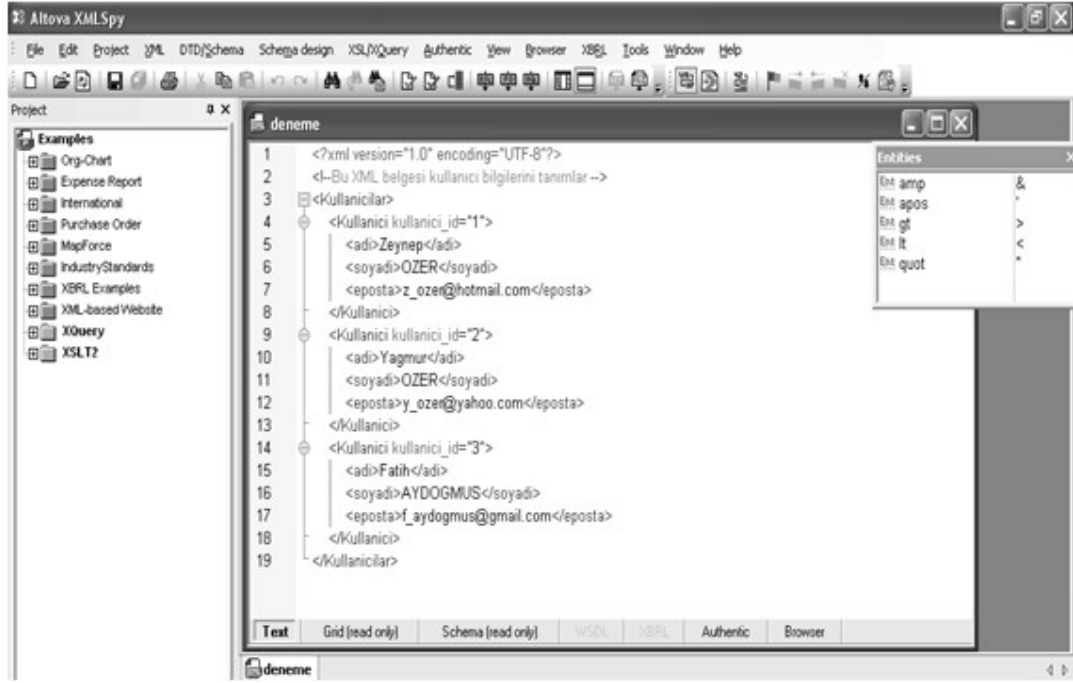
Şekil 3.20 deneme.xml dosyasının ağaç yapısı

Ağaç yapısı oluşturulduktan sonra kodlama kısmına geçilir. XML’de kod yazımını kolaylaştırmak için Altova XMLSpy, MoreMotion XML Editor, XMLwriter XML Editor 2.6 gibi çeşitli editör programları kullanılabilir. Bu tez çalışmasında oldukça işlevsel özelliklere sahip Altova XMLSpy yazılımı kullanılmıştır. Editör kullanarak dokümandaki metinlerin yapısal elemanları yazı özellikleri (renk, stil, biçim, vb) belirginleştirilir. İmlerin otomatik

olarak kapanması sağlanabilir. Oluşturulan yapıların doğruluğu ve geçerliliği kolayca kontrol edilebilir. Altova XMLSpy editörüyle oluşturulan belge aşağıdaki kodlardan oluşmaktadır.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Bu XML belgesi kullanıcı bilgilerini tanımlar -->
<Kullanicilar>
  <Kullanici kullanıcı_id="1">
    <adi>Zeynep</adi>
    <soyadi>OZER</soyadi>
    <eposta>z_ozar@hotmail.com</eposta>
  </Kullanici>
  <Kullanici kullanıcı_id="2">
    <adi>Yagmur</adi>
    <soyadi>OZER</soyadi>
    <eposta>y_ozar@yahoo.com</eposta>
  </Kullanici>
  <Kullanici kullanıcı_id="3">
    <adi>Fatih</adi>
    <soyadi>AYDOGMUS</soyadi>
    <eposta>f_aydogmus@gmail.com</eposta>
  </Kullanici>
</Kullanicilar>
```

Hazırlanan doküman iyi biçimlendirilmiş (well-formed) bir yapıya sahiptir. Şekil 3.21’ de Altova XMLSpy editörüyle oluşturulan belgenin ekran görüntüsünde deneme.xml görülmektedir.



Şekil 3.21 XML belgesi

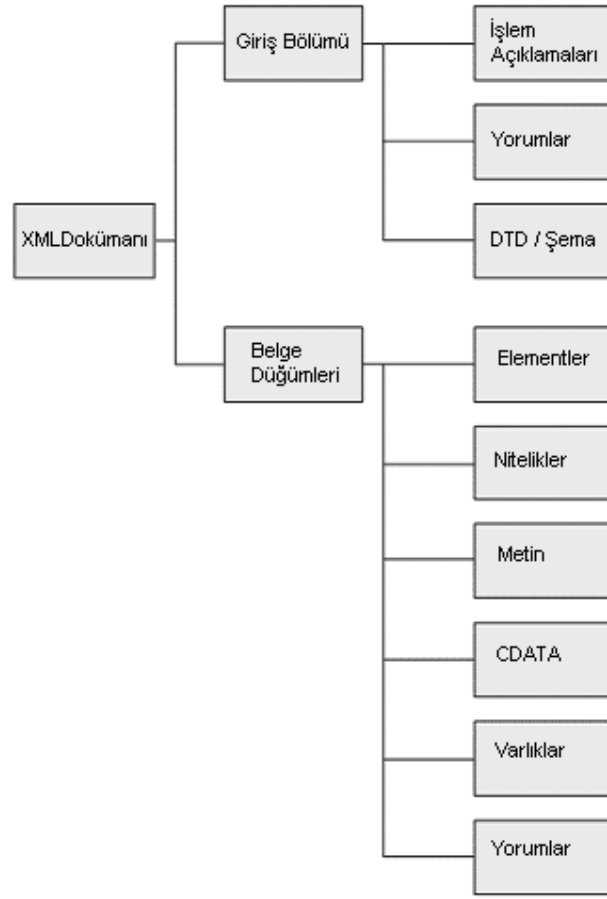
3.8.2 XML Dokümanının Yapısı

XML ile veriler yapı bakımından modülerlik kazanmaktadır. Bu sayede verilerin içerik, yapı ve sunum kısımları ayrı modüller halinde farklı XML dokümanlarında tutulmaktadır.

Her XML belgesini oluşturan modüller iki bölümde incelenecektir. Bu modüller giriş bölümü (prolog) ve belge ya da kök düğüm (document node) olarak ayrılmıştır.

Prolog XML belgesinin başında görüntülenir ve belge hakkında içerikle ilgili bilgiler içerir. XML bildirimleri işlem açıklamaları, yorumlar ve DTD (Document Type Definition)/XML Şema içerebilir.

İyi düzenlenmiş bir XML dokümanı tek ve temel bir kök düğüm üzerine kurulur. Elementler, nitelikler, metin, CDATA (character data), varlık ve yorumlar isteğe bağlı olarak eklenebilir.



Şekil 3.22 Örnek bir XML doküman yapısı [66].

3.8.2.1 Giriş Bölümü (Prolog)

Bu bölümde XML belgesinde bulunabilecek ön tanımlama konuları incelenecektir.

3.8.2.2 XML Bildirimleri

XML dokümanları genellikle aşağıdaki bildirim satırıyla başlar. Bildirim satırından önce başka komut satır yazılmamalıdır. Tersi bir durumda XML dokümanı içeriği işleme koyarken doğru bir şekilde okuyamaz ve hata mesajı üretir.

```
<?xml version= "1.0" encoding= "UTF-8" ?>
```

Bu işlem satırında başlangıç etiketi ve bitiş etiketi yoktur. XML bildiriminin yapıldığı `<?xml` ve `?>` işlem komut etiketleridir. İşlem komutları örnekteki gibi bitişik yazılır. Bu örnek satırda 2 tane öznitelik barınır. Birinci öznitelik xml dokümanının hangi sürüm olduğunu bildirir ve yazılması zorunludur. En yaygın kullanılan ve en güncel sürümü olan "1.0" yerine başka bir

değer yazıldığında dosya işlenmeyebilir. İkinci öznelik olan encoding belge işlenirken kullanılacak dil seçiminin belirtildiği alandır. Örnekteki kod birçok dili destekleyen UTF-8 karakter dil setini tanımlar.

3.8.2.2.1 İşlem Açıklamaları (Processing Instructions)

Giriş bölümünde diğer uygulamalara XML dokümanı hakkında bilgi veren işlem açıklamaları (Processing Instructions- PIs) dâhil edilebilir. İşlem açıklamaları <? karakterleriyle başlar ve ?> karakterleriyle biter. PIs XML belgesinin diğer bölümlerinde görülebilmese rağmen genellikle giriş bölümünde görülür.

XML bildirimleri <?xml karakter grubuyla başlar. İşlem açıklamalarıyla benzerlik gösterir. Fakat aralarındaki fark XML bildirimleri <?xml karakter grubuyla başlarken, işlem açıklamaları ise <? karakterleriyle başlar.

Bir XSL stylesheet'i referans gösteren PI gösterimi:

```
<?xml-stylesheet type="text/xsl" href="stylesheet.xml"?>
```

PI'nin ilk ögesi bir isimdir. PI hedefi olarak adlandırılır. Yukarıda gösterilen PI xml-stylesheet ismine sahiptir. İsimler <?xml karakter grubundan “-” sembolüyle ayrılır. PI'ler type= “text/xsl” href= “stylesheet.xml” metin karakter dizesine sahiptir. Örnek kod satırında stylesheet adında metin içerikli bir XSL dosyası referans gösterilmiştir.

3.8.2.2.2 Yorumlar (Comments)

Yorum satırları XML belgesinin hemen hemen her yerinde isteğe bağlı olarak kullanılabilir. XML yorum satırları XHTML belgelerindeki yapıyla aynıdır. Aynı karakterlerle başlarlar.

```
<!--Bu alana yorumlar yazılabilir. -->
```

Yorum satırları XML belgesi işlenirken etkilenmezler. Bu satırlar programcılarının ya da kullanıcıların program kodlarını kolayca anlayabilmeleri için yazılır [68].

3.8.2.2.3 Doküman Tipi Tanımlama-The Document Type Definition (DTD) ve XML Şemalar

XML belgesinin standartlaştırılması için kullanılan bir yapıdır. XML belgesinin hangi elementleri içereceğini ve bu elementlerin özniteliklerinin ne olacağını, elementlerin neleri taşıyacağını, doküman içerisindeki tekrar şeklini ve sayısını kurallarla belirtir [69].

3.8.2.2.3.1 DTD kullanmanın Avantajları

Veri depolama XML'nin temel amaçlarından birisidir. Örneğin yazılım geliştirecek bir proje ekibi çalışırken ortak bir DTD'nin oluşturulmasıyla gelişigüzel yeni etiketler eklenmesinin, etiketlerle depolanan bilginin hatalı sıralanmasının, etiketlere hatalı nitelikler eklenmesinin önüne geçilmiş olur. Ayrıca işlem yapan programın da bu belgeleri tanıması ve işlemesi kesinlik kazanır [70]. Belge işlenirken geçerlilik derleyicileri (validating parsers) XML belgesinin DTD geçerliliği sağlanana kadar tekrar tekrar kontrol eder. Eğer geçerlilik kontrolünden geçemezse, derleyici hata mesajı verir [66].

İçeriğin bir standarda oturtulmasıyla AJAX ve dinamik DOM manipülasyonunda istenmeyen gereksiz kod parçalarının ve çalışmayan işaretlemelerin önüne geçilir. Ayrıca daha hızlı HTML derleyicileri ve daha iyi çapraz tarayıcı performansı için gelişmiş arama motoru optimizasyonu sağlanır. Bir DOCTYPE belirlemenin en önemli etkisi HTML içeriğinin nasıl ayrıştırıldığını ve yorumlandığının Web tarayıcıya gösterilmesi gerektiğini belirtmesidir. Ayrıca DOCTYPE bağlı olarak, CSS yorumlanabilir ve HTML farklı şekilde görüntülenir [71].

Hangi işlem moduna dâhil olduğu tespit edilerek Web tarayıcıları DOCTYPE dönüşümünü destekler. Web tarayıcıları bir Web sayfasının nasıl HTML ve CSS içeriğini yorumladığını ve değerlendirdiğini sayfaların DOCTYPE bildirimlerine dayanarak değiştirir [71].

3.8.2.2.3.2 DTD tanımlama

DTD oluşumu iki yolla yapılabilir. Birinci yol belge içerisinde aşağıdaki örnekte olduğu ilk standart satırdan sonra <!DOCTYPE ile başlayan ve > işareti ile biten aralıkla DTD tanımı yapılmasıdır. DTD alanında bir kok_element_adi etiketi tanımlanmıştır. “[]” işaretleri arasında içsel olarak tanımlanmıştır. Bu etiketin içerisine element_icerigi olarak alt alanlar belirtilir. XML dokümanının içerik kısmına tanımlanan kok_element_adi etiketleri arasına istenilen metin girişi yapılır.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE kok_element_adi
    [ <!ELEMENT kok_element_adi (element_icerigi) > ]
>
< kok_element_adi > Bu basit bir DTD'li XML ornegidir. </ kok_element_adi >
```

İkinci yol ise Web sayfalarının içeriği bir geçerlilik servisi kullanılarak onaylanmasıdır. Doğruluğu W3C1 tarafından garantiye alınan, bir DTD ile geçerlilik kontrolü sağlanabilir [71]. HTML, XHTML, SMIL, MathML,vb Web dokümanlarının <http://validator.w3.org/> adresinden URI adresi, dosyaları upload edilerek ya da direk işaretleme girişi yapılarak geçerlilik kontrolü yapılabilir [72].

DTD tanımı aşağıdaki örnekteki gibi HTML belgesine harici olarak <http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd> adresindeki hazır kural yapısı dâhil edilir. Kural akışı bu adresten sağlanır [72]. Aşağıdaki kod satırında XML veri tanımlamasına göre `<code></code>` etiketleri oluşturulmuştur.

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<body>
<p> Bu alana serbest metin girişi yapılabilir.
    <code>
        Bu alanda ise XML veri tanımlaması DTD tanımına göre yapılmıştır.
    </code>
</p>
</body>
</html>
```

Programcıların en çok ihtiyaç duyabileceği 6 tane öncelikli DOCTYPE belgelerinin listesi aşağıda verilmiştir [72]:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
```

```
"http://www.w3.org/TR/html4/frameset.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

3.8.2.2.3.3 DTD İçinde Element Tipi Bildirimleri

Elementler hakkında bilgi veren kodlardır. Kullanılan yapı !ELEMENT ile başlar element adı ve element içeriğinin eklenmesiyle sonlanır.

```
<!ELEMENT elementName (elementContents)>
```

Element içeriği aşağıdaki gibi boş olarak ta tanımlanabilir.

```
<!ELEMENT elementName (EMPTY)>
```

Element içeriğinde alt dallar-düğümmler oluşturulacaksa aşağıdaki örnekte olduğu gibi DVD kök düğümüne “,” le ayrılarak sırasıyla baslik, format, tarz element isimleri yazılır. Belirlenen sıralama XML belgesi içinde görünecek sıralamayı da oluşturur.

```
<!ELEMENT DVD (baslik, format, tarz)>
```

Parsed Character Data (PCDATA) veri türü element içeriğinin metinsel ifade olacağını bildirir. XML ayrıştırıcısı bu metindeki karakter ve varlık referanslarını çözümler. Aşağıda <baslik>, <format>, <tarz> bildirimlerinin içerik türü PCDATA olarak tanımlanır:

```
<!ELEMENT baslik (#PCDATA)>
<!ELEMENT format (#PCDATA)>
<!ELEMENT tarz (#PCDATA)>
```

XML belgelerinde tanımlanan bir element herhangi bir veri kombinasyonunu içerebilir. Bu durumda içerik türü ANY olarak tanımlanır. En serbest içerik tanıımıdır. Örnek kullanım aşağıda gösterilmektedir.

<!ELEMENT KATALOG ANY>

Elementlerin alabilecekleri nitelikler DTD içerisinde tanımlanır. Element içerikleri hakkında daha detaylı bilgiye ulaşılmasını sağlar. Tablo 3.6’da kullanılabilecek niteleyiciler gösterilmektedir.

Tablo 3.6 DTD karakterleri ve açıklamaları

Karakter	Açıklama
*	Elemandan sıfır veya daha çok bulunabilir
,	Alt elemanları sırayla belirler
	Bir grup element arasında seçim yapılmasına izin verir
+	Elemandan bir veya daha çok bulunabilir
?	Elemandan sıfır veya 1 tane bulunabilir
	Sembol olmaması o elementin bir kez görünmesi gerektiğini bildirir

Örnek gösterimde KULLANICI elementine ait alt element olan AD ögesi birden fazla değer alabileceğini gösterir. TEL? bildirimi ile kullanıcıya ait diğer bilgiler girilirken telefon numarası girişi eksik olarak kalabileceğini ve sonradan eklendiğinde ise yeni kural tanımlamasına gerek kalmadan etiket tanımlamasının yapılabileceğini bildirir.

<!ELEMENT KULLANICI (ADI+, TEL?)>

3.8.2.2.3.4 DTD İçinde Nitelik Liste Bildirimleri

Nitelik bildirimleri element bildirimlerinden sonra görünür ve biraz daha karmaşık bir yapıya sahiptir. Bir elementin sahip olduğu özellikler bir liste halinde belirtilebilir.

Nitelik liste bildirimi aşağıdaki yapıyla gösterilir.

<!ATTLIST Element_adi Nitelik_adi Nitelik_tipi varsayılan_deger>			
↓	↓	↓	↓
<!ATTLIST DVD	id	CDATA	#REQUIRED>

Kod parçasında yazılan varsayılan_deger niteliğın gerekli olup olmadığını belirtir. Nitelik tipleri Tablo 3.7'deki değeri alabilir

Tablo 3.7 DTD içindeki niteliklerin değeri ve açıklamaları [73].

Değer	Açıklaması
CDATA	Değer karakter verisi olmalı
(en1 en2 ..)	Değer parantez içindeki değerlerden birisi olmalı
ID	a unique identifier- benzersiz bir tanımlayıcı-kimlik
IDREF	Değer başka bir elemanın ID si olmalı
IDREFS	Değer başka ID 'lerin listesi olmalı
NMTOKEN	Değer geçerli XML adı olmalı
NMTOKENS	Değer geçerli XML adı listesi olmalı
ENTITY	Değer bir varlık olmalı
ENTITIES	Değer bir varlık listesi olmalı
NOTATION	Değer bir notasyon olmalı
xml:	Değer önceden tanımlamış bir XML değeri olmalı

Varsayılan değeri ise Tablo 3.8'deki değeri alabilir.

Tablo 3.8 Varsayılan değeri ve açıklamaları

Değer	Açıklama
deger	Niteliğın Default değeri
#REQUIRED	Nitelik değeri Eleman içinde geçmek zorundadır.
#IMPLIED	Nitelik değeri Eleman içinde geçmek zorunda değildir.
#FIXED value	Nitelik değeri sabittir. Eğer nitelik değeri belirtilirse default değeri yazmak zorunludur.

Aşağıdaki örnekte URUN elementinin RENK niteliğı seçimli olarak kırmızı, yeşil ya da mavi olası değeri tanımlanmıştır. Varsayılan değeri ise kırmızı olarak belirtilmiştir [73].

```
<!ATTLIST URUN RENK (red|green|blue) "red">
```

3.8.2.2.3.5 DTD İçinde Varlık (Entity) Bildirimleri

Varlık bildirimleri XML belgesinde kullanılan değişik tür ve biçimlerdeki verilerin XML belgesine aktarılmasını sağlar. Diğer programlama dillerinde kullanılan modül ya da makrolar gibi sıklıkla kullanılan metin blokları varlık olarak tanımlanarak sayfanın istenilen yerine kolayca eklenebilir. Eklenen metin üzerindeki değişiklikler varlık bildirimiyle düzenlenebilir [73].

Ayrıca dışarıdan bir dosyayı varlık olarak tanımlayarak içindeki veri XML belgesinde kullanılabilir. Bu veri metin olmak zorunda değildir. DTD bir varlık olarak tanımlamak için aşağıdaki satırın kullanılması gerekmektedir [73]:

```
<!ENTITY varlık_adı "varlık_degeri">  
<!ENTITY copyright "Copyright 2009 Yayınevi">
```

Harici bir dosyadan alınacak bir varlık için söz dizimi ise aşağıdaki gibidir.

```
<!ENTITY varlık_adı SYSTEM "URI/URL">  
<!ENTITY tableOfContents SYSTEM "entities/deneme.xml">
```

(Not: URL = Uniform Resource Locator / URI = Uniform Resource Identifier)

3.8.2.2.4 XML Şeması

XML Şeması, DTD'lere benzer şekilde XML belgesinin yapısını ve bulundurabileceği verilerin türünü tanımlamakta kullanılır. XML şemaları .xsd dosya uzantılıdır. Şema, XML belgesinde kullanılabilecek öğeleri, nitelikleri ve veri türlerinin yanı sıra, XML belgesinin söz konusu XML Şeması için geçerli olmasını sağlamak amacıyla uyulması gereken yapıyı da tanımlar [74].

XML şemalar DTD yapılarına göre daha güçlü bir yapıya sahiptir. DTD içerisinde, eleman içeriği string ve diğer birkaç veri tipi ile sınırlıdır. XML şema integer, floating point sayılar, date, time gibi daha geniş bir veri tipi desteği sağlamaktadır. XML şema ayrıca açık içerik modeli (open content model) ve isim uzayı (namespace integration) gibi ek özellikler de sunmaktadır. XML şema belgeleri oluşturulması sırasında XML sözdizimi kullanılır. Bu DTD lere göre çok belirgin bir üstünlük sağlar. Çünkü yeni bir söz dizimi öğrenilmesi yerine, XML'in XML şemaya özel komutlarının öğrenilmesi yeterli olur [75].

Aşağıdaki örnek şema tanımlamasında xs: schema elementi ile başlayan bir kök öge vardır.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  .../...
</xs:schema>
```

Şema elementi iki tane özelliğe sahiptir:

- name
- xmlns

Şema için bir isim tanımlaması name özelliği ile sağlanır. xmlns özelliğinin ise oldukça önemli bir görevi vardır. Bu özellik sayesinde örneğin Microsoft'un schema belirtimlerini kullanabilmesi için "urn:schemas-microsoft-com:xml-data" değerine ayarlanması gerekir. Aşağıdaki örnekte XML şema belgesi için name ve xmlns niteliklerinin basit kullanımı gösterilmektedir [66, 75].

```
<schema name="kutuphane" xmlns= urn:schemas-microsoft-com:xml-data">
<! - Şema içeriği buraya yazılır- ->
</schema>
```

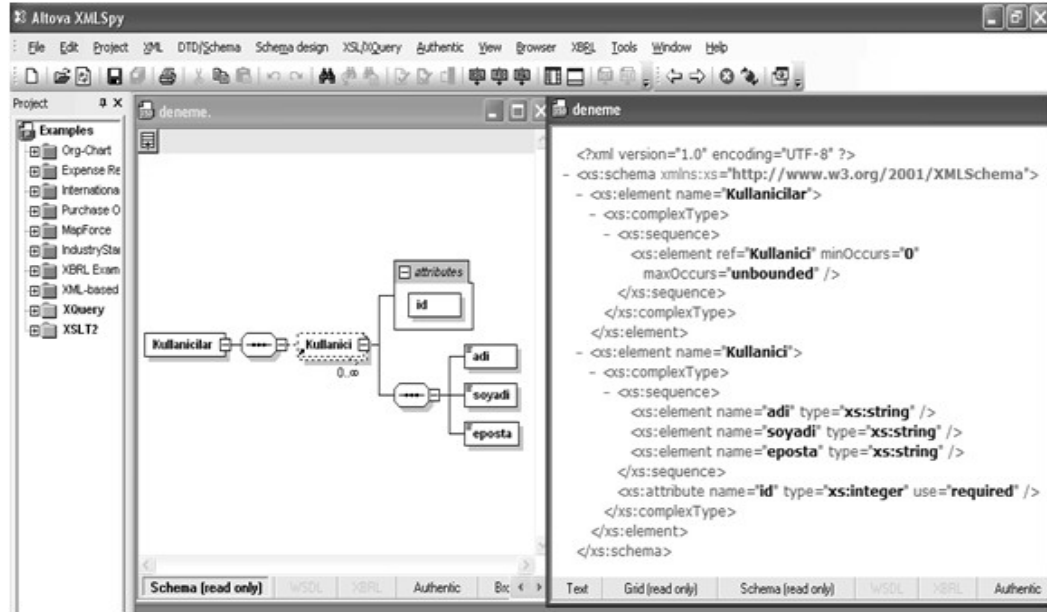
Örnek XML belgesi oluşturma başlığı altında verilen örneğin şema dosyası olan deneme.xsd'nin kodları ve ekran görüntüsü Şekil 3.23'deki gibidir.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Kullanıcılar">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Kullanici" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Kullanici">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="adi" type="xs:string"/>
        <xs:element name="soyadi" type="xs:string"/>
        <xs:element name="eposta" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:integer" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

</xs:complexType>
</xs:element>
</xs:schema>

```



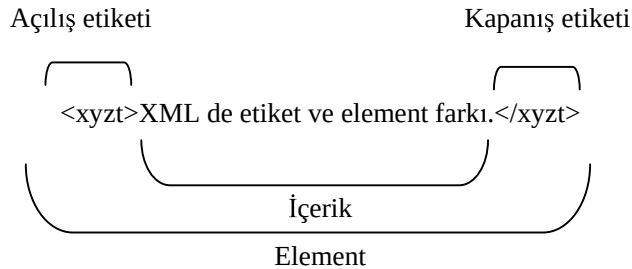
Şekil 3.23 deneme.xsd'nin ekran görüntüsü

3.8.2.3 Belge Düğümleri

XML dokümanlarındaki belge düğümleri aşağıda incelenecektir.

3.8.2.3.1 Elementler

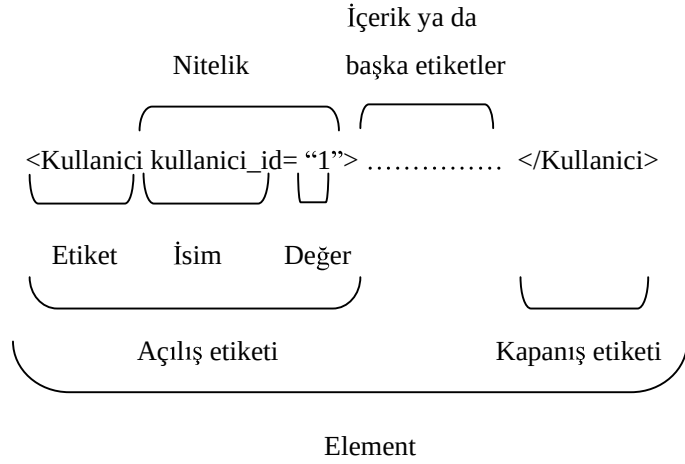
XML dokümanlarından bahsederken etiket ve element terimleri kullanılmaktadır. İlk başta birbirinin yerine kullanılabilir gibi görünse de aslında farklı iki terimdir. Element terimi herhangi bir içerikle birlikte etiketlerin açılıp kapanması olarak tanımlanabilir. Etiket elementin bir parçasıdır. Etiketler de köşeli parantez açılarak başlanır ve sonunda kapatılarak bitirilir. Aşağıdaki örnek satırda <xyzt> etiketlerini içeren tam bir element görülmektedir.



3.8.2.3.2 Nitelikler

Her XML belgesi karakter verisi ve biçimlendirmeden oluşur. Genellikle karakter verisi başlangıç ve bitiş etiketleri arasında metinsel bilgi içerir.

XML etiketlerine bir ya da daha fazla nitelik tanımlanabilir.



3.8.2.3.3 Metin

Açılan ve kapatılan tüm etiketlerin içeriğini oluşturur. CDATA gibi metin işaretlemesi yapılmadıkça, tüm metin XML kural yapısına göre işlenecektir [66].

3.8.2.3.4 CDATA Bölümü

CDATA, XML yapısı içerisinde işlenmeyecek metinleri bloklar halinde işaretlemeye yarar. XML derleyici bu bölümde metin içerisinde yer alan işaretleme ayraçlarını ve etiketlerini normal karakterler olarak algılar. CDATA yapısı <!CDATA[ile başlar ve]] köşeli parantezlerin kapanmasıyla sona erer.

```
<baslik><!CDATA[ ... ]></baslik>
```

CDATA bölümleri Javascript gibi gömülü kodları XML belgelerinde kullanabilir ve bu bölümlere işleme ihtiyaç duyulmayan içerik eklenebilir. Örneğin bir uygulamada veritabanından veri okuması ve işaretlemesi, bütün içeriğin XML belgesine gömme bilgisini vermede karakterler işlenirken açıkça ayrılmasının önüne geçebilmek için CDATA bölümleri kullanılır [66].

3.8.2.3.5 Varlıklar

Bir karakteri temsil eden sembol grubuna varlık denir. Örneğin Ampersand işareti yerine yani “ve” yerine kullanılan “&” ya da bir karakterlik boşluk bırakmak için nonbreaking space yerine “ ” sembolü kullanılır.

```
<matematik>18 < 36</matematik>
```

Yukarıdaki kod satırında bu element işlenmeye çalışıldığında hata iletisi verecektir. Çünkü 36 değerinin önündeki etiket açma işareti olarak algılanacak ve kod satırının işleme alınmasında yanlışlığa neden olacaktır. Bu hatayı düzeltmek için kod satırının aşağıdaki gibi yazılması yeterli olacaktır.

```
<matemetik>18 &lt; 36</matemetik>
```

Tablo 3.9’da sık kullanılan karakterlerin varlık sembolleri gösterilmektedir.

Tablo 3.9 XML Belgelerinde kullanılan varlık bildirimleri

Karakter	Varlık
&	&
'	'
>	>
<	<
"	"

3.8.3 XML’nin Söz Dizim Kuralları

XML içeriğin işaretlenmesi için etiketleme dili kullanılır. Bu etiketlemeler yapılırken bazı kurallar göz önünde bulundurulur. Şimdi bu kurallar incelenecektir.

- 1) XML dili büyük küçük harfe karşı duyarlıdır. XML’de küçük harf kullanımı esastır. Örnek yazım aşağıdaki gibidir [76].

```
<table WIDTH= “100%”> kullanımı yanlıştır.  
<table width= “100%”> kullanımı doğrudur.
```

- 2) Her etiket kapatılmalıdır. Örneğin <kelime> etiketini takip eden satır sonuna </kelime> etiketi de eklenmelidir.
- 3) Dokümanda bütün elementleri içeren bir kök eleman olması gerekir. Örnek yazımda olduğu gibi alt öğeler düzenli bir yapıda yazılmalıdır [76].

```
<root>
<child> ... </ child>
    <subchild> ... </ subchild>
</ root>
```

- 4) Etiketler belirlenirken bir element başka elementleri içine alacak şekilde tasarlanabilir. İçten dışa doğru etiketlerin hepsi kapatılmalıdır.

```
<b><i>Bu metin koyu ve eğik</b></i> yanlış kullanım
<b> <i> Bu metin koyu ve eğik </ i> </ b> doğru kullanım
```

- 5) XML elementleri özel karakter verileri, boş elementler, CDATA ya da yorumlar barındırabilir.
- 6) Öznitelik değerleri tırnak içine alınmış olmalıdır. Tek tırnak ya da çift tırnak olabilir.

```
<table width=100%> kullanımı yanlıştır.
<table width= "100%"> kullanımı doğrudur.
```

- 7) Element adları harfler, rakamlar, tire, alt çizgi, iki nokta işaretlerini ve noktaları içerebilir. İki nokta işareti sadece namespace kullanımı içindir, xml ile başlayan eleman isimleri önceden tanımlanmış özel isimlerdir [86].

```
<kullanilabilirIsimler>
    <isim/>
    <xsl:copy-of/>
    <Uzun_bir_isim/>
    <Noktalarla.ayrilmis.bir.isim/>
    <h823323123-231-231/>
    <_23/>
</kullanilabilirIsimler>
```

- 8) Etiketleme yapılırken rakamla başlamamasına dikkat edilmelidir. Boşluk yer almamalıdır.

3.8.4 XML Dokümanlarının İşleyişi ve Sunumu

XML işlemleri ağaç tabanlı ve olay tabanlı olmak üzere iki kategoriye ayrılır. Verilerin işlenmesinde XML parser yani ayrıştırıcı kullanılır. XML işlemciler genellikle XML ayrıştırıcı olarak da adlandırılır. Ağaç tabanlı ayrıştırıcılar, DOM (Document Object Model) yaklaşımına göre, olay tabanlı ayrıştırıcılar ise SAX (Simple API for XML) yaklaşımına göre veriyi şema yapısı ile birlikte XML'ye özgü hiyerarşik yapıda işler. İşlenen veriler uygulama ve sorgulamalara hazır hale getirir [66, 76].

Genellikle bir XML dokümanını XML formatlı veri içerik dosyası, XSD (DTD yerine) formatlı şema dosyası ve XSL (CSS yerine) formatlı stylesheet dosyası olmak üzere 3 dosyayla tanımlanabilir. Bu üç dosya tipi de XML destekli tüm sunucu ve veri tabanlarında çalıştırılabilir. Verinin yalın (native) olarak saklandığı veritabanı yönetim sistemleri, diğer veritabanlarında karşılaşılan harici veri tipleri veya farklı format dönüşümlerine gerek duymadığından; sorgu, transfer ve entegrasyon işlemlerinin performansları oldukça yüksektir. Ayrıca XML veritabanı yönetim sistemlerinde, verinin yalın biçimde kendi iç formatında tutuluyor olmasının bir avantajı da erişimi kolaylaştırmasıdır. Başka veritabanı yönetim sistemi yaklaşımlarında, buna benzer bir arama için karmaşık çoklu tablolar araştırılır ve daha sonra XML verisine dönüştürülür. Bu da işlem performansının düşmesine ve maliyet artırıcı ek uygulamalar eklenmesine neden olur [66, 76].

3.9 AJAX Prensipleri

AJAX yeni bir teknoloji olması itibarıyla henüz emekleme döneminde. Bununla birlikte birkaç Web tasarımcısı bu yeniliği başa çıkılması gereken bir sorun gibi algılamaktadır. Bu sorun iyi bir AJAX Web uygulaması yapabilmek için sıradan ya da kötü bir uygulamaya karşı ne yapılması gerektiği olarak tanımlanabilir. Michael Mahemoff (www.mahemoff.com), bir yazılım geliştirici ve kullanılabilirlik uzmanı, iyi Ajax uygulamaları için birkaç temel ilke belirlemiştir [37].

- **Minimal Trafik:** İnternet trafiği, internet üzerindeki veri akışıdır. Gönderilen ve alınan mesajların miktarı, yönü, web sitesine giren ziyaretçilerin sayısı, ziyaret ettikleri sayfalar; çevrimiçi trafiği oluşturmaktadır. AJAX uygulamalarında amaç bu internet trafiğini en alt seviyede tutmak için sunucudan mümkün olabildiği kadar az bilgi gönderip almaktır. Ayrıca bilgi akışında gereksiz bilgilerin her seferinde gönderilip alınmasının önüne geçilmesiyle uygulamaya sağlamlık katılır.

- **Kullanıcı etkileşimli model:** Genellikle AJAX uygulamaları geleneksel Web uygulamalarından farklı olarak değişik kullanıcı etkileşimli modeller sunar. Web standartlarındaki tıkla-bekle anlayışına karşı AJAX uygulamalarında anında sürükleyip bırak ya da çift tıklamayla yakınlaştırma gibi diğer kullanıcı arayüz paradigmaları vardır. Eğer kullanıcı etkileşimli modeller seçilirse, kullanıcı ne yapacağını bilir.
- **Sade arayüz:** Yanıp sönen sayfa bölümü ya da sürekli dönen animasyonlar gibi dikkat dağıtıcı ve gereksiz sayfa elementlerinden kaçınılır. Kullanıcılar göze hitabeden, uyumlu ve sade sayfaları tercih etmektedir.
- **Erişilebilirlik:** İnsanların bağlantı hızlarından, fiziksel engellerine, yaşadığı ülkenin toplumsal yapısından yaşlılığa kadar birçok yönden erişilebilirlik standartları incelenebilir. Birincil ve ikincil kullanıcıların beklentilerine en uygun olacak AJAX uygulamalarına erişim seçenekleri göz önünde bulundurulmalıdır. Özel yazılımlarla mı yoksa kullanılacak tarayıcı özellikleriyle mi adaptasyonun sağlanacağı belirlenmelidir. Böylece yeni hedef kitleler de tamamen dışarıda bırakılmış olmayacaktır.
- **Sayfanın tamamını indirilmesinden kaçınma:** Tetikleyici sayfa indirildikten sonra tüm sunucu iletişimi AJAX motoru tarafından yönetilmektedir. Böylece küçük bilgi alış-verişleri için tüm sayfanın tekrar yüklenmesinin önüne geçilmiş olur.
- **İlk kullanıcı:** AJAX uygulamalarında tasarım kullanıcıların daha önce karşılaşmadıkları bir biçimdedir. Etkileyici efektler ve sıkıcı olmayan bilgilendirici reklamlarla genel kullanım durumlarını kolayca başarmaktadır.

Bütün bu ilkelerde bulunan ortak nokta kullanılabilirliktir. AJAX, öncelikle, kullanıcılar için web deneyimini geliştirmeye çalışırken arkasındaki teknoloji sadece bunun için bir araçtır. Yukarıdaki ilkelere sadık kalarak, AJAX uygulamalar oldukça faydalı ve kullanışlı olacaktır [37].

3.10 AJAX Kütüphaneleri ve Araçları

Kütüphaneler hazır fonksiyonlara kolayca erişim sağlar. Ayrıca, sık kullanılan var olan kodları tekrar yazarak zaman kaybedilmemesini sağlar. Aşağıda en çok kullanılan AJAX kütüphanelerine örnekler verilmiştir.

3.10.1 Prototype

Prototype, dinamik web uygulamaları yaratmak için kullanabilecek gelişmiş bir kütüphanedir. Açık kaynaklı ve nesne tabanlı kod kütüphanesidir. Ajax için yazılan

kütüphanelerin ilk örneklerindendir ve diğer hazır kütüphanelerin bir kısmı prototype.js kullanmaktadır [78].

3.10.2 AJAX Gear Toolkit

AJAX olarak bilinen istemci taraflı tekniğin bütün avantajlarından faydalanmayı mümkün kılan bir araçtır. XMLHttpRequest nesnelerini kullanarak internet tarayıcısının asenkron çağrı yapmasını sağlar [78].

3.10.3 Asp.Net Ajax

ASP.NET AJAX, ASP.NET ile yapılan ve tüm İnternet tarayıcılarında çalışabilen uygulamalar için Microsoft tarafından geliştirilen bir kütüphanedir. Önce ‘Atlas’, daha sonra ‘ASP.NET Ajax’ adını almıştır. ASP.NET 2.0 ile kolay entegrasyon, ASP.NET 3.5 ile tam entegrasyon sağlar. Yeni nesil etkileşimli arayüzler oluşturmak için kullanılır [78].

3.10.4 Dojo Toolkit

Açık kaynak kodlu bir javascript toolkit olup dhtml sorunlarını büyük ölçüde çözmüştür. Arayüz programlama, karmaşık kodların basite indirgenmesi toolkitin sağladığı avantajlardır. Ayrıca dinamik web sitesi ya da başka ortamlar hazırlamayı kolaylaştıran ve sitenin daha kullanışlı olmasını sağlayan içerikler sunar [78].

3.10.5 MochiKit

Dreamweaver web uygulamaları için kullanılan ajax kütüphanelerinden birisidir. Eş zamanlı olmayan davranışlar için erteleme uygulamasını kullandığından, sunucuyla etkileşim halinde olan web sayfalarının geliştirilmesinde tercih edilir [78].

3.10.6 PHPLiveX

PHP ile yapılan uygulamalar için kullanılan bir kütüphanedir. Versiyon 2.5 ile gelen yenilikler ve optimizasyonlar, kütüphanenin potansiyelini ciddi biçimde arttırmıştır. Diğer php tabanlı kütüphaneler gibi php fonksiyonlarını javascript koduyla çağırır; fakat diğerlerine göre daha az kodla kullanımı sağlar [78].

3.10.7 Sardalya (Sarmal Dinamik Altyapısı)

Sardalya Dinamik AJAX Kütüphanesi, DOM desteği olan tüm güncel tarayıcılarda çalışmak için tasarlanmış bir 'AJAX Kütüphanesi'dir. Linux, windows, freebsd, osx, sun gibi platformlarda çalışabilir. Tüm tarayıcılardan bağımsız olarak çalışır ve kullanımı sırasında az yer kaplar [78].

3.10.8 Sajax

Sajax'ın açılımı Simple Ajax Toolkit'tir. ASP ve PHP tabanlı programlarda AJAX uygulamaları geliştirmek için kullanılan bir kütüphanedir. Bütün sunucu taraflı platformlara çağrı gönderebilme yeteneğine sahiptir [78].

3.10.9 Solvent

AJAX uygulamalarını modüler bir yapıda gerçekleştiren farklı bir kütüphanedir. JavaScript ile yazılmıştır ve tüm tarayıcılardan bağımsız olarak çalışan bir AJAX uygulamasıdır. Web uygulamalarını kuvvetlendirmek ve uygulamaların geliştirme sürecini hızlandırmak için kullanılır [78].

3.10.10 AjaxAC

AjaxAC, açık kaynak kodlu php tabanlı bir kütüphanedir. AjaxAC ile varolan PHP sınıflarına ve MySQL veritabanına, ek istek verilerini döndürmek için geçiş oldukça kolaydır [78].

3.10.11 Xajax

Xajax Php, JavaScript, HTML, XMLHttpRequest ve XML kullanarak tarayıcı ve sunucu arasındaki alışverişi ve verileri göstermeyi gerçekleştirir. Xajax ile geliştirilmiş uygulamalar, sunucudaki php fonksiyonlarını çağırabilir ve sayfayı yeniden yüklemeyi içerdiğini güncelleştirebilir [78].

3.10.12 Scriptaculous

Prototype tabanlı bir js framework'tür. Yapılan AJAX işlemleri prototype içerisinde bulunan nesneler ve fonksiyonlar ile yapılır [78].

3.11 AJAX Uygulamalarında Kullanılan Diğer Teknolojiler

AJAX istemci ile sunucu arasında eş zamanlı olmayan iletişim sağlayan bir teknolojiler bütünüdür.

3.11.1 HTML/XHTML

İstemci tarafı (client side) bir metin işaretleme dili olan XHTML (Extensible HyperText Markup Language- Geliştirilebilir Metin İşaretleme Dili) dili XML söz dizimi ile oluşturulmuştur. XHTML standart etiket ayarlarını tanımlarken spesifik bir yol kullanılmalıdır. World Wide Web Consortium (W3C) dünyadaki Web geliştiriciler için oluşturulmuş aynı standartlaştırılmış yaklaşımı benimserler [66].

3.11.2 CSS

XHTML'de stil biçimlendirmesini sağlar. Yüzlerce sayfanın tek merkezden yönetilmesini sağlar. Geleneksel yöntemlerde her sayfanın biçimlendirme ayarları birçok kod satırı arasından stillerle ilgili olanların seçilip düzeltilmesiyle yapılır. CSS kullanılan yöntemlerde aşağıdaki gibi bir kod parçası tüm sayfalardaki stil ayarlarının yapılmasını tek merkeze bağlamıştır.

```
#kabuk{  
width:760px;  
background-color:#EE7700;  
}  
.spot{  
color:#EE7700;  
font-family:Verdana, Comic Sans MS, Helvetica, sans-serif;  
}
```


AJAX ve CSS birlikte kullanılarak, masaüstü çalıştırılabilir EXE yazılımlar ile aynı hızda ve performansta çalışan web uygulamaları geliştirmek mümkün hale gelmiştir. Google belgelerinde Gmail, bunun en büyük örneğidir.

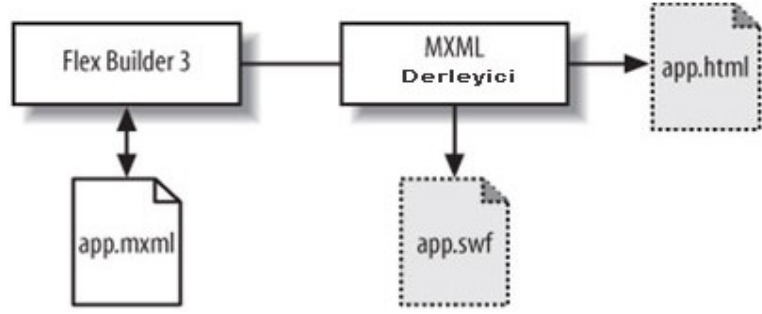
Web sayfalarındaki tüm içerik, biçimlendirmeler, resimler vb istemciye HTML/XHTML kodları halinde gönderilir. Bu kodlar sade ve basittir. Bu da oldukça fazla kullanıcıya hitap edebilen web siteleri için önemlidir. Çünkü internete bağlandığında kısıtlı bir bant genişliğine sahip kullanıcılar daha az sayıda karakter içeren sayfaları daha kısa sürede indirebilecektir. Böylece kullanıcılar, toplam sürede, daha fazla sayfa ziyaret edecektir. Sayfaların hızlı yüklenmesi, kullanıcıların o siteler hakkındaki memnuniyetini artıracaktır. Ayrıca bu durum yazıcı dostu sayfaların oluşturulmasında etkilidir [79].

3.11.3 XSLT

XSLT (Extensible Stylesheet Language Transformations, Genişletilebilir Biçimlendirme Dili Dönüşümleri), XML tabanlı dökümanların orijinal halini değiştirmeden yeni bir döküman oluşturmak için kullanılan bir dildir [80]. Örneğin bir XML dökümanını XHTML'ye dönüştürmek için XSL dönüşümü (transformation) kullanılabilir. XML'nin içindeki "hangi" data "nasıl" dönüştürülecek sorularından hangi sorusunun cevabı XPath ile belirtilirken, nasıl sorusunun cevabı ise yine XML ile belirtilir [81].

4 FLEX

Flex MXML, ActionScript 3.0 ve Flex sınıf kütüphanelerinin birleşiminden oluşan sunucu-istemci tabanlı yeni bir teknolojidir. Ayrıca Flex Flash uygulamaları ile derlenebilen XML tabanlı bir dil olarak tanımlanabilir [84]. Flex'in çalışma sistematığının işlem akış şeması Şekil 4.1'de gösterilmektedir.

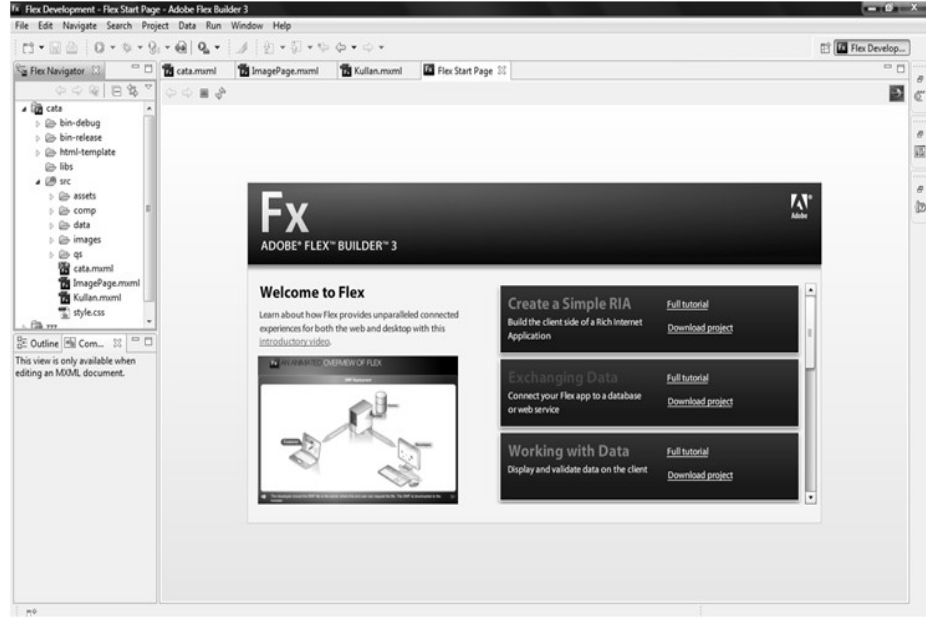


Şekil 4.1 Flex uygulamalarının işlem akış şeması

Şekil 4.1'de sol alt köşeden sağ üst köşeye doğru işlem akış şeması takip edildiğinde Flex Builder 3 ile geliştirilen app.mxml uygulaması MXML derleyicide çalıştırıldığında bir swf dosyası üretmektedir. Ayrıca üretilen swf dosyasının sunucu üzerinde test edilebilmesi için birde html uzantılı dosya oluşturulur. Flex'in en önemli özelliği üretilen son ürünün swf formatında olmasıdır. Ayrıca olay tutucu komut dizelerinin kolay kullanımı yazılımın oluşturulmasında Flex programını tercih nedeni olmasını sağlamıştır.

Adobe® Flex® 3, yaygın olarak kullanılan tarayıcılar, masaüstleri ve işletim sistemlerinde kolaylıkla etkileyici web uygulamaları oluşturmaya ve bakımlarını yapmaya yönelik, verimliliği yüksek bir programdır. Açık kaynak (open source) altyapısına sahiptir. Eclipse tabanlı bir plugin kümesidir.

Adobe firması tarafından Flex Builder 3 Pro lisansı herhangi bir eğitim kurumuna öğrenci, eğitmen ya da çalışan olarak bağlı olduklarını belgeleyenlere tamamen ücretsiz olarak verilmektedir. Kurulum tamamlandıktan sonra ilk açılış ekranı Şekil 4.2'de gösterildiği gibidir [83].



Şekil 4.2 Flex arayüzü

4.1 Adobe® Flex® 3 Kurulumu İçin Gereken Minimum Sistem Gereksinimleri

Uygulama gerçekleştirmek için farklı işletim sistemleri için gerekli sistem gereksinimleri aşağıda belirtilmiştir.

4.1.1 Windows İçin Flex Builder 3 (Standart ve Profesyonel)

Flex'in Windows kurulumu için gerekli minimum sistem gereksinimleri aşağıda gösterilmektedir [83]:

- Intel® Pentium® 4 işlemci
- Microsoft® Windows® XP Service Pack 2 veya Windows Vista® Home (Premium veya Basic), Business veya Ultimate
- 1 GB RAM (2 GB önerilir)
- 500 MB kullanılabilir sabit disk alanı (eklenti yapılandırması için ek 500 MB gerekir)
- Java™ Virtual Machine: Sun™ JRE 1.4.2, Sun JRE 1.5 (dâhildir), IBM® JRE 1.5 veya Sun JRE 1.6
- Eklenti yapılandırması için Eclipse™ 3.2.2, 3.3 ve 3.4 (Eclipse 3.3, Windows Vista için önerilir)
- Adobe Flash® Player 9 yazılımı
- BEA Workshop 10.1

- IBM Rational Software Architect 7.0.0.3 (Yalnızca Eclipse 3.3 eklenti yapılandırması)

4.1.2 Macintosh İçin Flex Builder 3 (Standart ve Profesyonel)

Macintosh bilgisayarlar için gerekli minimum sistem gereksinimleridir [83].

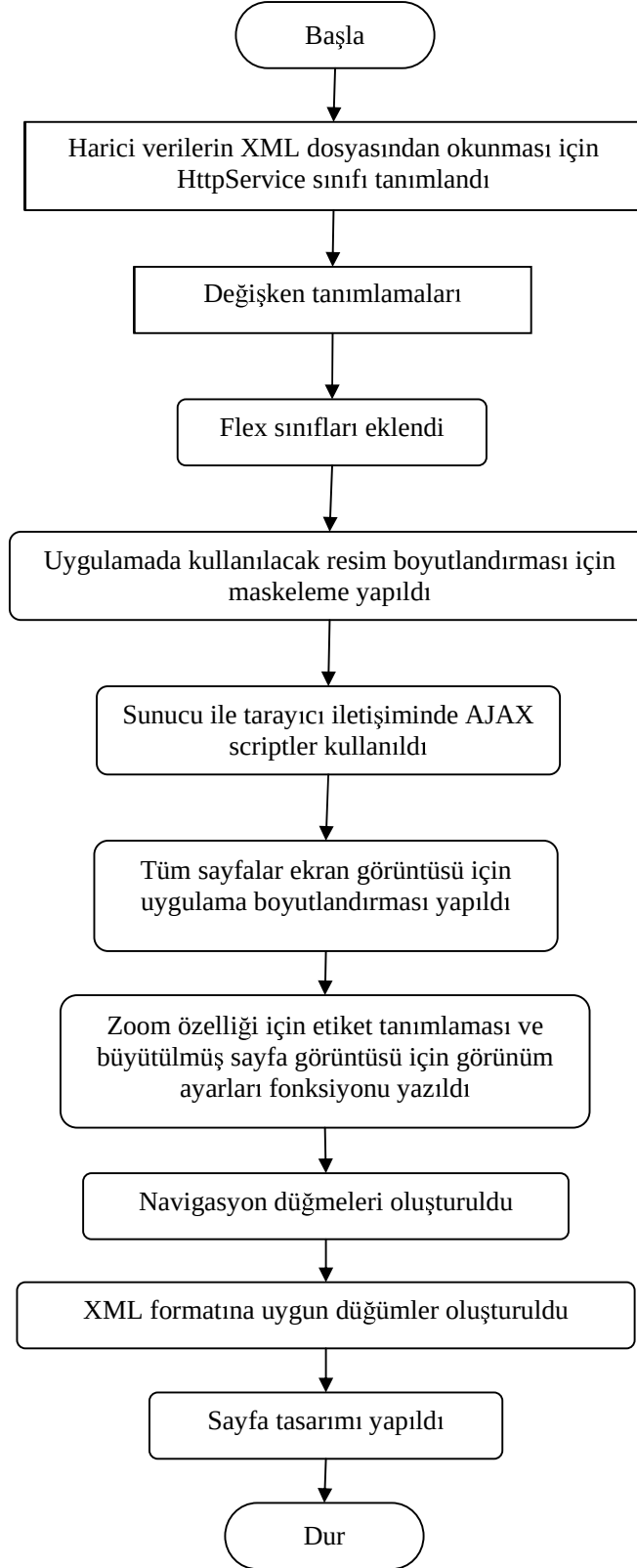
- PowerPC® G4 1,25 GHz veya Intel işlemci
- Mac OS X v10.4.7–10.4.10 veya 10.5
- 1 GB RAM (2 GB önerilir)
- 500 MB kullanılabilir sabit disk alanı
- Java Virtual Machine: Apple'dan JRE 1.5 veya JRE 1.6
- Eclipse 3.2.2, 3.3 ve 3.4 (eklenti yapılandırması için)
- Adobe Flash Player 9 yazılımı

5 ÇEVİRİMİÇİ KATALOG TASARIMI: Fırat Üniversitesi Örneği

AJAX teknolojisi, Adobe/Macromedia firmasının geliştirdiği Flex programı ve bileşenleri kullanılarak Fırat Üniversitesinin 2009 çevrimiçi katalog çalışması herhangi bir teknik bilgi gerektirmeden güncellenebilen alt yapısıyla gerçekleştirilmiştir. Kullanıcı arayüzü oluşturulurken XML tabanlı bir gramer yapısı ve Flex programı kullanılmıştır. Bu tez çalışmasında sunucu olarak WampServer 2.0 kullanılmıştır. Katalog uygulamasının kaynak kodlarına ek bölümünden bakılabilir.

AJAX teknolojisiyle ortaya çıkmaya başlayan çevrimiçi broşürler ve kataloglar internetteki Web uygulamaları üzerinde çevrimiçi hedeflerin önemli bir bölümüne hâkim olmaktadır [37]. Bu tez çalışması da buna örnek teşkil etmektedir.

Çalışmanın oluşum haritasına literatür taramasıyla başlanmıştır. Bir sonraki adımda hazırlanacak olan programın özellikleri belirlendi. Bu aşamadan sonra yazılım için kullanılacak programın seçimi yapıldı. Yazılım algoritması planlanıp Şekil 5.1'deki kodlama algoritması oluşturulmuştur. Yazılım test edildiğinde oluşan hatalar giderilmiştir. Çalışmanın sayfa tasarımı yapılarak son şekli verilmiştir.



Şekil 5.1 Kodlama algoritması

Aşağıdaki kod satırlarıyla katalog çalışmasının uygulama dosyası olan cata.mxml dosyasına Flex içinde çalışma yöntemlerini belirleyen faydalı sınıflar dâhil edilmiştir.

```
<![CDATA[  
    import mx.utils.StringUtil;  
    import mx.rpc.soap.LoadEvent;  
    import mx.controls.Alert;  
    import mx.rpc.events.FaultEvent;  
    import qs.controls.FlexBook;  
    import qs.controls.flexBookClasses.FlexBookPage;  
    import qs.controls.flexBookClasses.FlexBookEvent;  
    import qs.controls.flexBookClasses.WBookPageImpl;  
    import qs.caching.ContentCache;  
    import mx.rpc.events.ResultEvent;  
    import mx.rpc.http.HTTPService;  
    import mx.core.UIComponent;  
    import mx.managers.PopUpManager;  
    import comp.ResimSec;  
    import Kullan;  
    import mx.core.ScrollPolicy;  
    import mx.collections.ArrayCollection;
```

XML tanımlamaları, diziler, sunucu bağlantıları ve zoom işlemleri için gerekli değişken tanımlamaları aşağıdaki kod satırlarıyla yapılmıştır.

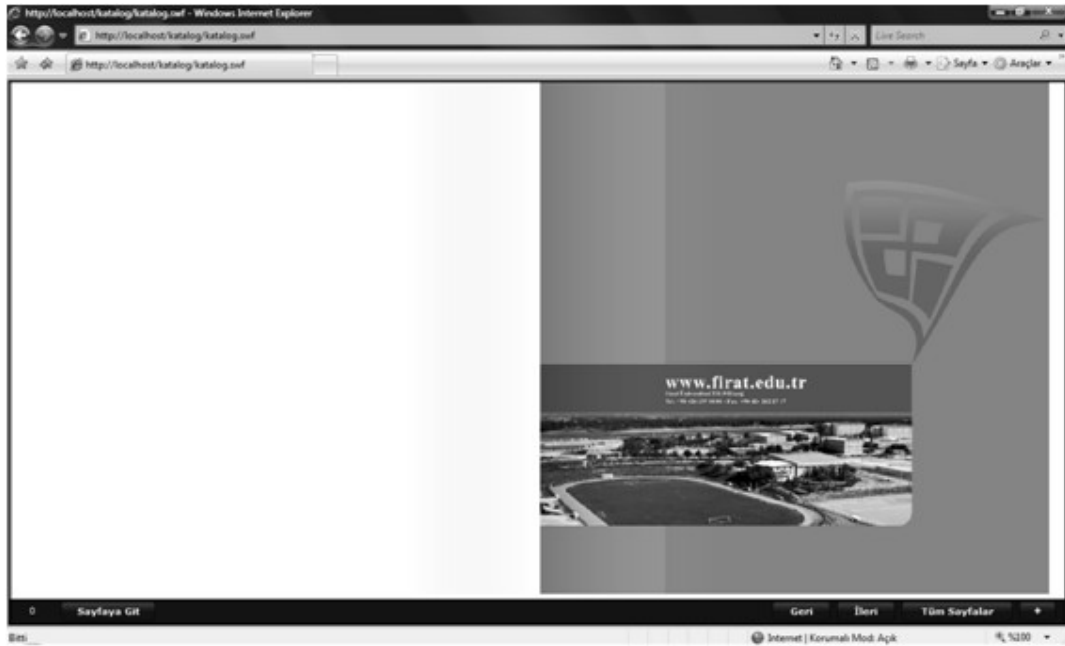
```
private var a:XML;  
    [Bindable]  
private var resimler:ArrayCollection;  
private var dataSet:HTTPService;  
private var resimdizi:Array=new Array();  
    private var zoomYap:Boolean=false;
```

Katalogun local sunucu üzerinde çalıştırılabilmesi için kaynak dosyası Wamp kurulumu sonrasında oluşan www klasörü içine yerleştirilir. E-katalog resimleri dinamik olarak data dizindeki images.xml dosyasında çekilmektedir. Katalog resimlerini değiştirmek için bu dosyadaki xml düğümlerinin değiştirilmesi yeterlidir. Değişiklik yapılacak düğüm örneği aşağıda gösterilmektedir.

```
<image source="images/2.jpg" thumb="images/2.jpg" zoom="images/2.jpg"/>
```

Yukardaki satırdaki düğümleri açıklamak gerekirse, source düğümünde katalogda görünecek resim, thumb düğümünde eğer asıl resim (source) resmi yüklenene kadar başka bir resim görünmesi isteniyorsa o resmin kaynağı gösterilir. Zoom düğümü ise sayfaya zoom yapılması esnasında görünecek resmin kaynağını gösterir.

Yukardaki 3 düğümden thumb düğümü isteğe bağlı olarak kullanılır. Zoom düğümündeki resim asıl resmin %100 büyük hali olmalıdır.



Şekil 5.2 Kataloğun ilk sayfasının ekran görüntüsü

E-katalog sayfaların boyutunu katalog.html'de tanımlı olan object nesnesinin genişlik ve yükseklik özelliklerinden almaktadır. Şekil 5.2'nin kaynak kodu aşağıda verilmiştir.

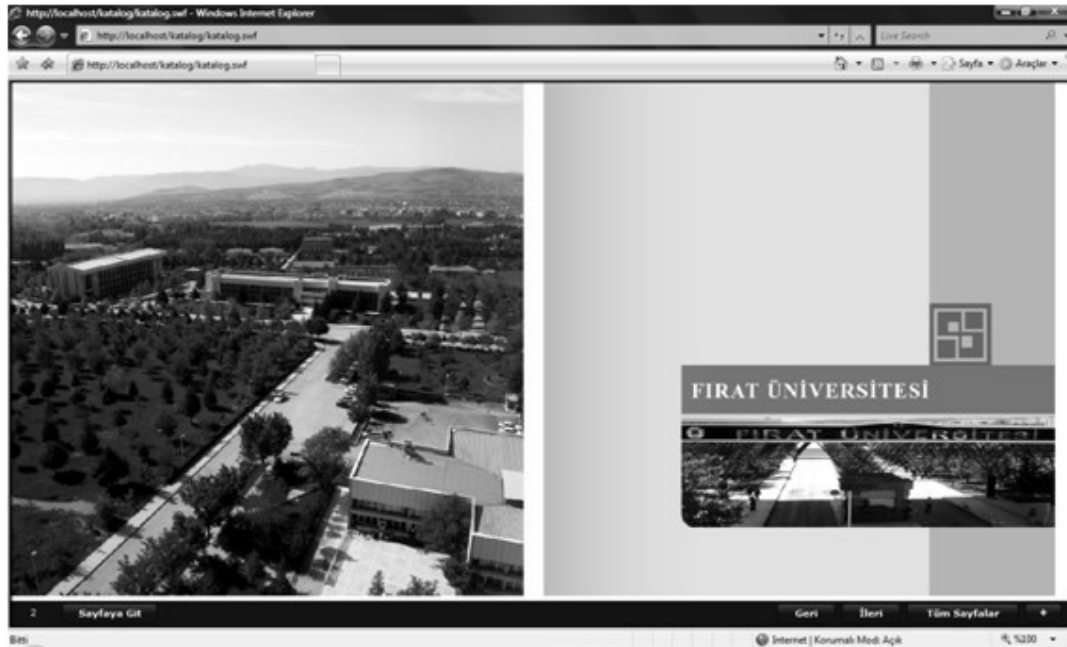
```
<script src="Scripts/AC_RunActiveContent.js" type="text/javascript"></script>
<script type="text/javascript">
AC_FL_RunContent( 'id','katalog','width','894','height','653','codebase','http://fpdownload.m
acromedia.com/get/flashplayer/current/swflash.cab','movie','katalog','quality','high','bgcolor',
'#000000','allowsriptaccess','sameDomain' ); //end AC code
</script><noscript><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
id="katalog" width="894" height="653"
```



```

codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">
  <param name="movie" value="katalog.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#000000" />
  <param name="allowScriptAccess" value="sameDomain" />
  <embed src="katalog.swf" quality="high" bgcolor="#000000"
    width="984" height="653" name="katalog" align="middle"
    play="true"
    loop="false"
    quality="high"
    allowScriptAccess="sameDomain"
    type="application/x-shockwave-flash"
    pluginspage="http://www.adobe.com/go/getflashplayer">
  </embed>
</object></noscript>

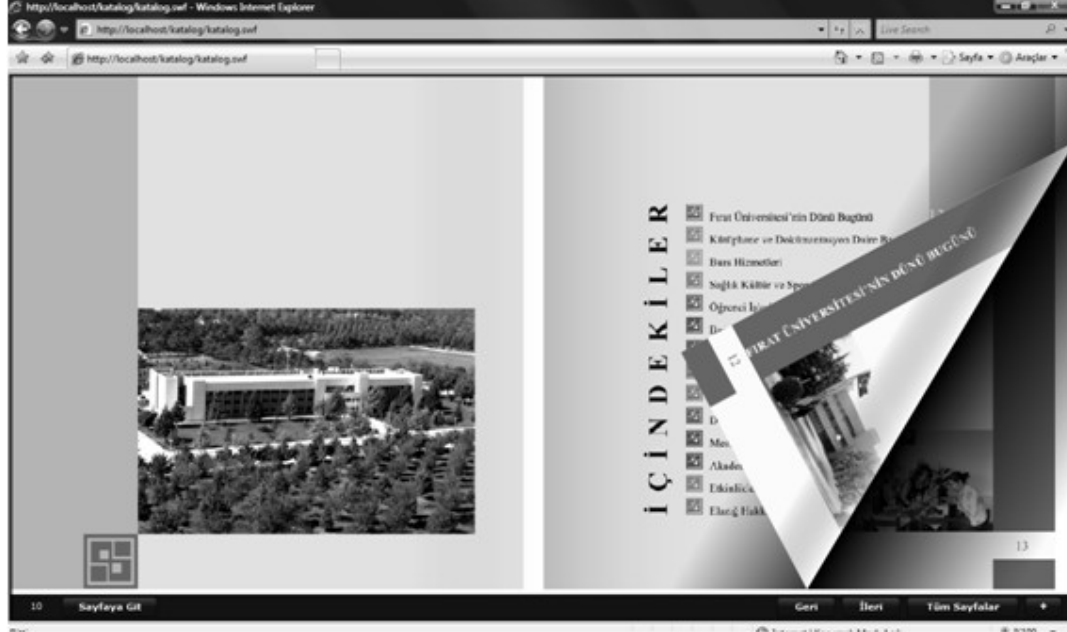
```



Şekil 5.3 Kataloğun 2/3 sayfa görüntüsü

Örneğin Şekil 5.3'te kullanılan resimlerin width="447" height="613" boyutunda ise resimlerin sayfada görünmesi için html etiketindeki width özelliğini 447x2=984 ve height

özelliğini $613+40=653$ şeklinde düzenlenmesi gereklidir. Yani $\text{width}=\text{resim genişliği} \times 2$ ve $\text{height}=\text{resim yüksekliği} + 40$ olmalıdır.



Şekil 5.4 Kataloğun sayfa çevirme efektli görüntüsü

Şekil 5.4'te görülen e-katalogun sayfayı çevirme özelliği ile hızı gibi işlemleri style.css dosyasında tutulmaktadır. Gerekli olmadıkça bu dosya üzerinde değişiklik yapılmaması tavsiye edilir. Oluşturulan style.css dosyasının kodları aşağıda gösterilmiştir.

```
FlexBook {
    color: #000000;
    textRollOverColor: #000000;
    border-thickness: 0;
    border-style: none;
    page-slope: 0.6;
    active-grab-area: page;
    page-shadow-strength: 1;
    curve-shadow-strength: 1;
    auto-turn-duration: 1500;
}
Application {
    color: #F1F1CC;
    textRollOverColor: #FFFFFF;
```

```

        backgroundGradientColors: #FFFFFF, #FFFFFF;
        backgroundColor: #FFFFFF;
    }
    SuperImage {
        border-thickness: 0;
        border-style: none;
    }
}

```

Şekil 5.5’de görülen navigasyon çubuğunda 1 numara ile gösterilen alan istenilen sayfaya konumlandırma yapılabilmesi için sayfa numarasının girildiği kutudur. 2 numara ile gösterilen alan “Sayfaya Git” düğmesine tıklandığında istenilen sayfanın açılmasını sağlayan düğmedir. 3 ve 4 numara ile gösterilen düğmeler “Geri”, “İleri” düğmeleridir. Fare kontrolü olmadan da sayfalar arasında gezinti yapılabilmesini sağlar. 5 numara ile gösterilen alan “Tüm Sayfalar” düğmesidir.



Şekil 5.5 Kataloğun navigasyon çubuğu

Tüm sayfaların açılır pencere içerisinde görüntülenmesini sağlayan fonksiyon aşağıdaki gibidir. Kullan.mxml sayfasında çekilen resimler pencerede göster komutuyla genişlik ve yüksekliği ayarlanır. Seçilen nesneler custom.Move.end() komutuyla ekrana taşınır.

```

private function tumsayfalar():void {
    Kullan.show_window(this,false,application.width-10,(application.height-60)/2);
    customMove.end();
    customMove.play();
}

```

Şekil 5.6’da görüldüğü üzere çevrimiçi katalogdaki yüklü olan tüm sayfaları yukarıdan açılan bir pencere yardımıyla gösterilir. Tüm sayfaları göster düğmesi tıklandığında görünecek resimler ile ilgi ayarlar ise data klasöründeki onizle.xml dosyasından ayarlanabilir. Xml’nin esnek etiketleme yapısı sayesinde onizle.xml dosyasında oluşturulan <sayfalar> etiketi arasına aşağıdaki kod satırları çoğaltılarak oluşturulmuştur.

<sayfa>

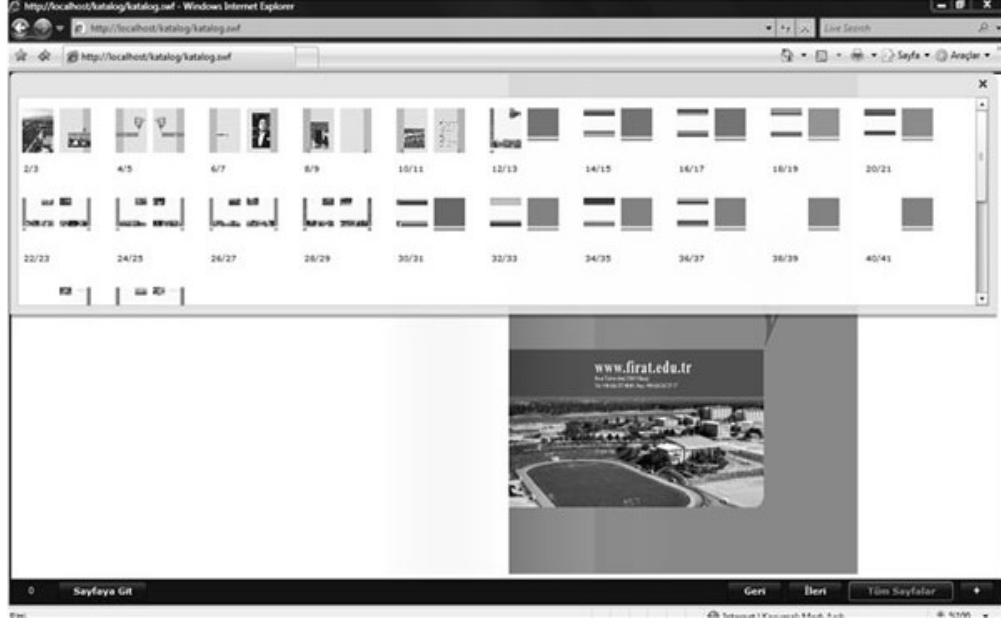
<resim1>images/2.jpg</resim1>

<resim2>images/3.jpg</resim2>

<git>2</git>

<pages>2/3</pages>

</sayfa>



Şekil 5.6 Tüm Sayfalar düğmesi tıklandığındaki ekran görüntüsü

Son olarak 6 numaralı düğme tıklandığında “+” ve “-” konumlarını alarak büyüteç özelliği kullanılır. Şekil 5.7’de gösterildiği gibi imleç sayfa üzerinde gezdirilirken bu alan üzerinde zoom yapılarak yan sayfada büyütülmüş görüntü sağlanır.



Şekil 5.7 Zoom özelliği

Zoom özelliğinde resimler üzerindeki genişlik ve yükseklik değerleri için if else blok yapısı kullanılarak maskeleme fonksiyonu yazılmıştır. Maskeleme fonksiyonu aşağıdaki kod satırlarında gösterilmektedir. Fare'nin x-y koordinatlarında alacağı değerlere göre fare'nin büyütülecek alan üzerinde gezinmesiyle yan sayfada büyütülmüş görüntü elde edilmiştir.

```
private function maskele(e:MouseEvent):void{
    if(zoomYap){

        if((mouseX>(application.width/2)) && (mouseY<application.height-40) ){
            maske.x=0
            img2.visible=false;
            img1.visible=true;
            img1.x=(application.width/2)-mouseX;
            img1.y=-1*mouseY
        }

        else if((mouseX<(application.width/2)) && (mouseY<application.height-40) ){
            maske.x=application.width/2;
            img1.visible=false;
            img2.visible=true;

            img2.x=-1*mouseX;
            img2.y=-1*mouseY
        }
    }
}
```

Sayfa görüntülerinin maskelenmesinden sonra resimlerin kaynak dosyadan çekilerek ekrana büyütülmüş hallerinin yansıtılması için aşağıdaki fonksiyon oluşturulmuştur.

```

private function zoom():void {
    if(zoomer.label==""){
        zoomer.label="-"
    }
    if(img2.source!=a.image[book.currentPageIndex*2].@zoom){
        yukleme.visible=true;
    }

    img2.addEventListener(Event.COMPLETE,buyuk2_yuklendi);
    img2.source=a.image[book.currentPageIndex*2].@zoom;

    if(a.image.length()>book.currentPageIndex*2+1){
        img1.addEventListener(Event.COMPLETE,buyuk1_yuklendi);
        img1.source=a.image[book.currentPageIndex*2+1].@zoom;
    }

    zoomYap=true;
}
else if(zoomer.label=="-"){
    zoomer.label="";
    setGorunum();
}

}

private function setGorunum():void {
    zoomYap=false;
    zoomer.label="";
    img1.visible=false;
    img2.visible=false;

}

```

6 SONUÇ

Web 2.0 akımı ve zengin internet uygulamaları günümüz popüler konularındandır. Bu nedenle tez çalışması bu amaca yönelik olarak son teknolojiler kullanılarak geliştirilmiştir.

AJAX teknolojisi kullanılarak kullanıcının istekleriyle sayfanın tamamen yenilenmeden ilgili işlemin yürütülebilmesi sağlanmaktadır. Geliştirilen sistem çapraz tarayıcı desteğine sahip, esnek ve kolay kullanılabilir bir yapıya sahip olduğundan herhangi bir programlama bilgisine sahip olmadan kişilerin istenilen değişiklikleri yapılabileceği bir alt yapıya sahiptir. Bu çalışma ışığında en yenilikçi internet yazılımları konusunda uygulama yapmak isteyen yazılım uzmanlarına kaynak teşkil edecek bir çalışma gerçekleştirilmiştir. Yazılım geliştiricileri ve yazılım mimarları AJAX'ı Flash, Flex, Appletler ve diğer teknolojileri uygulamalarına kolayca entegre edebilirler.

Bu tez çalışması kapsamında “Web’de Ajax Tekniğinin Bant Genişlik Optimizasyonuna Etkisi” başlıklı bir makale hazırlanmıştır [85]. Bu makalede Web Performance Suite 3.5.5941 programıyla AJAX'ın performans testi yapılmıştır. Bu ölçümler sonucunda elde edilen değerler göstermiştir ki, bir şebeke üzerinden gönderilecek sinyalin en düşük ve en yüksek frekansları arasında gereksiz yere kullanılan aralığın optime edilmesinde, AJAX etkili bir yöntemdir. AJAX'ın çalışma prensibi sayesinde uygulamalar en iyi performansla gerçekleştirilir.

AJAX teknolojisinin hala gelişimi devam etmekte olduğundan bazı eksik yanları da bulunmaktadır. Örneğin tarayıcı üzerindeki “Geri” fonksiyonu bazen beklediği gibi çalışmayabilir. Dinamik olarak oluşturulan sayfalar tarayıcının önceden işlediği sayfaları yöneten mekanizmalarına kendini otomatik olarak kaydetmez, bu yüzden tarayıcının “Geri” tuşuna basıldığında, istenen sayfa elde edilemeyebilir [86]. Başka bir dezavantaj ise AJAX ile oluşturulan içeriğin kodları kaynak kodlar kısmında görülememektedir. Aynı zamanda Google ve Yahoo gibi arama motorlarının Web sayfalarına gönderdiği robot yazılımlar da bu kodları görememektedir. Dolayısıyla tamamen AJAX ile oluşturulan içerik, arama motorlarının sonuç sayfalarında listelenmezler. Günümüzde, yapılan Web sayfasının arama motorlarında görüntülenmesi oldukça önemlidir. Fakat bir Web uygulaması, ziyaretçiler hakkında önemli kişisel bilgilere sahipse, bu bilgilerin arama motorlarında herkes tarafından görüntülenmesini sağlaması bir avantaja dönüşür [87]. AJAX yerinde ve doğru bir şekilde kullanıldığında uygulamalarda performans artışını sağlar.

Bu tez çalışmasında Web 2.0 akımının ortaya çıkış nedenleri ve RIA konseptine duyulan ihtiyacın temeli irdelenmiştir. Web 2.0 akımında öne çıkan uygulamalar ve çalışma sistematiği incelenmiştir. Öne çıkan bu uygulamaların site tasarımında dikkat çeken

özelliklerine değinilmiştir. AJAX teknolojisini oluşturan bileşenler incelenmiştir. Bu inceleme ve araştırma sonucunda zengin internet uygulamalarına örnek teşkil edecek Web 2.0 akımının çekirdeği olan AJAX teknolojisi kullanılarak Fırat Üniversitesi için çevrimiçi bir katalog tasarlanmıştır. Bu tasarımda flexBook, flexbookpage, landscape, wrapper, zoomer gibi ek bileşenler kullanılmıştır. Navigasyon paneli tasarlanmıştır. Oluşturulan tasarımda yerli ve yabancı literatür taraması yapılarak emsallerindeki kullanışlı özellikler projeye dahil edilmiştir. Projenin geliştirilmesi sırasında izlenen adımlar resimlenerek ortaya bir yol haritası çıkarılmıştır.

6.1 Öneriler

Bu tez çalışması esas alınarak, ileride bu uygulamalar mobil cihazlar aracılığıyla kullanılabilirliği üzerine araştırmalar sürdürülebilir. AJAX'ın üstün özellikleri sayesinde Web üzerinden yayıncılık ve mobil iletişim alanları farklı bir boyut kazanmıştır. Birçok şirket artık taşınabilir cihazlar için Web 2.0 uygulamalarını AJAX'la optimize etmeye başlamıştır. Yeni bir teknoloji olan AJAX'la taşınabilir cihazlardaki performans, bellek yönetimi, sınırlı tuş girişi gibi sorunların üstesinden gelinmesi hedeflenmelidir.

AJAX kullanılarak IDE (Integrated Development Environment) entegrasyonu Eclipse için bağlantılar yazılabilir veya yapım servisi aracılığıyla doğrudan IDE'den yapım paketleri geliştirilebilir. Günümüzde oldukça popüler olan açık kaynak kodlu yazılım geliştirme ortamlarında entegre uygulamalar geliştirilebilir.

Kullanıcı Arabirimi İşaretleme Betiği (User Interface Markup Language) sınıfına dâhil edilecek geleceğin en popüler web tabanlı teknolojilerinden biridir. Flex geliştiricileri Adobe Flex Builder - IDE ortamında LiveCycle Data Services, Java, Coldfusion, ASP.NET, JSP, PHP gibi popüler web programlama dilleri yardımıyla etkileşimli ve güvenli web tabanlı projeler geliştirilebilir [82].

KAYNAKLAR

1. İnternet: Web 2.0, http://tr.wikipedia.org/wiki/Web_2.0, Erişim tarihi: 03.09.2008
2. Özer Z., “Web Teknolojileri ve AJAX”, Seminer Çalışması, Fen Bilimleri Enstitüsü, sf:2, 2007
3. İnternet: Web 2.0 Nedir, www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, Erişim tarihi: 04.10.2008
4. İnternet: <http://ravenyoung.spaces.live.com/blog/cns!17376F4C11A91E0E!3717.entry>, Erişim tarihi: 03.09.2008
5. McLuhan M., Povers, B. R., Çeviren: Bahar Öcal Düzgören, “Global Köy, 21. Yüzyılda Yeryüzü Yaşamında ve Medyada Meydana Gelecek Dönüşümler”, Scala Yayıncılık, 2001
6. İnternet: RIA konsepti, <http://www.ademaktepe.com/GunlukDetay.aspx?Id=122>, Erişim Tarihi: 26.06.2008
7. Onat F., Aşman Ö., Kılıç A., “Sosyal Ağ Sitelerinin Reklam Ve Halkla İlişkiler Ortamları Olarak Değerlendirilmesi”, Sayfalar: 3-6, 2008
8. İnternet: <http://i8.tinypic.com/6kymb0k.gif>, Erişim tarihi: 11.07.2008
9. İnternet: DFA, <http://www.doubleclick.com/products/dfa/index.aspx>, Erişim tarihi: 14.02.2009
10. İnternet: Adsense.com, <https://www.google.com/adsense/support/bin/answer.py?answer=9712&topic=13488>, Erişim tarihi: 15.11.2008
11. İnternet: DoubleClick.com, www.doubleclick.com, Erişim: 13.10.2008
12. İnternet: Adsense hoş geldiniz, www.google.com/adsense, Erişim: 18.11.2008
13. İnternet: Flickr Tanımı, <http://tr.wikipedia.org/wiki/Flickr>, Erişim tarihi: 09.03.2009
14. İnternet: Flickr Fotoğraf Paylaşım Sitesi, <http://www.flickr.com/>, Erişim: 18.02.2009
15. İnternet: BitTorrent, <http://tr.wikipedia.org/wiki/BitTorrent>, Erişim tarihi: 05.01.2009
16. İnternet: Bittorrent kullanım rehberi, <http://www.bittorrent.com/btusers/guides/beginners-guide>, Erişim tarihi: 05.01.2009
17. İnternet: Bittorrent kullanım kılavuzu, http://www.shukko.com/bittorrent_kilavuzu/, Erişim tarihi: 05.01.2009
18. İnternet: Bittorrent çalışma mantığı, <http://www.forumel.biz/bittorrent-nedirgne-ise-yarargresimli-detayli-super-anlatim-t12836.html?s=17f7b3147d97799d31c32b09e87d43e4&am>, Erişim tarihi: 05.01.2009
19. İnternet: Sözlük,

- http://www.parliament.vic.gov.au/SARC/E-Democracy/Final_Report/Glossary.htm, Eriřim tarihi: 05.01.2009
20. FU F., LIU L., WANG L., “Empirical Analysis of Online Social Networks in the Age of Web 2.0”, 2007, Physica A: Statistical Mechanics and its Applications, Volume 387, Issues 2-3, Sayfalar: 675-684.
21. İnternet: Facebook, <http://tr.wikipedia.org/wiki/Facebook>, Eriřim tarihi: 06.01.2009
22. İnternet: Facebook makale, <http://turk.internet.com/haber/yazigoster.php3?yaziid=22787>, Eriřim tarihi: 06.01.2009
23. İnternet: Facebook istatistiksel bilgiler, <http://www.insidefacebook.com/2009/01/07/facebook-growth-continuing-surpasses-150-million-monthly-active-users/>, Eriřim tarihi: 06.01.2009
24. İnternet: Peter Thiel Facebook haberleri, http://www.businessweek.com/technology/content/mar2009/tc2009031_743025.htm, Eriřim tarihi: 06.01.2009
25. İnternet: Blog, <http://tr.wikipedia.org/wiki/Blog>, Eriřim tarihi: 13.02.2009
26. İnternet: Technorati, <http://technoratimedia.com/>, Eriřim tarihi: 13.02.2009
27. İnternet: Blog arařtırması, <http://www.mmistanbul.com/makale/title/bloglarin-satin-alma-kararlarina-etkisi-uzerine-bir-arastirma>, Eriřim tarihi: 13.02.2009
28. İnternet: Wiki nedir, <http://yunus.hacettepe.edu.tr/~b0343579/baglantiler/wikinedir.htm>, Eriřim tarihi: 13.02.2009
29. İnternet: İnternette wikinin yeri, <http://ogrenci.hacettepe.edu.tr/~besst/index.php/2008/06/03/internette-wikinin-yeri/>, Eriřim tarihi: 14.02.2009
30. Bergstedt S., Wiegrefte S., Wittmann J., Moller D., “Content Management Systems and e-Learning-Systems – A Symbiosis?”, Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT’03)
31. İnternet: İerik Ynetim Sistemleri, http://tr.wikipedia.org/wiki/%C4%B0%C3%A7erik_y%C3%B6netim_sistemi, Eriřim tarihi: 17.02.2009
32. İnternet: İerik Ynetim Sistemleri, <http://www.msxlab.org/forum/internet-bilgisayar-dunyasi/22262-icerik-yonetim-sistemi-cms-content-management-system.html>, Eriřim tarihi: 17.02.2009
33. İnternet: Web 2.0 Tasarım Araları, <http://www.eburhan.com/hazir-web-20-tasarim-araclari/>, Eriřim tarihi: 06.02.2009

34. İnternet: Tasarım rehberi, <http://www.hasanyalcin.com/web-20-tasarim-rehberi-yansima-ikon-ve-flaslar/?ref=Guzels.TV>, Erişim tarihi: 06.02.2009
35. İnternet: AJAX kaynak makalesi, <http://adaptivepath.com/publications/essays/archives/000385.php>, Erişim tarihi: 06.02.2009
36. Keith J., “Bulletproof Ajax”, New Riders yayıncılık, Sayfa: 3, 2007
37. Zakas N. C., McPeak J., Fawcett J., “Professional Ajax, 2nd Edition”, Wiley Publishing Inc, Sayfalar: 5-18, 2007
38. Gehtland J., Galbraith B., Almaer D., “Pragmatic Ajax, A Web 2.0 Primer”, The Pragmatic Programmers LLC, sf:1, 2005
39. Babin L., “Beginning Ajax with PHP, From Novice to Professional”, Apress Yayıncılık, Sayfa: 6, 2007
40. Crane D., Bibeault B., Sonneveld J., Goddard T., Gray C., Venkataraman R. Ve Walker J., “Ajax in Practice”, Manning Publications Co., Sayfalar: 5,6, 2007
41. Eichorn J., “Understanding AJAX: Using JavaScript to Create Rich Internet Applications”, August 11, 2006, Sayfalar:5-50
42. Ballard P., “SAMS Teach Yourself Ajax in 10 Minutes”, Sayfalar: 6-78, 28 Nisan 2006
43. İnternet: <http://www.ipucu.web.tr/aspders17.php>, Erişim tarihi:
44. Ullman C., Dykes L., “Beginnig AJAX”, Wiley Publishing, Inc., Sayfa: 44-100, 2007
45. Zakas N. C., McPeak J., Fawcett J., “Professional Ajax”, Wrox Press, Chapter 2: Ajax Basics, Ajax Communication Techniques, 2006
46. İnternet: Cross-Browser (Çapraz Tarayıcı Desteği), <http://blog.ceyhunaksan.com/cross-browser-capraz-tarayici-destegi>, Erişim: 11.03.2009
47. İnternet: <http://www.yazgelistir.net/viewtopic.php?p=160&sid=a49d3c26dd8e14e02e65bb222d6bcc39>, Erişim: 31.03.2009
48. Jacobi J., Fallows J. R., “Pro JSF and Ajax: Building Rich Internet Components”, Apress, Sayfa: 175, 2006
49. Schutta N. T., Asleson R., “Pro Ajax and Java”, Apress, Sayfa: 10, 2006
50. Lauriat S. M., “Ajax Architecture and Best Practices”, Pearson Education, Inc., Sayfa: 25, 2008
51. Powers S., “Adding Ajax”, O'Reilly yayıncılık, June 01, 2007
52. İnternet: Asenkron ve Senkron Meselesi, <http://AJAX-tr.com/asenkron-ve-senkron-meselesi>, Erişim tarihi: 02.11.2008
53. Holzner S., “Ajax For Dummies”, Wiley Publishing, Inc., sf: 40, 2006

54. İnternet: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>, Erişim tarihi: 06.03.2009
55. Darie C, Brinzarea B., Cherecheş-Toşa F. , Bucica M., “AJAX and PHP, Building Responsive Web Applications” , Packt Publishing, Sayfa:42, 2006
56. Olson S. D., “Ajax on Java”, O'Reilly Yayıncılık, Chapter 2, February 23, 2007
57. İnternet: <http://iibf.erciyes.edu.tr/gg/web/javascript.pdf>, Erişim tarihi: 06.03.2009
58. İnternet: <http://www.mutasyon.net/kitaplar.asp> hakkı öcal javascript ders notu sf:8, Erişim tarihi: 06.03.2009
59. İnternet: <http://www.bilgisayarogren.com/java5.doc>, Erişim tarihi: 06.03.2009
60. Asleson R, Schutta N. T., “Foundations of Ajax”, 2006, sf:29
61. Wells C., “Securing Ajax Applications”, O'Reilly Media Inc, sf:86, July 2006
62. İnternet: http://www.w3schools.com/dom/dom_intro.asp, Erişim tarihi: 19.03.2009
63. İnternet: <http://www.w3.org/TR/REC-DOM-Level-1/introduction.html>, Erişim tarihi: 19.03.2009
64. İnternet: <http://www.berkerpeksag.com/blog/dom-document-object-model-belge-nesne-modeli>, Erişim tarihi: 19.03.2009
65. İnternet: http://www.rssnedir.com/xml_nedir.php, Erişim tarihi: 19.03.2009
66. Jacobs S., “Beginning XML with DOM and Ajax From Novice to Professional”, Apress Yayıncılık, sf:26,31 , 2006
67. İnternet: [http://www.cclub.metu.edu.tr/nenedir/XML+\(Extensible+Markup+Language\)+Nedir%3F](http://www.cclub.metu.edu.tr/nenedir/XML+(Extensible+Markup+Language)+Nedir%3F), Erişim tarihi: 20.03.2009
68. İnternet: http://belgeler.org/elkitabi/xmldoc-howto_struct.html, Erişim tarihi: 20.03.2009
69. İnternet: <http://www.verivizyon.com/detail.asp?cid=149>, Erişim tarihi: 20.03.2009
70. İnternet: <http://www.bilgisayardershanesi.com/xml1.htm>, Erişim tarihi: 21.03.2009
71. Johnson D., White A., Charland A., “ENTERPRISE AJAX, Strategies for Building High Performance Web Applications”, Prentice Hall Yayıncılık, Sayfa: 108, 2008
72. İnternet: <http://www.w3.org/TR/xhtml1/#dtds>, Erişim tarihi: 01.04.2009
73. İnternet: <http://www.csharpnedir.com/makalegoster.asp?MIId=51>, 52, Erişim tarihi: 01.04.2009
74. İnternet: <http://office.microsoft.com/tr-tr/infopath/HP010967301055.aspx>, Erişim tarihi: 01.04.2009
75. İnternet: <http://www.kodpaylas.com/viewtopic.php?t=959&sid=e5167bdea935943ed9bbdc91ee218c53>, Erişim tarihi: 01.04.2009

76. İnternet: <http://www.w3schools.com/Xhtml/>, Erişim tarihi: 02.04.2009
77. İnternet: http://www.zvon.org/xxl/XMLTutorial/Output_tur/example6.html, Erişim tarihi: 02.04.2009
78. İnternet: <http://www.bidb.itu.edu.tr/?d=837>, Erişim tarihi: 03.04.2009
79. İnternet: <http://www.cssnedir.com/>, Erişim tarihi: 03.04.2009
80. İnternet: <http://tr.wikipedia.org/wiki/XSLT>, Erişim tarihi: 03.04.2009
81. İnternet: <http://www.bilisim-kulubu.com/sozluk/sozluk.php?e=XSLT>, Erişim tarihi: 08.04.2009
82. İnternet: <http://www.sanalkurs.net/adobe-flex-nedir--1081.html>, Erişim tarihi: 10.04.2009
83. İnternet: <http://www.adobe.com/tr/products/flex>, Erişim tarihi: 11.04.2009
84. İnternet: <http://www.trflex.com/?p=4>, Erişim tarihi: 11.04.2009
85. Özer Z., Varol A., “Web’de Ajax Tekniğinin Bant Genişlik Optimizasyonuna Etkisi”, Bilimde Modern Yöntemler Sempozyumu, 2008
86. İnternet: [http://tr.wikipedia.org/wiki/AJAX\(programlama\)](http://tr.wikipedia.org/wiki/AJAX(programlama)), Erişim tarihi: 01.06.2009
87. İnternet: <http://ajax-tr.com/ajaxi-anlamak/>, Erişim tarihi: 01.06.2009

ÖZGEÇMİŞ

Zeynep ÖZER AYDOĞMUŞ

zeynep_ozer@msn.com

Fırat Üniversitesi Teknik Eğitim Fakültesi

Elektronik Bilgisayar Eğitimi Bölümü

ELAZIĞ

1983 yılında Elazığ'da doğdu. İlk, orta ve lise öğrenimini Elazığ'da tamamladıktan sonra 2002 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Bilgisayar Öğretmenliği Bölümü'nü kazandı. 2006 yılında bu bölümden mezun oldu. 2006 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Ana Bilim Dalı Bilgisayar Sistemleri Bilim Dalı'nda yüksek lisans eğitimine başladı. 2007 yılı şubat ayında Elazığ'ın Karakoçan ilçesinin Yenice köyünde Şh. J. Kd. Ütğm. Mahir Özdemir İlköğretim Okulu'nda Bilgisayar Öğretmeni olarak göreve başladı. Şuan Elazığ Korgeneral Hulusi Sayın Lisesi'nde Bilgisayar Öğretmeni olarak görevini sürdürmektedir. Yabancı dili İngilizce'dir. Fatih AYDOĞMUŞ ile evlidir.

EK

Katalog çalışmasının uygulama dosyası olan cata.mxml, kullan.mxml ve imagepage.mxml dosyalarının kaynak kodları aşağıdaki gibidir.

Cata.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
    xmlns:controls="qs.controls.*"
    backgroundColor="#FFFFFF"
    creationComplete="init();" color="#FFFFFF" xmlns:local="*" xmlns:comp="comp.*"
    horizontalScrollPolicy="off" verticalScrollPolicy="off" borderColor="#050505" borderStyle="solid"
    borderThickness="2">
    <mx:Style source="style.css" />
    <mx:Script>
    <![CDATA[
        import mx.utils.StringUtil;
        import mx.rpc.soap.LoadEvent;
        import mx.controls.Alert;
        import mx.rpc.events.FaultEvent;
        import qs.controls.FlexBook;
        import qs.controls.flexBookClasses.FlexBookPage;
        import qs.controls.flexBookClasses.FlexBookEvent;
        import qs.controls.flexBookClasses.WBookPageImpl;
        import qs.caching.ContentCache;

        import mx.rpc.events.ResultEvent;
        import mx.rpc.http.HTTPService;
        import mx.core.UIComponent;
        import mx.managers.PopUpManager;
        import comp.ResimSec;
        import Kullan;
        import mx.core.ScrollPolicy;
        import mx.collections.ArrayCollection;
        private var a:XML;

        [Bindable]

        private var resimler:ArrayCollection;//comboresim
```

```

private var dataSet:HTTPService;
private var resimdizi:Array=new Array();
                private var zoomYap:Boolean=false;
private function init():void {

    maske.x=0;
    maske.y=0;
    maske.width=application.width/2
    maske.height=application.height-40
    addEventListener(MouseEvent.MOUSE_MOVE,maskele);

    addEventListener("tamam", sayfa_secildi);

    dataSet = new HTTPService();
    dataSet.url = ("data/images.xml");
    dataSet.resultFormat = "e4x";
    dataSet.send();
    dataSet.addEventListener(FaultEvent.FAULT, onFault);
    dataSet.addEventListener(ResultEvent.RESULT, onResult);
    var w:Number=new Number;
    var h:Number=new Number;
    w=Application.application.width;
    h=Application.application.height;

    book.width=w;
    book.height=h-40;

    yukleme.visible=false;
    yukleme.source="assets/loading.gif"
        yukleme.addEventListener(Event.COMPLETE,yukleme_yuklendi);
}

private function yukleme_yuklendi(event:Event):void {
    yukleme.x=(application.width/2)-100
    yukleme.y=(application.height-40)/2;
}

private function maskele(e:MouseEvent):void{
if(zoomYap){

```



```

        if((mouseX>(application.width/2)) && (mouseY<application.height-40) ){
            maske.x=0
            img2.visible=false;
            img1.visible=true;

img1.x=(application.width/2)-mouseX;
img1.y=-1*mouseY
        }

else if((mouseX<(application.width/2)) && (mouseY<application.height-40) ){
            maske.x=application.width/2;
            img1.visible=false;
            img2.visible=true;

            img2.x=-1*mouseX;
            img2.y=-1*mouseY
        }
    }

    //maske.x=mouseX;
    //maske.y=mouseY;
    }

    private function onFault(e:Event):void {
        trace(e);
    }

    private function onResult(e:ResultEvent):void {

        a=new XML(e.result);
        book.content = dataSet.lastResult..image;
        book.addEventListener(FlexBookEvent.TURN_START,loadContent);
        book.addEventListener(FlexBookEvent.TURN_END,loadContent);
    }

    private function sayfa_secildi(event:ResimSec):void
{

```

```

        sayfa.text=event.yeniResim;
        git();
        //mx.controls.Alert.show("ss",event.yeniResim);
    }

    private function loadContent(event:FlexBookEvent):void {
        setGorunum();
        var page:ImagePage = ImagePage(event.renderer);
        page.load();

        sayfa.text=(book.currentPageIndex*2).toString();
        // resimdizi[book.currentPageIndex]=page.thumb.source;
        // mx.controls.Alert.show("",book.currentPageIndex.toString());
    }

    private function buyuk1_yuklendi(event:Event):void {
        yukleme.visible=false;
        img1.width=application.width;
        img1.height=(application.height-40)*2;
        img1.x=0;
        img1.y=0;
    }

    private function buyuk2_yuklendi(event:Event):void {
        yukleme.visible=false;
        img2.width=application.width;
        img2.height=(application.height-40)*2;
        img2.x=0;
        img2.y=0;
    }

    private function tumsayfalar():void {
        Kullan.show_window(this,false,application.width-10,(application.height-60)/2);
        customMove.end();
        customMove.play();
    }

    private function zoom():void {
        if(zoomer.label==""){
            zoomer.label="-"
        }
        if(img2.source!=a.image[book.currentPageIndex*2].@zoom){
            yukleme.visible=true;
        }
        img2.addEventListener(Event.COMPLETE,buyuk2_yuklendi);
    }

```

```

img2.source=a.image[book.currentPageIndex*2].@zoom;

if(a.image.length()>book.currentPageIndex*2+1){
    img1.addEventListener(Event.COMPLETE,buyuk1_yuklendi);
    img1.source=a.image[book.currentPageIndex*2+1].@zoom;
}
zoomYap=true;
}
else if(zoomer.label=="-"){
    zoomer.label="+";
    setGorunum();
}
}

private function setGorunum():void {
zoomYap=false;
zoomer.label="+";
img1.visible=false;
img2.visible=false;

}

private function next():void {
//mx.controls.Alert.show(book.currentPageIndex.toString(), book.pageCount.toString());

if(book.currentPageIndex+1 < book.pageCount){
// mx.controls.Alert.show("--",resimdizi[book.currentPageIndex])
setGorunum();
book.turnToPage(book.currentPageIndex + 1);
}
}

private function previous():void {
if(book.currentPageIndex > 0){
setGorunum();
book.turnToPage(book.currentPageIndex -1);
}
}

private function git():void {

var s_no:uint = new uint;
s_no=Math.floor(uint(sayfa.text)/2)

```

```

        if(s_no>0 && s_no<book.pageCount){
            book.turnToPage(s_no);
            setGorunum();
        }
    }
]]>
</mx:Script>
<!-- 614x453 -->
        <mx:Parallel id="customMove" target="{Kullan}">
            <mx:Move duration="2000" xTo="{(100) / 2}" yTo="{(200) / 2}" />
            <mx:WipeRight duration="2000" />
        </mx:Parallel>

        <mx:VBox>
<mx:HBox>
        <controls:FlexBook id="book" width="100" height="100" x="0" y="0" horizontalCenter="0"
verticalCenter="0"
        animateCurrentPageIndex="true"
        showCornerTease="false"
        edgeAndCornerSize="150"
        itemRenderer="ImagePage"
        animatePagesOnTurn="true"
        itemSize="halfPage"          cornerRadius="0"          themeColor="#1D99E8"
backgroundAlpha="1.0" alpha="1.0">
        </controls:FlexBook>

    </mx:HBox>

        <mx:HBox height="30" width="100%" verticalAlign="middle"
horizontalAlign="center" color="#B65F5F" backgroundColor="#121313">
            <mx:Spacer width="10">

                </mx:Spacer>
                <mx:TextInput borderColor="FFFFFF" width="34" height="24" id="sayfa"
color="FFFFFF" fontSize="11" text="0" alpha="0.01"/>
                <mx:Button click="git()" color="ffffff" fontWeight="bold" fontSize="11"
width="114" height="24" borderColor="#000000" alpha="0.64" fillAlphas="[0.0, 0.0, 0.0, 0.0]"
label="Sayfaya Git" themeColor="#EDF2F5" textRollOverColor="#EDD617"/>
                <mx:Spacer width="60%">

```

```

        </mx:Spacer>
        <mx:Button click="previous()" color="#ffffff" fontWeight="bold" fontSize="11"
width="70" height="24" borderColor="#000000" alpha="0.64" fillAlphas="[0.0, 0.0, 0.0, 0.0]"
label="Geri" themeColor="#EDF2F5" textRollOverColor="#EDD617"/>
        <mx:Button click="next()" color="#ffffff" fontWeight="bold" fontSize="11"
width="70" height="24" borderColor="#000000" alpha="0.64" fillAlphas="[0.0, 0.0, 0.0, 0.0]"
label="İleri" themeColor="#EDF2F5" textRollOverColor="#EDD617"/>
        <mx:Button click="tumsayfalar()" color="#ffffff" fontWeight="bold" fontSize="11"
width="134" height="24" borderColor="#000000" alpha="0.64" fillAlphas="[0.0, 0.0, 0.0, 0.0]"
label="Tüm Sayfalar" themeColor="#EDF2F5" textRollOverColor="#EDD617"/>
        <mx:Button id="zoomer" click="zoom()" color="#ffffff" fontWeight="bold"
fontSize="14" width="44" height="24" borderColor="#000000" alpha="0.64" fillAlphas="[0.0, 0.0, 0.0,
0.0]" label="+" themeColor="#EDF2F5" textRollOverColor="#EDD617" fontFamily="Arial"/>
        <mx:Spacer width="5"/>
        </mx:HBox>
        <mx:Spacer height="10"/>
    </mx:VBox>
    <mx:Canvas id="maske" x="224" y="168" width="200" height="168"
verticalScrollPolicy="off" horizontalScrollPolicy="off" borderThickness="0" borderColor="#000000"
borderStyle="solid">
        <mx:Image id="img1" x="0" y="0"/>
        <mx:Image id="img2" x="40" y="0"/>
    </mx:Canvas>
    <mx:Image x="10" y="144" id="yukleme"/>
</mx:Application>

```

Kullan.mxml

```
<?xml version="1.0" encoding="utf-8"?>

<mx:TitleWindow xmlns="*" xmlns:mx="http://www.adobe.com/2006/mxml"
    creationComplete="on_creation_complete()"
    styleName="popupWindow" showCloseButton="true"
    paddingBottom="0" paddingTop="0" paddingLeft="0" paddingRight="0"
    height="200" width="100%" alpha="0.9" borderAlpha="1">
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            import mx.collections.ArrayCollection;
            import mx.rpc.events.ResultEvent;

            [Bindable]
            private var sayfalar:ArrayCollection;

[Bindable]
private var intro:String = "";
import mx.events.CloseEvent;
import mx.managers.PopUpManager;
import mx.core.Application;
import mx.events.DragEvent;
import mx.managers.DragManager;
import mx.core.DragSource;
import comp.ResimSec;
//import korax.controls.ColorPicker.ResimSecEvent;
private var m_parent:DisplayObject = null;
private function myResult(event:ResultEvent):void
    {
        sayfalar = event.result.sayfalar.sayfa;
    }
    private function sayfaac(event:Event):void
    {
        wnd.with_last_preved();
        //mx.controls.Alert.show("",myTile.selectedItem.git.toString());
        //uyeler = event.result.sayfalar.sayfa;
    }
        ]]>
    </mx:Script>
</mx:TitleWindow>
```

```

private function on_creation_complete():void
{
    myHTTP.send();
}

private static var wnd:Kullan = null;
public static function show_window(parent:DisplayObject,
modal:Boolean,w:Number,h:Number):void
{
    if (wnd == null) {
        wnd = new Kullan();
        wnd.addEventListener(mx.events.CloseEvent.CLOSE, on_close_event);
    }
    wnd.m_parent = parent;
    if (!wnd.isPopUp) {
        PopUpManager.addPopUp(wnd, DisplayObject(Application.application), modal);
        PopUpManager.centerPopUp(wnd);
    }
    else {
        PopUpManager.bringToFront(wnd);
        //      wnd.start_init();
    }
    wnd.x=0;
    wnd.y=0;
    wnd.width=w;
    wnd.height=h;
    wnd.myTile.width=w-20;
    wnd.myTile.height=h-40;
}

private function with_last_preved():void
{
    if (m_parent != null) m_parent.dispatchEvent(new ResimSec("tamam",
myTile.selectedItem.git));
}

private static function on_close_event(event:CloseEvent):void
{
    if (wnd == null) return;
    PopUpManager.removePopUp(wnd);
    wnd = null;
}

```

```

}
private function yardim():void
{
//      Kullan.show_window(this,false);
}

    ]]>
</mx:Script>
    <mx:HTTPService id="myHTTP" url="data/onizle.xml"
        result="{myResult(event)}" showBusyCursor="true"/>

    <mx:TileList      id="myTile"          dataProvider="{sayfalar}"      change="sayfaac(event)"
itemRenderer="comp.UyeSayfasi" width="100%" height="147" rowHeight="110" columnWidth="120"
rowCount="2"/>
</mx:TitleWindow>

```


ImagePage.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml" width="600" height="400"
xmlns:controls="qs.controls.*"
    backgroundColor="#FFFFFF" horizontalScrollPolicy="off" verticalScrollPolicy="off">

    <mx:states>
        <mx:State name="loadingState">
            <mx:AddChild position="lastChild">
                <mx:Image id="loadingContent" horizontalCenter="0" verticalCenter="-6"
source="assets/loading.gif" />
            </mx:AddChild>
        </mx:State>
    </mx:states>

    <mx:transitions>
        <mx:Transition fromState="" toState="loadingState">
            <mx:Fade alphaTo="1" target="{loadingContent}" duration="1000"/>
        </mx:Transition>
        <mx:Transition toState="" fromState="loadingState">
            <mx:Sequence>
                <mx:Fade alphaTo="0" target="{loadingContent}" duration="1000"/>
                <mx:RemoveChildAction target="{loadingContent}" />
            </mx:Sequence>
        </mx:Transition>
    </mx:transitions>

    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            private var _loaded:Boolean = false;
            public function load():void {
                if(_loaded)
                    return;
                _loaded = true;
                full.source = data.@source.toString();

                // mx.controls.Alert.show("",full.source)
            }
        ]]>
    </mx:Script>
</mx:Canvas>
```

```
        currentState = "loadingState";
    }
    ]]>
</mx:Script>
<controls:SuperImage    left="0"    right="0"    top="0"    bottom="0"    id="thumb"
source="{data.@thumb.toString()}" />
    <controls:SuperImage left="0" right="0" top="0" bottom="0" id="full" complete="currentState = ""/>
</mx:Canvas>
```

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.