

**T.C.
AKDENİZ ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
Biyostatistik ve Tıp Bilişimi Anabilim Dalı**

**İDARİ, EĞİTİM VE AKADEMİK
AMAÇLI MEDİKAL ENFORMASYON
PORTALI GELİŞTİRİLMESİ**

Yılmaz Kemal YÜCE

Yüksek Lisans Tezi

Antalya, 2006

**T.C.
AKDENİZ ÜNİVERSİTESİ**

SAĞLIK BİLİMLERİ ENSTİTÜSÜ
Biyostatistik ve Tıp Bilişimi Anabilim Dalı

İDARİ, EĞİTİM VE AKADEMİK
AMAÇLI MEDİKAL ENFORMASYON
PORTALI GELİŞTİRİLMESİ

Yılmaz Kemal YÜCE

Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Uğur Bilge

“Kaynakça Gösterilerek Tezinden Yararlanılabilir”

Antalya, 2006

Saęlık Bilimleri Enstitüsü Müdürlüğü'ne;

Bu alıřma, jürimiz tarafından Biyoistatistik ve Tıp Biliřimi Anabilim Dalı'nda Tıp Biliřimi yüksek lisans tezi olarak kabul edilmiřtir. / /

Tez Danıřmanı: Yrd.Do.Dr. Uęur Bilge
Akdeniz Üniversitesi
Tıp Fakóltesi
Biyoistatistik ve Tıp Biliřimi Anabilim Dalı

Üye: Prof. Dr. Osman Saka
Akdeniz Üniversitesi
Tıp Fakóltesi
Biyoistatistik ve Tıp Biliřimi Anabilim Dalı

Üye. Prof. Dr. Gültekin Süleymanlar
Akdeniz Üniversitesi
Tıp Fakóltesi
İ Hastalıkları Anabilim Dalı

Üye: Do. Dr. Utku řenol
Akdeniz Üniversitesi
Tıp Fakóltesi
Radyoloji Anabilim Dalı

Üye: Yrd. Do. Dr. K. Hakan Gülkesen
Akdeniz Üniversitesi
Tıp Fakóltesi
Biyoistatistik ve Tıp Biliřimi Anabilim Dalı

ONAY:

Bu tez, Enstitüsü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından uygun görölmüş ve Enstitü Yönetim Kurulu'nun / / 2006 tarih ve /..... sayılı kararıyla kabul edilmiřtir.

Prof. Dr. Nurettin OęUZ
Enstitü Müdürü

ÖZET

Internet'in gelişimi ile ortaya çıkan yeni teknolojilerin ulaştığı noktada, artık bilgi ve belge yönetimi/paylaşımı portal adı verilen ve "süper site" olarak da bilinen sanal ortamlarda gerçekleştirilmektedir. Portallar bilginin, "arama" işlevi ile kolayca ulaşılabildiği farklı servislerden oluşurlar. Bu servisler, farklı amaçlara yönelik bilgi ve belge paylaşımının niteliğini farklı kullanıcı tiplerine göre uyarlayarak sağlayabilirler. Portalların, özellikle organizasyonel açıdan sağlayabileceği kolaylıklar ile kurumlar çalışma ortamlarının temel işlevlerini bu ortamlara taşıyarak kurumiçi bilgiye erişimi kolaylaştırma ve iş verimliliğini artırma hedeflerine ulaşabilirler.

Bu çalışmada Biyoistatistik ve Tıp Bilişimi Anabilim Dalı'nda belge yönetimi için kullanılması tasarlanan; bunun yanında Akdeniz Üniversitesi Tıp Fakültesi öğrencilerine yönelik olarak tıbbi terimler sözlüğü ve akademik takvim servislerini içeren bir portal geliştirilmiştir. Belge yönetimi servisi, anabilim dalının işleyişine özel belge kütüphaneleri (gelen evrak, giden evrak, vd.) ve belge çalışma alanları (tez, ödev çalışma alanları) içermektedir. Ayrıca, kullanıcılara "çalışma alanı" olarak ayrılan bölümler bulunmaktadır. Bir kullanıcı çalışma alanı, kullanıcının tercihlerine göre uyarlanabilir niteliktedir ve kullanıcıya görev listesi, hatırlatmalar, duyurular gibi servisler verir. Tıbbi sözlük servisi tıbbi terimlerin anlamlarının Web üzerinden sorgulanmasına izin veren bir servistir. Bu servis tıpkı akademik takvim servisi gibi bir Web veritabanı uygulamasıdır. Akademik takvim servisi ise her akademik yıl için o yıl başlamadan belirlenen ders akışına, öğrencilere Web üzerinden ulaşma imkanı sağlamaktadır. Portalın belge yönetimi servisi için kullanılan temel platform, Microsoft SharePoint Portal Server (SPPS)'dir. Diğer servislerin ve Biyoistatistik ve Tıp Bilişimi Anabilim Dalı'nın resmi Web sitesinin geliştirilmesi için ise ASP.NET, JavaScript, HTML gibi Internet teknolojilerinin yanında veritabanı teknolojilerinden de yararlanılmıştır. Servislerin geliştirilmesinde 3 katmanlı (three-tier) yazılım geliştirme mimarisi kullanılmıştır. Servislerin gerektirdiği veritabanlarının veri modelleri ise Varlık-İlişki modeli ile tasarlanmıştır.

Bu çalışmada ayrıca tıbbi terimler sözlüğü servisinin, gerektirdiği veritabanı boyutunun giderek artması ile gerçekleştirdiği veritabanı işlemlerinin servis gecikmesine etkisi ölçülmüştür. Buna göre, farklı sayıda tıbbi terim saklayan veritabanları ile gerçekleştirilen ölçümler sonunda ortaya çıkan ortalama veritabanı işlem süresinin servis süresi dahilinde kabul edilebilir olduğu sonucuna varılmıştır.

Bu çalışmada medikal bilgiye erişimi sağlayan servislerden oluşan bir portal geliştirilmiştir. Bu prototip portala, Biyoistatistik ve Tıp Bilişimi Anabilim Dalı çalışanları ve Tıp Fakültesi öğrencilerinin kullanımına yönelik, gerek eğitim gerekse akademik tabanlı yeni servisler eklenmesi planlanmaktadır. Bu amaçla, kullanıcıların servis kullanım logları incelenecek, kullanıcıların servisleri kullanırken edindikleri deneyimler geribildirimlerle toplanacaktır.

Anahtar kelimeler: portal, medikal portal, belge yönetimi, tıbbi sözlük, Sharepoint Portal Server, ders takvimi.

ABSTRACT

The growth of the Internet has led to new technological advances that yielded new approaches to document and information management/sharing solutions called portals or super sites. A portal consists of different services providing search function. These services can allow customization of service properties with respect to different user types' different preferences. Portals may help organizations reach their targets by providing organizational facilities like easy and fast access of information, provided that they move their business and working environment into the portals.

In this study, a portal that consists of a document management service for the Biostatistics and Medical Informatics Department and two other services; medical dictionary and academic schedule; for the students of Medical School are designed and developed. For document management purposes, the portal contains specific document workspaces and document libraries for incoming documents, outgoing documents, thesis, and assignments. The portal assigns a "working space" for each user/employee. A typical working space contains task lists, contact lists and announcements. The medical dictionary service provides a querying facility for the meaning of medical terms. A user can query the meaning of a medical term using a simple Web-based interface designed using Microsoft's Web Forms technology. The academic schedule service allows students of Medical School to access the schedule of a specific date in an academic year. A student can query the schedule of a specific date based on the class and committee information using a stepwise interface, which was designed using Microsoft's Web Forms technology, as well. The platform that was adopted to realize the departmental document management and sharing service is Microsoft's Sharepoint Portal Server 2003. For the development of other two services three-tier software architecture was implemented. The data management layers of the services were designed based on the Entity-Relationship data model and the design was implemented using MS Access. The business and process management layers were implemented using ASP.NET platform with Visual Basic programming language. The presentation layers, which basically contain the interfaces, were implemented using HTML and Web Forms.

Additionally, in this study, the time that it takes to complete necessary database operations as the database for the medical terms dictionary service enlarges was measured in order to estimate the response time of the service to a visitor/user. As a result, the measurements realized using medical terms databases that contains different number of medical terms have yielded a tolerable mean response time for database operations.

In this study, a prototype portal that consists of services, which allows users/visitors access to medical information, has been developed. New services based on academic and educational needs will be added later. For this purpose, Web logs for service usage and feedbacks from the users/visitors will be collected and examined.

Keywords: World Wide Web, information services, documentation, etymology, vocabulary, dictionary, medical dictionary.

TEŐEKKÜR

Bu araŐtırmanın planlanması, projelendirilmesi ve gerçekteŐmesinde deęerli katkıları ve eleŐtirileriyle bana yol gÖsteren sevgili hocam Prof. Dr. Osman SAKA'ya, projenin yürütülmesinde katkılarını hiç esirgemeyen deęerli danıŐmanım Yrd.Doç.Dr. Uęur BİLGE'ye, özellikle kaynak arayıŐında deneyimlerini paylaŐarak beni yönlendiren Yrd. Doç. Dr. Kemal Hakan Gülkesen ve Dr. NeŐe ZAYİM'e, destek ve yardımları ile yanımda olan tüm Biyoistatistik ve Tıp BiliŐimi Anabilim Dalı AraŐtırma Görevlisi arkadaşlarım, Filiz İŐleyen, Özgür Tosun ve Esra Sümen'e ve sınırsız destek ve sabırları ile hep yanımda olan eŐime ve aileme minnetlerimi sunarım.

İÇİNDEKİLER

	SAYFA
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ	x
GİRİŞ	1
GENEL BİLGİLER	5
2.1. Portal ve Site Kavramları	5
2.2. Portalların Sınıflandırılması	5
2.2.1. Medikal ve Sağlık İçerikli Portallar	6
2.3. Bir Portalın Temel Bileşenleri	11
2.4. Bir Portalın Kurumsal Yararları	12
2.5. Portal Çözüm Teknolojileri	13
2.5.1 Microsoft Sharepoint Portal Server ve Windows Sharepoint Services	13
2.5.2. Sun Java System Portal Server	16
2.5.3 Portal Çözümlerinde Kullanılan Bazı Ortak Teknolojiler	21
GEREÇ VE YÖNTEMLER	18
3.1 Servisler Tanımlamaları ve Gereklilikleri	18
3.2 Servislerin Gerçekleştirimi İçin Tercihler ve Geliştirme Araçları	19
3.2.1 Belge Yönetimi Servisinin Gerçeklendirimi İçin Seçilen Portal Çözümü	19
3.2.2 Diğer Servisler İçin Benimsenen Geliştirme Prensipleri ve Gerçeklendirim Ortamı	20
3.3 Web Veritabanı Uygulama Geliştirme Mimarisi ve ASP.NET	20
3.3.1. 3-Katmanlı Uygulama Mimarisi	21
3.3.2. Veritabanı Modelleme ve MS Access	24
3.3.3. ADO.NET ve Web Formları	25
3.4 Tıbbi Terimler Sözlüğü Servisinin Veritabanı Boyutunun Servis Süresine Etkisi	26
3.4.1 Ölçüm Şartları ve Varsayımlar	27

BULGULAR	28
4.1. Tıbbi Sözlük Servisi	28
4.1.1 Sözlük Veritabanı Fiziksel Yapısı-Veri Yönetimi Katmanı	28
4.1.2 Tıbbi Terim Arama ve Ekleme Arayüzü- Sunum Katmanı	29
4.1.3 Servis Gerçekleştirmede Olaylar ve Olay İşleyiciler	32
4.1.4 Servis Algoritması	34
4.1.5 Servis Veritabanı İşlemlerinin Performans Ölçümü Sonuçları	36
4.2. Akademik Yıl Ders Takvimi Servisi	39
4.2.1 Ders Takvimi Veritabanı Fiziksel Yapısı	39
4.2.2 Akademik Yıl Ders Takvimi Sorgulama ve Ekleme Arayüzü	39
4.2.3 Servis Gerçekleştirmede Olaylar ve Olay İşleyiciler	43
4.2.4 Servis Algoritması	45
4.3. Anabilim Dalı Belge Yönetimi Servisi	47
4.4. Biyoistatistik ve Tıp Bilişimi A.D.'nin Resmi Web Sitesi	48
TARTIŞMA	50
5.1 Servislerin Genişletilmesi ve İyileştirilmesi	50
5.2 Tıbbi Terimler Sözlüğü Servisi için Servis Bilgisi Kalitesi	51
5.3 Servis/Yazılım Mimarisi Seçimi	51
5.4 Belge Yönetimi Servisi İçin Portal Çözümü Seçimi	52
5.5 Tıbbi Terimler Sözlüğü Servisinin Veritabanı Performansı	52
SONUÇLAR	53
KAYNAKLAR	55
EKLER	58
Ek I: Tıbbi Terimler Sözlüğü ve Akademik Takvim Ekleme ve Servis Kaynak Kodları	
ÖZGEÇMİŞ	87

ŞEKİLLER DİZİNİ

<u>SEKİL</u>	<u>SAYFA</u>
2.1. Yatay Türde Haber Portalı Örneği.	6
2.2. Portal Türlerinde İşbirliği Uygulamaları ve Karmaşıklık Seviyesi	7
2.3. Beslenme ve Diyet Portalı HPB Online	8
2.4. Hastaların Hekimleriyle Elektronik Ortamdan Haberleşmelerini Sağlayan Patient Gateway Portalı	9
2.5. John Hopkins Üniversitesi’de Tıp Fakültesi Öğrencileri ve Öğretim Üyeleri İçin Hazırlanan Portal	10
2.6. MinAstma; Astım Hastaları İçin Astım Takip Portalı	11
2.7. Windows Sharepoint Portal Server Mimarisi	14
2.8. Sharepoint’in Çalışma Prensipleri	15
2.9. Sun Java Enterprise System Mimarisi	16
3.1. İstemci/Sunucu İletişim Mimarisi	21
3.2. Servislerin Geliştirilmesinde Kullanılan Yazılım Mimarisi	23
3.3. Tıbbi Sözlük Servisi Veritabanı Tabloları	26
3.4. Ders Programı Servisi Veritabanı Tabloları	26
4.1. Tıbbi Sözlük Servisi Veritabanı Tabloları Arasındaki İlişkiler	29
4.2. Terimler Tablosundan Bir Görüntü	30
4.3. Anlamlar Tablosundan Bir Görüntü	31
4.4. Tıbbi Sözlük Servisi Arayüzü (Sunum Katmanı)	31
4.5. Tıbbi Sözlük Servisi’nin “Dize ile Başlayan Terimler Kriterine Göre Kullanılması Sonucu	32
4.6. Tıbbi Sözlüğe Terim ve Anlam Ekleme Arayüzü	33
4.7. Tıbbi Terimler Sözlüğü Servisinin Veritabanı İşlemleri Performansı	37
4.8. Günlük Ders Programı Servisi Veritabanı İlişkileri	40
4.9. Günlük Ders Programı Servisi Kullanım Arayüzü	41
4.10. Günlük Ders Programı Servisi ile Yapılan Bir Sorgu Sonucu	42
4.11. Akademik Yıl Takvimi Servisinin Ders Programı Ekleme Arayüzü	43
4.12. Belge Yönetim Servisi Giriş Sayfası	48
4.13. Bölüm Web Sitesi Ana Sayfası	49

TABLolar DİZİNİ

<u>TABLO</u>	<u>SAYFA</u>
3.1. 2-Katmanlı ve 3-Katmanlı Yazılım Mimarilerinin Karşılaştırılması	22
4.1. Veritabanında Saklanan Terim Sayısına Göre Servisin Gerektirdiği Veritabanı İşlemlerinin Tamamlanma Süresi Ortalamaları ve Standard Sapmaları	37

GİRİŞ

İnternet, ilk kurulduğu günden bu yana gelişiminde ulaştığı noktada, artık bilgisayarların bağlı olduğu basit bir bilgisayar ağı olmaktan çıkmış, giderek daha fazla insanın, hayatında daha fazla zaman ayırdığı ikinci bir habitat haline almaya başlamıştır. Temel gereksinimlerden “birbirleri ile iletişim”in ötesinde, artık insanlar günlük hayatlarının parçası olan aktiviteler için dahi İnternet’i kullanmayı bir refleks olarak hayatlarına dahil etmişlerdir. Diğer taraftan İnternet artık bilginin temel kaynağı olarak algılandığından giderek büyük ama düzensiz bir bilgi sistemi olma yolunda ilerlemektedir. İşte bu ilerlemenin düzenli bir halde yapılabilmesi için portallar geliştirilmeye başlanmıştır.

Portal denildiğinde aslında ilk akla gelen genellikle “site” terimidir. Teknik açıdan bir portal siteler topluluğu olarak açıklanabilir, fakat işlevsel anlamda son kullanıcı için yapılan tanımlamalarda; belli başlı bilgi kaynak ve servislerini bünyesinde barındıran ve onlara geçişi sağlayan ve yöneten “süper site”ler olarak adlandırılmaktadırlar. Bu tanımlamalarda geçen bilgi kaynak ve servisleri portalların tiplerine bağlı olarak değişse de bazı temel işlevler tüm portallar için ortak ve değişmezdir. Bu işlevlerin başında ise arama gelir (1). Portallar hiyerarşik yapıdadırlar ve belli bir düzen taşırlar. İşte bu yapı ve arama olanakları ile İnternet’in düzenli büyümesi ve böylelikle istenilen bilgiye ulaşılabilirliğin kolaylaşmasıyla İnternet’in bir büyük bilgi çöplüğü formundan kurtularak kullanılabilir bir bilgi kaynağı olmasına katkı sağlarlar. Bugün arama olanağının bulunmadığı site toplulukları portal olarak anılmazlar. Diğer taraftan portallar “düzenli büyüme”nin birincil çözümü olarak başka işlevler de yüklenmişlerdir. Örneğin kurumsal işleyişin bir parçası, hatta temel parçalarından biri haline almaları da bunun bir göstergesidir.

İnternet, Gelişimi ve Portalların Ortaya Çıkışı: İnternet, 1960’lı yıllarda bilimsel ve askeri alanlarda araştırma-geliştirme amaçlı bilgi paylaşımını bilgisayarlar üzerinden gerçekleştirebilme fikri üzerine geliştirilmeye başlanmıştır. İlk defa 1962’de J.C.R. Licklider tarafından fikir olarak ortaya atılırken “küresel bilgisayarlar ağı” betimlemesi kullanılmıştır (2). 1969 yılında, Amerika Birleşik Devletleri’ndeki 4 üniversitede (UCLA, Stanford Research Institute, UCSB, University of Utah) bulunan 4 mainframe’in birbirleri ile bağlantıya geçmesi ile bugünkü İnternet’in temelleri atılmıştır. Bu 4 bilgisayar ile kurulan ilk ağın adı ARPAnet’tir. Zamanla bu 4 bilgisayarın oluşturduğu ağa diğer araştırma-geliştirme merkezleri, kütüphaneler, enstitüler ve üniversitelerin bilgisayarları da bağlanmıştır (3).

ARPAnet’in geliştirilmesindeki birincil amaç, soğuk savaş döneminde muhtemel bir saldırı karşısında Amerika Birleşik Devletleri’nin askeri üsleri arasındaki bilgi alışverişini sağlamak olarak belirtilmiştir. Bir diğer amaç ise ARPAnet’i geliştiren ARPA (Advanced Research Projects Agency) adlı askeri ajansın başında bulunan Charles M. Herzfeld tarafından “ülkede oldukça az sayıda bulunan güçlü bilgisayarlara coğrafi ayrılık nedeni ile ulaşamayan birçok araştırmacının onlardan yararlanmalarını sağlamak” olarak bildirilmiştir.

Nitekim, İnternet'in ilk yıllarındaki kullanıcıları bilgisayar uzmanları, mühendisler, bilim adamları ve kütüphanecilerdi ve kullanabilmek için oldukça karmaşık bir sistemi öğrenmek zorundaydılar. İnternet, 1970lerde yeni servisler ve protokoller geliştirilerek eklenince olgunlaşmaya başladı. 1972 yılında İnternet'in gelişiminde önemli bir kilometre taşı olan ve Ray Tomlinson tarafından geliştirilen elektronik posta servisi kullanılmaya başlandı (4,5). Bunun yanında İnternet'in ilk temel iletişim protokolü olan ve oldukça kısıtlı sayıda bilgisayarın İnternet'e bağlanmasına izin veren NCP (Network Control Protocol) yerine bugün henüz kullanımda olan TCP/IP (Transmission Control Protocol/İnternet Protocol) adı verilen ve yerini zamanla IPv6 adındaki yeni protokol mimarisine bırakması beklenen protokol mimarisi geliştirildi (6,7). Fakat TCP/IP'nin benimsenmesi 1983'te gerçekleşti.

Bu tarihten sonraki gelişmeler İnternet üzerinde bilgi paylaşımını sağlayan, kullanımı kolay yeni teknikler üzerine odaklandı. E-posta, FTP ve Telnet adı verilen bilgi paylaşımı servislerinin standardize edilerek karmaşıklıklarından kurtarılması ile daha fazla insan İnternet'i kullanıma yöneldi. Bu yöneliş ise İnternet'teki bilginin giderek daha fazlalaşmasına ve daha fazla bilgi paylaşım alanının açılmasına neden oldu. İnternet'in giderek büyümesi, diğer bir ifade ile bilgi kaynağı alanların sayısının artması bilginin ve alanlarının takibini zorlaştırmaya başlamıştı. Bunun sonucunda İnternet'teki kaynakları dizinleme fikri doğdu ve bu çabanın ilk ürünü olarak 1989 yılında Archie adında, FTP sitelerini, herbirinin bulundurduğu dosyaları ve yazılımları listeleen yeni bir yazılım ve protokol çifti geliştirildi (8).

Yine aynı yılda, İnternet'in ve İnternet'te bilgi paylaşımının bugün bu kadar yaygın hale gelmesini sağlayan temel gelişme yaşandı. İnternet üzerinde bilgi paylaşımının ve bilgiye ulaşmanın çok daha kolaylaştırılabileceğine inanan Tim Berners-Lee ve arkadaşları bilgi paylaşımı ve dağıtımı için yeni bir servis ve protokol önerisi getirdiler. Aslında bu servis, Tim Berners-Lee'nin çalışmakta olduğu şirkete şirketin projeler sırasında ürettiği bilgi ve belgelerin çalışanlar arasında paylaşımını sürekli kılacak, kolaylaştıracak ve organizasyon ile birlikte büyüyecek bir "bağ" sistemi oluşturulması önerisinden türedi (9).

Hiyerarşik ve ağaç yapılı sistemlerden farklı olarak sistemin sınırlarını saklanan bilgi üzerine kurmaması gerektiğini ve bilgi depolanan her belgenin veya çalışanın bir diğerine ulaşılması için bir bağ içereceğini, böylelikle daireler ve oklarla oluşturulan yeni bir ilişkilendirme ortaya çıkacağını ileri sürüyordu Tim Berners-Lee önerisinde. Teorik olarak sistemde her belge veya insan bir daire (nod) ve içinde bulundurduğu ve bir diğer belge veya insana ulaşmayı sağlayan her bir bağ ise bir ok ile temsil ediliyordu. Tim Berners-Lee bu bağa daha önce başka çalışmalarda kullanılan "hypertext" adını veriyordu ve birbirleri arasında bağ içeren her türlü nodun bir şekilde ilişkili olduklarını fakat bu ilişkinin tipinin yapıda kesinlikle bağlayıcı olmadığını ama sistemin hem bu ilişkiyi hem de nodun türünü (insan, belge, vd.) bilmesi gerektiğini belirtiyordu önerisinde. Ayrıca bu sistemin temel parçaları olarak bir "Hypertext Sunucusu" ve "Tarayıcı" tanımlanıyordu. Tim Berners-Lee sistemin mimarisini o sıralarda yeni geliştirilen istemci/sunucu (client/server) mimarisi üzerine kurmuştu. Yapısal olarak Tim Berners-Lee'nin

sunduğu bu sistem bir Çizge Kuramı (Graph Theory) uygulaması olarak değerlendirilebilir (9).

Tim Berners-Lee'ye göre genelleştirilmişlik ve ulaşılabilirliğin, süslü grafik tekniklerden ve karmaşık servislerden daha önemli olduğu evrensel bir bilgi bağ sistemi kurulmalıydı. İşte bu amaçla önerdiği geliştirilmeye başlanan sistemin önemli yapıtaşlarından biri yazılım tipi olan “Tarayıcı”, Mosaic adı ile NCSA'da çalışan Marc Andreessen ve arkadaşları tarafından geliştirildi (2). Bu gelişme Web'in gelişimine büyük bir hız kazandırdı.

Bu hızlı gelişimin içerisine dahil olmaya çalışan ticari kuruluşlar İnternet için ve onun üzerinden işe yarar, daha doğrusu kazançlı ve karlı iş modelleri geliştirmeye koyuldular. Bilgi paylaşımı, tanıtım, reklam gibi amaçlarla geliştirilen siteler bir pazarlama alanı halini kazandı İnternet, sitelerdeki reklamların çoğalmasi ile İnternet üzerinden alışveriş ilk defa kitap ve müzik albümlerinin satışı ile başlatıldı ve büyük bir hızla ürün yelpazesi genişledi. Geliştirilen ve “işe yarayan” iş modeli ise portallardı. Fakat daha önce site kavramı ortaya çıktı. Bir site, Tim Berners Lee'nin ortaya attığı birbirine bağ içeren başlangıçta sadece metin içerikli belge tipi olarak tanımlanan sayfa denem ögeler kümesi olarak adlandırıldı (9). Daha sonraları bu tip belgeler resim ve ses gibi çokluortam öğeleri de içerecek şekilde geliştirildi. Bu belge tipine HTML adı verilen bir işaretleme dili kullanılarak hazırlandığından HTML belgesi adı da verilir. Dolayısıyla HTML İnternet'in ilk dili olarak da anılır.

Fakat HTML “statik içerik” hazırlanabilen bir dildir. Bu yüzden sunduğu olanaklar birçok açıdan (örneğin site ziyaretçisinin kişisel tercihlerine göre görüntülenme) sınırlı olan HTML'nin yapamadıklarını yapabilmek için, 90'ların ortalarına doğru dinamik içerik ve içerik yönetimi sağlayabilecek yeni teknolojiler geliştirilmeye başlanmıştır. Dinamik içerik yetisi istemci tarafında belli sınırlar dahilinde gerçekleştirilebilir bir hal almıştır. Bunu sağlayan Javascript ve Java Applet teknolojileridir. Bu iki teknoloji bugün sayfa ve servis tasarımlarında hala kullanılmaktadırlar fakat sağladıkları dinamik içerik yetisi istemci tarafında yapılabileceklerin kısıtlılığı ve özellikle de istemci tarafındaki güvenlik sorunu nedeni ile yetersizdir . Bu yetersizlik bilgisayar dünyasını sunucu tarafı teknolojiler geliştirmeye yöneltmiştir. İlk çalışmalar CGI, PHP, ASP ve JSP adında dört önemli teknolojiyi öne çıkarmıştır. Bu dört teknoloji öncelikle ziyaretçilerden sayfalardaki formlar aracılığı ile toplanan verileri sunucu tarafında işleyerek, sonuca göre HTML belgesi veya bir diğer deyişle sayfa üretmek sureti ile ziyaretçilerine dinamik içerik ve servisler sağlamak isteyenler için alternatif olmuşlardır. Bu teknolojilerden PHP, ASP ve JSP sonraları daha da geliştirilerek şu anda Web'in geneline hakim iki teknoloji halini almışlardır.

JSP ve ASP gibi teknolojilerin kullanımı Web veritabanı uygulamalarının geliştirilmesinin önünü açmıştır. Bu sayede İnternet üzerinden birçok servis verilmeye başlanmıştır. Yine de bu teknolojilerin sağladıkları avantaj ve olanakların sonucu ortaya çıkan ayrık servisler belli bir noktada yetersiz kalmaya başlamışlardır. İşte bu noktada portallar ortaya çıkmışlardır. Özünde portal kavramı Tim Berners Lee'nin 1989 yılında çalıştığı şirket olan CERN'e belge ve bilgi yönetimi önerisinin bir devamı niteliğindedir. O günden bugüne, özünde aynı kalmakla birlikte bugün bir

portal gelişen İnternet teknolojileri ile oldukça geniş bir işlev yelpazesine sahiptir. Bir portal artık bilgi ve belge yönetimi birimi olarak bilgi veya belgeyi görüntülemek isteyen kişinin kimliğine, görüntülenmek istenen bilginin niteliğine, vd. bakarak içerik, belge ya da bilginin yerinin belirlenmesi, saklanması, kişilere uyarlanması ve sunumu gibi birçok işlevi birarada yapabilen bir uygulamalar bütünü olmanın ötesinde, bu sayede sadece kurum ya da kuruluşların iç düzene sahip olmasına değil aynı zamanda İnternet'in de belli bir iç düzene sahip olmasını sağlayarak bir bilgi çöplüğü halini almasını engelleyen sanal bir organizmadır.

Bu çalışmada idari, akademik ve eğitim alanlarında belge ve bilgi yönetimi sağlayacak bir portal geliştirilmiştir. İdari olarak Biyoistatistik ve Tıp Bilişimi Anabilim Dalı'nın işleyişine adapte olabilecek ve belge yönetiminin verimli şekilde gerçekleştirilmesini sağlayacak sanal alanlardan ve anabilim dalının resmi sitesinden oluşan; akademik eğitim anlamında ise özellikle Akdeniz Üniversitesi Tıp Fakültesi öğrencilerinin tıbbi terimlerin anlamlarına ulaşmalarını ve eğitim yılı içerisinde herhangi bir güne ait ders saatlerini öğrenebilmelerini sağlayacak iki servis içeren bir portal hazırlanmıştır.

Bu tez raporunun geri kalan kısımları şu şekilde planlanmıştır; 2.bölüm olan Genel Bilgiler'de portal kavramının tanımı, türleri üzerinde durulduktan sonra medikal anlamda servislere sahip ve yayın yapan portallardan örnekler verilmiştir. Bu portallar ve kullanıcıları dikkate alınarak yapılan bazı bilimsel çalışmalar sunulmuştur. Ardından bazı portal çözümleri ve mimarileri açıklanmıştır. Araç ve Yöntem bölümünde portalda yer alması planlanan servislerin tanımları yapılmış, gereklilikleri saptanmıştır. Bu tanım ve gereklilikler göz önünde bulundurularak servislerin geliştirilmesi için kullanılmasına karar verilen portal çözümü, seçilen uygulama mimarileri ve geliştirme ortamları ve neden seçildikleri açıklanmıştır. Bulgular bölümünde geliştirilen servis ve sitelerin yapıları tanıtılmış, gerçekleştirim safhasındaki ara ürünler tanımlanmış ve servislerin kullanımları üzerine örnekler sunulmuştur. Tartışma'da ise portalın iyi bir servis yelpazesine sahip olabilmesi için yapılması gereken çalışmalar ve mevcut servislerin geliştirilmesi için alınması gereken kararlar üzerinde durulmuştur.

GENEL BİLGİLER

2.1. Portal ve Site Kavramları

Portal ve site kavramları birbiri içine geçmiş anlamlar barındırdığından kullanımları da çoğu zaman karışmıştır. Site kelimesi Türk Dil Kurumu tarafından hazırlanan “Bilgisayar Terimleri Karşılıklar Kılavuzu”nda “bölge” ve “küme” kelimeleri ile tanımlanmıştır. Portallar ise işte bu “bölge” ya da “küme”leri bir yığın halinde saklayabilen ve onlara, dolayısıyla buldukları belge ve bilgiye çeşitli servisler sayesinde geçişi sağlayan, kısaca onları yöneten ortamlardır.

2.2. Portalların Sınıflandırılması

Portallar ilk kez ortaya çıktıkları tarihten itibaren farklı şekillerde sınıflandırıldılar. Hedef kitle açısından sınıflandırma bunlar arasında ilk akla gelen olabilir. Hedef kitlenin büyüklüğü, portalın verdiği servislerle direkt olarak ilişkilidir.

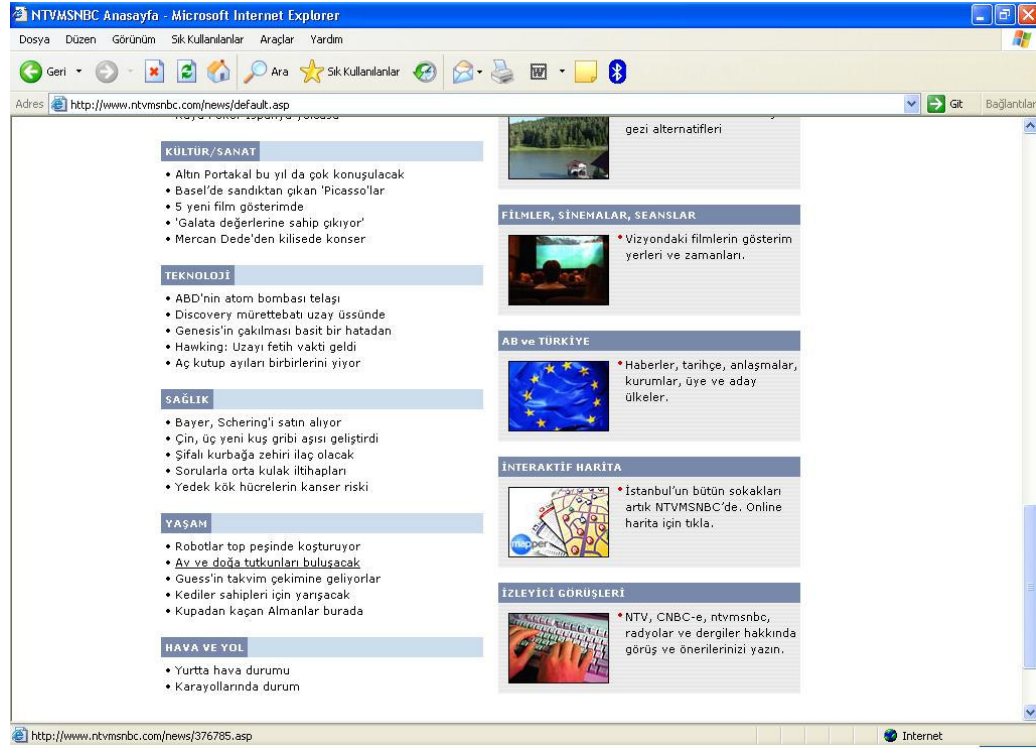
Buna göre portallar; (i) yatay portallar ve (ii) dikey portallar (iii) kurumsal portalları olmak üzere üç sınıfa ayrılırlar (10). Yatay portallar oldukça geniş bir kitleyi hedef alan Yahoo, Lycos, CNET, Microsoft Network gibi portallardır. Verilen servisler genellikle e-posta, haber, hobi ve ilgi alanlarını içerir. Elbette buna hedef kitle içindekilerin e-posta servisi dışında iletişimini sağlamak da dahildir; örneğin forumlar aracılığıyla. Şekil 2.1’de yatay türde bir portal örneği gösterilmektedir.

Dikey portallar ise belli bir hedef kitle hesaba katılarak geliştirilen portal türleridir. Kullanıcı popülasyonu, belli bir ortak paydaya sahip bireylerden oluşur ve içerik de bu ortak paydaya odaklıdır. Örneğin; belli bir alana ilgi duyanların oluşturduğu hedef kitle üzerine geliştirilen portallar dikey portal sınıfına girerler. Girişim portalları ise kurum ya da kurumu oluşturan birimleri hedef alır. Bu tür portallar genellikle şirket veya kamu kuruluşları için geliştirildiğinden hedef kitle çalışanlardır. Fakat elbette kurum dışı kullanıcılar da gerekli görülürse geliştirme sürecinde dikkate alınır.

Bunun yanında portallar içerik temelli de sınıflandırılırlar. İçerik temelli sınıflandırmada esas olan sağlanan servis ve içeriğin karmaşıklığı olduğundan özünde bu sınıflandırma karmaşıklık temelli bir sınıflandırmadır. Karmaşıklık seviyesi portalların “birlikte çalışma” ya da “işbirliği” için sağladığı olanaklar ile doğru orantılıdır. Portalların sağladığı işbirliği olanakları ve karmaşıklık seviyesi arasındaki ilişki Şekil 2.2’de gösterilmektedir.

İçeriğe göre sınıflandırıldıklarında portalları (i) kişisel portallar, (ii) iş portalları ve (iii) bilgi-tabanlı portallar olmak üzere üç ayrılırlar. Kişisel portallar, jenerik içerik sağlayan, ziyaretçinin bulunduğu yer veya ilgi alanlarına göre

şekillendirilebilen, haber, hava durumu, spor içerikli servisler sunan ve arama motorlarına bağlantılar barındıran portallardır.



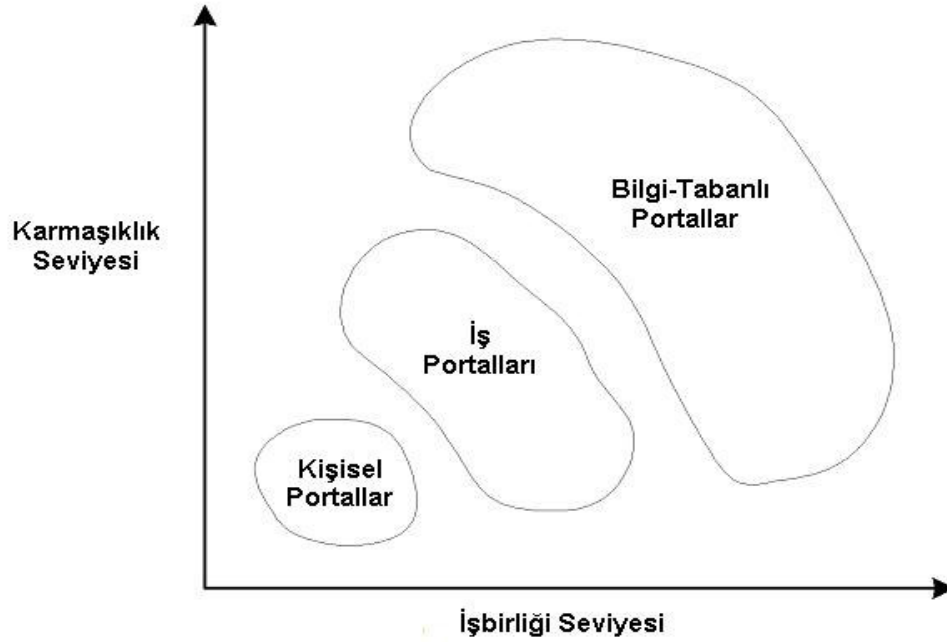
Şekil 2.1. Yatay Türde Haber Portalı Örneği (11).

İş portalları ise iş ve iş yönetimi içeriği sunan (örneğin; kaynak takvimlendirme ve yönetimi), küçük ölçekli işbirliği ve sınırlı kişiselleştirme ve uyarlanabilme yetenekleri barındıran tipte portallardır. Bu tip portallar aynı zamanda bilgi ve belgeler için gelişmiş arama olanaklarına da sahiptirler. Bilgi-tabanlı portalları diğerlerinden ayıran en önemli özellikleri ise geniş işbirliği yetenekleri barındırmasıdır. Bunun yanında öne çıkan bir diğer özelliği ise gelişmiş arama servisleridir. Bilgi-tabanlı portallarda her tür belge, iletişim bilgisi, kısacası kurumsal bilgi yığını içerisinde istenilen bilgiye kolayca ulaşmayı sağlayan gelişmiş arama olanakları bulunur.

Bu çalışmada geliştirilen portal iş-tabanlı bir portal olarak ve bu tipte portal geliştirmeye olanak veren araçlarla geliştirilmiştir. Bu araçlar ve kullanılan geliştirme yöntemleri 3. bölümde ele alınmıştır.

2.2.1. Medikal ve Sağlık İçerikli Portallar

Bu çalışma sırasında sağlık alanında veya tıbbi alanda hizmet veren portallar incelenmiştir. İncelenen portallar farklı hedef kitlelere, farklı hizmetler sunmaktadır.

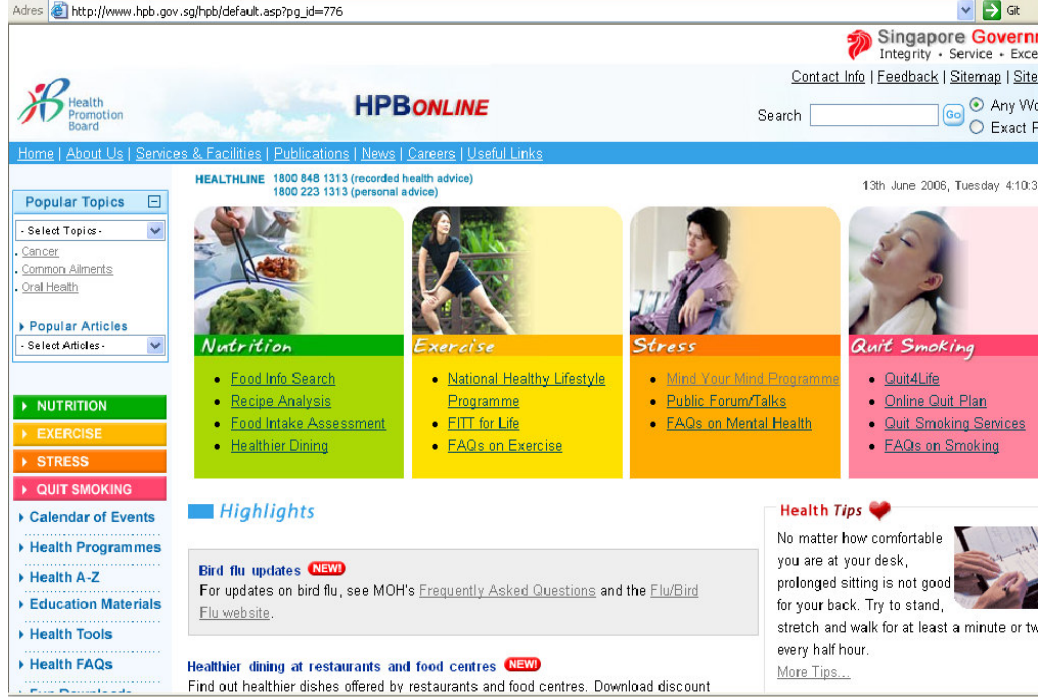


Şekil 2.2. Portal Türlerinde İşbirliği Uygulamaları ve Karmaşıklık Seviyesi

2001 yılında Singapur'da Sağlık Destek Kurulu (Health Promotion Board) bir portal geliştirmiş ve sağlık eğitimi vererek halkı bilinçlendirme yollarına İnternet'i eklemiştir. Sağlık eğitimi portalı, genel olarak, gıda bilgisi arama (beslenme esasları, 6000 farklı gıdanın kompozisyonu, haftalık beslenme ihtiyaçlarının karşılandığına dair kişisel değerlendirme, vd.), hastalıklar üzerine bilinçlendirici makaleler ve arama, günlük sağlıklı yaşam ipuçları, uzmana sor gibi servislerden oluşmaktadır. Portal İngilizce dilinde servis vermektedir (12).

2006'da Singapur Medikal Jurnal'inde, HPB Online adlı bu portalın halk tarafından Ocak 2002 ve Aralık 2004 tarihleri arasında kullanımı ve benimsenmesi üzerine yapılan bir çalışma yayımlanmıştır (13). Bu çalışmada, aylık ziyaret sayıları dikkate alınarak yapılan istatistiksel hesaplamalara göre, regresyon analizi sonucu, hem aylık ziyaret sayısındaki artış hem de tekrar ziyaretleri istatistiksel olarak anlamlı bulunmuştur. Yine aynı çalışmada, anlık bir ölçüm olarak, Nisan 2003 yılında baş gösteren SARS salgınında ziyaret ve tekrar ziyaret sayısındaki ciddi artış portalın kullanımının geliştirilme amacına hizmet ettiğinin bir göstergesi olarak sunulmuştur. HPB Online hali hazırda çevrimiçidir ve kullanılmaktadır. Şekil 2.3'te HPB Online'in giriş sayfası gösterilmiştir.

Sağlık alanında hizmet veren bir başka portal ise Patient Gateway (PG) adını taşımaktadır. Patient Gateway, hastaların hekimleri e-posta tipi mesajlaşma ile iletişim kurmalarını, randevu ve sevk almalarını sağlayan hizmetler sunmaktadır. Ayrıca hastalar kendi tıbbi bilgilerinin bir bölümüne istediklerinde portal aracılığı ile ulaşabilmektedir.



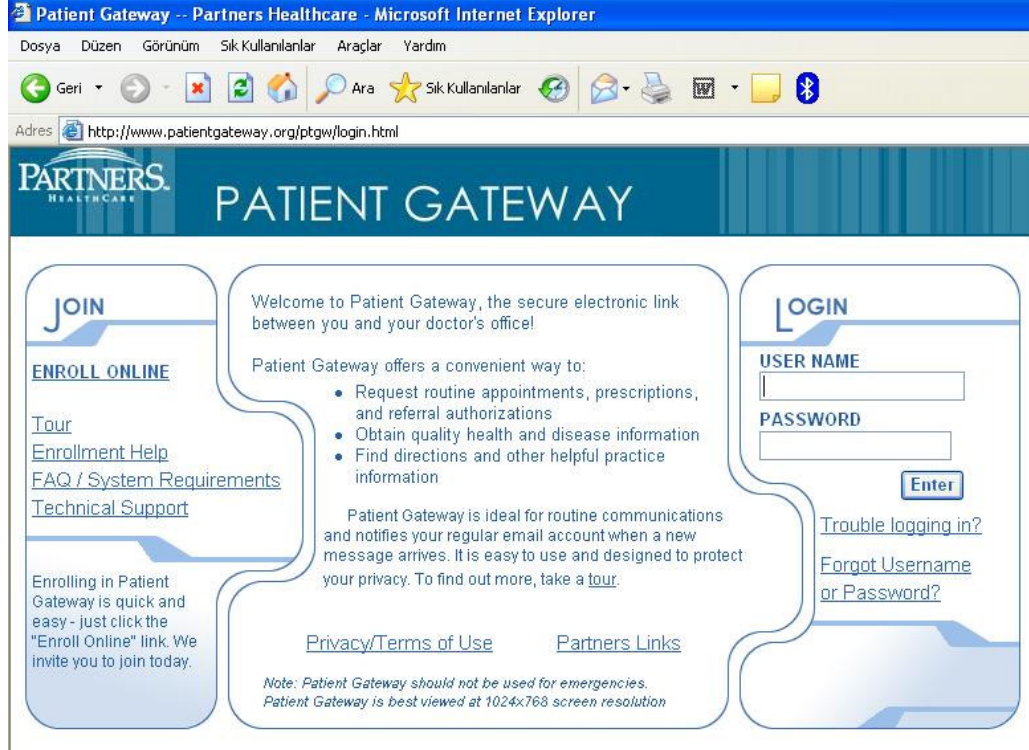
Şekil 2.3. Beslenme ve Diyet Portalı HPB Online.

Bunun yanında portal bir çevrimiçi sağlık kütüphanesi servisi de vermektedir. Şekil 2.4’te hali hazırda çevrimiçi durumda olan Patient Gateway’in giriş sayfası görülmektedir (14).

Anne F. Kitler ve arkadaşları tarafından yapılan ve 2004 yılında Informatics in Primary Care isimli jurnalda yayınlanan bir çalışma hekimlerin, hastalarının onlarla PG’deki elektronik iletişim servisi aracılığıyla iletişim kurmalarına yaklaşımları ve PG’nin çalışma düzenleri üzerine etkileri hakkındaki görüşlerini araştırmıştır (15).

Bu çalışmada 43 hekime, altı aylık PG kullanımlarını takiben bir anket postalanmış ve yirmidört hekim geri bildirim yapmıştır. Ankete katılan hekimlerin %84’ü sevk isteklerinin idaresinin PG sayesinde geliştiği, %71’i PG’nin ofis verimliliklerini artırdığı, ve yine aynı oranda hekim PG randevu alma işleminin de etkin olduğu yönünde fikir beyan etmiştir (15).

Bir başka portal çalışması ise hali hazırda çevrimiçi olan ve John Hopkins Üniversitesi’nde, “çevrimiçi tıbbi kütüphane” fikrinden hareketle geliştirilen MyWelch’tir (17). Kullanıcılarına, bünyesinde barındırdığı elektronik tıbbi kaynaklar arasında gelişmiş arama imkanı sunmasının yanında İnternet üzerinde ulaştıkları kaynakların bağlantılarını saklama, kendi aralarında bilgi alışverişini sağlayan elektronik ortamlar sağlayan bir portaldır aynı zamanda.

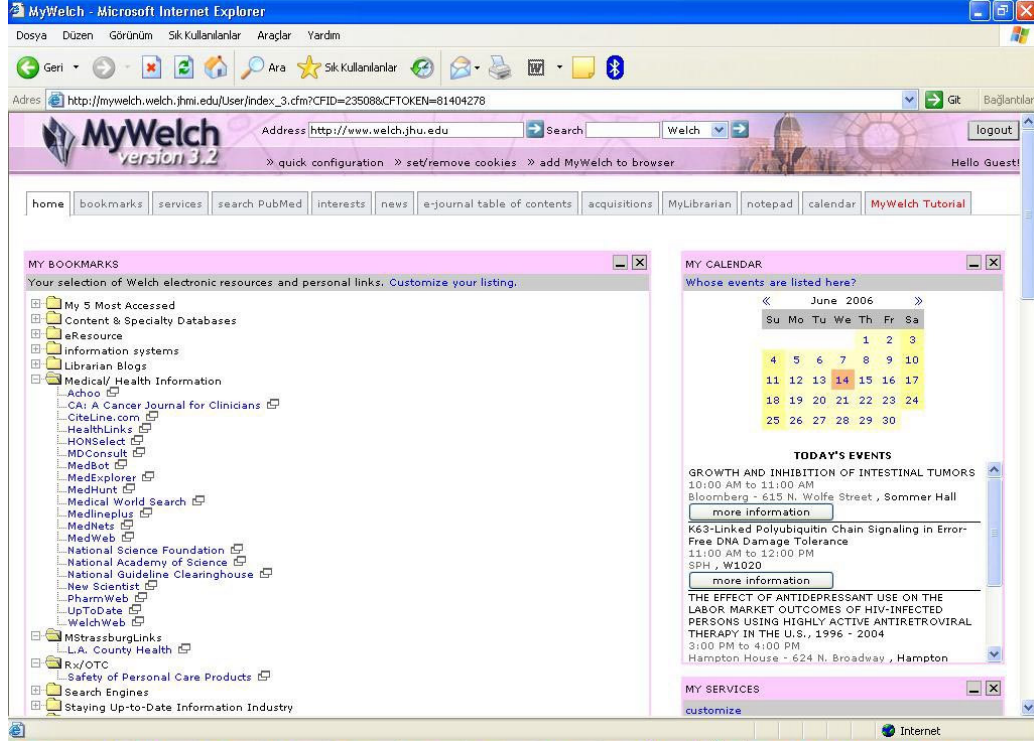


Şekil 2.4. Hastaların Hekimleriyle Elektronik Ortamdan Haberleşmelerini Sağlayan Patient Gateway Portalı (14).

MyWelch, servisleri ve ilişkili arayüzleri kullanıcı tercihlerine uyarlanabilir ve kişiselleştirilebilir bir portaldır. MyWelch'in hedef kitlesi John Hopkins Üniversitesi Tıp Fakültesi öğrencileri, öğretim üyeleri ve görevlileri olarak belirlenmiştir. Şekil 2.5'te MyWelch'in bir kullanıcı çalışma alanı sayfası gösterilmektedir (16,17).

2000 yılında yapılan bir başka çalışmada, astım tedavisinde kullanılan astım takip ve değerlendirme süreci, bir portalın geliştirilme temelini oluşturmuştur. LinkMedica adındaki bu portal, astımlıların kağıt kalem marifeti ile tuttıkları "astım günlüğü" adı verilen ve gün içi temel astım göstergelerini kaydettikleri sistemin elektronik ortama taşınmış halini temsil etmektedir (18). Portal üzerinden astımın takibi, hastanın soluk verme akımı tepe değeri değişkeni başta olmak üzere diğer birkaç değişkeni (kullanılan ilaç dozajı, gün içinde astım semptomlarının ortaya çıkıp çıkmadığı, vd.) portalı her gün en az bir kez ziyaret ederek kaydetmesi ile mümkündür.

Portala kayıtlı hastalar, günlük verilerini bir elektronik form aracılığı ile her kayıt edişlerinde sistemden geçmiş ve henüz girdikleri verilere dayanarak astımlarının kontrol altında olup olmadığına dair hızlı bir cevap alırlar. Eğer hastanın astımı kontrol altında değilse, hastaya yapması gerekenleri detaylı şekilde açıklayan bu cevap; örneğin hasta iki gün ardarda yüksek astım semptomları taşıdığını rapor etti ise önleyici nitelikteki kortikosteroid kullanım dozajının iki katına çıkarılması gibi önerilerde bulunmaktadır (19).



Şekil 2.5. John Hopkins Üniversitesi'nde Tıp Fakültesi Öğrencileri ve Öğretim Üyeleri için Hazırlanan Portal (16)

Bu servis, LinkMedica'nın Astım Yönetim Merkezi adı verilen bölümlerinden birini oluşturmaktadır. Bir diğer bölümü ise astım ve allerji üzerine haber, bildiri ve makale niteliğindeki elektronik formattaki belgeleri içeren, "Bilgi Merkezi" adlı bölümdür. Portalın bir diğer bölümü ise hastaların, hemşirelerin, hekimlerin buluşup fikir alışverişinde bulunabildiği forumdur. Portalın forum ve bilgi merkezi servisleri herkese açıktır. Astım Yönetim Merkezi ise hastaların ya da sağlık görevlilerinin kayıt olmasını gerektirmektedir (19).

LinkMedica, Danimarka ve Büyük Britanya'da, Danca ve İngilizce olmak üzere iki farklı dilde servis vermiştir. Hali hazırda LinkMedica, Danimarka'da MinAstma adı ile servise devam etmektedir. Büyük Britanya'da ise kapatılmıştır. Şekil 2.6'da MinAstma'nın Danimarka'da halen çevrimiçi olan sürümünün giriş sayfası görülebilir.

Jacob Anhoj ve arkadaşları tarafından 2004 yılında, temel amacı Web-tabanlı bir astım takip ve değerlendirme sisteminin kullanıcılar tarafından nasıl algılandığını belirlemek ve günlük hayatlarına nasıl bir etkileşim getirdiğini ölçmek olan bir çalışma yapılmıştır (19). Buna göre, LinkMedica'ya üye olarak Astım Yönetim Merkezi servisinden yararlanan gönüllü hasta ve sağlık görevlileri arasında gerçekleştirilen anket ve mülakat sonuçları, servisin hastalar tarafından kullanım istatistikleri ve girdikleri astım günlüğü verilerinin istatistiksel ve tıbbi değerlendirmesi sonucu LinkMedica'nın genel olarak güvenilir bir portal olarak algılandığı, akciğer fonksiyonlarını ve bronşiyal hiperaktiviteyi geleneksel tedavi

yöntemlerine nispeten iyileştirdiği tespit edilmesine karşın hastaların belli bir süre sonunda servisi kullanımlarında ortaya çıkan ciddi azalmanın temel sebebinin genelde İnternet'in, özelde ise LinkMedica'nın insanların günlük hayatlarına kolaylıkla entegre olamaması olarak belirlenmiştir. Bu nedenle LinkMedica'nın insanların yaşam şartlarına kolaylıkla adapte edebilecekleri bir hal alması gerektiği sonucu çıkarılmıştır (19).

Şekil 2.6. MinAstma; Astım Hastaları İçin Astım Takip Portalı (18)

2.3. Bir Portalın Temel Bileşenleri

Bir portalı oluşturan bileşenlerin başında site yığını ve kullanıcıları sayılmalıdır. Bunların yanında arama, uyarı gibi servislerle birlikte portala entegre edilen servisler ve uygulamalar sayılmalıdır. Fakat site yığınlarının bütün bu bileşenler arasında farklı bir yeri vardır. Siteler kendi içlerinde portalın kullanımını kolaylaştıran ve kurumsal yapıyı destekleyen bazı alt bileşenler içermelidirler.

Bu alt bileşenler site dizini, doküman kütüphaneleri, listeler ve çalışma alanlarıdır (20).

Site dizini; portalın içerdiği site yığını yönetmek, yığın içerisinde istenen siteyi bulmak, siteleri organize etmek gibi bazı işlevlerin gerçekleştirilmesini sağlayan bileşendir (20).

Doküman kütüphaneleri, uygun yetkilere sahip kullanıcılar tarafından belgelerin yaratıldığı, birlikte düzenlendiği, düzeltmeler yapılarak sürümlerinin

oluşturulduğu ve paylaşıldığı sanal ortamlardır. Doküman kütüphaneleri dizin içermelidirler ve arama işlevine sahip olmalıdırlar (20).

Listeler; duyurular, kontaklar, takvim, görevler ve herhangi bir türde bilgi saklamak amaçlı yaratılan hatırlatmaları içeren sonradan hazırlanmış alt bileşenlerdir. Kontaklar bulunduğu sitenin sahibinin direkt kontak kurduğu kişilerle iletişim kurmasını sağlayacak bilgileri sağlar (örneğin e-posta adresi). Duyurular, kurum içi yazışmalar gibi bilginin hızlı yayılmasını sağlayan yollardan biridir (20).

Takvim ve görevler listesi ise sitenin sahibi olan çalışan ya da birimin ajandası niteliğindedir.

Çalışma alanları, belge çalışma alanları ve toplantı çalışma alanları olarak ikiye ayrılır. Her iki tip alanda da işbirliği çalışması için araç gereçler bulunmalıdır. Belge çalışma alanlarında doküman geliştirmeye yönelik araçlar bulunurken toplantı alanlarında gerekli bilgi kaynakları ve dokümanlar saklanabilir. Bunun yanında listeler, üye bağlantıları gibi öğeler de içerebilirler (20).

2.4. Bir Portalın Kurumsal Yararları

En basit hali ile bir portal, insanları; örneğin bir şirketin çalışanlarını ve bilgiyi birbirine bağlamak, ulaştırmak için vardır. Bu temel işlev farklı profilde insanlar ile farklı tip ve yoğunluktaki, farklı yetki seviyesi gerektiren bilgilerin eşleştirilmesi olarak algılanmalıdır.

İşte bu eşleştirmenin bir sonucu olarak özellikle orta ve büyük ölçekli organizasyonlar için geliştirilecek portalların temel özellikleri arasında; entegrasyon, işbirliği, kişiselleştirme ve uyulanabilme yer almalıdır (21).

Entegrasyon, bir organizasyonun elektronik ortamdaki iş akışı için bir ihtiyaçtır. Organizasyondaki tüm bilgi ve uygulama sistemlerine tek bir yerden erişme yetisi kazandıran entegrasyon, portalın ölçeklenebilir olmasını ve dağıtık bir mimariyi (örneğin; coğrafi olarak ayrı yerlerde bulunan bilgi adacıkları) desteklemesini gerektirir. Bu sayede bir portal, organizasyonun elektronik ortamda da ihtiyaçlarına göre büyümesini sağlayan bir yapıya sahip olur (21).

Bir organizasyonun çalışanları arasındaki organizasyonel yapı işleyişin önemli bir parçasıdır. Çalışan seviyesinde bilgi paylaşımı iş akışının bir gerekliliğidir. Fakat daha önemli ve zor olan çalışan gruplarından oluşan takım, departman, kurul gibi yapısal birimlerin birbirleri ve üyeleri arasındaki iletişimi ve gerekli bilgiye zamanında ulaşabilmesidir. İşbirliği, bilgi, deneyim paylaşımı ve iletişim gerektirir. Portallar, işbirliği ve bu tip iletişim için sanal “çalışma ortamları”nı bir çözüm olarak bünyesinde barındırır. “Çalışma ortamları” sanal toplantı salonları olarak algılanabilir. Bunun yanında portallar, her bir kurumsal veya yapısal birime ait belge ve bilgilerin saklandığı ve çalışanların veya dışarıdan ziyaretçilerin yetkilerine göre ulaşma hakkına sahip oldukları sanal “doküman kütüphaneleri” ve siteler de işbirliği yapılarak belgelerin hazırlandığı, sürümlendiği, kontrol edildiği, onaylandığı ve paylaşıldığı ortamlar sunarlar. Birimler arası ve birimlere ait çalışma ortamları, doküman kütüphaneleri ve siteler sayesinde aktif

iletişim ve paylaşım içinde iş verimliliğine özellikle de aynı işin tekrar yapılmamasını sağlayarak katkıda bulunurlar.

Portalların çalışan ve kurumsal birimler seviyesinde bu kadar işlevsel olmaları beraberinde bazı kullanım zorlukları getirmektedir. Kullanımdaki bu zorluklar portalların servis veya işlevini kullananın tercihlerine göre görüntülenmesi sayesinde aşılır. Çalışanların veya kurumsal birimlerin sitelerinde periyodik olarak gerçekleştirilen görevlere bağlı olan çalışma ortamları, doküman kütüphaneleri gibi sanal ortamlara ya da sık kullandıkları arama ve uygulama servislerine kolayca ulaşmalarını sağlayacak bağlantılar ve bilgiler ekleyebilirler. Kullanıcılar ve birimler sitelerini oldukça kolay şekilde tasarlayarak isterlerse sitelerinin çalışma alanı işlevselliğini hatırlatmalar, listeler ekleyerek genişletebilirler. Kişiselleştirme ve uyarlanabilme, kullanıcılara özel kullanım alışkanları açısından da yarar sağladığından, çalışanların elektronik ortamda çalışma ilkesini benimsemelerine yardımcı olur. Kurumlar bu özellik sayesinde önemli bilginin kurum içinde dağıtımını da gerçekleştirebilirler. Bunun yanında portal içeriğindeki bilgilere ulaşım da hızlanır (21).

2.5. Portal Çözüm Teknolojileri

Bir portalın geliştirilme sürecindeki en önemli adımlardan biri modelleme adımıdır. Tasarımı yapılan bir portalın gerçekleştirimi için iki yol izlenebilir. Birincisi modellemenin gerekliliklerini tamamı ile yerine getiren bir gerçekleştirim yapmaktır. Seçilen yazılım geliştirme sürecinin belli başlı adımlarının ardından, gerçekleştirim yolunun benimsenmesi beraberinde alınması gereken yeni ciddi kararlar (gerçekleştirme platformu ve araç gereçlerinin belirlenmesi, vd.) ve beyin gücü, zaman gibi maaliyetler de getirir. Alternatif yol ise mevcut portal sunucu teknolojilerinden yararlanmaktır. Portal geliştirmek için bir portal sunucu teknolojisine kesin bir ihtiyaç olmasa da birçok alt süreç, portal sunucusunun kullanımı ile ortadan kaldırılabilir.

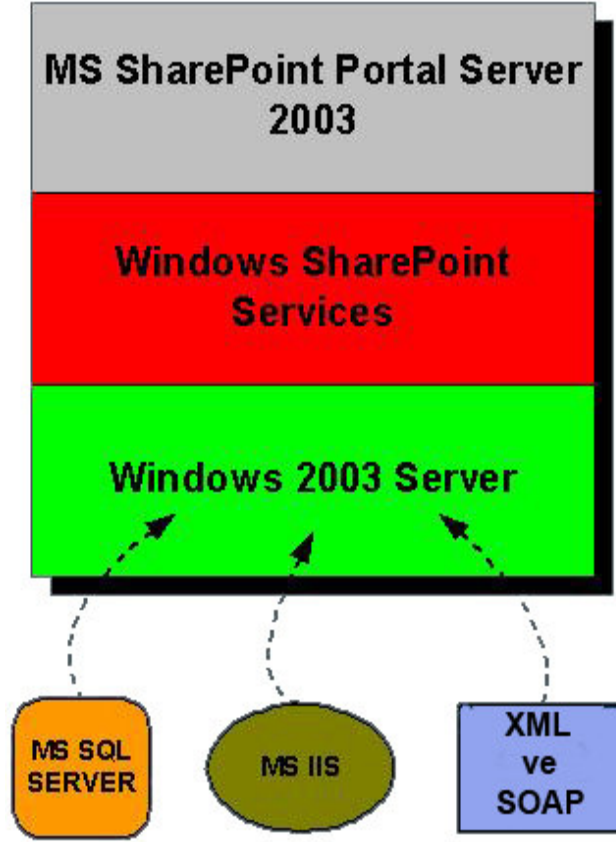
Bugün piyasada birçok şirketin portal geliştirme ve sunucu çözümleri bulunmaktadır. Bunlar arasında Microsoft'un ürünü olan Sharepoint Portal Server, Sun'ın geliştirdiği ve ücretsiz kullanımına izin verdiği Sun Java Enterprise System, IBM'in WebSphere Portal Server, ve Plumtree Corporate Portal adlı çözümleri sayılabilir. Mevcut çözümler, her ne kadar ortak temel teknolojiler benimsenerek geliştirilmiş olsalar ve aralarında ciddi benzerlikler gösterseler de ciddi farklılıklar da barındırmaktadırlar. Bu çözümlerden bazıları bu bölümde incelenecektir. Ayrıca bu bölümde incelenen çözümlerin ortak şekilde kullandıkları temel teknolojiler de açıklanacaktır.

2.5.1. Microsoft Sharepoint Portal Server

Microsoft, Sharepoint Portal Server ürününü ilk defa 1999'da piyasaya sürmüştür (20). Sharepoint Portal Server, bir portal çözümü olarak tüm işlevleri tek başına gerçekleştiren bir yapıda geliştirilmemiştir. Sharepoint Portal Server, Microsoft'un Windows Sharepoint Services adını verdiği bir platform üzerine geliştirilmiştir. Şekil 2.7'de Microsoft'un benimsediği geliştirme mimarisi gösterilmektedir (22).

Windows Sharepoint Services, motor katman görevindedir. Bu katmanda bir portalın tüm işlevselliği yönetilir. Örneğin, güvenlik bu katmanın temel işlevlerinden biridir. Windows Sharepoint Services aynı zamanda bir uygulama geliştirme platformudur. Bir portala yeni servis, uygulama ve site yığınlarının eklenmesi, kısaca bir portalın genişlemesi de bu katmanda, Windows Sharepoint Services teknolojisi kullanılarak gerçekleştirilir.

Microsoft, bir portalın servis veya uygulama genişlemesinin Sharepoint Services platformu temelli gerçekleştirimi için “Web Part” adı verilen temel bir teknoloji geliştirmiştir.



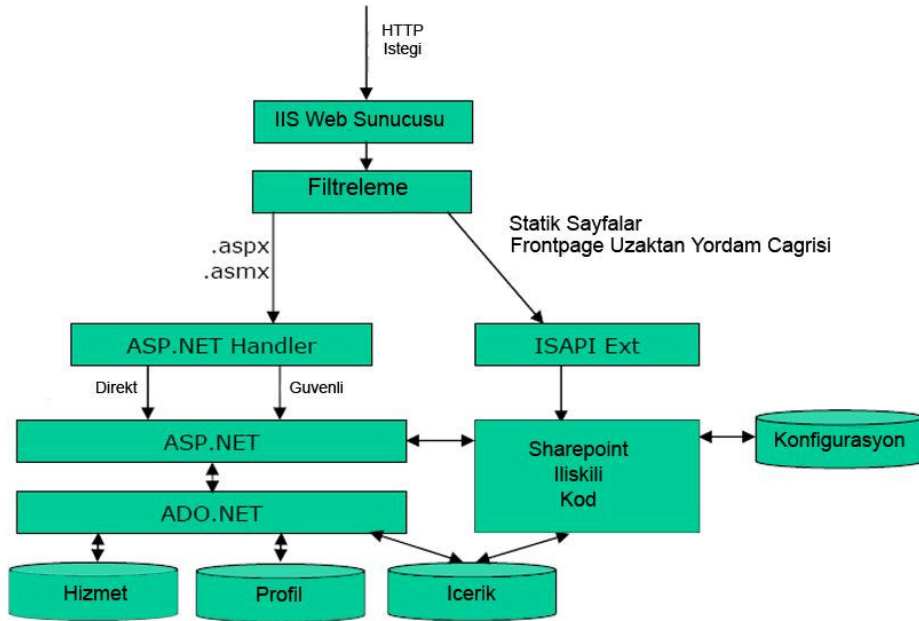
Şekil 2.7. Windows Sharepoint Portal Server Mimarisi

Temel olarak portalın geliştirilme amacına uygun olarak genel işleyiş prensiplerinin parçalar halinde eklenmesi ve sunucu kontrolleri haline alması için kullanılır bu teknoloji. Web Part teknolojisi Microsoft'un Collaborative Application Mark-up Language (CAML) adını verdiği bir işaretleme dili üzerine kurulmuştur. Bu işaretleme dili “işbirliği” gerektiren uygulama ve hizmetlerin geliştirilmesi için bir standard belirler.

Sharepoint üzerine kurulu olduğu mimarinin en alt katmanında ise modüler bir yapı gözetilmiştir. Bu katmanda portalın işlevselliği için gerekli olan temel

iletişim protokollerini gerçekleştiren sunucu teknolojileri ve veritabanı yönetim sistemi yer alır.

Bir sunucu topluluğu teknolojisi olan bu katman Windows Server teknolojisi olarak adlandırılmıştır. Windows Server teknolojisi, veritabanı yönetim sistemi olan Microsoft SQL Server, Web sunucusu olan Microsoft İnternet Information Server gibi sunuculardan oluşur. Bunların yanında XML ve SOAP gibi iletişim teknolojileri de bu katmanda gerçekleştirilmiştir. Sharepoint'in bu mimari üzerine kurulu çalışma ilkeleri Şekil 2.8'de gösterilmektedir.



Şekil 2.8. Sharepoint'in Çalışma Prensibi

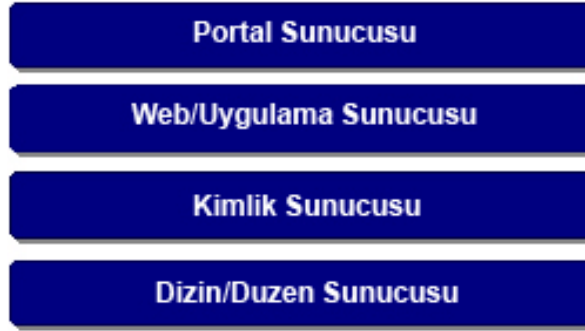
Sharepoint Portal Server'in en önemli özelliklerinden biri dağıtık sunucu mimarisini destekliyor olmasıdır. Sunucular kümelendirilerek verinin bulunabilirliği geliştirilebilir. Böylelikle, örneğin bir sunucu çökse de kullanıcılar belge, uygulama, servis ve verilere diğer sunucu veya sunuculardan kesintisiz ulaşmaya devam edebilirler. Sharepoint Portal Server bunun yanında servis ve uygulamalardan doğan işlem yükünün sunucular arasında dağıtımını yaparak güvenilirlik gerektiren servislerdeki yüklenmenin tepe altında tutulmasını başaran "Ağ Yük Dengeleme" teknolojisine de sahiptir.

Sharepoint Portal Server, belge-tabanlı işlevlerinde Microsoft'un masaüstü ve ofis uygulamaları ile sıkı bir şekilde entegre edilmiştir. Belgelerin yaratılması, değiştirilmesi, sürümlendirilmesi işlemleri Windows Office uygulamaları ile ilişkilendirildiğinden kolaylıkla gerçekleştirilebilir.

2.5.2. Sun Java Enterprise System

Sun firmasının bir ürünü olan bu sistem, oldukça modüler bir yapıda geliştirilmiştir. Portal sunucusunun sunabildiği servis ve işlevler modüler yapıdaki bu sistemde şekilde gösterilen mimaridedir.

Şekil 2.9'da da görüldüğü gibi portal sunucusu mimarinin en üst katmanında, diğer modüllerin üstünde bulunur. Web/Uygulama sunucusu, portalda yer alan/alacak Web uygulamalarının çalıştırılma ortamıdır. Bu katman portalın servislerinin genişletilebilmesini sağlamaktadır. Kimlik sunucusu, kullanıcı kimliklerinin yönetiminin yanında, ağ servis ve kaynaklarına uygun geçiş politikalarının yönetimini sağlar (23).



Şekil 2.9. Sun Java Enterprise System Mimarisi

Buna ek olarak kimlik sunucusu, kullanıcı profilleri, kullanıcı rolleri, kullanıcı haklarının da tanımlarını barındırır. Bu tanımlamalar sayesinde kullanıcıların kimliklendirme ve yetkilendirmeleri uygun şekilde yapılarak servis ve kaynaklara ulaşmaları yönetilir. En alt katmanda yer alan Dizin/Düzen sunucusu ise Kimlik sunucusu tarafından toplanan ve yönetilen bilgilerin saklama bileşeni olarak tasarlanmıştır (23).

2.5.3. Portal Çözümlerinde Kullanılan Bazı Ortak Teknolojiler

Yukarıda değinilen portal teknolojileri farklı şirketlerin farklı ürünleri olsa da bazı ortak temel teknolojiler kullanılarak geliştirilmişlerdir. Söz konusu temel ortak teknolojiler, kullanıcının portala bir noktadan tek güvenli girişinden (single sign-on) belge yönetimine kadar birçok kolaylığın geliştirilmesinde rol oynamışlardır.

XML: XML, özünde, basit ve esnek bir belge formatıdır. Geliştirilme amacı elektronik yayın, bilgi depolama ve dağıtma teknolojilerinin esneklik ilkesi ile genişletilebilmesi ve bu yapılırken tekniğin oldukça basit tutulmasının sağlanmasıdır. Fakat XML yapısı itibarı ile farklı teknikler için de kullanılabilir bulunduğundan, hali hazırda SOAP gibi iletişim protokolünün tabanını da oluşturmaktadır (24).

XML, temelde bir işaretleme dilidir. Ataları olan; Hypertext Mark-up Language (HTML) ve Standardized Generalized Mark-up Language (SGML) dikkate alınarak geliştirilmiştir. HTML'nin Web'de içerik sağlama yeteneklerinin kısıtlı olması nedeni ile zamanla yeni ve daha geniş yeteneklere sahip bir dile ihtiyaç duyulduğu fark edilmiştir. SGML ise oldukça geniş yeteneklere sahip ilk işaretleme

dili ve işaretleme dillerinin anası olarak öğrenilmesi zor ve karmaşık bir işaretleme dili olması nedeni ile İnternet’te kullanımı uygun görülmemiştir. İşte bu iki işaretleme dilinin dezavantajlarını yok eden; HTML kadar basit, SGML kadar yetenekli bir işaretleme dili olarak geliştirilmiştir XML.

XML, tek başına tek bir belge tipi tanımlamaz. Yeni belge tipi tanımlama ve türetmeyi sağlar. Diğer bir ifade ile yeni belge tipi tanımlama yeteneğine sahip bir dildir. Bu yeteneğe sahip olduğundan “genişletilebilir” bir işaretleme dili olarak bilinmektedir. Bu yeteneği XML tabanlı birçok yeni işaretleme dili geliştirilmesini sağlamıştır. Hali hazırda bilimsel verinin saklanması, işlenmesi için geliştirilmiş ve kullanılmakta olan birçok yeni işaretleme dili mevcuttur. Bunlar arasında Geographic Mark-up Language (GML), BSML (Bioinformatic Sequence Mark-up Language), Object Description Language (ODL) sayılabilir. Bu diller aynı zamanda XML’in bir alt kümesi olarak adlandırılırlar. XML bunun yanı sıra belge yönetim sistemlerinin ve işbirliği uygulamalarının da temelini oluşturmaktadır.

SOAP: En basit hali ile SOAP bir protokoldür. SOAP, merkezi olmayan, dağıtık mimariye sahip ortamlarda belli bir tipte ve yapılandırılmış bilginin ortamı oluşturan uygulama eşleri arasında değişimi için geliştirilmiş hafif bir mekanizma üzerine kuruludur. SOAP, gerçekleştirime veya uygulamaya özel anlamdizim tanımlamaz; uygulama anlamdiziminin ifade edilmesi için verinin modüller içinde kodlama mekanizmaları ve modüler paketleme modelleri tanımlar (25).

SOAP, XML kullanılarak geliştirilmiştir. Bir SOAP mesajı üç bölümden oluşur; zarf, başlık ve gövde. Bölümler farklı türde ve anlamda veriler taşır (25).

GEREÇ VE YÖNTEMLER

3.1. Servis Tanımlamaları ve Gereklilikleri

Portalın geliştirilme süreci tasarımı ile başlatılmıştır. Tasarım sürecinin ilk adımı, portalın ilgililerinin keskin çizgilerle belirlenmesidir. Buna göre; bu çalışmada geliştirilen portalın hedef kitesini Akdeniz Üniversitesi Tıp Fakültesi öğrencileri ve Biyoistatistik ve Tıp Bilişimi Anabilimdalı çalışanları oluşturmaktadır. Sağlanacak servisler hedef kitleye veya kitlelere yönelik olduğundan içerik de onların uygulamalarına göre yapılandırılmalıdır. Özellikle servis arayüzleri hedef kitle ihtiyaçlarına göre geliştirilmelidir. Böylelikle genel portal tasarımı ve içeriği belirlenen hedeflerle uyuşabilir.

Tıbbi Sözlük Servisi, tıbbi terimlerin sadece anlamlarına ulaşılmasını sağlayan bir servis olarak tasarlanmıştır. Kullanıcıların, anlamına ulaşmak istedikleri terimi girip arama yapmalarının ötesinde arama kriterini de belirleyerek istedikleri terime ulaşmalarının kolaylaşması amaçlanmıştır. Dört farklı arama kriteri tanımlanmıştır. Buna göre, kullanıcı “Girilen Dize ile Başlayan Terimleri”, “Girilen Dize ile Biten Terimleri”, “Girilen Dizenin İçinde Geçtiği Terimleri” ve “Girilen Dizeye Eşit Olan Terimleri” kriterlerinden birini seçerek aramayı gerçekleştirebilir. Farklı arama kriterlerinin olması servisin kullanım amacını sınırlarını da genişletmektedir. Örneğin; herhangi bir tıbbi terimin bütünü aklına gelmeyen bir öğrenci terimin başladığı, bittiği, ya da içinde geçen hece veya heceleri yazarak arama yapabilir. Böylelikle eğer terim veritabanına kayıtlı ise arama sonuçları arasında çıkacağından terime ulaşabilir.

Akademik ders takvimi servisi eğitim-öğretim yılının tüm programının saklandığı ve öğrencilerin istedikleri günün saat saat programını öğrenebilmelerini sağlayan bir servistir. Bu servis tasarlanırken her öğretim yılı başında Akdeniz Üniversitesi Tıp Fakültesi tarafından yayınlanan Eğitim-Öğretim kitapçısından yararlanılmıştır. Bir eğitim-öğretim yılına ait programın veritabanında saklanabilir hale dönüştürülmesi için öncelikle eğitim-öğretim programının üzerine kurulu olduğu dönem ve kurul soyut yapılarının arasındaki ilişki üzerine mantıksal veri modeli geliştirilmelidir. Bu çalışmada geliştirilen model “Varlık-İlişki Veri Modeli” (26)nin prensiplerine göre tasarlanmıştır. Bu model öğrencilerin ulaşmak istedikleri ders programının hangi kurulda bulunduğunu bildikleri faraziyesi üzerine kuruludur. Öğrencilerin istedikleri günün ders programına ulaşabilmeleri için oldukça efektif bir arayüz tasarlanmıştır. Söz konusu veri modeli ve arayüz bir sonraki bölümde incelenmiştir.

Biyoistatistik ve Tıp Bilişimi Anabilim Dalı için geliştirilen Belge Yönetimi servisi ise anabilim dalındaki yazışma süreçleri, anabilim dalının bünyesindeki kurul ve alt birimler göz önüne alınarak tasarlanmıştır.

3.2. Servislerin Gerçekleştirimi İçin Tercihler ve Geliştirme Araçları

Servislerin tasarımının ardından gerçekleştirimi için hangi ortam ve araç gereçlerin kullanılacağına karar verilmiştir. Gerçekleştirim yapılırken her ne kadar servisler karakteristik açıdan farklı olsalar da kullanılacak ortam ve araç-gereç çeşitliliği yaratmama ilkesi gözetilmiştir. Bu sayede farklı şirketlerin modüler ve birbirleri ile uyum sorunu yaşayabilecek çözümlerinin biraraya getirilmesinden çok aynı şirketin birbirinden bağımsız çalışabildiği gibi birbiriyle işleyebilirlik yeteneğine sahip ortam ve araç-gereçlerinin kullanılması sonucuna ulaşılmıştır.

3.2.1. Belge Yönetimi Servisinin Gerçekleştirimi İçin Seçilen Portal Çözümü

Portal geliştirme sürecinde verilen önemli kararlardan biri servis gerçekleştirimi için benimsenecek olan portal çözüm ve geliştirme ortamının seçimidir. Günümüzde bilgi teknolojileri üreten şirketler birbirlerine alternatif portal çözüm ve geliştirme ortamları piyasaya sürmektedirler. Alternatif portal çözümleri, farklı yetenekleri ile öne çıkmaktadır.

Mevcut portal çözümleri arasından gerçekleştirim için birinin seçilmesi bu çözümler arasında yetenekleri itibarı ile geliştirilecek olan portalın işlevsellik amacına en uygun olanının belirlenmesi de gereklidir. Zira mevcut portal çözümleri farklı avantaj ve dezavantajlar barındırmaktadır.

Yukarıda bahsedilen ilkeler gözetilerek Microsoft şirketinin Windows Sharepoint Services teknolojisi üzerine kurulu Sharepoint Portal Server sunucusu ve yine Microsoft'un ASP.NET adını verdiği ve Web Uygulamaları geliştirme teknolojisi olarak da isimlendirilebilecek geliştirme ortamının kullanımına karar verilmiştir.

Yukarıda bahsedilen temel ilkeden farklı olarak belge yönetim servisi için Microsoft Sharepoint Portal Server teknolojisinin seçilmesinde ayrıca nedenler bulunmaktadır. Öncelikle Sharepoint Portal Server belge yönetimi merkezli geliştirilmiş bir portal çözümü olduğundan belge paylaşımı, yayını, gibi işlevler oldukça basit şekilde gerçekleştirilebilmektedir. Bunun yanında, Sharepoint Portal Server, bünyesinde barındırdığı siteler ve belgeleri indeksleyerek saklamaktadır. Böylece kullanıcılar, Sharepoint Portal Server'in gelişmiş belge arama (site, e-posta, Word dosyası, vd.) imkanlarından yararlanarak aradıkları belgeye kolaylıkla ulaşabilirler.

Ayrıca Sharepoint Portal Server'in diğer bir Microsoft ürünü olan ve belge hazırlama uygulamaları olan, Microsoft Word, Microsoft Excel, Microsoft Powerpoint gibi ofis ürünleri ile bütünleşik şekilde kullanılabilmesi portalın belge yönetimi servisinin kullanımını daha da kolaylaştırmaktadır.

Bir başka neden ise Sharepoint Portal Server ve üzerine kurulu olduğu Windows Sharepoint Services'in portaldaki uygulama ve servislerin genişletilmesini sağlayacak teknolojiler barındırmasıdır. Microsoft, Web Part adını verdiği ve bir portalın uyarlanabilirlik ve kişiselleştirilebilirlik yeteneklerinin yanında portalın hizmet ettiği amaca özel küçük uygulamacıklar geliştirilmesini sağlayan teknolojiyi

geliştirmiştir. Microsoft Sharepoint Portal Server, temel Web Partlar bulundurur fakat işleyişe özel Web Part'lar geliştirilerek kolaylıkla eklenebilir.

Ayrıca, portalın geliştirilme hedefine bağlı olarak kullanılacağı yer olan anabilim dalımızda Microsoft ürünü işletim sistemi ve ofis yazılımlarının geniş olarak kullanılması da temel sebepler arasında sayılabilir.

Portal çözümleri genellikle organizasyonel çözümler barındırdığından ciddi fiyatlarla satılan ürünlerdir. Fakat bazı firmalar portal çözümü ürünlerinin ücretsiz kullanımına izin vermektedir. Maliyetin ciddi bir problem olarak görüldüğü gerçekleştirimlerde bu önemli bir belirleyici olarak karşımıza çıkmaktadır. Maliyetin ön planda olduğu durumlarda bir başka çözüm yaklaşımı ise portalın tasarımı ile birlikte gerçekleştiriminin yapılmasıdır. Bu türde çalışmalar, spesifik uygulamalar için yapılmıştır ve prototip olarak kullanımdadır (31).

3.2.2. Diğer Servisler İçin Benimsenen Geliştirme Prensipleri ve Gerçeklendirim Ortamı

Portalın diğer servislerinin geliştirilmesi bazı yazılım geliştirme prensiplerinin benimsenmesini gerektirmiştir. Servislerin Web üzerinden hizmet verecek olması ve veritabanı içermesi nedeniyle Web veritabanı uygulamaları geliştirmek için en uygun yazılım mimarisinin ve veritabanının kavramsal tasarımının yapılmasını sağlayacak olan uygun veri modelleme yönteminin belirlenmesi birincil adımdır. Bu çalışmada 3-Katmanlı yazılım mimarisi ve Varlık-İlişki veri modeli benimsenmiştir.

Ne tip olursa olsun, veritabanı uygulamalarının geliştirilmesi için öncelikle veritabanında verinin saklanması, veriye ulaşılması gibi temel işlevlerin gerçekleşmesini sağlayacak olan veri modelinin tasarlanması gerekir. Servislerin gerektirdiği mantıksal veri modellerinin tasarımı için Varlık-İlişki Modellemesi benimsenmiştir (26). Veri modellerinin fiziksel tasarımı ise yine Microsoft'un bir ürünü olan Microsoft Access ile gerçekleştirilmiştir. Microsoft Access, küçük ve orta ölçekli veritabanlarının yönetimini gerçekleştirebilir.

Tıbbi sözlük ve ders takviminin gerçekleştirimi için ise bir başka Microsoft ürünü olan ASP.NET platformu kullanılmıştır. Servisler yapıları itibarıyla veritabanı uygulamaları olmakla beraber sunucu tarafı işleme gerektirmektedirler. Dolayısıyla bu servisler Web Veritabanı Uygulaması adı da verilmektedir.

Bu servislerin geliştirilmesi için bir diğer gereklilik uygulamaların sunucu tarafı işlem gerektirdiğinden bir Web sunucusudur. Bu çalışmada Microsoft'un ASP.NET geliştirme platformunu destekleyen yine Microsoft'un ürünü olan Web sunucusu İnternet Information Server kullanılmıştır.

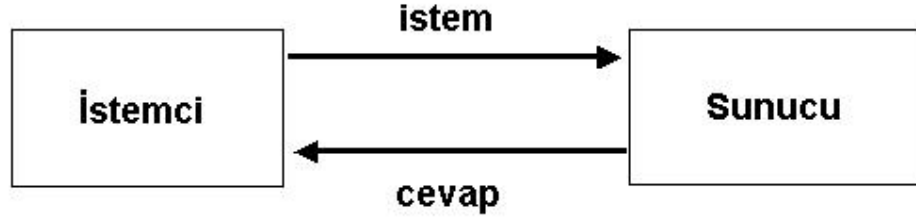
3.3. Web Veritabanı Uygulama Geliştirme Mimarisi ve ASP.NET

Tıbbi sözlük ve ders takvimi servislerinin uygulama mimarisi olarak 3-katmanlı uygulama mimarisi benimsenmiştir. Gerçekleştirimi için ise ASP.NET platformu tercih edilmiştir. Bu alt bölümde benimsenen mimari ve geliştirme platformuna değinilmiştir.

3.3.1. 3-Katmanlı Uygulama Mimarisi

Uygulama mimarisi bir uygulamanın gerçekleştirmesi için geliştirileceği işin karakteristiği de göz önüne alınarak belirlenir. Ağ teknolojilerinin henüz bulunmadığı dönemlerde bilgisayarlar tek katmanlı olarak adlandırabileceğimiz, yalnızca üzerinde çalıştığı bilgisayarın kaynaklarını kullanan uygulamalardı. Bilgisayar ağlarının ortaya çıkması ile dağıtık sistemler ve uygulamalar da geliştirilmeye başlandı. Dağıtık uygulamalar, ortak bir iş için farklı bilgisayarlarda çalışan, iş sorumlulukları gerçekleştirmeleriyle tanımlanmış ve birbirleri ile iletişim kurmak için belli bir protokolleri olan uygulamalardan oluşan sistemlerdir.

Dağıtık sistemler geliştirilirken kullanılan temel mimarilerinden biri “istemci/sunucu” mimarisidir. Temel olarak istemci/sunucu mimarisi mesaj-tabanlı istem ve cevap ikilileri dizisi üzerine kurulu bir iletişim şekli benimser. Buna göre sunucu bir servisi sağlayan, istemci ise servis için istemde bulundur (27,28). Bu mimari mainframe mimarisinin dezavantajlarını yok ederek, dağıtık sistemlerde kullanılabilirlik, esneklik, ölçeklenebilirliği artırmak için geliştirilmiştir. Şekil 3.1’de bu mimarinin basit işleyişi gösterilmektedir.



Şekil 3.1. İstemci/Sunucu İletişim Mimarisi

İstemci/sunucu mimarisi zaman geçtikçe farklı alt mimari jenerasyonları doğurmuştur. 2-katmanlı mimari, istemci/sunucu mimarisinin bir örneğidir. Bu mimaride iş yönetimi istemci ve sunucu arasında bölünmüştür. Buna göre iş yönetiminin bir kısmı; örneğin bir dağıtık veritabanı sisteminde veritabanındaki bilgiye ulaşmak isteyen bir kullanıcının bilgisayarında çalışan sistem arayüzü tarafından gerçekleştirilirken, geriye kalan kısmı ise veritabanı yönetim sistemi (sunucusu) tarafından gerçekleştirilir (28).

2-katmanlı mimari oniki ile 100 istemcili sistemler için oldukça iyi bir çözümdür. Fakat istemci sayısı 100’ü geçtiğinde özellikle ağ üzerindeki iletişim ağırlığı nedeni ile performansın giderek kötüleştiği saptanmıştır. Bu mimarinin bir başka dezavantajı özellikle veritabanı uygulamalarında ciddi esneklik kısıtlamalarının olmasıdır. Bunun nedeni, iş yönetim hizmetlerinin farklı şirketlerin ürünü olan farklı veritabanı yönetim sistemlerinde farklı şekilde ya da sırada gerçekleştirilmesidir (28).

3-katmanlı mimari, 2-katmanlı mimarinin belirtilen dezavantajları göz önünde bulundurularak geliştirilmiş bir 2.kuşak istemci/sunucu mimarisidir. Geliştirilmesindeki temel amaç 2-katmanlı mimarinin dağıtık sistemler için sağlayamadığı (100'den fazla istemci biriminden sonra performans düşüklüğü) yüksek ölçeklenebilirliği sağlayarak özellikle yüksek kullanıcı sayısı gerektiren dağıtık sistemler geliştirebilmektir. Bunun yanında 3-katmanlı mimari istemcisini/kullanıcısını dağıtık işlemin gerektirdiği karmaşıklıktan da ciddi bir seviyede kurtarmıştır. Bu özellikleri ile 3-katmanlı mimari İnternet uygulamaları için geçerli bir mimari halini almıştır (28,29). Tablo 3.1'de 2-katmanlı ve 3-katmanlı yazılım mimarilerinin bir karşılaştırmasını göstermektedir.

Tablo 3.1 2-Katmanlı ve 3-Katmanlı Yazılım Mimarilerinin Karşılaştırması

Kriter	2-Katmanlı Mimari	3-Katmanlı Mimari
Birlikte Çalışabilirlik	Sınırlı	Yüksek
Taşınabilirlik	Az Taşınabilir	Daha Fazla Taşınabilir
Yeniden Kullanılabilirlik	Sınırlı	Yeniden Kullanılabilir
Karmaşıklık	Yalın	Karmaşık
Bakım	Zor	Kolay
Ölçeklenebilirlik	100 Kullanıcıya kadar	1000 Kullanıcıya kadar

3-katmanlı mimariye göre, bir dağıtık sistemin sunum, iş/süreç yönetimi ve veri yönetimi katmanları vardır. Bu katmanları, sırasıyla istemci, ara sunucu ve asıl sunucu oluşturur. Diğer bir ifadeyle istemci sunumu, aradaki sunucu iş yönetimini ve diğer sunucu da veriyi yönetir (saklar). Bu mimaride orta katman işleyişin temel belirleyicisidir (28).

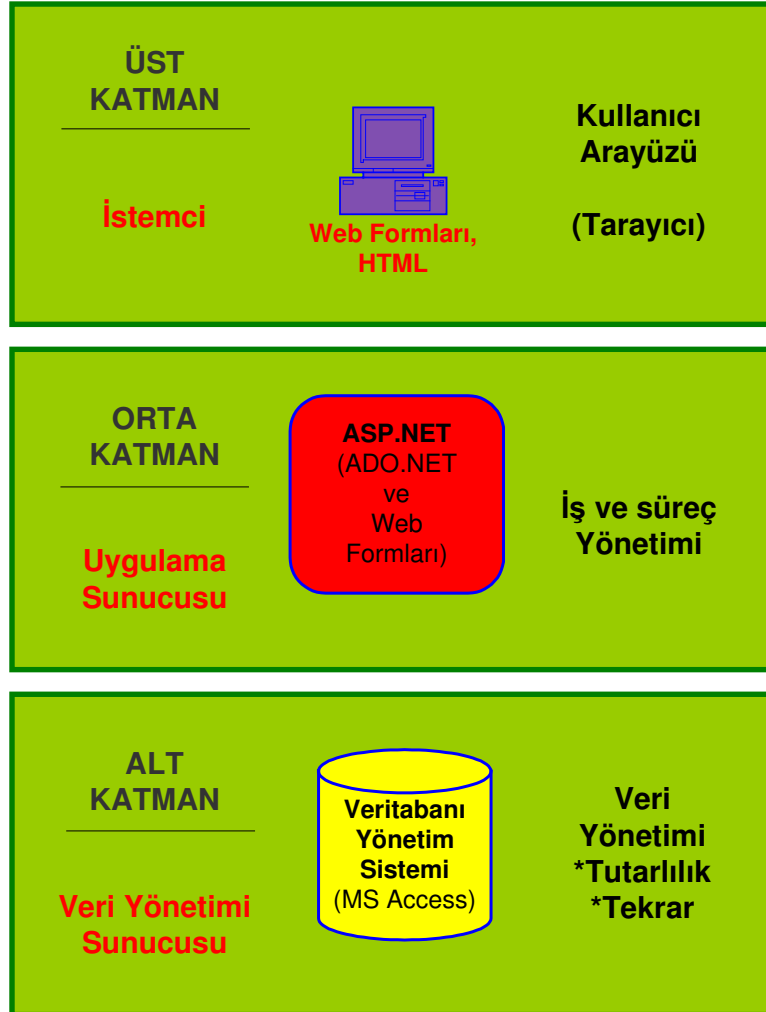
En alt katman olan veri yönetimi katmanı, verilerin saklandığı ve ulaşıldığı katmandır. Örneğin bir Web Veritabanı Uygulaması için; bu katman, modellenmiş ve fiziksel olarak bir veritabanı yönetim sisteminde saklanan veritabanının olduğu katmandır. Uygulama bu tür olduğunda bu katmanın temel görevleri veri tutarlılığı sağlamak, veri tekrarını engellemek ve veriyi güvenli bir şekilde saklamak olarak belirtilebilir (28).

2. katmana uygulama/iş mantığı katmanı adı verilir. Bu katmanda ilk katmandaki veriye ulaşılmasını sağlayacak olan kaynaklar yönetilmekte ve veriye ulaşım kuralları uygulanmaktadır. Kısaca bu katman süreç yönetim servisleri sağlamaktadır(29). 3-katmanlı mimarinin orta katmanı, 1.kuşak istemci/sunucu mimarisinde istemci ve sunucu arasında paylaşılan iş/uygulama mantığı yönetimini tek başına yaparak bu işlevin merkezileşmesini sağlamıştır. Bu katmanda merkezileştirilen iş/süreç yönetimi beraberinde ciddi bir ölçeklenebilirlik ve performans kazanımı sağlamıştır.

İstemci seviyesinde gerçekleştirilen sunum katmanı temelde uygulama kullanıcısının uygulama verisi ile etkileşim birimidir. Bu katman kullanıcının sistem

ile etkileşimini formlar marifeti ile oluşturulan bir arayüz bulundurarak sağlar. Arayüz, etkileşim için gerekli verilere göre hazırlanır. Verilerin geçerlik kuralları bu katmanda arayüz sayesinde uygulanmalıdır.

3-katmanlı mimari yaklaşımı yazılım geliştirme sürecini de kolaylaştırmaktadır. Orta katman sayesinde kazanılan modülerite her bir katmanın birbirinden bağımsız platformlarda, farklı programlama/işaretleme dilleri kullanılarak geliştirilmesine olanak tanımaktadır. Bu uygulama geliştiriciler için ciddi bir esneklik sağlamaktadır (28,29).



Şekil 3.2. Servislerin Geliştirilmesinde Kullanılan Yazılım Mimarisi

Bütün bu avantajlarına rağmen 3-katmanlı mimari kendi içinde ciddi bir dezavantaj bulundurmaktadır. 3-katmanlı mimarinin belli bir çözüme dönük tasarımı sırasında arayüz, iş/uygulama/süreç yönetimi ve veri yönetimi katmanları yeterince keskin sınırlar ile belirlenemeyebilir. Bu genellikle iş/süreç yönetiminin her bir katmanda farklı şekillerde ortaya çıkmasından kaynaklanır (29).

Bu çalışmadaki tıbbi sözlük ve ders takvimi servisleri de 3-katmanlı mimari benimsenerek geliştirilmişlerdir. Mimarinin orta katmanı bu mimariyi destekleyen bir Web uygulaması geliştirme platformu olan ASP.NET kullanılarak gerçekleştirilmiştir. ASP.NET, 3-katmanlı mimari üzerine geliştirilecek küçük, orta ya da büyük ölçekli bir Web veritabanı uygulamasının orta katmanının gerçekleştirimi için gerekli tüm uygulama protokol arayüzlerini, kaynaklarını ve bunların yönetimini sağlayan programlama arayüzlerini bünyesinde barındırmaktadır (30). Bütün bu arayüz ve kaynaklar, Microsoft'un geliştirdiği ve ActiveX Data Objects .NET(ADO.NET) adını verdiği veri kaynağı bağlantı ve veri işleme teknolojisinin bir parçasıdır.

Veri yönetimi katmanı ise Microsoft Access kullanılarak gerçekleştirilmiştir. En üst katman olan sunum katmanı HTML işaretleme dili ve ASP.NET'in bünyesinde barındırdığı Web Forms teknolojisi kullanılarak geliştirilen bir arayüzden oluşmaktadır. Bu arayüz Web tarayıcıları tarafından sorunsuz görüntülenebilmektedir. Şekil 3.2'de kullanılan mimarinin katmanları ve servislerin geliştirilmesinde bunların karşılıkları gösterilmektedir.

3.3.2. Veritabanı Modelleme ve MS Access

Bir veritabanı uygulaması geliştirmek öncelikle veritabanı yapısının iskeletini oluşturan ve veritabanında saklanacak veriyi, veri ilişkilerini tanımlayan, veri anlamlarını ve veri üzerindeki kısıtlamaları belirleyen bir veri modeli tasarlamayı gerektirmektedir. Veri modelleri üç temel sınıfta toplanır; nesne-tabanlı modeller, kayıt-tabanlı mantıksal modeller ve fiziksel modeller. Bu modeller de kendi içlerinde farklı modeller içermektedir.

Bu çalışmada veri modeli olarak nesne-tabanlı modellerden biri olan Varlık-İlişki modeli benimsenmiştir. Varlık-İlişki modeli geniş kabul görmüş ve halen uygulamalarda kullanılmakta olan en popüler veri modelidir.

Varlık-İlişki (E-R) veri modeli, varlık adı verilen nesnelere ve bu nesnelere kendi aralarındaki ilişkilerini tanımlayan bir gerçek dünya algısının matematiksel ilkelerle tasarımıdır. Buna göre, varlık ismi ile anlatılmak istenen, diğer nesnelere belirli özellikleri sayesinde ayırt edilebilen nesnedir. Varlıklar arası ilişki ise varlıkların birbirleri ile var olma tanımlamalarını yapan kavramsal bir yapıdır. Bir E-R veri modeli aynı tür varlıkların birarada olduğu varlık kümeleri ve ilişki kümelerinden meydana gelmektedir.

E-R veri modeli varlık ve ilişkilerinin yanında bir veritabanının sahip olması gereken kısıtlamaları da tanımlayabilir. İlişkilerin tipini de belirleyen "eşleştirme sayısı" bu kısıtlamalardan sadece biridir.

Veri modeli hazırlandıktan sonra normalizasyon adı verilen ve modelin matematiksel bütünlüğünün sağlanması olarak açıklanabilecek bir safhaya sokulur. Bu safhada beş temel normalizasyon uygulanarak veri modeline yeni tablolar eklenebilir, mevcut tablolar silinebilir veya bölünerek iki ya da daha fazla tabloya dönüşebilir. Bu safha geliştirilen modelin sağlam bir yapıya sahip olmasını sağlasa da çoğu geliştirici bu safhayı atlar.

Veritabanı uygulamalarının geliştirme süreçlerinde veri modelinin hazırlanmasının ardından gelen adım veritabanının fiziksel olarak yaratılması aşamasıdır. Bu aşamada geliştiricinin, veritabanında saklanacak verinin ve onu işleyecek uygulamanın karakteristik özelliklerini göz önünde tutarak küçük, orta ya da büyük ölçekli veritabanı profillerinden hangisine girdiğine karar vermesi gereklidir. Verdiği bu karara göre geliştirdiği veri modelinin fiziksel gerçekleştirimini yapacağı bir veritabanı yönetim sistemi seçecektir.

Bu çalışmada, yukarıda sözü edilen özellikler dikkate alınarak geliştirilen veritabanlarının küçük ve orta ölçekli arası bir veritabanı olduğu sonucuna varılmıştır. Tıbbi sözlük ve ders takvimi servislerinin kullanıcı sayısının; diğer bir ifade ile öğrencilerin bu servislere ilgileri ölçüldükten sonra sistem daha keskin bir şekilde değerlendirilebilir ve buna göre sisteme eklemeler yapılabilir.

Kullanılacak veritabanlarının küçük ve orta ölçekli arası olması veritabanının fiziksel yaratımı için Microsoft'un Access adlı uygulamasının yeterli olduğu sonucuna varıldığından tasarlanan veri modelleri bu uygulama kullanılarak yaratılmışlardır. Buna göre, tasarlanan veri modelindeki varlıklar ve ilişkiler, ilişki tipleri göz önünde bulundurularak tablolara dönüştürülmüş, birincil anahtarları belirlenmiş ve gösterim bütünlüğü gibi kısıtlamalar uygulandıktan sonra tablolar arası ilişkiler tanımlanarak veritabanı yaratılmıştır.

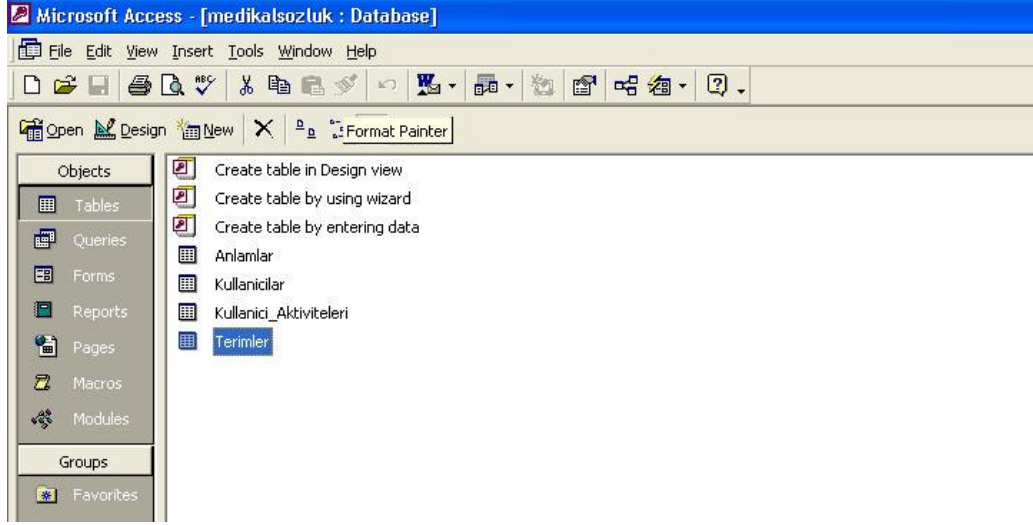
Şekil 3.3'te tıbbi sözlük için yaratılan veritabanının tablo listesi, Şekil 3.4'te ise ders takvimi için yaratılan veritabanının tablo listesi Microsoft Access'ta yaratıldığı gibi gösterilmiştir.

3.3.3. ADO.NET ve Web Formları

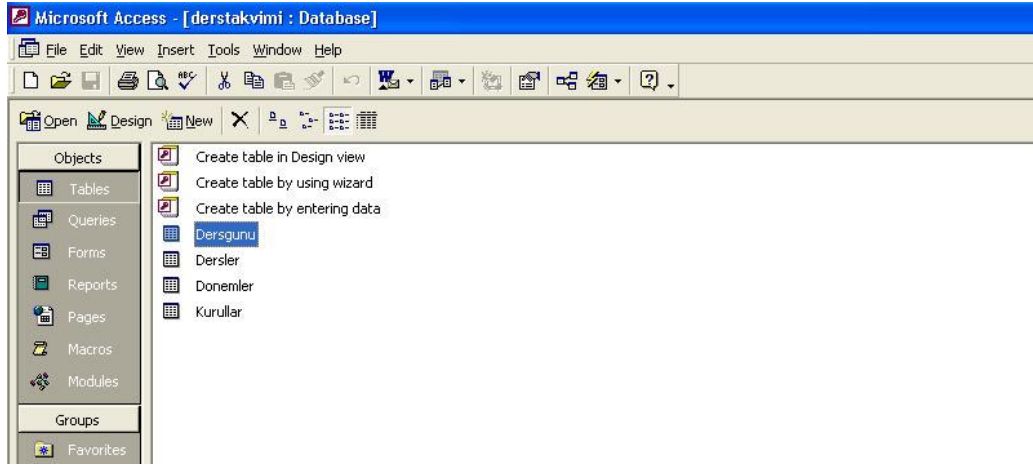
ADO, Microsoft'un ilk defa 1996 yılında piyasaya sürdüğü ve OLE DB ve ODBC gibi sağlayıcılar marifeti ile veritabanı gibi veri kaynaklarındaki veriye ulaşım ve işleme olanakları sağlayan bir nesne modelidir. Özünde ADO, programlama dilleri ile geliştirilen uygulamalar ile sağlayıcılar arasında bir katmandır. Microsoft, ADO'nun 1996'da tanıttığı ilk sürümünden sonra .NET platformu ile beraber ADO.NET sürümünü geliştirmiş ve piyasaya sürmüştür.

ADO.NET, kullanım kolaylığı ve düşük hafıza kullanım maliyeti ile özellikle veritabanı uygulama geliştiricileri tarafından en çok kullanılan veritabanına ulaşım aracıdır.

ADO.NET, tıpkı diğer birçok veri ulaşım yöntemi gibi uygulama geliştiricilere, veri kaynağına bağlantı, veri işleme komutlarının çalıştırılması (SQL), komut sonuçlarının alınması, veri ekleme, silme, ve güncelleme, işlem ve olay yönetimi ve hata sonlu işlem yönetimi gibi uygulama görevleri için kullanımı oldukça basit uygulama programlama arabirimleri ve bu arabirimlerin uygun kullanımı için de farklı tipte veri yapıları sağlar. Fakat ADO.NET'i diğerlerinden ayıran en önemli özelliklerinden biri veri kaynağı ile yaptığı işlemleri gerçek-zamanlı gerçekleştirmesidir.



Şekil 3.3. Tıbbi Sözlük Servisi Veritabanı Tabloları



Şekil 3.4. Ders Programı Servisi Veritabanı Tabloları

3.4. Tıbbi Terimler Sözlüğü Servisi Veritabanı Boyutunun Servis Süresine Etkisi

Tıbbi terimler sözlüğü servisi veritabanı, tıbbi terimlerin veritabanına eklenmesi ile binler hatta onbinlerle ifade edilir sayıda tıbbi terim ve bunlara ait anlam saklayacaktır. Bu açıdan değerlendirildiğinde veritabanı boyutunun büyümesi ile servis performansının bundan nasıl etkileneceği üzerine fikir vermesi açısından bazı ölçümler yapılmıştır.

3.4.1. Ölçüm Şartları ve Varsayımları

Ölçümler, istemci (tarayıcı), sunucu (Web sunucusu) ve veritabanının aynı fiziksel makinada yer alır şekilde gerçekleştirilmiştir. Dolayısıyla ağ performansı bu ölçümlerin dışında bırakılmıştır.

Ölçümler sırasında kullanılan bilgisayar konfigürasyonu aşağıda belirtilmiştir;

- İşlemci: Intel Centrino 715, 1.5 GHz
- Hafıza: 512 MB
- İşletim Sistemi: Microsoft Windows XP Professional 5.1.2600 Service Pack 2 Yapı 2600
- Sabit Disk: 40 GB (2 partiyonlu, 1.8 GB boş alan)

Ölçümler sırasında kullanılan sunucu, istemci, geliştirme platformu ve yazılımlar, ve veri sağlayıcısı ise;

- İstemci: İnternet Explorer 6.0.2900.2180 SP2
- Sunucu: İnternet Information Services 5.1
- Geliştirme Platformu: ASP.NET, ADO.NET
- Veritabanı Yönetim Sistemi: MS Access 2000 (v9.0)
- Veri Sağlayıcısı: Microsoft Jet OLEDB 4.0

Ölçümlerin gerçekleştirilebilmesi için geliştirilen veritabanı çöplük veriler ile doldurulmuştur. Fakat doldurma işlemi gerçekleştirilmeden evvel tablolarda saklamak için sabit bir çöplük veri formatı belirlenmiştir. Buna göre, tıbbi terimler için onbir karakterden oluşan dizeler ve anlamlar için 30 karakterden oluşan dizeler girilmesi benimsenmiştir. Tablolarda yer alan diğer alanlar (örneğin; terimin eklenme tarihi, değiştirilme tarihi) sistem tarafından girilmektedir.

Performans ölçümü için farklı büyüklüklerde(farklı sayıda terim içeren) altı veritabanı oluşturulmuştur.

Ölçümler veritabanının, boyutu büyüdükçe performansının nasıl etkileneceğini ölçmek adına yapıldığından benimsenen yazılım mimarisinin orta katmanında tamamlanmakta ve tıbbi terimler sözlük servisi için gerçekleştirilen veritabanı işlemlerinin sunucuya ulaşan istemin ardından ne kadar zaman gerektirdiği üzerine kuruludur. Diğer bir ifade ile performans ölçümleri bir istem için gerekli veritabanı işleminin ne sunucu tarafında ne kadar zaman aldığı hesaplamak için yapılmıştır.

Ölçümler orta katmanda sunucu tarafından işlenen ve servis ile ilişkili kaynak kodun başına ve sonuna zaman ölçer fonksiyonlar yerleştirilerek gerçekleştirilmiştir. Ölçüm değerleri milisaniye cinsindedir.

BULGULAR

4.1. Tıbbi Sözlük Servisi

Bu servisin geliştirilme amaçlarının başında özellikle Akdeniz Üniversitesi Tıp Fakültesi öğrencilerinin çalışmaları sırasında ilk kez okudukları, derste veya hastanede duydukları tıbbi terimlerin anlamlarını öğrenebilecekleri, kendi üniversiteleri bünyesinde bulunan bir çevrimiçi kaynak oluşturmaktır. Bunun yanında öğrencileri, çalışma ve araştırma süreçlerinde İnternet'i kullanmaya alıştırmaktır. Öğrenciler, böylesi bir arama servisinin kullanımı ile İnternet kullanımının temel adımlarına daha aşına olacaklar ve böylelikle İnternet'i giderek çalışma ve araştırmalarının birincil kaynağı olarak düşünebileceklerdir.

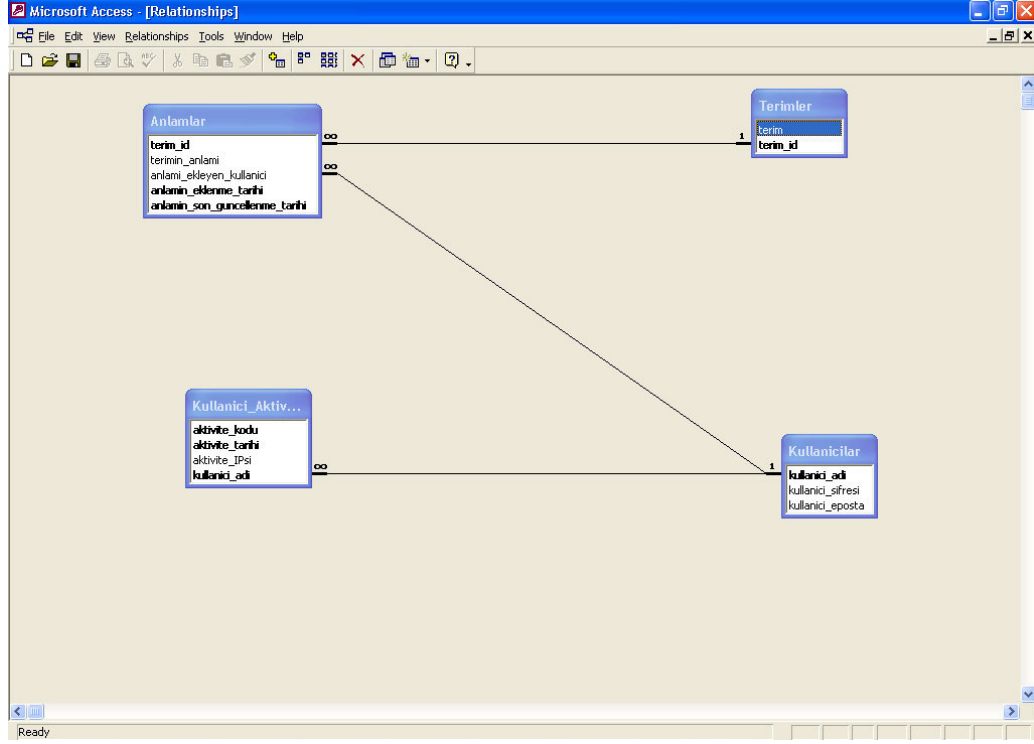
4.1.1. Sözlük Veritabanı Fiziksel Yapısı- Veri Yönetimi Katmanı

Tıbbi sözlük servisi önceki bölümlerde anlatılan veritabanı teknolojileri, uygulama geliştirme mimarileri ve ortamları benimsenerek meydana getirilmiştir. Buna göre, servisin veritabanını oluşturan tablolar, özellikleri, birincil anahtarları ve aralarındaki ilişkiler Şekil 4.1'de gösterilmiştir.

Veritabanı dört tablodan oluşmaktadır. Bunlardan Terimler adlı tablo veritabanında bulunan terimleri saklamaktadır. Terimler tablosunu terim metni ve terimin veritabanına eklenmesi sırasında veritabanı tarafından verilen ve eşsiz olan terim numarası özellikleri oluşturur. Anlamlar tablosu ise Terimler tablosuna eklenerek dahil olan, bu tabloda bulunan terimlerin anlamlarının özellik olarak bulunduğu tablodur. Anlamlar tablosu, terim numarası, terimin anlamı, anlamı ekleyen kullanıcı, anlamın eklendiği tarihi ve anlamın son güncellenme tarihi özelliklerinden oluşur.

Terimler tablosu ve Anlamlar tablosu arasında Bire-Çok tipinde bir ilişki bulunmaktadır. Bu ilişki sayesinde veritabanında bir terime dair birden fazla anlam saklanabilmektedir. Şekil 4.2'de Terimler tablosunun, Şekil 4.3'de ise Anlamlar tablosunun sakladıkları verilerden bir bölüm gösterilmektedir.

Tıbbi servisin veritabanı üzerinde şimdilik üç farklı işlem gerçekleştirilebilir; terim ve anlam arama, terim ve anlam ekleme ve silme. Önceki bölümde de belirtildiği gibi bu işlemler ASP.NET ve HTML teknolojileri kullanılarak geliştirilmiştir. Öğrencilerin kullanımına ise sadece arama işlemi sunulmuştur. Veritabanına yeni terimler ekleme, veritabanındaki mevcut terim veya anlamları silme işlemleri yetkilendirilmiş kullanıcılar tarafından gerçekleştirilebilir.



Şekil 4.1. Tıbbi Sözlük Servisi Veritabanı Tabloları Arasındaki İlişkiler

4.1.2. Tıbbi Terim Arama ve Ekleme Arayüzü- Sunum Katmanı

Veritabanı işlemlerinin gerçekleştirimi için hazırlanan arayüzlerden arama arayüzü, aramanın üç adımda gerçekleştirilmesini sağlar. Şekil 4.4’de üç adımlı bu arayüz gösterilmektedir. 1.adımda metin kutusuna aranan terim veya dize girilmesi istenmektedir. 2.adımda aramayı yapan en uygun arama kriterini seçtikten sonra, 3.adımda arama düğmesine basılması istenmektedir. Farklı arama kriterlerine göre arama yapılabilmesini sağlayan arayüz; örneğin bir öğrencinin aradığı terimi dört arama kriterinden birine göre aramayı gerçekleştirmesine izin verir.

Bu dört kriterden hangisine göre arama yapılacağı ise öğrenciye bağlıdır. Eğer öğrenci, aradığı terimin ilk birkaç harfini ya da birkaç kelimedenden oluşan bir tıbbi terimin ilk kelimesini hatırlayabiliyor, geriye kalan kısmını hatırlayamıyorsa hatırladığı kısmı metin kutusuna yazdıktan sonra arama kriteri olarak “Dize ile Başlayan Terimleri” kriterini seçtikten sonra ARA düğmesine bastığı anda arama gerçekleştirilir.

Bu arama kriterinin sonucu bir terim ve ona ait anlam(lar)ı değildir. Öğrenciye sağlanan arama sonucu ara metni ile başlayan tıbbi terimlerin listesidir. Böylelikle öğrenci hatırlayamadığı tam terimi bu liste içinden arayıp bulabilir ve daha sonra bir başka arama kriteri olan “Dizeye Eşit Olan Terimleri” kriteri ile bir arama yaptığında terimin veritabanında kayıtlı anlamlarına ulaşabilecektir. Şekil 4.5’te “Dize ile Başlayan Terimleri” arama kriterine göre yapılan bir arama sonucu gösterilmektedir.

	terim	terim_id
+	bakteriyemi	50
+	ballismus	51
+	benign	52
+	beriberi	53
+	biyopsi	54
+	blefarit	55
+	brakiyalji	56
+	bronchiolitis	57
+	büllöz	58
+	calcaneus	59
+	candida	60
+	cerrahi	61
+	cerumen	62
+	cestodiasis	63
+	clavicula	64
+	cor	65
+	coxae	66
+	dakriyoadenit	67
+	dakriyolit	68
+	dakriyosistekto	69
+	dakriyosistit	70
+	dakriyosistogra	71
+	dakriyosistorinc	72
+	daltonizm	73
+	dansimetre	74
+	debilite	75
+	defekasyon	76
+	defekt	77
+	defibrilatör	78
+	deflorasyon	79
+	deformasyon	80
+	deformite	81
+	dejenarasyon	82
+	dejenarasyon	83

Şekil 4.2. Terimler Tablosundan Bir Görüntü

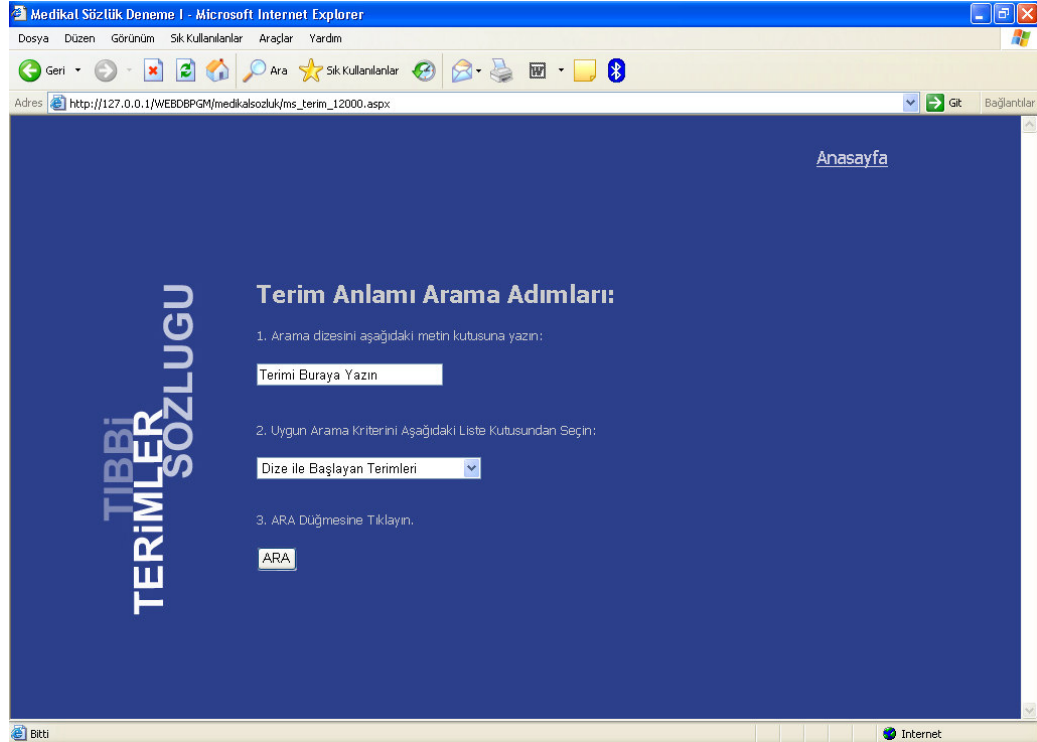
Diğer arama kriterleri olan; “Dize ile Biten Terimleri” ve “Dizeyi İçinde Bulunduran Terimleri” ile yapılan aramalar da “Dize ile Başlayan Terimleri” arama kriteri gibi bir terimler listesi sonucu verir. Farklı olan, “Dize ile Biten Terimleri” arama kriteri sonucu çıkan listede bulunan terimler girilen arama metnini sonunda bulunduran terimler iken “Dizeyi İçinde Bulunduran Terimleri” arama kriteri sonucu ortaya çıkan listedeki terimler ise arama metninin terimin herhangi bir bölümünde bulunduğu saptanan terimlerdir.

Arama arabiriminin haricinde veritabanına yeni terim ve anlam ekleme arabirimi de geliştirilmiştir. Bu arabirim, henüz uygun kullanıcı altyapısı tamamlanmamış olmakla birlikte yalnızca yetkilendirilmiş kullanıcılar tarafından kullanılabilir bir arayüzdür. Şekil 4.6’da bu arabirim gösterilmektedir.

Terim ve anlam ekleme işlemi yapan bir kullanıcı, veritabanının gönderdiği sonuca bağlı olarak bilgilendirilmektedir. Örneğin; eğer kullanıcının girdiği terim ve anlam başarı ile veritabanına kaydedildi ise işlemin başarıyla sonlandırıldığını bildiren bir cevap sayfası tarayıcıda görüntülenmektedir.

terim_id	terimin_anlam	anlami_ekley	anlamin_ekle	anlamin_son
50	bakterilerin vey	ykyuce	I.2006 16:53:28	I.2006 16:53:28
51	Kol ve bacaklar	ykyuce	I.2006 16:54:55	I.2006 16:54:55
52	İyi huylu	ykyuce	I.2006 16:55:31	I.2006 16:55:31
53	B vitamini eksik	ykyuce	I.2006 16:56:01	I.2006 16:56:01
54	Canlı bir dokud	ykyuce	I.2006 16:56:56	I.2006 16:56:56
55	Göz kapaklarını	ykyuce	I.2006 16:57:29	I.2006 16:57:29
56	Kol ağrısı.	ykyuce	I.2006 16:57:50	I.2006 16:57:50
57	Solunum sisten	ykyuce	I.2006 16:58:52	I.2006 16:58:52
58	Büllerden oluşa	ykyuce	I.2006 16:59:12	I.2006 16:59:12
59	Topuk kemiği.	ykyuce	I.2006 16:59:46	I.2006 16:59:46
60	Bir mantar çeşi	ykyuce	I.2006 16:59:58	I.2006 16:59:58
61	Tibbin en eski c	ykyuce	I.2006 17:02:18	I.2006 17:02:18
62	Kulak kiri. İnsa	ykyuce	I.2006 17:03:43	I.2006 17:03:43
63	Yassı solucan e	ykyuce	I.2006 17:04:05	I.2006 17:04:05
64	Köprücük kemi	ykyuce	I.2006 17:04:25	I.2006 17:04:25
65	Kalp.	ykyuce	I.2006 17:04:34	I.2006 17:04:34
66	Kalça kemiği.	ykyuce	I.2006 17:04:48	I.2006 17:04:48
67	Gözyaşı bezi ilt	ykyuce	I.2006 17:05:24	I.2006 17:05:24
68	Gözyaşı taşı.	ykyuce	I.2006 17:05:43	I.2006 17:05:43
69	Gözyaşı kesesi	ykyuce	I.2006 17:06:07	I.2006 17:06:07
70	Gözyaşı kesesi	ykyuce	I.2006 17:06:30	I.2006 17:06:30
71	kontrast madde	ykyuce	I.2006 17:07:21	I.2006 17:07:21
72	Gözyaşı kanalın	ykyuce	I.2006 17:08:24	I.2006 17:08:24
73	renk körlüğü.	ykyuce	I.2006 17:08:37	I.2006 17:08:37
74	Yoğunluk ölçen	ykyuce	I.2006 17:08:58	I.2006 17:08:58
75	Zeka geriliği.	ykyuce	I.2006 17:09:13	I.2006 17:09:13
76	Dışkının dışarı	ykyuce	I.2006 17:09:35	I.2006 17:09:35
77	Eksiklik, kusur.	ykyuce	I.2006 17:09:48	I.2006 17:09:48
78	Kalbin normal d	ykyuce	I.2006 17:10:37	I.2006 17:10:37
79	Kızlık zannın y	ykyuce	I.2006 17:10:55	I.2006 17:10:55
80	Şekli bozma.	ykyuce	I.2006 17:11:15	I.2006 17:11:15
81	Şekil bozukluğu	ykyuce	I.2006 17:11:30	I.2006 17:11:30
82	Dokuların norm	ykyuce	I.2006 17:12:04	I.2006 17:12:04
83	Beş uyanma adı	ykyuce	I.2006 17:12:29	I.2006 17:12:29

Şekil 4.3. Anlamlar Tablosundan Bir Görüntü

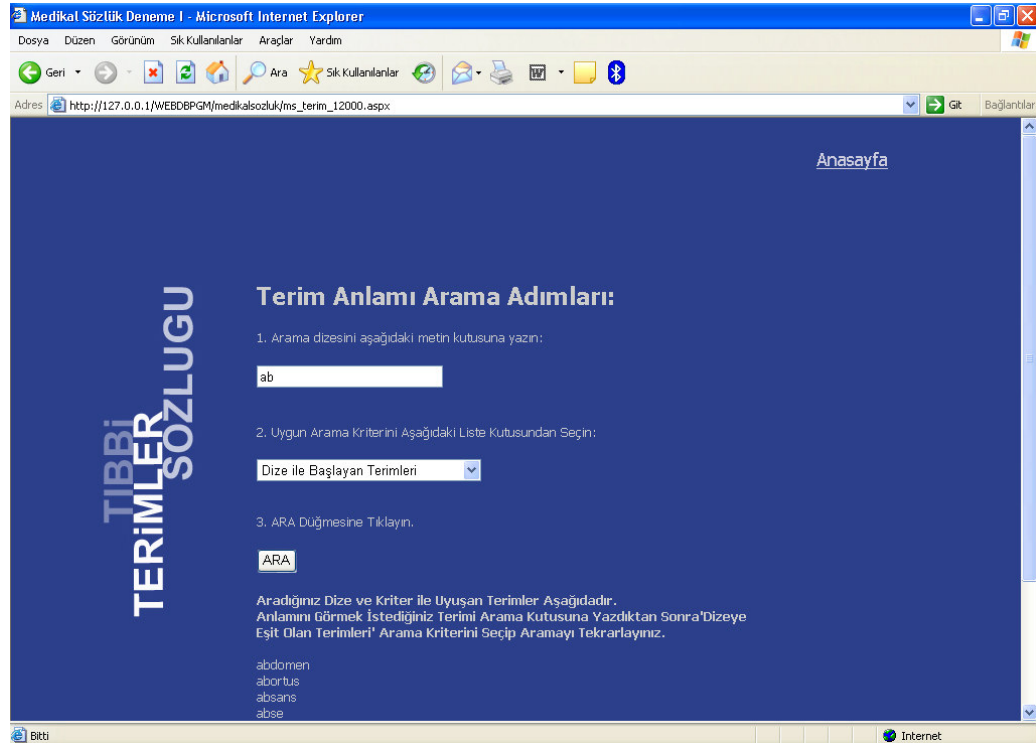


Şekil 4.4. Tıbbi Sözlük Servisi Arayüzü (Sunum Katmanı)

Eğer kullanıcının eklediği terimin veritabanında hali hazırda bir ya da daha fazla anlamı kayıtlı ise bu durumda kullanıcıyı girdiği terim için yeni bir anlam kaydettiği şeklinde bilgilendirici bir başka cevap sayfası görüntülenir. Terimin hali hazırda veritabanında kayıtlı olduğu bu durumlar için; aynı anlamın farklı cümlelerle de olsa birden fazla şekilde yer almamasını sağlamak amacıyla kullanıcı “Gönder” tuşuna bastıktan sonra kullanıcıya terimin veritabanında kayıtlı anlamlarını göstererek, ancak kendi girdiği anlam kayıtlı anlamlar arasında yoksa yeni anlamın kaydını onaylamasını isteyecek bir ara adım eklenmesi kullanıcı arabiriminin daha efektif çalışması için planlanmaktadır.

4.1.3. Servis Gerçekleştirmede Olaylar ve Olay İşleyiciler

Servisin çalışma esası bütünü ile en önemli katmanı sayılan orta katmanda bulunan Web Forms üzerine kuruludur. Web Forms, “olaya dayalı programlama”(event-driven) prensibi ile kullanılabilen bir arayüz geliştirme paketidir. Buna göre, formu oluşturan her tip bileşenin kendine özgü olayları ve bu olaylar gerçekleştiğinde yapılması gerekenleri gerçekleştiren “olay işleyici”leri vardır. Uygulama/servis için geliştiricinin arayüzde gerekli bulduğu olaylar atışlenebilir. Ateşlenmesi planlanan her bir olay için ise geliştirici bir olay işleyici yazmak zorundadır. Bu serviste atışlenebilir olay, formun düğme bileşeninin OnClick olayıdır. Diğer bir ifade ile “ARA” düğmesi tıklandığında ya da “Enter” tuşuna basıldığında form öğelerinin tüm değerleri bir HTTP komutu olan POST yöntemi ile sunucuya gönderilir. Sunucu aldığı verileri yine Web Forms’un çalışma ilkeleri doğrultusunda işler.

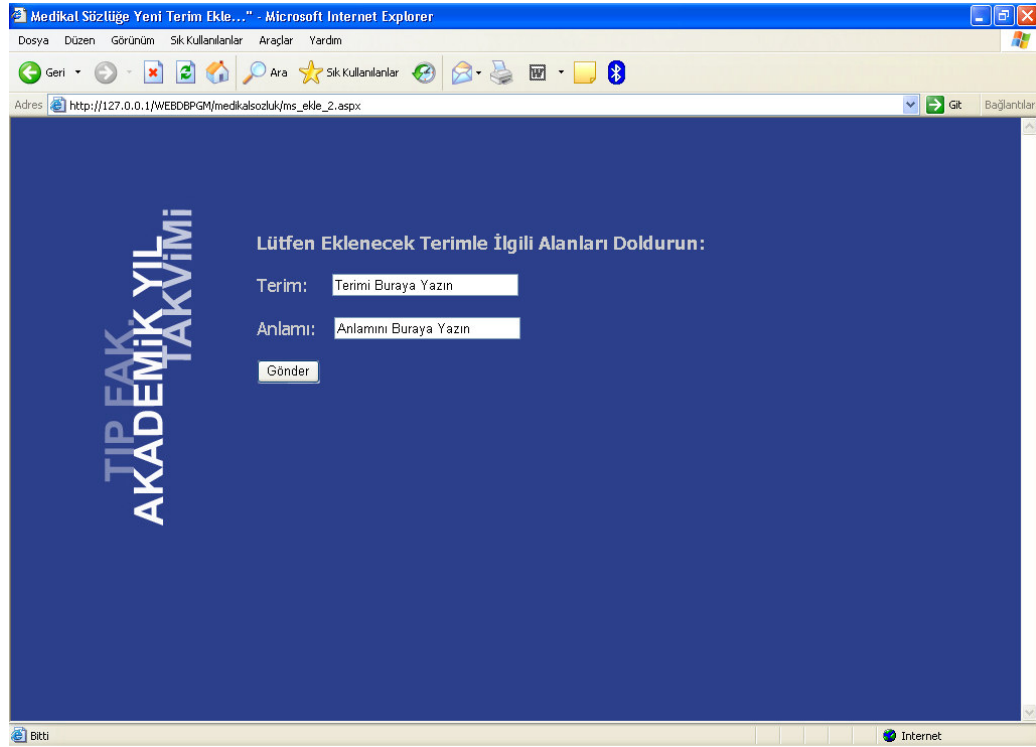


Şekil 4.5. Tıbbi Sözlük Servisi' nin “Dize ile Başlayan Terimler” Kriterine Göre Kullanılması Sonucu

Web Forms tekniđi POST yöntemi ile gönderilen form deđerlerinin işlenmesi için aksi koda belirtilmemiş ise aynı servis kodu ile işlenmesini sağlar. Bu diđer bir ifade ile bir servisin tek bir sunucu tarafı uygulaması ile sağlanması olarak özetlenebilir. Eski teknolojilerde servisin gerçekleştirimi birden fazla sunucu tarafı uygulaması gerektirebiliyor ve bu da geliştiricileri güvenlik gibi ciddi sorunlar ve gerçekleştirim zorlukları ile karşı karşıya getirebiliyordu.

Web Forms öğelerinin verileri sunucu tarafına ulaştığında, sunucu ilk olarak kendi tarafında yarattığı form öğelerine deđerlerini atar. Bundan sonra ulaşan veri kümesi içinde ateşlenen olaya bakar. “ARA” düğmesinin tıklanması ya da “Enter” tuşuna basılması ile ateşlenen onClick olayının işlenmesi gerektiđi sonucuna varır. Sunucu, bu olay sanki sunucu tarafında var olan form öğelerinden biri üzerinde gerçekleşmiş gibi uygun olay işleyicisini arar, bulur ve çalıştırır.

Form verilerinin sunucu tarafına gönderilmesi ve işlenmesini sağlayan aslında form öğesi düğmenin onClick olayı olarak görünse de olay işleyicisi gerçekleştirimde farklıdır. Ziyaretçinin girdiđi verilerin servisi sağlayan tek ve aynı koda gönderilmesi WebForms tekniđinin bir nesne olarak kabul ettiđi “sayfa” ile ilişkilendirdiđi AutoPostBack olayı sayesinde gerçekleşir.



Şekil 4.6. Tıbbi Sözlüğe Terim ve Anlam Ekleme Arayüzü

Dolayısıyla aslında her ne kadar onClick olayı ateşlenmişse de servisin veri giriş sayfası, verilerle geri postalandıđından verilerin işlenerek servis sonucu alınmak istenmiştir. Bu sebeple her yeni servis isteđi sonucu oluşturulan bir sayfa yüklenecektir. Servise her seferinde veri gönderilerek servis cevabı için yeni sayfa

oluşturulmasını sağlayan olay sayfa nesnesinin IsPostBack olayıdır ve bu olay ile Web Forms teknolojisinde varsayılan olarak Page_Load olay işleyicisi ilgilenebilir. Bu servisin gerçekleştirilmesinde de Page_Load olay işleyicisi kodlanmıştır.

Bu olay işleyicisi IsPostBack'in gerçekleştiğini kontrol eder ve gerçekleşti ise bir veritabanındaAra adında bir yordam çağırarak şekilde kodlanmıştır. Çağırıldığı yordam, gönderilen veriler ile veritabanı yani en alt katmana bağlantıyı gerçekleştirdikten sonra sorgulamalar yaparak servisin gerçekleşmesini ve yeni cevap sayfasının oluşturulmasını sağlamaktadır. Olay işleyici, veritabanına, seçili arama kriterine uygun sorguyu yarattıktan sonra bağlantı acip Terimler tablosunu sorgulamaktadır. Eğer seçili arama kriteri açılır liste öğesinde listelenen ilk üç kriterden biri ise kullanıcının girdiği dize arama kriterine göre yapılandırılmış sorgu sonucu sıfır, bir ya da daha fazla terimin bulunduğu bir liste bulur. Bu liste kullanıcıya sunulur. Eğer dördüncü arama kriteri seçili ise girilen dize ile birebir uyuşan terimler arandığından sorgu sonucu sıfır ya da bir terimdir. Terim bulundu ise anlamı da uygun sorgu yaratılarak, Anlamlar tablosunda sorgulanır. Bahsedilen bütün bu işlemler, ayrıntılı şekilde bir sonraki seksiyonda geçen psuedo-code'da görülebilir.

4.1.4 Servis Algoritması

Tıbbi terimler sözlüğü servis mimarisinin orta katmanı olan ve sunum katmanı aracılığı ile kullanıcının girdiği verileri kullanarak en alt katmandaki veritabanında aramanın yapılmasını sağlayan iş katmanı geliştirilirken öncelikle algoritmik bir çözüm hazırlanmıştır. Bu çözüm adımlı psuedo-code şeklinde aşağıda tanımlıdır.

1. Set bağlantıDizesi:= "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" [Uygun veritabanı sağlayıcısını belirten bağlantı dizesini oluştur.]
2. Set fizikselYol:=server→MAPPATH ("medikalsozluk.mdb") [Veritabanının bulunduğu göreceli sanal yolu, uygun fiziksel yola çevir.]
3. Set bağlantıDizesi:= bağlantıDizesi + fizikselYol [fizikselYolu bağlantıDizesi'ne ekle.]
4. Set sorguKoku:= "SELECT * FROM Terimler WHERE terim"
Set sorgu_2:= "SELECT terimin_anlami FROM ANLAMLAR WHERE terim_id=" [Sorguların ana yapısını hazırla.]
5. Set
sorgu[0] := sorguKoku + " LIKE "" + kullanıcıDizesi + "%"
sorgu[1] := sorguKoku + " LIKE '%" + kullanıcıDizesi + "%"
sorgu[2] := sorguKoku + " LIKE '%" + kullanıcıDizesi + "%"
sorgu[3] := sorguKoku + "="" + kullanıcıDizesi
[Seçilmiş arama kriterine uygun sorguyu oluştur.]
6. Set bağlantı:= New OLEDBCONNECTION(bağlantıDizesi)
[Oluşturduğunuz bağlantı dizesini kullanarak veritabanı bağlantı nesnesi yarat.]
- 7.sorguYurutucu:=New OLEDBCOMMAND(sorgu [seciliAramaKriteriIndeksi] , bağlantı)
[Bağlantı nesnesi ve uygun sorgu ile Sorgu Yürütme Nesnesi yarat.]

8. bağlantici → Open ()
[Veritabanı Bağlantısını Aç]

9. Set veriOkuyucu := sorguYurutucu→EXECUTEREADER ()
[Sorguyu yürüt ve sonuç tablodan satırları okumak için okuyucu nesne yarat.]

10. Set bulunanTerimSayisi := 0

11. If seciliAramaKriteriIndeksi>=0 And seciliAramaKriteriIndeksi<=2 Then
Repeat While veriOkuyucu→READ () <> Null
bulunanTerimSayisi := bulunanTerimSayisi +1
If bulunanTerimSayisi = 2 Then
Exit Repeat While
[End If]
[End Repeat]
[Arama kriteri indeksi [0,2] aralığındaysa arama sonucu kaç terim çıkacağını bul.]

veriOkuyucu → CLOSE ()
[okuyucuyu kapat.]

Set veriOkuyucu:= sorguYurutucu→EXECUTEREADER ()
[Aynı sorguyu veritabanında tekrar yürüt ve veri okuyucu yarat.]

If bulunanTerimSayisi >= 1 Then
Set mesajMetni := "Aradığımız Dize ve Kriter ile Uyuşan Terimler
Aşağıdadır.
Anlamını Görmek İsteddiğiniz Terimi
Arama Kutusuna Yazdıktan Sonra 'Dizeye Eşit Olan
Terimleri' Arama Kriterini Seçip Aramayı
Tekrarlayınız.
"
Repeat While veriOkuyucu→READ () <> Null
Set mesajMetni := mesajMetni + veriOkuyucu→ITEM("terim")
[End Repeat]
[Sorgu sonucu bulunan terim sayısı 1 veya daha fazla ise terimleri
kullanıcıya listele]

Else If bulunanTerimSayisi = 0 Then
Set mesajMetni := "Veritabanımızda Verilen
Dize ve Arama Kriteriyle Uyuşan Terim Bulunamadı..."
[Sorgu sonucu bulunan terim sayısı 0 ise kullanıcıya uygun mesajı ver]
[End If]

Else If seciliAramaKriteriIndeksi=3 Then
Set mesajMetni:= "Veritabanımızda Verilen Dize ve Arama Kriteriyle Uyuşan
Terim Bulunamadı..."
If veriOkuyucu→READ () <> Null Then
Set sorgu_2:= sorgu_2 + veriOkuyucu→ITEM("terim_id")
veriOkuyucu→CLOSE ()
sorguYurutucu→DISPOSE ()
Set sorguYurutucu:=New OLEDBCOMMAND (sorgu_2,baglantici)
veriOkuyucu→EXECUTEREADER ()
If veriOkuyucu→READ () <> Null
Set mesajMetni:="<i>" + kullanıcıDizesi + "</i>"
+ veriOkuyucu.ITEM(0) + ""

[Arama kriteri indeksi 3 ise, 1.sorgu sonucu 1 terim bulundu ise onu kimliklendiren
değer ile2.sorguyu hazırla ve çalıştır. Bulunan anlamı kullanıcıya sağla. 1.sorgu
sonucu terim bulunamadı ise kullanıcıya uygun mesajı ver]

[End If]
[End If]
[End If]

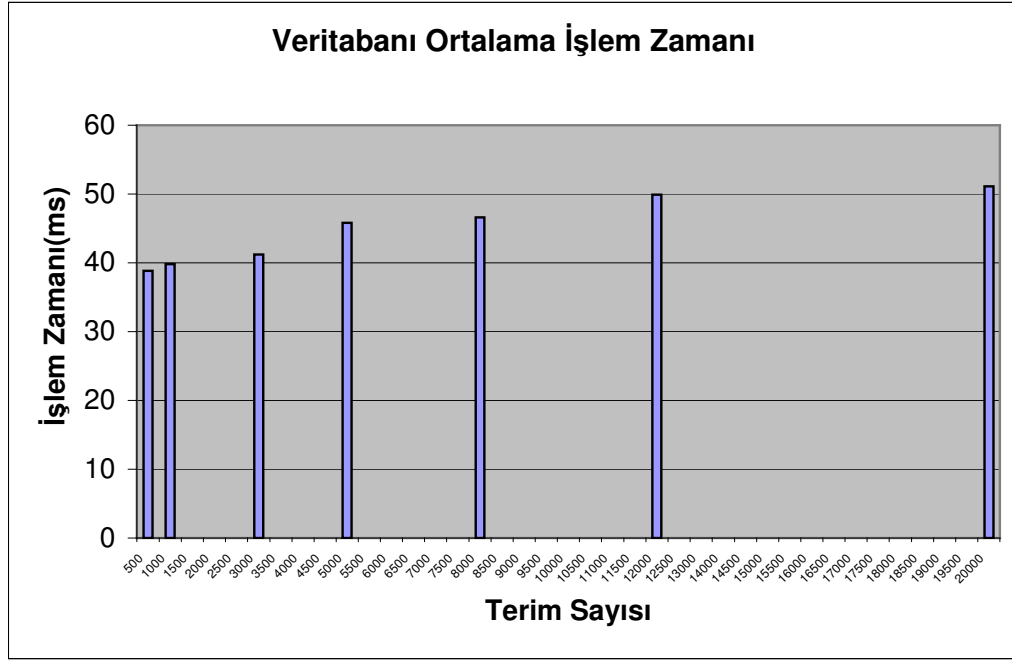
Bu psuedo-code, ziyaretçinin arama için gerekli olan dizeyi girip, uygun arama kriterini seçtikten sonra “ARA” düğmesine tıklaması ve verilerin sunucuya ulaşması ile sunucu tarafından çalıştırılan koddur. Psuedo-code’da büyük harflerle yazılı olanlar ADO.NET tarafından sağlanan nesnelere ait işlevleri(yöntemleri) temsil etmektedir. Bu işlevler veritabanı işlevinin gerçekleşmesi için gereklidirler. Bu kodda ADO.NET’in sağladığı üç tip nesne ve bu tip nesnelere ait işlevleri olan sekiz farklı işlev kullanılmıştır. Kullanılan nesne tipleri OleDbConnection, OleDbCommand ve OleDbDataReader’dır. Bu tipte nesnelere sırası ile veritabanı ile bağlantı kurmaya ve veritabanını açmaya, açık bağlantı üzerinden veritabanında sorgu yürütmeye ve sorgu sonucu tablodaki verileri okumayı gerçekleştirmek için kullanılırlar.

4.1.5. Servis Veritabanı İşlemlerinin Performans Ölçümü Sonuçları

Ölçümler, veritabanı çöplük veri ile doldurulduktan sonra gerçekleştirilmiştir. Bunun gerçekleştirilmesi için öncelikle çöplük veriler için bir veri uzunluğu belirlenmiştir. Terim on, terim anlamı ise 30 karakter uzunluğunda kabul edilmiştir. Tüm veritabanı bu formatta çöplük verilerle doldurulduktan sonra ölçümlere geçilmiştir. Tüm ölçümler sırasında “Dizeye Eşit Olan Terimleri” arama kriteri seçilmiştir. Bunun nedeni bu kriterin en fazla veritabanı işlemi gerektirmesidir. Ölçümler, bir istemci (tarayıcı) tarafından birbiri ardına arama istemleri gönderilerek yapılmıştır.

Ölçümler farklı veritabanı büyüklükleri için gerçekleştirilmiştir; 500 terim, 1000 terim, 3000 terim, 5000 terim, 8000 terim, 12000 terim ve 20000 terim. Bu büyüklükte veritabanları oluşturmak için öncelikle veritabanını istenilen büyüklüğe getiren, aynı zamanda junk data jeneratörü de olan bir uygulama hazırlanmıştır. Her bir veritabanı bu uygulama kullanılarak birer birer yaratıldıktan sonra ölçümlere geçilmiştir. Her veritabanı için 80 ölçüm yapılmış ve bunların ortalamaları ve standard sapmaları hesaplanmıştır. Şekil 4.7, hesaplanan veritabanı işlem zamanı ortalamalarını göstermektedir. Tablo 4.1’de ise farklı terim sayısına göre veritabanı işlemleri tamamlanma süresi ortalamaları ve standard sapmaları listelenmiştir.

Ölçümler, terim sayısının ya da veritabanı büyüklüğünün artması ile tıbbi sözlük servisinin gerektirdiği veritabanı işlemlerinin giderek daha fazla zaman gerektirdiğini göstermektedir. Çizgisel bir artış olmamakla birlikte, işlem zamanının, özellikle 3000 ve 8000 terimden sonra gösterdiği yaklaşık 4.6 ms ve 3.3 ms’lik ani artışlar dikkat çekmektedir. Ayrıca, 500, 1000 ve 3000 terimlik veritabanları ile yapılan ölçümler dikkate alındığında veritabanına bağlantı açma işlemi, arama işlevinden daha fazla zaman gerektirmektedir. Diğer dikkat çekici nokta ise 12000 ve 20000 terimlik veritabanları arasında ciddi bir işlem zamanı farkı bulunmamasıdır.



Şekil 4.7. Tıbbi Terimler Sözlüğü Servisinin Veritabanı İşlemleri Performansı

Fakat burada dikkat edilmesi gereken asıl nokta binler hatta onbinlerle ölçülebilir sayıda terim saklayacak olan veritabanının gerçekleştireceği işlemlerin gerektirdiği sürenin servis kullanıcıları tarafından tolere edilip edilemeyeceğidir. 20000 terimlik veritabanının işlem için gerektirdiği süre olan ortalama 51,1 ms'nin, servis süresi içinde tolere edilebileceği rahatlıkla söylenebilir.

Veritabanı işlemlerinin gerektirdiği işlem süresinin veritabanı boyutu ile büyümesi engellenemez. Diğer bir ifade ile veritabanı genişledikçe veritabanı yönetim sisteminin gerçekleştirdiği işlemler için gereken sürenin uzaması beklenen bir sonuçtur. Zira daha fazla veri işlemek, daha fazla zaman demektir. Fakat işlem süresi iyileştirilebilir. İyileştirme için birçok farklı yola başvurulabilir.

Tablo 4.1 Veritabanında Saklanan Terim Sayısına Göre Servisin Gerektirdiği Veritabanı İşlemlerinin Tamamlanma Süresi Ortalamaları ve Standard Sapmaları

	500 Terim	1000 Terim	3000 Terim	5000 Terim	8000 Terim	12000 Terim	20000 Terim
Ortalama (ms)	38,82	38,93	41,25	45,81	46,61	49,97	51,03
Std. Sapma (ms)	7,21	7,10	7,73	14,48	13,50	12,42	16,45

Ölçümlerde kullanılan arama kriteri kullanılarak servisin gerçekleştirilmesi için veritabanı orta katman tarafından iki kez sorgulanmaktadır. Diğer bir ifade ile bir terimin anlamının bulunabilmesi için veritabanının iki farklı tablosu iki farklı sorguya maruz kalmaktadır. Bu yöntem tablolar arası join işlemi kullanılarak tek bir

sorguya indirgenebilir. Fakat sorgu sayısını azaltmak için kullanılabilir bu yöntem tablolardaki verilerin kartezyen çarpımını da gerektirdiğinden sorgunun daha karmaşık yapıda olmasına neden olacak ve işlem sonucu ortaya çıkan tabloda istenilen verilerin ayıklanması daha fazla zaman alacaktır.

Veritabanı uygulamalarında veritabanı işlem süresinin azaltmanın bir başka yolu ise yine gerçekleştirim sırasında bu çalışmada da kullanılan gerçek zamanlı çalışan veri okuyucu adı verilen veri yapılarından değil de Dataset adı verilen ve işlemlerin gerçekleştirilmesi için gerekli tablolardaki verilerin geliştiricinin insiyatifinde olan süre içerisinde sunucunun hafızasına yüklenip işlemler yapıldıkça oradan okunmasını sağlayan veri yapıları kullanmaktır. Fakat böyle bir gerçekleştirim ciddi hafıza alanı/yeteneği/hızı gerektirir. Özellikle büyük ve orta ölçekli veritabanlarında gerçekleştirilecek işlemlerde Dataset kullanılması geliştirici için bir kolaylık olsa da sunucunun servisi tamamlama süresi açısından özellikle aynı anda yapılan birden fazla istem söz konusu ise ciddi sıkıntılar doğurabilir. Bu sebeple gerçekleştirimde Dataset kullanımı sadece düşük boyutlu tablolar için pratik bir çözüm olarak tercih edilmelidir.

Bu çalışmada, işlem süresini asıl belirleyici faktör olarak veri sağlayıcı görülmektedir. Farklı veri sağlayıcılarının, özellikle veritabanına bağlantı açma işleminde oldukça farklı performansları olduğu raporlanmıştır (32). Ayrıca, MS Access, veritabanı yönetim sistemleri arasında yapılan testlerde veritabanı bağlantısı açma işlemini en uzun sürede gerçekleştiren yönetim sistemi olarak raporlanmıştır (32).

Veritabanı uygulamaları geliştirilirken veritabanı yönetim sistemlerinin uygun veri sağlayıcılar ile eşleştirilmesi işlem süresinde ciddi bir iyileştirme yaratabilir (32). Bu çalışmada, Access gibi bir Microsoft ürünü olan Microsoft Jet OLEDB veri sağlayıcısı kullanılmıştır. Fakat gösterilmiştir ki (32) Access, ODBC veri sağlayıcısı ile erişildiğinde, en azından bağlantı açmak için çok ciddi bir performans farkı yaratmaktadır. (32)'de yapılan testlerde Access'e Microsoft Jet OLEDB veri sağlayıcısı ile ilk erişim (bağlantı açma) yaklaşık 50 ms alırken, Microsoft'un ODBC kullanan istemciler için geliştirdiği veri sağlayıcısı aynı şartlar altında bağlantı açma işlemini 4ms gibi kısa bir sürede gerçekleştirmektedir. Dolayısıyla MS Access ile kullanıldığında daha kısa bağlantı açma ve işlem zamanı sağlayan farklı veri sağlayıcıları ve motorları ile ölçümler tekrar edilmelidir.

Bir diğer dikkat edilmesi gereken nokta seçilen veritabanı yönetim sisteminin kısıtlamalarıdır. Bu kısıtlamalardan biri, yönetim sisteminin servis veritabanının ne kadar büyümesine izin vereceğidir. MS Access, bir veritabanının 2 GB'a kadar büyümesine izin verebilmektedir. Fakat veritabanının 2 GB'a ulaşmasının ardından MS Access kullanılmaya devam edilecek ise veritabanının genişlemesi için alternatif bir yol kullanılabilir. Aynı yapıda yeni bir veritabanı daha hazırlayarak veritabanı dosyaları arasında tablolar marifeti ile kurulacak bir mantıksal bağ, desteklenebilir maksimum veritabanı (dosya) boyutu sorununa bir çözüm getirebilir. Fakat bu çözümün işlem performansını nasıl etkileyeceği ölçümlenmelidir.

Sonuç olarak, veritabanının gerektirdiği işlem süresi servisin karakteristiği ve gerektirdiği cevap verme süresi içinde tolere edilebilir görünmektedir. Bununla birlikte, veritabanının performansı farklı tablo alanı büyüklüklerinde (bu ölçümlerde onbir karakter, 30 karakter ve on karakter olarak kullanılmışlardır) de test edilmelidir. Bunun yanında, özellikle aynı anda ulaşan istem sayısının performansa etkisi ölçülmelidir. Farklı istemci sayısı ve istem bombardımanı ile gerçekleştirilecek testler performans hakkında ciddi fikirler verebilir. Microsoft Access, aynı anda 255 kullanıcının veritabanına ulaşmasını sağlasa da çok kullanıcıli/istemli bir ortamda veritabanı performansı servis cevap süresini tek başına belirler hale gelebilir.

4.2. Akademik Yıl Ders Takvimi Servisi

Akademik yıl ders takvimi servisi, Akdeniz Üniversitesi Tıp Fakültesi öğrencilerinin yıllık akademik takvimdeki herhangi bir gün hangi saatte hangi dersin olduğunu öğrenebilmelerini sağlayan bir hizmettir. Bu hizmetin kullanımı bir arabirim ile oldukça basitleştirilmiştir.

4.2.1. Ders Takvimi Veritabanı Fiziksel Yapısı

Ders Programı servisi de tıpkı Tıbbi sözlük servisi gibi önceki bölümde anlatılan veritabanı teknolojileri, uygulama geliştirme mimarileri ve ortamları benimsenerek geliştirilmiştir. Buna göre, servisin veritabanını oluşturan tablolar, özellikleri, birincil anahtarları ve aralarındaki ilişkiler Şekil 4.8'de gösterilmiştir.

Servis için kullanılacak olan veritabanının veri modeli dönem-kurul ilişkisi temel alınarak tasarlanmıştır. Veri modelinin MS Access'te fiziksel yaratımı sonucu veritabanı dört tablodan oluşmuştur.

Bunlardan Dönemler adlı tabloda her bir tıp fakültesi tarafından tanımlanmış her bir dönem, dönem numarası ve o dönemi oluşturan kurul sayısı ile saklanmaktadır. Kurullar tablosu ise kurul numarası, kurulun bağlı olduğu dönem numarası, varsa kurul adı, kurulun başlangıç ve bitiş tarihlerini ve her bir kurula ait eşsiz kurul kimliği özelliklerinden oluşmaktadır.

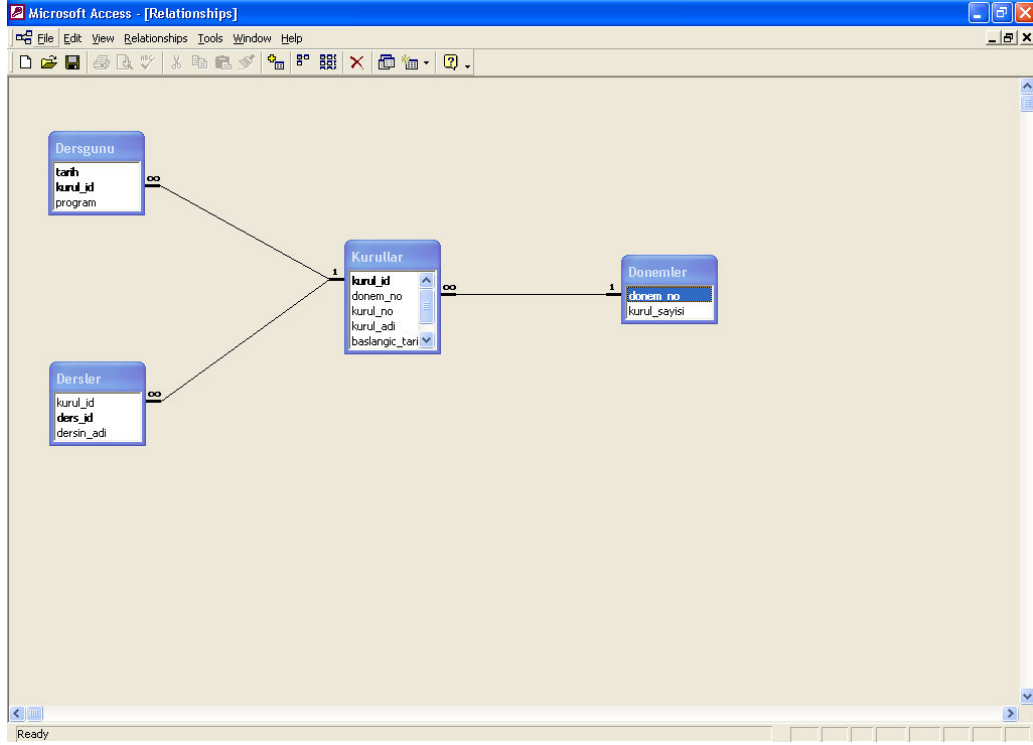
Dönemler ve Kurullar tabloları arasında Bire-Çok bir ilişki tipi vardır, zira uygulamada bir dönem birçok kuruldaki oluşmaktadır. Kurullar tablosu veritabanının diğer iki tablosu ile de ilişkiindedir. Bunlardan Dersler tablosunda, verilen her bir ders adı, verildiği kurul kimliği ve eşsiz bir ders numarası ile temsil edilmektedir. Bu iki tablo arasında da Bire-Çok bir ilişki bulunmaktadır.

Kurullar tablosunun yine Bire-Çok tipte ilişkili olduğu bir diğer tablo ise Dersgünü tablosudur. Bu tabloda akademik takvimdeki her bir gün, o günün tarihi, günün içinde bulunduğu kurulun kimliği ve günün programının sakladığı program özellikleri ile saklanır.

4.2.2. Akademik Yıl Ders Takvimi Sorgulama ve Ekleme Arayüzü

Öğrencilerin istedikleri tarihe ait ders programına ulaşabilmelerini sağlamak için kullanımı oldukça basit bir arabirim tasarlanmıştır. Bu arayüz üç açılır liste, bir metin kutusu, bir takvim ve metin kutusu aracılığı ile girilen yıl değerinin geçerlilik kontrolü için kullanılan fakat formun fiziksel yapısında görünmeyen bir validatörden

oluşmaktadır. Arayüzde bulunan birimler tıpkı tıbbi terimler sözlük servisinde olduğu gibi bir Web Formu içinde tanımlıdır.

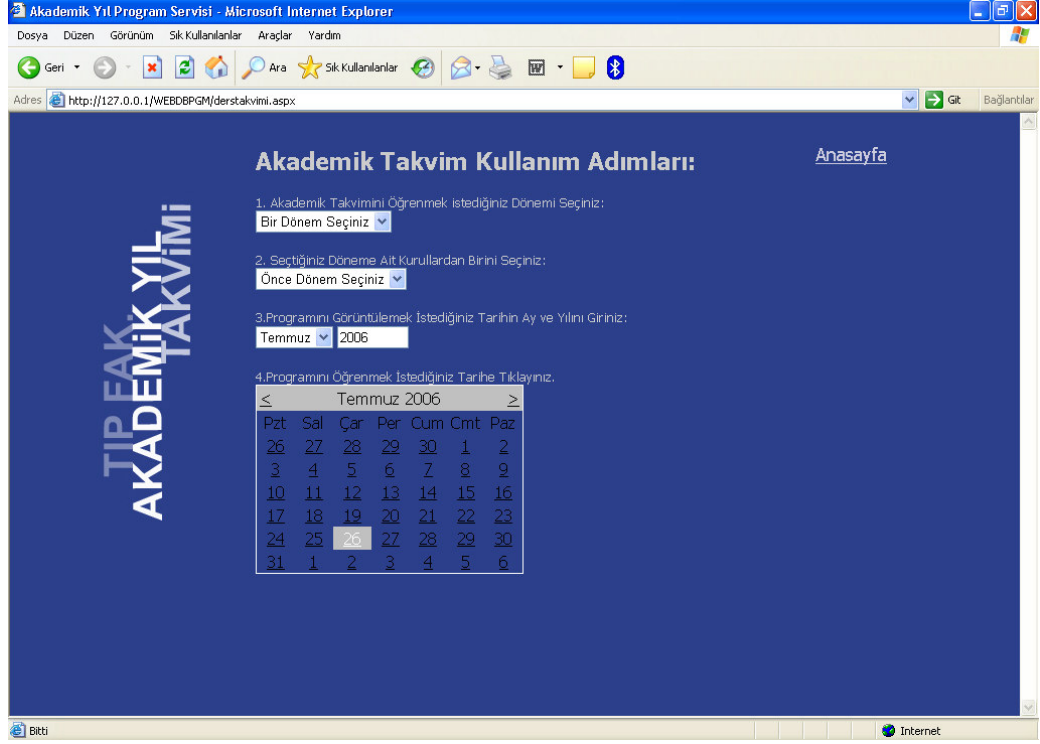


Şekil 4.8. Günlük Ders Programı Servisi Veritabanı İlişkileri

Bu servisin arayüzü de veri girişi için adımlara bölünmüştür. Buna göre arayüz kullanıcıdan (öğrenciden) ilk olarak bir açılır liste vasıtasıyla dönem seçmesini ister. Seçilen döneme göre veritabanı sorgulanarak 2.adımda seçilmesi gereken ve seçili döneme bağlı kurullar bulunur. Sorgu sonucu elde edilen kurul adlarını veya numaralarını, 2.adım veri girişini belirleyen açılır liste maddeleri yapan servis bu kez kullanıcıdan kurul seçmesini ister. Kullanıcılar bu mekanizma ile veri giriş adımlarına uymak zorunda bırakılmışlardır.

Uygun kurulun seçiminin ardından akademik yıl ve o yıla ait ay seçildiğinde, seçilen ayın takvimi görüntülenir. Görüntülenen takvimde programını öğrenmek istediği güne tıklayarak kullanıcı o güne ait ders programına ulaşır. Sistem eğer seçili güne ait kayıtlı bir program bulamaz ya da haftasonuna ait bir gün seçildiğini tespit etti ise kullanıcıyı bu yönde bilgilendiren bir cevap sayfası gönderir. Şekil 4.9'da servisin arayüzü gösterilmiştir.

Şekil 4.10'da ise Dönem II Kurul I için 17 Haziran 2006 tarihli günündeki program için yapılan sorgu sonucu görülmektedir.

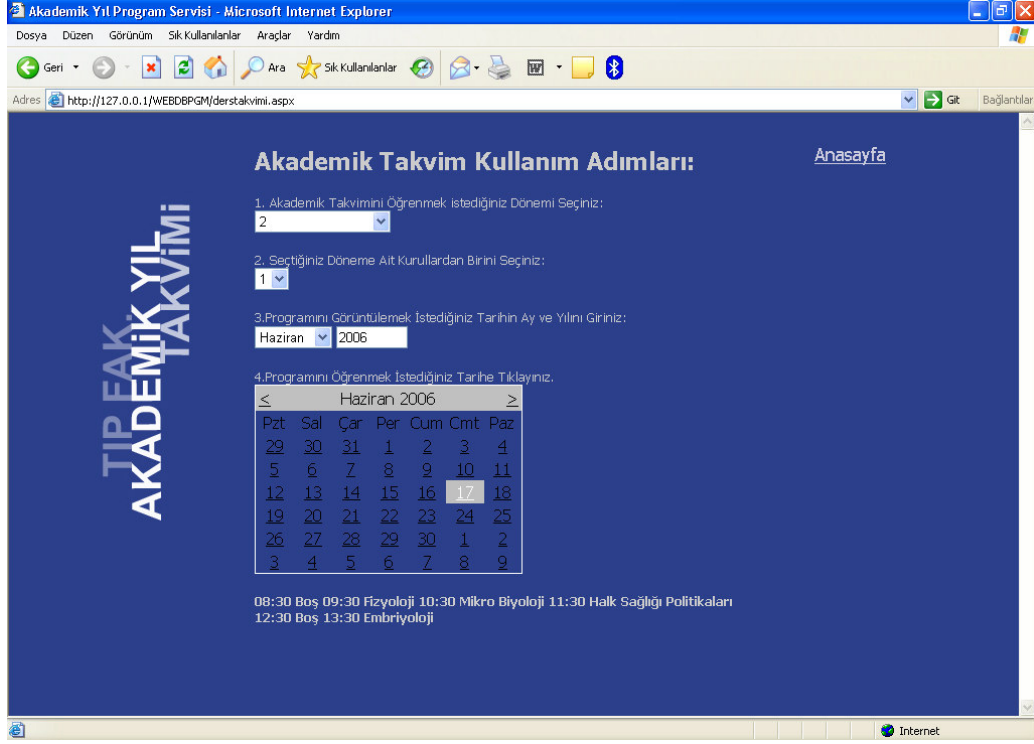


Şekil 4.9. Günlük Ders Programı Servisi Kullanım Arayüzü

Akademik yıl takvim servisinin kullanılabilmesi için her akademik yıl başlangıcında o yıla ait akademik takvimin (ders programının), servisin veritabanına girilmesi gereklidir. Bunun gerçekleştirilebilmesi için bir ekleme arayüzü geliştirilmiştir. Ekleme arayüzü Şekil 4.11’de gösterilmektedir. Arayüz, kullanıcıları yönlendirici 5 adıma bölünmüştür.

Ekleme arayüzü, beş açılır liste, bir metin kutusu, bir takvim, üç düğme ve iki etiket ögesi kullanılarak oluşturulmuştur. Arayüzde en başta yer alan açılır listelerden ilk üçü hangi dönem, kurul ve aya ait ekleme yapılacaksa onların seçimini sağlamaktadır. Kurul listesi, ekleme yapılacak dönem seçildikten sonra o döneme ait kurulları listeler. Listelenecek kurullar, kullanıcı dönem seçer seçmez seçtiği döneme ait kurulların veritabanında sorgulanması için arayüz istek gönderir. Ardından, veritabanında o döneme ait kurullar sunucu tarafından açılır listeye eklendikten sonra arayüz kullanıcının tarayıcısına geri postalanır. Ayın belirlenebildiği açılır liste ve arayüzdeki tek metin kutusu, ekleme yapılacak günün ait olduğu ay ve yılın seçilerek takvimde gösterilmesini sağlamak için vardır. Takvim ise seçili yıl ve yine o yıla ait seçili ayın günlerini görüntüler. Takvim, kullanıcının eklemek istediği ders programının hangi güne ait olduğunu seçmesini sağlar.

Dönem-Kurul ve tarih/gün seçimi yapıldıktan sonra kullanıcının, o günün hangi ders saatinde o dönem-kurul ikilisine kayıtlı derslerden hangisinin olduğunu belirleyebilmesi için açılır listelerden ilki saati, diğeri de o saatte yer alan dersi seçmesini sağlar. Fakat bu açılır listelerdeki değerlerin görüntülenebilmesi için kullanıcının dönemin ardından bir kurul seçmesi gerekmektedir.

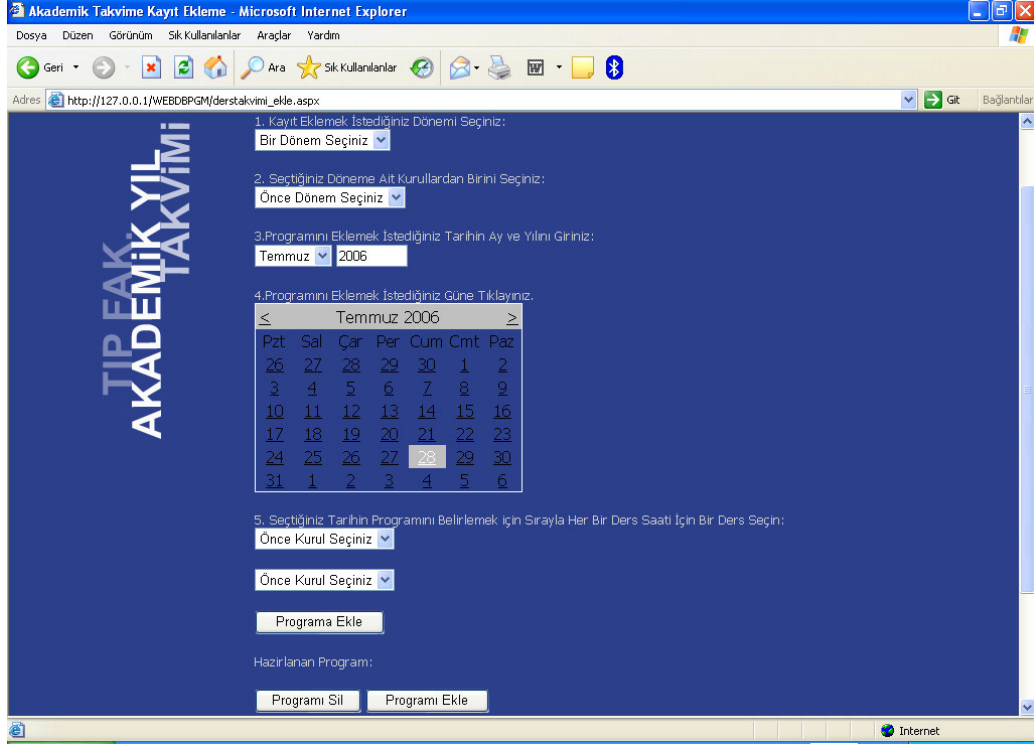


Şekil 4.10. Günlük Ders Programı Servisi ile Yapılan Bir Sorgu Sonucu

Kullanıcı uygun kurulu seçtiği anda, veritabanında, seçili (dönem, kurul) ikilisine ait dersler sorgulanır. Bulunan dersler alttaki açılır listede, ders saatleri ise üstteki açılır listede görüntülenir. Bu iki açılır listenin hemen altında “Programa Ekle” düğmesi bulunur. Saat ve ders seçimini yapan kullanıcı bu düğmeye tıkladığında açılır listelerden seçilmiş (saat,ders) ikilisi takvimde seçili tarihin/günün programına eklenir.

Bu işlem günün 8 saatini dolduracak şekilde tekrarlanır ve sonuçta günün programı arayüzdeki etikette her bir (saat, ders) ikilisi eklentisinin ardından görülebilir. Kullanıcı, (saat,ders) ikilisi eklemesi sırasında bir hata yaparsa “Programı Sil” düğmesini kullanarak o tarihe/güne ait programı baştan yaratmaya başlayabilir. Eğer o tarihe/güne ait program başarı ile oluşturulmuş ise kullanıcı, “Programı Ekle” düğmesine basarak ekleme işlemini gerçekleştirebilir.

Akademik takvim servisine bir tarihe/güne ait ders programı girişi yapan bir kullanıcı işlem sonucuna göre bilgilendirilir. Eğer işlem başarılı ise eklenen program ve işlemin başarılı şekilde tamamlandığına dair kısa bir mesaj görüntülenir. Diğer taraftan, eğer kullanıcının ders programı girmek istediği dönem, kurul ve tarih üçlüsüne ait bir program veritabanında zaten kayıtlı ise kullanıcıya durumu bildirir kısa bir mesaj ulaşır cevap sayfası olarak. Kullanıcı eğer kayıtlı programda bir değişiklik yapmak isterse hazırlanma aşamasında olan güncelleme arayüzünden bu işlemi gerçekleştirebilir.



Şekil 4.11. Akademik Yıl Takvimi Servisinin Ders Programı Ekleme Arayüzü

4.2.3. Servis Gerçekleştirmede Olaylar ve Olay İşleyiciler

Akademik Ders Takvimi Servisi'nin gerçekleştirim mimarisinin orta katmanı Tıbbi Sözlük Servisi'nin orta katmanına nispeten daha karmaşık yapıdadır. Bu servis için geliştirilen sorgulama arayüzünün içerdiği görsel veri giriş ögesi daha fazla ve çeşitlidir. Bu öge tipi çeşitliliği ve öge sayısındaki artış arayüzün kullanımı için gerekli olan olayların sayısı ve çeşitliliğini de beraberinde getirmiştir. Örneğin takvim ögesinin uygun şekilde kullanımı için takvim üzerinden ateşlenen iki farklı olay gereklidir ve bunlarla ilişkilendirilmiş olay işleyicileri kodlanmıştır.

Bunun yanı sıra, farklı tipte öğeler vasıtası ile ateşlenen olaylar diğer arayüz öğelerini de etkileyebildiğinden bu servisin orta katman gerçekleştirimi daha karmaşıktır. Fakat servisin gerçekleştirimini daha da karmaşıklaştıran, öğeler kadar kullanılması gerekli olan veri tipleridir. Özellikle Tarih/Zaman veri tipinin, veritabanında gerektirdiği veri kısıtlamaları (örneğin servis 2006-2007 akademik yılından itibaren günlük dersleri saklayacağından) da göz önünde bulundurulunca arayüzü bazı girdi kontrolleri yapacak şekilde de tasarlamak ve gerçekleştirmek gerekliliği fark edilmiştir.

Bu serviste bir olay işleyicisi, ilk kez sayfa istemci tarafından istenip bu istem sunucuya ulaştığında çağrılır. Çağrılan işleyici Page_Load olay işleyicisidir ve ararımın servisin gerçekleştirilmesi için gerek duyduğu ilk verileri veritabanı bağlantısı ve sorgulaması yaparak bulur. Bu sorgulama veritabanında kaç dönem için akademik yıl saklandığını bularak kullanıcının ders programına bakacağı dönemi seçmesini sağlayacak olan ilk adımdaki açılır listenin değerler listesini belirlemek

için yapılmaktadır. Buna göre, veritabanında kayıtlı dört dönem bulunmaktadır ve bunlar açılır liste değerleri olarak belirlendikten sonra istemciye (tarayıcı) arayüz görüntülenmek üzere gönderilir.

Potansiyel olarak ateşlenebilen diğer tüm olaylar kullanıcı tarafından ateşlenebilir. Kullanıcının ateşleyebileceği olaylardan biri ilk açılır listedeki dönemlerden uygun olanı seçmesi ile ateşlenen OnSelectedIndexChanged olayıdır. Açılır listedeki seçili değer değiştiği anda bu olay ateşlenir. Bu durumda bu açılır liste için ateşlendiğinde seçili dönem değeri değiştiğinden ve bu öge için "AutoPostBack" değeri "true" olarak belirlendiğinden istemci ateşlenen bu olay için yeni bir istem oluşturup gönderir. İstemi alan sunucu gelen veriden açılır liste ögesi için OnSelectedIndexChanged olayının ateşlendiğini belirler ve ilişkilendirilmiş olay işleyicisini bulup çalıştırır.

Bu serviste ilk açılır menü için değer değiştirildiğinde servis, veritabanına bağlanıp seçilen dönem için kaç kurul olduğunu sorgulayıp saptadıktan sonra kurulların listelendiği açılır listenin değerlerini değiştirir. Buna göre 1. açılır listeden Dönem II seçildiğinde, 2. açılır liste 1, 2, 3 ve 4 değerlerinden oluşan bir listeye, Dönem III seçildiğinde ise 1, 2, 3, 4 ve 5 değerlerinden bir listeye görüntülenecektir.

Kullanıcının ateşleyebileceği bir diğer olay akademik yıl değerinin değiştirilmesini sağlayan metin kutusunun "OnTextChanged", yani metin kutusunda yazılı metnin değişmesi olayıdır. Metin kutusu istenilen akademik yıla ait takvimin görüntülenmesini sağlayan ögedir. Takvim ögesi istenilen akademik yıla geçmeyi birkaç tıklama ile halletse de kullanıcı rahatlığı düşünüldüğünden tek bir girdi ile istenilen akademik yıla geçiş yapılması metin kutusu kullanılarak sağlanmıştır.

Arayüzde görünmeyen fakat form içinde bulunan bir öge olan validatör (RangeValidator), metin kutusuna kullanıcı tarafından girilen akademik yıla ait verinin veritabanında bulunup bulunmadığını kontrol etmeye gerek kalmadan metin kutusu değeri olarak girilebilecek değer aralığını tanımlamak için kullanılmıştır. Diğer bir ifade ile metin kutusuna girilebilecek değerler kümesini belirleyen, sınırlayan ve girilen değeri kontrol eden bir mekanizma sağlar validatör. Böylelikle kullanıcı metin kutusuna yeni bir yıl değeri girdiğinde eğer girilen yıla ait akademik takvim bilgileri saklı değilse o akademik yıla ait takvim görüntülenmeyecek ve böylece gereksiz veritabanı sorgulamalarının önüne arayüz sayesinde geçilmiş olunacaktır. Kullanıcı geçersiz bir akademik yıl girdiğinde veritabanının gereksiz sorgulanmasını engellemek için, validatör, ilişkilendirildiği metin kutusunun hemen yanında kullanıcıya "Girdiğiniz Akademik Yıl İçin Veritabanında Kayıt Yoktur" uyarısını da görüntüler.

Arabirimim bir diğer ögesi olan 3. açılır liste, takvimin göstereceği ayı belirlemek için vardır. Takvimin göstereceği ay birkaç tıklama ile belirlenebilse de kullanıcı takvimin gösterilen ayı değiştiren sağ ve sol oklara tıklayarak istediği aya gidebilse de daha rahat bir şekilde açılır listeden iki tıklama ile istediği ayın takvimde görüntülenmesini sağlayabilmektedir. Bu ögenin de OnSelectedIndexChanged olayı dikkate alınmış ve olay işleyicisi kodlanmıştır (ayListesindeAyDegisti). Kullanıcı

seçili ayı değiştirdiği anda takvim o sırada geçerli yılın (metin kutusu ile belirlenir) seçili ayını görüntüler.

Takvim, arabirimin en önemli ögesidir. Kullanıcı tarafından tanımlanmış yıla ait yine kullanıcının seçtiği ayı gösterir. Arayüz tarayıcı tarafından ilk kez görüntülediğinde takvim, sunucuda o günün tarihini seçili olarak gösterir. Daha sonra kullanıcı isterse gösterilen yıl ve/veya yıla ait ayı değiştirebilir. Gösterilen ay takvimin başlığında adı yazılı olandır ve bu adın iki yanındaki '<' ve '>' işaretleri düğme işlevi görmektedir. Bunlardan '<'e kullanıcı tıkladığında gösterilmekte olan aydan bir önceki ay gösterilir. '>'e tıkladığında ise takvimde bir sonraki ay görüntülenir. Bu işlem aslında bir olayın ateşlenmesi sonucu gerçekleşir. Bu düğmelere tıkladığında ateşlenen olay onVisibleMonthChanged olayıdır ve takvime özgüdür.

Takvimin gösterdiği aya ait herhangi bir güne tıkladığında ise seçili tarih değişmiş olur. Bu durum da takvim ögesi için bir başka olaydır ve adı onSelectionChanged'dir. Kullanıcının takvim üzerinde seçili günü değiştirmesi ile ateşlenen bu olaya bağlı olay işleyicisi veritabanına bir bağlantı açar ve seçili gün tarihine ait ders programını sorgular.

Olay işleyicisi, bunu gerçekleştirmek için önce dönem ve kurul seçiminin yapıldığı açılır listelerdeki seçili değerleri kullanarak bir sorgu oluşturur ve seçile döneme ait kurulun veritabanındaki Kurullar tablosundan kimliklendirici değerini bulur. Ardından bu değer ve takvim üzerinde seçili tarihi (kullanıcı tarafından tıklanmış) kullanarak ikinci bir sorgu hazırlar ve veritabanındaki Dersgunu tablosundaki program özelliğini sorgular. Eğer veritabanında seçili döneme ait seçili kuruldaki seçili tarihe kayıtlı bir program bulunuyorsa tarayıcıya sayfa içinde metin şeklinde gönderilir. Bulunamadı ise uygun mesaj ile kullanıcı bilgilendirilir.

4.2.4. Servis Algoritması

Olayların ve bunlarla ilişkili olay işleyicilerinin sayısının çok olması nedeni ile arabirimde önemli olduğuna inanılan olay işleyicilerinden üçünün psuedo-code'u aşağıda verilmiştir. Bunlardan ilki takvim ögesinin onSelectionChanged olayını işleyen ve servisin gerektirdiği işi yapan işleyicinin kodudur.

1. Set baglantiDizesi:= "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" [Uygun veritabanı sağlayıcısını belirten bağlantı dizesini oluştur.]
2. Set fizikselYol:=server→MAPPATH ("derstakvimi.mdb") [Veritabanının bulunduğu göreceli sanal yolu, uygun fiziksel yola çevir.]
3. Set baglantiDizesi:= baglantiDizesi + fizikselYol [fizikselYolu baglantiDizesi'ne ekle.]
4. Set kurulKimligiSorgusu:= "SELECT kurul_id FROM Kurullar WHERE donem_no=" + seciliDonem + " AND kurul_no=" + seciliKurul
5. Set programSorgusu:= "SELECT program FROM Dersgunu WHERE tarih=" programSorgusuKuyruk:= " AND kurul_id="
6. Set baglantici:= New OLEDBCONNECTION(baglantiDizesi)

[Oluşturduğun bağlantı dizesini kullanarak veritabanı bağlantı nesnesi yarat.]

7. Set sorguYurutucu:=New OLEDBCOMMAND(kurulKimligiSorgusu , baglantici)
[Baglanti nesnesi ve uygun sorgu ile Sorgu Yürütme Nesnesi yarat.]
8. bağlantici → Open ()
[Veritabanı Bağlantısını Aç]
9. Set veriOkuyucu := sorguYurutucu→EXECUTEREADER ()
[Sorguyu yürüt ve sonuç tablodan satırları okumak için okuyucu nesne yarat.]
10. If veriOkuyucu→READ () <> Null Then
Set kurulKimligi:=veriOkuyucu→ITEM (0)
Set programSorgusu:= programSorgusu + “#” + takvim→SelectedDate→DAY() +
“/” + takvim→SelectedDate→MONTH () + “/” +
takvim→SelectedDate→YEAR () + “#”
Set programSorgusu:= programSorgusu+programSorgusuKuyruk+kurulKimligi
[Seçili Dönem ve Kurul numaralarını kullanarak o kurulun kimliklendiricisini
veritabanını sorgulayarak bul ve seçili güne ait ders programını bulmak için
hazırlanan sorguda seçili tarihi de kullanarak 2. sorguyu hazırla.]
[Tarih/Zaman tipinde veri sorgulamak için sorguyu uygun formatta hazırla.]

veriOkuyucu→CLOSE ()
sorguYurutucu→DISPOSE ()

Set sorguYurutucu:= New OLEDBCOMMAND (programSorgusu, baglantici)
Set veriOkuyucu:= sorguYurutucu→EXECUTEREADER ()
[Seçili tarihe ait ders programını veritabanında sorgula.]

If veriOkuyucu→READ() <> Null Then
Set mesajMetni:= “” + veriOkuyucu→ITEM (0) + “”
Else
Set mesajMetni:= “Bu tarihe ait ders programı bulunamadı.”
[End If]
Else
mesajMetni:= “Veritabanında seçili dönem ve kurula ait veri bulunamadı. Bu
durumu e-posta yoluyla yöneticiye bildirebilirsiniz.”
[End If]

Seçili dönem, kurul ve tarihe ait ders programı kayıtlı ise kullanıcı ders programını tarayıcısında sunucu tarafından oluşturulan cevap sayfasında görebilecektir. Eğer bulunamadı ise sunucu kullanıcının tarayıcısına durumu uygun mesaj ile bildiren sayfayı yollayacaktır.

Kullanıcı arayüz sayfasını ilk kez görüntülediğinde kurulları listeleyen açılır liste boştur. Bu açılır liste kullanıcı tarafından dönem seçilirse seçilen döneme ait kurullar ile doldurulacaktır. Bu işlem de sunucu-tarayıcı arası bir gidiş-dönüş işlemidir ve seçilen döneme ait kurul verileri veritabanından okunduktan sonra sunucu tarafından açılır liste hazırlanır ve arayüz tekrar postalanır. Böylelikle sadece veritabanında yer alan dönemler ve bu dönemlere ait kurullar için sorgulamalar yapılması sağlanmıştır. Bunu gerçekleştiren dönemlerin listelendiği açılır liste öğesinin olayı onSelectedIndexChanged’in işleyicisidir. İşleyici yordama donemListesindeDonemSecildi adı verilmiştir. Aşağıda bu yordama ait psuedo-code verilmiştir.

1. Set baglantiDizesi:= "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
 - [Uygun veritabanı sağlayıcısını belirten bağlantı dizesini oluştur.]
2. Set fizikselYol:=server→MAPPATH ("derstakvimi.mdb")
 - [Veritabanının bulunduğu göreceli sanal yolu, uygun fiziksel yola çevir.]
3. Set baglantiDizesi:= baglantiDizesi + fizikselYol
 - [fizikselYolu baglantiDizesi'ne ekle.]
4. Set kurulSayisiSorgusu:= "SELECT kurul_sayisi FROM Donemler WHERE donem_no="
5. Set baglantici:= New OLEDBCONNECTION(baglantiDizesi)
 - [Oluşturduğın bağlantı dizesini kullanarak veritabanı bağlantı nesnesi yarat.]
6. Set kurulSayisiSorgusu:= kurulSayisiSorgusu + secilenDonem
 - []
7. Set sorguYurutucu:=New OLEDBCOMMAND(kurulSayisiSorgusu , baglantici)
 - [Baglantı nesnesi ve uygun sorgu ile Sorgu Yürütme Nesnesi yarat.]
8. baglantici → Open ()
 - [Veritabanı Bağlantısını Aç]
9. Set veriOkuyucu := sorguYurutucu→EXECUTEREADER ()
 - [Sorguyu yürüt ve sonuç tablodan satırları okumak için okuyucu nesne yarat.]
10. If veriOkuyucu→READ () <> Null Then
 - Set kurulSayisi := veriOkuyucu→ITEM (0)
 - Else
 - Set mesajMetni := "Seçtiğiniz Döneme ait kayıt bulunamadı. Bu durumu sistem yöneticisine e-posta ile bildirebilirsiniz."
 - [End If]
 - []
 - kurulListesi→Items→CLEAR ()
 - [seçili döneme ait kurulları listeleyen açılır listeyi temizle.]
 - Repeat for i=1 to kurulSayisi
 - kurulListesi→Items→INSERT (i-1, New LISTITEM (i , i))
 - [End of Repeat]
 - [kurulListesi açılır listesine seçili döneme ait listeleyeceği kurulları ekle.]

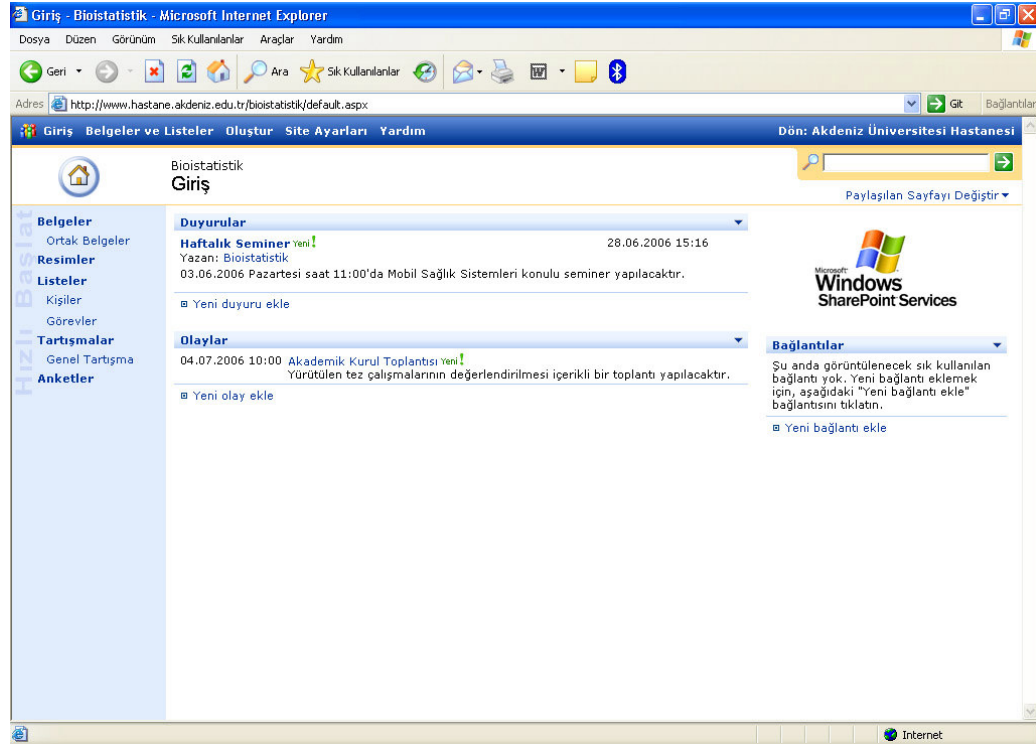
4.3. Anabilim Dalı Belge Yönetimi Servisi

Belge yönetim servisi geliştirilirken dikkate alınması gereken en önemli konulardan biri servisin halledeceği departmandaki belge hareketinin çözümlenmesidir. Bu çözümlenme sayesinde uygun sınıflandırma ve adlandırma ile belge kitaplıkları yaratılır ve hazırlanır. Bu çözümlenmede dikkat edilmesi gereken bir nokta da hangi belge kitaplığına kimlerin geçiş hakkına sahip olacağıdır. Departman çalışanları, hiyerarşi ve belge hareketi üçlüsü tasarım yapılırken incelenmiştir. Bu incelemelerden sonra belge yönetimi sağlayan portal çözümlerinden departmanın ihtiyaçlarını karşılayabilecek olanlar arasından bir seçim yapılarak kurulumu yapılmıştır. Portal çözümü seçimi sırasında bir diğer belirleyici anabilim dalı personelinin çözümü geliştiren firmanın diğer teknolojileri ile aşinalığıdır. Microsoft'un portal çözümü olarak şu anda piyasada bulunan Sharepoint Portal

Server'in seçilmesinde anabilim dalı personelinin bu firmanın teknolojilerini kullanıyor olmalarının da etkisi vardır.

Servis gereği anabilim dalı çalışanları için kullanıcı isimleri ve şifre ikilileri oluşturulmuştur. Kullanıcıların belge kitaplıklarına erişim hakları düzenlenmiştir. Portal her kullanıcısı için ayrıca bir alan tahsis etmiştir.

Belge yönetimi servisi <http://www.hastane.akdeniz.edu.tr/bioistatistik> adresinde, dekanlığa ait bir sunucu üzerinde kurulu olan Sharepoint Portal sunucusu kullanılarak hazırlanmıştır. Şekil 4.12'de servis için giriş sayfası görüntülenmektedir.



Şekil 4.12. Belge Yönetim Servisi Giriş Sayfası

Belge kitaplıkları ve çalışma alanları anabilim dalında yapılan çalışmalara paralel olarak yaratılmıştır. Bunlar arasında seminerler belge kitaplığı, tezler belge kitaplığı sayılabilir. Çalışma alanları ise kurul toplantıları için vardır. Anabilim dalımızda bulunan Akademik Kurul toplantıları için bir çalışma alanı yaratılmıştır.

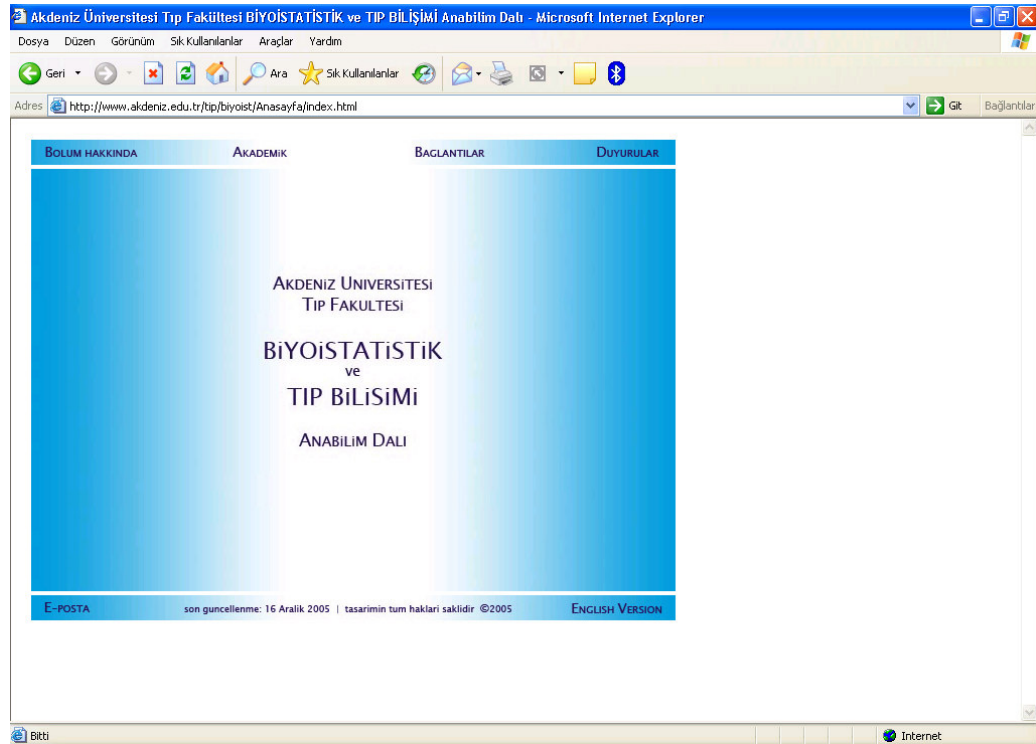
4.4. Anabilim Dalı Resmi Web Sitesi

Servislerin yanında hazırlanmış site toplulukları da barındırır bir portal. Bu çalışmada geliştirilen portal anabilim dalımızın sitesini de barındırır. Site, JavaScript, Dinamik HTML ve CSS gibi istemci tarafı teknolojiler kullanılarak geliştirilmiştir.

Söz konusu teknolojilerin kullanımı ile site içeriğinin neredeyse hepsinin ilk istemde sunucu tarafından istemciye transmisyonu gerçekleştirilir. Fakat tarayıcının görüntülediği anasayfadır. Ana sayfa üzerindeki menü linklerinden ziyaretçi mevcut

içeriğe ulaşabilir. Fakat bu linkler bahsedildiği gibi sunucuya bir başka sayfa içeriğinin görüntülenmesi için istem göndermez. Onun yerine katmanlama tekniği ile saklanan ve ilk istemde sunucu tarafından gönderilen içeriğin istediği bölümüne ulaşmak isteyen ziyaretçi karşılık gelen linke tıkladığında tarayıcıda görüntülenen anasayfa katmanı yok olur ve onun yerine tıklanan link ile ilişkili içeriğin saklandığı katman görüntülenir. Sitenin statik içeriği, üst üste onyediy farklı katmana yerleştirilmiştir.

Bu tekniğin amacı, sunucuya statik içerikli sitelerin statik sayfalarını görüntülemek için tekrar tekrar ziyaretçi tarafından istem gönderilmesini engelleyerek, sunucuyu statik içerikli sayfaların transmisyonu için daha az meşgul etmek ve ASP.NET, PHP gibi sunucu tarafı çalışarak dinamik içerikli sayfalar üretilmesini sağlayan sistemlere daha fazla zaman ayırmasını sağlamaktır. Şekil 4.13'de bölüm Web sitesinin giriş sayfası gösterilmektedir.



Şekil 4.13. Bölüm Web Sitesi Ana Sayfası

Bu tekniğin uygulamasındaki en önemli nokta, transmisyonu yapılacak olan statik içeriğin büyüklüğü ve bağlantı hızıdır. Statik içerik boyutu büyüdükçe transmisyonu gerçekleştirilen veri artacağından statik de olsa site içeriğinin çoğunluğunun ya da hepsinin bir istemde gönderilmesi özellikle çevirmeli ağ bağlantısı olan ziyaretçiler için uzunca bir işkenceye dönüşebilir.

Sonuç olarak bu teknik statik içeriğin toplam boyutu küçük olan statik içerikli siteler için uygulanabilir bir tekniktir. Küçük boyutu sağlayabilecek olan verinin başında da metin veri gelmektedir.

TARTIŞMA

İnternet'in zamanla giderek daha büyük bir "bilgi çöplüğü" halini almasının önüne "portal" adı verilen ve bir servis ve siteler yığını olarak açıklanan yapılarla geçilebilir. Portalların yapıları ve temel işlevi olan arama, bilgi ve belgeye ulaşımın daha efektif şekilde gerçekleştirilebilmesini sağlar. Portallar siteler yığnında, sitelerin hem kendi içinde hem de birbirleri ile ilişkisel bir düzende farklı türde belge ve bilgi barındıracak şekilde saklanmalarını ve kullanıcıların/ziyaretçilerin de bu bilgi ve belgelere geçişlerini düzenleyerek izin verir.

Belli başlı servislere sahip bir portal geliştirmek ciddi bir inceleme ve çözümleme gerektirmektedir. Kullanıcı ihtiyaçları belirlenmeli ve belirlenen ihtiyaçlara yönelik çözümler mevcut portal çözümleri ve uygulama geliştirme araçları kullanılarak gerçekleştirilmelidir. Fakat bunun kadar önemli bir başka nokta geliştirilen portalın kullanılması, buna bağlı olarak mevcut servislerin iyileştirilmesi ve servis çeşitliliğinin artırılmasıdır.

Bu çalışmada; Akdeniz Üniversitesi Tıp Fakültesi Biyoistatistik ve Tıp Bilişimi Anabilim Dalı'na yönelik bir belge yönetimi servisi; Akdeniz Üniversitesi Tıp Fakültesi öğrencilerine yönelik tıbbi terimler sözlüğü ve akademik yıl ders takvimi servislerinden oluşan bir portal geliştirilmiştir. Bu portalın servislerinin ve servis sayısının, kullanıcı davranışları ve istekleri göz önünde bulundurularak iyileştirilmesi ve artırılması planlanmalıdır.

5.1. Servislerin Genişletilmesi ve İyileştirilmesi

Bir tematik portalın verdiği servislerin iyileştirilmesi ve yeni servisler eklenebilmesi için ziyaretçi sayısı, ziyaret sayısı, servis işlem sayısı ve ziyaretçi memnuniyeti ölçümleri yapılmalıdır. Bunun yanı sıra, portalın verdiği her bir servis ile alakalı olarak ziyaretçilerin servislerle ilgili yaşadıkları tecrübelerini paylaşmalarını sağlayan, böylelikle servislerin ve portalın geliştirilmesine katkıda bulunmuş olacakları mesajını içeren bir geri bildirim servisi de portala eklenmelidir.

Fakat kullanıcılardan gelecek olan geri bildirimlerin yanında özellikle bazı istatistiki çalışmalarla servislerin kullanımı üzerine çözümlenmeler yapılmalıdır. Bu amaçla başlangıçta, her bir servisten belli zaman dilimlerinde yararlanan anlık toplam kullanıcı sayısı, bir servisi düzenli şekilde kullanan kullanıcı sayısı ve servislerin kullanıcılar tarafından kullanım süreleri gibi bazı değişkenler ölçümlenmelidir. Bu verilere ulaşmak, kullanılan portal sunucusunun ve Web sunucusun tuttuğu logların incelenmesi ile mümkündür. Ölçümlerden elde edilen veriler üzerinde yapılacak istatistiki çözümlenmelerle birlikte kullanıcı/ziyaretçilerden alınan geri bildirimler, portal içeriği ve servislerinin hangi yönde geliştirilmesi gerektiği konusunda karar verilmesine temel hazırlayacaktır. Elbette portalın ve servislerinin kullanıcılar tarafından benimsenip benimsenmediği de böylelikle değerlendirilebilir.

5.2. Tıbbi Terimler Sözlüğü İçin Servis Bilgisi Kalitesi

Tıbbi terimler sözlüğü servisi, öğrencilerin eğitim hayatları boyunca, ders sırasında ya da çalışırken karşılaştıkları tıbbi terimlerin anlamlarını kolayca arayıp bulmalarını sağlamak için geliştirilmiştir. Fakat eğitim boyutu itibarı ile veritabanında yer alacak terimlerin anlamları ehil kişilerce ve genel kabul görmüş ifadeler kullanılarak girilmelidir. Bu, servisin kalitesini belirleyici, önemli bir unsurdur. Bir diğer ifade ile servisin zayıf noktasıdır. Zira tıbbi terimler için verilen anlamsal bilginin doğruluğu, geçerliliği, tutarlılığı sağlanmalıdır ki eğitim amaçlı bu serviste sağlanan bilgi kaliteli olabilsin.

Bu amaçla, Tıp Fakültesi'ndeki her bir anabilim dalı içinde kişi ya da kişiler seçilerek o anabilim dalına giren terimlerin anlamlarının veritabanına girilmesi için görevlendirilebilir. Anlamlar, Dorland's Medical Dictionary gibi genel kabul kazanmış kaynaklardan yararlanılarak veritabanına eklenebilir.

Bunun yanında Tıbbi terimler sözlüğü servisi bazı iyileştirmelerle daha nitelikli hale getirilebilir. Söz konusu iyileştirmeler arasında tıbbi terimlerin İngilizce yazılışlarının da sözlüğe eklenerek terimlerin bu dildeki yazılışları ile arama sağlamak sayılabilir. Ayrıca, tıbbi terimlerin kısaltmaları ve terimler arasındaki eşanlamlılık servise dahil edilebilir. Örneğin; arama kriterleri arasına "Eşanlamlı terim" kriteri eklenebilir ve/veya anlam arama sonuçlarında aranan terimin eşanlamlı olduğu terime de yer verilebilir.

Tıbbi Terimler Sözlüğü servisi için bir diğer iyileştirme, servis ürününü; terimlerin anlamlarının çokluortam formatında veri ile desteklenmesini sağlayarak servisi saf metin-tabanlı olmaktan çıkarmaktır. Böylelikle servis, kullanıcıya terimin anlamı ile alakalı daha yüksek kapasitede bilgi sağlayabilecek şekilde sunulabilir.

5.3. Servis/Yazılım Mimarisi Seçimi

Tıbbi terimler sözlüğü ve akademik ders takvimi servislerinin geliştirilme mimarileri için 3-katmanlı mimari benimsenmiştir. 3-katmanlı mimarinin 2-katmanlı mimariye tercih edilmesinin ardındaki temel neden daha gelişmiş bir ölçeklenebilirlik sağlayabilmesidir. Başlangıçta küçük bir hedef kitleye sahip üç servisli bir portalın servisleri için bu mimarinin benimsenmesi mantıklı görünmese de portala yeni servisler eklenip belli seviyede bir servis zenginliğine ulaşıldığında mimarinin ölçeklenebilirlik yeteneğinin aslında bir gereklilik olduğu anlaşılacaktır. Zira servis sayısında artış, servislerden yararlanan kullanıcı/ziyaretçi sayısında da artış anlamına gelmektedir. Bu nedenle ölçeklenebilirlik servis sayısı yüksek bir portal için aranması gereken bir özelliktir.

Servislerin veri yönetimi katmanı için Microsoft'un Access veritabanı yönetim servisinin tercih edilmesi, servislerin gerektirdiği veritabanı ölçeğinin ya da diğer bir ifade ile veritabanı büyüklüğünün, küçük/orta ölçekli veritabanı sınıfına girmesidir.

Mimarinin sunum katmanı olarak adlandırılan en üst katmanın gerçekleştirimi ise Web Forms adı verilen yine bir Microsoft teknolojisi ile

yapılmıştır. Web Forms istemci yani tarayıcı tarafından görüntülenebilen fakat sunucu tarafında yaratılan geniş bir form ögesi yelpazesi sağlamaktadır. Fakat daha da önemlisi bu bileşenlerin sunucu tarafında yönetimini sağlayarak işlevselliğini genişletmiş olmasıdır. Web Forms yerine kullanılabilir bir başka teknoloji HTML formlarıdır. Fakat HTML formları, Web Forms'un sağladığı bu esnek işlevselliği ve form bileşeni zenginliğini geliştiricilere tanıyamadıklarından servislerin geliştirilmesinde tercih edilmemişlerdir.

5.4. Belge Yönetimi Servisi İçin Portal Çözümü Seçimi

Belge yönetimi servisinin tasarımı için benimsenen Sharepoint Portal Server'in seçilmesinde ise iki nokta göz önünde bulundurulmuştur. Öncelikle belge-merkezli işlev ve kolaylık zenginliği, Sharepoint'in bölümsel belge yönetimi yanını diğer portal çözümlerine nispeten kuvvetlendirmektedir. Bunun yanında her tip dosya için gelişmiş bir indeksleme ve arama fasilitasi ile birlikte kişisel alan sağlamaktadır. Sharepoint bu işlev ve özellikleri birarada bulundurduğundan geliştirilen portal için uygun çözüm olarak düşünülmüştür. Diğer çözümlerde bu gibi özellikler modülerite adına eklentiler şeklinde geliştirilmiştir. Yine de alternatif portal çözümleri de belge yönetim servisi için tercih edilebilir.

5.5. Tıbbi Terimler Sözlüğü Servisinin Veritabanı Performansı

Tıbbi terimler sözlüğü servisi veritabanı tıbbi terimlerin girilmesi ile onbinlerle ifade edilir sayıda tıbbi terim ve bunlara ait anlam saklayacaktır. Bu açıdan değerlendirildiğinde veritabanı boyutunun büyümesi ile servis performansının bundan nasıl etkileneceği üzerine fikir vermesi açısından bazı ölçümler yapılmıştır. Bu ölçümlerde, terim aram isteğinin sunucuya ulaştığı an ile veritabanı işlemleri'nin tamamlandığı an arasında geçen veritabanı işlem süresi hesaplanmıştır. Farklı boyutta (terim sayısı) veritabanlarında, sabit boyutta arama dizeleri ile "Girilen Dizeye Eşit Terimler" arama kriteri kullanılarak yapılan ölçümler sonucu minimum, maksimum, ortalama veritabanı işlem sürelerinin yanında standard sapma da hesaplanmıştır.

Sonuç olarak, tıbbi terimler sözlüğü servisinin gerektirdiği veritabanı sorgulama işlem süresinin toplam servis süresine etkisi veritabanının boyutuna bağlı olarak artsa da yapılan ölçümler bu etkinin servis toplam süresinin yanında (istem tarayıcıda yaratılması, sunucuya ağ üzerinden gönderilmesi, veritabanı sorgulama işlemleri, istem yanıtının yaratılması ve yanıtın tarayıcıya ağ üzerinden gönderilmesi) kabul edilebilir olduğu anlaşılmıştır. Diğer bir ifade ile veritabanı sorgulama işlemleri veritabanı boyutu artsa da servis süresinde ciddi bir etki yaratmayacaktır. Kullanıcılar açısından, veritabanı boyutu büyüdükçe artan sorgulama süresi bir servis gecikmesine yol açmamaktadır.

SONUÇLAR

Bu çalışmada geliştirilen medikal enformasyon portalı, gerek idari, gerekse akademik ve eğitsel anlamda bir başlangıçtır. Sağladığı servisler itibarı ile oldukça basit fakat yararlı görünen bu portal, gelişime açıktır. Kullanılmaya başlanmasının ardından özellikle var olan servislerin iyileştirilmesi ve yeni servislerin eklenmesi planlanmaktadır.

Tıbbi terimler sözlüğü için gerçekleştirilebilecek iyileştirmeler arasında tıbbi terimlerin İngilizce yazılışlarının da sözlüğe eklenerek terimlerin bu dildeki yazılışları ile arama sağlamak sayılabilir. Bunun yanı sıra, tıbbi terimlerin kısaltmaları ve aralarındaki eşanlamlılık düşünülerek servisin yetenekleri geliştirilebilir. Servis, tıbbi terimlerin eşanlamlılık temelinde aranması kriterini sağlayabilir; hatta anlam arama sonuçlarında sadece anlamsal açıklamaya değil terimin eşanlamlı olduğu terime de yer verebilir.

Serviste bir diğer iyileştirme, terimlerin anlamlarının grafik, şekil, resim, hatta düşük kalite video gibi çokluortam formatında veri ile desteklenmesidir. Böylelikle servis ürünü, sadece düz metin olmaktan çıkarak kullanıcıya terimin anlamı ile alakalı daha yüksek kapasitede bilgi sağlayabilecek şekilde sunulabilir. Bu servis, her ne kadar metin tabanlı veri ile de olsa özellikle öğrenciler için iyi bir kaynak olabilir. Bu özelliği ile servis, iyi bir eğitim aracı haline alabilir.

Bu servisin kaliteli bilgi sunabilmesi için terimler ve anlamları, veritabanına, ehil kişilerce girilmeli ve genel kabul görmüş anlamlar kullanılmalıdır. Aksi durumda, amacına ters düşen sonuçlar yaratabilir, yanlış/yanlı/eksik anlamsal bilgi sağlayabilir. Bu amaçla, Dorland's Medical Dictionary(34) gibi genel kabul görmüş kaynaklardan yararlanılarak terimler ve anlamlar servis veritabanına eklenebilir.

Ayrıca, tıbbi terimler sözlüğü servisinin veritabanı işlemleri açısından performansı ölçülmüş ve sabitlenmiş alan büyüklükleri için ortalama işlem süreleri hesaplanmıştır. Bu ölçümler farklı terim sayısında veritabanları için yapılmış ölçümlerden elde edilen sonuca göre, kullanıcılar düşünüldüğünde, 20000 terimlik bir kapasitede ortalama 0.51 saniyelik gerekli işlem süresi kabul edilebilir.

Kullanıcı kitlesi öğrenciler olarak planan bir diğer servis ise akademik yıl takvimi servisedir. Bu servis her akademik yılın başında belirlenen takvime öğrencilerin elektronik ortamdan ulaşabilmesini sağlayarak seçtikleri bir güne ait ders programını öğrenmelerini sağlar. Bu servis, daha çok portalın öğrenciler tarafından kullanımını artırmaya yöneliktir. Servisin kullanımı, geliştirilen arayüz sayesinde oldukça kolaydır. Bu servisin ders saatleri ve adlarının yanında dersin yapılacağı bina ve oda kodunu da sağlaması bir servis iyileştirmesi olarak sayılabilir.

Belge yönetim servisi, Biyoistatistik ve Tıp Bilişimi Anabilim Dalı'nda belge işlemlerinin elektronik ortamda yapılması için geliştirilmiştir. Bu servisin tasarlanması için Microsoft'un portal çözümü olan Sharepoint Portal Server benimsenmiştir.

KAYNAKLAR

- 1- <http://foldoc.org/>, Free Online Dictionary of Computing, 25.06.2006
- 2- <http://www.walthowe.com/navnet/history.html>, A Brief History of the Internet, 21.12.2005
- 3- Denning PJ: The ARPANET After Twenty Years. American Scientist 77: 530-535, 1989.
- 4- Bhushan A, Progran K, Tomlinson R, White J: Request For Comments 561, Standardizing Network Mail Headers, 1973
- 5- Crocker DH, Vittal JJ, Progran KT, Henderson DA Jr II: Request For Comments 733, Standard For The Format Of ARPA Network Messages (1), 1977
- 6- Cerf V, Kahn R: A Protocol for Packet Network Intercommunication. IEEE Transactions on Communications COM-22: 637-648, 1974
- 7- Request For Comments 793, Transmission Control Protocol, DARPA İnternet Program Protocol Specification, 1981
- 8- Emtage A, Deutsch P: "archie: An Electronic Directory Service For The Internet". Winter Usenix Conference Proceedings: 93-110, 1992
- 9- <http://www.w3.org/History/1989/proposal.html>, World Wide Web Consortium, The Original Proposal of the WWW, 25.06.2006.
- 10- <http://anzheug.anu.edu.au/blog/wpcontent/uploads/2006/04/presentations/pres4/sigportal/sld005.htm>, Types of Portals, 11.05.2006.
- 11- <http://www.ntvmsnbc.com>, Yatay Türde Haber Portalı, 20.07.2006

- 12- <http://www.hpb.gov.sg>, Health Promotion Board Online, 12.04.2006
- 13- Vijaya K, Chan SP, Ho HPC, Lim YYL, Lim R: HPB Online: an electronic health education portal in Singapore. Singapore Medical Journal 47(1): 8-13, 2006.
- 14- <http://www.patientgateway.org>, Patient Gateway, 13.04.2006.
- 15- Kittler AF, Carlson GL, Harris C, Lippincott M, Pizziferri L, Volk LA, Jagannath Y, Wald JS, Bates DW: Primary care physician attitudes towards using a secure Web-based portal designed to facilitate electronic communication with patients. Informatics in Primary Care 12: 129-138, 2004.
- 16- <http://mywelch.welch.jhni.edu>, John Hopkins Üniversitesi Tıp Fak. Portalı, 11.07.2006.
- 17- Wang X, Zhang D: Managing and Sharing Knowledge through Portal- The John Hopkins University Libraries Experience. 3rd China/US Conference on Libraries: Knowledge Management in the Digital Age, 2005.
- 18- <http://www.minastma.dk>, 15.04.2006.
- 19- Anhoj J, Nielsen L: Quantitative and Qualitative Usage Data of an Internet-Based Asthma Monitoring Tool. J Med Internet Res. 6(3): ,2003.
- 20- Microsoft Office Sharepoint Portal Server 2003. Customer Evaluation Guide, Microsoft, 2003.
- 21- Benefits of Microsoft Office Sharepoint Portal Server 2003. White Paper, Microsoft, 2004.
- 22- <http://msdn.microsoft.com>, Microsoft Developer Network, 19.05.2006.
- 23- Kosinska J, Slowikowski P: Technical Aspects of Portal

Technology Application for E-health Systems.

- 24- Young M, Adım Adım XML. Arkadaş Yayınevi, Ankara, 2001.
- 25- <http://www.w3.org> , SOAP, 15.06.2006
- 26- Korth HF, Silberschatz A: Database System Concepts. 2nd Edition, McGraw Hill, 1991.
- 27- Tanenbaum AS, Computer Networks. 2nd Edition, Prentice Hall, 1996.
- 28- Arifođlu A, Dođru A: Yazılım Mühendisliđi. 1.Baskı. SAS Bilişim Yayınları, Ankara, 2001.
- 29- http://www.sei.cmu.edu/str/descriptions/threetier_body.html, 2006, 19.04.2006.
- 30- Buyens J, Web Database Development Step by Step .NET Edition. 2nd Edition. Microsoft Press, 2002
- 31- Joubert M, Aymard S, Staccini P, Fieschi M, Designing and Implementing Medical Web Portals: spreading Educational and Research Materials on the İnternet. Proceedings, AMIA Symposium, 299-303, 2001.
- 32- http://www.motobit.com/tips/detpg_Perfdata/, Veritabanı Veri Sađlayıcıları-Motorları Benchmark Testi, 11.07.2006
- 33- Yarırmađan Ü, Veritabanı Sistemleri. Hacettepe Üniversitesi, Akademi Yayın Hizmetleri, Ankara, 2000.
- 34- Dorland's Medical Dictionary, Saunders, 30. Baskı, Haziran 2003.

EKLER

Tıbbi Terimler Sözlüğü Servisi, Terim Arama Kaynak Kodu

```
<%@Page language="vb" explicit="true" debug="true" %>
<%@Import Namespace="System.Data" %>
<%@Import Namespace="System.Data.OleDb" %>
<script runat="server">
```

Sub terimiVeritabanindaAra ()

```
Dim baglantici As OleDbConnection
Dim baglantiDizesi As String
Dim sorguYurutucu As OleDbCommand
Dim veriOkuyucu As OleDbDataReader
Dim sorgular(4) As String
Dim sorguKoku As String
Dim sorgu_2 As String
Dim bulunanTerimSayisi As Integer
Dim terimID As String
Dim aciklama As String
```

Randomize

```
baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & _
    server.mappath("medikalsozluk.mdb") & ";"
```

```
sorgu_2 = "SELECT terimin_anlami FROM ANLAMLAR WHERE terim_id="
sorguKoku = "SELECT * FROM Terimler WHERE terim"
sorgular (0) = sorguKoku & " LIKE '" & mk_1.text & "%'"
sorgular (1) = sorguKoku & " LIKE '%" & mk_1.text & ""'"
sorgular (2) = sorguKoku & " LIKE '%" & mk_1.text & "%'"
sorgular (3) = sorguKoku & "='" & mk_1.text & ""'"
```

```
l_1.text = ""
```

```
bulunanTerimSayisi = 0
```

```
Try
```

```
    baglantici = New OleDbConnection(baglantiDizesi)
    sorguYurutucu = New OleDbCommand( sorgular(lk_1.SelectedIndex) ,
```

baglantici)

```

baglantici.Open()
veriOkuyucu = sorguYurutucu.ExecuteReader()

Select Case lk_1.SelectedIndex
    Case 0 to 2
        While veriOkuyucu.Read ( )
            bulunanTerimSayisi += 1
            If bulunanTerimSayisi = 2 Then
                Exit While
            End If
        End While
        veriOkuyucu.Close()

        veriOkuyucu = sorguYurutucu.ExecuteReader()
        If bulunanTerimSayisi >= 1 Then
            l_1.text = "<b>Aradığınız Dize ve Kriter ile
Uyuşan Terimler Aşağıdadır." & _
                "<br>Anlamını Görmek İsteddiğiniz Terimi
Arama Kutusuna Yazdıktan Sonra" & _
                "'Dizeye Eşit Olan Terimleri' Arama Kriterini
Seçip Aramayı Tekrarlayınız.</b><br>"

            While veriOkuyucu.Read ( )
                l_1.text &= "<br>" &
                veriOkuyucu.Item("terim")
            End While

            ElseIf bulunanTerimSayisi = 0 Then
                l_1.text="Veritabanımızda Verilen Dize ve
Arama Kriteriyle Uyuşan Terim Bulunamadı..."
            End If

        Case 3

            l_1.text = "Veritabanımızda Verilen Dize ve
Arama Kriteriyle Uyuşan Terim Bulunamadı..."

            If veriOkuyucu.Read ( ) Then
                sorgu_2 &= veriOkuyucu.Item
                ("terim_id")
            End If
        End Case
    End Select

```

```

OleDbCommand ( sorgu_2 , baglantici )
sorguYurutucu.ExecuteReader ( )
& "</i>: <b>" & veriOkuyucu.Item (0) & "</b>"

veriOkuyucu.Close ( )
sorguYurutucu.Dispose ( )
sorguYurutucu = New
veriOkuyucu =
If veriOkuyucu.Read ( )
    l_1.text = "<i>" & mk_1.text
End If
End If

```

End Select

```

veriOkuyucu.Close()
sorguYurutucu.Dispose()

```

Catch exc As OleDbException

```

l_1.text = "Beklenmeyen Özel bir programatik durum ortaya çıktı." & _
    "Sözlüğü kullanamadığımız için üzgünüz." & _
    "Durum Özeti aşağıdaki gibidir: " & exc.Message

```

Finally

```

baglantici.Close()

```

End Try

End Sub

Sub Page_Load (sender As Object , e As EventArgs)

```

Dim baglantici As OleDbConnection
Dim baglantiDizesi As String
Dim sorguYurutucu As OleDbCommand
Dim veriOkuyucu As OleDbDataReader
Dim sorgu As String

```

```

Dim baslangic, bitis As DateTime
Dim i As Integer

baslangic = now ( )
If Page.IsPostBack Then
    terimiVeritabanindaAra ( )
End If
bitis = now ( )

olcum.Text = "Baslangic<br>saniye: " & baslangic.Second & "   milisaniye: " &
baslangic.Millisecond & "<br><br>Bitis<br>saniye: " & bitis.Second & "   milisaniye: " &
bitis.Millisecond

baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & _
    server.mappath("olcum.mdb") & ";"

sorgu = "INSERT INTO olcumler ( baslangicSaniye, baslangicMilisaniye,
bitisSaniye, bitisMilisaniye, aramaDizesi ) VALUES (" & baslangic.Second & "," &
baslangic.Millisecond & "," & bitis.Second & "," & bitis.Millisecond & "," & mk_1.text & ")"

Try

    baglantici = New OleDbConnection(baglantiDizesi)
    sorguYurutucu = New OleDbCommand( sorgu , baglantici)
    baglantici.Open()

    If sorguYurutucu.ExecuteNonQuery ( ) > 0 Then
        sorguYurutucu.Dispose ( )
        olcum.Text &= " ölçümler başarı ile eklendi"
    Else
        olcum.Text = "ölçümler eklenemedi!"
    End If

Catch exc As OleDbException

    l_1.text = "Beklenmeyen Özel bir programatik durum ortaya çıktı." & _
        "Sözlüğü kullanamadığınız için üzgünüz." & _
        "Durum Özeti aşağıdaki gibidir: " & exc.Message

Finally

```


baglantici.Close()

End Try

End Sub

</script>

<html>

<head>

<title>Medikal Sözlük Deneme I</title>

</head>

<body bgcolor="aabbcc">

<h1>Tıbbi Terimler Sözlüğü v0.1</h1>

<h2>Arama Adımları:</h2>

1. Arama dizesini aşağıdaki metin kutusuna yazın:

<form runat="server">

<p><asp:Textbox Text="Terimi Buraya Yazın" id="mk_1" runat="server" Maxlength="249"

Columns="25" /></p>

2. Uygun Arama Kriterini Aşağıdaki Liste Kutusundan Seçin:

<p><asp:DropDownList id="lk_1" runat="server" AutoPostBack="false">

<asp:ListItem value="1">Dize ile Başlayan Terimleri</asp:ListItem>

<asp:ListItem value="2">Dize ile Biten Terimleri</asp:ListItem>

<asp:ListItem value="3">Dizeyi içinde Bulunduran Terimleri</asp:ListItem>

<asp:ListItem value="4">Dizeye Eşit Olan Terimleri</asp:ListItem>

</asp:DropDownList>

3. ARA Düğmesine Tıklayın.

<p><asp:Button Text="ARA" id="d_1" runat="server" /></p>

<p><asp:Label Text="" id="l_1" runat="server"/></p>

<p><asp:Label Text="" id="olcum" runat="server"/></p>

</form>

</body>

</html>

Tıbbi Terimler Sözlüğü Servisi, Terim Ekleme Kaynak Kodu.

```

<%@Page language="vb" explicit="true" debug="true" %>
<%@Import Namespace="System.Data" %>
<%@Import Namespace="System.Data.OleDb" %>
<script runat="server">

```

```

Sub Page_Load ( sender As Object , e As EventArgs )
    If Page.IsPostBack Then
        terimiVeritabanınaEkle ( )
    End If
End Sub

```

```

Sub terimiVeritabanınaEkle ( )

```

```

    Dim baglantici As OleDbConnection
    Dim sorguYurutucu As OleDbCommand
    Dim veriOkuyucu As OleDbDataReader
    Dim veritabanıİslemiYurutucu As OleDbTransaction
    Dim baglantiDizesi As String
    Dim sorgu As String
    Dim yeniTerimID As Integer
    Dim eklemeSorgusuBasariBayragi As Integer

```

```

    baglantiDizesi = "Provider=Microsoft.Jet.OleDb.4.0;" & _
        "Data Source=" & _
        server.mappath("medikalsozluk_1000.mdb") & ";"

```

```

Try

```

```

    baglantici = New OleDbConnection ( baglantiDizesi )
    baglantici.open ( )
    veritabanıİslemiYurutucu = baglantici.BeginTransaction ( )

```

```

    sorgu = "INSERT INTO Terimler (" & _
        "terim" & _
        ") VALUES (" & mk_1.text & ")"

```

```

    sorguYurutucu = New OleDbCommand ( sorgu , baglantici )
    sorguYurutucu.Transaction = veritabanıİslemiYurutucu
    eklemeSorgusuBasariBayragi = sorguYurutucu.ExecuteNonQuery ( )

```

```

    If eklemeSorgusuBasariBayragi > 0 Then

```

```

sorguYurutucu.Dispose ( )
sorgu = "SELECT terim_id FROM Terimler WHERE terim=" &
mk_1.text & ""

sorguYurutucu = New OleDbCommand ( sorgu , baglantici )
sorguYurutucu.Transaction = veritabaniIslemiYurutucu
veriOkuyucu = sorguYurutucu.ExecuteReader ( )
If veriOkuyucu.Read Then
    yeniTerimID = veriOkuyucu.Item ( 0 )
    veriOkuyucu.Close ( )
    sorguYurutucu.Dispose ( )
    sorgu = "INSERT INTO Anlamlar (" & _
        "terim_id ," & _
        "termin_anlami ," & _
        "anlami_ekleyen_kullanici , "
& _
        "anlamin_eklenme_tarihi , "
& _
        "anlamin_son_guncellenme_tarihi" & _
        ") VALUES (" & yeniTerimID & "," & _
        mk_2.text & "," & _
        mk_3.text & "," & _
        Now ( ) & "," & _
        Now ( ) & ")"
sorguYurutucu = New OleDbCommand ( sorgu ,
baglantici )
sorguYurutucu.Transaction = veritabaniIslemiYurutucu
eklemeSorgusuBasariBayragi =
sorguYurutucu.ExecuteNonQuery ( )
If eklemeSorgusuBasariBayragi > 0 Then
    l_1.text = "Terim veritabanına eklendi. Katkınız
için teşekkür ederiz."

    veritabaniIslemiYurutucu.Commit ( )
    veriOkuyucu.Close ( )
    sorguYurutucu.Dispose ( )
    baglantici.Close ( )
    mk_1.text = ""
    mk_2.text = ""
    mk_3.text = ""

```

```
Else
    l_1.text = "Hata 1 Beklenmedik bir hata oluştu.
Lütfen Sunucu Yöneticisine bu durumu bildirin."
    ' veritabaniIslemiYurutucu.Rollback ( )
    <%-- Burada Terimler tablosuna yapılan ekleme
silinmeli --%>
End If
```

```
Else
    l_1.text = "Beklenmedik bir programmatik hata oluştu.
Lütfen Sunucu Yöneticisine bu durumu bildirin."
    <%-- Burada Terimler tablosuna yapılan ekleme
silinmeli --%>
End If
```

```
End If
```

```
Catch exc As OleDbException
    mk_1.text = sorgu
    l_1.text = "Özel bir programmatik durum ile karşılaşıldığından işleminiz
tamamlanamayacak. Durum mesajı: " & exc.Message
    veritabaniIslemiYurutucu.Rollback ( )
    <%--veriOkuyucu.Close ( )
    sorguYurutucu.Dispose ( ) --%>
    bağlantici.Close ( )
Finally
End Try
```

```
End Sub
```

```
</script>
```

```
<html>
```

```
<head>
```

```
<title>Medikal Sözlüğe Yeni Terim Ekle...</title>
```

```
</head>
```

```
<body bgcolor="aabbcc">
```

```
<font face="Verdana" size="2"><b>Lütfen Eklenecek Terimle İlgili Alanları Doldurun: </b>
```


Dim donemSayisiSorgusu As String

Page.Validate ()

If Page.IsPostBack Then

 intPostBackCnt = ViewState ("PostBackCnt")

 intPostBackCnt = intPostBackCnt+1

 ViewState ("PostBackCnt") = intPostBackCnt

 tarihEtiketi.text = "If Then calisti"

Else

 takvim.SelectedDate = now ()

 yilMetinKutusu.text = takvim.SelectedDate.Year

 ayListesi.SelectedIndex = takvim.SelectedDate.Month - 1

 baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
 "Data Source=" & _
 server.mappath("derstakvimi.mdb") & ";"

 donemSayisiSorgusu = "SELECT donem_no FROM Donemler"

 donemSayisi = 0

 postBackSayisi.text=""

 Try

 baglantici = New OleDbConnection(baglantiDizesi)

 sorguYurutucu = New OleDbCommand(donemSayisiSorgusu ,

baglantici)

 baglantici.Open()

 veriOkuyucu = sorguYurutucu.ExecuteReader()

 While veriOkuyucu.Read ()

 donemSayisi += 1

 End While

 veriOkuyucu.Close ()

 sorguYurutucu.Dispose ()

 Catch exc As OleDbException

 yilMetinKutusu.text = donemSayisiSorgusu

```

        yilMetinKutusu.Text = exc.Message

    Finally

        baglantici.Close ()

    End Try

    donemListesi.items.Clear()
    For i=1 to donemSayisi
        donemListesi.items.insert ( i-1 , new listItem( i , i ) )
    Next

End If

End Sub

Sub donemListesindeDonemSecildi ( sender As Object , e As EventArgs )

    Dim baglantici As OleDbConnection
    Dim baglantiDizesi As String
    Dim sorguYurutucu As OleDbCommand
    Dim veriOkuyucu As OleDbDataReader
    Dim kurulSayisiSorgusu As String
    Dim kurulSayisi As Integer
    Dim i As Integer

    baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & _
        server.MapPath("derstakvimi.mdb") & ";"
    kurulSayisiSorgusu = "SELECT kurul_sayisi FROM Donemler WHERE
donem_no="

    Try

        baglantici = New OleDbConnection(baglantiDizesi)
        kurulSayisiSorgusu += donemListesi.SelectedItem.Value
        sorguYurutucu = New OleDbCommand( kurulSayisiSorgusu , baglantici)
        baglantici.Open()

```

```

        veriOkuyucu = sorguYurutucu.ExecuteReader()
        tarihEtiketi.Text = "Deneme"
        If veriOkuyucu.Read ( ) Then
            kurulSayisi = veriOkuyucu.Item ( 0 )
        Else
            tarihEtiketi.Text = "Seçtiğiniz Döneme ait kayıt bulunamadı. Bu
durumu sistem yöneticisine e-posta ile bildirebilirsiniz."
        End If

        veriOkuyucu.Close ( )
        sorguYurutucu.Dispose ( )

    Catch exc As OleDbException

        yilMetinKutusu.text = kurulSayisiSorgusu
        yilMetinKutusu.Text = exc.Message

    Finally

        baglantici.Close ( )

    End Try

    kurulListesi.items.Clear()
    For i=1 to kurulSayisi
        kurulListesi.items.insert ( i-1 , new listItem( i , i ) )
    Next

End Sub

Sub ayListesindeAyDegisti ( sender As Object , e As EventArgs )

    takvim.VisibleDate = New DateTime ( yilMetinKutusu.text ,
ayListesi.SelectedIndex+1 , 1 )

End Sub

Sub metinKutusundaYilDegisti ( sender As Object , e As EventArgs )

```



```
        If Page.IsValid Then
            takvim.VisibleDate = New DateTime ( yılMetinKutusu.text ,
ayListesi.SelectedIndex+1 , 1 )
        End If
```

```
End Sub
```

```
Sub takvimdeAyDegisti ( sender As Object , e As MonthChangedEventArgs )
```

```
    yılMetinKutusu.text = e.NewDate.Year
    yılDegeriKontrolcusu.Validate ( )
    ayListesi.SelectedIndex = e.NewDate.Month - 1
```

```
End Sub
```

```
Sub takvimdeGunSecildi ( sender As Object , e As EventArgs )
```

```
    Dim baglantici As OleDbConnection
    Dim baglantiDizesi As String
    Dim sorguYurutucu, sorguYurutucu_2 As OleDbCommand
    Dim veriOkuyucu, veriOkuyucu_2 As OleDbDataReader
    Dim kurulKimligiSorgusu As String
    Dim programSorgusu As String
    Dim programSorgusuKuyruk As String
    Dim kurulKimligi As Integer
    Dim tiklananTarih As DateTime
    Dim i As Integer
    Dim tz As DateTime
```

```
    baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & _
        server.mappath("derstakvimi.mdb") & ";"
```

```
    kurulKimligiSorgusu = "SELECT kurul_id FROM Kurullar WHERE donem_no="
& ( donemListesi.SelectedItem.Value ) & " AND kurul_no=" & kurulListesi.SelectedItem.Value
    programSorgusu = "SELECT program FROM Dersgunu WHERE tarih="
    programSorgusuKuyruk = " AND kurul_id="
```

```
Try
```

```

baglantici = New OleDbConnection(baglantiDizesi)
sorguYurutucu = New OleDbCommand( kurulKimligiSorgusu , baglantici)
baglantici.Open ( )
veriOkuyucu = sorguYurutucu.ExecuteReader( )

If veriOkuyucu.Read ( ) Then
    kurulKimligi = veriOkuyucu.Item ( 0 )
    tz = now ( )
    postBackSayisi.Text = "Bulunan kurul kimligi " & kurulKimligi
& " saniye: " & tz.Second & " milisaniye: " & tz.Millisecond

    programSorgusu &= "#" & takvim.SelectedDate.Day ( ) & "/" &
takvim.SelectedDate.Month ( ) & "/" & takvim.SelectedDate.Year ( ) & "#"
    programSorgusu &= programSorgusuKuyruk & kurulKimligi
    veriOkuyucu.Close ( )
    sorguYurutucu.Dispose ( )

    sorguYurutucu = New OleDbCommand( programSorgusu ,
baglantici)

    veriOkuyucu = sorguYurutucu.ExecuteReader ( )

    If veriOkuyucu.Read( ) Then
        tarihEtiketi.Text = "<b>" & veriOkuyucu.Item ( 0 ) &
"</b>"

    Else
        tarihEtiketi.Text = "<b>Bu tarihe ait veri
bulunamadı</b>" & " sorgu: " & programSorgusu
    End If

    Else
        yılMetinKutusu.Text = "Bulunamadı " &
kurulListesi.SelectedItem.Value
        tarihEtiketi.Text = kurulKimligiSorgusu
    End If

Catch exc As OleDbException

    tarihEtiketi.Text = exc.Message & " " & programSorgusu

```

Finally

baglantici.Close ()

End Try

End Sub

</script>

<html>

<head>

<title>Akademik Yıl Program Servisi</title>

</head>

<body bgcolor="aabbcc">

<form runat="server">

<p>Adım 1: Dönem Seçiniz

<asp:DropDownList id="donemListesi" runat="server" AutoPostBack="True"

OnSelectedIndexChanged="donemListesindeDonemSecildi">

<asp:ListItem value="1">Dönem I</asp:ListItem>

<asp:ListItem value="2">Dönem II</asp:ListItem>

<asp:ListItem value="3">Dönem III</asp:ListItem>

<asp:ListItem value="4">Dönem IV</asp:ListItem>

</asp:DropDownList>

</p>

<p>Adım 2: Kurul Seçiniz

<asp:DropDownList id="kurulListesi" runat="server" AutoPostBack="False">

<asp:ListItem value="1">Önce Dönem Seçiniz</asp:ListItem>

</asp:DropDownList>

</p>

<p>

<asp:DropDownList id="ayListesi" runat="server" AutoPostBack="True"

OnSelectedIndexChanged="ayListesindeAyDegisti">

```

        <asp:ListItem value="1">Ocak</asp:ListItem>
        <asp:ListItem value="2">Şubat</asp:ListItem>
        <asp:ListItem value="3">Mart</asp:ListItem>
        <asp:ListItem value="4">Nisan</asp:ListItem>
        <asp:ListItem value="5">Mayıs</asp:ListItem>
        <asp:ListItem value="6">Haziran</asp:ListItem>
        <asp:ListItem value="7">Temmuz</asp:ListItem>
        <asp:ListItem value="8">Ağustos</asp:ListItem>
        <asp:ListItem value="9">Eylül</asp:ListItem>
        <asp:ListItem value="10">Ekim</asp:ListItem>
        <asp:ListItem value="11">Kasım</asp:ListItem>
        <asp:ListItem value="12">Aralık</asp:ListItem>
    </asp:DropDownList>

    <asp:TextBox id="yilMetinKutusu" runat="server" columns="6" MaxLength="4"
    AutoPostBack="True" OnTextChanged="metinKutusundaYilDegisti" />
    <asp:RangeValidator id="yilDegeriKontrolcusu" runat="server"
        ControlToValidate="yilMetinKutusu"
        Type="integer"
        MinimumValue="2005"
        MaximumValue="2007"
        Display="Dynamic"
        ErrorMessage="Girdiğiniz Yıl Değeri İçin Ders Programı Görüntülenemiyor." />
</p>
<p>
    <asp:calendar id="takvim" runat="server" onVisibleMonthChanged="takvimdeAyDegisti"
    onSelectionChanged="takvimdeGunSecildi" />
</p>
<p>
    <asp:Literal id="tarihEtiketi" runat="server"/>
</p>
<p>
    <asp:Literal id="postBackSayisi" runat="server"/>
</p>
</form>
</body>
</html>

```

Akademik Yıl Takvimi Servisi Ders Programı Ekleme Kaynak Kodu

```

<%@Page language="vb" explicit="true" debug="true" %>
<%@Import Namespace="System.Data" %>
<%@Import Namespace="System.Data.OleDb" %>
<script runat="server">

```

```

Sub Page_Load ( sender As Object , e As EventArgs )

```

```

    Dim intPostBackCnt, donemSayisi, i As Integer

```

```

    Dim baglantici As OleDbConnection

```

```

    Dim baglantiDizesi As String

```

```

    Dim sorguYurutucu As OleDbCommand

```

```

    Dim veriOkuyucu As OleDbDataReader

```

```

    Dim donemSayisiSorgusu As String

```

```

Page.Validate ( )

```

```

If Page.IsPostBack Then

```

```

    intPostBackCnt = ViewState ( "PostBackCnt" )

```

```

    intPostBackCnt = intPostBackCnt+1

```

```

    ViewState ( "PostBackCnt" ) = intPostBackCnt

```

```

    <%--tarihEtiketi.text = "If Then calisti"--%>

```

```

Else

```

```

    takvim.SelectedDate = now ( )

```

```

    yilMetinKutusu.text = takvim.SelectedDate.Year

```

```

    ayListesi.SelectedIndex = takvim.SelectedDate.Month - 1

```

```

    baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _

```

```

        "Data Source=" & _

```

```

        server.mappath("derstakvimi.mdb") & ";"

```

```

    donemSayisiSorgusu = "SELECT donem_no FROM Donemler"

```

```

    donemSayisi = 0

```

```

Try

```

```

    baglantici = New OleDbConnection(baglantiDizesi)

```

```

    sorguYurutucu = New OleDbCommand( donemSayisiSorgusu ,

```

```

    baglantici)

```

```

        baglantici.Open()
        veriOkuyucu = sorguYurutucu.ExecuteReader()

        While veriOkuyucu.Read ( )
            donemSayisi += 1
        End While

        veriOkuyucu.Close ( )
        sorguYurutucu.Dispose ( )

    Catch exc As OleDbException

        tarihEtiketi.Text = "Seçili Döneme Ait Kurulları Sorgularken Bir
        istisnai durum ortaya çıktı. Sistem Mesajı: "
        tarihEtiketi.Text &= exc.Message

    Finally

        baglantici.Close ( )

    End Try

    donemListesi.items.Clear ( )
    donemListesi.items.insert ( 0 , new listItem ( "Bir Dönem Seçiniz" , 0 ) )
    For i=2 to donemSayisi+1
        donemListesi.items.insert ( i-1 , new listItem( i-1 , i-1 ) )
    Next

    End If
End Sub

Sub donemListesindeDonemSecildi ( sender As Object , e As EventArgs )

    Dim baglantici As OleDbConnection
    Dim baglantiDizesi As String
    Dim sorguYurutucu As OleDbCommand
    Dim veriOkuyucu As OleDbDataReader
    Dim kurulSayisiSorgusu As String
    Dim kurulSayisi As Integer

```

Dim i As Integer

```
baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
                "Data Source=" & _  
                server.mappath("derstakvimi.mdb") & ";"  
kurulSayisiSorgusu = "SELECT kurul_sayisi FROM Donemler WHERE  
donem_no="
```

Try

```
    baglantici = New OleDbConnection(baglantiDizesi)  
    kurulSayisiSorgusu += donemListesi.SelectedItem.Value  
    sorguYurutucu = New OleDbCommand( kurulSayisiSorgusu , baglantici)  
    baglantici.Open()  
    veriOkuyucu = sorguYurutucu.ExecuteReader()  
  
    If veriOkuyucu.Read ( ) Then  
        kurulSayisi = veriOkuyucu.Item ( 0 )  
    Else  
        tarihEtiketi.Text = "Seçtiğiniz Döneme ait kayıt bulunamadı. Bu  
durumu sistem yöneticisine e-posta ile bildirebilirsiniz."  
    End If
```

```
    veriOkuyucu.Close ( )  
    sorguYurutucu.Dispose ( )
```

Catch exc As OleDbException

```
    tarihEtiketi.Text = kurulSayisiSorgusu & " Seçili Döneme Ait Kurulları  
Sorgularken Bir istisnai durum ortaya çıktı. Sistem Mesajı: "  
    tarihEtiketi.Text &= exc.Message
```

Finally

```
    baglantici.Close ( )
```

End Try

```
kurulListesi.items.Clear( )
```

```

        kurulListesi.items.insert ( 0 , new listItem ( "Bir Kurul Seçiniz" , 0 ) )
    For i=2 to kurulSayisi+1
        kurulListesi.items.insert ( i-1 , new listItem( i-1 , i-1 ) )
    Next

    dersListesi.items.Clear ( )
    dersListesi.items.insert ( 0 , new listItem ( "Önce Kurul Seçiniz" , 0 ) )
    saatListesi.items.Clear ( )
    saatListesi.items.insert ( 0 , new listItem ( "Önce Kurul Seçiniz" , 0 ) )

End Sub

Sub kurulListesindekurulSecildi ( sender As Object , e As EventArgs )

    Dim baglantici As OleDbConnection
    Dim baglantiDizesi As String
    Dim sorguYurutucu As OleDbCommand
    Dim veriOkuyucu As OleDbDataReader
    Dim kurulIDSorgusu, dersAdlariSorgusu As String
    Dim dersAdlari(10) As String
    Dim kurulIDSorgusuKuyrugu As String
    Dim dersSaatleri() As String = {"08:30", "09:30", "10:30", "11:30", "12:30",
"13:30", "14:30", "15:30", "16:30" }
    Dim dersSayisi As Integer
    Dim i As Integer

    baglantiDizesi = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & _
        server.MapPath("derstakvimi.mdb") & ";"
    kurulIDSorgusu = "SELECT kurul_id FROM Kurullar WHERE donem_no="
    dersAdlariSorgusu = "SELECT dersin_adi FROM Dersler WHERE kurul_id="
    kurulIDSorgusuKuyrugu = " and kurul_no="
    Try

        baglantici = New OleDbConnection(baglantiDizesi)
        kurulIDSorgusu &= donemListesi.SelectedItem.Value
        kurulIDSorgusuKuyrugu &= kurulListesi.SelectedItem.Value
        kurulIDSorgusu &= kurulIDSorgusuKuyrugu
        sorguYurutucu = New OleDbCommand( kurulIDSorgusu , baglantici)

```



```

baglantici.Open()
veriOkuyucu = sorguYurutucu.ExecuteReader()

If veriOkuyucu.Read ( ) Then
    dersAdlariSorgusu &= veriOkuyucu.Item ( 0 )
    veriOkuyucu.Close ( )
    sorguYurutucu.Dispose ( )

    sorguYurutucu = New OleDbCommand( dersAdlariSorgusu ,
baglantici)

    veriOkuyucu = sorguYurutucu.ExecuteReader()

    dersSayisi = 0
    While veriOkuyucu.Read ( )
        dersAdlari(dersSayisi) = veriOkuyucu.Item( 0 )
        dersSayisi += 1
    End While

    If dersSayisi > 0 Then
        dersListesi.items.Clear ( )
        dersListesi.items.insert ( 0 , new listItem( "Bir Ders
Seçiniz" , 0 ) )
        dersListesi.items.insert ( 1 , new listItem( "Boş" , "Boş" )
)
        For i=2 to dersSayisi+1
            dersListesi.items.insert ( i , new listItem(
dersAdlari(i-2) , dersAdlari(i-2) ) )
        Next

        saatListesi.items.Clear ( )
        For i=1 to 9
            saatListesi.items.insert ( i-1 , new listItem(
dersSaatleri(i-1) , dersSaatleri(i-1) ) )
        Next
    Else
        saatListesi.items.Clear ( )
        saatListesi.items.insert ( 0 , new listItem ( "Bu Dönem-
Kurul İkilisine ait Ders Bulunamadı" , 0 ) )

```

```

                dersListesi.items.Clear ( )
                dersListesi.items.insert ( 0 , new listItem ( "Bu Dönem-
Kurul İkilisine ait Ders Bulunamadı" , 0 ) )
            End If
        Else
            tarihEtiketi.Text = "Sorgulanan Dönem-Kurul İkilisi ile Uyuşan
Bir KurulID Bulunamadı. Bu durumu Yöneticiye bildirebilirsiniz."
        End If

        veriOkuyucu.Close ( )
        sorguYurutucu.Dispose ( )

    Catch exc As OleDbException

        tarihEtiketi.Text = " Seçili Kurula Ait Dersleri Sorgularken Bir istisnai
durum ortaya çıktı. Sistem Mesajı: "
        tarihEtiketi.Text &= exc.Message
    Finally

        baglantici.Close ( )

    End Try

End Sub

Sub ayListesindeAyDegisti ( sender As Object , e As EventArgs )

    takvim.VisibleDate = New DateTime ( yilMetinKutusu.text ,
ayListesi.SelectedIndex+1 , 1 )

End Sub

Sub takvimdeGunSecildi ( sender As Object , e As EventArgs )

End Sub

Sub metinKutusundaYilDegisti ( sender As Object , e As EventArgs )

```

```

        If Page.IsValid Then
            takvim.VisibleDate = New DateTime ( yılMetinKutusu.text ,
ayListesi.SelectedIndex+1 , 1 )
        End If

```

```

End Sub

```

```

Sub takvimdeAyDegisti ( sender As Object , e As MonthChangedEventArgs )

```

```

    yılMetinKutusu.text = e.NewDate.Year
    yılDegeriKontrolcusu.Validate ( )
    ayListesi.SelectedIndex = e.NewDate.Month - 1

```

```

End Sub

```

```

Sub programaEkleDugmesineBasildi ( sender As Object , e As EventArgs )

```

```

    Dim geriPostalamaSayisi As Integer
    If Page.IsPostBack Then
        geriPostalamaSayisi = ViewState( "PostBackCnt" )
        ViewState( "PostBackCnt" ) = geriPostalamaSayisi
    End If
    tarihEtiketi.Text &= saatListesi.SelectedItem.Value & " " &
dersListesi.SelectedItem.Value & " "

```

```

End Sub

```

```

Sub programiSilDugmesineBasildi ( sender As Object , e As EventArgs )

```

```

    tarihEtiketi.Text = ""

```

```

End Sub

```

```

Sub programiEkleDugmesineBasildi ( sender As Object , e As EventArgs )

```

```

    Dim baglantici As OleDbConnection
    Dim sorguYurutucu As OleDbCommand
    Dim veriOkuyucu As OleDbDataReader

```

```

Dim veritabaniIslemiYurutucu As OleDbTransaction
Dim baglantiDizesi As String
Dim sorgu As String
Dim eklemeSorgusuBasariBayragi As Integer
Dim ekleneTerimSayisi, karakterSayisi, indeks As Integer
Dim seciliTarih As String

baglantiDizesi = "Provider=Microsoft.Jet.OleDb.4.0;" & _
                "Data Source=" & _
                server.mappath("derstakvimi.mdb") & ";"

Try
    baglantici = New OleDbConnection ( baglantiDizesi )
    baglantici.open ( )
    veritabaniIslemiYurutucu = baglantici.BeginTransaction ( )

    seciliTarih = takvim.SelectedDate.Day ( ) & "." &
takvim.SelectedDate.Month ( ) & "." & takvim.SelectedDate.Year ( )
    sorgu = "INSERT INTO Dersgunu (" & _
            "tarih, " & _
            "kurul_id, " & _
            "program" & _
            ") VALUES (" & _
            seciliTarih & ", " & _
            kurulListesi.SelectedItem.Value & ", " & _
            tarihEtiketi.Text & ")"

    sorguYurutucu = New OleDbCommand ( sorgu , baglantici )
    sorguYurutucu.Transaction = veritabaniIslemiYurutucu
    eklemeSorgusuBasariBayragi = sorguYurutucu.ExecuteNonQuery ( )
    If eklemeSorgusuBasariBayragi > 0 Then
        sabitEtiket.text = "Program veritabanına başarıyla eklendi."
        veritabaniIslemiYurutucu.Commit ( )
        <%--veriOkuyucu.Close ( ) --%>
        sorguYurutucu.Dispose ( )
        baglantici.Close ( )
    Else
        sabitEtiket.text = "Hata 1 Beklenmedik bir hata oluştu. Lütfen
Sunucu Yöneticisine bu durumu bildirin."
        veritabaniIslemiYurutucu.Rollback ( )

```

```
<!-- Burada Terimler tablosuna yapılan ekleme silinmeli -->
```

```
End If
```

```
Catch exc As OleDbException
```

```
    sabitEtiket.text = "Özel bir programatik durum ile karşılaşıldığından  
işleminiz tamamlanamayacak. Durum mesajı: " & exc.Message & " Eklenen Terim Sayısı: " &  
eklenenTerimSayisi & " secili tarih:" & seciliTarih
```

```
    <!--veritabaniIslemiYurutucu.Rollback ()
```

```
    veriOkuyucu.Close ()
```

```
    sorguYurutucu.Dispose () -->
```

```
    baglantici.Close ()
```

```
Finally
```

```
End Try
```

```
End Sub
```

```
</script>
```

```
<html>
```

```
<head>
```

```
<title>Akademik Takvime Kayıt Ekleme</title>
```

```
</head>
```

```
<body bgcolor="#2E408C" vlink="#CCCCDD">
```

```
<font face="Tahoma" color="#cccccc" size="2">
```

```
<div style="position: absolute; visibility: visible; width: 201; height: 445; left: 44; top: 20">
```

```

```

```
</div>
```

```
<div style="position: absolute; visibility: visible; width: 599; height: 800; left: 244; top: 33">
```

```
<form runat="server">
```

```
<h2>Akademik Takvime Kayıt Ekleme Adımları:</h2>
```

```
<p>1. Kayıt Ekleme İstedığınız Dönemi Seçiniz:<br>
```

```
<asp:DropDownList id="donemListesi" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="donemListesindeDonemSecildi">
    <asp:ListItem value="1">Dönem I</asp:ListItem>
    <asp:ListItem value="2">Dönem II</asp:ListItem>
    <asp:ListItem value="3">Dönem III</asp:ListItem>
    <asp:ListItem value="4">Dönem IV</asp:ListItem>
</asp:DropDownList>
</p>
```

```
<p>2. Seçtiğiniz Döneme Ait Kurullardan Birini Seçiniz:<br>
<asp:DropDownList id="kurulListesi" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="kurulListesindeKurulSecildi">
    <asp:ListItem value="1">Önce Dönem Seçiniz</asp:ListItem>
</asp:DropDownList>
</p>
```

```
<p>
3.Programını Ekleme İstedığınız Tarihin Ay ve Yılı Giriniz:<br>
<asp:DropDownList id="ayListesi" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ayListesindeAyDegisti">
    <asp:ListItem value="1">Ocak</asp:ListItem>
    <asp:ListItem value="2">Şubat</asp:ListItem>
    <asp:ListItem value="3">Mart</asp:ListItem>
    <asp:ListItem value="4">Nisan</asp:ListItem>
    <asp:ListItem value="5">Mayıs</asp:ListItem>
    <asp:ListItem value="6">Haziran</asp:ListItem>
    <asp:ListItem value="7">Temmuz</asp:ListItem>
    <asp:ListItem value="8">Ağustos</asp:ListItem>
    <asp:ListItem value="9">Eylül</asp:ListItem>
    <asp:ListItem value="10">Ekim</asp:ListItem>
    <asp:ListItem value="11">Kasım</asp:ListItem>
    <asp:ListItem value="12">Aralık</asp:ListItem>
</asp:DropDownList>
```

```
<asp:TextBox id="yilMetinKutusu" runat="server" columns="6" MaxLength="4"
AutoPostBack="True" OnTextChanged="metinKutusundaYilDegisti" />
<asp:RangeValidator id="yilDegeriKontrolcusu" runat="server"
ControlToValidate="yilMetinKutusu"
Type="integer">
```

```
        MinimumValue="2005"
        MaximumValue="2007"
        Display="Dynamic"
        ErrorMessage="Girdiğiniz Yıl Değeri İçin Ders Programı Girilemez." />
    </p>
```

```
<p>
```

4.Programını Ekleme İstedığınız Güne Tıklayınız.


```
<asp:calendar id="takvim" runat="server" onVisibleMonthChanged="takvimdeAyDegisti"
onSelectionChanged="takvimdeGunSecildi" />
</p>
```

<p>5. Seçtiğiniz Tarihin Programını Belirlemek için Sırayla Her Bir Ders Saati İçin Bir Ders Seçin:


```
<asp:DropDownList id="saatlistesi" runat="server" AutoPostBack="False">
    <asp:ListItem value="1">Önce Kurul Seçiniz</asp:ListItem>
```

```
</asp:DropDownList>
```

```
</p>
```

```
<asp:DropDownList id="derslistesi" runat="server" AutoPostBack="False">
    <asp:ListItem value="1">Önce Kurul Seçiniz</asp:ListItem>
</asp:DropDownList>
```

```
<p><asp:Button Text="Programa Ekle" id="programaEkleDugmesi" runat="server"
onClick="programaEkleDugmesineBasildi"/></p>
```

```
<p>
```

```
<asp:Literal Text="Hazirlanan Program: " id="sabitEtiket" runat="server"/>
```

```
<asp:Literal Text="" id="tarihEtiketi" runat="server"/>
```

```
</p>
```

```
<p><asp:Button Text="Programı Sil" id="programiSilDugmesi" runat="server"
onClick="programiSilDugmesineBasildi" />
```

```
<asp:Button Text="Programı Ekle" id="programiEkleDugmesi" runat="server"
onClick="programiEkleDugmesineBasildi" />
```

```
</p>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```


ÖZGEÇMİŞ

Yılmaz Kemal YÜCE, 1977 yılında Çanakkale’de doğdu. İlk ve orta öğrenimini Antalya’da tamamladıktan sonra 1995 yılında Doğu Akdeniz Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü’nde üniversite eğitime başladı. 2000 yılında lisans eğitimini tamamladıktan sonra yine aynı yıl Doğu Akdeniz Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü’nde dağıtık sistemler üzerine Yüksek Lisans eğitime başladı. 2003 yılında bu programı başarıyla tamamladıktan sonra, Akdeniz Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Anabilim Dalı’nda Yüksek Lisans eğitime başladı. Halen Akdeniz Üniversitesi Sağlık Bilimleri Enstitüsü Biyoistatistik Anabilim Dalı’nda Araştırma Görevlisi olarak çalışmaktadır. Yabancı dili İngilizce’dir. Evlidir.