

Üniversite Ortamında Yazılım Geliştirme Deneyimleri

Tuğkan Tuğlular¹, Fatih Algan², Hüseyin Hışıl³, Ozan Akdemir⁴,
Oya Altay⁵, Yılmaz Örer⁶

^{1,2,3,4,5,6} İzmir Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Bölümü, 35430, Urla, İzmir

¹ tugkantuglular@iyte.edu.tr, ² fatihalgan@iyte.edu.tr, ³ huseyinhisil@hotmail.com,

⁴ ozanakdemir@hotmail.com, ⁵ oyalтай@superonline.com, ⁶ yilmazorer@superonline.com

Özet. Üniversitelerin yazılım geliştirme birimlerinde özellikle güncel yazılım teknolojileri kullanılarak projeler üretilmesi ve hayata geçirilmesi anılan teknolojiler hakkında yeterli bilgi birikimi olmamasından dolayı oldukça zordur. Güncel teknolojiler ancak bilgisayar mühendisliği bölümlerinde bu alanda ders veren öğretim üyeleri ile birlikte çalıştıkları lisansüstü öğrencileri tarafından sürekli izlenmekte ve öğrencilere aktarılmaktadır. Özellikle bu derslerde başarılı olan lisans öğrencilerinin katkıları ile proje ekipleri kurulabilmektedir. İzmir Yüksek Teknoloji Enstitüsü'nde böyle bir ekip ile IBM WebSphere ailesi uygulama geliştirme ortamı üzerinde bir proje geliştirilmektedir. Bu bildiriye, ekibin yapısı ve görev dağılımı, çalışma ilkeleri ve yöntemi ile sekiz aylık bir sürede edinilen deneyimler anlatılacaktır. Anılan ekibin özelliği, öğretim üyesi için ders verme ve öğrenciler için ders alma gibi birçok başka çalışmanın yanısıra proje etkinliklerini gerçekleştirebilmesidir.

1 Giriş

Her kurum gibi üniversiteler de işlerini bilgisayar ve hatta İnternet ortamına aktarmak için çalışmaktadır. Kullanılacak yazılımlar genellikle üniversite bünyesinde kurulu ve görevlerinden biri yazılım geliştirmek olan bir birim tarafından üretilmektedir. Bu tür birimler çoğu zaman iş yoğunluğundan yeni teknolojileri izleyip bu teknolojilerle yazılım geliştirme olanağı bulamamaktadır. Nitekim bu nedenle, anılan birimler üniversitelerin bilgisayar mühendisliği bölümleri ile yakın ilişkiye geçerek bu eksikliklerini kapama yoluna gitmektedir. Zaten derslerinde yeni teknolojileri anlatmakla yükümlü olan bilgisayar mühendisliği bölümleri hali hazırda yazılım geliştirme birimlerinin gereksinimi olan bu bilgiye sahiptir. İyi bir işbirliği ile üniversitenin gereksinimi olan yazılımlar son teknolojiler ile geliştirilebilir. Bu bildirinin yazarları da böyle bir amaç için yola çıkmıştır.

Geliştirilecek olan yazılım, Öğrenci İşleri Bilgi Sistemi üniversite bünyesinde en fazla kullanıcıyı kapsayacak projelerden biridir. Kullanıcılar arasında Öğrenci İşleri Daire Başkanlığı ile Mühendislik ve Fen Bilimleri Enstitüsü gibi idari birimler, üniversitede bulunan tüm öğretim elemanları ve öğrenciler yer almaktadır. Başlangıçta sadece idari birimler açısından ciddi anlamda verimlilik ve sıfır hata beklentileri ile yola çıkmış olsa da zaman içinde böyle bir projenin öğretim elemanları ve öğrenciler için de önemli kolaylıklar sağlayacağı ve genel bir kullanıcı memnuniyeti yaratacağı görülmüştür. Doğal olarak İnternet üzerinde yaşayan bir bilgi sistemi, üniversitenin e-dönüşüm vizyonunun hayata geçirilmesine büyük katkıda bulunacak ve kurum kültüründe değişim ve hatta nihayetinde dönüşümlere yol açacaktır.

Bu bildiriye, böyle bir projeyi gerçekleştirmek üzere yola çıkmış ekibin yapısı ve görev dağılımı, çalışma ilkeleri ve yöntemi ile sekiz aylık bir sürede edinilen deneyimler anlatılacaktır.

2 Projenin Tanıtımı

Geliştirilmekte olan Öğrenci İşleri Bilgi Sistemi Projesi'nin amacı; öğrencilerin, öğretim üyelerinin ve ilgili idari birimlerin öğrenci ile ilgili bütün işlemleri İnternet üzerinden yürütebileceği bir sistem geliştirmektir. Anılan bilgi sisteminin üniversite dışı çeşitli kurumlar (örneğin, ÖSYM) ile entegre olabilmesi diğer bir hedef olarak belirlenmiştir. Uç tarafta ise, çok çeşitli son kullanıcı arayüzleri (web, thick-client, kablosuz cihazlar gibi) ile beraber çalışabilecek yetenekte bir orta katman mimarisine sahip olması amaçlanmaktadır.

Bu hedefler göz önüne alındığında, çalışma ortamı bütün bu gereksinimlere yanıt verebildiğini ispatlamış olan J2EE platformu olarak belirlenmiştir. Orta katman mimarisi olarak J2EE platformunun getirdiği çözümlerden biri olan EJB teknolojisinin seçilmesinin iki temel nedeni bulunmaktadır: değişik istemci tiplerine tek bir merkezi modülden cevap verebilme ve güvenli erişim. Uygulamanın üzerinde çalışacağı platform olarak IBM'in J2EE çözümü olan *WebSphere* ailesi seçilmiştir. *WebSphere* uygulama sunucusu olarak güçlülüğünü ve tutarlılığını kanıtlamış bir ürün olmasının yanında tüm tasarım, geliştirme ve test sürecini tek bir IDE içerisinde gerçekleştirebilme olanağı sunmaktadır.

Öğrenci İşleri Bilgi Sistemi başlangıçta sadece Web tabanlı istemciler için hizmet verecek şekilde planlanmıştır. Projenin web katmanında J2EE uygulamalarında yaygın olarak görülen bir *Model-View-Controller* (MVC) çerçevesi olan *Struts* kullanılmıştır. Bunun dışında yoklama işlemleri için geliştirilmiş başka bir çerçeve olan *Log4Java* da yazılım orta katmanına entegre edilmiştir.

3 Yazılım Projesi Kaynakları

Bilgi sistemlerinin geliştirilmesi, uygulamaya sokulması ve yönetilmesi ancak doğru şekilde seçilmiş proje kaynakları ile mümkündür. Bu kaynaklar insan(ekip), donanım ve yazılım kaynakları olarak sınıflandırılabilir [1]. Anılan kaynakların en önemli olanı insan olarak belirmektedir. Yazılım projelerinde çok kritik olan bitirme zamanı göz önüne alındığında takım çalışması, takım içi uyum, haberleşme, insan psikolojisi gibi konular ön plana çıkmaktadır.

3.1 Ekip

Çalışmanın ilk sekiz aylık bölümünde proje ekibi bir öğretim üyesi, bir yüksek lisans öğrencisi ve dört lisans tez öğrencisi olmak üzere altı kişiden oluşmaktaydı. Özellikle çalışma ortamının oluşturulması ve çalışma yönteminin oturtulması adımlarının atılacağı bu ilk bölümde ekibin küçük olmasına özen gösterildi. Proje lideri olan öğretim üyesi, yazılım geliştirme projelerinin farklı aşamalarında çeşitli görevlerde bulunmuş ve bu anlamda ekibin en deneyimli kişiydi. Yüksek lisans öğrencisi ise, üretim için seçilen IBM uygulama geliştirme ve çalıştırma ailesini iyi tanıyan ve bu konuda iki yıllık bir deneyime sahip proje mühendisi idi. Anılan iki kişi projenin başarıya ulaşması için gerekli en kritik görevleri üstlendi. Lisans öğrencileri ise, bölümün en başarılı (bölüm birincisi, bölüm üçüncüsü gibi) ve gelecek iş yaşamlarında özellikle yazılım geliştirme alanında çalışmayı planlayan öğrenciler arasından seçildi. Bu öğrenciler, projeye hazırlanmak için yaz stajlarını benzer bir ürün ailesini kullanan bir şirkette üç aylık bir sürede tamamlamışlardır.

Proje liderinin beş temel görevi bulunmaktaydı. Birincisi, proje çalışmasının planlanması ve olası riskleri öngörerek önlemlerin alınması olarak açıklanabilir. İkincisi, proje çalışmasının hedeflere uygun bir biçimde zaman kısıtları içinde ilerlemesini sağlamaktır. Bir diğeri ise, çalışma ilkelerini belirlemek ve bunları gözetmektir. Dördüncüsü, proje ekibinin uyum içinde ve bir bütün olarak çalışmasını sağlamak olarak düşünülebilir. Son olarak, projenin dış dünya ile bağlantısını kurmak ve bu bağlantıda oluşan problemleri proje ekibine hissettirmeden çözmek idi.

Proje kapsamında geliştirilecek yazılımın tasarımından genel olarak ve orta katmanın geliştirilmesinin tümünden sorumlu olan proje mühendisi, proje elemanları ile sürekli birlikte çalışmakla görevlendirilmişti. Proje elemanlarının yetiştirilmesi, tasarım ve kod bazında yapılanların denetlenmesi yine proje mühendisinin görevleri arasında bulunmaktaydı. Özellikle başarılı bir ekip ile çalışırken, ekibin önüne sürekli yeni hedefler konması ve ekibin hedefe koşmadan önce bilgi ve yetenek açısından hazırlanması önemlidir. Proje mühendisi öğrenciler ile bu görevi yürütürken aynı zamanda ileri aşamalarda karşılaşılması olası problemleri öngörerek bunları proje lideri ile tartışmış ve bu problemler büyük bir oranda başarı ile atlatılmıştır.

Projede kod üreten elemanların yazılım geliştirme sürecine ve onu oluşturan parçalarının gerekliliğine olan inancı bu projenin ilk bölümünü başarılı kılmıştır. Çözümleme, tasarım, kodlama, sınama ve sürekli belgeleme ekibin inancı ile mümkün olmuştur. İlk aşama bittiğinde çalışan modüller ile birlikte belgeleri de hazır idi. Birçok fiziksel kısıt (kampüsün yerleşim merkezine uzaklığı gibi) ve yetersizlik (bilgisayar konfigürasyonu gibi) proje elemanlarının gayretleri ile aşılmıştır. Ekipte yer alan lisans öğrencilerinin iki kişilik iki küçük ekibe ayrılması düşünülmüş, böylece çözümleme ve tasarım aşamalarında verilecek kararların tartışılarak en iyinin bulunması yolu açılmıştır. Bunun ötesinde küçük ekipteki bir kişinin diğer işleriyle meşgul olması ya da rahatsızlanması durumunda bile işlerin bitirilmesi sağlanmıştır.

3.2 Yazılım

Programlama Dili ve Platform.

Sistem içinde tanımlanmış temel yapılar akademik birimler, akademik personel, ilgili personel ve öğrencilerden oluşmaktadır. Temel yapılar birbirlerinden çok farklı olsalar da ortak özellikler içerir. Bu yapılara birer nesne gözüyle bakarsak hepsinin doğduğu, belli bir süre yaşadığı ve sonra öldüğü gözlenmektedir. Sistemde tanımlanan her nesne aslında gerçek hayatın bir modellemesidir. Dolayısıyla bu nesnelerin tüm özelliklerinin kolay, hızlı ve tam olarak tanımlanabilmesi gerekmektedir. Ayrıca aynı sınıfa ait nesnelerin herbiri için benzer yazılım parçacıkları üretmek anlamsız olacaktır. Bu açıdan bakıldığında nesneye yönelimli bir platformda çalışmak modelleme, zaman ve yeniden kullanılabilirlik anlamında avantaj sağlayacaktır. Projede bu gereksinimleri eksiksiz olarak sağlayabilen Sun Microsystems tarafından geliştirilmiş JAVA^(TM) programlama dili seçilmiştir.

Öğrenci İşleri Bilgi Sistemi yapısı gereği karmaşık sunucu servisleri, çok sayıda ve değişik tipte istemciler, güvenlik altyapısı ve ölçeklenebilirlik gibi servislere gereksinim duymaktaydı. Bu gereksinimler dahilinde bilgi sistemi üreten pek çok kuruluş yıllardır özünde aynı olan problemlere geçici ve farklı yaklaşımlar geliştirip uygulamaktadır. Bu çözümler bir standarttan uzak olduğu için bilgi sistemlerinin geliştirilmesi ve güncelleştirilmesi zorlaşmaktadır. Sun Microsystems bu konuya platform oluşturarak yaklaşmış, sık karşılaşılan problemlere uygulanan başarılı çözümleri bir araya getirmiş ve J2EE (Java^(TM) 2 Platform, Enterprise Edition) adı altında piyasaya sürmüştür [2]. Bu durum göz önüne alınarak, çalışma ortamı anılan problemlere cevap verebildiğini ispatlamış olan J2EE platformu olarak belirlenmiştir.

J2EE platformunun getirdiği çözümlerden biri olan EJB teknolojisinin seçilmesinin iki temel nedeni bulunmaktadır: değişik istemci tiplerine tek bir merkezi modülden cevap verebilme ve güvenli erişim [3].

Web Katmanı.

Proje ismi her ne kadar Öğrenci İşleri Bilgi Sistemi olarak belirlendiyse de, sistemin kullanıcı yelpazesinde İYTE Öğrenci İşleri personelinin yanısıra, İYTE Fen Bilimleri Enstitüsü personeli, akademisyenler, memurlar ve öğrenciler yer almaktadır. Kullanıcı çeşitliliği göz önüne alındığına oluşturulacak yazılım altyapısının; performans, güvenlik, role göre öncelik belirleme ve kolay kurulum gibi özellikleri bünyesinde barındırması beklenmektedir. Bu gereksinimleri en iyi sağlayan yaklaşım Web tabanlı uygulama yaklaşımıdır. Bu sayede, kullanıcıların bilgisayar kullanımı ve program kurulumlarına ilişkin bilgilerinin sınırlı olması bir problem olmaktan çıkmıştır. Web tabanlı yazılımlarda kullanıcılar işletim sistemleri ile birlikte sağlanan ya da tercihen kullanacakları bir Web Tarayıcı ile sisteme erişebilir. Sistem içinde çalışan iş mantığı ile erişim denetimleri kullanıcıdan tamamı ile gizlendiği için güvenlik çözümlerinin kolaylıkla uygulanabileceği bir ortam sağlanmıştır.

Web tabanlı JAVA^(TM) uygulamalarında genellikle *Servlet* ve *JSP* kullanılmaktadır. Bu iki klasik yaklaşımdan *Servlet* verinin içeriğine, *JSP* ise verinin biçimine çözüm getirmeye çalışmıştır [4]. Verinin hem içerik hem de gösterim özelliklerinin yoğun kullanımı gerektiğinde bu iki teknoloji bir arada kullanılmakta, dolayısıyla sistemin geliştirilmesi ve yönetilmesindeki zorluk artmaktadır. Projede, bu iki teknolojiyi Model-View-Controller (MVC) tasarım paradigmasını kullanarak bir araya getiren *Struts* çerçevesinden faydalanılmıştır. *Struts* çerçevesinin J2EE platformuyla kolay uyum sağlayabilmesi de bu seçimde etkili olmuştur [5].

Yazılım Geliştirme Ortamı ve Veritabanı Yönetim Sistemi.

Proje geliştirme için seçilen ortamda aranan en önemli özellik entegrasyona açık olma olmuştur. Böylece geliştirme sürecinde ortaya çıkan gereksinimlere göre yeni modüller geliştirme ortamına eklenebilmektedir. Bu konuda günümüzde yaygın tercih edilen ortam IDE (Integrated Development Environment - Entegre Edilmiş Geliştirme Ortamı)'lerdir. Proje için seçilen *IBM WebSphere* IDE'sinin önemli özellikleri mevcuttur. Bu özelliklerden biri, *WebSphere*'in içinde *IBM WebSphere Uygulama Sunucusunun* bir kopyası olan test sunucusunu barındırmasıdır. Bu sunucu, geliştirme sürecinde yazılım ile senkron kalarak onu her an test edilmeye hazır tutar ve *IBM WebSphere Application Server*'a göre daha az sistem kaynağı tüketir [6]. *IBM WebSphere Application Server* ise proje bitiminde kullanılacak esas sunucudur. *WebSphere*'in bir diğer özelliği plug-in mantığı ile tasarlanmış olmasıdır [7]. Bu sayede IDE kurulumundan sonra da ilaveler ile daha kullanışlı duruma gelir. Örneğin, bir modelleme aracı olan *Rational XDE* yazılımı projede kullanılmak üzere *WebSphere*'e entegre edilmiştir [8]. *WebSphere* takım çalışmalarında gereksinim duyulan paylaşım, versiyonlama ve yedekleme konularında da veri havuzu çözümleri sunmaktadır. Kolay kullanımı ve network performansı nedeniyle *CVS* veri havuzu seçilmiştir [9].

WebSphere'in kurumsal uygulamalarda en iyi *IBM DB2* veritabanı yönetim sistemi ile uyum sağladığı görülmüştür ve veritabanı olarak *DB2* kullanılmıştır. Özellikle otomatik *EJB* nesnelere üretiminde bu uyumluluk ekibe zaman kazandırmıştır [10].

Açık Kaynak Yazılımlar.

Proje başlangıcında öngörülen fakat çözümleri netleşmeyen iki problem de açık kaynak yazılımlar ile aşılmıştır. Bunlardan ilki olan *ANT*, *XML* tabanlı bir derleme destek aracıdır. Proje içinde *ANT*, yazılım geliştirmesinde gereksinim duyulacak veritabanı bilgilerini üretmede kullanılmıştır. Böylece veritabanı yedeklerine gereksinim duyulmadan tüm tanımlamalar ve veri girişleri birkaç saniyede otomatik olarak gerçekleştirilmektedir [11, 12].

İkinci açık kaynak yazılım, *Log4Java*, uygulama çalışırken veri değişikliğine yol açan hareketlerin kayıtlarını tutmaya yarar. *Log4Java* orta katmana entegre edilmiş durumdadır. *Log4Java* ile sistemi kullanan bir kullanıcının gerçekleştirdiği her eylemin içeriği, zamanı ve süresi kaydedilmektedir. Sistem güvenliği açısından *Log4Java* destekleyici bir yazılım olarak görülmektedir [13].

3.3 Donanım

Donanım, projede kullanılan yazılımlar kadar önem taşımaktadır. Kişisel bilgisayarlarla kıyaslandığında daha fazla kaynağa gereksinim duyulmaktadır. Bu nedenle yetersiz bir donanım çok zaman kaybettireceği gibi iyi bir donanım da gözardı edilemeyecek kadar zaman kazancı sağlayacaktır.

Proje Geliştirme Ortamı Donanım Özellikleri.

Proje geliştirme sürecinde sistem kaynaklarının en çok zorlandığı noktalar; orta katman derlemelerinin yapılması ve test sunucusunun hata arama (debug) modunda işletilmesi olmuştur. İşletim sistemi'nin bu noktalarda 500-800 MB bellek gereksinimi doğmuştur. Bu açıdan bakıldığında en kritik öge ana bellek olarak görülmektedir. Kullanılan yazılım geliştirme ortamında ideal bellek kapasitesi 1GB olarak tespit edilmiştir. Sıkça yapılan derleme işlemlerinin Pentium-III işlemcili bilgisayarlarda 5-10 dakika gibi süreler alması, Pentium-IV işlemcili bilgisayarlar ile çalışmayı zorunlu kılmaktadır. *WebSphere* ve *Rational XDE*'nin kullanıcı ekranları pek çok kısa yol simgesi içerdiği için monitör çözünürlüğünün en azından 1024x768 piksel seviyesinde tutulması gerekmektedir.

Uygulama Sunucusu Donanım Özellikleri.

Proje bitimi ile beraber üretilen yazılım *IBM WebSphere Application Server* üzerinde çalışmaya başlayacak ve akademik takvimde öngörülen zamanlarda aynı anda pek çok kullanıcıya hizmet verecektir. Sunucunun çalışacağı bilgisayarda aynı anda veritabanının da işletilmesi düşünülmektedir. Ancak veritabanı sunucusunu ayırmanın performans üzerinde olumlu etkileri olabileceği yönünde görüşler de bulunmaktadır. Şu an öngörülen temel konfigürasyon 2.8 GHz x 2 işlemcili 2 GB bellek kapasiteli bir bilgisayardır.

4 Çalışma Yöntemi

Çalışmanın Ekim 2002 – Mayıs 2003 arasındaki birinci sekiz aylık bölümü iki aşamadan oluştu. İlk aşamada proje ekibi için bir çalışma ortamı oluşturuldu. Bu süre, lisans öğrencilerinin yazılım geliştirme ortamını öğrenmeleri, diğer çalışmalarını ile paralel olarak proje çalışmalarını yürütmeyi başarmaları ve birbirleri, proje mühendisi ve proje lideri ile birlikte çalışmaya alışmaları açısından çok faydalı oldu. Ayrıca, proje lideri ve mühendisi geliştirilecek olan yazılımı sıfır düzeyinde dört modüle böldü ve genel hatları ile bu modülleri tanımladı. Üç ayın sonunda ekip yazılım geliştirmeye hazır duruma gelmişti.

İkinci aşamada en önemli hedef, içinde bulunulan koşullara uygun yazılım geliştirme yönteminin oturtulması idi. Kalan beş ay için, yöntem oturtmanın yanısıra projede tüm kullanıcıları ilgilendiren ders kayıt aşamasına gelinmesi, diğer bir deyişle dört modülden üçünün bitirilmesi, hedefi de konulmuştu. Bu kısa sürede hedeflere ulaşmak ve proje elemanlarından en yüksek verimi alabilmek için ardışık çalışma biçimi benimsendi. Böylece her çalışan elindeki işi bitirdiğinde kendisini yeni bir iş bekliyor olacaktı. Bu işler proje mühendisi için veritabanı tasarımı, bileşen tasarımı ve kodlanması, bileşen testleri, fonksiyonel testler, veritabanı yönetimi, uygulama sunucusu yönetimi, veri havuzu yönetimi olarak belirlenmişti. Proje küçük ekiplerinin işleri ise senaryo oluşturulması, arayüz tasarımı, statik sayfa oluşturulması, dinamik sayfa oluşturulması, belgeleme olarak tanımlanmıştı. Bu işler arasına idari birimler ile haftalık görüşmeler serpiştirilmiş ve böylece sürekli bir geri-besleme ortamı yaratıldığı gibi idari birimlerin, yani kullanıcıların, projeye etkin katılımları ve sahiplenmeleri sağlanmıştır.

Yazılım geliştirme yöntemi oturtulmasında en kritik unsurun seçenekler arasından seçim yaparken dikkate alınacak ilkeler olduğu öngörüldü. Bu neden ile ikinci aşamanın başlangıcında anılan ilkeler belirlendi. Bunlar arasında “gerektiği kadar basit”, “kalıba sadık kalma”, “ortam koşullarını sabitleme” ve “sürekli belgeleme” ilkeleri sayılabilir. Örneğin, “gerektiği kadar basit” ilkesi arayüz tasarımlarını sadeleştirdi ve öğrenilmesi ile kullanılmasını kolaylaştırdı. “Kalıba sadık kalma” ilkesi ile üretim süreci hızlandı. Nitekim, kalıba bir kez karar verildikten sonra her yerde aynı kalıp kullanıldı. Böylece, geliştirme ve belgeleme kolaylaştı ve yapılı duruma geldi. “Ortam koşullarını sabitleme” ilkesi de çalışanların problemleri çözmek için yazılım geliştirme ortamına yeni araçlar sokmasını engelledi. Bu sayede sürekli olarak yeni araçların öğrenilmesi, bunların sürece entegre edilmesi gibi zaman alıcı ve heves kırıcı işlerden kurtulmuş oldu.

5 Sonuç

Projenin birinci sekiz aylık bölümü için belirlenen öncelikli hedefler arasında çalışma ortamının oluşturulması ve yazılım geliştirme yönteminin oturtulması bulunuyordu. Çalışma ortamının doğru oluşturulduğu, ilk üç aylık dönemi izleyen ikinci beş aylık aşamada yöntemin oturtulabilmesi ile ortaya çıktı. Yazılım geliştirme yönteminin oturtulması hedefine ulaşıldığı ise, lisans tez öğrencilerinin mezuniyeti ile birlikte proje ekibindeki görev değişimi sonrasında, iki hafta içinde yeni lisans öğrencilerinin web katmanında kod üretebilmesi ile görülmüştür.

Projenin ikinci sekiz aylık bölümü Haziran 2003’de başladı. Bu ikinci bölümde ana hedefler fiziksel ortamda gerçekleştirilen işlerin sıfır hata ile sanal ortama taşınması ve güvenlik gereksinimlerini karşılayacak düzeneklerin bilgi sistemine entegrasyonu olarak belirlendi. Ayrıca, proje ekibinde yer alan lisans öğrencilerinin sayısının iki katına çıkarılması yoluna gidilerek birim zamanda daha çok iş yapılması amaçlandı.

Kaynakça

1. Pressman, R. S., *Software Engineering A Practitioner’s Approach*, 3rd Edition, McGraw Hill International Editions, Computer Science Series, 1992, ISBN: 0-07-112779-8.
2. The JAVA^(TM) 2 Enterprise Edition Developer’s Guide Version 1.2.1, May 2000.
3. Monson-Haefel, R., *Enterprise Java Beans*, O’Reilly&Associates, 2001, ISBN 0596002262.
4. Hall, M., *Core Servlets and Java Server Pages*, Java2 Platform, Enterprise Edition Series, ISBN 0-13-089340-4.
5. Goodwill, J., *Mastering Jakarta Struts*, Wiley Publishing, Inc., ISBN 0-471-21302-0.
6. Nartovich, A., Bennett, K., Burgmann, K., Edwards, C., Lowery, E. ve Poole, R., *WebSphere Application Server Enterprise Edition 4.0: A Programmer’s Guide*, IBM Redbooks Collections, February 2002, ISBN 0738423998.
7. Sadtler, C. ve Jacob, A., *An Introduction to IBM WebSphere Studio Application Developer*, IBM Redbooks Collections, October

2001, <http://www-900.ibm.com/developerWorks/cn/wsdd/support/redbook/wsad-redp0414.pdf>.

8. Rational Developer Network, Rational® XDE™ Professional v2002 Release 2 – Java™ Platform Edition Tutorial, 2002, http://www.rational.com/alliances/IBM_Project/ps-2059/ps-2059.html.
9. Tagagiwa, O., Korchmar, J., Lindquist, A. ve Vojtko, M., WebSphere Studio Application Developer Programming Guide, IBM RedBooks Collections, August 2002, ISBN 0738425826.
10. Baklarz, G. ve Wong, B., DB2 Universal Database v8, Data Administration Guide, IBM Press, ISBN 0-13-046361-2.
11. Hightower, R. ve Lesiecki, N., Java Tools for Extreme Programming, Wiley Publishing, ISBN 0-471-20708-X.
12. Apache Software Foundation, The Apache Ant Project, <http://ant.apache.org/>.
13. Apache Software Foundation, The Apache Jakarta Project, <http://jakarta.apache.org/log4j/docs/index.html>.