

**ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**  
**YÜKSEK LİSANS TEZİ**

**AB BOLOGNA SÜRECİNDE**  
**ÜNİVERSİTELER İÇİN WEB TABANLI**  
**AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ**

**Yücel ŞAHİN**

**Yrd.Doç.Dr. İsmail KADAYIF**

**Ocak, 2008**  
**ÇANAKKALE**

**AB BOLOGNA SÜRECİNDE  
ÜNİVERSİTELER İÇİN WEB TABANLI  
AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ**

**Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü  
Yüksek Lisans Tezi  
Bilgisayar Mühendisliği Anabilim Dalı**

---

**Hazırlayan  
Yücel ŞAHİN**

**Danışman  
Yrd.Doç.Dr. İsmail KADAYIF**

**Ocak, 2008  
ÇANAKKALE**

**Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Müdürlüğüne,**

Bu araştırma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

**Başkan** : Yrd. Doç. Dr. İsmail KADAYIF

**Üye** : Doç. Dr. İsmail TARHAN

**Üye** : Yrd. Doç. Dr. İbrahim TÜRKYILMAZ

**Kod No :**

**Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.**

**Enstitü Müdürü**

## YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

YÜCEL ŞAHİN tarafından YRD.DOÇ.DR. İSMAİL KADAYIF yönetiminde hazırlanan “AB BOLOGNA SÜRECİNDE ÜNİVERSİTELER İÇİN WEB TABANLI AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Yrd.Doç.Dr. İsmail KADAYIF

---

Yönetici

Doç.Dr. İsmail TARHAN

---

Jüri Üyesi

Yrd.Doç.Dr. İbrahim TÜRKYILMAZ

---

Jüri Üyesi

Prof.Dr. Mehmet Emin ÖZEL

---

Müdür

Fen Bilimleri Enstitüsü

## TEŞEKKÜR

Öncelikle bu çalışmanın ortaya çıkmasında bilgi ve deneyim desteklerini esirgemeyen danışmanım Yrd.Doç.Dr. İsmail KADAYIF'a saygı ve teşekkürlerimi sunuyorum.

Maddi ve manevi destekleri ile beni bugünlere getiren aileme sonsuz sevgilerimi iletiyorum.

Tez ve proje çalışmalarımda gerek bilgi ve tecrübeleriyle gerekse manevi destekleriyle her zaman yanımda olan çalışma arkadaşlarım en başta Yrd.Doç.Dr. Adnan ÇEVİK, Uzman Banu TEKİN, Uzman İsmail İŞLEYEN ve Bilgi İşlem Dairesi çalışanlarına teşekkür ediyorum.

Yücel ŞAHİN

2008

## SİMGELER VE KISALTMALAR

<b>AB</b>	: Avrupa Birliđi
<b>ACVS</b>	: Academic CV System
<b>AÖBS</b>	: Akademik Özgeçmiş Bilgi Sistemi
<b>AKTS</b>	: Avrupa Kredi Transfer Sistemi
<b>ECTS</b>	: European Credit Transfer System
<b>AKTS DKS</b>	: AKTS Ders Katalog Sistemi
<b>ECTS CCS</b>	: ECTS Course Catalogue System
<b>BT</b>	: Bilişim Teknolojisi
<b>ASP</b>	: Active Server Pages
<b>AJAX</b>	: Asynchronous JavaScript and XML
<b>CSS</b>	: Cascading Style Sheet
<b>DB</b>	: Database
<b>DNS</b>	: Domain Name Server
<b>HTML</b>	: Hyper Text Markup Language
<b>IIS</b>	: Microsoft Internet Information Server
<b>PEAR</b>	: PHP Extension and Application Repository
<b>PHP</b>	: Hypertext Preprocessor
<b>PWS</b>	: Personal Web Server
<b>URL</b>	: Unified Resources Locator
<b>YAGE</b>	: Yukarda adı geçen eser
<b>YÖK</b>	: Yüksek Öğretim Kurumu
<b>XML</b>	: Extensible Markup Language
<b>LDAP</b>	: Lightweight Directory Access Protocol
<b>IMAP</b>	: Internet Message Access Protocol
<b>SNMP</b>	: Simple Network Management Protocol
<b>NNTP</b>	: Network News Transfer Protocol
<b>POP3</b>	: Post Office Protocol
<b>HTTP</b>	: Hypertext Transfer Protocol
<b>ODBC</b>	: Open Database Connectivity
<b>SQL</b>	: Structured Query Language

# **ACADEMIC CV SYSTEM FOR THE UNIVERSITIES IN ACCORDANCE WITH THE REGULATIONS OF EU BOLOGNA PROCESS**

## **ABSTRACT**

This study aims to develop, design and configure two web-based databases which store the CVs of academic personnel and provides information about contents and other details of courses taught at the universities. Although Canakkale Onsekiz Mart University is chosen to implement the project, the system can be applicable to all universities in general. Academic CV System (ACVS) and European Credit Transfer System Course Catalogue System (ECTS CCS) databases contain complete user and administration/management interfaces.

Databases and their essential parts, such as administration applications, support the decision making process/mechanisms by transforming the data and/or records into accurate and up-to-date information in a very short time. Taking into consideration that in today's world, the success is mostly depended on personnel's abilities and skills; and specially at the universities, well-educated and experienced academic personnel is the most important asset to improve the scientific quality; the necessity of up-to-date and easily manageable CV system is obvious. The database will be multi-functional and include search/management system and will be accessible by all users.

Apart from this general need, under the EU Bologna Process, the universities are required to provide information about "what, how and who" to improve the quality and transparency in education. As a result of this study, ECTS Course Catalogue System and Academic CV System will provide up-to-date, accurate and fast answers to the questions "what, how and who" and the answers will be presented to the decision makers and other users as an information.

**Keywords** : Internet, database, Academic CV System, ECTS, AB Bologna, Web

# AB BOLOGNA SÜRECİNDE ÜNİVERSİTELER İÇİN WEB TABANLI AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ

## Özet

Bu çalışma, genelde üniversiteler, özelde Çanakkale Onsekiz Mart Üniversitesi akademik personeli özgeçmiş verilerinin ve üniversite bünyesinde okutulan tüm derslerle ilgili içerik bilgilerinin veritabanı ortamına aktarılması ve verilerin kullanımı amaçlarıyla, kullanıcı ve yönetim arayüzlerini kapsayan web tabanlı iki uygulamanın AKTS Ders Katalog Sistemi (AKTS DKS) ve Akademik Özgeçmiş Bilgi Sistemi (AÖBS) tasarlanması, kurulumu ve geliştirilmesi üzerinedir.

Veritabanı ve ayrılmaz parçaları sorgulama ve yönetim uygulamaları, veri ve/veya kayıtların çok kısa zamanda, doğru ve güncel olarak karar alma mekanizmalarına yardımcı olacak biçimde enformasyona dönüştürülmesini kolaylaştırmaktadır. Günümüzde başarının kesinlikle insan odaklı olduğu, üniversitelerin yine yetişmiş akademik personel ile bilimsel kalitelerini arttırdığı göz önüne alınacak olursa, yoğun akademik çalışmaların gerçekleştirildiği üniversitelerde özgeçmiş sistemlerinin sürekli güncel, akademik personel tarafından yönetilebilir ve tüm kullanıcılar tarafından kullanılabilir çoklu-faydalı bir veritabanı ve sorgu/yönetim sistemine sahip olması gerektiği açıktır. Tüm bu genel ihtiyacın yanında Avrupa Birliği Bologna Süreci kapsamında üniversitelerin “neyi, nasıl ve kiminle” yaptıkları sorularına verilecek yanıtlar eğitimde kalite için bir zorunluluktur. Çanakkale Onsekiz Mart Üniversitesi baz alınarak yapılan bu çalışmada AKTS DKS ve AÖBS projeleri ile her üç soru birbiriyle ilişkili olarak güncel, doğru ve hızlı bir biçimde yanıtlanacak ve yanıtlar kullanıcılar ile karar verme mekanizmalarına enformasyon olarak sunulmuş olacaktır.

**Anahtar sözcükler:** İnternet, veritabanı, Akademik Özgeçmiş Sistemi, ECTS, AB Bologna, Web



YÜKSEK LİSANS TEZİ SINAV SONUÇ BELGESİ .....	ii
TEŞEKKÜR .....	iii
SİMGELER VE KISALTMALAR .....	iv
ABSTRACT .....	v
ÖZET .....	vi
<b>BÖLÜM 1 – GİRİŞ .....</b>	<b>1</b>
<b>BÖLÜM 2 – KULLANILAN TEKNOLOJİLER .....</b>	<b>3</b>
2.1 PHP .....	4
2.2 MYSQL.....	6
2.3 PEAR .....	10
2.4 AJAX.....	12
<b>BÖLÜM 3 – AKTS DERS KATALOG SİSTEMİ.....</b>	<b>15</b>
3.1 İşletme Politikası.....	16
3.2 Sistemin Veri Tabanı Yapısı.....	17
3.3 Arayüzler ve Kullanıcılar .....	17
<b>BÖLÜM 4 – AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ .....</b>	<b>20</b>
4.1 Üniversitemizde Akademik Özgeçmiş Verilerinde Önceki Durum ....	20
4.2 Akademik Özgeçmiş Bilgi Sisteminin Getireceği Kazanımlar .....	22
4.3 Veri Tabanı .....	24
4.3.1 Tablolar ve Yapıları.....	27
4.3.2 Veri Yapıları.....	28
4.3.3 Tablolar Arası İlişki Yapısı.....	30
4.4 Sistemdeki Güvenlik İşlemleri.....	31
4.4.1 Session (Oturum) Kullanımı .....	31
4.4.2 Veri Şifreleme Yöntemi .....	32
4.4.3 Ağ Üzerinde Şifreli Veri Gönderimi .....	34

4.4.4 Otomatik Form Anahtarlama.....	34
4.4.5 Veri İşlemlerinde Form Anahtarı ve Oturum Kontrolü .....	36
4.5 Sistem Hazır Değerleri .....	37
4.6 Sınıf (Class) Yapılarının Kullanımı .....	38
4.7 Sistemin Ana Bölümleri .....	39
4.7.1 Veri Giriş ve Yönetim.....	39
4.7.1.1 Akademik Özgeçmiş Ana Konu Başlıkları.....	40
4.7.1.2 Kullanıcı Tarafı Form Etkileşimi ve Hata Kontrolleri .....	43
4.7.1.3 Veri Görüntüleme .....	44
4.7.1.4 Kişisel Ayarlar.....	45
4.7.1.5 Fotoğraf Yükleme .....	45
4.7.2 Sistem Yönetim.....	48
4.7.2.1 Yetkilendirmeler .....	48
4.7.2.2 Kullanıcı İşlemleri .....	48
4.7.2.3 Veri Yedekleme .....	49
4.7.3 Arama ve Görüntüleme.....	49
4.7.3.1 Hızlı Arama .....	49
4.7.3.2 Gelişmiş Arama.....	51
4.7.3.3 Akademik Etkinlik Arama.....	51
4.7.4 İstatistikler.....	52
4.8 Sistemde Dil Desteği .....	52
4.8.1 Dil Dosyaları.....	52
4.8.2 Sistemde Dil Dosyaları Kullanımı .....	53
4.8.3 Yeni Dil Eklenmesi.....	53
4.9 AÖBS ve AKTS DKS Etkileşimi .....	54
<b>BÖLÜM 5 – SONUÇ VE TARTIŞMA .....</b>	<b>55</b>
<b>KAYNAKLAR .....</b>	<b>I</b>
<b>Çizelgeler .....</b>	<b>IV</b>
<b>Şekiller .....</b>	<b>V</b>
<b>Yaşam Öyküsü.....</b>	<b>VI</b>

# BÖLÜM 1

## GİRİŞ

1999'da yayınlanan, 29 Avrupa ülkesinde Eğitim Bakanları tarafından onaylanan Bologna Bildirisi'ne göre, yüksek öğrenimde Avrupa standardı oluşturulmaya çalışılması ve kalite güvencesi konusunda işbirliği yapılması kararlaştırılmıştır. Bu çerçevede, Diploma Eki oluşturulması ve Avrupa Kredi Transfer Sistemi/European Credit Transfer System (AKTS/ECTS) oluşturulmasına karar verilmiştir. YÖK tarafından üniversitelerde uygulanması istenen Bologna süreci işlemleri kapsamında, üniversitemiz tarafından Diploma Eki oluşturulması ve Avrupa Kredi Transfer Sistemi için gerekli olan bilgi paketinin, web sitesinin oluşturulması konusunda çalışmalar başlatılmıştır. Üniversitemizde bu kapsamda AB programlarını tanıtan İngilizce web sitesi hazırlanmış, sitede programlar ve prosedürler ayrıntılı olarak tanıtılmış ve ayrıca ÇOMÜ, Türkiye ve Çanakkale hakkında bilgilere yer verilmiştir. Bunlarında yanında üniversitemizde tüm bu çalışmalara ek olarak bazı yeni projelerin de gerçekleştirilmesi gerekmektedir.

Bu gereksinimlerden biri Avrupa Kredi Transfer Sistemi için internet ortamında bir ders veri tabanı oluşturulmasıdır (AKTS Ders Katalog Sistemi/ECTS Course Catalogue System (AKTS DKS/ECTS CCS)). Bu ders katalogunda öğrencilerin mümkün olan en güncel ve en doğru bilgiye en kısa zamanda ulaşmaları amaçlanmaktadır. Her akademik birimdeki dersler ile ilgili ders içerikleri, kaynak kitaplar, öğretim metotları, değerlendirme yöntemleri, haftalık saatleri, ÇOMÜ ve AKTS kredileri gibi bilgiler yer almalıdır. Ayrıca bu dersler ile ilgili olarak uluslararası alanda tüm ilgili öğrenci veya kişilerle etkileşimin en hızlı ve sistemli bir şekilde gerçekleştirilmesi de şarttır.

Avrupa Kredi Transfer Sistemi ile bağlantılı olarak Akademik Özgeçmiş Bilgi Sistemi/Academic CV System (AÖBS/ACVS) veri tabanı yapılması gerekliliği duyulmuştur. Ders Katalog Sistemi ile neyi nasıl sorularına yanıt bulunulmaya çalışılmaktadır. Ancak kiminle sorusu yanıtız kalmaktadır. AKTS bilgi paketine göre, öğrencinin dersi veren öğretim elemanı hakkında bilgi sahibi olması gerekliliği

belirtilmektedir. AKTS DKS ile etkileşimli bir şekilde çalışan Akademik Özgeçmiş Bilgi Sistemi, öğretim elemanlarının özgeçmişlerinin güncel olmasını ve ilgili dersle ilgilenen kişiler tarafından anında görüntülenebilmesini hedeflemektedir.

AKTS DKS ile öğretim programları şeffaf hale getirilirken, akademik özgeçmiş sistemi bu programları uygulayacak kişileri tanıtmayı amaçlar. Bunun doğal sonucu, Bologna sürecinin yukarıda sözü edilen üç basamağının gerçekleşmesine önemli katkıdır. Dahası, AÖBS tam anlamıyla çalıştığı zaman yine Bologna sürecinin bir başka vazgeçilmezi “stratejik planlama” için önemli veriler sağlayacaktır. Böylece, AÖBS’ni Bologna süreci için temel destek sistemlerinden biri olarak adlandırmak mümkündür.

Üniversitemizde uygulanacak bu projeler sayesinde, neyi nasıl kiminle sorularına verilecek yanıtlar ile, Avrupa Birliği ülkelerinin eğitimde kalite güvencesi konusundaki şeffaflık arayışlarına karşılık verilmiş olacaktır.

Bu tezin ilerleyen bölümleri şu şekilde düzenlenmiştir: Bölüm 2’de AKTS DKS ve AÖBS yapımında kullanılan internet teknolojileri ve bu teknolojilerin kullanılma sebeplerine kısaca değinilmiştir. Bölüm 3’de AKTS DKS ile ilgili genel bilgilere değinilmiştir. Bölüm 4’de AÖBS yapısı, geliştirilme yöntemleri ve işleyiş yapısı ile AKTS DKS ve AÖBS arasındaki etkileşime değinilmiştir. Sonuçların değerlendirilmesine ise Bölüm 6’da değinilmiştir.

## BÖLÜM 2

### KULLANILAN TEKNOLOJİLER

AKTS DKS ve AÖBS internet tabanlı uygulamalardır. Sistemlerin çalışması için kullanılacak sunucu, teknik özellikleri ve ihtiyaçlar bakımından farklılık gösterebilir. Ancak sunucu makine üzerinde çalışacak web sunucu yazılımları, bu yazılımlar üzerinde çalışacak uygulamaların istek ve ihtiyaçlarını karşılar nitelikte olmalıdır. Ek bir lisans ihtiyacı ve masrafı gerektirmeme, kullanılabilirlik, yaygın kullanım ve destek, performans ve uygulama geliştirmeye dair diğer artılar sistemin kurulumu öncesinde düşünülmesi gereklidir. Üniversitemizde mevcut web sunucu makineleri üzerinde Linux işletim sistemi ve Apache web sunucuları hizmet vermektedir ve bu sunucular uygulama alanı olarak kullanılmıştır. İki sistemde de genel olarak bilinen, web yazılım literatürünün oldukça geniş bir bölümünü oluşturan ve web uygulamalarının temeli olan HTML, JavaScript ve CSS kullanılmıştır Pfaffenberger (2004) ve Zakas (2005) kaynaklarında bu teknolojilere detaylı değinilmiştir.

Gosney (2003)'e göre 430 dan fazla üye organizasyonu bulunan W3C (World Wide Web Consortium) kendini, Web geliştirmelerini, tüm muhtemel teknik gereksinimlerini, ortak üretim hedefleriyle protokollerini ve Web'in tüm taraflarına karşı doğru iş birliktelik için fonksiyonel altyapıyı sağlamaya adanmıştır. W3C ile ilgili bilgilere kendi web sitesinden de erişilebilir (<http://www.w3c.org>).

HTML'nin tarihine bakıldığında basit dört iterasyon yoluyla geçmişten günümüze geldiği görülür (Gosney, 2003).

**HTML 2.0.** : İlk standartları ve çekirdek gelişmeleri barındıran versiyondur.

**HTML 3.2.** : W3C nin üzerinde ilk çalıştığı versiyondur. Superscript, subscript, tables gibi popüler gelişmeleri içerir.

**HTML 4.0.** : Programcıları için altın değerinde gelişmeler bu versiyonda yer aldı ve daha sonra **HTML 4.0.1.** yerini aldı.

Bunlar dışında sistemlerin yapımında kullanılan ve sunucu üzerinde çalışan diğer web teknolojilerine aşağıda kısaca değinilmiştir.

## 2.1 PHP

PHP'nin açılımı "Hypertext Preprocessor" dır. Tüm dünyada web geliştiricileri tarafından yaygın olarak kullanılan bir betik (script) dilidir. Sunucu taraflı uygulama geliştirme, komut satırı uygulamaları, istemci taraflı uygulamalar ve daha farklı uygulamalar PHP ile yapılabilmektedir. Bir çok işletim sisteminde (Linux ve Unix türevleri, MS Windows, Mac OS X, RISC OS) kullanılabilir ve web sunucuların büyük kısmında (Apache, IIS, PWS, iPlanet vd.) modül olarak bulunur ve desteklenir. Bunun dışında PHP (MySQL , Oracle (OCI7 ve OCI8), dBase, InterBase, ODBC, Unix dbm, FrontBase, PostgreSQL, Informix vd.) veritabanları desteğine sahiptir. LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (Windows platformunda) ve daha birçok protokolü kullanarak diğer hizmetler ile konuşabilir. PHP betikleri sunucu tarafından çalıştırılır (JavaScript dilinin tam tersine) ve çıktısı tarayıcıya gönderilir. PHP ifadeleri ve fonksiyonları HTML dokümanları içine yazılabilir ve çalıştırılabilir. Böylelikle dinamik web sayfaları oluşturulabilir. Web sunucusu yazılımı web sayfası içerisinde PHP diline ait ifadeleri gördüğünde, bunları yorumlayıp, çıktısını HTML ifadeleri yazılı yerlerin arasına yerleştirir (Achour, ve arkadaşları, 2007).

PHP'nin tarihçesine bakıldığında Apache ekibinin üyesi yazılım mühendisi Rasmus Lerdorf tarafından bulunduğu ve Lerdorf'un PHP'nin ardındaki gerçek itici güç olduğu görülür. Lerdorf'un PHP'nin ilk bölümünü kendi kişisel kullanımı için 1994 yılının sonlarında geliştirdiği bildirilmiştir. Oluşturduğu sayfasının ziyaretçilerini izlemek için kullanmıştır. Bir sonraki yıl, yaptığı sayfayı gören veya duyanların isteği üzerine Personal Home Page Tools (Kişisel Sayfa Araçları) adı altında bir paketi hazırladı. İkinci Versiyon PHP/FI adı altında ve bir form yorumlayıcısı ve SQL sorgularını yorumlayan bir araç içerecek şekilde hazırlandı. 1997 yılının ortalarında PHP dünya çapında ortalama 50.000 sitede kullanılmaktaydı. Daha sonra sadece Rasmus'un geliştirebileceğinin ötesine geçen PHP, tüm dünyadan

kullanıcı ve geliştiricilerin katılımıyla açık kaynak “yardımsever grup (benevolent junta)” modeliyle bir geliştirme ekibi oluşturuldu. PHP3 ve PHP4 yorumlayıcılarını geliştiren İsraili programcılar Zeev Suraski ve Andi Gutmans ayrıca çalışmalarını Zend Şirketinin bir bölümü altında geliştirdi ve genellediler. 1998 yılının ikinci çeyreğinden itibaren PHP patlaması yaşandı. En iyi tahminle Ekim 1998 yılında 100.000 farklı alan adında (domain) bir şekilde kullanılmaya başlandı. Sadece bir yıl sonra PHP 10 milyon alan adı rakamını aştı (Converse, 2004).

PHP'nin bilinen en büyük destekleyicisi Zend isimli kuruluştur. Komut ve uygulama geliştirme için kullanılan Zend Studio, PHP kodlarının şifrelenmesi ve korunmasına yönelik Zend SafeGuard, PHP yorumlama hızını artırmaya ve diğer Zend ürünlerini çalıştırmaya yönelik Zend optimizer motoru kuruluş ürünlerinden birkaçıdır.

Sistemlerin geliştirilirken PHP'nin tercih edilmesinde PHP'nin şu avantajları etkili olmuştur:

- Platform çeşitliliği (UNIX, Linux, Solaris, HP-UX, IRIX, FreeBSD, Windows vb.),
- Performans (Zend optimizer motoru ile artmıştır),
- Büyük ölçekli veri tabanı uygulamaları için ideal (ODBC, MySQL, PostgreSQL, Oracle ve diğerleri için arayüz desteği),
- İnternet standartlarına uyumluluk (HTTP, LDAP, IMAP, FTP gibi standartlara uyum),
- Gelişmiş özellikler (oturum yönetimi, semafor ve paylaşımlı hafıza kullanım, sürekli veritabanı bağlantıları, regular expressions),
- GPL (General Public License) ile ücretsiz dağıtılmaktadır.

## 2.2 MYSQL

MySQL'in tarihçesi, yaratıcısı Monty Widenius'un bir İsviçre BT firmasında çalıştığı zaman olan 1979 yılına kadar uzanır. 1994 yılında aynı firma Web uygulamaları üstüne çalışmalarını arttırmıştır. Bunun üzerine David Hughes tarafından geliştirilen mSQL üzerinde çalışılmıştır. Böylece fikir olarak ilk MySQL doğmuş oldu. 1995 yılında ilk versiyon üretilmiştir. Bir yıl sonrasında Monty'nin çalıştığı firma adını MySQL AB adı altında geliştirmiştir. On yıl içinde MySQL dünya çapında bir ürün haline gelmiştir. Günümüzde MySQL ticari olarak üretilen Oracle ve Informix gibi veritabanı uygulamalarıyla rekabet edecek seviyeye ulaşmıştır. Hatta 4.x ve üzeri versiyonlarında çok aranan ikinci anahtar (foreign key) ve hareket yönetimi (transaction) özellikleri eklenmiştir. Bu özellikleriyle MySQL dünyanın en aranan açık kaynak veritabanı uygulaması haline gelmiştir (Converse, 2004).

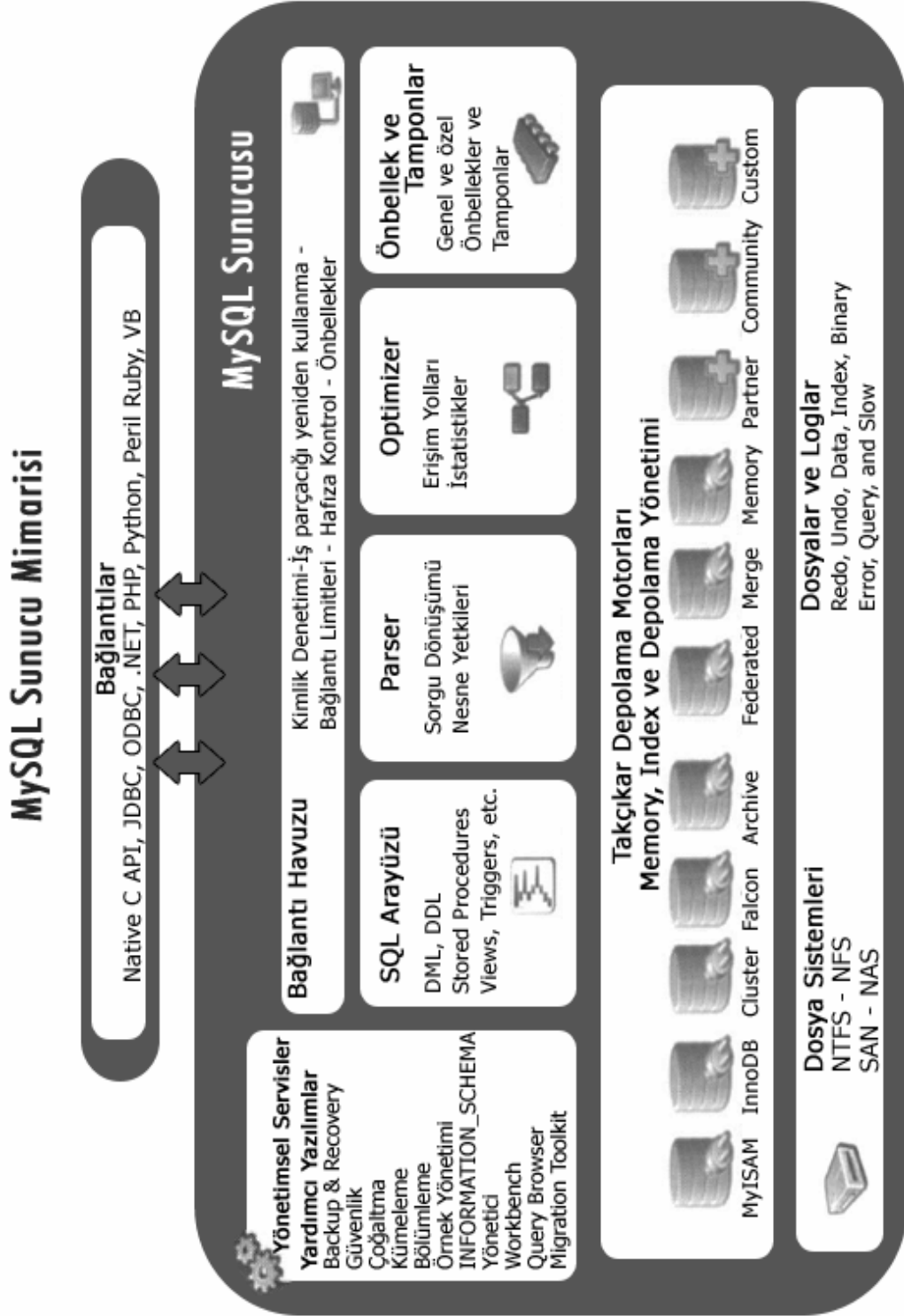
MySQL çok popüler ve yaygınlaşmış gelişmiş bir açık kaynak SQL (Open Source SQL) veritabanı yönetim sistemidir. MySQL geliştiricileri tarafından kurulmuş MySQL AB isimli ticari bir kuruluş tarafından desteklenir. C ve C++ ile yazılmıştır. MySQL ve MySQL AB ile ilgili detaylı bilgiye <http://www.mysql.com> adresinden ulaşılabilir (Anonymous, 2007). MySQL'in temel özelliklerine kaynağa göre maddeler halinde şöyle değinebiliriz:

- MySQL ilişkisel bir veritabanı yönetim sistemidir: Bir ilişkisel veritabanı verileri tek büyük bir tabloda tutmak yerine bunları ayrı tablolarda tutarak işlemlerde hız ve esneklik sağlar. SQL Yapısal Sorgulama Dili (Structured Query Language) demektir. ANSI/ISO SQL standardı ile tanımlanır.
- MySQL açık kaynaklı bir yazılımdır: MySQL yazılımı GPL (GNU General Public Licence) (GNU Genel Halk Lisansı) lisansını kullanmaktadır (<http://www.fsf.org/licenses/>). Bu lisans yazılım ile farklı durumlarda neler yapılıp yapılamayacağını tanımlar.



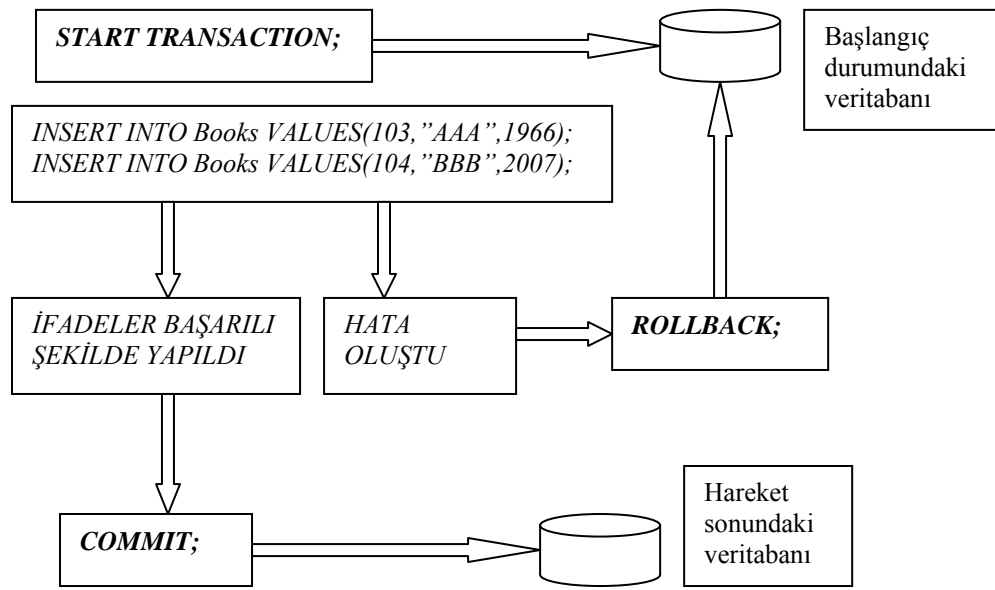
- MySQL Sunucusu mevcut çözümlerden daha hızlı bir şekilde çok büyük boyutlu veritabanları ile çalışmak için geliştirildi ve uzun süredir de başarılı şekilde kullanılmaktadır. MySQL sunucusu günümüzde çok zengin ve kullanışlı fonksiyonlar sunmaktadır. İlişkisel, hızı ve güvenliği MySQL sunucusunun internet üzerinden erişilmek için son derece uygun olmasını sağlamaktadır.
- MySQL Server istemci/sunucu (client/server) veya gömülü sistemlerde (embedded system) çalışır. MySQL veritabanı yazılımı bir istemci/sunucu sistemidir. Bu sistem farklı istemcilere hizmet verebilen bir çok iş parçacıklı (multi-threaded) SQL sunucusundan, yönetimsel araçlardan ve bir geniş ölçekli uygulama programlama arayüzleri (API) içermektedir. Ayrıca MySQL sunucusunda gömülü çok iş parçacıklı çalışabilen bir kütüphane bulunmaktadır. Bu şekilde küçük, hızlı ve kolay yönetilebilen tek başına çalışan uygulamalar geliştirilebilir.
- İşletim sisteminin çekirdek iş parçacıklarını (kernel thread) kullanan çok iş parçacıklı çalışan bellek yer ayırma sistemine sahip bir yazılımdır. Eğer varsa kolaylıkla birden fazla işlemciyi kullanabilir.
- MySQL kodu Purify isimli ticari bir bellek sızıntısı test eden program ve açık kaynaklı Valgrind ile test edilir.

MySQL sunucu mimari yapısı Şekil 2.1'deki gibidir.



Şekil 2.1 : MySQL Sunucu Mimarisi (Zoratti, 2006)

MySQL veritabanı InnoDB ve BDB tablo tiplerinde hareket (transaction) yönetimini destekler. Diğer tüm tablo tiplerinde hareket (transaction) desteklenmez. InnoDB tablo tipi hareket yönetimi için en ideal olanıdır. Hareketi kullanabilmek için `START TRANSACTION`, `COMMIT` ve `ROLLBACK` ifadelerini içerir. `START TRANSACTION` bir hareketi başlatır. `COMMIT` ifadesi yapılan SQL ifadelerinin gerçekten veritabanına işlenmesini sağlar. `ROLLBACK` ifadesi yapılan işlemlerin iptal edilmesini ve veritabanının hareketten önceki haline geri gelmesini sağlar (Sheldon, 2005). Kaynağa da dayanarak bu durumları aşağıdaki gibi ifade edebiliriz:



**Şekil 2.2 :** MySQL’de hareket yönetimi

Hareketler (transactions) veri bütünlüğünün ve doğruluğunun sağlanması amacıyla kullanılırlar. Hareket ilişkisel veritabanlarında birden fazla birbiri ile ilişkili olan SQL ifadelerinin grubu anlamına gelmektedir. Bu gruptaki SQL ifadelerinin ya hepsi doğru şekilde gerçekleştirilir ya da hepsi birden yapılmaz (Sheldon, 2005).

## 2.3 PEAR

PEAR kelime olarak “PHP Extension and Application Repository” (PHP eklentisi ve uygulama deposu) kısaltılmasıdır. 1999 yılında Tig S. Bakken tarafından geliştirilmeye başlatılmıştır ve birçok serbest yazılım gönüllüsünün desteği ile gelişerek günümüze kadar gelmiştir. Ayrıca 2003 yılında PHP Community tarafından kabul görmüştür (Anonymous, 2004). Geniş çaplı, farklı fonksiyonlar için ücretsiz PHP modülleri ve sınıflarından oluşan çevrimiçi (online) depodur. XML ayrıştırma (parsing), veritabanı işlemleri, kullanıcı kimlik denetimi ve bunlar gibi birçok amaç için kullanılabilen birçok sınıf ve modülden oluşur. Birçok durumda, bu modüller geliştiricinin çok karmaşık işlemleri kolay bir arayüz yardımıyla hızlı ve kolay bir şekilde gerçekleştirmesini sağlar. Bu işleme soyutlama (abstraction) denilir ve kullanılan arayüz (interface) modülüne soyutlama katmanı (abstraction layer) denilir. PHP nin 4.0.4 yada daha üst versiyonları ile birlikte kullanılmak üzere tasarlanmıştır (Naramore, 2005). PEAR’ın genel amaçlarını şöyle sıralayabiliriz:

- PHP kullanıcıları için açık kaynak kodlu kütüphaneler oluşturmak,
- Yazılmış PHP kodları için standartlar getirmek,
- Kod dağıtımı ve paket yönetimi için bir sistem oluşturmak,
- PHP kullanıcıları ve geliştiricileri için ortak bir iletişim platformu oluşturmak.

PEAR için birçok modül bulunmaktadır. Bu modüllere paketler (packages) denilir ve paketlere “<http://pear.php.net/packages.php>” adresinden ulaşılabilir. Paketler hem HTML hem de XML arayüzler ile açık kaynak kodlu olarak bulunmaktadırlar. Paketler PHP geliştiricileri tarafından belirli kodlama standartlarına bağlı kalınarak yazılırlar. Her geliştirilen paket dokümantasyon ve bağımlılık listesi içerir. Paketler bir ağaç üzerinde birer düğüm olarak görüntülenir. Örneğin; XML\_Parser düğümü (node) diğer XML paketleri ile birlikte paketlenmiştir. Bu grupta sadece organizasyonu sağlamak içindir. Her PHP geliştiricisi bir paket geliştirip bu paketi PEAR paket ağacına ekleyebilir.

PEAR paketleri kategoriler halinde toplanmıştır. Bu kategoriler aşağıdaki gibidir (Valade, 2004).

- Kimlik Denetleme (Authentication)
- Karşılaştırmalı Test (Benchmarking)
- Ön Bellek (Caching)
- Ayar Konsolu (Configuration Console)
- Veri Tabanı (Database)
- Tarih Zaman(Date and Time)
- Şifreleme (Encryption)
- Dosya Formatları (File Formats)
- Dosya Sistemi (File System)
- Grafik Bileşenleri (Gtk Components)
- HTML
- HTTP
- Görüntü (Images)
- Uluslararasılaştırma (Internationalization)
- Günlük Tutma (Logging)
- Web Servisleri (Web Services)
- Mail
- Matematik (Math)
- Bilgisayar Ağı (Networking)
- Rakamlar (Numbers)
- Ödeme (Payment)
- PEAR
- PHP
- İşleme (Processing)
- Bilim (Science)
- Akımlar (Streams)
- Yapılar (Structures)
- Metin (Text)
- Araçlar ve Yardımcı Programlar (Tools and Utilities)
- XML

## 2.4 AJAX

IEEE Software editörü Christof Ebert'in belirttiği gibi; tüm sayfayı yeniden yüklemeyen içeriğin değiştirilmesine olanak veren asenkron web sayfaları son zamanlarda popüler olmuştur. Bu sayfalar, yerleştirilmesi, erişimi ve tarayıcının "geri" tuş özelliği ile karmaşıklık yaratmasına rağmen daha esnek bir içerik sunumu sağlar. Böylece sayfanın yeniden yüklenme performansının artışına neden olur. Ajax bu gibi web sayfalarının yaratılması için çoğunlukla kullanılır (Serrano ve Aroztegi, 2007).

Bilgisayar ağları günümüzde, tek başına duran bilgisayarların ötesinde, genel çalışma alanı haline gelmiş ve böylece geçmişin masaüstü uygulamalarının web uygulaması haline dönüşmesi hızlanmıştır. İşte bu noktada Ajax devreye girmektedir. Yukarıdaki kaynağa göre Jesse James Garret, Ajax terimini (Asynchronous Java Script and XML) 2005 yılında ortaya attı. Geliştiriciler bir süredir Web sayfalarını ve uygulamalarını daha dinamik hale getirmek için çalışıyordu. Özellikle bu çalışmalar HTML, World Wide Web Consortium's Document Object Model, betik dilleri (scripting languages) ve XMLHttpRequest nesnesini içermekteydi. Betik dilleri DOM'u yönetirken web uygulamalarının HTML içerik, yapı ve biçimine dinamik erişimine ve bunları güncellemesine izin verir. Web uygulamalarına HTTP istek-yanıt davranışı sırasında yol veren XMLHttpRequest, tüm sayfayı yeniden yüklemeksizin tarayıcı ve sunucu arasındaki iletişimi destekler. Geliştiriciler gizli çerçeve (hidden frame) gibi bunu sağlayan farklı yollar bulmuş olmasına rağmen, incelikli çalışmalar için bu kodlar yazılmayacak kadar güç ve karmaşıktır.

Yine yukarı adı geçen esere göre 2005'ten bu yana Google Maps, Netvibes ve Zimbra Collaboration Suite gibi geliştiriciler Ajax'ı Web uygulamalarında kullanmaktadırlar. Kullanıcılar masaüstü uygulamalarda olduğu kadar ve hatta belki daha iyi şekilde bu uygulamalarla etkileşime girebilmektedirler. Fakat Ajax'ın özellikleri bu uygulamalarda ayrıntılı kodlama gerektirir.

Ajax paketleri (frameworks) kullanıcının talep ettiği Ajax özelliklerini kapsayan uygulamaları geliştirmeyi ve oluşturmayı kolaylaştıran araçlardır. Çok çeşitli türde Ajax paketi vardır. Bazıları kullanıcı ara yüzünün kontrolünü arttırmaya ve diğerleri sunucu fonksiyonlarına daha basit erişim sağlamaya odaklıdır (yage).

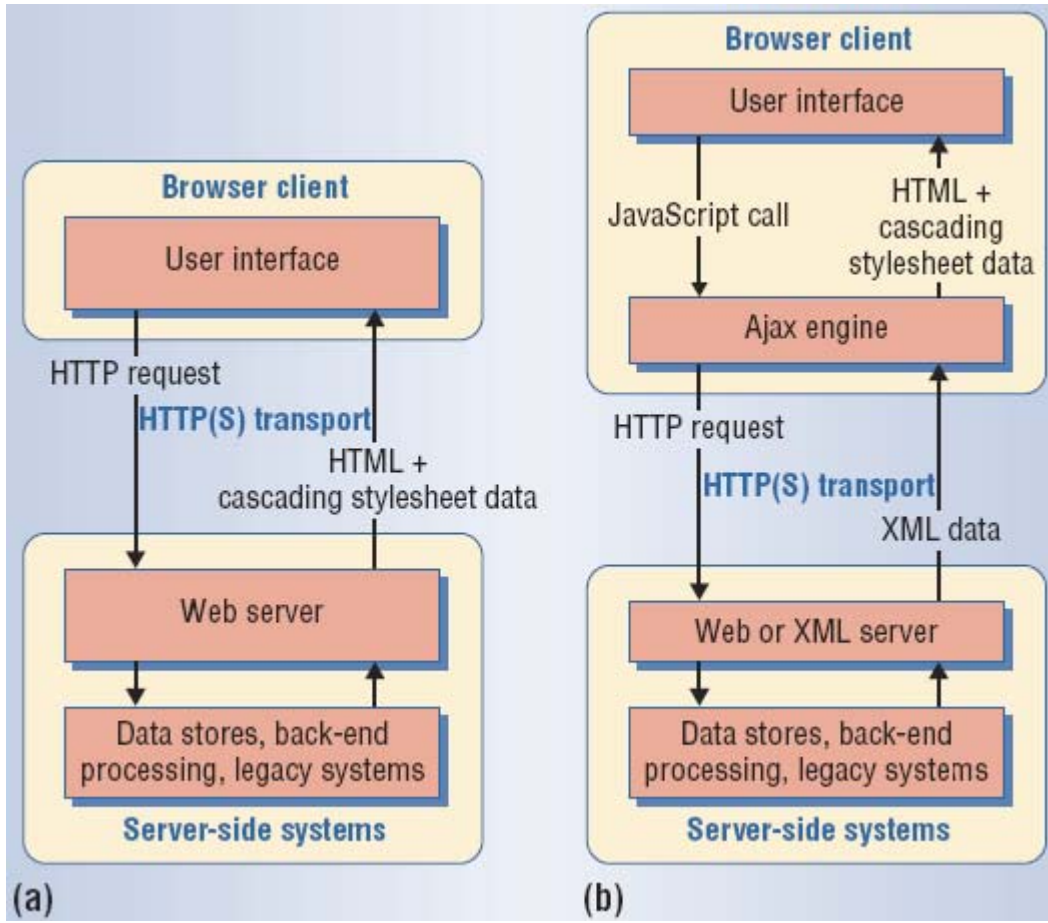
Geleneksel tarayıcı tabanlı web uygulamaları kullanıcının sunucuya bir istek göndermesi ve sunucunun bu isteği işlemesini ve yanıt oluşturmasını ve sonra tarayıcının sonuçları arayüzde güncellemesini beklemeyi gerektirir. Bu istek-yanıt bekle-bekle yapısı kesinlikle iletişimi aksatıcı ve verimliliği azaltıcı niteliktedir (Smith, 2006).

Aşırı ağ gecikmesi, arayüz karmaşıklığı ve yavaş sunucu yanıt vericiliği (server response) kullanıcı deneyimini olumsuz etkiler ve müşterinin mutluluğunu azaltır, ziyaretçilerin web sitesine daha az sıklıkla ve az süreyle girmesine yol açar. Ayrıca kesinlikle e-ticaretin sürekliliğini baltalar. Asenkron JavaScript ve XML (AJAX) web tabanlı uygulamaların yanıt vericiliğini, etkileşimliliğini ve ayarlanabilirliğini gerçekleştirmek (kısaca diğer masaüstü uygulamalarının kesintisiz kullanıcı deneyimini yeniden yaratmak) için tasarlanmış standart tabanlı bir programlama tekniğidir (Smith, 2006).

Şubat 2005 tarihinde “Ajax: A New Approach to web Applications” başlıklı makalesinde Jesse James Garrett Ajax kullanılarak inşa edilen uygulamaların beş anahtar karakterini listelemiştir (Smith, 2006):

1. DHTML (Dynamic Hypertext Markup Language) gibi açık standart ile yapılandırılmış bir kullanıcı arayüzü,
2. Dokuman Nesne Modeli (DOM) ile gerçekleştirilen dinamik, etkileşimli kullanıcı deneyimi,
3. XML (Extensible Markup Language) ve ESLT (Extensible Stylesheet Language Transformations) kullanılarak veri değişimi ve transformasyonu,
4. XMLHttpRequest yoluyla asenkron istemci/sunucu iletişimi,
5. Tüm bileşenleri bir araya getiren ortak dil olarak JavaScript.

AJAX'ın uygulama çalışma mantığına ait bir gösterim aşağıdadır.



**Şekil 2.3:** AJAX uygulama çalışma mantığı: Geleneksel bir Web uygulamasında (a) kullanıcı hareketleri (actions) web sunucusuna gönderilen bir HTTP isteği (request) tetikler. Bu işlem sonucunda sunucu bu isteğe uygun bir web sayfasını istemciye geri döndürür. Ek istekler sayfa güncelleninceye kadar uygulamayı kilitler. (b) Ajax uygulamaları tarayıcıda çalışan bir JavaScript temelli motor (engine) oluştururlar. Motor kullanıcı isteklerinin yolunu keser, istenilen materyali gösterir ve istemci tarafında gerçekleşen birçok reaksiyon ile ilgilenir. Eğer motor daha fazla bilgiye ihtiyaç duyarsa, arka planda sunucudan ihtiyaç duyduğu veriyi ister. Bu isteme aşamasında uygulama kilitlemez ve kullanıcı uygulama ile etkileşimde bulunmaya devam eder (Paulson, 2005).



### BÖLÜM 3

#### AKTS DERS KATALOG SİSTEMİ

1999'da yayınlanan ve 29 Avrupa ülkesinde Eğitim Bakanları seviyesinde onaylanan Bologna Bildirisi'nde, Avrupa Yüksek Öğretim Sistemi'nin uluslararası alanda rekabet edebilir bir düzeye yükseltilmesi, 2010'da Avrupa Yüksek Öğrenim Alanı'nın oluşturulması, bu süreçte iki yılda bir gelişmelerin ülkeler seviyesinde ve Avrupa genelinde izlenmesi (trends raporlarıyla) hedeflenmiştir (Bologna, 2004). Bildiride kapsanan konular (aktivite alanları) şunlardır:

- Kolay anlaşılır ve karşılaştırılabilir dereceler (Diploma Eki),
- Temel olarak üç seviye yüksek öğrenim: lisans, lisansüstü, doktora,
- Ortak kredi sistemi (ECTS),
- Öğrenci/öğretim üyesi hareketliliği,
- Kalite güvencesi konusunda işbirliği,
- Yüksek öğrenimde Avrupa boyutu.

Üniversitelerden beklenti bu sürece olumlu yaklaşılması ve etkin katkıda bulunulmasıdır. Bologna sürecini büyük bir şemsiye olarak kabul edersek, bu şemsiyenin üniversiteleri ilgilendiren önemli basamakları şunlardır.

- Diploma Eki etiketi,
- ECTS (Avrupa Kredi Transfer Sistemi) etiketi,
- Sokrates/Erasmus.

Her basamağın önemle üzerinde durduğu nokta açıklıktır. Bu her yükseköğretim kurumunun ne olduğunu, ne olmak istediğini ve bunu hangi koşullarda yaptığını tüm açıklığıyla ortaya koyması ile mümkündür. Bu da öğretimin üç değişmez unsuru olan öğrenciler, öğretim elemanları ve öğretim planları unsurlarının şeffaf hale getirilmesi demektir.

Tüm bu sebep ve ihtiyaçlar göz önüne alındığında üniversitemiz genelinde internet üzerinden hizmet verecek bir ders veri tabanı ve katalog sistemi oluşturulmasına karar verilmiştir.

Planlanan AKTS DKS'ne göre bir ders için geçerli olan şu verilerden söz edilebilir: akademik birimi, bölümü, ana bilim dalı (varsa), yıl, sınıf, s0mestr, dersin adı, ders kodu, durumu, 0ğretim elemanı/elemanları, sicil noları, dersin dili, OMÜ kredisi, ECTS kredisi, haftalık ders saati (teorik), haftalık ders saati (pratik), haftalık ders saati (laboratuvar), 0n kořullar, ders tanımlaması, 0ğretim metotları, 0ğretim materyalleri, deęerlendirme metotları, kaynaklar ve g0r0řler. Sistem t0m 0niversite genelinde okutulan dersleri tek bir ortamda birleřtirmiřtir. Dolayısıyla bu, 0niversitedeki b0l0mler arası koordinasyonu ve iřbirlięini gerektirmektedir. AKTS DKS'nin 0niversitemizdeki aktif versiyonuna <http://ects.comu.edu.tr/katalog> adresinden eriřilebilir.

### **3.1 İřletme Politikası**

Bologna s0recinin, AKTS ve AKTS DKS'nin gereksinimlerini planlamak ve karřılamak 0zere 0niversitemizde bir ECTS komisyonu kurulmuřtur. Ayrıca t0m akademik birim ve b0l0mler bazındaki ihtiyaları karřılamak ve devamlılıęı saęlamak adına b0l0m koordinat0rleri belirlenmiřtir. AKTS DKS'nde koordinat0rler sorumlusu olduęu b0l0mler iin yetkilendirilmiřlerdir. S0z konusu koordinat0r sorumlu olduęu b0l0m ile ilgili ders verilerini temin etme ve sisteme girme g0revlerini yerine getirir. Sistemde dersler ile ilgili ders adı, kodu, kredisi, yıl, sınıf vb. genel veriler hazır olarak listelenir. Bu 0niversitenin 0ęrenci iřleri sistemi etkileřimi ile saęlanır. Koordinat0r yalnızca dersin 0ğretim metodu, ders tanımı, kaynaklar vb. ierik bilgilerini temin eder ve sisteme yerleřtirir.

### 3.2 Sistemin Veri Tabanı Yapısı

AKTS DKS Türkçe ve İngilizce olmak üzere iki dilde ders verilerini içerir. Buna göre, veri tabanı yapısı oluşturulurken ders ile ilgili sayısal verilerin (ders kodu, yıl, sınıf, sömestr, kredi bilgileri) dil ayrımı gözetmeyeceği dolayısıyla tekrarlanmaması düşünülmelidir. Buna göre ders verilerini 2 tablo halinde depolayabiliriz. Bunlar;

**course\_general Tablosu:** Dersin birim bölüm nolarını, ders adı (Türkçe ve İngilizce olarak), ders kodu, sömestr sınıf gibi sayısal verilerini ve dersin öğretim elemanı isim ve sicil no bilgilerini içerir. Burada sicil no ileride anlatacağımız Akademik Özgeçmiş Bilgi Sistemi ile bağlantı için kullanılmaktadır.

**course\_detail Tablosu:** Dersin ders tanımlaması, öğretim metotları gibi diğer içerik bilgilerini barındırır. course\_general tablosundaki her bir kayda karşılık course\_detail tablosunda 2 kayıt bulunmaktadır.

Ayrıca veri tabanında, sistemde kullanılan diğer sabit verileri barındırmak için tablolar bulunmaktadır. Bunlar dışında AKTS DKS işletimi ve işlevi için çok önemli olan “ders görüşleri” tablosu tanımlanmıştır. Bu tabloda herhangi bir ders hakkında dünya üzerindeki tüm ilgili kullanıcıların belirttikleri görüş ve istekler barındırılır. Sistemin genel tarama arayüzü üzerinden gönderilebilen bu görüş ve istekler ilgili dersi veren öğretim elemanına anında iletilir. Böylelikle bir ders iletişim mekanizması oluşturulmuş olur.

### 3.3 Arayüzler ve Kullanıcılar

AKTS DKS genel olarak 3 arayüzden oluşur. Bu arayüzler sistemle bağlantısı olacak kullanıcı kitlesinin yetkilerine göre belirlenmiştir. Bu kullanıcılar dersle ilgili veri girişi sağlayacak koordinatörler, sistemi yönetecek kişiler ve tüm ders verilerini arama ve listeleme eylemi gösterecek internet kullanıcılarıdır. Şimdi bu arayüzleri kısaca inceleyelim.

- **Koordinatör Arayüzü**

Bu arayüz üniversitedeki birimler ve bölümler bazında belirlenmiş koordinatörlerin erişim yetkisinde olan ve ders içeriklerinin yönetildiği kısımdır. Her koordinatörün yalnızca kendi yetkili olduğu bölüm derslerine erişim ve düzenleme izni vardır.

Arayüz listeleme mantığı, grafiksel görünüm ve içerik durum alanları ile koordinatörün sistemi çok kolayca kullanımına destek verir. Her ders için yedi içerik bilgisi söz konusudur. Bu ders içerikleri ile ilgili durum bilgisi ve ayrıca öğretim elemanı bilgisi, her ders kaydının yanında durum çubuğu şeklinde aşağıda gösterildiği gibi belirtilir.



Şekil 3.1 : Koordinatör paneli ders içerik durumu gösterimi

- **Genel Tarama Arayüzü**

Üniversite genelinde AKTS DKS'ne eklenen tüm dersler bu arayüz aracılığı ile internet kullanıcılarına açılır. Kullanıcı arama ekranından istenen birim, bölüm ve nitelikteki dersleri listeleterek detaylı içerik bilgilerine ulaşabilir. Ayrıca ders iletişim formu aracılığı ile görüş ve isteklerini koordinatörlere ve dersi veren öğretim elemanına iletebilirler.

- **Yönetim Arayüzleri**

AKTS DKS'nde iki tip yönetim arayüzü bulunmaktadır. Bunlar genel sistem yöneticisinin kullandığı, koordinatör tanımlarının, parola işlemlerinin ve genel sistem işlemlerinin yapıldığı sistem yöneticisi arayüzü, diğeri de Öğrenci İşleri Dairesi Başkanlığı'nca kullanılan ve ders girişlerinin yönetildiği ders yönetim arayüzüdür. Sisteme koordinatörler tarafından giriş yapılan ders içeriği verilerinin yöneticiler tarafından doğruluğunun kontrolü ve sorgulanması söz konusu değildir. Bu konuda yetki ve sorumluluklar koordinatörlere aittir.

## BÖLÜM 4

### AKADEMİK ÖZGEÇMİŞ BİLGİ SİSTEMİ

ECTS (European Credit Transfer System) bilgi paketi kapsamında, bir üniversitede okutulan derslerle ilgili genel tanımlar ile temel içerik bilgileri ve bunun yanında dersi veren öğretim elemanları hakkında özgeçmiş bilgileri açık olarak belirtilmelidir. AB ülkelerinin üzerinde önemle durduğu eğitimde kalite güvencesi konusu ve bu konudaki çalışma ve arayışlara göre, bir öğrencinin yüksek öğrenimde kazandığı her türlü bilgi, deneyim ve mesleki kazanımları açık ve belirgin olmalıdır. AKTS DKS ile bağlantılı çalışan AÖBS, öğretim elemanlarının özgeçmişlerinin güncel ve anında erişilebilir olmasını sağlayacaktır.

Akademik Özgeçmiş Bilgi Sistemi'nin üniversitemizdeki aktif versiyonuna <http://akademik.comu.edu.tr> adresinden ulaşılabilir.

#### 4.1 Üniversitemizde Akademik Özgeçmiş Verilerinde Önceki Durum

Üniversitemizdeki akademisyen özgeçmiş verilerinde, bu tezde anlatılan çalışma ve projeler uygulanmadan önce, bilgi teknolojilerinin gerektiği şekilde yapılandırılmamış, dağınık, tutarsız ve ulaşılması zor bir sunum söz konusuydu. Mevcut durumda bu konudaki sorun ve eksiklikler aşağıda özetlenmiştir.

- **Karmaşıklık, birbirinden farklı format ve dağınık veriler (Standartsızlık):** Önceki durumda üniversitemizde akademik özgeçmiş verileri, html, pdf, doc gibi çeşitli formatlarda yayınlanmaya çalışılmaktaydı. Bu örnekler başka birçok üniversitede de görülmektedir. Genel bir standarda oturtulmamış, karmaşık, dağınık şekildeki bu veriler, statik veri kaynaklarıdır. Veri sahipleri kendi kişisel zevklerine uygun format, renk, font, sıralama vb. seçeneklerini kullanarak belgelerini hazırlayabilirler. Ancak üniversite genelinde bakıldığında, bu çeşitliliğin içinden çıkılmaz bir karmaşaya dönüştüğü görülmektedir.

- **Zaman, yer ve platform bağımlılığı ve güncelleme zorluğu:** Önceki durumda, bahsedilen formatlarda hazırlanan ve internet ortamında bulunan özgeçmiş verilerinin güncellemesi gerektiğinde, söz konusu kişinin dünya'nın neresinde olursa olsun, ilgili formatın düzenleme aracına yani editörüne, internet ftp hesabına erişebileceği diğer yazılımlara vb. ihtiyacı olacaktır. Bu da her mekan ve zamanda mümkün olmayabilir. Aynı zamanda özgeçmiş verileri her ne kadar düzenli şekilde yazılmış olsalar bile, genelde düz metin niteliğinde olduğundan, veri üzerinde herhangi bir değişiklik, ekleme, silme işlemleri güçlükle yapılabilecektir. Dinamik olarak değişen çok sayıdaki yayının ve diğer akademik verilerin doğru, hızlı ve kolayca güncellenebilmesi önemlidir.
- **Güncel olmayan veriler:** Gerçekten de önceki duruma bakıldığında, internet ortamında bulunan özgeçmiş verilerinin, çok azının düzenli olarak güncellendiği görülmektedir. Bunun nedenleri yukarıda sayılan nedenler olabilir. Ayrıca bir üniversitede ftp erişimi yetki hiyerarşisi düşünüldüğünde, her akademik özgeçmiş sahibinin kendi özgeçmiş verilerine erişilmeye yetkisi olmaması ve işlemlerini web sorumluları aracılığıyla dolaylı yoldan yapabilmesi de bir nedendir. Bunun yanında akademisyenin ftp erişim yetkisi olsa bile, özgeçmiş ile ilgili web sayfasının düzenlenmesi konusunda bilgi eksikliğinin olabilmesi bu nedenlere eklenebilir.
- **Arama ve veri bulma zorluğu:** Bir üniversite genelinde, bu şekilde karmaşık ve dağınık şekilde özgeçmiş verileri düşünüldüğünde, aranılan nitelik veya duruma uygun kişiyi, veriyi bulma oldukça zordur. En basit bir örnekle, üniversitede Almanca'sı iyi düzeyde olan profesörlerin özgeçmiş verilerine ulaşmak çok güçtür. Bunu genişlettiğinizde, Almanca'sı iyi düzeydeki doçentleri ve çalıştığı birimi eğitim fakültesi olanları ve de fizik konusunda çalışanları vb. bulmak çok zordur. Önceki durumda yalnızca ilgili birimin web sayfası üzerinden o birimde çalışan akademisyenlerin çoğunlukla güncel olmayan özgeçmiş verilerine ulaşılabilir.

- **İstatistiksel veri elde edememek:** Tüm bir üniversite geneline yayıldığında, örneğin o üniversitenin Science Citation Index'deki makale sayısı, yada geçmiş yıllara göre yayınlardaki artış miktarı, bunların birimlere göre dağılımları, birimlere göre öğretim elemanı sayıları, veya bir akademisyenin özgeçmişinin günde kaç kişi tarafından ziyaret edildiği ve en son ne zaman güncellendiği vb. bilgilerine ulaşmak, önceki durumda gerçekten zor ve vakit alacak işlemlerdi.

#### 4.2 Akademik Özgeçmiş Bilgi Sisteminin Getireceği Kazanımlar

Yukarıda bahsedilen eksikliklerin giderilmesi ve Akademik Özgeçmiş Bilgi Sisteminin getireceği faydaları genel olarak şöyle sıralayabiliriz :

- **Konu başlıkları yapısı, birbirinden bağımsız konularda bilgiyi giriş ve düzenleme imkanı:** AÖBS çeşitli konu başlıkları halinde tasarlanmıştır. Her konu için bilgi girişleri ve düzenleme işlemleri birbirinden bağımsız olarak yapılmaktadır. Bu bağımsızlık sayesinde kullanıcılar örneğin; yeni bir makaleyi kaydedip sistemden çıkabilir, başka bir zaman diliminde sisteme tekrar girip başka bilgilerini güncelleyebilirler.
- **İstenilen konulara göre çıktı üretme ve sunum olanağı:** Yine AÖBS'nin ayrı konular halinde olmasının başka bir avantajı da çıktı alma ve görüntüleme aşamasında karşımıza çıkmaktadır. Eğer özgeçmişin tümü değil de belirlenen konulardaki bilgilerin çıktısı alınmak istenirse sistem buna da olanak sağlamaktadır. Verilerin doc uzantılı Word dokümanı şeklinde bilgisayar ortamına kaydedilmesi ve özgeçmişin başka kurum isteklerine uygun şekilde düzenlenebilmesi de mümkündür.



- **Zaman, yer ve platform kısıtlaması olmadan güncelleme serbestliđi:** Sistemin tasarımında kullanılan internet teknolojileri sayesinde, dünyanın neresinde ve hangi zaman diliminde olunursa olunsun, başka hiçbir düzenleme aracı programı yada editörü kullanımına gerek kalmadan veriler güncellenebilir ve bilgilere anında ulaşılabilir.
- **Kullanıcı arkadaşı arayüz:** Akademik personelin kendilerine ait özgeçmiş verilerini giriş ve düzenlemeleri için tasarlanan arayüzler tamamen kullanıcı arkadaşı olarak hazırlanmıştır. Bu arayüzler sayesinde kullanıcıların hata yapma olasılıkları alt seviyeye çekilmiş olurken, en rahat ve hızlı şekilde veri işlemlerine de olanak sağlanmıştır.
- **Türkçe ve İngilizce olarak veri girişi ve yayınlama imkanı:** AÖBS Türkçe ve İngilizce olmak üzere iki dilde tasarlanmıştır. Böylelikle akademik personel özgeçmişini iki dilde birden girme ve yayınlayabilme olanağı bulmaktadır.
- **Kişisel Ayarlar:** Kullanıcı, fotoğrafının görünüp görünmemesi yada özgeçmişinin renk ve arka plan gibi tema özelliklerine karar verebilir yada sisteme giriş parolasını değiştirebilir. Yine sisteme son giriş zamanını, yada özgeçmişinin bugün ve toplamda kaç kişi tarafından ziyaret edildiđi verilerine ulaşabilir. Ayrıca özgeçmişe fotoğraf ekleme işlemi de hızlı ve kolayca yapılabilir.

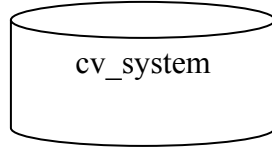
### 4.3 Veri Tabanı

Akademik Özgeçmiş Bilgi Sistemi internet temelli bir uygulamadır. Üniversitemiz mevcut web sunucuları LINUX tabanlı işletim sistemleri ile çalışmaktadır. AÖBS için MySQL veri tabanı yönetim sistemi kullanılmıştır. PHP dilinin MySQL ile diğer veritabanlarına göre daha hızlı ve performanslı çalışması MySQL'i tercih sebeplerinden biridir. Diğer yandan, MySQL ile internet temelli uygulamalar için kullanılabilen diğer veritabanlarından üç tanesi üzerinde yapılan bir özellik kıyaslaması Çizelge 4.1'de gösterilmiştir.

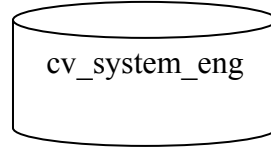
	MySQL	PostgreSQL	Oracle	MSSQL
<b>Veri Depolama</b>				
Depolama Modelleri	MyISAM, InnoDB, Berkeley DB, full-text	Postgres	Bit haritalı, B-tree, IOT, fonksiyon tabanlı	Kümelenmiş, kümelenmemiş
Okunabilirlik	Yüksek / Çok Yüksek	Yüksek	Yüksek / Çok Yüksek	Yüksek / Çok Yüksek
Ölçeklenebilirlik	Büyük / Çok büyük	Büyük	Büyük / Çok büyük	Büyük / Çok Büyük
<b>Indexler</b> Tekli ve çoklu sütun, birincil anahtar ve tam metin	Evet	Evet	Evet	Evet
<b>Veri Bütünlüğü</b> ACID uyumluluğu, düşük seviye kilitleme, anında yedekleme, kısmi geri yükleme	Evet	Evet	Evet	Evet
<b>Replication</b>				
Single master	Evet	Evet	Evet	Evet
Multimaster	Evet	Evet/ Hayır *	Evet	Evet
Kümeleme	Evet	Hayır	Evet	Evet
<b>Arayüz Metodları</b> ODBC/JDBC, C/C++, ve Java	Evet	Evet	Evet	Evet
<b>Gelişmiş Özellikler</b> Stored Prosedurs, views, triggers, sequences, and cursors	Evet (versiyon 5.X den beri)	Evet	Evet	Evet

**Çizelge 4.1 :** Dört farklı veri tabanı mimarisi özelliklerinin kıyaslanması (Di Giacomo, 2005)

AÖBS'nin üniversitemizdeki çalışma yapısı düşünülürken, Türkçe ve İngilizce olarak iki dilde hizmet vermesi planlanmıştır. Bu bakımdan sistem ile ilgili genel veri değerleri ve kullanıcılar hakkındaki genel bilgiler dışında her türlü veri iki farklı dilde depolanmaktadır. Buna göre AÖBS veri tabanı, Türkçe ve İngilizce olmak üzere iki adet veri tabanından oluşmaktadır. Bu veritabanları şöyledir:



Sistem genel verileri ve Türkçe özgeçmiş verilerini depolar.



İngilizce özgeçmiş verilerini depolar.

Bu veritabanları aynı veritabanı sunucusu üzerinde barındırılmaktadır ve tanımlanan MySQL kullanıcısı her iki veritabanına da erişim hakkına sahiptir. Bu kullanıcının veritabanları üzerindeki yetkilerini şöyle sıralayabiliriz :

### **Veri üzerindeki yetkileri**

- SELECT: Veri okunmasına izin verir.
- INSERT: Verinin eklenmesine ve yerinin değiştirilmesine izin verir.
- UPDATE: Veri değiştirilmesine izin verir.
- DELETE: Veri silinmesine izin verir.
- FILE: Verinin içeri ve dışarı aktarılmasına izin verir.

### **Yapı üzerindeki yetkileri**

- CREATE: Yeni veritabanları ve tabloların oluşturulmasına izin verir.
- ALTER: Varolan tabloların yapısının değiştirilmesine izin verir.
- INDEX: İndekslerin oluşturulmasına ve kaldırılmasına izin verir.
- DROP: Veritabanları ve tabloların kaldırılmasına izin verir.
- CREATE TEMPORARY TABLES: Geçici tablolara oluşturulmasına izin verir.
- CREATE VIEW SHOW: Yeni görünümünün oluşturulmasına izin verir.
- SHOW VIEW: Create View sorgularının yapılmasına izin verir.
- CREATE ROUTINE: Depolanan yordamların oluşturulmasına izin verir.
- ALTER ROUTINE: Depolanan yordamların değiştirilmesine ve kaldırılmasına izin verir.
- EXECUTE: Depolanan yordamların yürütülmesine izin verir.

Şimdi bu veritabanlarında bulunan tablolara ve yapılarına bakalım.

### 4.3.1 Tablolar ve Yapıları

Veri tabanları (cv\_system ve cv\_system\_eng) üzerinde bulunan tablolar, genel sistem verilerini barındıran tablolar hariç, hem isim hem yapı itibariyle her iki veritabanında da aynıdır. Bu tablolar ve barındırdıkları verilere göre açıklamaları şöyledir:

<b>aa_ext_lecturer</b>	: Üniversitede dışardan gelen öğretim elemanları bilgilerini tutar.
<b>aa_gen_statistic</b>	: Genel sistem istatistiklerini depolar.
<b>aa_logs</b>	: Sisteme girişlerin durumlarını tutar.
<b>aa_masters</b>	: Sistem yöneticileri ile ilgili yetki bilgilerini barındırır.
<b>aa_sayac</b>	: Sistem kullanıcıları özgeçmişleri için sayaç bilgilerini tutar.
<b>aa_types</b>	: Sistem genel akademik veri tipi verilerini barındırır.
<b>pdm1_aunvanlar</b>	: Sistem hazır değerleri tablosu. Akademik unvanlar.
<b>pdm2_birimler</b>	: Sistem hazır değerleri tablosu. Üniversite birim ve bölümleri
<b>pdm3_universiteler</b>	: Sistem hazır değerleri tablosu. Türkiye'deki üniversiteler
<b>pdm4_enstituler</b>	: Sistem hazır değerleri tablosu. Türkiye'deki enstitüler
<b>pdm5_tezkonular</b>	: YÖK te kayıtlı bulunan tez konuları
<b>pdm6_idarigorev</b>	: Sistem hazır değerleri tablosu. İdari görevler
<b>pdm7_ydiller</b>	: Sistem hazır değerleri tablosu. KPDS kapsamındaki yabancı diller
<b>pdm9_ulkeler</b>	: Sistem hazır değerleri tablosu. BM ye kayıtlı ülkeler
<b>pdm10_iunvanlar</b>	: Sistem hazır değerleri tablosu. İdari unvanlar
<b>t0_pconfig</b>	: Akademik kullanıcıyı konfigürasyon bilgileri
<b>t1_kbilgiler</b>	: Akademik kullanıcının genel bilgileri
<b>t2_gorevbirim</b>	: Akademik kullanıcının görev yaptığı birimler
<b>t3_idarigorev</b>	: Akademik kullanıcının idari görevleri
<b>t4_kuruluye</b>	: Kurul komisyon üyelikleri
<b>t5_koorgorev</b>	: Koordinatörlük görevleri
<b>t6_0_lisans_on</b>	: Ön lisans ve lisans bilgileri
<b>t6_1_ylisans</b>	: Yüksek lisans bilgileri
<b>t6_2_sanaty</b>	: Sanatta yeterlik bilgileri
<b>t6_3_doktora</b>	: Doktora bilgileri

<b>t6_4_tuzman</b>	: Tıpta uzmanlık bilgileri
<b>t7_akeunvan</b>	: Akademik unvanları
<b>t8_uzmanalan</b>	: Uzmanlık alanları
<b>t9_ydil</b>	: Bildiği diller ve düzeyleri
<b>t10_ydilsinav</b>	: Girdiği dil sınavları ile ilgili sonuçlar
<b>t11_uyekurulus</b>	: Üyesi olduğu kuruluşlar
<b>t12_oduller</b>	: Almış olduğu ödüller
<b>t13_makaleler</b>	: Makaleler ile ilgili tüm bilgiler
<b>t14_sanatdaluret</b>	: Sanat dallarında üretimleri ile ilgili bilgiler
<b>t15_bildiriler</b>	: Bildiriler
<b>t16_patentler</b>	: Patentler
<b>t17_kitaplar</b>	: Akademik kullanıcının kitap yazarlıkları konusundaki veriler
<b>t18_editorluk</b>	: Akademik kullanıcının editörlükleri
<b>t19_egitimetk</b>	: Verilen dersler, ylisans ve doktora tez yönetimleri
<b>t20_projeler</b>	: Araştırma projeleri ve raporlar
<b>t21_digetkinlik</b>	: Diğer akademik etkinlikleri
<b>t22_burslar</b>	: Almış olduğu burslar
<b>t23_toplumetk</b>	: Toplumsal etkinlikleri
<b>t24_diger</b>	: Tüm hepsinin dışında kalan etkinlik ve çalışmaları
<b>t25_abproje</b>	: AB projeleri kapsamında yapmış olduğu çalışmaları
<b>t26_hobiler</b>	: Hobi ve ilgi alanları
<b>t27_atiflar</b>	: SCI ve AHCI kapsamında yayınlarına yapılmış atıf sayıları

#### 4.3.2 Veri Yapıları

Tüm tablolar genel yapı itibariyle birbirine benzer özellik gösterir. Çoğu tablo içerebileceği maksimum kayıt sayısına göre tür özelliği almış bir anahtardan ve barındıracağı verilerle ilgili tür özellikleri belirlenmiş alanlardan oluşur. Örnek olarak bir kaç tablonun barındırdığı veri alanlarını ve veri türlerini inceleyelim.

t1\_kbilgiler tablosu:

Alan	Türü	Karşılaştırma	Öznitelik	Varsayılan	Ekstra
<b>cv_id</b>	smallint(5)		Unsigned		auto_increment
<b>sicil_no</b>	varchar(10)	<i>latin5_turkish_ci</i>			unique
<b>adi</b>	varchar(30)	“			
<b>soyadi</b>	varchar(40)	“			
<b>foto</b>	enum('0', '1')	“		0	
<b>unvan_id</b>	smallint(5)		Unsigned	0	
<b>dogum_yeri</b>	varchar(15)	<i>latin5_turkish_ci</i>			
<b>dogum_tar</b>	date			0000-00-00	
<b>uyrugu</b>	char(2)	<i>latin5_turkish_ci</i>		TC	
<b>uyrugu_dig</b>	varchar(25)	“			
<b>ana_dil</b>	varchar(100)	“			
<b>cinsiyet</b>	enum('0', '1', '2')	”		2	
<b>tel1</b>	varchar(15)	<i>latin5_turkish_ci</i>			
<b>tel2</b>	varchar(15)	“			
<b>telis_1</b>	varchar(22)	“			
<b>telis_2</b>	varchar(22)	“			
<b>fax</b>	varchar(15)	“			
<b>postaadres</b>	tinytext	“			
<b>eposta1</b>	varchar(50)	“			
<b>eposta2</b>	varchar(50)	“			
<b>web</b>	varchar(50)	“			
<b>kdr_birimid</b>	smallint(5)		Unsigned	0	
<b>kdr_bolumid</b>	smallint(5)		Unsigned	0	
<b>kdr_adalid</b>	smallint(5)		Unsigned	0	
<b>last_login</b>	datetime			0000-00-00 00:00:00	
<b>ip</b>	varchar(15)	<i>latin5_turkish_ci</i>			
<b>dis</b>	enum('0', '1')	<i>latin5_turkish_ci</i>		0	
<b>belge_durum</b>	enum('0', '1')	<i>latin5_turkish_ci</i>		0	

t6\_1\_ylisans tablosu:

Alan	Türü	Karşılaştırma	Öznitelik	Varsayılan	Ekstra
<u>rec_id</u>	int(10)		Unsigned		auto_increment
sicil_no	varchar(10)	<i>latin5_turkish_ci</i>			
yl_uni_id	smallint(5)		Unsigned	0	
diger_uni	varchar(120)	<i>latin5_turkish_ci</i>			
yl_enst_id	smallint(5)		Unsigned	0	
diger_enst	varchar(120)	<i>latin5_turkish_ci</i>			
yl_ulke_id	smallint(5)		Unsigned	0	
yl_tez_id	smallint(5)		Unsigned	0	
yl_diger_tez	varchar(150)	<i>latin5_turkish_ci</i>			
yl_tez_baslik	varchar(200)	<i>latin5_turkish_ci</i>			
yl_yil	varchar(4)	<i>latin5_turkish_ci</i>			

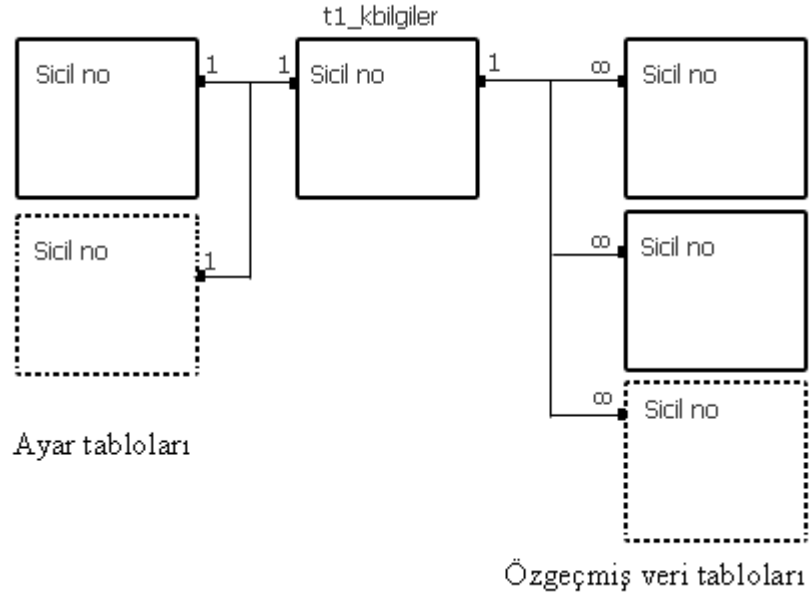
#### 4.3.3 Tablolar Arası İlişki Yapısı

AÖBS veritabanında yer alan “t1\_kbilgiler” tablosu ile diğer özgeçmiş verilerini barındıran tablolar arasında bire-çok ilişki türü vardır. Bu ilişkilerde t1\_kbilgiler tablosu dışındaki özgeçmiş veri tablolarında, akademik kullanıcının kurum sicil numarası ikincil anahtar (foreign key) görevi görmektedir. Ayrıca her tabloda otomatik artış özellikli ve tablo kayıt nosunu tutan birincil anahtar bulunur. Tabloların normalizasyonu da sistemin işleyiş gereksinimlerine göre sağlanmıştır.

Veritabanı normalizasyonu önemli kurallardan oluşur. Bu kurallar kullanılarak veritabanı daha sonraki büyüme ve ilişkiler için daha esnek bir yapı kazanır. Normalleştirilmemiş düz bir tablo birçok sütundan oluşan Excel tablosu gibidir. Birden fazla düz tablo arasında hiçbir ilişki yoktur. Bir düz tablo diskte normalleştirilmiş tablolara göre çok daha fazla fiziksel alan kaplar (Meloni, 2003).

Genel kullanıcı verileri kayıt tablosu (t1\_kbilgiler) ile diğer tablolar arasındaki ilişki yapısı aşağıdaki gibi ifade edilebilir.





**Şekil 4.1:** Veri tabloları ilişki yapısının genel gösterimi

#### 4.4 Sistemdeki Güvenlik İşlemleri

AÖBS ile ilgili güvenlik işlemlerine genel başlıklar halinde aşağıda değinilmiştir.

##### 4.4.1 Session (Oturum) Kullanımı

Web uygulamalarında, sunucu ile istemci arasında yetki (authorization) gerektiren işlemler yürütmek gerektiğinde, sunucu tarafında oturum (session) açma yöntemi yaygın olarak kullanılır. Birden fazla sayfaya erişim yetkisi veya sunucu üzerinde çalışan kullanıcı bazlı uygulamalar için yetki kontrolü bu şekilde yapılabilir. Yetki gerektiren işlemler dışında session yöntemi, web istemcisinin işlettiği süreçlere yönelik saklanması gereken verileri geçici süreyle sunucuda depolamayı da sağlar. Genelde basit uygulamalarda yetki bilgilerinin kontrolü ve bu yetki bilgilerinin işlem yapılan her sayfaya (POST veya GET yöntemleriyle) gönderilmesi ve tekrar kontrolü sıkça yapılan bir seçenektir. Ancak yetki bilgilerinin her sayfada tekrarlanması ve ağ

üzerinde bu bilgilerin yinelenmesi güvenlik açığı ortaya çıkaracaktır. Bu soruna yol açmamak için AÖBS'nin “Akademik Özgeçmiş Giriş” ve “Sistem Yönetimi” kısımlarında oturum açma yöntemi (session) kullanılmıştır.

Oturum açmada genel işlem mantığı, sisteme giriş yetkisi olan kişinin kullanıcı adı ve parolasını bir giriş (login) ekranından girmesiyle başlar. Giriş sayfasının form aracılığı ile işlem dosyasına gönderilmesi (submit) sonrasında yetki bilgileri veri tabanında sorgulanır. Eşleşme varsa eşleşen kayıttan bazı veriler geri döndürülür. Genelde form üzerinden parola gönderme işlemleri söz konusu olduğunda http üzerinde POST metodu kullanılır. Çünkü POST metodu form üzerinden girilen verileri ağ üzerinde belirli bir takım algoritma ve gizlilikler içerisinde gönderir. GET metodunda ise girilen veriler olduğu gibi URL adresine yansır.

Formun gönderildiği işlem sayfasında gerekli veri tabanı bağlantıları ve kontroller yapılır. Eğer yetki bilgileri onaylanırsa oturum başlatılır ve sunucu üzerinde uzantısı olmayan bir geçici metin dosyası (session file) oluşturulur. Oturum değişkenleri belirlenerek alacak değerleriyle birlikte bu dosyaya yazılır. Sunucu üzerinde oturum başlatma ve oturum değişkenleri kullanımı için PHP komut betiği şöyledir:

```
session_start();           // Start Session  
session_register('user_id'); // Sunucu tarafında bir oturum değişkeni tanımlanır.  
$_SESSION['user_id'] = $row[0]; // tanımlanan değişkene veri tabanından gelen veri kaydedilir.
```

#### **4.4.2 Veri Şifreleme Yöntemi**

Veri tabanı olan bir uygulamada verilerin güvenliği çok önemlidir. Üstelik bu bir Web uygulaması ise dikkat edilmesi gereken güvenlik tehditleri çok fazladır. Bu yüzden veri tabanında barındırdığımız bazı yetki bilgilerini (parola gibi) veya tahmin edilmesini istemediğimiz bazı değerleri şifreleme (encryption) yoluyla koruyabiliriz. Bilinen bir çok veri şifreleme algoritmaları vardır.

PHP çift yönlü şifreleme sağlayan ENCODE() fonksiyonunun aksine, tek yönlü şifreleme (one-way encryption) sağlayan birkaç fonksiyon sunmaktadır. Tek yönlü şifrelemenin avantajı anahtarın ele geçirilme riskinin çok daha az olmasıdır. Bir yönlü şifreleme ile ayrıca deşifreleme anahtarının tutulması işlemine de gerek olmaz (Sheldon, 2005).

Kaynağa göre PHP tek yönlü şifreleme fonksiyonlarına kısaca şöyle değinebiliriz:

**PASSWORD(dizgi(string)):** Bu fonksiyon belirtilen dizgiyi (string) 41-byte hash değeri şeklinde şifreler (hash ≈ kargaşa).

**MD5(dizgi(string)) :** MD5 fonksiyonu kullanarak 128-bit şifrelenmiş bir veri elde edilebilir.

**SHA(dizgi(string)):** Eğer 160-bit şifrelenmiş bir veri elde etmek istenirse SHA() ya da SHA1() fonksiyonunu kullanılabilir.

MySQL tablolar hakkında tutulan yetkilendirme parolalarını PASSWORD() fonksiyonunu kullanarak saklamaktadır. PASSWORD() fonksiyonu her platformda desteklenmiyor olabilir. MySQL ile bir uygulama geliştiriliyorsa ve kullanıcıların parolaları veritabanında saklanmak isteniyorsa daha fazla genel olan ve tercih edilen MD5() ya da SHA() fonksiyonları tercih edilebilir. AÖBS'nde MD5 ve SHA fonksiyonları kullanılmıştır. Ayrıca bir MySQL sunucusuna bağlanıldığında tüm parola işlemleri otomatik olarak şifrelenerek gerçekleştirilir ve bu şekilde koruma sağlanmış olur.

AÖBS'nde veri şifreleme birçok kısımda kullanılmıştır. Parola ve benzeri verilerin veri tabanında şifrelenmiş bir şekilde yer alması bir güvenlik önlemidir. Bunun yanı sıra birazdan anlatılacak otomatik form anahtarlama tekniğinde yine veri şifreleme kullanılmıştır.

### 4.4.3 Ağ Üzerinde Şifreli Veri Gönderimi

İnternet üzerinde bir web uygulaması yaptığımızda gerçekten düşünülmesi gereken çok faktör vardır. Örneğin kullanıcı giriş ekranından parolasını girdiği anda verinin çeşitli yöntemlerle ağdaki veri trafiğini dinleyen kötü niyetli kişiler tarafından saptanma ihtimali olabilir. Bu yüzden bu tehditlere karşı bir diğer yöntem de gizlemek istediğiniz veriyi şifreli şekilde ağda dolaştırmaktır.

AÖBS’nde bu işlem kullanıcının sisteme giriş aşamasında kullanılmıştır. Kullanıcı giriş ekranına parolasına girip giriş düğmesine tıkladığı anda kullanıcı etkileşimli bir komut betiği (JavaScript) aracılığı ile girilen metin şifrelenir ve göndermenin hemen öncesinde şifreli hali eski metin üzerine yazılır. Böylelikle parola ağ üzerinde karıştırılmış (şifrelenmiş) şekilde yol alır. Kullanılan örnek JavaScript komut betiği şöyledir:

```
function alanKontrol(){  
    var str;  
    str = hex_md5(document.frm_login.password.value);  
    document.frm_login.password.value = str;  
    return true;  
}
```

### 4.4.4 Otomatik Form Anahtarlama

Otomatik form anahtarlama, genel anlamda belirli bir form üzerinde tekil (unique) bir veri tanımlama ve işlem dosyasında bu verinin kontrolü üzerine çalışır. Bu kontrol formun güvenliği için gereklidir. Örneğin şöyle bir durum ortaya çıkabilir. Bir kişi bilerek veya yanlışlıkla kayıt işlem dosyasını birden fazla çalıştırıp sisteme gereksiz kayıt ekleyebilir. Bunu engellemenin bir yöntemi her form gönderme (submit) işleminde farklı bir geçici veri üretmek ve bunun kontrolünü sağlamaktır.

Form anahtarı üretilecek sayfada önceden tanımlanmış anahtar üretme sayfası şöyle çağırılır:

```
require("form_key_generate.php");
```

Sonrasında üretilen form anahtarının saklanması ve işlem göreceği sayfaya gidebilmesi için form içerisinde aşağıdaki gibi gizli bir input nesnesi oluşturulur.

```
<input type="hidden" name="frm_key" value="<?=$gted_form_key?>">
```

Anahtar üretme sayfasında harf ve rakamlardan oluşan rasgele karakter dizisi üretme ve devamında üretilen karakter dizisinin şifrenip oturum değişkeninde saklanması işlemleri yapılır. Sistemde form anahtarı için rasgele 10 karakterli bir dizinin şifreli biçimi kullanılmıştır. Form anahtarı üretim PHP kodları şöyledir :

#### **form\_key\_generate.php** dosyası

```
function worduret($harfsayi, $sayiuret, $harfuret)
{
    $sharf = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r',
's', 't', 'u', 'v', 'y', 'z', 'q', 'w', 'x');
    $sharf_b = array('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',
'P', 'R', 'S', 'T', 'U', 'V', 'Y', 'Z', 'Q', 'W', 'X');
    $sayi = array('1', '2', '3', '4', '5', '6', '7', '8', '9', '0');

    if (($sayiuret == "1") && ($harfuret == "1")) { $denetim = "3"; }
    elseif (($sayiuret == "1") && ($harfuret == "0")) { $salla = "2"; }
    elseif (($sayiuret == "0") && ($harfuret == "1")) { $salla = "1"; }

    for($i=0; $i<$harfsayi; $i++)
    {
        if ($denetim == "3") { $salla = rand(1,3); }
        if ($salla == "3") { $gidecek = $gidecek . "" . $sharf_b[rand(0,25)]; }
        elseif ($salla == "2") { $gidecek = $gidecek . "" . $sayi[rand(0,9)]; }
        elseif ($salla == "1") { $gidecek = $gidecek . "" . $sharf[rand(0,25)]; }
    }
    return $gidecek;
}
function kript_word($word)
{
    $skript_word = md5($word);
    return $skript_word;
}
```

```

$rnd_word = worduret(10,1,1);
$gted_form_key = kript_word($rnd_word);

//-----Session'a frm_key yazılır -----

if(!session_is_registered("var_frmkey"))
{
    session_register('var_frmkey');
}
$_SESSION['var_frmkey'] = $gted_form_key;

```

#### 4.4.5 Veri İşlemlerinde Form Anahtarı ve Oturum Kontrolü

Daha önce bahsettiğimiz otomatik form anahtarı ve kullanıcı girişi sırasında oluşturulan oturum değişkenleri, kullanıcı veri tabanı üzerinde her değişiklik yapmak istediğinde (insert, update, delete işlemlerinde) kontrol edilir. Bu kontrol işlemleri, her akademik özgeçmiş konu başlığı ile ilgili işlem dosyaları çalıştırılmak istendiğinde tekrarlanırlar.

Örnek bir veri işlemi için kontrollerin nasıl yapıldığını ve kontrollere göre işlem süreçlerinin nasıl değerlendirildiği aşağıdaki gibi gösterilebilir.

```

<?
// Oturum başlatılır ve oturum değişkeni kontrol edilir.
session_start();
if(($HTTP_SESSION_VARS['ussicil']=="" ) OR (!session_is_registered('ussicil')))
{
    include("login.php");
    exit;
}
// otomatik form anahtarı ve oturum değişkeni formdan gelen değerlerle eşleştirilir.
İf(($HTTP_SESSION_VARS["var_frmkey"] == @$HTTP_POST_VARS["frm_key"]) AND
(@$HTTP_POST_VARS["sicil_no"]==$_SESSION_VARS['ussicil'])) {
.....
// Kontroller doğru. Geçerli veri tabanı işlemi gerçekleştirilir.
.....
} else {
// Kontroller yanlış. Kullanıcı erişim hatası mesajı içeren dosyaya yönlendirilir.

```

```
header("Location:http://".$_SERVER['HTTP_HOST'].
dirname($_SERVER['PHP_SELF'])."/english_err.php");
}
?>
```

#### 4.5 Sistem Hazır Değerleri

AÖBS’nde veritabanı üzerinde önceden tanımlanmış ve sistemde bir çok konu başlığında kullanılmak üzere veri tabloları oluşturulmuştur. Bu tablolarda barındırılan verilerden sistem hazır değerleri olarak bahsedebiliriz. Bu değerlerin amaçlarını şöyle sıralayabiliriz:

- Sistemde sık kullanılan verileri standartlaştırmak ve sınıflandırabilmek,
- Kullanıcının veri girişi sırasında hata yapmasını önlemek,
- Veri standartlarını yakalayarak arama ve görüntülemeyi mümkün kılmak,
- Tüm kullanıcıların eriştiği verilerin Türkçe ve İngilizce olarak veritabanında saklanarak sistemde kullanımını kolaylaştırmak,
- Verilerin rahat yönetilebilmesi.

Sistem hazır değerleri için ana türlerin neler olduğunu ve barındırdıkları verileri inceleyelim.

- **Akademik Unvanlar:** Akademik unvanları barındırır. (Prof.Dr., Doç.Dr., Uzman vs. gibi)
- **Üniversiteler:** Türkiye’deki üniversite isimlerini barındırır.
- **Enstitüler:** Türkiye’deki enstitü isimlerini barındırır.
- **Lisansüstü Tez Konuları:** YÖK’te kabul edilen lisansüstü tez konu başlıklarını içerir.
- **Üniversitemiz Birim ve Bölümleri:** Üniversitemizde birim, bölüm ve anabilim dalı hiyerarşisini içerir.
- **İdari Unvanlar:** İdari unvanları içerir.

- **İdari Görev Listesi:** En sık bilinen idari görevler (rektör, daire bşk., bölüm bşk vs.)
- **Ülkeler:** Birleşmiş Milletlere kayıtlı tüm ülkelerin isimleri
- **Yabancı Diller:** KPDS kapsamında tüm yabancı dil isimleri

#### 4.6 Sınıf (Class) Yapılarının Kullanımı

PHP nesneye yönelik (object-oriented) bir dildir. Dolayısıyla nesneye yönelik programlama dillerinin çoğu özelliğini barındırır. Nesneye yönelik programlamada sınıflar nesne oluşturmak için birer şablondurlar ve bir nesnenin nasıl oluşturulacağı bilgisini tutarlar. Bir nesne bir sınıfın örneğidir (instance). Sınıfı bir pasta kalıbına, nesnelere ise bu pasta kalıbı kullanılarak yapılan pastalara benzetebiliriz. (Lavin, 2006).

Genel olarak PHP’de bir sınıf tanımlaması aşağıdaki gibi yapılır ve AÖBS’nde benzer örnekler kullanılmıştır.

```
class myfunc {  
    function func_name ($variable, $var2)  
    {  
        .....  
    }  
}
```

Tanımlanan sınıfın PHP içerisinde kullanılması ve nesne oluşturulmasına bir örnek de aşağıdaki gibidir.

```
require("../class_file.php");  
$object = new myfunc();  
$return_value = $object->func_name($variable1,$var2);
```



Lavin (2006)'e göre PHP'de çoklu kalıtım (multiple inheritance) yoktur. PHP'nin nesneye yönelik programlama mantığında, bir nesnenin sadece tek bir ebeveyn sınıfı olabilir. PHP 5'in oluşturucuları sınıflardaki çoklu kalıtımı reddetmişlerdir. Bir sınıf birden fazla sınıftan türetilmiş olsa örneğin; turna kuşunu kuş sınıfından ve nesli tükenen türler (endangered species) sınıfından türetebiliriz. Görünüşte bir tehlike yoktur fakat yeme (eat) metodunda hangi ebeveynin yeme davranışı kullanılacaktır sorusu çoklu kalıtımda bir problem oluşturmaktadır.

Tekli kalıtım basit ve daha doğru bir yaklaşım sağlamaktadır fakat bazen bir sınıfta farklı sınıfların davranışlarına ihtiyaç olabilir. Bir turna kuşu hem bir kuştur hem de nesli tükenmektedir. PHP bu problemi arayüz (interface) kavramı ile çözmektedir. Bir arayüz veri üyeleri, veri tiplerinden oluşan değişkenler (data members) içermez, sadece kodu yazılmamış fonksiyon adlarından oluşur. Bu arayüzden türetilen bir sınıf arayüz içerisinde kodu yazılmamış olan bu fonksiyonların kodunu kendi içerisinde yazmak zorundadır. Bir sınıf sadece bir sınıftan türetililebilir fakat birden fazla arayüzü kullanabilir (Lavin, 2006).

#### **4.7 Sistemin Ana Bölümleri**

AÖBS aynı veritabanları üzerinde işlem yapan ve birbiriyle etkileşimli çalışan 4 ana bölümden oluşmaktadır. Bunlar “**Veri Giriş ve Yönetim**”, “**Sistem Yönetim**”, “**Arama ve Görüntüleme**” ve “**İstatistikler**” bölümleridir. Şimdi bu bölümleri inceleyelim.

##### **4.7.1 Veri Giriş ve Yönetim**

Veri giriş ve yönetim arayüzü yalnızca akademik kullanıcıların kullanım yetkisinde olan ve sistemin en yoğun kullanılan parçasıdır. Karşılıklı olarak sunucu ile istemci arasında en fazla veri akışı bu bölümde olur. Her akademik kullanıcı, herhangi bir tarih, saat yada yer gözetmeksizin sisteme bağlanıp, veri tabanı üzerinde kendileriyle ilgili bilgileri değiştirip yönetebilirler.

AÖBS geliştirilmeye başlandığında, yapılan arařtırmalar, anketler ve yönetimsel kriterler göz önüne alınarak, akademik özgeçmiş bilgilerinin belirli ana konu başlıkları altında toplanması kararlařtırılmıştır. Şimdi bu ana konu başlıklarını inceleyelim.

#### **4.7.1.1 Akademik Özgeçmiş Ana Konu Başlıkları**

AÖBS ana konu başlıkları, projenin üniversitemizde çalıştırılacağı düşünöldüğü için özellikle ÇOMÜ Atama ve Yükseltme Kriterleri değeriendirilerek belirlenmiştir. Şimdi kısaca bu konu başlıklarına değinelim.

Kişisel Bilgiler: Akademik kullanıcı ile ilgili ad, soyad, unvan, doğum tarihi-yeri, uyruk gibi genel bilgileri, telefon, e-posta, adres gibi her türlü iletişim bilgilerini ve kadrosunun bağı olduğı birim/bölüm/anabilim dalı bilgilerini içerir.

Öğrenim Durumu: Kişinin ön lisans, lisans, yüksek lisans, doktora, sanatta yeterlik ve tıpta uzmanlık ile ilgili her türlü fakülte, enstitü, yıl, tez ve tez konusu bilgilerini barındırır. Ayrıca birden fazla yapılan eğitim etkinliklerini de (örneğin birden fazla lisans bitirmek) desteklemektedir.

Yabancı Dil Bilgisi: Kullanıcıların KPDS kapsamında belirtilen veya bunların dışında özel olarak bildiğı dillerdeki seviyelerini yazabilmesine, aynı zamanda bu diller ile ilgili girdikleri sınavları, sınav sonuçlarını ve yıllarını belirtebilmelerine olanak sağlar.

Akademik Unvanları: Akademik kullanıcının sahip olduğı unvana ulaşıncaaya kadar almış olduğı unvanları, kurumları ve yılları barındırır.

İdari Görevler: Kişinin yapmış olduğı idari görevleri, kurumları ve görev yıllarını belirtebileceğı bir alandır.

Görev Yaptığı Birimler: Akademisyenin şu anda üniversitemizde görev yaptığı birimleri veya bölümleri belirtebileceği bir alandır.

Uzmanlık Alanları: Kişinin çalıştığı uzmanlık alanlarını belirttiği kısımdır. Diğer ana konu başlıklarında olduğu gibi bu kısımda da kayıt sınırı yoktur. Kişi sınırsız uzmanlık alanına sahip olabilir.

Kurul/Komisyon Üyeliği: Akademisyenin kurul veya komisyon üyeliklerini ve tarih aralıklarını belirtebildiği alandır.

Koordinatörlük Görevleri: Aynı şekilde akademisyenin eskiden bu yana yapmış olduğu koordinatörlük görevlerini ve tarih aralıklarını belirtebildiği kısımdır.

Üyesi Olduğu Bilimsel/Mesleki Kuruluşlar: Yine kişi aktif olarak üyesi olduğu ulusal veya uluslararası nitelikteki bilimsel veya mesleki kuruluşları belirtebilir.

Kazandığı Ödüller: Ulusal veya uluslararası kazandığı ödüller belirtilir.

Makaleler: Bu kısımda üniversitemiz "ÇOMÜ Atama ve Yükseltme Kriterleri" değerlendirilerek alt başlıklar belirlenmiştir. Bu alana eklenecek makaleler bu belirlenen alt başlıklara göre gruplanırlar. (Örneğin, SCI (Science Citation Index), SSCI (Social Science Citation Index), AHCI (Arts and Humanities Citation Index) tarafından taranan dergilerde yayımlanan teknik not, editöre mektup, tartışma, vaka takdimi ve özet türünden yayınlar dışındaki makaleler bir grup olarak veya Ulusal hakemli dergilerde yayımlanan teknik not, editöre mektup, tartışma, vaka takdimi ve özet türünden yayınlar başka bir grup olarak belirlenmiştir.)

Sanat Dallarında Üretim: Makalelere benzer şekilde iki alt grup halinde veri girişine imkan sağlar. Bunlar "uluslararası" ve "ulusal" olarak sergiler, festivallere katılım, icralar, ödül almış eserler, sanat hizmetleri, mimari proje sunumları olarak değerlendirilirler.

Bildiriler: Yine bildiler başlığı da “uluslararası” ve “ulusal” , “tam metin” ve “özet metin” gibi alt gruplara ayrılır. Örneğin; uluslararası kongre, sempozyum, panel gibi bilimsel toplantılarda sunularak, programda yer alan özet metin olarak yayınlanan bildiri ya da poster veya gösteri vb.

Kitaplar: Alanında yurtiçinde ya da yurtdışında yayınlanan, kitap veya ünite yazarlığı gibi alt gruplarda veri girişi sağlar.

Patentler: Kişinin almış olduğu bilimsel patentlerini girebileceği alandır.

Editörlükler: Yine bu bölümde belirli alt gruplarda veri girişine ve ilgili editörlük hakkında diğer bilgilere yer verilir. Örneğin; SCI, SSCI ve AHCI tarafından taranan dergilerde editörlük.

Eğitim ve Öğretim Etkinlikleri: Son iki yılda verilen lisansüstü, lisans veya önlisans dersleri, tamamlanan veya devam eden doktora yada yüksek lisans tezi yönetimleri belirtilir.

Araştırma Projeleri ve Raporlar: Uluslararası veya ulusal kuruluşlarca desteklenen proje yürütücülüğü, projede görev alma, final raporu, destekleyen kuruluş, proje tarih aralığı gibi bilgiler yer alır.

Diğer Akademik Etkinlikler: Alanı ile ilgili uluslararası ulusal kongre çalıştay sempozyum vb. etkinliklerde aldığı başkanlık veya görevler yer alır.

Kazanılan Burslar: Alanında bilimsel/sanatsal çalışma ve araştırmalar için kazandığı uluslararası veya ulusal burs bilgileridir.

Toplumsal Etkinlikler: Uluslararası/ulusal bilimsel, mesleki, sosyal, kültürel ve ekonomik kurum ve kuruluşların yerel şubelerinin üst yönetiminde alınan görevler, kamu kurum ve kuruluşlarında alınan görevler ya da katkılar vb bilgiler yer alır.

AB Eğitim/Araştırma Programları: Çerçeve programları kapsamında araştırma projesi koordinatörlükleri, Socrates, Leonardo programlarında yürütülen projeler ve koordinatörlükler belirtilir.

ISI Web of Science Atıflar: Web of Science tarafından taranan dergilerdeki toplam atıf sayısı belirtilir.

Hobi ve İlgi Alanları: Akademik personelin her türlü sosyal ve kültürel çalışmaları, hobileri yer alır.

Diğer: Tüm alanların dışında kalan bilgiler kaydedilir.

#### **4.7.1.2 Kullanıcı Tarafı Form Etkileşimi ve Hata Kontrolleri**

Akademik veri girişi arayüzünde, her özgeçmiş konu başlığı için ayrı veri girişi formu vardır. Dolayısıyla girilecek konu tipine göre doğru kontrol edilecek veri formatları da farklılaşabilir. Sistemdeki kullanıcıların veri girişleri sırasındaki hatalarını mümkün olduğunca azaltmak ve veri aramada veri bütünlüğünü sağlamak amacıyla, bu giriş formları JavaScript kod betikleri ile kontrol edilirler. Sistemde dil seçeneği de olduğu için bu kod betikleri sonucunda kullanıcılara iletilecek hata raporları da dil farkını desteklemelidir. Bu yüzden dinamik PHP dil dosyalarından gelen mesajları algılayabilmek için JavaScript betikleri PHP içerisinde yazdırılmıştır. Aşağıda akademik yayınlara yapılan atıflarla ilgili veri girişi formunun kontrol betiği örnek olarak verilmiştir.

```
<?
require("control_messages.php");
echo("<SCRIPT language=JavaScript>
<!--
function alanKontrol(){
    var alan = new Array;
```

```

var form = document.form_atif;
alan[0] = form.ebx_atif.value;
if (!alan[0]) { alert("\$.contmsg_atif."); return false; }
return true;
}
function isChar() {
if ((event.keyCode < 48 || event.keyCode > 57)) event.returnValue = false;
}
// -->
</SCRIPT>");
?>

```

#### 4.7.1.3 Veri Görüntüleme

AÖBS bağımsız konu başlıkları şeklinde tasarlandığından, veri görüntüleme aşamasında kullanıcıya farklı seçenekler sunar. Sisteme giriş yapan kullanıcı CV önizleme kısmından görüntülemek veya çıktı almak istediği konu başlıklarını seçer.

<input checked="" type="checkbox"/> Kişisel Bilgiler	<input checked="" type="checkbox"/> Öğrenim Durumu
<input checked="" type="checkbox"/> Yabancı Dil Bilgisi	<input type="checkbox"/> Akademik Unvanları
<input checked="" type="checkbox"/> İdari Görevler	<input type="checkbox"/> Görev Yaptığı Birimler
<input type="checkbox"/> Uzmanlık Alanları	<input type="checkbox"/> Kurul / Komisyon Üyeliği
<input type="checkbox"/> Koordinatörlük Görevleri	<input type="checkbox"/> Görev Yaptığı Kurum / Kurumlar
<input type="checkbox"/> Kurumlar	<input type="checkbox"/> Kurumlar

**Şekil 4.2 :** Veri görüntüleme konu seçimi

Diğer internet kullanıcıları ise, sistemden arayıp buldukları akademisyenlerin özgeçmiş verilerinin tamamını görebilirler.

#### 4.7.1.4 Kişisel Ayarlar

Sistemde kullanıcı kendi özgeçmiş verilerinin görüntülenmesine ve izlenmesine yönelik bazı tercihleri değiştirebilir ve görebilir. Özgeçmiş sayfasının renk ve görünüm tercihleri için belirlenmiş temalardan seçim yapabilir. Aynı zamanda özgeçmişinin günlük ve genelde izlenme sayılarını da görebilir. Bunlar dışında sistem giriş parolasını değiştirmesi de mümkündür.

#### 4.7.1.5 Fotoğraf Yükleme

AÖBS’nde, kullanıcı bilgisayar ortamında bulunan ve belirli formatlara uygun olan herhangi bir fotoğrafını kendi özgeçmişi ile ilgili kişisel bilgiler kısmına dolayısıyla özgeçmişinin en üst kısmına ekleyebilir. Sisteme fotoğraf ekleyen akademik personelin fotoğrafları belirli bir düzen içerisinde sunucuda bulunan bir dizinde depolanır. Aynı zamanda bu fotoğraflar veri tabanı ile de ilişkilendirilerek görüntüleme işlemine olanak sağlanır. Şimdi fotoğrafın sunucuya yüklenmesi kısmını inceleyelim.

- **Sunucuya fotoğraf yükleme (Photo upload)**

Akademik personel, kişisel ayarları ile ilgili bölüme girdiğinde eğer daha önce fotoğraf yüklemeyse, “fotoğraf yükle” düğmesi aktif ve görünür haldedir. Bu düğme yanında da “foto yok” veya dil seçeneğine göre “no photo” imajı bulunur. Eğer daha önce fotoğraf yüklemiş ise düğme “fotoğrafı değiştir” şeklinde görünür ve önceden yüklenmiş olan fotoğraf yan tarafta görüntülenir. Düğmeye tıklandığında açılan pencerede mevcut bilinen HTML kodları yanı sıra PHP ve JavaScript komutları ile yazılmış bir takım kullanıcı etkileşimi kodları bulunur. Veri tabanı üzerinden kişinin önceki fotoğraf durumu getirilir. JavaScript komutları ile de, kullanıcı bilgisayarından fotoğraf seçtiği anda ilgili alanda ön izlemesinin görülebilmesi sağlanır.

```

var htmlfile = document.getElementById('file_foto');
var file = htmlfile.value
var himg = document.getElementById('himg1');
if (file.length>0) himg.src = 'file://' + file;
else { himg.src = ''; };

```

Aynı zamanda fotoğrafın belirli boyut ve format kısıtlamalarına uyması için JavaScript komutlarıyla yazılmış bir kod süzgeç görevi görür. Sunucuya yüklenmek istenen fotoğraflar yalnızca **JPG** formatında ve dosya boyutları da en fazla **800 kb** olmalıdır. Her ihtimale karşı client (istemci) tarafında yanlış yorumlanan yada düzgün çalışmayan script komutlarına önlem olarak, sunucuda çalıştırılmak üzere yazılmış Php komutları da aynı kontrolleri geçekleştirecektir.

```

İf(($file_foto_size==0) AND ($file_foto_type==""))
{.....}
İf(($file_foto_type<>"image/jpeg") AND ($file_foto_type<>"image/pjpeg"))
{.....}
İf(($file_foto_size > 800000) OR ($file_foto_size <= 0))
{.....}

```

- **Sunucu üzerinde PHP ile imaj işlemleri**

Akademik kullanıcının sunucuya yüklemek istediği fotoğraf sistem için geçerli olmayan format ve dosya boyutlarındadır. Bu sorunu aşmak ve sistemdeki fotoğraf formatlarını standartlaştırmak için sunucuya yüklenmek istenen imajların, sunucu üzerinde çalışan bir kod aracılığı ile işlenmesi gereklidir. İlgili PHP kodu aşağıdaki gibi tanımlanabilir.

```

//---Foto boyutları-----
$size=GetImageSize($HTTP_POST_FILES['file_foto']['tmp_name']);
$genislik=$size[0];
$yükseklik=$size[1];
$resim_adi="cv_".$HTTP_SESSION_VARS['ussicil'].".jpg";
if (is_uploaded_file($HTTP_POST_FILES['file_foto']['tmp_name'])) {

```



```

$imj = $HTTP_POST_FILES['file_foto']['tmp_name'];
$imj_yol_b = "photos/".$resim_adi;
$skucultme_yuzdesi_b = 10000 / $genislik;
$swb = ($genislik / 100)*$skucultme_yuzdesi_b;
$shb = ($yukseklk / 100)*$skucultme_yuzdesi_b;
$thumb=imagecreatetruecolor($swb,$shb);
$imj=ImageCreateFromJpeg($imj);
ImageCopyResampled($thumb,$imj,0,0,0,0,$swb,$shb,$genislik,$yukseklk);
imageJpeg($thumb,$imj_yol_b,90);
imagedestroy($thumb);
imagedestroy($imj);
clearstatcache();
..... // diđer veri tabanı işlemleri devam eder.
}

```

Sunucunun geçici belleğinde tutulan gerçek imaj, yine sunucuda oluşturulan sanal bir imaja, tanımlanmış format ve kriterlere uygun olarak aktarılır. Daha sonra bu imaj sunucuda belirlenmiş dizine kopyalanır. Bu fotoğraf işlemi sayesinde hem sunucu yükü hafiflerken hem de sistemin performansı korunmuş olur.

- **Depolama ve Görüntüleme**

İstemci tarafından sunucuya gönderilen imajlar, imaj sahibinin sicil bilgilerinden oluşan bir isim ile belirlenmiş bir klasöre kaydedilirler. Bu kaydetme işlemi istemci tarafından gönderilen imaj nesnesinin belirlenen format ve özelliklere göre sunucuda işlenmiş halidir.

Belirlenen resim adıyla sunucuya kaydedilen resim için veritabanında kullanıcıya ait bir alanda işaretleme yapılır. Böylelikle özgeçmiş görüntülenecek akademisyenin, önce veritabanındaki imaj değeri daha sonrada sunucu üzerindeki imaj dosyası kontrol edilir.

## **4.7.2 Sistem Yönetimi**

Sistemin tüm hazır değerleri kontrolleri, kullanıcı kontrolleri ve yedekleme, veri kontrolü ve diğer yönetici işlemleri bu kısımda yapılır. Genel yönetici ve genel yönetici tarafından tanımlanan yetkileri kısıtlı yöneticiler tanımlıdır.

### **4.7.2.1 Yetkilendirmeler**

Sistem yönetim kısmına erişimi olan kullanıcılar yalnızca tam erişim yetkisi olan genel yöneticinin belirlediği alanlara erişebilirler. Sistemin hazır değerleri veya sistem değişkenleri gibi kritik verilerin değiştirilmesi yine yetkilendirilmiş kullanıcılar tarafından yapılabilir.

### **4.7.2.2 Kullanıcı İşlemleri**

AÖBS’nde iki tip akademik kullanıcı vardır. İlki üniversitede kadrolu olarak çalışan ve üniversitenin tanımladığı sicil numarası ile görev yapan akademik kullanıcılardır. Diğeri ise üniversiteye dışarıdan gelen ve YÖK tarafından belirlenmiş yönetmeliklere göre üniversite içerisinde ders verebilen akademisyenlerdir. Bu durumda AÖBS bu tip kullanıcılara yalnızca sistemin kullanımı için tanımlanmış akademik sicil numaraları atar. Her iki kullanıcı tipinde de sicil numarası anahtar ve tekil veridir. Sistem yöneticisi bu anahtarlar bazında kullanıcıları sisteme dahil eder, sistemden çıkarır veya parola atama gibi diğer kullanıcı yetki işlemlerini gerçekleştirir.

### **4.7.2.3 Veri Yedekleme**

Sistemin kullanıcı sayısı, güncellenen veri sayısı ve akademik verilerin önemi düşünüldüğünde sistemin veri yedeklemesi son derece önemlidir. Bu yüzden Sistem Yönetim kısmında yer alan yedekleme işlemleri ve genel yönetici ile sunucu yöneticisi tarafından yapılan veri tabanı yedekleme işlemleri çok sıklıkla gerçekleştirilir. Yöneticilerin kullanıcı silme hatalarına karşı da sistemde kullanıcı verilerini yedekleme mekanizması bulunmaktadır.

### **4.7.3 Arama ve Görüntüleme**

“Arama ve Görüntüleme” bölümü tüm internet kullanıcıları kullanımına açıktır. Bu kısımda “Veri Giriş ve Yönetim” bölümü aracılığı ile akademik kullanıcıların girmiş oldukları tüm veriler belirli arama arayüzleri sayesinde bulunabilir ve görüntülenebilir. Üç türlü arama arayüzü vardır. Bunlar; “hızlı arama”, “gelişmiş arama” ve “akademik etkinlik arama” dır. Aramada amaç, kullanıcıların aradığı nitelikteki akademisyenleri en hızlı şekilde bulmasıdır. Şimdi bu arama arayüzlerini inceleyelim.

#### **4.7.3.1 Hızlı Arama**

AÖBS’ne girilen verilerin belirli kriterlere göre aranıp görüntülenmesi çok önemlidir. Hızlı aramada aranmak istenen akademisyen ile ilgili isim, soy isim, bulunduğu birim gibi temel bilgiler kriter alınır.

Arama formunda yer alan isim alanında, kullanıcının gireceği değeri dinamik olarak veri tabanından sorgulayıp sonuç döndüren bir AJAX etkileşimi mevcuttur. Bu etkileşim sayesinde kullanıcı, girmek istediği ismi veri tabanında bulunan isimlerle sayfa yenilenmeden (reloading the page) kıyaslama olanağı bulur. AJAX teknolojisi PHP kütüphaneleri, JavaScript ve XML mimarilerinin birlikte kullanımı prensibine dayanır.

Sistemde AJAX kullanılırken aynı zamanda sunucuda yüklü bulunan PEAR kütüphanelerine başvurulmuştur. PEAR kütüphaneleri ile veritabanı bağlantısının yapılmasına bir örnek aşağıdaki gibidir.

#### **veritabani.php** dosyası

```
ini_set('include_path', './usr/share/pear/');
require_once 'DB.php';
$DSN = 'mysql://root:pass@localhost/cv_system';
$options = array('debug' => 2, 'portability' => DB_PORTABILITY_ALL);
$DB =& DB::connect($DSN, $options);
if (DB::isError($DB)) { die($DB->getMessage()); }
```

Bunun dışında GPL lisanslı bir AJAX işlem sınıflarından (xajaxResponse class) oluşan hazır bir kütüphane kullanılmıştır (xajax.inc.php dosyası). Sonrasında bir PHP fonksiyonu ile veritabanından gelen istenen kriterdeki kayıtlara göre bir JavaScript ve XML oluşturulur ve döndürülür.

```
mb_internal_encoding("iso-8859-9");
require_once "xajax.inc.php";
require_once "veritabani.php";
$xajax = new xajax();
$xajax->registerFunction("personel");
$xajax->processRequests();
....
Function personel($deger) {
    $deger = mb_convert_encoding($deger, "ISO-8859-9", "auto");
    $objResponse = new xajaxResponse();
    global $DB;
    $sql = "SELECT adi, soyadi FROM personel WHERE adi LIKE '". $deger. "%'";
    $res = $DB->query($sql);
    $kayit_sayi = $res->numRows();

    if($kayit_sayi>0) {
        for($k=0; $k < $kayit_sayi; $k++) {
            $row = $res->fetchRow(DB_FETCHMODE_DEFAULT,$k);

            if($k==0) $option_atama = "opt = new Array; for(j=500; j>=0; j--) {
                document.form1.hatir_select.options[j] = null; } ";

            $option_atama = $option_atama . "opt[\".$k.\"] = new Option(\\\"".$row[0].\"
                \". $row[1].\"\\"); opt[\".$k.\"].value = \\\"".$row[0].\"\\\"";

            $option_atama=$option_atama."document.form1.hatir_select.options[\".$k.\"]=opt[\".$k.\"];
            ";
        }
    }
}
```

```
}  
else { $objResponse->addAlert("Böyle Bir Kayıt Yok"); }  
$objResponse->addScript($option_atama);  
return $objResponse->getXML();
```

#### **4.7.3.2 Gelişmiş Arama**

Gelişmiş arama kısmı hızlı aramadan farklı olarak kişiler ile ilgili bazı detay bilgileri sorgulamak için hazırlanmıştır. Yine kişinin isim unvan bilgileri yanında, yapmış olduğu idari görevlerine, görev yaptığı birimlere, bildiği dil ve seviyelerine göre veya hobi ve ilgi alanlarına göre arama yaptırılabilir.

#### **4.7.3.3 Akademik Etkinlik Arama**

Akademik etkinlik arama diğer iki aramadan farklı olarak akademik personelin akademik etkinliklerini aramak içindir. Diğer iki yöntemden elde edilen sonuçlar üzerinde veya direkt olarak akademik etkinlik konularına göre arama gerçekleştirebilir. Sonuçlar üzerinde tekrar arama yapılarak (refine) kriterler istendiği kadar daraltılabilir ve aranan kişilere ulaşılabilir. Akademik kullanıcıların etkinlik verilerini girişleri sırasında belirttikleri anahtar kelimeler (keywords), yayın künyeleri ve yayın yılları bu arama yönteminde kullanılır. Tüm akademik etkinlik ana konularına göre arama yapılabilir. Örneğin; Mühendislik ve Mimarlık Fakültesi'ndeki tüm doçentleri hızlı arama yaparak listeledikten sonra bunların içerisinde, bu yıl içerisinde Science Citation Index tarafından taranan dergilerde yayınlanmış makalesi olanları ve bu makalelerinin künyelerinde ve anahtar kelimelerinde "computer network" ve "CPU" terimlerinden her ikisini birden içeren kişileri listeleyebiliriz. Sonuçlar üzerinde uluslararası ödülü olanları ve son 3 yıl içinde ulusal bildiri yayınlamış olanları göster şeklinde sorguyu genişletebiliriz. Bu şekilde yüzlerce sorgu kombinasyonu düşünülebilir. Tabi ki ileride bu veriler dahilinde çok daha detaylı sonuçlara ihtiyaç duyulursa, sistem üzerinde daha karmaşık ve uzun sorgular üretilebilir.

#### 4.7.4 İstatistikler

Sistemin İstatistikler bölümünde kullanıcılara sürekli güncellenen veri tabanı verilerine göre döküm yapılır. Bu bölüme giriş yetkisi tüm internet kullanıcılarındadır. İstatistikler için iki bölümden söz edilebilir. Bunlar şöyledir:

**Genel Kullanıcı Sayıları:** Bu istatistik bölümünde sistemde kayıtlı olan akademisyen sayıları birimler bazında görüntülenir. Bu akademisyen sayıları yükselme ve atama kriterlerine göre sürekli değişkenlik gösterebilirler.

**Akademik Yayın İstatistikleri:** Bu bölümde ise, sisteme kaydedilen makale, bildiri, kitap vb. gibi yayın istatistikleri görüntülenir. İstatistikleri tüm internet kullanıcılarına sunulur. Üniversite bünyesinde son iki yıl içinde yapılmış olan her türlü yayın istatistiği sayısal olarak ifade edilir. Bu sayısal ifade, her makale bildiri yada kitap tipine göre ayrı yapılır.

#### 4.8 Sistemde Dil Desteği

AÖBS'nin Sistem Yönetim bölümü dışındaki tüm bölümleri dil desteğini barındırmaktadır. Akademik Giriş, Arama ve Görüntüleme ve İstatistikler bölümleri, sunucu üzerinde belirlenmiş dil dosyalarına göre Türkçe ve İngilizce olmak üzere iki dilde çalışabilmektedirler.

##### 4.8.1 Dil Dosyaları

Sunucu üzerinde çalışacak olan dil dosyaları tanımlanırken, dil desteğini barındıracak olan tüm bölümler ve alt bölümlerine göre ayrı dosyalar halinde hazırlanmışlardır. Dil dosyaları basitçe içerlerinde sistemde kullanılan kelime veya tanımları barındırırlar. Bu kelime ve tanımlar PHP değişkenlerinde tutulurlar. Örneğin tüm sistem genelinde kullanılan kelime ve tanımlar için “turkce.lng” ve “english.lng” isimli 2 adet genel dil dosyası oluşturulmuştur. Benzer şekilde sistemin her alt bölümü için ayrı klasörlerde tutulan dil dosyaları vardır. Bir dil dosyasının

içeriği kullanılacağı (sayfada görüntüleneceği) bölüme uygun olarak aşağıdaki gibi değişken atamaları içerir.

```
$cvlang_maintitle = "ÇOMÜ – Academic CV System";  
$cvlang_guncelle = "UPDATE";  
$cvlang_duzenle = "Edit Record";  
.....
```

#### **4.8.2 Sistemde Dil Dosyalarının Kullanımı**

Hazırlanmış olan dil dosyalarının, kullanılacağı ilgili sistem bölümünden çağırılması için aşağıdaki komutun kullanılması yeterli olur.

```
require($cvlang_areafolder."lang_makale.lng");
```

Burada *\$cvlang\_areafolder* değişkeni, yüklenecek olan dile göre dil dosyasının bulunduğu dizin tanımını içerir (Örn: language\turkce).

#### **4.8.3 Yeni Dil Eklenmesi**

Sisteme ihtiyaç halinde sonradan eklenecek bir dil için, yapılması gereken Türkçe ve İngilizce için hazırlanmış olan dil dosyalarının kopyasını oluşturarak benzer dizin yapısını kurmaktır. Sonrasında dil dosyalarının içeriğinde bulunan değişkenler istenen dile göre uyarlanırlar. Bunun dışında tüm sistem üzerinde dil tanımlamaları da güncellenir.

#### 4.9 AÖBS ve AKTS DKS Etkileşimi

AKTS DKS ile AÖBS aynı sunucu üzerinde çalışmaktadırlar. AKTS DKS’nde herhangi bir ders için tanımlanan öğretim elemanı ve bu öğretim elemanın kurum sicil numarası AÖBS sistemiyle etkileşimi sağlar. AKTS DKS’nde listelenen bir derste eğer öğretim elemanı ve kurum sicil numarası girişi yapıldıysa, ders detay ekranında öğretim elemanı ismi yanındaki CV düğmesi aktif hale gelir ve bu sayede AÖBS’deki özgeçmiş görüntüleme arayüzüne direk bağlantı yapılır. Dolayısıyla ders bilgilerini ve detaylarını inceleyen kullanıcı dersi veren öğretim elemanı hakkındaki bilgilere de anında erişebilir. Bu ders detay ekranı aşağıdaki gibidir.



**ECTS** > **CCS** **Course Catalogue System** Canakkale Onsekiz Mart University

v.2 Course Catalogue System **CLOSE**

**COURSE DETAILS** **TÜRKÇE**

<b>Academic Unit :</b>	Faculty of Engineering& Architecture
<b>Department :</b>	Computer Engineering
<b>Date/Year of Study :</b>	2007-2008 / 2nd Year
<b>Term :</b>	Autumn
<b>Course Title :</b>	<b>OBJECT ORIENTED PROGRAMMING</b>
<b>Course Code :</b>	209
<b>Status :</b>	Compulsory
<b>Lecturer 1 :</b>	<b>CV</b>
<b>Language of Course :</b>	Turkish
<b>COMU Credit :</b>	3.0
<b>ECTS Credit :</b>	<b>6.0</b>
<b>Lecture hours per week :</b>	2
<b>Practical hours per week :</b>	2

Şekil 4.4 : AKTS DKS Ders detay ekranı



## **BÖLÜM 5**

### **SONUÇ VE TARTIŞMA**

Gerçekleştirilen çalışmaların sonucu iki temel başlık altında ele alınabilir: 1) Web uygulamalarının üniversite eğitimine yaptığı katkılar ve 2) günümüzde yaygın olarak kullanılan Web programlama dilleriyle hedef odaklı uygulama geliştirilmesi ve veri standardizasyonu.

Birinci başlık altında elde edilen olumlu sonuçlar şöyle özetlenebilir. Bu tez çalışmasında genel hatlarıyla anlatılmış olan iki sistem AKTS DKS (Ders katalog sistemi) ve AÖBS (Akademik Özgeçmiş Bilgi Sistemi) ile ve ÇOMÜ’de AB Bologna sürecine yönelik yapılmış diğer tüm çalışmalar sayesinde ÇOMÜ AB Diploma Eki Etiket (AB Diploma Supplement Label) hakkını elde etmiştir. Ayrıca söz konusu iki proje de üniversitede eğitim kalitesinin iyileştirilmesi yönünde yapılan akreditasyon çalışmalarına enformasyon sağlayarak önemli katkılarda bulunmuştur. Tüm bu katkılardan ayrı olarak, ÇOMÜ akademik personeli özgeçmişleri, Web sayfası sorumluları aracılığı ile akademik birimlerin altındaki sayfalardan AÖBS’ye bağlantı verilerek güncel ve sıkıntısız şekilde yansıtıldığı gözlemlenmektedir.

Web programlama dilleriyle hedef odaklı uygulama geliştirilmesi ve veri standardizasyonu başlığı altında elde edilen sonuçlar ise şöyle özetlenebilir. Her iki sistemin, özellikle 2010 yılında sona erecek olan AB Bologna Süreci kapsamında kayda değer katkısı olduğunun altı çizilmekle birlikte, Bologna Süreci’nin sonunda ortaya çıkması olası akreditasyon, eğitim ayarlanması (tuning) gibi süreçler için de temel oluşturması beklenmektedir. Buna ek olarak, genelde tüm üniversiteler özelde ÇOMÜ kapsamında “neyi, nasıl ve kiminle” sorularının yanıtlarının güncel, doğru ve hızlı biçimde verilmesini sağlayan bu iki sistemin, gelecekte yanıtlanması istenecek “izleme, yeniden değerlendirme ve öğretim programı geliştirme” ile ilgili soruların çıkış noktasını oluşturacağı ön görülmektedir.

Oluşturulan sistemlerin olumlu sonuçlarının yanında ortaya çıkması olası sorun ve sıkıntılar da yine “Bilişim teknolojisi sorunları” ve “Sosyal boyuttaki sorunlar” olarak iki başlık altında ele alınabilir.

Bilişim teknolojileri açısından her iki sistemin bakım-onarımı, geliştirilmesi, veri güvenliği ve yedeklemesi gibi ihtiyaçların ve artan veri boyutları sonucunda ortaya çıkacak donanımsal ve yönetsel gereksinimlerin karşılanması açık bir zorunluluktur. Buna bağlı olarak yaşanacak sorunların önceden veya başlangıç aşamasında tespit edilmesi ve gerekli önlemlerin alınarak en kısa zamanda çözümlerin üretilmesi gereklidir. Yine de doğru bilgiye en hızlı biçimde ulaşma yönünde yapılan bu çalışmalar teknolojik gelişmeler veya sonradan beliren ihtiyaçlar çerçevesinde her türlü yeniliklere ve geliştirmelere açıktır. AÖBS’nde ileride yapılması planlanan, girilen yayın verilerine göre akademisyenlerin akademik yükselme puanlarının otomatik hesaplanması işlemi bu gelişmelere bir örnek olarak gösterilebilir.

Sosyal boyut ile ilgili sorunların daha çok kullanıcı odaklı olarak ortaya çıkabileceği düşünülmektedir. Bilgisayar okur-yazarlığının genelin üstünde bir seviyede olmasını gerektiren her iki sistemin kullanımının, kullanıcı açısından sıkıntılar oluşturabileceği düşünülmektedir. Ancak, yine bu sıkıntıların sistemlerin kullanılması sonucu kurumsal bazda elde edilecek ECTS Diploma Eki Etiketleri ve diğer kazanımların getireceği motivasyon ile azalacağı düşünülmektedir.

Gerçekte her iki modülün diğer üniversite bilgi sistemleriyle birlikte AB Bologna Süreci dahilinde bir üniversiteler portalı biçiminde tasarlanması ve buna bağlı farklı, çoklu etkileşimli veritabanları oluşturulmasının daha fütürist bir yaklaşım olacağı açıktır. Her iki sistemin bu yaklaşımın başlangıç adımı olarak kabul edilmesi, gelecekteki vizyonun belirlenmesini sağlayacaktır.

## KAYNAKLAR

Achour, M., Betz, F., Dovgal, A., Lopes, N., Olson, P., Richter, G., Seguy, D., and Vrana, J., 2007. PHP Manual, Ed. Hojtsy, G. 1997 - 2007 the PHP Documentation Group, Retrieved August 10, 2007, <http://www.php.net/download-docs.php>.

Anonymous 2004. PEAR Reference Manual. Retrieved August 10, 2007, <http://pear.php.net>.

Anonymous 2007. MySQL Reference Manual. Retrieved August 10, 2007, <http://dev.mysql.com/doc/>.

Bologna Rehberleri 2004. Avrupa Birliđi Eđitim ve Gençlik Programları Merkezi Başkanlıđı. The Bologna Process Towards the European Higher Education Area (January 2004), 10 Eylül 2007, <http://www.bologna.gov.tr/documents/files/BolognaBasicInformation.doc>.

Converse, T. 2004. PHP5 and MySQL Bible. Hoboken, NJ, USA: John Wiley & Sons, Incorporated, 2004. Retrieved February 1, 2007, from <http://site.ebrary.com/lib/canakkale>.

Di Giacomo, M. 2005. MySQL: lessons learned on a digital library , IEEE Software, 22 (3):10-13.

Gosney, J. 2003. HTML Profesional Projects. Independence, KY, USA: Premier Press, Incorporated, 2003. Retrieved January 10, 2007, from <http://site.ebrary.com/lib/canakkale>.

Lavin, P. 2006. Object Oriented PHP. San Francisco, CA, USA: No Starch Pres, Incorporated, 2006. Retrieved April 22, 2007, from <http://site.ebrary.com/lib/canakkale>.

Meloni, J. 2003. PHP Essentials. Boston, MA, USA: Course Technology, 2003. Retrieved December 12, 2007, from <http://site.ebrary.com/lib/canakkale>.

Naramore, E. 2005. Beginning PHP5, Apache, and MySQL Web Development. Hoboken, NJ, USA : John Wiley & Sons, Incorporated, 2005. Retrieved March 14, 2007, from <http://site.ebrary.com/lib/canakkale>.

Paulson, L.D 2005. Building rich web applications with Ajax. Computer Volume 38, Issue 10, Oct. 2005.

Pfaffenberger, B. 2004. HTML, XHTML, and CSS Bible (3rd Edition). Hoboken, NJ, USA: John Wiley & Sons, Incorporated, 2004. Retrieved January 4, 2007, from <http://site.ebrary.com/lib/canakkale>.

Serrano, N., Aroztegi, J.P. (2007). Ajax frameworks for interactive web apps, IEEE Software 24 (5): 12-14.

Sheldon, R. 2005. Beginning MySQL. Hoboken, NJ, USA : John Wiley & Sons, Incorporated, 2005. Retrieved March 15, 2007, from <http://site.ebrary.com/lib/canakkale>.

Smith K. 2006. Simplifying Ajax-style Web development, COMPUTER 39 (5): 98-101 MAY 2006. Retrieved December 12, 2007, from <http://site.ebrary.com/lib/canakkale>.

Valade, J. 2004. PHP 5 for Dummies. Hoboken, NJ, USA: John Wiley & Sons, Incorporated, 2004. Retrieved October 10, 2007, from <http://site.ebrary.com/lib/canakkale>.

Zakas, N.C. (2005). Professional JavaScript for Web Developers. Hoboken, NJ, USA: John Wiley & Sons, Incorporated, 2005. Retrieved September 20, 2007, from <http://site.ebrary.com/lib/canakkale>.

Zoratti, I. 2006. MYSQL Security Best Practices , Crime and Security, 2006.  
The Institution of Engineering and Technology Conference on Jun. 2006. Retrieved  
December 24, 2007, from <http://site.ebrary.com/lib/canakkale>.

## ÇİZELGELER

Çizelge No	Çizelge Adı	Sayfa
Çizelge 4.1	4 veri tabanı mimarisi özelliklerinin kıyaslanması.....	25

## ŞEKİLLER

Şekil No	Şekil Adı	Sayfa
Şekil 2.1	– MySQL Sunucu Mimarisi .....	8
Şekil 2.2	– MySQL de hareket yönetimi .....	9
Şekil 2.3	– AJAX uygulama çalışma mantığı.....	14
Şekil 3.1	– Koordinatör paneli ders içerik durumu gösterimi.....	18
Şekil 4.1	– Veri tabloları ilişki yapısının genel gösterimi .....	31
Şekil 4.2	– Veri görüntüleme konu seçimi .....	44
Şekil 4.4	– AKTS DKS Ders detay ekranı.....	54

## YAŞAM ÖYKÜSÜ

### **Kişisel Bilgiler**

Adı Soyadı : Yücel ŞAHİN  
Doğum yeri ve yılı : Balıkesir / Bandırma-1982  
Adres : Çanakkale Onsekiz Mart Üniversitesi  
Bilgi İşlem Dairesi Başkanlığı  
Terzioğlu Yerleşkesi, Rektörlük 17100 ÇANAKKALE  
Tel : 0 544 372 52 25  
E-posta : yucelsahin@comu.edu.tr

### **Eğitim Durumu**

2001-2004 : Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği  
1999-2001 : Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Programcılığı  
1995-1998 : Bandırma Ticaret Meslek Lisesi, Bandırma

### **Akademik Unvanları**

2005 - : Çanakkale Onsekiz Mart Üniversitesi - Uzman

### **Mesleki Deneyim**

06.2004 - 09.2004 : Figen Yazılım, Ericsson Mobility World, Maslak/İSTANBUL  
11.2004 - : Çanakkale Onsekiz Mart Üniversitesi, Bilgi İşlem Dairesi Başkanlığı,  
Uzman

### **Çalışma ve İlgili Alanları**

İnternet, İnternet Yazılım, Web Tasarım ve Programlama, Programlama Dilleri,  
Veritabanı Yönetimi, Grafik Tasarım