

Web Part ve AJAX Teknolojileri ile Kişiselleştirilebilir Portal Uygulaması

Şehra ŞEN, Ata ÖNAL

Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İZMİR
sehra_sen@yahoo.com, ata.onal@ege.edu.tr

Özet: Geliştirilmekte olan projenin amacı ASP .NET 2.0 ve AJAX (Asynchronous JavaScript and XML) teknolojileri kullanılarak verilerin kişiselleştirmeye izin verecek bağımsız birimler halinde sunulduğu bir portal tasarımının ve gerçekleştiriminin yapılmasıdır. Uygulama kapsamında Ege Üniversitesi web sitesi üzerinde yer alan “egetel”, “egeduyuru” isimli hizmetleri ve benzeri hizmetler web servislerine dönüştürülerek portal üzerinde kişiselleştirilebilir bağımsız birimler olarak sunulacaktır. Ayrıca projenin gerçekleştirim aşamalarında kullanılan teknolojilerin anlatımlarını içeren bir yol haritasının oluşturulması hedeflenmektedir.

Anahtar Sözcükler: Portal, ASP .NET 2.0, AJAX, Web Servisi.

A Personalizable Portal Application Using Web Part And Ajax Technologies

Abstract: The goal of the project being developed is to design and implement a personalizable portal using ASP .NET 2.0 and AJAX (Asynchronous JavaScript and XML) technologies. In this context the services named “egetel”, “egeduyuru” and similar services on the web site of Ege University will be reimplemented as web services. These services will be presented as personalizable, independent modules on the new portal. Furthermore, a roadmap including the technologies used at implementation will be formed.

Keywords: Portal, ASP .NET 2.0, AJAX, Web Service.

1. Giriş

Günümüzde web siteleri büyük bilgi kaynaklarıdır. Kötü tasarlanmış bir web sitesi kullanıcının bilgi gereksinimini yeterince karşılamayabilir. Günümüzde portallar genellikle verilerini belirli bir derecede kişiselleştirmeyi sağlayacak şekilde bağımsız birimler olarak düzenlemektedir. Kullanıcılar bu parçaları kendi bireysel çalışma biçimlerine uygun şekilde düzenleyebilmektedir [1].

Web kişiselleştirme bir web sitesinin sunumunun, web sitesi tarafından sağlanan bilgi veya servislerin bir veya bir küme kullanıcının açık veya üstü kapalı tercihlerine uygun şekilde

değiştirilebilmesidir [2][3]. Web kişiselleştirme bir web kullanıcısının kendi portal web sayfasının içeriğini ve görünümünü uyarlayabilme yeteneğidir. Günümüzdeki web yapısı ile bilgiye erişim siteler üzerinde gezinmeyi gerektirmektedir. Oysa kişiselleştirilmiş bir sayfa bilgiyi bize getirmektedir. Böylece istediğimiz şeyi istediğimiz zamanda görmemizi sağlamaktadır. Kişiselleştirilmiş bir sayfa ile web üzerinde dağılmış olan bilgiyi kullanarak kendimize özel bilgi alma kaynağımızı oluşturabilme yeteneği oldukça değerlidir [4].

Ege Üniversitesi web sitesinin mevcut yapısı kişiselleştirmeye izin vermemektedir. Proje kapsamında ASP .NET 2.0 teknolojileri kulla-

nılarak Ege Üniversitesi web sitesi kullanıcılarının bilgi gereksinimlerini karşılayabilmeleri için belirli bir oranda kişiselleştirmenin sağlanabildiği bir portalın tasarlanması ve oluşturulması hedeflenmektedir. ASP .NET 2.0'ın sağlamış olduğu yenilikler ile Google ig ve Live.com gibi sitelerin sağlamış olduğu gelişmiş özelliklere sahip bir portal geliştirilebilecektir. AJAX teknolojisi ile ağ uygulamalarının daha hızlı yanıt vermesi sağlanabilecektir. Ayrıca Ege Üniversitesi web sitesi üzerinde yer alan “egetel” isimli telefon rehberi, “ege-duyuru” isimli duyuru hizmetleri ve benzeri hizmetler portal üzerinde kişiselleştirilebilir bağımsız birimler olarak sunulmak üzere web servislerine dönüştürülecektir. Böylece veritabanı erişiminde soyutlama sağlanarak bu hizmetlerin farklı uygulamalar tarafından da kullanılabilmesi sağlanacaktır.

2. Web Servisleri

Web servisleri platform ve programlama dilinden bağımsız, açık Internet standartlarına dayanan, veri ve nesne paylaşabilmemizi sağlayan modüllerdir. Web servisleri HTTP, XML, SOAP, WSDL, DISCO, UDDI gibi standartlar üzerinde çalışır.

Web servisleri HTTP(Hypertext Transfer Protocol) protokolü üzerinden bilgi alışverişi yapmaktadır. Bu nedenle Internet üzerinden bir web servisine erişilebilmektedir. XML(Extensible Markup Language), web servisleri ve servis istemcisi arasında bilgi alışverişinin sağlanmasında kullanılan bir standarttır. SOAP(Simple Object Access Protocol) web servisine platformdan bağımsız çağırımlar yapılabilmesini sağlamaktadır. WSDL(Web Service Description Language), web servisinde bulunan fonksiyonların aldığı parametrelerin ve sonuç olarak döndürdüğü bilginin türünü tanımlamak için kullanılır. DISCO(Discovery Protocol) bir sunucu üzerinde, UDDI(Universal Description, Disco-

very and Integration) ise Internet üzerinde paylaşımına açılmış bütün web servislerinin düzenlenmesinde kullanılır.

Web servisleri dağıtık yapıdaki uygulamaların bir parçası olarak geliştirilirler. Bir web servisine erişmek ve servise ait metodların çalıştırılabilmesi için servis istemci rolündeki kullanıcının ilgili servisin sadece adresini bilmesi yeterlidir. Kullanılacak web servisinin adresi bilinmiyorsa bir UDDI dizini yardımı ile bir web servisi bulunur. UDDI servisi istemciye web servisinin .disco uzantılı dosyasının adresini döndürür. DISCO dosyası içerisinde web servisini tanımlayan kaynaklara bağlantı bulunmaktadır. DISCO dosyasına yapılan istek ile web servisinin tanımlama dosyasına yönlendirilir. Web servisinin tanımlama dosyası olan WSDL dosyasına istek yapılarak servis biçimi öğrenilir. Sonu olarak web servisinin metodları istemci tarafından çalıştırılmaya hazırdır [5].

Visual Studio.NET ile yeni bir web servisi oluşturmak için izlenmesi gereken adımlar kısaca şöyledir:

- File menüsünde New Project seçeneğine tıklanır.

- Açılan pencerede uygulama dili seçildikten sonra ASP.NET Web Service seçeneğine tıklanır ve seçim onaylanır.

Visual Studio.NET yerel makinede web servisi için gerekli klasörleri otomatik olarak oluşturacaktır. Bir web servis projesinde yer alan dosyalar ve sınıfların oluşturulma amacına göre şöyle özetleyebiliriz:

- Global.asax ve Global.asax.cs : Uygulama ve oturum bazında olay yönetimi yapılabilirliği sağlar.

- Service.asmx ve Service.asmx.cs : .asmx uzantılı dosyalar web servisi uygulama

dosyalarıdır. Service.asmx'in görevi IIS tarafında web servisinin bulunabilmesini sağlamak ve web servisi için giriş noktası olmaktır. Service.asmx.cs dosyası yazılan program kodlarını içeren dosyadır.

Web servisindeki metodların web metodu olarak algılanması için fonksiyonların başında [WebMethod] ifadesi yer almalıdır. Aksi halde web servisimizin arayüzünde bu metod bulunmayacaktır [5].

```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

[WebService(Namespace = "http://localhost/", Description = "E.Ü. Telefon rehberi ile ilgili metodlar içerir.", Name="egetelWS")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

public class Service :
System.Web.Services.WebService
{
    [WebMethod(Description = "Kullanıcıya ait telefon numarasını döndürür.")]
    public int GetPhoneNumber(string userName) {
        int phoneNumber;
        //İlgili işlemler burada yapılır.
        return phoneNumber;
    }
}
```

Şekil 1 C#.NET ile örnek web servisi oluşturmak

Web servisi için gerekli dll dosyası proje derlendiğinde otomatik olarak bin klasörü içerisinde oluşturulmaktadır. Oluşturulan web servisini test etmek için projenin çalıştırılması gerekmektedir. Web servisinin ilgili metodu, ilgili bağlantı üzerine tıklanarak çalıştırılabilir.

egetelWS

E.Ü. Telefon rehberi ile ilgili metodlar içerir.

The following operations are supported. For a formal definition, please review the [Service Description](#).

- **GetPhoneNumber**
Kullanıcıya ait telefon numarasını döndürür.

Şekil 2 Proje çalıştırıldıktan sonra web servisine ait metodların gösterildiği ekran görüntüsü.

Web servisini kullanacak uygulama bir Windows uygulaması, bir ASP .NET uygulaması veya bir mobil uygulama olabilir. Visual Studio.NET ile geliştirilen uygulamaya bir web servisi eklemek için izlenmesi gereken adımlar kısaca şöyledir:

- Solution Explorer'da proje öğesinin üzerine sağ tıklanır.
- Açılan menüden Add Web Reference seçeneğine tıklanır.
- Ekranı gelen web servisi ekleme seçenekleri sunan Add Web Reference diyalog penceresinde Web services on the local machine bağlantısına tıkladığında localhost üzerindeki tüm web servislerinin bir listesini elde ederiz.
- Uygulaya eklemek istenilen web servisi üzerine tıklandıktan sonra Add Reference butonuna tıklanarak web servisi uygulamaya eklenir.

Web servisi uygulamaya eklendikten sonra metodunu kullanmak istediğimiz sınıfa ait bir nesne oluşturularak, ilgili sınıfın metodlarına erişim sağlanmaktadır.

```
localhost.egetelWS egetelWS = new  
localhost.egetelWS();  
TextBox1.Text =  
egetelWS.GetPhoneNumber(userName)  
.ToString();
```

Şekil 3 Uygulamaya eklenen web servisi metodlarının kullanılması.

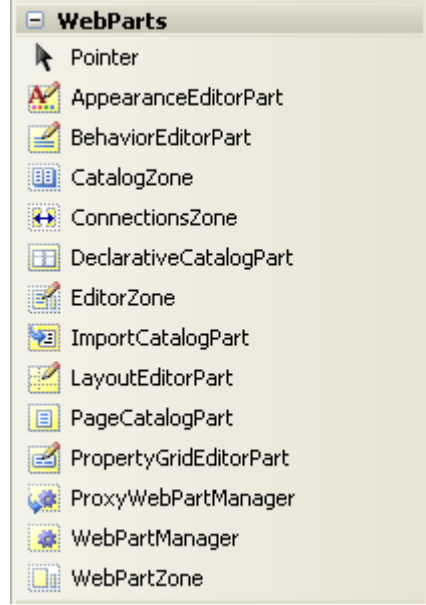
3. Web Part

Bir portal sunduğu seçeneklerle kullanıcıya kendi sayfasını düzenleyebilme olanağı verir. Farklı firmaların farklı portal çözümleri mevcuttur. ASP .NET 2.0 diğer geliştirme ortamlarından farklı olarak kolay bir şekilde kişiselleştirilebilen portal uygulamaları geliştirmek için hazır kontroller ve arabirimler sunmaktadır. ASP .NET 2.0'da portal web part'lerden oluşur. Bir web part görsel bir bileşendir. ASP .NET 2.0'da bir web part bir sunucu kontrolü, kullanıcı kontrolü, kullanıcı tarafından oluşturulmuş özel bir kontrol veya başkaları tarafından oluşturulmuş bir bileşen olabilir [6].

ASP .NET 2.0 WebPart kontrolleri kullanıcılara web sayfalarının içeriğini, görünümünü ve davranışını doğrudan bir tarayıcı ile değiştirebilme olanağı sağlayan bir küme tümleşik kontroldür. Yapılan değişiklikler kullanıcı bazlı kaydedilmektedir. Tüm bu işlevselliğin sağlanması çok fazla kodlamayı gerektirmektedir [7].

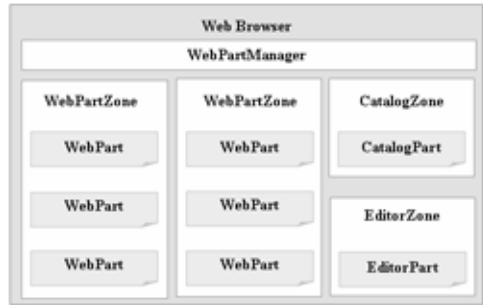
Visual Web Developer 2005 Express veya Visual Studio 2005 editörlerinde web part'lar için oluşturulan kontroller aşağıda yer alan şekilde (Şekil 4) gösterilmektedir.

WebPart tipi abstract bir sınıf olup, bir web part bileşeni için gerekli tüm temel alt yapıyı sunmaktadır [10]. Web part'lar web zone'lar içinde yer alırlar ve web zone'lar da web part manager'lar tarafından yönetilir.



Şekil 4 Web Part'lar için oluşturulan kontroller.

ASP .NET 2.0'da ToolBox'ta yer alan standart bir sunucu kontrolü bir web part olarak kullanılabilir. Bunun için yapılması gereken ilgili kontrolü WebPartZone'a eklemektir. Kontrol kendi özelliklerini kaybetmeden bir web part muamelesi görür.



Şekil 5 Web Part'ları kullanan bir aspx sayfasının mimarisi.

- WebPartManager Kontrolü

WebPartManager kontrolünün görevi web

form üzerindeki web part ve zone'lar için yönetimsel bir altyapı sunmaktır. Web part kontrolleri içeren her sayfa bir WebPartManager kontrolüne sahip olmalıdır [5]. Bu kontrol sayfanın herhangi bir yerine bırakılabilir çünkü çalışma zamanında görünür bir arayüze sahip değildir.

```
<asp:WebPartManager id=
"WebPartManager1"
runat="server" />
```

Şekil 6 WebPartManager belirtimi.

- WebPartZone Kontrolü

Web part'lar ancak WebPartZone kontrolleri içerisinde yer alabilirler. WebPartZone kontrolü şablon bir kontroldür ve içerdiği web part kontrollerinin görsel bütünlüğünü ayarlar. Editörümüzün özellikler kısmından baktığımızda WebPartZone'un çok fazla özelliğe sahip olduğunu görürüz. Bir aspx sayfasında birden fazla web part zone bulunabilir.

```
<asp:WebPartZone id="WebPartZone1"
runat="server" Title="Title1">
<ZoneTemplate>
...
</ZoneTemplate>
</asp:WebPartZone>
```

Şekil 7 WebPartZone belirtimi.

- EditorZone Kontrolü

EditorZone kontrolü kullanılarak kullanıcılara web part'ların görünümünü değiştirebilmelerini sağlayacak AppearanceEditorPart kontrolü, yerleşimini değiştirebilmelerini sağlayacak LayoutEditorPart kontrolü ve davranışını değiştirebilmelerini sağlayacak BehaviorEditorPart kontrolü sunulabilir. EditorZone kontrolünün görüntülenebilmesi için WebPartManager kontrolü "Edit" moduna getirilmelidir.

```
<asp:EditorZone id=
"EditorZone1" runat="server"
<ZoneTemplate>
<asp:AppearanceEditorPart id=
"AppearanceEditorPart1"
runat="server" />
<asp:LayoutEditorPart id=
"LayoutEditorPart1"
runat="server" />
<asp:BehaviorEditorPart id=
"BehaviorEditorPart1"
runat="server" />
</ZoneTemplate>
</asp:EditorZone>
```

Şekil 8 EditorZone belirtimi.

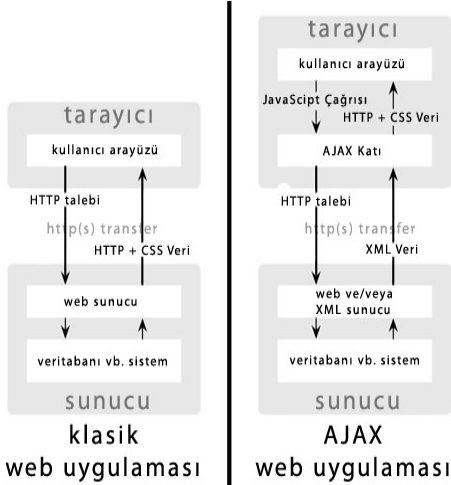
4. Ajax

Günümüzde Internetin yaygınlaşması ile birlikte Internet kullanıcıları web sayfalarının da masaüstü programlar gibi çalışabilmesini istemektedir. Bunu sağlamak için birkaç teknoloji bir araya getirilerek yeni bir yaklaşım oluşturulmuştur. Bu yaklaşıma AJAX adı verilmiştir.

Web sayfaları bir sunucu bilgisayar üzerinde belirli kod parçalarının işletilmesiyle hazırlanır ve istemci bilgisayara gönderilir. Kullanıcının web sayfası üzerinde yaptığı her işlem için sunucuya bir istek gönderilir ve sunucu bilgisayar ilgili web sayfasını yeniden oluşturup istemci bilgisayara gönderir. AJAX teknolojisi sayesinde kullanıcının belirli bir komut vermesiyle birlikte sayfanın tamamen yenilenmeden ilgili işlemin yürütülebilmesi sağlanmaktadır.

AJAX daha hızlı, daha dinamik ve eş zamanlı çalışan web uygulamalarının geliştirilmesi için kullanılan bir yaklaşımdır [11]. Birçok web teknolojisi bir arada kullanılarak, sayfalar üzerinde birbirinden özgür alanlar oluşturabilmekte ve bu alanlar ayrı ayrı takip edilebilmekte

tedir [7]. AJAX, Javascript sayesinde tarayıcı içerisinde çalışan çeşitli metodlara sahiptir. Bu metodlar yardımıyla sayfa üzerinde oluşan çeşitli olaylarda (buton tıklama, üzerine gelme vs.) sunucu ile veri alışverişinde bulunulur ve gelen verilere göre sayfa üzerinde çeşitli değişiklikler yapılır.



Şekil 9 Klasik web uygulaması ve AJAX uygulamasının karşılaştırılması [7][8].

Klasik web uygulamalarında sunucu ile iletişim senkron olarak yürütülmektedir. Ancak AJAX uygulamalarında sunucu iletişimi asenkron olarak gerçekleştirilmektedir.

AJAX'ın temelinde XMLHttpRequest nesnesi yer alır. AJAX uygulamalarını masaüstü uygulamalara benzer kılan teknoloji bu nesnedir. XMLHttpRequest nesnesi bize, kendisiyle gönderilen sorguları, başka bir fonksiyon içerisinde yorumlama imkanı sağlar. Bu demektir ki XMLHttpRequest ile bir dosyaya bir sorgu gönderilir, alınan sonuç başka bir sayfa içerisine yüklenme ihtiyacı hissedilmeden kullanılır [9].

Yukarıdaki özellikler sayesinde web sayfalarının hızlı veri alışverişi yapabilmesi ve bir ma-

saüstü uygulama kadar kolay kullanılabilmesi sağlanmaktadır. Veri iletişiminin hızlı olması, AJAX destekli sayfalarda kullanıcı ile web sayfası arasındaki iletişimin ve web sayfasının görsel özelliklerinin klasik sayfalara göre daha fazla gelişmesine imkan sağlamaktadır. Kullanıcının yazı girişlerinin otomatik tamamlanması, yüklenmekte olan kısmın “yükleniyor” gibi uyarı verebilmesi gibi birçok pratik işlem gerçekleştirilebilmektedir. AJAX ile eski web uygulamalarına göre çok hızlı çalışan ve çok daha kullanışlı web siteleri geliştirmek mümkündür. AJAX web dünyasının yüzünü değiştirmeye aday bir teknoloji ve yaklaşımdır [7].

5. Sonuç

Proje kapsamında yukarıda yer alan teknolojiler kullanılarak Ege Üniversitesi web sitesi kullanıcılarının bilgi gereksinimlerini daha etkin bir şekilde karşılayabilmeleri için site üzerinde yer alan bir takım hizmetlerin web servisi halinde ayrı birimler olarak sunulduğu, belirli bir oranda kişiselleştirmenin sağlanabildiği bir web sitesinin tasarlanması ve oluşturulması hedeflenmektedir. Böylece web sitesinde sunulan bilgi içeriğine alternatif bir erişim yöntemi sunulması ve web sitesi kullanıcılarının bilgi gereksinimlerinin daha etkin bir şekilde sağlanabilmesi hedeflenmektedir. Projenin geliştirilmesi sırasında izlenen adımlar belgelendirilerek ortaya bir yol haritası çıkacaktır.

6. Kaynaklar

- [1] Building Web Parts - Part 1, http://www.ondotnet.com/pub/a/dotnet/2005/05/23/webparts_1.html, O'Reilly.
- [2] Thomson, L., “A Standard Framework for Web Personalization”, May 2005, st International Workshop on Innovations In Web Infrastructure(IWI 2005).

- [3] Magdalini, P., E., “New Approaches To Web Personalization”, Ph.D. Thesis.
- [4] Kravatz, H., “Designing Web Personalization Features”, STC 2000, Orlando, Florida.
- [5] XML Web Servisleri, <http://www.yazgelistir.com/Makaleler/makaleler.aspx?KatId=1000000009&Kat=XML%20Web%20Servisleri>.
- [6] Demirkol, Z.,” C# ile ASP .NET 2”, Pusula Yayıncılık, 2007.
- [7] Küçüksille, E., U., Yıldırım M. S., “Web Dünyasında Yeni Bir Yaklaşım: AJAX”, 2006, Akademik Bilişim 2006.
- [8] Garrett, J., J., “AJAX: A New Approach to Web Applications”, <http://www.adaptivepath.com/publications/essays/archives/000385.php>, 2005.
- [9] AJAX, <http://www.nothinggrows.net/ajax/>, Nothinggrows.net.
- [10] Kendi Web Part Bileşenlerimizi Geliştirmek, <http://www.csharpnedir.com/makalegoster.asp?Mid=726>.
- [11] Sıvákumar, S., “ASP.NET 2.0”, Alfa Yayınları, 2007.
- [12] Ayers, D., Bruchez E., Fawcett J., Ver-net A., Vlist E. V. D., “Web 2.0 Teknolojileri”, Alfa Yayınları, 2007.