



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**E-ÖĞRENME PROJELERİNDE XML WEB  
SERVİSLERİNİN KULLANIMI VE ÖRNEK  
UYGULAMA**

**Bilgisayar ve Enformatik Müh. İbrahim KURU**

**Enformatik Bölümü  
Enformatik Anabilim Dalı**

**Danışman  
Doç. Dr. Hülya ÇALIŞKAN**

**Temmuz, 2006**

**İSTANBUL**



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**E-ÖĞRENME PROJELERİNDE XML WEB  
SERVİSLERİNİN KULLANIMI VE ÖRNEK  
UYGULAMA**

**Bilgisayar ve Enformatik Müh. İbrahim KURU**

**Enformatik Bölümü  
Enformatik Anabilim Dalı**

**Danışman  
Doç. Dr. Hülya ÇALIŞKAN**

**Temmuz, 2006**

**İSTANBUL**

Bu çalışma 26/09/2006 tarihinde ařağıdaki jüri tarafından ENFORMATİK Anabilim Dalı ENFORMATİK programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Doç.Dr. Hülya Çalışkan (Danışman)  
Gülseçen  
İstanbul Üniversitesi

Yrd.Doç.Dr. Sevinç  
İstanbul Üniversitesi

Yrd.Doç.Dr. Mesut Yalvaç  
İstanbul Üniversitesi

Doç.Dr. Mehpere Timor  
İstanbul Üniversitesi

Yrd.Doç.Dr. Zuhale Tanrıkulu  
Boğaziçi Üniversitesi

## **ÖNSÖZ**

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç.Dr.Hülya ÇALIŞKAN'a en içten dileklerle teşekkür ederim.

Bu çalışma boyunca yardımlarını esirgemeyen çalışma arkadaşlarıma ve göstermiş olduğu yardımlardan dolayı eşime teşekkürü borç bilirim.

**Temmuz, 2006**

**İbrahim KURU**

## İÇİNDEKİLER

ÖNSÖZ .....	i
İÇİNDEKİLER .....	ii
ŞEKİL LİSTESİ.....	iv
TABLO LİSTESİ .....	v
ÖZET .....	vi
SUMMARY .....	vii
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. XML WEB SERVİSLERİ .....</b>	<b>3</b>
<b>2.1. XML HAKKINDA GENEL BİLGİ .....</b>	<b>3</b>
2.1.1 XML Dokümanlarının Yapısı.....	4
2.1.2 Belgenin Tanımlanması.....	5
2.1.3 Verinin Tanımlanması .....	5
2.1.4 Elementler.....	6
2.1.5 Yorum Satırları .....	7
<b>2.2. WEB SERVİSİ NEDİR? .....</b>	<b>7</b>
<b>2.3. WEB SERVİSİ PROTOKOLLERİ.....</b>	<b>9</b>
2.3.1 UDDI.....	10
2.3.2 WSDL .....	11
2.3.3 SOAP .....	11
<b>2.4. XML WEB SERVİSLERİN AVANTAJLARI.....</b>	<b>12</b>
2.4.1. Web Servislerinin Sundukları.....	13
<b>3. E-ÖĞRENMEDE XML WEB SERVİSLER .....</b>	<b>14</b>
<b>3.1. E-ÖĞRENME .....</b>	<b>14</b>
<b>3.2. E-ÖĞRENME DE WEB SERVİS MİMARİLERİ .....</b>	<b>15</b>
3.2.1. Geleneksel Mimari .....	15
3.2.2. Web Servis ile Mimari.....	15
<b>4. ÖRNEK E-ÖĞRENME – XML WEB SERVİS UYGULAMASI ...</b>	<b>19</b>
<b>4.1. TOROS ÜNİVERSİTESİ.....</b>	<b>19</b>
4.1.1. Analiz .....	20
4.1.2. Tasarım.....	25
<b>4.2 CANİK ÜNİVERSİTESİ .....</b>	<b>30</b>
4.2.1. Analiz.....	31
4.2.2. Tasarım.....	36
<b>4.3 GELİŞTİRME ORTAMI VE VERİTABANI.....</b>	<b>41</b>
4.3.1. Geliştirme Ortamı .....	41
4.3.2. Veritabanı .....	41

<b>5. SONUÇ .....</b>	<b>42</b>
<b>KAYNAKLAR .....</b>	<b>43</b>
<b>EK-A .....</b>	<b>45</b>
<b>EK-B .....</b>	<b>47</b>
<b>EK-C .....</b>	<b>50</b>
<b>ÖZGEÇMİŞ.....</b>	<b>55</b>

## ŞEKİL LİSTESİ

<b>Şekil 2.1</b>	: Örnek bir XML dosyası .....	5
<b>Şekil 2.2</b>	: Basit bir DTD örneği.....	6
<b>Şekil 2.3</b>	: XML dosyası element yapısı.....	6
<b>Şekil 2.4</b>	: Basit haliyle XML Web Servislerinin hayatımızdaki yeri.....	9
<b>Şekil 2.5</b>	: Web servisi mimarisi katmanları.....	10
<b>Şekil 3.1</b>	: Ariadne'nin 3 katmanlı mimarisi.....	17
<b>Şekil 4.1</b>	: Tasarlanan sistemlerin birbirleri ile etkileşimleri .....	19
<b>Şekil 4.2</b>	: Toros Üniversitesi Kullanım Durumu Diyagramı .....	20
<b>Şekil 4.3</b>	: Toros Üniversitesi Genel Tasarım .....	25
<b>Şekil 4.4</b>	: Toros Üniversitesi Veritabanı Tasarımı (Tablo İlişkileri) .....	26
<b>Şekil 4.5</b>	: Toros Üniversitesi Veritabanı Tasarımı (Saklı Yordam İlişkileri).....	27
<b>Şekil 4.6</b>	: E-Toros Veri Erişim Katmanı .....	29
<b>Şekil 4.7</b>	: E-Toros Sayfa Sınıf Diyagramı .....	30
<b>Şekil 4.8</b>	: Canik Üniversitesi Kullanım Durumu Diyagramı.....	31
<b>Şekil 4.9</b>	: Canik Üniversitesi Genel Tasarım.....	37
<b>Şekil 4.10</b>	: Canik Veritabanı Tasarımı (Tablo İlişkileri).....	37
<b>Şekil 4.11</b>	: Canik Veritabanı Tasarımı (Saklı Yordam Tablo İlişkileri) .....	38
<b>Şekil 4.12</b>	: CanikLib Veri Erişim Katmanı.....	39
<b>Şekil 4.13</b>	: CanikWS Sınıf Diyagramı .....	40
<b>Şekil 4.14</b>	: CanikLearning Sunum Katmanı Sınıf Diyagramı .....	41

## TABLO LİSTESİ

<b>Tablo 2.1</b>	: XML de kullanılmayacak karakterlerin yerine kullanılacak ifadeler ...6
<b>Tablo 4.1</b>	: Toros Veritanını oluşturan tablo açıklamaları .....26
<b>Tablo 4.2</b>	: Toros Veritanınındaki saklı yordamlar ve kullanım yerleri .....28
<b>Tablo 4.3</b>	: Canik Veritanını oluşturan tablo açıklamaları .....38

## ÖZET

### **E-ÖĞRENME PROJELERİNDE XML WEB SERVİSLERİNİN KULLANIMI VE ÖRNEK UYGULAMA**

Bu çalışmada, XML web servislerinin e-öğrenme projelerinde kullanımı incelenmiş ve örneklendirilmiştir. Çalışmayı üç bölüme ayırmak mümkündür. İlk bölümde XML web servisleri hakkında bilgi verildikten sonra web servislerin e-öğrenmede kullanımı ve örnek projeler hakkındaki bölümler yer alır.

XML hakkında genel bilgi verildikten sonra, web servislerine giriş yapılmıştır. Web servislerinin çalışma mantığı anlatılıp, protokolleri hakkında bilgi verilmiştir. Ayrıca web servislerin sunmuş olduğu avantajlardan bahsedilmiştir.

Çalışmanın ikinci bölümünde XML web servislerinin e-öğrenme projelerinde kullanımı hakkında bilgi verilmektedir. E-öğrenme hakkında genel bilgiler verildikten sonra XML web servislerini e-öğrenme projelerinde kullanan çalışmalardan ikisi tanıtılmıştır.

Üçüncü bölümde, XML web servislerinin e-öğrenme projeleri içinde kullanımı üç proje ile örneklendirilmiştir. Örnek olarak, iki farklı üniversite için e-öğrenme projesi hazırlanmış, ayrıca iki sistemin birbirleriyle iletişimini sağlayacak bir web servis uygulaması geliştirilmiştir. Projeler için detaylı ihtiyaç analizleri çıkarılmış, ihtiyaçları karşılayan veritabanı tasarımları, veri erişim katmanı için sınıf tasarımları ve sayfa tasarımları yapılmış ve bu tasarımlar detaylı bir şekilde açıklanmıştır.

En son bölümde yapılan çalışma ile ilgili bir değerlendirme bulunmaktadır.

## **SUMMARY**

### **XML WEB SERVICES IN E-LEARNING PROJECTS AND A SAMPLE PROJECT**

In this study XML web Services in e-learning is presented and sample projects are developed. The study is separated in three parts. In the first part the XML web services discusses generally, then XML web services in e-learning and sample projects are detailed.

After explaining XML, web services are presented. Detailed information about web services runtime environment and web service protocols are presented, and web services advantages are explained.

The second part of the study contains web services in e-learning projects are discussed. Before presenting of two studies about XML web services in e-learning, e-learning was discussed generally.

In the third part of study, the usage of XML web services in e-learning is exemplified by three sample projects. Two of them are web based e-learning applications for two universities. The other is a web service application that performs interoperability between them. After the requirement analysis of three projects, database design that fits the requirements, and data access layer design and finally web pages design are performed and explained in detail.

Lastly, the evaluation of the study is given.

## 1. GİRİŞ

Bugün gelinen teknolojik seviye, hayatımızdaki her şeyi, alışkanlıklarımızı ve yaşama tarzımızı etkilemektedir. İnternetin her geçen gün etkinliğini artırmasından dolayı, hizmetlerini web üzerinden sunan etkinliklerin sayısı da hızla artmaktadır. Bu yüzden, aldığımız eğitimin de, bugüne kadar ulaşılan bilgi seviyesi paralelinde çağın gereklerine daha uygun, daha etkili ve daha kolay uygulanabilen, daha verimli sonuçlar alınabilen hale getirilmesi gerekmektedir.

Başlangıçta sadece statik belgelerin kullanıcılara sunumundan ibaret olan web, daha sonraları kurumların bazı iş süreçlerini de internet üzerinde yapabilmesini sağlayan uygulamaların sunulabildiği bir platform haline gelmiştir. İnternet devriminde ikinci aşama denilen son durumda ise, işletmelerin diğer işletmelerle olan iş süreçlerini bütünleştirme gereksinimi karşısında ortaya çıkan ve halen gelişmekte olan XML (eXtensible Markup Language) web servisleridir.

XML web servisler, farklı ortamlarda geliştirilmiş, farklı işletim sistemlerinde koşan, uygulamaların birbirleri ile veri alış verişinde bulunabilmesini sağlamaktadır. Bir başka deyişle XML web servisler uygulamaların dış dünyaya açılan standart ara yüzleri olarak tanımlanabilir. Veriyi taşımak için kullanmış olduğu standartlar, onu herhangi bir platforma bağlı kalmaktan kurtarır.

Elektronik öğrenim sistemlerinin yaygınlaşıp büyümesi ile bu sistemlerin gerek başka e-öğrenme sistemleri ile gerekse diğer uygulamalar ile veri alış verişi daha çok ihtiyaç haline gelmiştir. İşte bu noktada XML web servisler, ihtiyaç haline gelen veri paylaşımı konusunda, standart bir ara yüz sunmaktadır.

Web servislerin gelişmeye başlamasından bugüne e-öğrenme projelerinde kullanımı üzerine çeşitli araştırmalar mevcuttur. Gerek öğrenim nesnelere ulaşım konusunda gerekse e-öğrenme ana işlevlerini bir takım küçük parçalara bölüp bunları web servisler ile karşılama konusunda çalışmalar yapılmaktadır.

Web servis konularında yapılan bu arařtırmalar, alıřmanın ileriki blmlerinde detaylı bir Őekilde yer alacaktır. alıřmaya konu olan rnek iin iki adet e-ğrenme uygulaması geliřtirilmiř ve bu uygulamalar arasındaki iř birliktelięi saęlayacak bir web servis uygulaması tasarlanmıřtır. Uygulamaların tasarım ařamasından geliřtirme ařamasına kadar bu alanda en yaygın olarak kullanılan teknolojiler tercih edilmiřtir.

## **2. XML WEB SERVİSLERİ**

### **2.1. XML HAKKINDA GENEL BİLGİ**

Etkin olarak kullanılmayan bilgi önemini kaybeder. Bilgiyi paylaşabilmek, yeniden kullanabilmek, sınıflandırmak, işleyebilmek, depolayabilmek kısaca yönetebilmek istenir. Ancak yazılı, sözlü ve görsel veriler (dokümanlar), aşırı miktarlarda olmaları ve birden fazla platformda kullanılmaları sebebiyle, hem aktarılma hem sorgulama hem de başka bir sisteme yerleştirilmelerinde maliyet, zaman ve işgücü kaybına yol açarlar. Öyleyse ihtiyaç olunan, her ortamda donanım ve yazılımdan bağımsız veri transferini ve yönetimini kolaylaştıran bir veri standardıdır.

XML verilerin transferi, depolanması, sorgulanması ve yönetiminde; veriye içerik değeri katması, ihtiyaç duyduğumuz sistemi yaratabilme esnekliği sunması, dağınık verilerin kümelenmesi, karşılaştırma yapma kolaylığı, farklı veri formatlarını ve dilleri destekleyebiliyor olması ve tüm sistemlerle çalışabilme özelliğiyle bugün ve gelecekte ihtiyaç duyulan veri standardıdır. Veri öbekleri XML'le bilgiye dönüşür [6].

XML, değişik veri, kavram ve içeriklerin tanımlanması ve temsil edilmesi için uygun bir ortam sunmaktadır. Bu nedenle XML farklı alanlarda uygulama verilerinin tanımlanması ve taşınması için üretici, dil ve platformdan bağımsız stratejik bir araç olarak hızla yaygınlaşmaktadır.

SGML (Standard Generalized Markup Language), ISO (International Standards Organization) tarafından 1986 yılında kabul edilmiş ve onaylanmış bilgi yönetimi standardıdır (ISO 8879:1986) [1]. SGML, platform-bağımsız ve uygulama-bağımsız dokümanlar yaratmak ve sağlamak için oluşturulmuştur. Dilbilgisi gibi bir mekanizma kullanılarak dokümanların yapısını, özel tanımlanmış etiketler kullanarak tanımlamamıza yarar [3]. SGML, bir meta dildir. Yani yeni diller üretmeye müsait bir yapıya sahiptir.

Bağımsız bir kuruluş olan W3C (World Wide Web Consortium) organizasyonu tarafından tasarlanan ve herhangi bir kurumun tekelinde bulunmayan XML, Genişletilebilir İşaretleme Dili, SGML 'den türetilmiş, basit ve çok esnek bir işaretleme dildir. Başlangıçta büyük ölçekli elektronik yayınlardaki zorluklarla başa çıkmak için tasarlanan XML'in, daha sonraları geniş ölçekli verinin, web ve diğer ortamlar arasında değiş tokuşunda önemli rol oynamasıyla popülaritesi artmıştır [4].

SGML den türetilmiş işaretleme dilleri, uygulamalar içinde işlenecek ya da farklı ortamlarda bulunsalar bile uygulamalar arasında dağıtılabilecek verilerin tanımlanmasını ve dağıtımını kolaylaştırmaları bakımından önemi büyüktür. XML çok kısa zaman içinde HTTP (Hypertext Transfer Protocol) kullanarak veri transferi konusunda evrensel bir protokol haline gelmiştir. HTML (Hyper Text Markup Language)'in dokümanları internette yayınlamak için tek bir dil olmasının yanında, uygulama geliştiriciler XML'in veri transferi ve işlemesi konusundaki gücünden faydalanmaya başlamışlardır.

### **2.1.1 XML Dokümanlarının Yapısı**

XML, karmaşık problemlere çok basit çözümler sunmaktadır. Veriyi ya da bilgiyi, kendi kendini tanımlayan doküman formatı içinde standart bir yapı sunar. XML, veri yapısını kendi içinde tanımladığından dolayı platform bağımsızdır. Ayrıca düz metin olarak tutulmasından dolayı HTTP protokolü ile bir yerden başka bir yere kolayca transfer edilebilir. Böylelikle XML, verinin transferi ve uygulamalar arasında alış verişinde diğer herhangi bir kısıtlamaya maruz kalmaz [5].

2323	Ali Korkmaz	36
1212	Veli Tok	48

Yukarıdaki satırların gerçekten hangi tür bilgiyi gösterdiğini tahmin etmek çok kolay değildir. Ancak Şekil 2.1 deki XML dosyasının içeriğine bakıldığında, dosyada ne tür bir bilgi bulunduğu hakkında daha çok fikir sahibi olunmaktadır.

```

<?xml version="1.0" encoding="iso-8859-9" ?>
<!--Sınav sonuçları... -->
<notlar>
  <ogrenci no="2323">
    <adı>Ali Korkmaz</adı>
    <notu>36</notu>
  </ogrenci>
  <ogrenci no="1212">
    <adı>Veli Tok</adı>
    <notu>48</notu>
  </ogrenci>
</notlar>

```

Şekil 2.1: Örnek bir XML dosyası

### 2.1.2 Belgenin Tanımlanması

Şekil 2.1 deki XML dosyasında da olduğu gibi, ilk satır, dosyanın XML dosyası olduğunu ve burada bulunan XML'in sürüm bilgisi ile dosyadaki metnin karakter kodlama numarası bulunmaktadır.

Çok dilli belgeleri ve evrensel kodu destekleyen XML, uygulamaların uluslararası hale getirilmesinde önemli avantaj sağlar. Satırın kodlama parçası dosya içeriği işlenirken önem kazanır. Dosyada, belirtilen karakter kodlamasında bulunmayan bir karaktere ya da karakter dizisine rastlandığında hata üretilecektir. Burada belirtilen kodlama türü, çıktının kodlama türü değil, başlığı barındıran dosyanın karakter kodlamasıdır. Türkçe belgeler için "iso-8859-9" İngilizce belgeler için "iso-8859-1" ya da "utf-8" evrensel kodları kullanılmaktadır.

### 2.1.3 Verinin Tanımlanması

XML belgelerinde, HTML de olduğu gibi önceden belirlenmiş etiketler bulunmaz, etiketler kullanıcı tarafından esnek bir şekilde belirlenir. XML belgeleri, platformlar ve uygulamalar arası standart bir veri transfer yöntemi olması ve kullanıcı tanımlı genişleyebilir, evrensel bir yapıda olması dolayısıyla, paylaşılacak verinin dilbilgisi kuralının tanımlama bilgisine ihtiyaç vardır. Böylelikle uygulamalar, hangi tipte veri bekledikleri ve göndermesi konusunda karşılıklı hemfikir olurlar.

DTD (Data Type Definition) tanımı ya da XSD (XML Schema Definition), XML çözümleyicisine gerekli etiketleri, isteğe bağlı olan etiketleri ve bir etiketin izin verilen değerlerini belirtilerek belgenin dilbilgisini söylerler. Her ne kadar XML esnek bir yapıya sahipse de, DTD de ya da şemada belirtilen kurallara sıkı sıkıya uyulması gerekir. XML belgesinin bu kurallara uymayan kısmı olursa çözümleyici bir hata döndürür. Bu kurallara uygun olarak hazırlanmış XML belgesine, iyi biçimlenmiş (well-formed) belge denir. Şekil 2.2 deki örnek DTD' ye sahip XML belgesinin, *notlar* isimli kök elementin bulunduğunu ve bu elementin *ogrenci* elementlerinden oluştuğunu, *ogrenci* elementinin ise içinde veri bulduran *adi* ve *notu* isimli iki element ve *no* isimli zorunlu bir özniteliğe sahip olduğunu anlamaktayız.

```
<!ELEMENT notlar (ogrenci+)>
<!ELEMENT ogrenci (adi, notu)>
<!ATTLIST ogrenci no CDATA #REQUIRED>
<!ELEMENT adi (#PCDATA)>
<!ELEMENT notu (#PCDATA)>
```

Şekil 2.2: Basit bir DTD örneği

#### 2.1.4 Elementler

XML belgelerinde bilgi, etiketler arasında sunulur. Şekil 2.3 de gösterildiği gibi her bir belge bir adet kök elemente sahip olmalıdır. Bu kök içinde ağaç yapısı şeklinde diğer elementler sıralanabilirler.

```
<?xml version="1.0" encoding="iso-8859-9" ?>
<!--Yorum satırı... -->
<etiket1>
  <etiket2>
    <etiket3>
      ...
    </etiket3>
  </etiket2>
</etiket1>
```

Şekil 2.3: XML dosyası element yapısı

Etiketleme de dikkat edilecek bazı kurallar vardır. Etiket isimleri daima < ve > karakterleri arasında olmalıdır. Dolayısıyla element içindeki bilgi içinde < ve > işaretleri gibi XML belgeleri için anlam ifade eden karakterler kullanılamazlar. Bunların yerine Tablo 2.1 deki gibi ifadeler kullanılmalıdır.

Tablo 2.1 : XML de kullanılmayacak karakterlerin yerine kullanılacak ifadeler

<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

Etiket ve değişken isimleri daima bir harfle başlamalıdır. SGML ve HTML için etiket ismini oluşturan harflerin büyüklüğünün önemi yoksa da XML için küçük büyük harf duyarlılığı vardır. XML' deki bu özellik temel aldığı kodlamadan kaynaklanmaktadır.

Bir etiket tanımında, içinde veri içermesi öngörülmüşse, bir açılış etiketini daima bir kapanış etiketi izlemeli. HTML ve SGML deki bazı durumlarda bitiş etiketi verilmeyebilirse de XML için bitiş etiketleri zorunludur.

Bir etiketin bitiş belirleyicisi "/" işaretidir. Bir veriyi sarmalayan etiketlerde, bu işaret, etiket isminin başına, işlem etiketlerinde ise sona konur.

### 2.1.5 Yorum Satırları

XML belgesinin içine yazılacak olan yorum metinleri <!-- --> notasyonları arasına yazılmalıdır. Örnek,

```
<!-- Bu bir yorum satırıdır -->
```

## 2.2. WEB SERVİSİ NEDİR?

Günümüzde internet üzerindeki sistemler düşünüldüğünde çok fazla sayıda donanım ve bu donanımlar üzerinde kurulu çeşitli platformlar ve platform üzerinde çalışan değişik amaçlı yazılımlar mevcuttur. Bu çeşitlilik kendi içinde çok fazla problem gibi

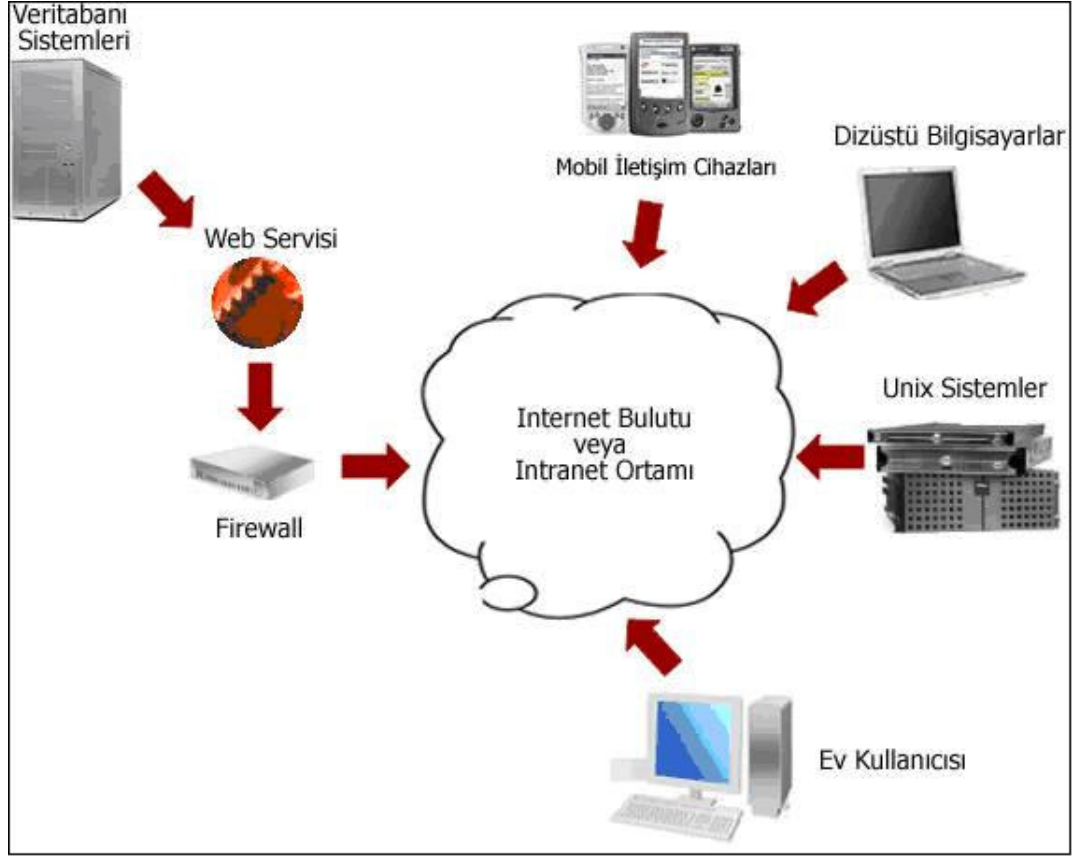
görünmese de, sistemler arasındaki iletişim söz konusu olduğunda problem haline gelebilmektedir. Bu problemi çözmek için bir yol, herkesin birbiriyle iletişim kurabilmesi için standartları oluşturmaktır. Web servisler her çeşit uygulamalar arasında iletişim kurmak için yeni bir yöntem sağlamaktadırlar.

W3C'ye göre web servis, bir ağ üzerindeki sistemlerin, birbirleri ile iletişimini sağlamak için tasarlanan yazılım sistemleridir. Web servisleri, diğer makineler tarafından işlenebilecek ara yüz tanımlarına sahiptir ve bu tanımlanmış olan ara yüzlere mesajlar geçilerek web servislerle iletişim kurulur [9].

XML web servisleri, yazılan program kodunu çalıştıran bileşenlerdir. Sıradan bileşenlerden farkı bu işlevi farklı bilgisayarlara da sunmasıdır. XML Web Servislerine farklı bir bilgisayar ve farklı bir platformdan istemci olunabilir. Yani XML web servisleri, farklı platformlar arasında ortak olarak kullanılabilen bileşenlerdir. XML web servisleri ile farklı platformlar arasında veri alış verişi mümkün kılınmaktadır. Bu farklı platformlar arası iletişimin sağlanması, web servislerin bazı standartlara uyması sayesinde olmaktadır. Bunlara örnek olarak, HTTP, XML, SOAP (Simple Object Access Protocol) standartları sayılabilir. HTTP, çoğu web tarayıcıların istemde bulunurken ve web sunucularının isteklere cevap verip web sayfalarını sunarken kullandıkları protokoldür. Bunun ışığında XML web servislerini, istemcilerden HTTP protokolü üzerinden istekleri kabul eden ve bunlara XML tabanlı cevaplar sunan sunucu üzerindeki nesnelere olarak, bir başka tanım getirebiliriz [10].

Kullanılan standartların sunmuş olduğu soyutlama sayesinde, web servislerini kullanırken, web servisinin hangi dilde yazıldığı, hangi platform üzerine sunulduğu, ya da istemci uygulamanın hangi dilde yazıldığı ve hangi platformda koştuğu bilgisi önemli değildir.

Web servisleri, merkezi uygulamalar olduklarından, verilerdeki değişiklikler bu servisleri kullanan tüm uygulamalar için de eş zamanlı ve eş güncellikte olacaktır. Web servisleri ve bu servisleri kullanan istemciler arasındaki ilişkiler, yüzeysel olarak bakıldığında şekil 2.4 de görüldüğü gibi değerlendirilebilir.

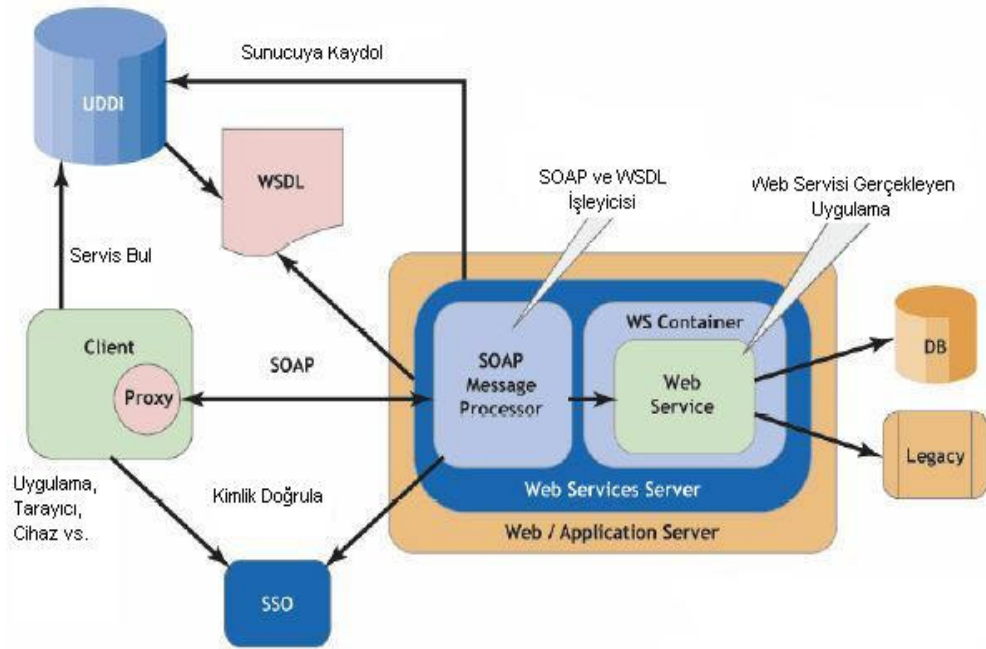


Şekil 2.4: Basit haliyle XML Web Servislerinin hayatımızdaki yeri [11].

Web servislerinin, onları kullanan istemciler ile arasındaki ilişki, şüphesiz ki bu şekilde görüldüğü kadar basit değildir. Her şeyden önce, web servislerini kullanan istemciler ile arada kurulacak ilişkinin belli standartlara dayandırılması gerekir. Her ne kadar, web servisleri HTML üzerinden gidecek XML veri parçalarını kullanıyor olsa da bunların, istemcilerin işleyebileceği ve anlayabileceği bir hale getirilmeleri gerekir. Bu noktada devreye web servisleri için gerekli protokoller devreye girmektedir.

### 2.3. WEB SERVİSİ PROTOKOLLERİ

Web servisleri açık internet standartlarına dayanır. Şekil 2.5 de web servislerinin temel protokolleri ve çalışma mantığı gösterilmektedir [19].



Şekil 2.5: Web servislerinin mimarisi katmanları

Bir web servisi kullanımında, istemci, aradığı hizmeti verebilecek sunucuları konumlandırmak için UDDI (Universal Description Discovery and Integration Protocol) kaydına başvurur. Daha sonra sunucudan bir WSDL (Web Services Description Language) dokümanı ister. Bu WSDL dokümanında, ilgili hizmetin desteklediği işlemler tanımlıdır. Servis almak için, HTTP üzerine kurulu SOAP (Simple Object Access Protocol) protokolü kullanılmaktadır.

### 2.3.1 UDDI

UDDI (Evrensel Açıklama, Keşif ve Entegrasyon), web servislerini internette tanımlamak için oluşturulmuş bir protokoldür. Hangi web servisinin nerede olduğunu ve ne işe yaradığını bildirmek için kullanılır. Kısaca UDDI, web servislerinin rehberidir de denilebilir. UDDI, web servisleri yayımlama ve keşfetme yollarını sunar. Web servisleri insanlara ve diğer üçüncü parti yazılımlara hizmet vermek için geliştirilmiş yazılımlardır. Dolayısıyla servisi kullanacak kişinin web servisin nerede olduğunu bilmesi gerekir. İşte web servisinin nerede olduğunu bulmak için, UDDI kullanılır.

UDDI İş(firma) kaydının yaptırılması anlamına gelmektedir. Bu bileşen, firmanın isminin ve bu firmaya ait web servisinin bulunduğu bir XML dosyasıdır. UDDI firma kaydı bileşenin içeriği temelde üç bölümden oluşur;

- a) Adres, iletişim ve genel bilinmesi gereken bilgilerin bulunduğu “beyaz sayfalar”,
- b) Endüstriyel sınıflandırmanın bulunduğu “sarı sayfalar”
- c) Web servisi sahibinin sunduğu, servisle ilgili teknik bilgilerin bulunduğu “yeşil sayfalar”

UDDI temelde XML şeması kullanır. Dil olarak XML seçilmesinin nedeni XML in platform bağımsız bir bilgi yapısı sunması ve kolayca okunup anlaşılmasıdır [15].

### 2.3.2 WSDL

Bir uygulamanın, bir web servisini kullanması için web servisinin nasıl çağırılacağı, parametrelerinin, protokollerin ve kodlama standartlarının belirtilmesi gerekir. WSDL, web servisini tanımlayan bir XML belgesidir [13]. Bir anlamda dağıtık programlamada kullanılan IDL'e (Interface Definition Language – Ara yüz Tanımlama Dili) benzer. Web servisinin tanımı, işlemleri, giren ve çıkan mesaj formatları, ağ ve bağlantı noktası numarası gibi bilgileri içerir.

Bir web servisi tanım belgesi aşağıdaki temel elemanları içerir:

- Types: Mesajlarda kullanılacak veri tiplerini belirtir.
- Message: İletişimde kullanılacak mesajları tanımlar.
- PortType: Web servisinin içerdiği metotları ve ilgili mesajları tanımlar.
- Binding: İşlem ve mesajlarda kullanılacak veri formatlarını tanımlar.
- Port: Ağ ve web adresinden oluşan servis noktasını tanımlar. Web adresi servisin çalıştırılacağı URL'dir.
- Service: Kullanılan bağlantı noktası numaraları kümesidir.

WSDL belgeleri, oldukça karmaşık oldukları için genellikle web servisi geliştirme araçları WSDL belgelerini otomatik olarak oluşturan programlar içermektedir [12].

### 2.3.3 SOAP

SOAP, uygulama bütünleştirilmesi için geliştirilmiş, XML ile formatlanmış bilgilerin iletişimini sağlayan, standartları W3C komitesi tarafından tanımlanan bir protokoldür [13]. Uygulama bütünleştirme için daha önce birçok çözümler sunulmuştur. RPC

(Remote Procedure Call), DCOM (Distributed component object model), IIOP (Internet Inter-ORB Protocol) ve Java RMI (Remote Method Invocation) bu çözümlerden bazılarıdır. Bu çözümler Internet ortamında iletişim sağlama konusunda yetersiz kalmaktadır. Bu nedenle XML tabanlı bir protokol olan SOAP giderek yaygınlaşmaktadır.

SOAP, istemcilerin, sunucularda olan nesne yöntemlerini çağırmasını ve sonuçların alınmasını sağlayan basit istek/yanıt protokolüdür. SOAP'ın en büyük avantajlarından birisi mevcut internet altyapısında olan yönlendirici, ateş duvarı ve vekil sunucularda herhangi bir değişiklik yapmadan kolayca çalışmasıdır.

SOAP, istemci ve sunucusu arasındaki iletişim kısaca şu şekilde gerçekleşir. İstemci SOAP uygulaması, bir SOAP istek mesajı oluşturarak bu isteği SOAP sunucusunda tanımlanmış servis uç noktalarından birisi tarafından çalıştırılması için gönderir. SOAP mesajı, HTTP POST metodu veri paketinin içinde gönderilir. SOAP mesajı, bir SOAP zarfından oluşur. SOAP zarfı, bir SOAP gövdesinden ve isteğe bağlı bir SOAP başlığından oluşur. SOAP gövdesi içinde kodlanarak gönderilen bu mesaj, web servisi tarafından çözülür, gerekli parametreler ve metod çağırımı bilgileri eşliğinde işlemlerini gerçekleştirdikten sonra, istemciye döndüreceği cevap bilgileri için, yine SOAP protokolüne uygun XML mesajlarını oluşturur. Bu mesajlar HTTP üzerinden istemci uygulamaya ulaşır, burada çözülür ve değerlendirilir [12]. HTTP protokolünü desteklediğinden, SOAP sayesinde web servisine, platformdan bağımsız çağırımlar yapılabilmektedir. Örneğin, web servislerinin farklı bileşenler üzerinde çalışacağı, .NET (Microsoft) veya J2EE (Sun) teknolojilerinden hangisinin kullanılacağı kararı, tamamen web servislerinin üzerinde geliştirildiği uygulama platformuyla ilgilidir [14].

#### **2.4. XML WEB SERVİSLERİN AVANTAJLARI**

XML web servisleri, bilişim teknolojileri alanında sistemlerin aralarında sorunsuzca konuşabilmeleri, bilgi paylaşabilmeleri için oluşturulmuş popüleritesini koruyan bir teknolojidir. Web servislerini kullanarak;

- Daha etkili yazılımlar yazmak mümkündür.
- Kurum içi farklı bölümler arasında, sorunsuzca bilgi alışverişinde bulunabilir.

- Farklı kurumlarla birlikte veri paylaşımı yapılabilir.
- İş akışını hızlandıran ve ekonomik olarak kurumları bilgi iletişimde masrafları azaltan bir teknolojidir.
- Farklı programlama dillerinde, farklı platformlarda yazılmış olan uygulamaları bir araya getirerek Internet üzerinden haberleşmelerine imkân tanır [16].

#### **2.4.1. Web Servislerinin Sundukları**

- a. İşbirliktelik: Farklı platform ve işletim sistemlerindeki farklı uygulamaların beraber çalışabilmelerini sağlaması web servislerinin en önemli avantajıdır.
- b. Entegrasyon: Web servisleri, farklı yer ve farklı yazılımlara sahip firmaların bütünleşmiş bir servis altında kolayca toplanabilmelerini sağlar.
- c. Tekrar kullanılabilirlik: Web servisleri, hizmetlerin farklı uygulamalarda kullanılabilmesini sağlar.
- d. Servis şeklinde yazılım: Yazılımların servisler şeklinde sunulmasına olanak sağlayarak farklı platformlarda rahat kullanım sağlar. Bileşen olarak kullanılabilmesi esnek bir yapı sunar.
- e. Erişilebilirlik: Servisler platform bağımsız olmaları sayesinde kişisel bilgisayar, cep bilgisayarı, cep telefonları gibi birçok istemciye hizmet verebilir.
- f. İzolasyon: Web servisleri sağlayıcılarına Internet ortamında hizmet sunma konusunda yeni bir katman oluşturarak hizmeti diğer faktörlerden izole eder ve risk faktörünü azaltır [17].
- g. Kolay kurulabilirlik: Web servisleri standart Internet teknolojileri üzerinden dağıtılır, bu nedenle dünyanın bir ucundaki ve hatta firewall kullanan sunuculara bile rahatlıkla ulaşılabilir [18].

### 3. E-ÖĞRENMEDE XML WEB SERVİSLER

#### 3.1. E-ÖĞRENME

Sözcük anlamına bakıldığında, elektronik olarak verilen eğitime e-öğrenme deniyor. Önceleri bu eğitim, CD-ROM gibi medyalar üzerinden sağlanırken artık genelde web tabanlı olarak internet veya iç ağ üzerinden sunuluyor. İnternet'in giderek yaygınlaşması ve internet teknolojilerindeki ilgili gelişmeler e-öğrenme deneyimini büyük ölçüde zenginleştirdi. Artık ses, video, animasyon ve grafikler daha kaliteli ve hızlı bir şekilde iletilebiliyor. Bu haliyle dersliklerin modern bir uzantısı olmaktan daha fazla bir anlam taşıyan e-öğrenme, bireylerin istedikleri her yerde, etkileşimli ve uygulamalı olarak öğrenmelerine olanak tanıyor [20].

E-öğrenme, son yıllarda araştırmacıların olduğu kadar uygulayıcıların da fazlaca dikkatini çekmeye başlamıştır. Kavram, içerik yazarlarının, eğitim tasarımcılarının, çoklu ortam uygulayıcılarının, öğretmenlerin, eğitmenlerin, veritabanı yöneticilerinin ve farklı alanlardan uzmanların bir öğrenci topluluğuna hizmet amaçlı bir araya gelmesi olarak tanımlanabilir [21]. Öğrenme ortamındaki insanların rollerine ilişkin genel kabulü, bu tip sistemlerden beklenen işlevsellik için de söylemek mümkündür. Tipik bir öğrenme ortamında birkaç gruptan söz edebiliriz. Yazarlar, e-öğrenim içeriğini mevcut yazım sistemini kullanarak oluşturacak öğretmen veya eğitim tasarımcıları olabilir. Yönetici kontrolündeki bir e-öğrenme sisteminin kalbinde tipik olarak bir LMS (Learning Management System), öğrenim yönetim sistemi veya LCMS (Learning Content Management System), öğrenim içerik yönetim sistemi, bulunur. LMS, öğrenci ve profil yönetimi, gelişmelerinin ve ilerlemelerinin izlenmesi, işbirliğinin kolaylaşması ve olayların bir takvime bağlanması gibi işlevsellikler sunar. LCMS ise çoğu zaman bir veritabanında tutulan öğrenim içeriğinin yönetiminden sorumludur. Buna ek olarak, LCMS, içeriğin yeniden kullanımını kolaylaştırır, içerik hazırlamada iş akışı desteği sunar ve önceden tanımlı ara yüzler ve sunum katmanları aracılığıyla içeriği taşır.

Bugün gelinen teknolojik seviye, hayatımızdaki her şeyi, alışkanlıklarımızı ve yaşama tarzımızı etkilemektedir. İnternetin her geçen gün etkinliğini artırmasından dolayı, hizmetlerini web üzerinden sunan etkinliklerin sayısı da hızla artmaktadır. Bu yüzden, aldığımız eğitimin de, bugüne kadar ulaşılan bilgi seviyesi paralelinde çağın gereklerine daha uygun, daha etkili ve daha kolay uygulanabilen, daha verimli sonuçlar alınabilen hale getirilmesi gerekmektedir.

Web servisleri, farklı e-öğrenme uygulamalarını bir arada yönetebilmenin zorluğunu hafifletmek ve birbirleriyle etkileşimi mümkün kılarak daha yüksek bir işbirliklilik seviyesini yakalamak için öğrenim teknolojileri alanına girmeye başlamış, bu konuda tamamlanmış ve halen yapılmakta olan pek çok proje imza atılmıştır [23].

## **3.2. E-ÖĞRENME DE WEB SERVİS MİMARİLERİ**

### **3.2.1. Geleneksel Mimari**

Mevcut geliştirilmiş olan sistemler, çoklukla istemci-sunucu veya P2P (eşler arası) mimarilerine dayalı sistemlerdir. Bu yüzden ölçeklenebilirlik problemleri ve karmaşık içerik alışverişleri, karşılaşılan önemli problemlerdendir. IEEE LOM (Learning Object Metadata) ve ADL (Advanced Distributed Learning), SCORM (Shareable Content Object Reference Model) gibi standartlara rağmen, eğitimsel içeriğin sunucular ve eşler arasında alışverişi halen tatmin edici şekilde çözülememiş bir problemdir. Diğer yaklaşımlar bu problemi tasarım kalıpları gibi mimari tasarıma odaklanarak veya uyumlu ek programlar kullanımına izin veren çerçeve sistemleriyle aşmaya çalışırlar. Farklı sistemlerin birbirleriyle olan bağlantılarını bulmak çoğu zaman eksik ve kötü dokümantasyon yüzünden kolay olmadığından bazı zorluklar halen devam etmektedir [22].

### **3.2.2. Web Servis ile Mimari**

E-öğrenme platformları ve işlevsellikleri büyük ölçüde birbirine benzer. E-öğrenme alanındaki standardizasyon çabaları öğrenim materyalinin yeniden kullanımı üzerinde yoğunlaşmış, ancak uygulama işlevselliklerinin yeniden kullanımı ile ilgili yeterince çalışılmamıştır. Geleneksel e-öğrenme platformlarının aksine, Muenster

Üniversitesi'nce geliştirilen LearnServe sistemi [21], tipik bir öğrenim sisteminin, öğrenciler ve öğrenim nesnelere şeklinde gelen uygun içerikle etkileşim halindeki aktivite ve süreç koleksiyonu olduğu varsayımı üzerine kurulmuştur. Bu, bir e-öğrenme sisteminin ana fonksiyonlarını, kendi başına çalışacak birtakım uygulamalara bölebilmemizi sağlar ki, bunlar da tek tek veya grup olarak web servisleri şeklinde gerçekleştirilebilir. Öte yandan, web servisleri, bir takım sağlayıcılar tarafından sunulur ve ders ünitelerini veya bütününe içerebilir.

Bu servislerin gerçekleştirilmesi, e-öğrenme platform işlevselliğinin yeniden kullanımını mümkün kılar. LearnServe sistemi, hem e-öğrenme hem de web servisleri alanında ortak standartlara dayalı olarak geliştirilmiştir. Dağıtık tarzda tasarımı, içerik ve servis yönetiminde bir takım zorluklar getirmiş de olsa, e-öğrenme servislerinin iş uygulamalarına direkt entegrasyonu veya o cihaz için uygun istemci olduğunda farklı cihazların öğrenim servislerine erişimi gibi ciddi potansiyelleri beraberinde getirmiştir.

Pankratius ve diğ [22] tarafından öne sürülen tasarımda, LCMS içinde, öğrenimle alakalı bütün aktivitelerin ve öğrenim materyallerinin yönetiminin koordinasyonu hedeflenmiştir. Öğrencinin kullandığı tarayıcı, oturum esnasında, direkt olarak LCMS ile etkileşir. LCMS'nin bütün web servisleri, kullanıcı sadece tarayıcısının kullanarak işini görebilsin diye, web sayfalarından erişilebilir olmalıdır.

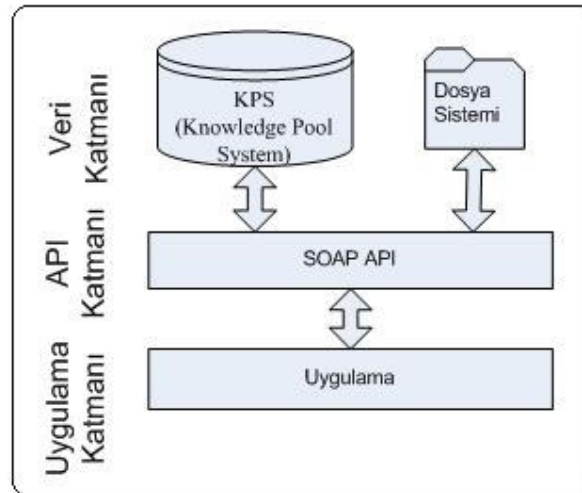
İlk etapta LCMS öğrencinin kimliğini doğrular, bu işlem LCMS giriş servisi tarafından gerçekleştirilir. Öğrenci sisteme girip kimliğini doğrulattıktan sonra, ders yönetimi için web sayfasına girebilir. Bu, ders yönetim servisinin gerçekleştirilmesi gereken bir işlevselliştir. Öğrenci, uygun dersleri yine bir web servisi olarak gerçekleştirilmiş ajan platformunun yardımıyla arayabilir, bulunan öğrenim nesnelere yine ajan platformunun yardımıyla kullanılabilir.

LCMS, başka servisler de içerebilir. Örneğin, tartışma gruplarında ve sohbet odalarında, öğrenciler, öğretmenler ve diğer öğrencilerle etkileşim kurar ve soru sorabilirler. Gelişim ve ilerleme monitörü öğrencinin katıldığı bütün derslerin değerlendirme sonuçlarının genel bir özetini verebilir. Sertifikaların hazırlanmasında da bu bileşen kullanılır. Muhasebe servisi finansal yönü olan tüm işlemlerin yönetimiyle ilgilidir. Kaydolunan

dersleri veya öğrencinin ödemesi gereken tutarı gösterir. LCMS'nin işlevselliği iç ağa veya web üzerindeki diğer sağlayıcılar aracılığı ile harici olarak genişletilebilir. Bu bakış açısıyla, bir LCMS, artık tek sunucu üzerinde çalışan bir sistem değil, dünyanın değişik bölgelerinden farklı gruplar tarafından sunulabilecek dağıtık servislerin bir koleksiyonu haline gelmektedir. Bu özellik sayesinde dış kaynak kullanımını da kolaylaştırmaktadır. Örneğin bir şirket, muhasebe ve faturalamada uzmanlaşmış ve bu servisleri birçok LCMS'ye entegre etmiş olabilir. Böyle bir mimarinin beklenen diğer avantajları, daha kolay ölçeklenebilirliği, bakımı ve yeniden konfigürasyonun yapılabilmesidir.

Web servislerinin e-öğrenme üzerinde kullanımı ile ilgili bir başka çalışma, Ariadne [24] projesidir. Vuorikari [23], web servislerinin işletmeler arası iş birlikteliği sağlamanın en etkili yollarından birisi olduğunu ve bu amaçla Ariadne Bilgi Havuzu Sistemi'nin bu temeller üzerinde geliştiğini belirtmiştir.

Ariadne, LOR (Learning Objects Repository), öğrenim nesnesi ambarlarına ulaşım için web servisine dayalı bir mimari sunmaktadır. Uygulama LOR'un öğrenim nesnesi üretip tüketen diğer uygulamalara entegrasyonu sayesinde indeksleme sürecini kolaylaştıracak ve diğer LOR'larla iş birlikteliği mümkün kılacak şekilde tasarlanmıştır.



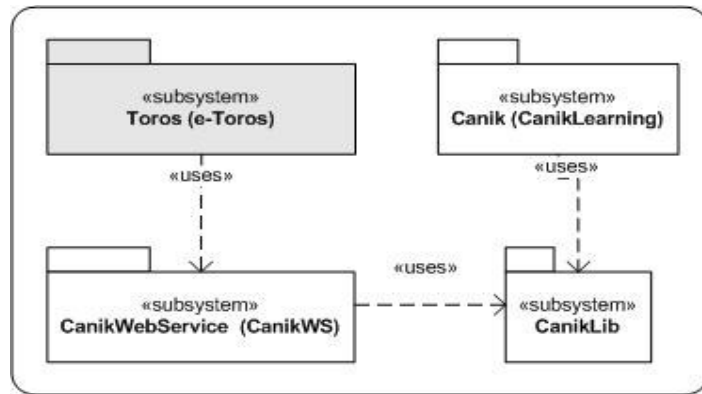
Şekil 3.1: Ariadne'nin 3 katmanlı mimarisini

Ariadne, Şekil 3.1 de görüldüğü gibi, üç katmanlı mimari ile uygulama katmanından veri katmanını gizlemeyi amaçlar ve veri yapısını ve veriyi nasıl erişileceği konusunda bir endişe duyulmadan gizler. API için iletişim protokolü olarak SOAP tercih edilmiştir. SOAP, herhangi bir programlama diline veya platforma bağlı değildir. Bunun bir alternatifi CORBA (Common Object Request Broker Architecture) gibi olgun teknolojiler kullanmak olabilirdi ancak SOAP'ın ateş duvarından etkilenmemesi de CORBA'ya kıyasla artı bir özelliğidir. Uygulama katmanı, bilgi havuzunu, bu havuzu kullanan uygulamalara taşır. Bilgi havuzuyla bir API (Application Programming Interface) üzerinden konuştukları için bu uygulamalar veritabanı mimarisinden bağımsızdırlar.

#### 4. ÖRNEK E-ÖĞRENME – XML WEB SERVİS UYGULAMASI

Web servislerinin en önemli işlevi farklı iki sistem arasındaki iş birlikteliği sağlamak olduğundan, örnek uygulama olarak birbirinden bağımsız iki farklı e-öğrenme sistemi tasarlanmıştır. Bunlardan birincisi teknik içerikli derslerin, diğeri ise sözel içerikli derslerin eğitimi veren üniversiteler olarak düşünülmüştür. Teknik üniversite, öğrencilerine içeriğini kendi hazırladığı teknik derslerin yanında, içeriğini, sözel ders eğitimi veren diğeri üniversiteden aldığı dersleri de sunmak ister. Hazırlanan web servisi burada devreye girerek, sözel ders içeriklerinin bir üniversiteden diğeri üniversiteye aktarımını yapar.

Sözel içerik sağlayan ve bunu web servisleri aracılığı ile dışarıya veren üniversiteye Canik Üniversitesi, öğrencilerine teknik dersler sunan üniversiteye de Toros isimleri verilmiştir. Bu iki farklı üniversiteler için, sırasıyla *CanikLearning* ve *E-Toros* adında iki adet web tabanlı e-öğrenme uygulamaları tasarlanmıştır. Daha sonra, CanikLearning sisteminde bulunan ders içerik bilgilerinden diğeri dış sistemlerin de yararlanmasını sağlayacak *CanikWS* isimli bir web servis geliştirilmiş ve iki sistem arasındaki iletişim bu web servis aracılığı ile sağlanmıştır. Sistemler arasındaki ilişkinin gösterildiği Şekil 4.1 deki gibi, CanikLib ise, Canik üniversitesi için geliştirilen CanikLearning ve CanikWS alt sistemlerinin ortak kullanacağı modüllerin bulunduğu kitaplık sistemidir.



Şekil 4.1: Tasarlanan sistemlerin birbirleri ile etkileşimleri

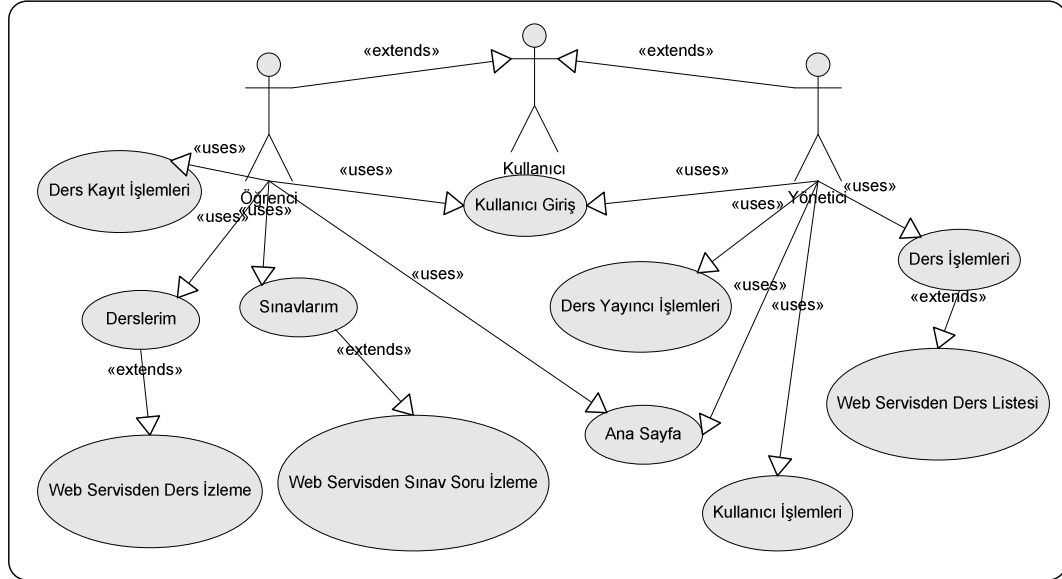
## 4.1. TOROS ÜNİVERSİTESİ

E-Toros, Toros Üniversitesi için tasarlanan bir e-öğrenme projesidir. Öğrencilerine, içeriğini kendisinin sunduğu derslerin yanı sıra, web servis aracılığı ile hizmet aldığı Çanık Üniversitesinin derslerini de uzaktan izleme imkânı vermektedir.

### 4.1.1. Analiz

E-Toros sisteminde iki çeşit aktör bulunmaktadır. Sistem yöneticisi aktörü, sistemdeki genel tanımlama işlemlerini yürüten kişidir. Ders yayıncılarını tanımlama ve kullanıcıları tanımlama bunlara örnek olarak verilebilir.

Öğrenci aktörü ise, sistemden asıl öğrenme hizmeti alacak olan aktördür. Sistemdeki derslere kayıt olmak ve kaydolduğu dersleri izlemek öğrenci aktörünün genel yaptığı işlemler arasında sayılabilir. E-Toros sistemini kullanacak aktörler ve bunların sistemdeki kullanım durumları Şekil 4.2 de gösterilmiştir.



Şekil 4.2: E-Toros Kullanım Durumu Diyagramı

#### 4.1.1.1. Kullanıcı Giriş

Toros elektronik öğrenme sistemini kullanacak sistem yöneticisi ve öğrencilerin kimlik doğrulamalarının yapıldığı arayüzdür.

İşlev:

- Kullanıcı kodu ve kullanıcı şifresi girilir

- Kullanıcı kodu ve şifresi sitemde tanımlı ise ana sayfaya yönlendirilir
- Kullanıcı kodu ve şifresi sistemde tanımlı değil ise hata mesajı döndürülür

#### 4.1.1.2. Ana Sayfa

Kullanıcı doğrulaması yapıldıktan sonra kullanıcının önüne gelen ilk ara yüzdür

İşlev:

- Ana sayfada kullanıcı tipine göre menü hakkı ayarlaması yapılır
  - Kullanıcı tipi sistem yöneticisi ise “Kullanıcı İşlemleri, Yayıncı İşlemleri, Ders İşlemleri” menülerine hak verilir
  - Kullanıcı tipi öğrenci ise “Derslerim, Konularım, Sınavlarım, Ders Kaydı” menülerine hak verilir

#### 4.1.1.3. Kullanıcı İşlemleri

Sistem yöneticisi hakkına sahip kişilerin kullanacağı, sisteme yeni kullanıcıların eklenebileceği, mevcutların bilgilerinin değiştirilip silinebileceği ara yüzdür.

Tanımlanacak bilgiler:

- Kullanıcı tipi, listeden seçmeli
- Adı, düz metin
- Soyadı, düz metin
- Kullanıcı kodu, düz metin
- Şifre, düz metin
- Öğrenci numarası, pozitif tam sayı

İşlev:

- Sistemde tanımlı kullanıcılar bir liste halinde sıralanır
- Listede kullanıcı tipi, kullanıcı adı, kullanıcı soyadı, kullanıcı kodu, öğrenci numarasına göre sıralama imkanı sunulur
- Sistem yöneticisi, sisteme yeni kullanıcı ekleyebilir
  - Tanımlanacak kullanıcının kullanıcı tipi, Öğrenci, Sistem Yöneticisi ya da Öğretmen olabilir.
  - Yeni bir kullanıcı eklenirken, eğer kullanıcı tipi öğrenci seçilmiş ise kullanıcının hangi öğrenciye ait olduğu bilgisinin anlaşılabilmesi için, bağlı bulunan öğrencinin numarası sisteme tanıtılır.

- Sistem yöneticisinin, sistemdeki mevcut kullanıcıların kullanıcı adı, soyadı, kullanıcı kodu, parola ve varsa öğrenci numaralarının değiştirilip güncellenmesine imkan verilir.
- Sistem yöneticisinin, sistemdeki mevcut kullanıcıları silmesine imkan verilir
  - Silinecek olan kullanıcıya bağlı öğrenci varsa, öğrencinin silinmeden önce, sistemde öğrenciye bağlı ders kaydı gibi başka kaydın olmaması gereklidir.

#### 4.1.1.4. Ders Yayıncı İşlemleri

Sistemde ders yayınlamak olan kurumların listelendiği ara yüzdür.

İşlev:

- Yayıncıların adları, açıklamaları, varsa sundukları web servis adları ve bu web servise bağlanırken kullanılacak kullanıcı kodu ve şifreleri bir liste halinde gösterilir.
- Listedeki yayıncı adı, web servis adı, web servis kullanıcı adı, sütunlarına göre sıralama yapabilme imkânı sunulur.
- Sistem yöneticisi, yayıncı kuruluşun web servis aracılığı ile ders içeriği sunması halinde, web servise bağlanmak için gereken web servis kullanıcı kodu ve kullanıcı şifresi bilgileri girip değiştirebilir.

#### 4.1.1.5. Ders İşlemleri

Sistemde öğrencilerin kayıt olup içeriklerini takip edebileceği derslerin eklendiği, güncellendiği ve silinebildiği ara yüzdür.

Tanımlanacak bilgiler:

- Dersi yayınlayan kurum, listeden seçmeli (Toros Üniversitesi, Canik Üniversitesi)
- Ders adı, düz metin
- Ders kodu, düz metin

İşlev:

- Sistemde tanımlı dersler bir liste halinde sıralanır.
- Listedeki dersleri, ders adı, ders kodu, yayıncı kurum alanlarına göre sıralama imkanı sunulur.
- Sistem yöneticisi, sisteme yeni dersler ekleyebilir. Dersler iki türlü olabilir. Birincisi, Kurumun (Toros Üniversitesi) kendi vermiş olduğu dersler, ikincisi

ise, kurumun web servisler aracılığı ile dışarıdaki başka kurumlardan alacağı derslerdir.

- Eğer dersi yayınlayan kurum olarak Toros Üniversitesi seçilmiş ise, ders kodu ve ders adı girilir.
- Eğer dersi yayınlayan kurum olarak web servis aracılığı ile hizmet veren dış bir kurum seçilmiş ise ders kodu ve adı hizmet alınan web servis aracılığı ile alınarak kullanıcının seçmesi için listelenir. Kullanıcı ders adını ve kodunu değiştiremez sadece seçebilir.
- Sistem yöneticisi, sistemden dersleri silebilir.
  - Eğer silinecek ders hiçbir öğrenci tarafından kayıt olunmamışsa silmeye izin verilir.
- Sistem yöneticisi, sistemdeki derslerin ismi ve kodunu değiştirebilir.
  - Eğer ders, web servis aracılığı ile dışarıdan alınıyorsa dersin adı ve kodu değiştirilemez.

#### 4.1.1.6. Derse Kayıt

Öğrencinin, kaydolup takip ettiği ders listesini düzenlemesini sağlayan ara yüzdür.

İşlev:

- Öğrenci, kayıt olduğu dersleri listeliyebilir.
- Listedeki dersleri, ders adı ve ders kodu, ve derse kayıt olma tarihine göre sıraya dizebilir.
- Öğrenci, sistemde mevcut, kaydolunmamış yeni bir derse kaydolup listeye ekleyebilir.
- Öğrenci, kaydolunmuş dersleri silip, listeden çıkarabilir.

#### 4.1.1.7. Derslerim

Öğrencinin, kaydolup takip ettiği ders listesini görüntülemesini ve seçim yapmasını sağlayan ara yüzdür.

İşlev:

- Öğrenci, kayıt olduğu dersleri listeliyebilir.
- Listedeki dersleri, ders adı ve ders kodu, ve derse kayıt olma tarihine göre sıraya dizebilir.
- Konularını ya da sınavlarını görmek istediği dersi listeden seçerek aktif hale getirir.

#### 4.1.1.7. Konularım

İşlev:

- Öğrencinin, derslerim menüsünde seçmiş olduğu aktif dersin konularını listeleyen ara yüzdür
  - Eğer seçilmiş olan aktif bir ders yoksa hata mesajı döndürülür.
- Seçili olan dersin sağlayıcısı üniversitenin kendisi ise, ders konuları üniversitenin kendi veritabanından getirilir.
- Seçili olan dersin sağlayıcısı dışarıdan web servisle hizmet veren bir kurum ise, ders konuları, dersin sağlayıcısının sunmuş olduğu web servis aracılığı ile yine bu sağlayıcı tarafından belirlenen kullanıcı adı ve parola kullanılarak getirilir.
- Listedeki konuları, konu adı, konu açıklaması ve içerik sayısına göre sıralama imkanı verilir.
- Öğrenci içeriğini görmek istediği konuyu listeden seçip aktif hale getirerek konunun içeriğini ayrı bir tabloda ya da sayfada görebilir.
- Aktif dersin sağlayıcısı üniversitenin kendisi ise, seçili olan konunun içeriği, üniversitenin kendi veritabanından getirilir.
- Aktif dersin sağlayıcısı dışarıdan web servisle hizmet veren bir kurum ise, seçili olan konunun içeriği, dersin sağlayıcısının sunmuş olduğu web servis aracılığı ile yine bu sağlayıcı tarafından belirlenen kullanıcı adı ve parola kullanılarak getirilir.
- Konu içerikleri her sayfada bir içerik gösterilecek şekilde sırasıyla gösterilir.
- İçerik sayfasında, içeriğin adı ve detayı gösterilir.

#### 4.1.1.8. Sınavlarım

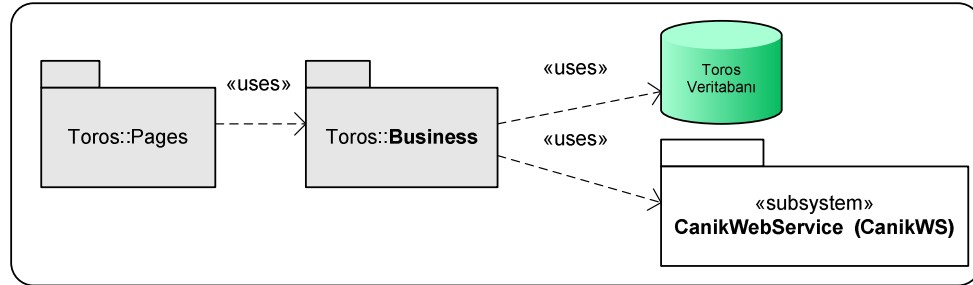
İşlev:

- Öğrencinin, derslerim menüsünde seçmiş olduğu aktif dersin deneme sınavlarını listeleyen ara yüzdür
  - Eğer seçilmiş olan aktif bir ders yoksa hata mesajı döndürülür.
- Seçili olan dersin sağlayıcısı üniversitenin kendisi ise, dersin sınav bilgileri üniversitenin kendi veritabanından getirilir.
- Seçili olan dersin sağlayıcısı dışarıdan web servisle hizmet veren bir kurum ise, dersin sınav bilgileri, dersin sağlayıcısının sunmuş olduğu web servis aracılığı ile yine bu sağlayıcı tarafından belirlenen kullanıcı adı ve parola kullanılarak getirilir.

- Listedeki sınavları, sınav adı, sınav açıklaması, sınav süresi ve soru sayısına göre sıralama imkanı verilir.
- Öğrenci, sınavı listeden seçip aktif hale getirerek sınavın soru detayları ayrı bir tabloda ya da sayfada görebilir.
- Aktif dersin sağlayıcısı üniversitenin kendisi ise, seçili olan sınavın soruları, üniversitenin kendi veritabanından getirilir.
- Aktif dersin sağlayıcısı dışarıdan web servisle hizmet veren bir kurum ise, seçili olan sınavın soruları, dersin sağlayıcısının sunmuş olduğu web servis aracılığı ile yine bu sağlayıcı tarafından belirlenen kullanıcı adı ve parola kullanılarak getirilir.
- Sınav soruları her sayfada bir içerik gösterilecek şekilde sırasıyla gösterilir.
- Soru sayfasında, sorunun sıra numarası, sorunun metni ve sorunun cevap seçenekleri gösterilir.
- Öğrencinin soruya cevap verme imkanı verilir.

#### 4.1.2. Tasarım

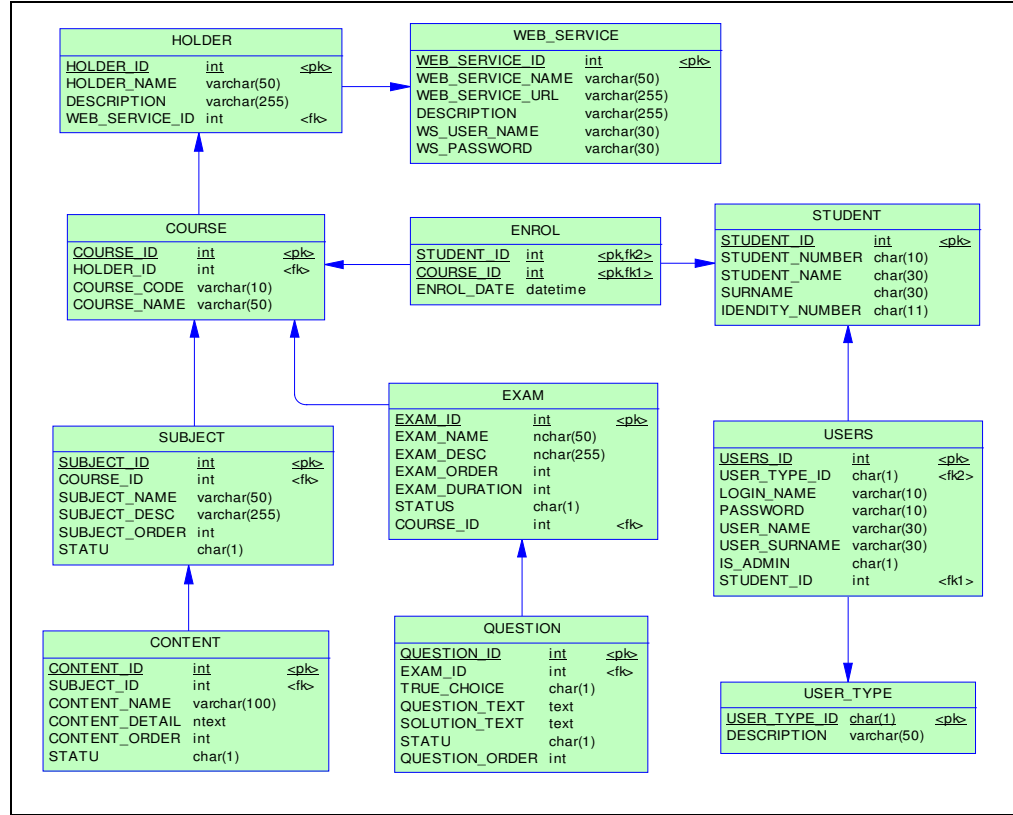
Toros Üniversitesi elektronik öğrenme ihtiyaç analizini karşılayacak sistem tasarımında, kullanıcı arayüzü ile veriler arasında yeni bir katman konulmuştur. Bu sayede kullanıcı arayüzü, arka plandaki verilerin kendi veritabanından mı yoksa web servis aracılığı ile bir başka sistemden mi geldiğiyle ilgilenmez. Bir başka deyişle veri katmanını kullanıcı arayüzünden soyutlanmış olmaktadır.



Şekil 4.3 Toros Üniversitesi Genel Tasarım

#### 4.1.2.1. Veritabanı Tasarımı

Toros Üniversitesinin kullanıcılarına hizmet verecek olan web uygulamasının verilerinin tutulacağı veritabanı ilişkisel olarak tasarlanmıştır. Her bir tablonun kimlik alanı mevcuttur ve diğer tablolar ile arasındaki ilişkiler bu alanlar üzerinden sağlanır.



Şekil 4.4: Toros Üniversitesi Veritabanı Tasarımı (Tablo İlişkileri)

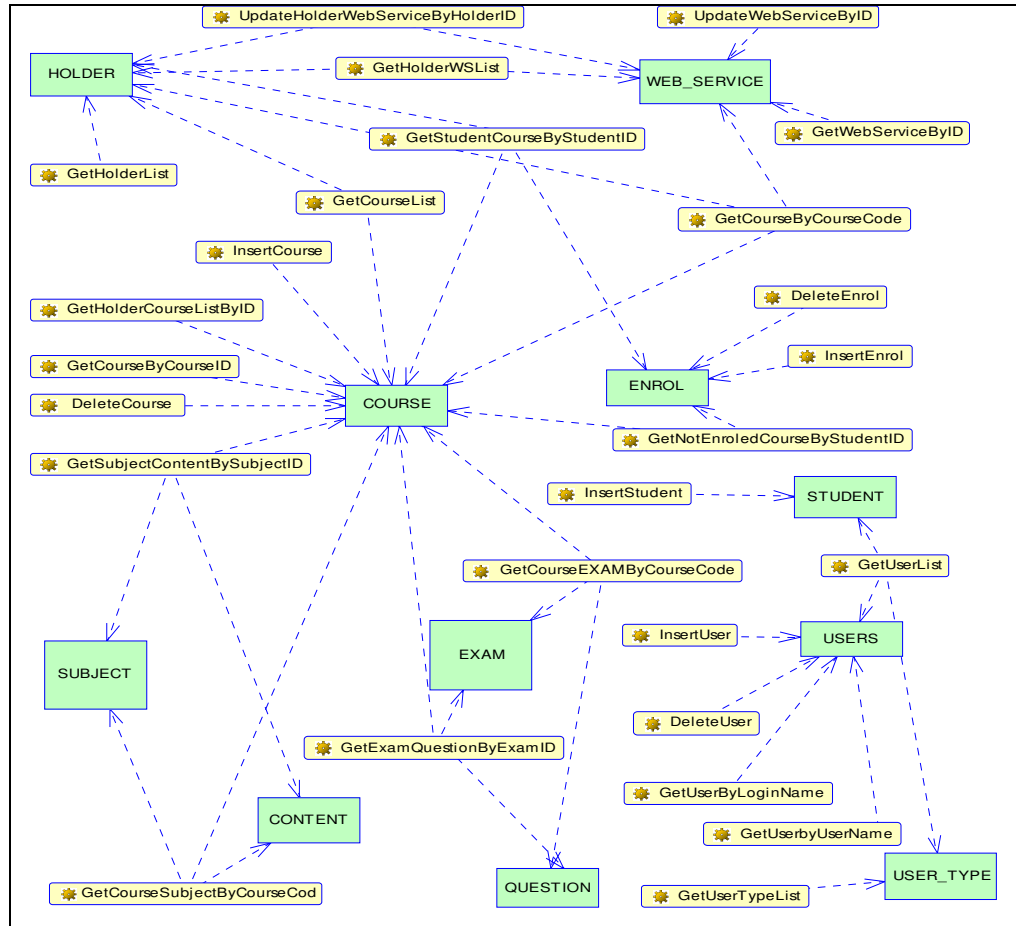
E-Toros veritabanının genel tasarımı ve tablolar arasındaki ilişkiler Şekil 4.4 de gösterilmiştir. Buna göre veritabanında yer alan tablolar ve bu tabloların kullanım amaçları aşağıdaki Tablo 4.1 de verilmiştir.

Tablo 4.1 : E-Toros Veritabanını oluşturan tablo açıklamaları

Tablo Adı	Açıklama
HOLDER	Derslerin içerik yayını yapan kurumların bilgisinin tutulduğu tablodur
WEB_SERVICE	İçerik yayını yapan kurumların varsa web servis bilgilerinin ve bu web servislere bağlanabilmek için gerekli kullanıcı adı ve şifre bilgilerinin tutulduğu tablodur.
USERS	E-Toros uygulamasını kullanabilecek kullanıcıların bilgisinin tutulduğu tablodur
USERS_TYPE	E-Toros sistemini kullanacak kullanıcıların yetkilendirilmelerini sağlayacak,

	kullanıcı tip bilgilerinin tutulduğu tablodur
COURSE	E-Toros bünyesinde öğrencilerin takibi için sunulan derslerin bilgilerinin tutulduğu tablodur
STUDENT	E-Toros bünyesindeki öğrencilerin bilgilerinin tutulduğu tablodur.
ENROL	Öğrencilerin takip etmek istedikleri derslerin listesinin tutulduğu tablodur
SUBJECT	İçeriği E-Toros tarafından sağlanan derslerin, konularının listelendiği tablodur
CONTENT	İçerik bilgileri E-Toros tarafından sağlanan derslerin konularına bağlı içeriklerin detay bilgilerinin tutulduğu tablodur
EXAM	Derslere bağlı test sınavların başlık bilgilerinin tutulduğu tablodur
QUESTION	Test sınavlarının soru detaylarının tutulduğu tablodur

Veritabanı üzerindeki tablolara erişim, yine veritabanı üzerindeki saklı yordamlar aracılığı ile sağlanmıştır.



Şekil 4.5: Toros Üniversitesi Veritabanı Tasarımı (Saklı Yordam İlişkileri)

Genel olarak her bir tabloya yeni kayıt eklemek, mevcut kayıtları güncellemek veya silmek ve tablolar arası ilişkilerden de yararlanarak ihtiyaç doğrultusunda sorgulara cevap verebilmek amaçlı geliştirilen yordamlar ve bu yordamların hangi tablolarla ilişki içinde olduğu Şekil 4.5 de gösterilmiştir. Böylelikle, saklı yordamların uygulamalara sağlamış olduğu hız, güvenlik ve soyutlama avantajlarından yararlanılmış olmaktadır. Tablo 4.2 de ise bu yordamların işlevleri sunulmuştur.

Tablo 4.2 : Toros Üniversitesi Veritabanındaki saklı yordamlar ve kullanım yerleri

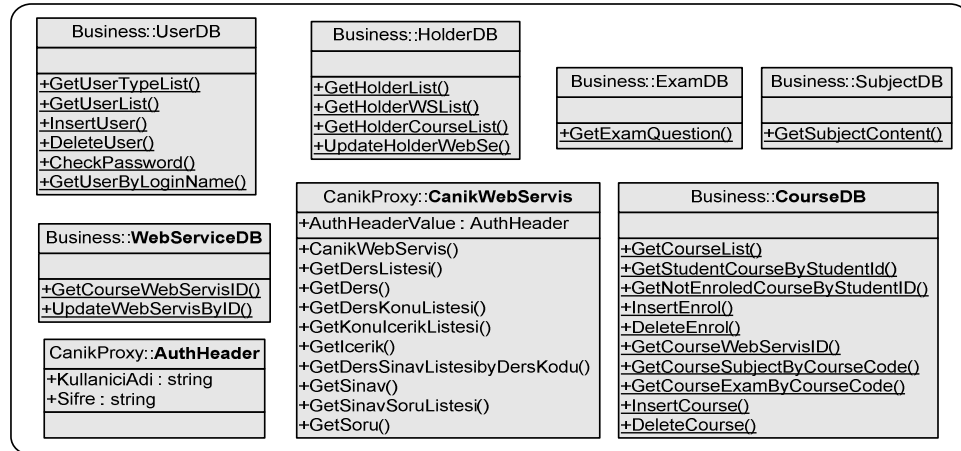
Tablo Adı	Açıklama
DeleteCourse	Kimliği verilen ders kaydını siler
DeleteEnrol	Kimliği verilen öğrenci ders kaydını siler
DeleteUser	Kimliği verilen kullanıcı kaydını siler
GetCourseByCourseCode	Ders kodu verilen dersin bilgilerini döner
GetCourseByCourseID	Kimliği verilen dersin bilgilerini döner
GetCourseExamByCourseCode	Ders kodu verilen dersin sınav bilgilerini liste halinde döner
GetCourseList	Kimliği verilen yayıncının vermiş olduğu ders listesini döner
GetCourseSubjectByCourseCode	Ders kodu verilen dersin konu bilgilerini liste halinde döner
GetCourseSubjectByCourseID	Kimliği verilen dersin konu bilgilerini liste halinde döner
GetExamQuestionByExamID	Kimliği verilen sınavın konu soru bilgilerini liste halinde döner
GetHolderCourseListByID	Kimliği verilen yayıncıya ait ders listesini döner
GetHolderList	Veritabanındaki yayıncı listesini döner
GetHolderWSList	Veritabanındaki yayıncı web servis listesini döner
GetNotEnroledCourseByStudentID	Kimliği verilen öğrencinin kayıt olmadığı derslerin listesini döner
GetStudentCourseByStudentID	Kimliği verilen öğrencinin kayıt yaptırdığı derslerin listesini döner
GetSubjectContentBySubjectID	Kimliği verilen ders konusunun içerik listesini döner
GetUserByLoginName	Kullanıcı kodu verilen kullanıcının bilgilerini döndürür
GetUserbyUserName	Kullanıcı adı verilen kullanıcının bilgilerini döndürür
GetUserList	Veritabanındaki kullanıcı listesini döner

GetUserTypeList	Veritabanındaki kullanıcı tipi listesini döner
GetWebServiceByID	Kimliği verilen web servisin bilgilerini döner
InsertCourse	Ders bilgileri verilen dersi veritabanına kaydeder
InsertEnrol	Öğrenciyi derse kaydeder
InsertStudent	Bilgileri verilen öğrenciyi veritabanına kaydeder
InsertUser	Bilgileri verilen kullanıcıyı veritabanına kaydeder
UpdateHolderWebServiceByHolderID	Kimliği verilen yayıncının bilgilerini günceller
UpdateWebServiceByID	Kimliği verilen web servisin bilgilerini günceller

#### 4.1.2.2. Veri Erişim Katmanı Tasarımı

Veri erişim katmanının görevi, ön yüz tarafından istenilen verinin kaynağından bulunup getirilmesini sağlamaktır. Böylelikle ara yüz ile ver kaynağı birbirlerinden soyutlanmış olur. E-Toros'da ön yüzde kullanıcının istediği bir bilgi, ön yüz tarafında veri erişim katmanına iletilir. Veri iletişim katmanı, isteğin cinsine göre veriyi ya Toros veritabanından ya da ilgili web servisinden bulup getirerek arayüzün kullanımına sunar. Sistem dış sistemlerden web servisler aracılığı ile hizmet alıyorsa, hizmet aldığı her bir web servis için vekil bir sınıf türetmek zorundadır. Web servisinden bilgiler bu vekil sınıflar aracılığı ile alınır ve ön yüze geçilir.

Şekil 4.6 da E-Toros için tasarlanan veri erişim katmanı yer almaktadır. Kullanıcı bilgilerine erişim için UserDB, ders yayıncı bilgilerine erişim için HolderDB, ders bilgilerine ulaşmak için CourseDB, ders sınav bilgilerine ulaşmak için ExamDB, ders konu içerik bilgilerine ulaşmak için SubjectDB sınıfları kullanılmaktadır.

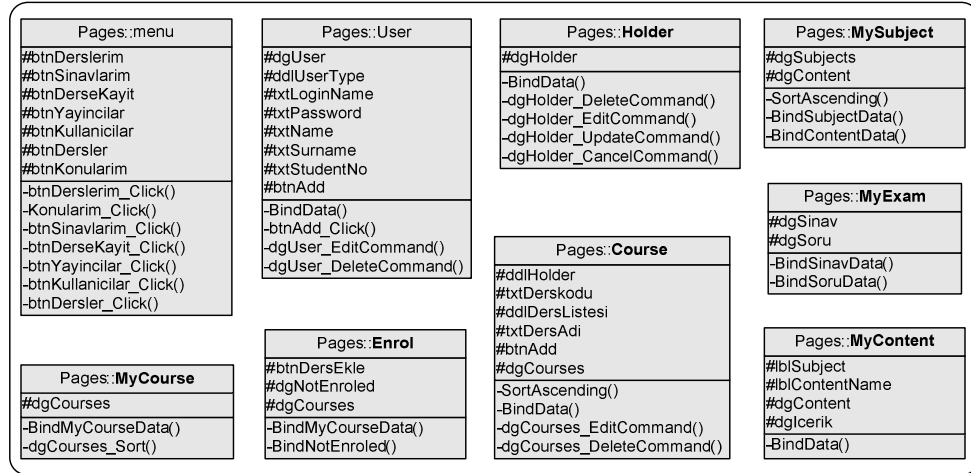


Şekil 4.6: E-Toros Veri Erişim Katmanı

CourseDB dışında bahsi geçen her bir sınıf, kendisinden istenilen verileri Toros veritabanından getirmektedir. CourseDB ise, dersin durumuna veritabanından bakarak hangi sistemden getireceğine karar verir. Eğer ders Toros Üniversitesinin kendi dersi ise o da diğerleri gibi veriyi Toros veritabanından çekerek isteği cevaplar. Ancak ders, web servis ile hizmet alınan dış bir sistemin dersi ise, dersin ait olduğu yayıncının web servis bilgileri, WebServisDB aracılığı ile sorgulanır. Bu bilgiler ışığında ilgili web servisin vekil sınıfından bir nesne türetilerek web servis kullanılarak ders bilgileri (ders içerik bilgisi ya da ders sınav bilgisi) alınarak isteğin geldiği ön yüze yollanarak veri erişim katmanının işlevi sonlanmış olur.

#### 4.1.2.3. Sayfa Tasarımı

E-Toros için geliştirilen sayfaların arka planındaki sınıf tasarımı şekil 4.7 de gösterilmiştir. Buna göre E-Toros bünyesinde yayıncı işlemleri, kullanıcı işlemleri, ders işlemleri, öğrenci ders kayıt işlemleri, ders konu içeriği ve sınav sorularını göstermeye yönelik sayfalar bulunmaktadır.



Şekil 4.7: E-Toros Sunum Katmanı Sınıf Diyagramı

## 4.2 CANIK ÜNİVERSİTESİ

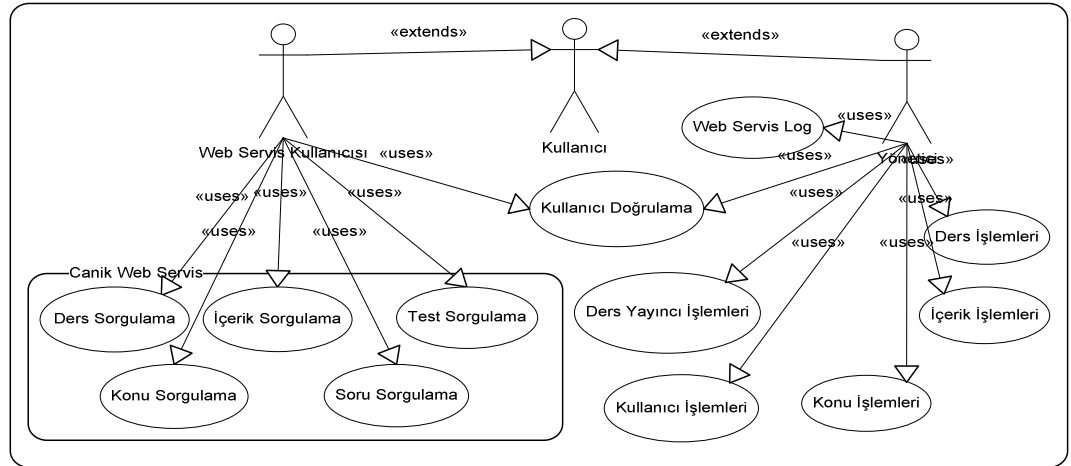
Canik Üniversitesi bünyesinde geliştirilecek elektronik öğrenme sistemi, hem üniversitenin kendi kullanıcılarına hizmet verirken aynı zamanda web servis arayüzü aracılığı ile dış sistemlere hizmet verecektir.

#### 4.2.1. Analiz

Canik elektronik öğrenme sisteminde iki çeşit aktör bulunmaktadır. Sistem yöneticisi sistemdeki genel tanımlamaları yapan ve sistemi yürüten kişidir. Ders ve kullanıcılar tanımlamak, mevcut web servisi kullanım durumlarını takip etmek sistem yöneticisinin görevidir.

Web servis kullanıcısı, Canik Üniversitesi tarafından sağlanmış web servislerinden, kendisine sağlanan yetki bünyesinde hizmet alan kullanıcıdır. Web servisi aracılığı ile ders içerik ve soru bilgisi sorgulama, web servisi kullanıcısının yaptığı işlemler arasında sayılabilir.

Şekil 4.8, Canik Üniversitesi bünyesindeki aktörleri ve kullanım durumlarını göstermiştir.



Şekil 4.8: Canik Üniversitesi Kullanım Durumu Diyagramı

##### 4.2.1.1. Kullanıcı Giriş

Canik sistemini kullanacak sistem yöneticisi ve öğrencilerin kimlik doğrulamalarının yapıldığı arayüzdür.

İşlev:

- Kullanıcı kodu ve kullanıcı şifresi girilir
- Kullanıcı kodu ve şifresi sitemde tanımlı ise ana sayfaya yönlendirilir

- Kullanıcı kodu ve şifresi sistemde tanımlı değil ise hata mesajı döndürülür

#### 4.2.1.2. Ana Sayfa

Kullanıcı doğrulaması yapıldıktan sonra kullanıcının önüne gelen ilk ara yüzdür

İşlev:

- Kullanıcıya yapmak istediği işlem için menü seçme imkânı verilir

#### 4.2.1.3. Kullanıcı İşlemleri

Sistem yöneticisinin kullanacağı, sisteme kullanıcılar ile ilgili işlemlerin yapılabileceği ara yüzdür.

Tanımlanacak bilgiler:

- Kullanıcı tipi, listeden seçmeli
- Adı, düz metin
- Soyadı, düz metin
- Kullanıcı kodu, düz metin
- Şifre, düz metin
- Öğrenci numarası, pozitif tam sayı

İşlev:

- Sistemde tanımlı kullanıcılar bir liste halinde sıralanır
- Listede kullanıcı tipi, kullanıcı adı, kullanıcı kodu, şifre, öğrenci numarasına göre sıralama imkanı sunulur
- Sistem yöneticisi, sisteme yeni kullanıcı ekleyebilir
  - Tanımlanacak kullanıcının kullanıcı tipi, *Öğrenci*, *Sistem Yöneticisi* ya da *Web Servis Kullanıcısı* olabilir.
  - Yeni bir kullanıcı eklenirken, eğer kullanıcı tipi öğrenci seçilmiş ise kullanıcının hangi öğrenciye ait olduğu bilgisinin anlaşılabilmesi için, bağlı bulunan öğrencinin numarası sisteme tanıtılır.
- Sistem yöneticisinin, sistemdeki mevcut kullanıcıları silmesine imkan verilir
  - Silinecek olan kullanıcıya bağlı öğrenci varsa, öğrencinin silinmeden önce, sistemde öğrenciye bağlı ders kaydı gibi başka kaydın olmaması gereklidir.
  - Silinecek olan web servis kullanıcılarına ait sistemde web servis kullanım geçmişi var ise, kullanıcının silinmesine izin verilmez.

#### 4.2.1.4. Ders İşlemleri

Sistem yöneticisinin kullanacağı, sistemdeki derslerle ilgili işlemlerin yapılabileceği ara yüzüdür.

Tanımlanacak bilgiler:

- Ders adı, düz metin
- Ders kodu, düz metin
- Durum; *aktif* seçili ise, öğrencilerin derse kayıt yapabildiklerini, eğer *pasif* ise derse kayıt yapamadıklarını gösterir.
- Web servisinde görünsün; *evet* ise, dersin içerik bilgilerinin web servis aracılığı ile dış sistemlere verilebileceğini, *hayır* ise verilemeyeceğini gösterir.

İşlev:

- Sistemde tanımlı dersler bir liste halinde sıralanır.
- Listedeki dersleri, ders adı, ders kodu, durum ve konu sayısı alanlarına göre sıralama imkanı sunulur.
- Sistem yöneticisi, sisteme yeni dersler ekleyebilir.
- Sistem yöneticisi, sistemden dersleri silebilir.
- Sistem yöneticisi, sistemdeki derslerin ismini, kodunu ve durum bilgilerini değiştirebilir.
- Sistem yöneticisi, seçtiği dersin konu işlemlerine yönlenebilmelidir.

#### 4.2.1.4. Konu İşlemleri

Sistem yöneticisinin kullanacağı, sistemdeki derslere konu ilgili işlemlerin yapılabileceği ara yüzüdür.

Tanımlanacak bilgiler:

- Konu Adı, düz metin
- Konu Sırası, sayı
- Durum; *aktif* seçili ise, öğrencilerin konuları görebilmelerini sağlar

İşlev:

- Seçili dersin tanımlı konuları bir liste halinde sıralanır.
- Listedeki konuları ad, açıklama, durum, içerik sayısı alanlarına göre sıralama imkanı sunulur.
- Sistem yöneticisi, derse yeni konu ekleyebilir.
- Sistem yöneticisi, mevcut konuları silebilir.

- Silinecek konuya ait sistemde herhangi bir içerik olmaması önşartı vardır.
- Sistem yöneticisi, sistemdeki konuların ismini, açıklamasını ve durum bilgilerini değiştirebilir.
- Sistem yöneticisi istediği konunun içerik işlemlerine yönlenebilmeli

#### 4.2.1.4. İçerik İşlemleri

Sistem yöneticisinin kullanacağı, sistemdeki konulara ait içerik ile ilgili işlemlerin yapılabileceği ara yüzdür.

Tanımlanacak bilgiler:

- İçerik adı, düz metin
- İçerik sırası, sayı
- Durum; *aktif* seçili ise, öğrencilerin içeriği görebilmelerini sağlar

İşlev:

- Seçili dersin tanımlı konuları bir liste halinde sıralanır.
- Listedeki içerikleri ad, açıklama ve durum alanlarına göre sıralama imkanı sunulur.
- Sistem yöneticisi, konuya HTML formatta yeni içerik ekleyebilir.
- Sistem yöneticisi, mevcut içerikleri silebilir.
- Sistem yöneticisi, sistemdeki içerik ismini, sırasını, durumu ve HTML olarak içerik detay bilgilerini değiştirebilir.

#### 4.2.1.5. Web Servis Kullanım Bilgileri

Sistem yöneticisinin Canik web servisinin kullanım durumunu izlemesine imkân veren ara yüzdür.

İşlev:

- Web servisin vermiş olduğu hizmetin geçmiş bilgilerini listelenir. Kullanılan web servisinin şu bilgilerinin izlenmesini sağlar.
  - a. Web servis kullanıcısı
  - b. Web servis metodu
  - c. Kullanılan metodun parametreleri
  - d. Kullanım tarihi

#### 4.2.1.6. Web Servis Kimlik Doğrulama

Web servise gelen istekleri cevaplamadan önce isteği yapan kişinin kimlik doğrulaması yapılır.

İşlev:

Web servis isteği içinde bulunan SOAP başlığındaki kullanıcı adı ve kullanıcı şifresi alınarak veritabanında kayıtlı olan kullanıcılar arasında olup olmadığına bakılarak, kullanıcının web servisini kullanma yetkisi kontrol edilir. İşlemin sonucuna göre web servisi isteğe cevap verir.

#### 4.2.1.7. Web Servis Ders Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu ders listesini öğrenmek için kullandığı metottur.

İşlev:

Ders adı, ders kodu bilgilerinden oluşan derslerin listesi XML paket olarak istemciye döndürülür.

#### 4.2.1.8. Web Servis Ders Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu ders listesini öğrenmek için kullandığı metottur.

İşlev:

Ders adı, ders kodu bilgilerinden oluşan derslerin listesi XML paket olarak istemciye döndürülür.

#### 4.2.1.9. Web Servis Konu Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu dersin konu listesini öğrenmek için kullandığı metottur.

İşlev:

Ders kodu girdi olarak verilen dersin, konu kimliği, konu adı, konu açıklaması, konu sırası bilgilerinden oluşan konu listesi XML paket olarak istemciye döndürülür.

#### 4.2.1.10. Web Servis İçerik Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu dersin bir konusuna bağlı içerik listesini öğrenmek için kullandığı metottur.

İşlev:

Konu kimliği girdi olarak verilen konunun, içerik adı, içerik detayı, içerik sırası bilgilerinden oluşan içerik listesi XML paket olarak istemciye döndürülür.

#### 4.2.1.11. Web Servis Sınav Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu dersin sınav listesini öğrenmek için kullandığı metottur.

İşlev:

Ders kodu girdi olarak verilen dersin, sınav kimliği, sınav adı, sınav açıklaması, sınav sırası, sınav süresi bilgilerinden oluşan sınav listesi XML paket olarak istemciye döndürülür.

#### 4.2.1.12. Web Servis Soru Sorgulama

Web servis kullanıcısının, Canik Üniversitesinin web servis aracılığı ile sunmuş olduğu dersin bir sınavına bağlı soru listesini öğrenmek için kullandığı metottur.

İşlev:

Sınav kimliği girdi olarak verilen sınavın, soru sırası, soru metni, cevap seçenekleri, doğru seçeneği bilgilerinden oluşan soru listesi XML paket olarak istemciye döndürülür.

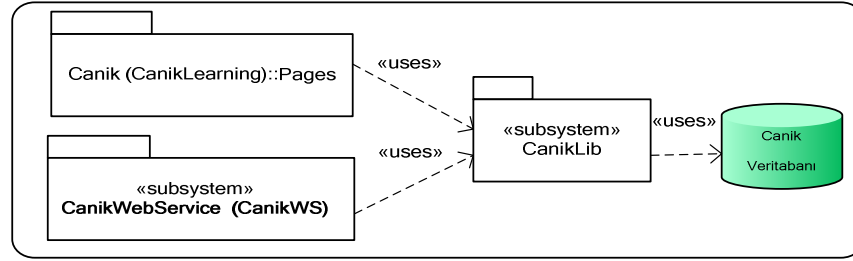
### 4.2.2. Tasarım

Canik Üniversitesi, elektronik öğrenme ihtiyaç analizini karşılamak için, CanikLearning ve CanikWS isimli iki alt sistem tasarlandı.

*CanikLearning*, Canik Üniversitesi bünyesindeki sistem yöneticisinin ders ve kullanıcı bilgilerini tanımlayarak sistemi kurgulayabileceği, gerekirse kendi öğrencilerinin de ders takibi yapabilmesi için sistemin genişletilebileceği, web tabanlı bir elektronik öğrenme sistemidir.

*CanikWS*, Canik Üniversitesinin sağlamış olduğu ders içerik bilgisinden dış sistemlerinde faydalanabilmesini sağlayacak web servis projesidir.

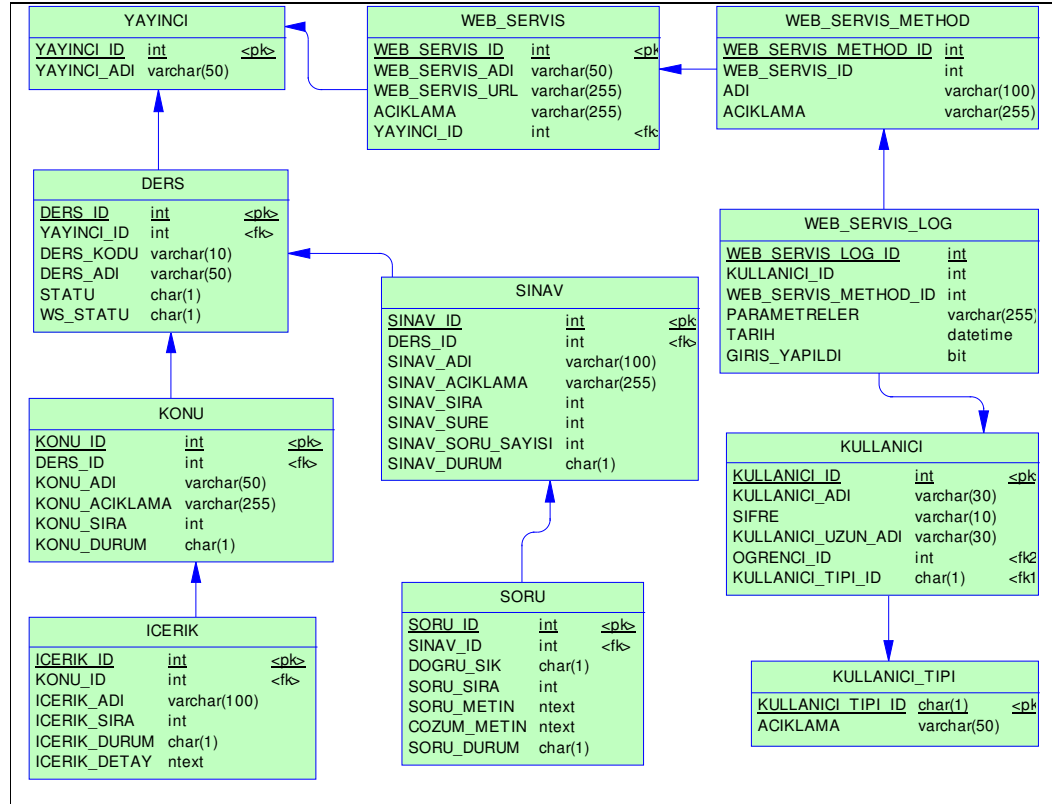
Şekil 4.9 a bakıldığında, her iki alt sistemin de Canik veritabanından hizmet aldığı görülmektedir. Her iki alt sistem için de ortak yapılacak işlemler, ayrı bir alt sistemde toplanabilir. *CanikLib*, bu mantıktan çıkılarak ortak bir kitaplık projesi olarak tasarlanmış, veritabanına erişim diğer alt sistemlerden bu şekilde soyutlanmıştır.



Şekil 4.9: Canik Üniversitesi Genel Tasarım

#### 4.2.2.1. Veritabanı Tasarımı

Canik Üniversitesi elektronik öğrenme sistemini için geliştirilen ilişkisel veritabanı bünyesindeki tablolar Şekil 4.10 da gösterilmiştir.



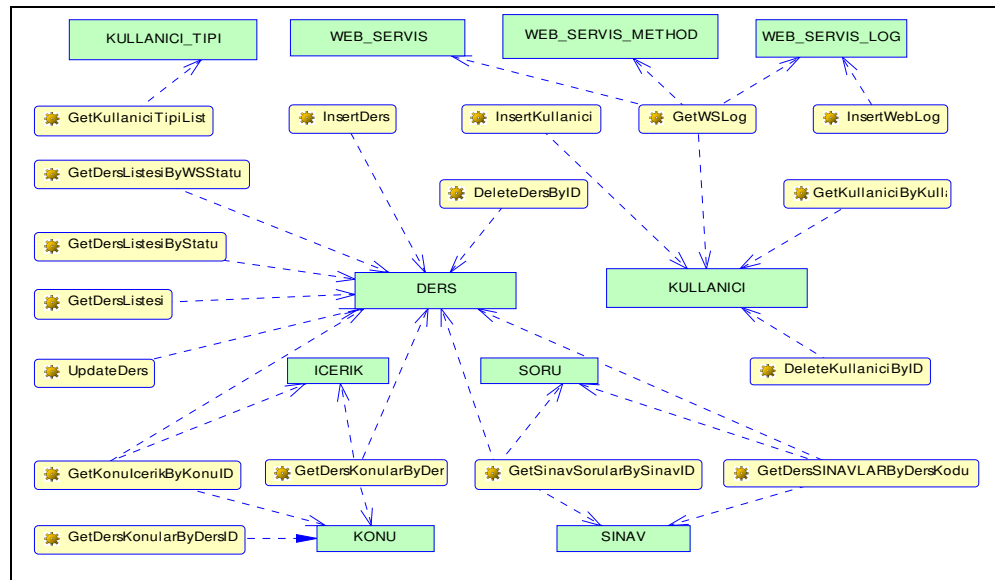
Şekil 4.10: Canik Veritabanı Tasarımı (Tablo İlişkileri)

Şekilde de görüldüğü gibi her tablo üzerinde tekliği sağlayacak kimlik alanları bulunmaktadır. Tablolar arasındaki ilişki kurulurken bu kimlik alanlarından yararlanılmaktadır. Canik veritabanında yer alan tablolar ve bu tabloların kullanım amaçları aşağıdaki Tablo 4.3 de verilmiştir.

Tablo 4.3 : Canik Veritabanını oluşturan tablo açıklamaları

Tablo Adı	Açıklama
KULLANICI	CanikLearning ve CanikWS uygulamalarını kullanabilecek kullanıcıların bilgilerinin tutulduğu tablodur
KULLANICI_TIPI	Kullanıcı tip bilgilerinin tutulduğu tablodur
YAYINCI	Derslerin içerik yayını yapan kurumların bilgisinin tutulduğu tablodur
WEB_SERVIS	Sistemde sunulan web servislerinin bilgilerinin tutulduğu tablodur.
WEB_SERVIS_METHOD	Sunulan web servise ait metod bilgileri tutulur.
WEB_SERVIS_LOG	Sunulan web servis metodlarının kullanım geçmişlerinin tutulduğu tablodur
DERS	Sistemdeki derslerin bilgilerinin tutulduğu tablodur
KONU	Derslere bağlı konularının tutulduğu tablodur
ICERIK	Ders konularının içeriklerinin tutulduğu tablodur
SINAV	Derslere bağlı test sınavların başlık bilgilerinin tutulduğu tablodur
SORU	Sınavların soru detaylarının tutulduğu tablodur

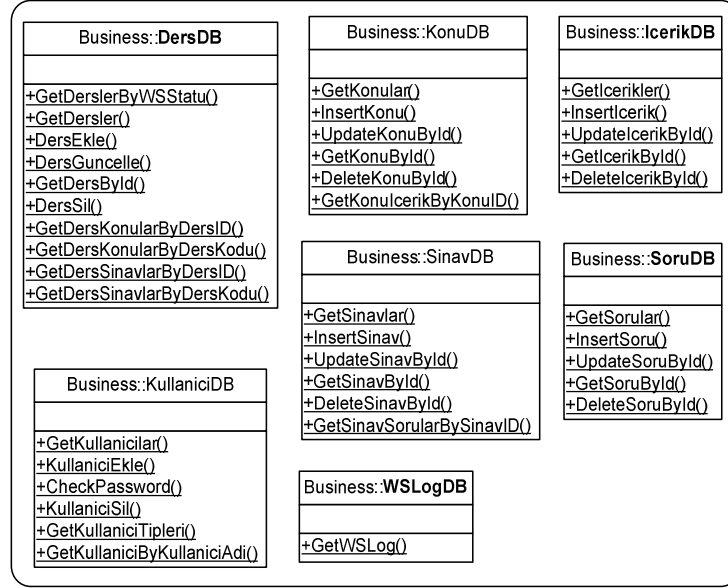
Canik veritabanı üzerindeki tablolara erişim yine veritabanı üzerindeki saklı yordamlar aracılığı ile sağlanmıştır. Genel olarak sistemdeki her bir tabloya yeni kayıt eklemek, mevcut kayıtları güncellemek veya silmek ve tablolar arası ilişkilerden de yararlanarak ihtiyaç doğrultusunda sorgulara cevap verebilmek amaçlı geliştirilen yordamlar ve bu yordamların hangi veritabanı tabloları üzerinde işlem gerçekleştirdiği Şekil 4.11 de gösterilmiştir.



Şekil 4.11: Canik Veritabanı Tasarımı (Saklı Yordam Tablo İlişkileri)

#### 4.2.2.2. Veri Erişim Katmanı Tasarımı

Canik e-öğrenme sistemi için veri erişim işlemleri CanikLib bünyesinde toplanmıştır. CanikLearning ve CanikWS alt sistemleri içinde veritabanı ihtiyaçları CanikLib bünyesindeki veri erişim katmanına yönlendirilir. Böylelikle her iki alt sistem için de veriye erişim tek bir kaynaktan yapılmış olur.

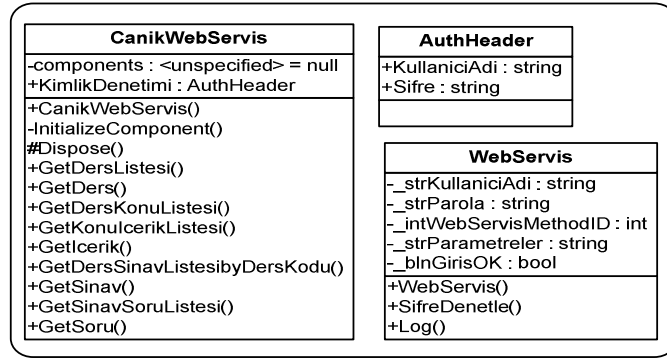


Şekil 4.12: CanikLib Veri Erişim Katmanı

Şekil 4.12 deki CanikLib için tasarlanan veri erişim katmanına göre, kullanıcı bilgilerine erişim için KullaniciDB, ders bilgilerine ulaşmak için DersDB, ders sınav bilgilerine ulaşmak için SinavDB, ders konu bilgilerine ulaşmak için KonuDB, içerik bilgilerine ulaşmak için IcerikDB, sınav soru bilgilerine ulaşmak için SoruDB ve web servis kullanım geçmişi bilgilerine ulaşmak için WLogDB sınıfları kullanılmaktadır.

#### 4.2.2.3. Web Servisi Tasarımı

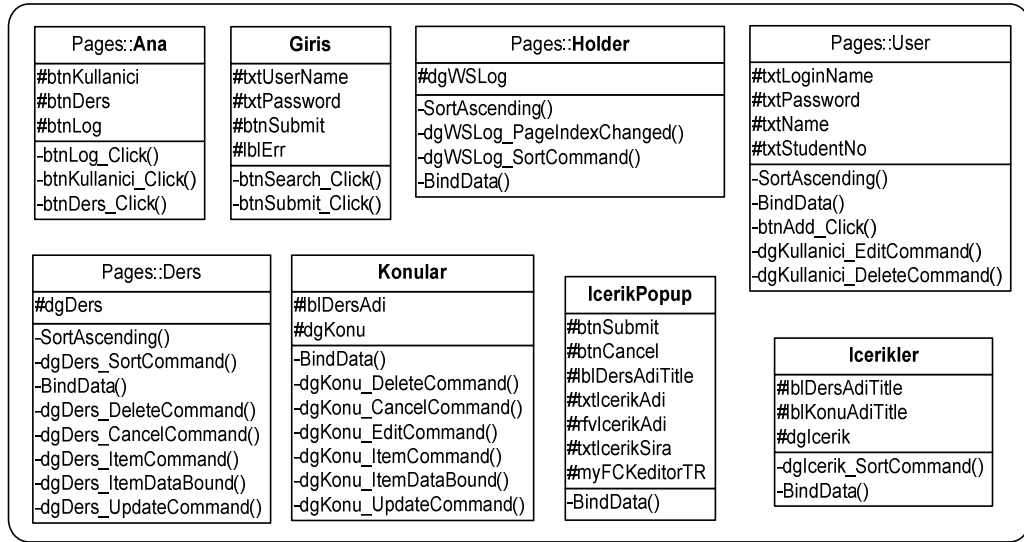
CanikWS, güvenlik için web servislerin SOAP başlık mekanizmasını kullanır. Üyelerine daha önceden bildirmiş olduğu kullanıcı adı ve şifresini, web servisinden istekte bulunmadan SOAP başlığı olarak girmelerini bekler. Kendine gelen istekte açılan SOAP başlık bilgisindeki kullanıcı adı ve şifresini veritabanındaki kullanıcı adı ve şifresi işe karşılaştırır. Sonuca göre, isteğe cevap verip vermeyeceğine karar verir.



Şekil 4.13: CanikWS Sınıf Diyagramı

#### 4.2.2.4. Sayfa Tasarımı

CanikLearning için geliştirilen sayfaların arka planındaki sınıf tasarımı şekil 4.14 de gösterilmiştir. Buna göre CanikLearning bünyesinde yayıncı işlemleri, kullanıcı işlemleri, ders işlemleri ve ders içeriği hazırlamaya yönelik sayfalar bulunmaktadır. Ders içeriklerin HTML olarak hazırlanmasına imkan vermek için FCKeditor isimli açık kaynak kodlu HTML düzenleyicisi geliştirilen sayfaya entegre edilmiştir [25].



Şekil 4.14: CanikLearning Sunum Katmanı Sınıf Diyagramı

### **4.3 GELİŞTİRME ORTAMI VE VERİTABANI**

#### **4.3.1. Geliştirme Ortamı**

Geliştirilecek olan iki elektronik eğitim sistemi ve aralarındaki iletişimi kuracak olan web servisinin geliştirme ortamları için ön koşul, bugün dünyada en çok tercih edilen işletim sistemi olarak öne çıkan Microsoft Windows üzerinde çalışacak şekilde olmasıdır.

Web uygulamalarının uygulama sunucusu için, Microsoft Windows üzerinde hazır olarak gelen, IIS (Internet Information Services) tercih edilmiştir. Uygulamalar, Microsoft Visual Studio 2003 sürümü geliştirme platformu kullanılarak, C# dilinde hazırlanmıştır. Dolayısıyla, geliştirilen kodlar, çalıştırılabilmesi için Microsoft .Net Framework 1.1.'e gereksinim duyarlar.

#### **4.3.2. Veritabanı**

Her iki üniversite için geliştirilen elektronik eğitim sisteminin verilerinin yer alacağı veritabanı olarak, açık kaynak kodlu veritabanları yerine kurumsal tedarikçilerden biri olan Microsoft SQL Server tercih edilmiştir. Diğerlerine göre toplam sahip olma maliyetinin daha uygun olması ve Microsoft.NET platformu ile uygunluğu sebebiyle Microsoft SQL seçilmiştir. Veritabanı tasarımı PowerDesigner yardımcı programı kullanılarak geliştirilmiştir. Veritabanındaki tablolar ve saklı yordamlar arasındaki ilişkiyi de gösterebilmesi PowerDesigner ürününün tercih edilmesine sebep olmuştur.

## 5. SONUÇ

XML web servislerin Haziran 2000 yılında çıkışından bu yana, firmaların yoğun ilgisi ile karşılanmıştır. Web servisleri alanında, sektörün büyük bir hızla gereksinimleri belirlenmiş ve standardizasyon çalışmalarına başlanmış ve olgunlaşmaya başlamıştır. Tanınmış olduğu yüksek esneklik sayesinde, uygulamalar arası iş birlikteliğini sağlaması konusunda web servislerinin hâkim ortak olacağı açıktır.

Mevcut e-öğrenme sistem mimarilerinin birçoğu esas olarak istemci-sunucu mimarilerine dayalı olması, halen ölçeklenebilirlik problemleri ve karmaşık içerik alışveriş işlemlerini beraberinde getirir. Ancak, öğrenim nesne ambarları ve öğrenim içerik yönetim sistemi gibi e-öğrenme uygulamaları arasında entegrasyonu sağlamanın etkili bir yolunun web servislerinin kullanımından geçtiği gözükmektedir. Böylece web servisleri kullanan e-öğrenme sistemleri için servis tabanlı, dağıtık bir mimari sunulmuş olmaktadır.

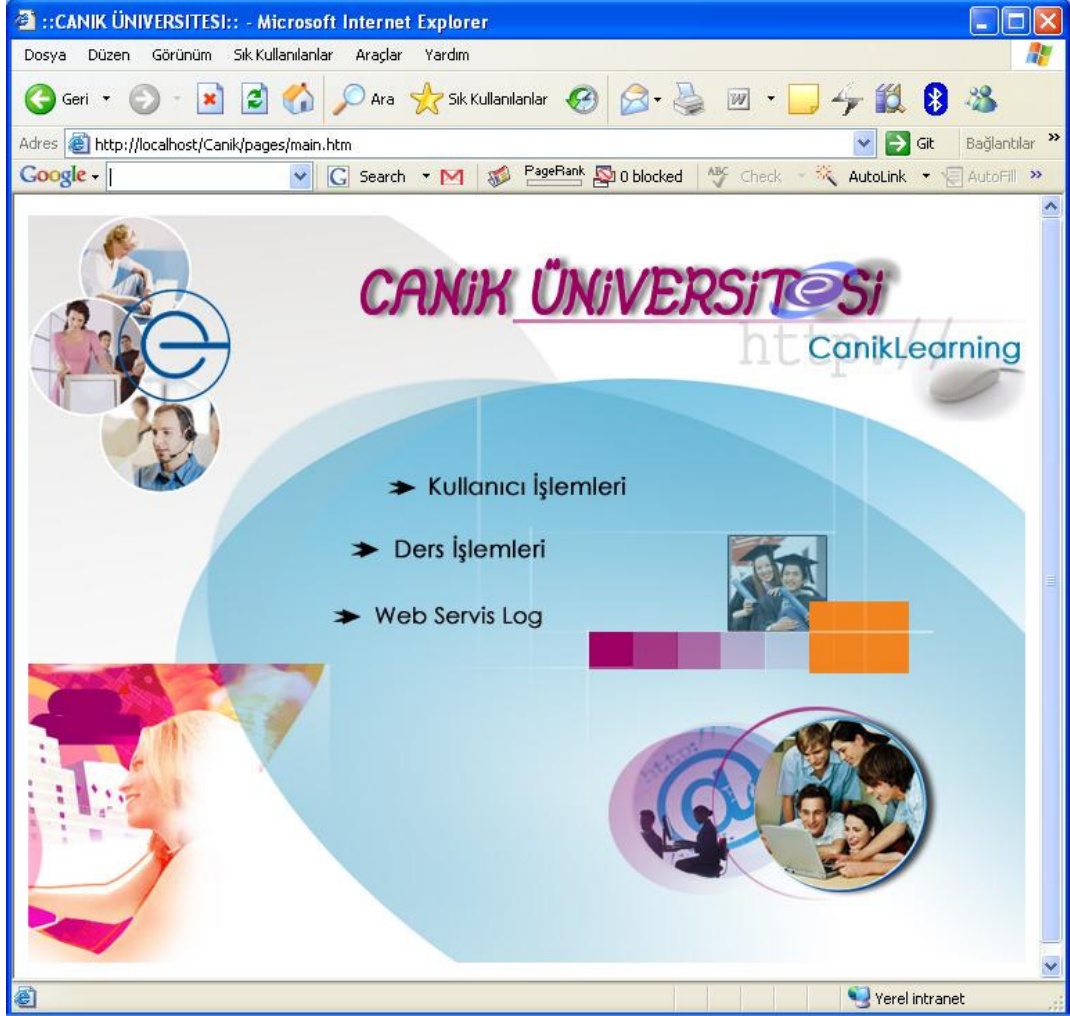
İleride e-öğrenme sistemlerinden sunmuş oldukları servisler aracılığı ile hizmet alan ve pek çok e-öğrenme nesnesini farklı servislerden bulup kullanıcının önüne getiren uygulamaların gelişeceği düşünülmektedir.

## KAYNAKLAR

1. MASON, Jim, 2006, *SGML and XML markup languages in vendor-neutral, machine-independent and human-readable format* [online], <http://www.iso.org/iso/en/commcentre/pdf/Sgml0002.pdf> [Ziyaret Tarihi: 11 Ocak 2006].
2. BİLGİLİ Emre, 2002, *SOAP Nedir? Avantajları Nelerdir – I* [online], <http://www.evcil.net/Makale/61.aspx> [Ziyaret Tarihi: 15 Ocak 2006].
3. AYAN, Evren, 2005, *XML Nedir?-Ne İşimize Yarar?* [online], [http://www.maxiasp.net/d\\_xml\\_nedir\\_ne\\_isimize\\_yarar.aspx](http://www.maxiasp.net/d_xml_nedir_ne_isimize_yarar.aspx) [Ziyaret Tarihi: 15 Ocak 2006].
4. QUIN, Liam, 2003, *Extensible Markup Language (XML)* [online], <http://www.w3.org/XML/> [Ziyaret Tarihi: 15 Ocak 2006].
5. JORGENSEN, David, 2002, *Developing .NET Web Services with XML*, Syngress, Rockland MA, 1-928994-81-4
6. Software AG, 2005, *XML Nedir?*, İstanbul, <http://bilisim.milliyet.com.tr/detay.asp?id=123> [Ziyaret Tarihi: 15 Ocak 2006].
7. Doug TIDWELL, James SNELL, Pavel KULCHENKO, 2001, *Programming Web Services with SOAP*, O'Reilly, Sebastopol CA, 0-596-00095-2
8. HAN, CENGİZ., 2004, *XML Web Servisleri – I* [online]. İstanbul, <http://www.yazgelistir.com/Makaleler/1000000422.ygpx> [23.02.2006].
9. 2006, *Web Service* [Online], [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service) [Ziyaret Tarihi: 14.02.2006]
10. BUTLER, JASON, 2002, *ASP.NET Database Programming Weekend Crash Course*, Hungry Minds, Inc., NY, 0-7645-4830-1
11. ŞENYURT Burak Selim, 2004, *XML Web Servisleri-I* [online], <http://www.csharpnedir.com/makalegoster.asp?MId=370> [Ziyaret Tarihi: 15 Haziran 2006].
12. AKYOKUŞ Selim, 2001, *Web Servisleri: İnternet Devriminde İkinci Aşama* [online], İstanbul, Doğu Üniversitesi, <http://sakyokus.ce.dogus.edu.tr/Presentations/wspresent.ppt> [Ziyaret Tarihi: 20 Haziran 2006]
13. TÜRKELİ Yener, 2004, *XML Web Servisleri* [online], İstanbul, [http://www.aspnedir.com/makaleler\\_icerik.asp?id=500](http://www.aspnedir.com/makaleler_icerik.asp?id=500) [Ziyaret Tarihi: 2 Haziran 2006]
14. ALGAN Sefer, 2003, *C# Web Servislerine Giriş* [online], İstanbul, <http://www.csharpnedir.com/makalegoster.asp?MId=49> [Ziyaret Tarihi: 2 Haziran 2006]
15. SULTAN Mehmet, 2004, *UDDI*, [online], İstanbul, [http://www.msakademik.net/makaleler\\_detay.aspx?id=269](http://www.msakademik.net/makaleler_detay.aspx?id=269) [Ziyaret Tarihi: 2 Haziran 2006]
16. ÜLKER M.Musa, *XML Web Servisleri*, <http://www.verivizyon.com/detail.asp?cid=193>

17. AÇIKEL Emre, 2005, XML Web Servislerine Giriş, <http://www.mobilnedir.com/makalegoster.asp?id=42>
18. *The Advantages of Using Web Services*, , [online], Burlington, MA, <http://www.systinet.com/doc/ssc-65/primer/html/using.web.services.html>[Ziyaret Tarihi: 2 Haziran 2006]
19. *Introduction to Web Services*, [online], Burlington, MA, <http://www.systinet.com/doc/ssc-65/primer/html/intro.to.web.services.html>[Ziyaret Tarihi: 2 Haziran 2006]
20. *Teknoloji Eğitimi Yaygınlaştırıyor: e-Öğrenim* [online], İstanbul, <http://www.msakademik.net/haberdetay.aspx?id=78>[Ziyaret Tarihi: 2 Haziran 2006]
21. WESTERKAMP, Peter, 2003, E-Learning as a Web Service, *7th International Database Engineering and Applications Symposium, 16-18 Temmuz 2003*, Hong Kong, IEEE Computer Society, 242
22. PANKRATIUS, V., SANDEL O., STUCKY, W., 2004, Retrieving Content With AgentsIn Web Service E-Learning Systems, *First IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), August 2004*, Toulouse, France, 91-100
23. VUORIKARI Riina, 2003, Web Services to Achieve Interoperability Between eLearning Applications [online], [http://www.eun.org/eun.org2/eun/en/Insight\\_Standardization/content.cfm?ov=27853&lang=en](http://www.eun.org/eun.org2/eun/en/Insight_Standardization/content.cfm?ov=27853&lang=en) [Ziyaret Tarihi: 20 Mayıs 2006].
24. TERNIER Stefaan, DUVAL Erik, 2003, *Web services for the ARIADNE Knowledge Pool System* [online], Heverlee, Belgium, <http://ariadne.cs.kuleuven.ac.be/ariadne/CONF2003/papers/TER2003.pdf> [Ziyaret Tarihi: 20 Mayıs 2006].
25. <http://www.fckeditor.net/> [Ziyaret Tarihi: 20 Haziran 2006].

## EK-A



Şekil A1: CanikLearning Ekran Görüntüsü

Address: http://localhost/Toros/pages/MainFrame.htm

**TOROS ÜNİVERSİTESİ**  
e-toros

Ana Sayfa | Çıkış |

**DERSLERİM**

Ders Kodu	Ders Adı	Kayıt Tarihi	
CS 202	Bilgisayar II	13.06.2006 17:24:30	<a href="#">Sil</a>
MATH 101	Matematiğe Giriş	01.01.2005 00:00:00	<a href="#">Sil</a>
TRH 101	Tarih	13.06.2006 17:34:03	<a href="#">Sil</a>
TRH 102	Tarih II	14.06.2006 22:46:39	<a href="#">Sil</a>
TRH 201	Osmanlı Tarihi I	22.06.2006 14:28:24	<a href="#">Sil</a>

1

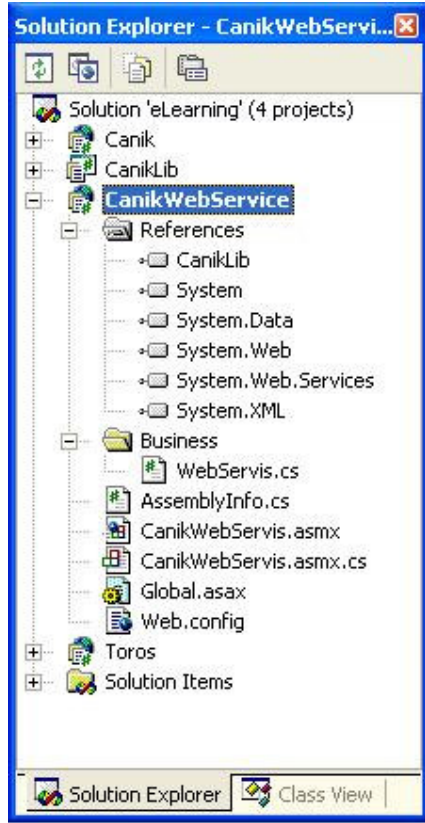
[Ders Ekle](#)

Yerel intranet

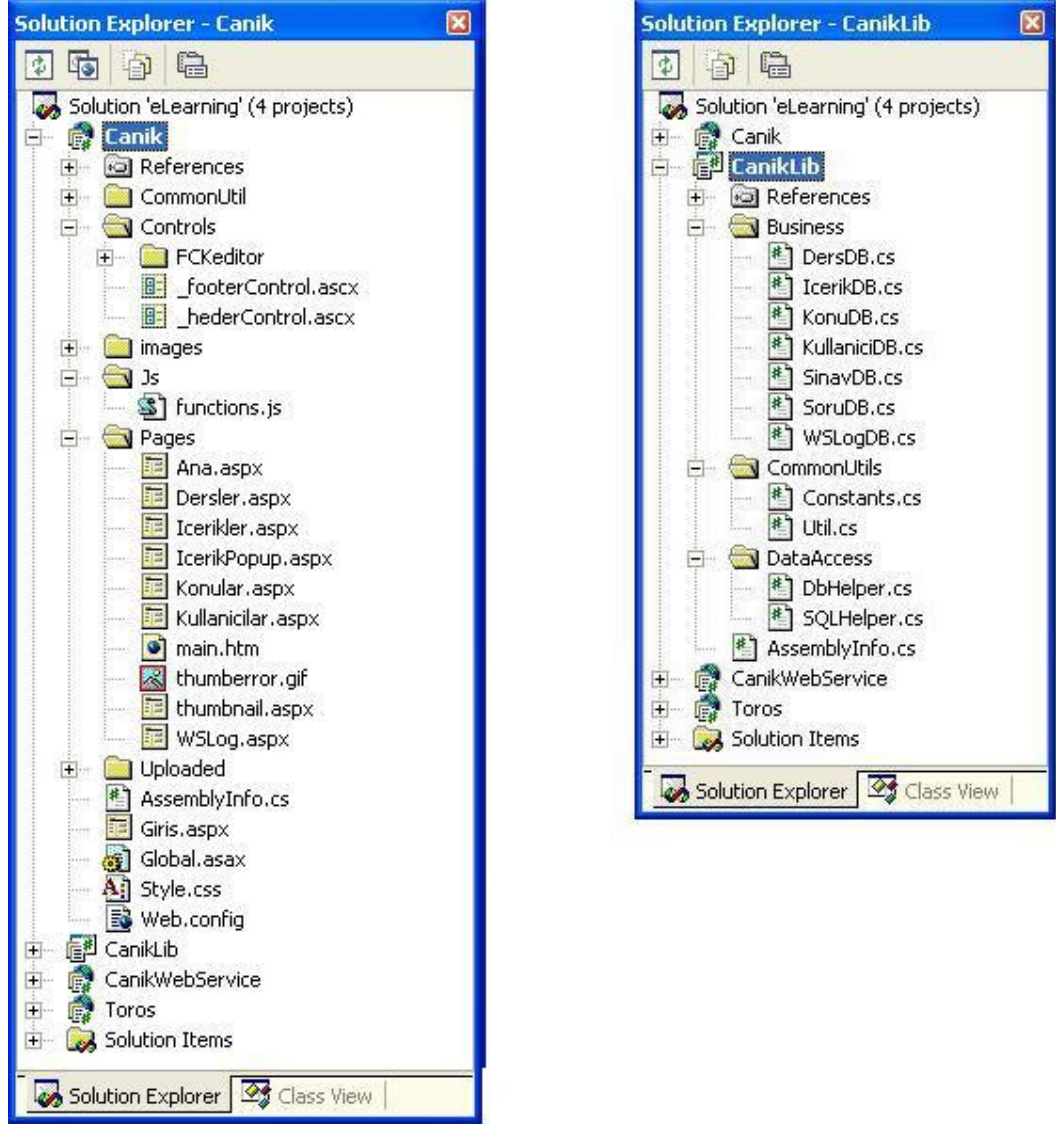
Şekil A.2: E-Toros Ekran Görüntüsü

**EK-B**

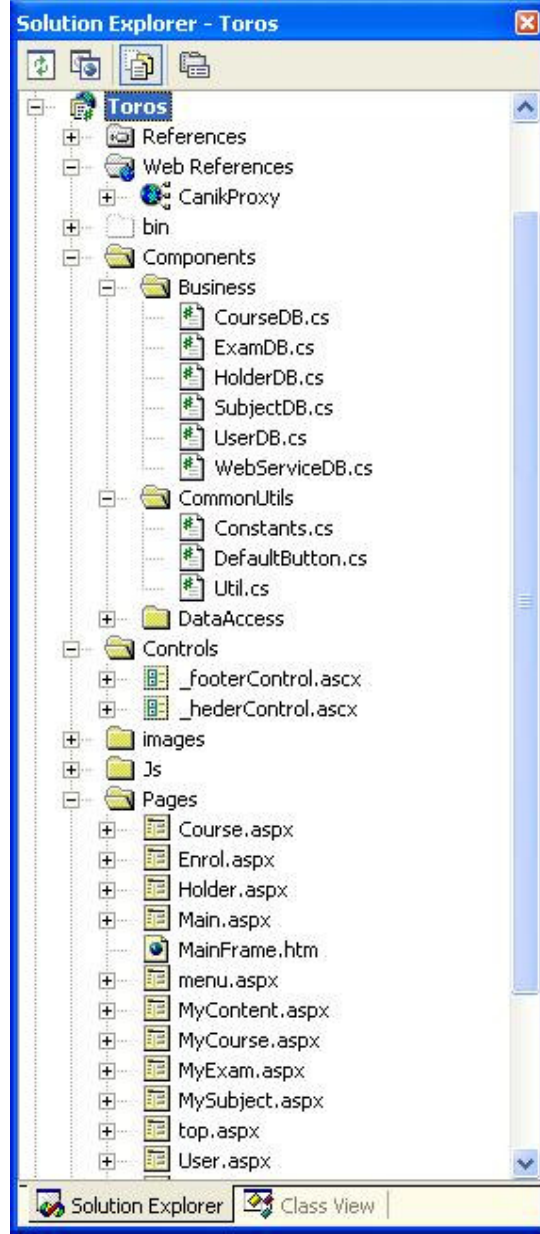
Şekil B.1: E-Öğrenme Çözümü İçin Gelistirilen 4 Proje Dosyasının VS 2003 Görüntüsü



Şekil B.2: CanikWS Proje Dosyasının VS 2003 Görüntüsü



Şekil B.3: Canik ve CanikLib Projeleri Dosyaları VS 2003 Görüntüsü



Şekil B.4: Toros Proje Dosyasının VS 2003 Görüntüsü

## EK-C

CanikWS projesindeki örnek iki adet kaynak kod:

### CanikWebServis.asmx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using CanikLib.Business;
using CanikLib;
using CanikWebService;

namespace CanikWebService
{
    /// <summary>
    /// Summary description for CanikLib.
    /// </summary>
    public class CanikWebServis: System.Web.Services.WebService
    {
        public CanikWebServis()
        {
            //CODEGEN: This call is required by the ASP.NET Web Services
            Designer.InitializeComponent();
        }

        #region Component Designer generated code

        //Required by the Web Services Designer
        private IContainer components = null;

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
        }

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if(disposing && components != null)
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #endregion

        public AuthHeader KimlikDenetimi;

        [SoapHeader("KimlikDenetimi")]
        [WebMethod]

```

```

public DataSet GetDersListesi()
{
    DataSet dsResult = null;
    WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetDersListesi, "");
    if (webServis.SifreDenetle())
    {
        dsResult =
DersDB.GetDerslerByWSStatu(Constants.WEB_SERVIS_DERSI);
    }
    webServis.Log();
    return dsResult;
}

[SoapHeader("KimlikDenetimi")]
[WebMethod]
public DataSet GetDers(int dersID)
{
    DataSet dsResult = null;
    WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetDers, dersID.ToString());
    if (webServis.SifreDenetle())
    {
        dsResult = DersDB.GetDersById(dersID);
    }
    webServis.Log();
    return dsResult;
}

[SoapHeader("KimlikDenetimi")]
[WebMethod]
public DataSet GetDersKonuListesi(string dersKodu)
{
    DataSet dsResult = null;
    WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetDersKonuListesi, dersKodu.ToString());
    if (webServis.SifreDenetle())
    {
        dsResult = DersDB.GetDersKonularByDersKodu(dersKodu, 'E');
    }
    webServis.Log();
    return dsResult;
}

[SoapHeader("KimlikDenetimi")]
[WebMethod]
public DataSet GetKonuIcerikListesi(int konuID)
{
    DataSet dsResult = null;
    WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetKonuIcerikListesi, konuID.ToString());
    if (webServis.SifreDenetle())
    {
        dsResult = KonuDB.GetKonuIcerikByKonuID(konuID);
    }
    webServis.Log();
    return dsResult;
}

[SoapHeader("KimlikDenetimi")]
[WebMethod]
public DataSet GetIcerik(int icerikID)
{

```

```

        DataSet dsResult = null;
        WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetIcerik, icerikID.ToString());
        if (webServis.SifreDenetle())
        {
            dsResult = IcerikDB.GetIcerikById(icerikID);

        }
        webServis.Log();
        return dsResult;
    }

    [SoapHeader("KimlikDenetimi")]
    [WebMethod]
    public DataSet GetDersSinavListesiByDersKodu(string dersKodu)
    {
        DataSet dsResult = null;
        WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetDersSinavListesi, dersKodu.ToString());
        if (webServis.SifreDenetle())
        {
            dsResult = DersDB.GetDersSinavlarByDersKodu(dersKodu,
'E');

        }
        webServis.Log();
        return dsResult;
    }

    [SoapHeader("KimlikDenetimi")]
    [WebMethod]
    public DataSet GetSinav(int sinavID)
    {
        DataSet dsResult = null;
        WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetSinav, sinavID.ToString());
        if (webServis.SifreDenetle())
        {
            dsResult = SinavDB.GetSinavById(sinavID);

        }
        webServis.Log();
        return dsResult;
    }

    [SoapHeader("KimlikDenetimi")]
    [WebMethod]
    public DataSet GetSinavSoruListesi(int sinavID)
    {
        DataSet dsResult = null;
        WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetSinavSoruListesi, sinavID.ToString());
        if (webServis.SifreDenetle())
        {
            dsResult = SinavDB.GetSinavSorularBySinavID(sinavID);

        }
        webServis.Log();
        return dsResult;
    }

    [SoapHeader("KimlikDenetimi")]
    [WebMethod]
    public DataSet GetSoru(int soruID)
    {
        DataSet dsResult = null;
        WebServis webServis = new WebServis(KimlikDenetimi.KullaniciAdi,
KimlikDenetimi.Sifre, Constants.INTGetSoru, soruID.ToString());
        if (webServis.SifreDenetle())
        {

```

```

        dsResult = SoruDB.GetSoruById(soruID);
    }
    webServis.Log();
    return dsResult;
}

}

public class AuthHeader: SoapHeader
{
    public string KullaniciAdi;
    public string Sifre;
}

}

```

## WebServis.cs

```

using System;
using System.Data;
using System.Data.SqlClient;
using CanikLib.Business;
using CanikLib;

namespace CanikWebService
{
    /// <summary>
    /// Summary description for Log.
    /// </summary>
    public class WebServis
    {
        private string _strKullaniciAdi;
        private string _strParola;
        private int _intWebServisMethodID;
        private string _strParametreler;
        private bool _blnGirisOK;

        public WebServis()
        {
            //
            // TODO: Add constructor logic here
            //
        }

        public WebServis(string strKullaniciAdi,
            string strParola,
            int intWebServisMethodID,
            string strParametreler)
        {
            _strKullaniciAdi = strKullaniciAdi;
            _strParola = strParola;
            _intWebServisMethodID = intWebServisMethodID;
            _strParametreler = strParametreler;
        }

        public bool SifreDenetle()
        {
            DataRow dr =
                KullaniciDB.GetKullaniciByKullaniciAdi(_strKullaniciAdi);

            if(dr == null)
            {
                _blnGirisOK = false;
                return _blnGirisOK;
            }

            string dbPassword = dr["SIFRE"].ToString();

            _blnGirisOK = _strParola == dbPassword;
            return _blnGirisOK;
        }

        public bool Log()
        {
            bool ret = true;
            try
            {

```

```
        DataRow dr =
KullaniciDB.GetKullaniciByKullaniciAdi(_strKullaniciAdi);
        if(dr != null)
        {
            int kullaniciID =
Convert.ToInt32(dr["KULLANICI_ID"].ToString());

            DbHelper.ExecuteNonQuery("InsertWebLog",
                kullaniciID,
                _intWebServisMethodID,
                _strParametreler,
                DateTime.Now,
                _blnGirisOK
            );
        }
    }
    Catch
        ret=false;
    }
    return ret;
}
}
}
```

## ÖZGEÇMİŞ

16.08.1975 yılında Samsun'un Terme ilçesinde doğdu. İlk orta ve lise eğitimini aynı ilçede tamamladıktan sonra 1991 yılında Bilkent Üniversitesi, Bilgisayar ve Enformatik Mühendisliği Bölümüne girdi. 1996 yılında lisans eğitimini tamamlamasının ardından özel sektörde çalışma hayatına atıldı. İş yaşamı sürerken 2003 yılında İstanbul Üniversitesi Enformatik Bölümü'nde yüksek lisans eğitimine başladı.

Çalışma hayatında, Kurumsal Kaynak Planlaması (ERP) dalında uzun yıllar analist-yazılımcı ve danışman olarak hizmet etti. Şirketlerde iş süreçlerinin iyileştirilmesi ve ERP sistemine geçiş konusunda danışmanlık hizmeti verdi.

Halen özel bir şirkette iş süreçlerinin, programlarının ve veri ambarlarının yeniden yapılanması konusunda analist-programcı olarak görev yapmaktadır.

Web servisleri kullanarak ERP sistemleri arasında entegrasyon kurulması özel ilgi alanıdır.