

Web Tabanlı Ders Yönetim ve Değerlendirmede Yeni Bir Yaklaşım

Selvihan N. Yavuzer¹, Orhan Gökçöl²

^{1,2} Bahçeşehir Üniversitesi, Mühendislik Fakültesi, 34538,
Bahçeşehir, Büyükçekmece, İstanbul

¹ selvi@eng.bahcesehir.edu.tr, ² orhan@eng.bahcesehir.edu.tr

Özet. İnternet teknolojilerinin yaygınlaşması ile web tabanlı ders yönetim sistemleri kavramdan gerçeğe dönüşmüştür. Öğretim elemanı ve öğrenci arasında iletişimi sağlayan ders yönetim sistemlerinin, tasarım arayüzü ve içerik yapılandırma, iletişim ve işbirliğini kolaylaştırma, ve ders yönetim desteği sağlama potansiyelleri vardır. Farklı araçlar sunan birçok ders yönetim sistemi mevcuttur. Bu sistemler çok sayıda fonksiyonellikler sunmalarına rağmen bir çoğu eğitim kurumlarındaki gerçek uygulamaları gerektiği şekilde destekleyecek yapıya sahip değildir ve az kullanılan ya da hiç kullanılmayan birçok özelliği barındırırlar. Bu çalışmada, bir üniversite öğretim üyesinin verdiği dersi değerlendirirken "akademik danışmanlık" katmanına da inebilen bir yaklaşımla yeni bir metodoloji ve yönetim modeli sunulacaktır. Model, sınav değerlendirmelerini ve ders danışmanlığı hizmetlerini ortak bir platformda tutmayı ve bu platformun esnek bir şekilde yönetimini amaçlamaktadır [1].

1 Giriş

Son yıllarda, internet teknolojilerinin uzaktan eğitim ve öğrenme yönetim sistemleri için kabul edilir araçlar haline geldikleri görülmektedir. Bu sistemlerin avantajları 30 yılı aşkın süredir belgelenmektedir. Geçtiğimiz on yılda yüksek öğretim alanında bir çok yapısal değişiklik meydana gelmiştir. Öğrenci sayısındaki artış ve daha modüler programlara olan talep bu değişikliklerden bazılarıdır. Üniversiteler öğrencilerin eğitim ortamlarının geliştirmesinde hedef uyumsuzlukları ile karşı karşıya da kalmışlardır [2]. Bu kısıtlamalar kurumları internet üzerinden erişilebilen bilgisayar tabanlı bir ders yönetim sistemine yöneltmiştir. Ders yönetim sistemleri derslerin kampüs dışı öğrencilere iletilmesi ve etkileşimli bir web arayüzü kullanılarak değerlendirilmesine odaklanmaktadır. Diğer yandan, normal verilen derslerin son yıllarda web-destekli (blended learning) olarak tasarmlandığını da görmekteyiz. Öğretim elemanları öğrencileri her ders için bir çok farklı yolla değerlendirir. Bazıları bir çok sınav yaparken, bazıları ödev veya laboratuvar çalışması, bazıları da projeler verir. Öğretim elemanları arasında sabit bir ders değerlendirme yaklaşımı olmadığı söylenebilir. Her dersin bağlı olduğu disiplinden gelen bir doğası vardır ve görsel sanatlar dersinin değerlendirilmesi kuşkusuz mühendislik matematiği dersinden farklı olacaktır.

Bu çalışmada, web-destekli ders tasarımlarında, her disiplinden dersin istenildiği şekilde tanımlanmasına ve ders danışmanlığı yapılmasına olanak tanıyan yeni bir metodoloji geliştirilmiş; .NET ortamı ve C# kullanılarak bu kavramlar bir web tabanlı uygulama haline getirilmiştir [1].

2 Ders Yönetim Sistemleri

Web tabanlı Ders Yönetim Sistemleri (DYS), dersi veren eğiticinin dersle ilgili her türlü bilgiyi öğrencilere güvenli bir ortamda sunmasını sağlar. [3]. Öğrenciler web arayüzleri ile ders sayfasına bağlanırlar ve sınav sonuçlarını öğrenme, ödev gönderme, dersle ilgili tartışma forumlarına katılma vb gibi aktiviteleri gerçekleştirirler. Dersi veren eğiticisi ise, yine güvenli bir ortamda, dersle ilgili her türlü bilgiyi günceller.

DYS, “*eğiticilerle öğrenciler arasında bir iletişim aracı gibi çalışan bilgisayar destekli bir ortamdır*” [1].

DYS’lerin işlevsellikleri 3 ana grupta toplanabilir : (1) arayüz tasarımı ve içerik yönetimi; (2) ders ile ilgili taraflar arasındaki etkileşim ve paylaşımın sağlanması; ve (3) ders yönetimi desteği. Günümüzde DYS, bir eğiticisi için ders kitabı ya da karatahta kadar gereklidir. Eğitimin etkinliğinin ölçülmesi ve gerçekçi değerlendirilmesi, içeriğin geliştirilmesi ve yeni metodolojiler bulunabilmesi için DYS’lerin kullanımı yardımcı olmaktadır [4].

Bu alanda yapılmış çalışmalar incelendiğinde, son yıllarda geliştirilen web tabanlı 100’ün üzerinde farklı DYS olduğunu görürüz. Bunlardan birçoğu uzaktan eğitim amacı ile de kullanılmaktadır. Bu sistemlerin birbirlerine göre üstünlükleri ve belli başlı özellikleri değişik kaynaklarda özetlenmiştir [1].

3 Ders Yönetim Sistemi Geliştirilmesi

DYS’ler incelendiğinde, kullanıcılar tarafından önemli olarak nitelendirilmeyen ve yaygın olarak da kullanılmayan birçok aracı barındırdıkları görülmektedir. Bazı araçlar kullanıcılar sisteme alıştıkça kullanılabilir olmalarına rağmen, o aşamaya kadar temel araçları içeren ve kullanımı kolay bir sistem tercih edilmektedir [5]. Sonuç olarak, bir DYS geliştirilirken, sistemin neler yapabileceğinden çok sistemin neler yapması beklendiğine odaklanılmalıdır. Ders tanımları ve ders değerlendirmeleri gibi DYS’ nin temel yapıları dikkatlice tasarlanmalı ve maksimum kullanılabilirlikle

uygulanmalıdır. Grup özellikleri sistemin birincil görevlerini yerine getirmesi sağlandıktan sonra uygulanabilir.

3.1 Uygulamanın Tanımlanması

Yukarıda açıklanan sebeplerden dolayı, ders yönetim sistemlerinde her türlü dersin ihtiyaçlarını karşılayacak bir yaklaşım gerekmektedir. Bu tarz bir sistem, ders kayıtlarını ve değerlendirme metodlarını tanımlayabilecek kadar esnek olmalıdır. Bu çalışmada her türlü dersin değerlendirmesini kapsayacak yeni bir model geliştirilmiştir [1]. Kullanılan yaklaşım aşağıda belirtilen alanlarda geleneksel yazılım paketlerinden farklıdır:

- Ders değerlendirmesindeki sınırlamaların dinamik ve esnek değerlendirme yapıları kurularak kaldırılması,
- Öğretim elemanına ders tanımlarını ve değerlendirme metodlarını dönem süresince değiştirme olanağı sağlanması ,
- Açık bir değerlendirme (öğretim elemanının isteği ile sınırlı) sağlayarak öğrencilere sınav kağıtları üzerine yorumları da içeren bir sonuç bilgisi sağlanması,
- Öğrencilere yalın bir sayı yığını yerine, notlar ve öğretim elemanı yorum ve önerilerini içeren anlamlı bir değerlendirme bilgisi sunulması,
- Derslerle ilgili problemleri ortadan kaldırarak akademik danışmanlara öğrencilere daha fazla yönelme fırsatı verilmesi.

3.2 Geliştirme Araçları

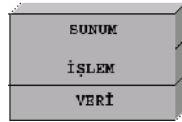
Web uygulama geliştirme platformu olarak Visual Studio.NET seçilmiştir [6]. .NET, veri ile çalışmak için bir çok araç sunması ve geliştirme araçlarının entegre kullanılması nedeniyle en iyi seçenek olarak görülmüştür.

.NET altında geliştirilen uygulama kodlamaları için C# programlama dili kullanılmıştır. C#, C ve C++' dan türetilmiş yeni bir nesne tabanlı programlama dili olarak geliştirme aşamasını hızlandırmaktadır [6].

Uygulama veritabanı SQL Server 2000 ile hazırlanmıştır. Tüm sistem, Windows 2000/XP Sunucularda çalışmaktadır.

3.3 Yazılım Katmanlama

Katmanlama genel anlamıyla bir yazılımın parçalarının diğer parçalardan oluştuğu ve/veya diğer parçalara ihtiyaç duyduğu çağırma sıradüzenidir. Yatay, dikey, bölünmüş, özelden genele gibi bir çok katmanlama yaklaşımı bulunmaktadır. Bu çalışmada Şekil 1' de gösterilen standart yatay katmanlama kullanılmıştır.



Şekil 1. Standart yatay katmanlama modeli

Burada, **Veri katmanı** verinin kalıcı olarak – örneğin bir veritabanına- depolanması işlemini yönetir. **İşlem Katmanı** tüm erişim kuralları ve geçişleri kontrol eder. Eğer yeni bir uygulama veriye erişmek istiyorsa, iş katmanı aracılığıyla çalışmak zorundadır. Bu uygulama veritabanına dolaysız olarak SQL gönderemez. Bu, katman nesneye yönelik bir uygulama olmasa da, nesneye yönelik tasarımıla örtüşmektedir – nesnelere veriyi ve mantığı enkapsüle eder -. **Sunum Katmanı** verileri gösterir, kullanıcının klavye ve fare hareketlerini kabul eder ve uygulama-özel navigasyon konularını yönetir. Sunum katmanı uygulamaya özel katmandır. İşlem katmanının parçaları devam eden bir işin kurallarını (satış, müşteri servisi vb) verirken, sunum katmanı işlem içindeki özel görevleri (yeni sipariş girişi, hesap özeti, vb.) belirler [7].

Yatay katmanlama modelinde bir katman, altındaki diğer katmanlardan birine bağımlı olabilir, fakat bunun tersi söz konusu değildir. Örneğin, sunum katmanı veriye erişmek için işlem katmanına bağımlıdır, fakat işlem katmanı sunum katmanına bağımlı olamaz [7]. Bu çalışmada kullanılan yatay katmanlama modelinde, katmanların iç yapısı Şekil 2' de görüldüğü gibidir.



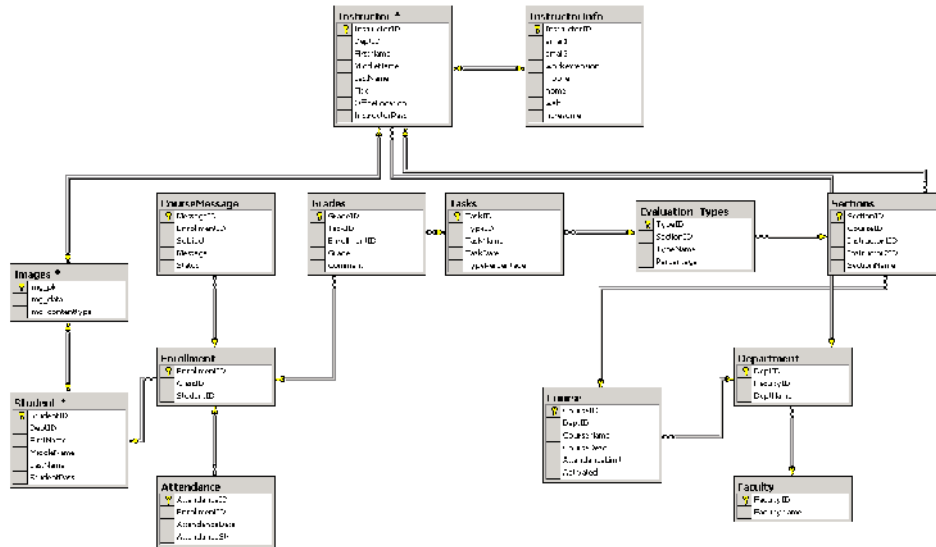
Şekil 2. Çalışmada kullanılan yatay katmanların iç yapısı

Örneğin, öğretim elemanı seçtiği öğrencinin belirli bir etkinlik için not ve yorum bilgilerini güncellemek istediğinde, öncelikle sunum katmanında gerekli alanlara bilgileri girer ve gerekli form ögesini (komut düğmesi, vb.) kullanarak isteğini uygulamaya bildirir. İlk aşamada kullanıcının girdiği bilgilerin uygunluğu (not bilgisinin sayısal değer olması vb.) kontrol edilir (GİRİŞ KONTROLÜ). İkinci aşamada kullanıcının not bilgisini değiştirmeye yetkili olup olmadığı (Kullanıcı sisteme uygun biçimde giriş yapmış mı? Kullanıcı öğretim görevlisi mi?) kontrol edilir (GÜVENLİK KONTROLÜ). Bu aşamadan sonra, öğretim elemanı servisinin sağladığı işlem için gereken modül çağrılır. Bu modül veri katmanını üzerindeki veritabanı erişim arayüzünü kullanarak veritabanına erişir ve işlemi gerçekleştirerek -var ise- sonuç bilgisini sunum katmanına iletilmek üzere işlem katmanına gönderir.

3.4 Veritabanı Tasarımı

Bu tip bir uygulamanın veri tasarımı aşamasında hedef kullanıcılar ile dikkatli bir görüşmenin yapılması ve uygulamanın gerçeğe uygunluğunun sağlanması için eğitim kurumlarının uygulamaları hakkında gerekli bilgi elde edildikten sonra modelleme aşamasına geçilmesi önemlidir.

Bu çalışmada da benzer bir yöntem izlenmiş, halen kullanılan sistemler ile öğretim elemanlarının ve öğrencilerin ihtiyaçları karşılaştırılmış, sonuç olarak mevcut model ortaya çıkmıştır. Şekil 3 bu çalışmada geliştirilen veritabanına ait E-R diyagramını göstermektedir.



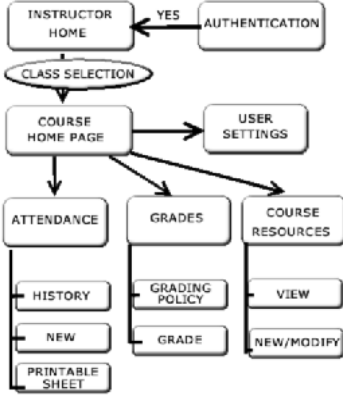
Şekil 3. E-R Diyagramı

Çalışmada veri erişimi için standart SQL komutları yerine kayıtlı yordamlar (stored-procedures) kullanılmıştır. Böylece veritabanına erişim belli metodlarla kısıtlanarak güvenlik artırılmıştır.

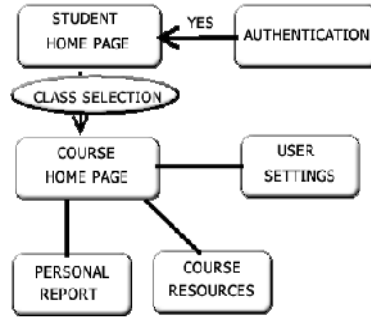
3.5 Uygulama Özellikleri

Varolan DYS' ler üzerinde yapılan testler sonucunda belirlenen tasarım karakteristikleri aşağıdaki gibidir;

- Her arayüzün ana yapısı sistemin tüm parçalarına bağlantılar içermelidir,
- Güvenlik kısıtlamaları farklı sayfalara erişimi zorlaştırmamalıdır,
- Tanımlayıcı yazılar kullanıcıların önemli sistem araçlarını nerede bulabileceklerine dair doğru tahminler yapmalarını sağlayacak nitelikte olmalıdır [5],
- Öğretim elemanının görevleri ve muhtemel senaryolar kullanılarak, Şekil 4' deki öğretim elemanı arayüz haritası tasarlanmıştır,
- Öğrencilerin beklentileri ve muhtemel senaryolar göz önünde bulundurularak, Şekil 5' deki öğrenci arayüz haritası tasarlanmıştır.



Şekil 4. Öğretim Elemanı Arayüz Haritası



Şekil 5. Öğrenci Arayüz Haritası

3.6 Esnek Değerlendirme İçin Matematiksel Model

Veritabanı tasarımı kısmında anlatıldığı gibi, çalışmada esnek bir değerlendirme aracı kullanılmıştır. Buna göre dersi veren eğitimci, dersini nasıl değerlendireceğini tanımlar. E değerlendirme tipini (yılıçi sınav, final, vb.) belirtmek üzere, istenilen sayıda (M) değerlendirme tipi olabilir.

$$E_i, i = 1 \dots M \quad (1)$$

(1)'de M değerlendirme sayısını; E_i ise her bir değerlendirme tipini (yıl içi sınav, ödev, proje, atölye çalışması, vb.) gösterir. Daha sonra, her bir değerlendirme tipinin toplam değerlendirme içindeki yüzdeleri (2)'de gösterildiği gibi verilir.

$$E_{wi}, i = 1 \dots M \quad (2)$$

(2) de her bir E_{wi} için $0 < E_{wi} \leq 1$ dir. Ayrıca,

$$\sum_{i=1}^M E_{wi} = 1 \quad (3)$$

şeklinde bir kısıt söz konusudur. Daha sonra, her bir değerlendirme tipinden kaç adet yapılacağı tanımlanır. Söz gelimi, dersin değerlendirme tiplerinden birisi "Proje" ise, kaç tane proje verileceği (4)'de gösterildiği gibi belirlenir.

$$E_{Ni}, i = 1 \dots M \quad (4)$$

Sonraki aşamada, değerlendirme tiplerinin ağırlıkları verilir.

$$E_{wN_{ij}}, \quad i = 1 \dots M, \quad j = 1, \dots, E_{N_i} \quad (5)$$

Burada, her bir E_i için $\sum_{j=1}^{E_{N_i}} E_{wN_{ij}} = 1$ olmalıdır. G_{ij} her bir deęerlendirmeye ait notu belirtmek üzere ilgili dersin geeme notu (6) daki gibi olacaktır.

$$g = \sum_{i=1}^M \sum_{j=1}^{E_{N_i}} E_{wN_{ij}} * G_{ij} \quad (6)$$

3.7 Arayüz Tasarımı ve Kodlama

Bu aşama, yazılım katmanlamada hazırlanan yöntemler ve temel sayfa modeli kullanılarak dinamik içerik hazırlanmasını kapsar. DYS'lerin kendi ders modüllerini geliştirmek için yeterli bilgisayar bilgisine sahip olmayan kullanıcılara gerekli servisi sağlaması gerekir. Bu nedenle, uygulamanın belirli bir senaryo göz önünde bulundurmadan tasarlanması, sistemin kontrol yapılarının teknik bilgisi olmayan kullanıcılardan gelebilecek kural dışı istekleri kavrayabilecek şekilde kurulması önemlidir.

4 Sonuçlar

Bu bölümde, ana hatları ve modeli verilen uygulamanın çalışması sırasındaki bazı ekran görüntüleri verilecek ve sonuçlar tartışılacaktır. Şekil 6, geliştirilen uygulamanın öğretim elemanı modülündeki ana bileşenlerini göstermektedir.



Şekil 6. Öğretim Elemanı Modülü Ana Bileşenleri



Şekil 7. Öğretim Elemanı Ders Deęerlendirme Tanımlamaları

Şekil 7 ise, öğretim elemanının ders deęerlendirmelerini tanımladığı modülü göstermektedir. Burada, istenildiği kadar farklı deęerlendirme ölçütü dinamik olarak girilebilmekte ve deęiştirilebilmektedir.

Şekil 8, öğretim elemanının ders deęerlendirmelerini girdiği ekranı göstermektedir. Şekil 9 ise, öğrencinin ilgili derse ait tüm deęerlendirmelerini göstermektedir. Öğrenci ayrıca, yoklama, sınıf içindeki baęlı başarı vb gibi bilgilerini de görebilmektedir.

Yapılan çalışma, ders deęerlendirmelerini ve ders danışmanlıklarını ortak bir platformda tutmayı ve esnek bir şekilde yönetimini amaçlamaktadır. Veri tabanı bileşenleri ve kullanıcı ara birimi geliştirilmesinde klasik DYS'lerde olmayan ama eğitim kurumlarında sıkça kullanılan bileşenlerin geliştirilmesine ağırlık verilmiş ve .NET'in sunduğu görsel ve dinamik tasarım araçları etkin bir şekilde kullanılmıştır. Mevcut çalışma, "öğrenci merkezli" bir yaklaşım yerine, "esnek ders deęerlendirmesi" merkezli yaklaşımı kullanmaktadır. Böylece, akademik ortamlarda *öğrenci-ders-öğretim elemanı* üçgeni daha belirli ve şeffaf hale gelmekte ve daha etkin öğrenci danışmanlıkları imkanı sağlanmaktadır.

	ÖĞRETMEN	GRUPLAR	LERNER
000750	ALTINBAĞ YUSUFKANAN	0	
000270	ARGAÇ FATİH	0	
000490	BAKICI SELMA SELENA	0	
000200	BOZTÜRK ÇAĞATAY	0	

Şekil 8. Öğretim Elemanı Ders Deęerlendirme Girişi

Öğrenci No	Öğrenci Adı	Öğrenci Soyadı	Öğrenci Durumu
000001	ALP	EREN	0
000002	AYDIN	EMRE	0
000003	BAKIR	EMRE	0
000004	BAKIR	EMRE	0
000005	BAKIR	EMRE	0
000006	BAKIR	EMRE	0
000007	BAKIR	EMRE	0
000008	BAKIR	EMRE	0
000009	BAKIR	EMRE	0
000010	BAKIR	EMRE	0

Şekil 9. Öğrenci Modülü için Ders Deęerlendirme Ekranı

Bu çalışmada sunulan yaklaşım 2002-2003 akademik yılında Bahçeşehir Üniversitesi Mühendislik Fakültesi' nde belirlenen derslerde denenmiş, öğrencilerin ders ile ilgili gelişmeleri sistem üzerinden takip etmeleri sağlanarak öğretim görevlisine olan bağımlılığının azaldığı ve ders etkinliğinin arttığı, ders değerlendirmelerinin daha gerçekçi ölçülebildiği gözlenmiştir. Ayrıntılı deneme çalışmaları devam etmektedir.

Kaynakça

1. Yavuzer, S.N., "A New Approach to Web-Based Course Management Systems", Lisans Tezi, Bahçeşehir Üniversitesi, Mühendislik Fakültesi, Bilg. Müh. Bölümü, 2003.
2. Crisp, G., "A Model for the Implementation and Sustainability of a Course Management System in a Research University", ASCILITE 2002 Konferansı, Aralık 8, 2002.
3. George Washington Üniversitesi, "Prometheus CMS", <http://prometheus.gwu.edu/>
4. Cohen, D., "Course Management Software: The Case for Integrating Libraries", CLIR Issues, Sayı 23, 2001.
5. Halloran, M.E., "Evaluation of Web-Based Course management Software from Faculty and Student User –Centered perspectives", <http://www.usafa.af.mil/iita/Publications.htm>.
6. Sharp, J. ve Jagger, J., "Microsoft Visual C#.NET Step by Step", Microsoft Yayınları, 2002.
7. Standley, J., Software Architecture, <http://www.surfscranton.com/architecture/>