

**T.C.  
MUĞLA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI**

**JAVA TEKNOLOJİSİ KULLANARAK  
İNTERNET TABANLI ÖĞRENCİ KAYIT SİSTEMİ**

**YÜKSEK LİSANS TEZİ**

**KÜRŞAT KURT**

**MAYIS 2005**

**MUĞLA**

**T.C.  
MUĞLA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI**

**JAVA TEKNOLOJİSİ KULLANARAK  
İNTERNET TABANLI ÖĞRENCİ KAYIT SİSTEMİ**

**YÜKSEK LİSANS TEZİ**

**KÜRŞAT KURT**

**MUĞLA 2005**

## ONAY SAYFASI

Dr. Mehmet KARAHASAN danışmanlığında Kürşat KURT tarafından hazırlanan bu çalışma, 06/04/2005 tarihinde aşağıdaki jüri tarafından İstatistik ve Bilgisayar Bilimleri Anabilim Dalı'nda yüksek lisans tezi olarak oybirliği ile kabul edilmiştir.

Başkan : Dr. Mehmet KARAHASAN İmza :

Üye : Yrd.Doç.Dr. Taner DİNÇER İmza :

Üye : Yrd.Doç.Dr. İlhan TARIMER İmza :

## ÖNSÖZ

Bu çalışma ile üniversitelerde el ile yapılan kayıt sistemlerindeki ve kullanılmakta olan İnternet kayıt sistemlerindeki ortak meseleleri çözebilen, öğrencilerin İnternet üzerinden kayıt yapabilmesine imkân veren öğrenci İnternet kayıt sistemi geliştirilmiştir. Çalışma sonucunda ortaya konan kayıt sistemi, Muğla Üniversitesi'nde kullanılmaktadır.

Çalışmamın geliştirilmesinde yardımcı olan ve yönlendiren hocalarım Yrd.Doç.Dr. Taner Dinçer, Dr. Mehmet Karahasan ve Muğla Üniversitesi Bilgi İşlem Daire Başkanı Osman Keleş'e teşekkürlerimi sunarım. Ayrıca çalışmanın geliştirilmesi aşamasında, diğer üniversitelerdeki kayıt sistemleri ile ilgili bilgiler veren ve yardımcı olan, kaynakçada adı geçen herkese teşekkür ederim.

## İÇİNDEKİLER DİZİNİ

ÖNSÖZ.....	I
İÇİNDEKİLER DİZİNİ.....	II
ÖZET.....	V
ABSTRACT.....	VI
ŞEKİLLER DİZİNİ.....	VII
TABLolar/ÇİZELGELER DİZİNİ.....	VIII
SEMBOLLER ve KISATMALAR DİZİNİ.....	IX
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	10
3. MATERYAL VE YÖNTEM.....	11
4. JAVA TEKNOLOJİSİ.....	13
4.1 Java Teknolojisi Yapısı.....	13
4.2 Java Kodlarının Çalışma Şekli.....	14
4.3 Yazılım Mimarileri ve Java'nın Yeri.....	16
4.3.1 İstemci-sunucu mimarisi.....	16
4.3.2 Üç katlı mimari.....	17
4.3.3 Çok katlı mimari.....	19
4.4 Servletler.....	21
4.5 Java İsimlendirme ve Rehber Arayüzü (JNDI).....	21
4.6 Java Veritabanı Bağı (Java Database Connectivity-JDBC).....	21
4.7 Java Elektronik Posta (Java Mail).....	22
4.8 Java İleti Hizmeti (Java Messaging Service-JMS).....	23
4.9 Ejb (Enterprise Java Beans).....	23

4.10 Java Applet.....	23
4.11 Java Sunucu Sayfaları (Java Server Pages-JSP).....	24
4.11.1 Jsp sayfalarının çalışma şekli.....	24
4.11.2 Jsp'nin avantajları.....	25
4.11.3 Java yazılım geliştirme araçları (JSDK).....	26
4.11.4 Java uygulama sunucuları.....	27
4.12 Java Teknolojisinde Güvenlik.....	28
4.12.1 Java programlama dili yapısındaki kod güvenliği.....	28
4.12.2 Java sistem güvenliği.....	29
4.13 Java Script.....	30
4.14 Java Teknolojisi Seçim Nedenleri.....	30
5.ÖĞRENCİ KAYIT SİSTEMİ TANIMI VE YAPISI.....	33
5.1 Klasik Kayıt Aşamalarının İncelenmesi.....	33
5.1.1 Kontrol ve ön hazırlık aşaması.....	33
5.1.2 Kayıt yenileme ve ders kaydı.....	33
5.2 El İle Kayıt Sisteminde Karşılaşılan Meseleler.....	33
5.3 MiKaS Aşamaları.....	34
5.3.1 Kontrol ve ön hazırlık aşaması.....	34
5.3.2 Kayıt yenileme ve ders kaydı.....	35
5.4 Kayıt Sistemi Tablo Yapıları.....	35
5.4.1 Öğrenci genel bilgileri tablo grubu.....	35
5.4.2 Öğrenci dönem bilgileri tablo grubu.....	36
5.4.3 Öğrenci ders bilgileri tablo grubu.....	37
5.4.4 Öğrenci bölüm bilgileri tablo grubu.....	38
5.4.5 Öğrenci harç bilgileri tablo grubu.....	39

5.4.6 Sistem dönem bilgileri tablo grubu.....	41
5.4.7 Öğrenci şifre bilgileri tablo grubu.....	42
5.5 Öğrenci Bilgileri Tablo İlişkileri.....	43
5.6 Kayıt Sistemi Tanımı.....	45
5.6.1 Birinci aşama: açıklama ve sistem bilgilerinin doğrulanması.....	45
5.6.2 İkinci aşama: harç denetimi ve kayıt yenileme.....	46
5.6.3 Üçüncü aşama (1.ana aşama): önceki dönemlerden başarısız olunan zorunlu derslerin seçimi.....	47
5.6.4 Dördüncü aşama (ikinci ana aşama): mevcut dönemden zorunlu derslerin seçimi.....	50
5.6.5 Beşinci aşama (üçüncü ana aşama): mevcut dönemden seçmeli derslerin seçimi.....	52
5.6.6 Altıncı aşama: alınan ders listesinin görüntülenmesi.....	54
5.7 Uygulama Sunucusu.....	56
5.7.1 Weblogic sunucusu kurulum aşamaları.....	58
5.7.2 Weblogic sunucusuna SSL kurulumu.....	58
5.8 Güvenlik Tanımı.....	59
5.9 Kayıt Sistemi Yapısının Düzenlenmesi ve Farklı Kayıt İşleyişlerine Uyarlanması.....	59
6. ARAŞTIRMA BULGULARI.....	60
7. SONUÇLAR VE TARTIŞMA.....	61
KAYNAKLAR .....	63
ÖZGEÇMİŞ.....	65

**JAVA TEKNOLOJİSİ KULLANARAK  
İNTERNET TABANLI ÖĞRENCİ KAYIT SİSTEMİ  
(Yüksek Lisans Tezi)**

**Kürşat KURT**

**MUĞLA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**2005**

**ÖZET**

Bu çalışma, klasik üniversite öğrenci kayıt sistemlerindeki aksaklıkları gidermek, kayıt sırasında oluşan kuyrukları önlemek ve hata oranını azaltmak amacıyla yapılmıştır. Çalışmanın birinci aşaması olarak mevcut öğrenci kayıt sistemleri incelenmiştir. Daha sonra eksikler ve meseleler ortaya konmuş ve çözümler üretilmiştir. Son olarak üretilen çözümler doğrultusunda internet üzerinden kayıt yapılabilmesi için bir sistem geliştirilmiştir. Kayıt sisteminin geliştirileceği platform olan Java teknolojisi incelenmiş, Java teknolojisi içerisinde kullanılacak alt teknolojiler belirlenmiş ve kayıt sistemi geliştirilmiştir.

Önerilen sistem hâlihazırda Muğla Üniversitesi'nde aktif olarak kullanılmaktadır. Uygulamaya geçirilmiş olan kayıt sistemi sayesinde kayıt tarihinin son günündeki öğrenci yığılmaları azalmıştır. Yapılan kayıt işlemleri denetim altına alınmıştır. Ayrıca öğrenciler üniversiteye gelmeden buldukları şehirden kayıt yenileme ve ders kaydı işlemlerini yapabilmişlerdir.

**Anahtar Kelimeler:** EJB, İnternet kayıt sistemi, Java, Jsp, Öğrenci kayıt sistemi.

Sayfa Adedi : 74

Tez yöneticisi : Dr. Mehmet KARAHASAN



**INTERNET BASED STUDENT REGISTRATION SYSTEM  
WITH JAVA TECHNOLOGY  
(M.Sc.Thesis)**

**Kürşat KURT  
MUĞLA UNIVERSITY  
ISNTITUTE of SCIENCE and TECHNOLOGY  
2005**

**ABSTRACT**

This study has been developed for the purpose of eradicating drawbacks of traditional student registration, preventing student gatherings in queues and minimising error rate. As the first step of this development, available student registration system has been analysed. After that, deficiencies and problems have been detected and solutions have been offered. Lastly, under the light of solutions offered, for the purpose of making it possible to do registration on the internet, the system has been developed. Java technology on the platform of which this system will be developed has been analysed. Subtechnologies to be used in java have been determined and registration system has been developed.

The proposed system has still been readily used actively at university of Muğla. By means of the registration system developed for application, the accumulation of students for registration on the last day of registration has been decreased. The registration procedure has been controlled. Besides; without coming to the university for registration, students have been able to renew their registrations and lesson registration procedures in their hometowns.

**Key Words** : EJB, Internet registration system, Java, Jsp, Student registration system.

**Page Number** : 74

**Advisor** : Dr. Mehmet KARAHASAN

## ŞEKİLLER DİZİNİ

<u>Şekil No</u>	<u>Sayfa No</u>
Şekil 1.1 El ile Kayıt ve MiKaS aşamaları .....	3
Şekil 4.1 İşletim Sistemleri ve JVM konumu .....	13
Şekil 4.2 Java kodlarının çalışma aşaması.....	15
Şekil 4.3 İki katlı mimari.....	16
Şekil 4.4 Üç katlı mimari.....	18
Şekil 4.5 Çok katlı mimari.....	19
Şekil 4.6 JSP Teknolojisi çalışma prensibi.....	24
Şekil 4.7 JSP kodlarının çalıştırılma işlemi.....	25
Şekil 5.1 Tablo ilişkileri.....	44
Şekil 5.2 Birinci aşama sayfa yapısı.....	46
Şekil 5.3 İkinci Aşama İşleyişi.....	47
Şekil 5.4 Üçüncü Aşama Ekranı.....	48
Şekil 5.5 Üçüncü aşama sayfa işleyişi.....	49
Şekil 5.6 Dördüncü aşama ekranı .....	51
Şekil 5.7 Dördüncü Aşama Sayfa Yapısı .....	52
Şekil 5.8 Beşinci Aşama Ekranı .....	53
Şekil 5.9 Beşinci Aşama Sayfa Yapısı .....	54
Şekil 5.10. Altıncı aşama ekranı .....	55
Şekil 5.11 Altıncı Aşama Sayfa Yapısı.....	55

## TABLolar/ÇİZELGELER DİZİNİ

<u>Şekil No</u>	<u>Sayfa No</u>
Tablo 3.1 Web sunucusu donanım özellikleri.....	12
Tablo 3.2 Veri Tabanı sunucusu donanım özellikleri.....	12
Tablo 4.1 Google arama motorundaki programlama dillerinin arama sonucu....	31
Tablo 4.2 Programlama dillerine göre proje sayıları.....	32
Tablo 5.1 Stud01 Tablosu .....	36
Tablo 5.2 Stud01Status Tablosu.....	36
Tablo 5.3 Stud01Course Tablosu.....	37
Tablo 5.4 Stud01CourseOverload Tablosu.....	38
Tablo 5.5 Stud01_Program Tablosu.....	38
Tablo 5.6 Stud01_Payments Tablosu.....	39
Tablo 5.7 PaymentTypes Tablosu.....	40
Tablo 5.8 RegistrationParameters.....	41
Tablo 5.9 EduSem Tablosu.....	41
Tablo 5.10 EduYear Tablosu .....	42
Tablo 5.11. StudentLogins Tablosu.....	42
Tablo 5.12 Weblogic Server kurulumu için donanım ve işletim sistemi gereksinimleri .....	56
Tablo 5.13 Weblogic Server sistemi için minimum sistem gereksinimleri .....	57

## SEMBOLLER ve KISATMALAR DİZİNİ

API	Application Programming Interface
CGI	Common Gateway Interface
EJB	Enterprise Java Beans
HTML	Hyper Text Markup Language
HTTP	Hyper Transfer Text Protocol
HTTPS	Secure Hyper Transfer Text Protocol
IDE	Integrated Development Environment
IMAP	Internet Message Access Protocol
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JSDK	Java Standart Development Kit
JSP	Java Server Pages
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MeYOY	Mesaja Yönelik Ortakat Yazılımları
MIME	Multipurpose Internet Mail Extensions
MiKaS	Muğla Üniversitesi İnternet Kayıt Sistemi
MOM	Message Oriented Middleware
ODBC	Open Database Connectivity
POP3	Post Office Protocol 3
SMTP	Simple Mail Trasfer Protocol
SQL	Structered Query Language
SSL	Secure Sockets Layer- Güvenli Yuva Tabakası
TCP/IP	Transmission Control Protocol/ İnternet Protocol
WAP	Wireless Application
XML	Extensible Markup Language

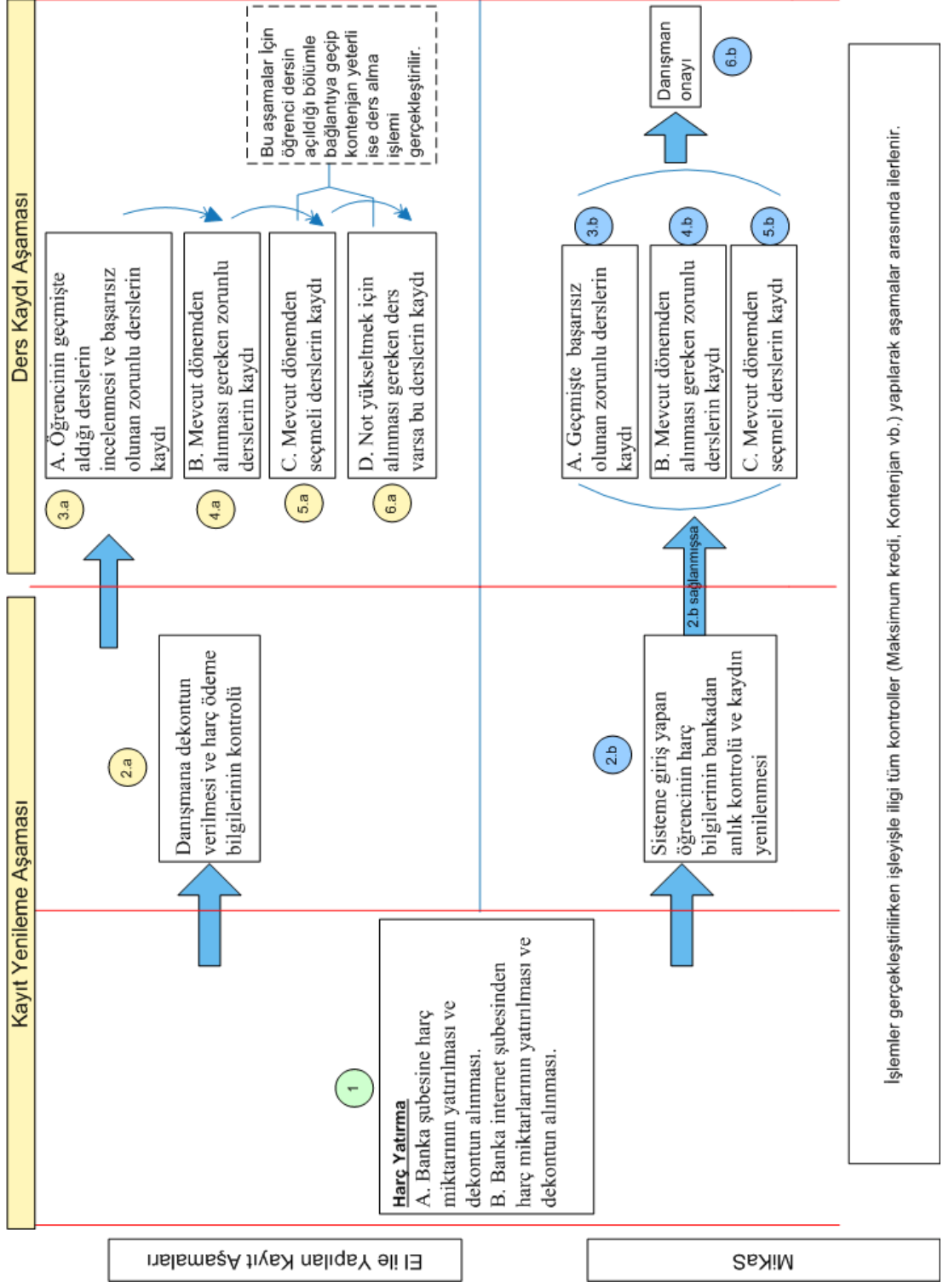
## 1. GİRİŞ

Bu çalışmada; üniversitelerde el ile yapılan kayıt sistemlerindeki ve kullanılmakta olan İnternet kayıt sistemlerindeki ortak meseleleri çözebilen, öğrencilerin İnternet üzerinden kayıt yapabilmesine imkân veren öğrenci İnternet kayıt sistemi geliştirilmiştir. Önerilen Muğla Üniversitesi İnternet Kayıt Sistemi (MiKaS), öncelikle Muğla Üniversitesi'nin mevcut kayıt işleyişini esas alacak şekilde, ek olarak diğer üniversitelerdeki öğrenci İnternet kayıt sistemleri üzerinden elde edilen tecrübelerin de dâhil edilmesi ile ortaya konmuştur. İncelemeler sonunda, el ile kayıt aşamasındaki adımlar ve bu adımlardaki meselelerin sayısal ortamda çözümleri üretilerek modele aktarılmıştır.

Üniversitelerde öğrenci kaydı, kayıt yenileme ve ders kaydı aşamalarından oluşmaktadır. El ile kayıt işlemi gerçekleştiren danışmanlar farklı nedenlerden dolayı zorluklarla karşılaşmaktadırlar. Bu zorluklar, danışmanın takibinde olan öğrenci sayısının fazlalığı yapılan ve takip edilmesi gereken işlemleri artırmaktadır. Ayrıca, kayıt sırasında yapılan kontrollerin sağlıklı ve doğru yapılabilmesi ise ancak her öğrencinin geçmiş dönem bilgilerinin kayıt esnasında erişilebilir olması ile sağlanabilmektedir. Bu ve benzeri el ile kayıta yaşanan olumsuzluklar sayısal ortamın kazanımları ile çözümlenebilmektedir. Bu nedenle üniversitelerin büyük bir çoğunluğu kayıt sistemlerini bilgisayar ortamına taşımış bulunmaktadır.

Kayıt sistemlerinin sayısal ortama taşınması, kayıt işlemlerinin danışmanlar tarafından daha hızlı ve kontrollü bir şekilde yapılabilmesine imkân vermiştir. Bu aşamadan sonra bir adım daha ileri gidilmiştir. Kayıt aşamalarının ve kayıt sırasında dikkat edilmesi gereken kuralların ve işleyişin sayısal ortama aktarılmasıyla, öğrencinin danışmana ihtiyaç duymadan kayıt yapabilmesine imkân sağlayan modeller ortaya çıkmıştır. Bu kayıt sistemi modellerini farklı şekillerde kayıt sistemlerinde kullanan üniversiteler bulunmaktadır.

MiKaS, diđer üniversitelerde kullanılan internetten öğrenci kayıt sistemleri ile karşılaştırılarak ortaya konmuştur. Şekil 1.1.'deki çizgede el ile yapılan kayıt aşamaları ve MiKaS içerisinde sayısal ortamda en iyileştirilmiş biçimleri karşılaştırmalı olarak verilmektedir.



Birinci aşama öğrencinin yatırmakla yükümlü olduğu harç miktarı ve kayıt gideri miktarının eksiksiz olarak üniversite banka hesabına yatırılması işlemidir (Yükseköğretim Kanunu,46.Madde). Bu işlemi öğrenci, üniversitenin anlaşmalı bulunduğu bankanın herhangi bir şubesine bizzat giderek yapabileceği gibi ayrıca bankanın İnternet şubesi üzerinden de harç yatırma işlemini gerçekleştirebilmektedir (Şekil 1.1-1). Bu noktadan sonra kayıt işleyişi öğrencinin kayıt işlemini yapacağı yola göre değişiklik gösterir.

El ile kayıt sisteminde öğrenci harcını yatırdıktan sonra dekontla birlikte bölümüne gelmek zorundadır. Danışmana dekontun verilmesiyle birlikte, derslerin seçimi ve onayı yapıldıktan sonra kayıt yenileme işlemini gerçekleştirmiş olur (Şekil 1.1-2.a). Öncelikle, öğrencinin geçmiş dönemlerden kaldığı zorunlu dersin olup olmadığı geçmiş döneme ait kayıt bilgilerinden bulunarak varsa zorunlu dersin kaydı yapılır (Şekil 1.1-3.a). Başarısız olunan derslerin kaydından sonra mevcut dönemde alınması gereken zorunlu derslerin kaydı, mevcut eğitim-öğretim programı incelenerek yapılır (Şekil 1.1-4.a). Bu aşamada hata yapılmaması için danışmanın bölüm eğitim-öğretim programına sahip olması gerekir (yerine saydırmalar, ders bölünmeleri veya ders birleştirme işlemleri gibi...). Bu tür elle kayıt sisteminde, eğitim-öğretim programında değişiklik yapılan bölümlerde bu aşamada meseleler yaşanmaktadır. Bu meseleler derslerin kapanması, bölünmesi veya birleştirilmesi gibi ders işlemlerinden oluşur ve danışman tarafından takibinin doğru olarak yapılamamasından kaynaklı meseleler ortaya koyar. Zorunlu derslerin kaydı bittikten sonra seçmeli derslerin kaydı yapılmaktadır (Şekil 1.1-5.a, Şekil 1.1-6.a). Bu aşamada seçmeli derslerin kontenjan kontrolü önemli bir yer tutmaktadır. Genel olarak elle yürütülen sistemlerde derse tahsis edilmiş kontenjan aşılmaması için, öğrencinin ilgili dersin açıldığı bölüme gidip yeterli kontenjan olduğu onayını alması ve bunu danışmanına iletmesi gerekmektedir (Adar N.,2005; Akgöbek Ö.,2005; Yiğit B.,2005; Bingöl A.,2005; Çakar F.,2005; Güven A.S.,2005; Kılıçarslan H.G.,2005; Öncü Z.,2005; Özdemir A.,2005; Yıldız b.,2005 ). Bu işlem aslen sayısal ortamın kazanımları ile ortadan rahatlıkla kalkmaktadır.



MiKaS’da öğrenciler harçlarını yatırdıktan sonra İnternet üzerinden sisteme giriş yapıp kayıtlarını oldukları yerden yenileyebilmektedirler. Kayıt yenileme aşamasında MiKaS, anlaşmalı bankadan gelen bilgileri kontrol eder. Diğer üniversitelerdeki sistemlerde anlaşmalı banka ile bilgi aktarımı ortalama olarak günde iki kez yapılmaktadır. MiKaS’ta ise saat başı bilgi güncellemesi yapılmaktadır. Böylece öğrencinin kayıt işlemlerini vakit kaybı olmaksızın yapabilmesi sağlanmaktadır. MiKaS, sisteme giriş yapmış olan öğrencinin harç ödeme bilgilerinde eğer bir eksiklik varsa, yatırmadığı miktarı ve bu miktarların türünü (harç, kayıt gideri gibi...) bu öğrenciye bildirmektedir. Harç yatırma işlemi MiKaS tarafından doğrulanır ise öğrencinin kaydı yenilenmektedir (Şekil 1.1-2.b). İstenirse bu aşamadan sonra öğrenci ders kaydı işlemlerine geçebilmekte veya daha sonra ders kaydı yapmak için sistemden çıkabilmektedir. Yani ders kaydı işlemi, kayıt yenileme işlemine bağımlıdır, fakat eş zamanlı sürdürülmesi zorunlu değildir. Gerekli ise öğrenci danışmanı ile bağlantıya geçip bilgi aldıktan sonra ders kaydını daha sonra kayıt süresi içinde İnternetten gerçekleştirebilir. MiKaS’ta diğer sistemlerden farklı olarak her kayıt aşamasında bir ‘sanal danışman’ bölmesi bulunmaktadır. *Sanal danışman*, ders kaydı sırasında dikkat edilmesi gereken (öğrencinin bulunduğu adıma kadar aldığı ders ve kredi bilgilerinin sunulması, not ortalamasına göre adımlar arasında geçişe izin verilmesi veya ders kaydının sonlandırılması, not ortalamasına göre öğrenciye öneriler sunulması gibi) noktaları vurgulamakta ve yönlendirmeler yapmaktadır. MiKaS’ta ders kaydı üç aşamaya ayrılmıştır. Ayrılan her aşamada öğrencinin dersi seçimi ile ilgili bilgilendirmeler yapılarak adımlar arasında ilerleme sağlanmaktadır. Birinci aşama başarısız olunan zorunlu derslerin kaydı aşamasıdır (Şekil 1.1-3.b) . Bu aşamada sistem, öğrencinin geçmiş dönemlerde alıp da başarısız olduğu zorunlu dersleri listeler. Başarısız olunan zorunlu derslerin kredi toplamı öğrencinin alabileceği en yüksek krediden az ise bu dersler sistem tarafından seçilmiş olarak listelenir. Böylece öğrencinin başarısız olduğu zorunlu derslerin kaydı aşaması bitmiş olur. İkinci aşama, mevcut dönemdeki eğitim-öğretim programından zorunlu derslerin kaydı aşamasıdır (Şekil 1.1-4.b). Zorunlu derslerin kaydından sonra seçmeli derslerin kaydı

aşamasına geçilir. Bu aşamadan sonra ders kaydı işlemleri bitirilmiş olur (Şekil 1.1-5b).

Tüm ders kaydı aşamalarında bölüm ve programlarda tanımlı kurallar kontrollü olarak yapılarak ders kayıtları gerçekleştirilir. Sonradan kural veya kontroller sisteme eklenebilmektedir.

Bazı üniversitelerde kullanılan sayısal kayıt sistemlerinde, ders kaydı aşamasında dersler öğrenciye tek bir liste şeklinde verilmektedir (Adar N.,2005;Akgöbek Ö.,2005;Yiğit B.,2005). Bu listede seçmeli derslerin hangi şartlar doğrultusunda listeye alındığı bildirilmediği için, öğrenci ders seçimi konusunda kararsız kalmaktadır. MiKaS'da öğrenciye hangi dersin neden ve nereden verildiği, verilemiyorsa hangi sebeple verilemediği gibi detaylı açıklamalar sunulmaktadır. Bunu MiKaS'da ders kaydının aşamalara ayrılmış olması ve bu sayede öğrenci ile etkileşimli olarak üniversite yönetmeliklerindeki kuralların kontrolü eşliğinde seçimlerin yönlendirilmesi sağlamaktadır.

MiKaS'ta yapılan tüm işlemler sisteme gerçek zamanlı kaydedilir, geçici yapılarda tutulmaz. Yapılan tüm kayıt hareketleri sistemde saklanır. Böylece gerektiğinde işlemler incelenebilmektedir. Danışmanlar, danışman oldukları öğrencilerin kayıt hareketlerini ve bilgisini eş zamanlı görebilmekte ve gerekirse müdahalede bulunabilmektedirler. Bazı üniversitelerde kullanılan sayısal kayıt sistemlerinde, İnternet üzerinden yapılan kayıt işlemleri geçici tablolarda saklanmaktadır (Adar N.,2005;Akgöbek Ö.,2005; Bingöl A.,2005; Çakar F.,2005; Yiğit B.,2005; ). Bu işleyiş, danışmanların İnternet üzerinden yapılan kayıtları takip edememesine, derslerin kontenjan takibinin yapılamamasına ve bir dizi ek sürecin sisteme fazladan eklenmesine sebep olmaktadır (geçici kayıt dosyasının ana tablolara aktarılması, kontrollerin sonradan yapılması, kayıt esnasında karşılaşılan ve eş zamanlı çalışıldığında düzeltilebilecek meselelerin ana sisteme giriş yapılana kadar ertelenmesi gibi).

El ile kayıt sisteminde bazı danışmanların kayıt işlemlerini kâğıt üzerinde geçici olarak yaptıkları görülmüştür (Muğla Üniversitesi Öğrenci İşleri Daire Başkanlığı,2005). Bu, ders kontenjanlarında yanılığa yol açmakta daha sonradan öğrencilerin aldıkları dersleri değiştirmek zorunda kaldıkları sonucunu ortaya

koymaktadır. Bazı ders kayıtları yoğunluk nedeniyle unutulmuş kâğıda işlenemeyebilmekte ve bu da eksik ders kayıtlarına yol açmaktadır. MiKaS'taki gerçek zamanlı kayıt bu problemlerin önüne geçmektedir.

MiKaS sayesinde, öğrenciler buldukları yerden üniversiteye gelmeden kayıt yenileme ve ders kaydını gerçekleştirmektedirler. Sadece ders kaydı tarihinin son günü danışmanlarına işlemlerini onaylatmak için gelmektedirler.

Bazı üniversitelerde kullanılan kayıt sistemleri (bazı fakülte veya yüksekokullarda) kısmi olarak çalışmakta (Yiğit B.,2005) ve tüm derslerin kaydına imkân vermemektedir (sadece seçmeli derslerin kaydı gibi...). MiKaS, Muğla Üniversitesinin tüm fakülte ve yüksekokullarında kullanılmakta, zorunlu ve seçmeli derslerin tamamının kaydına imkân vermektedir.

MiKaS'ta, teknik altyapıda kullanılan sunucularda veri güvenliği mevcuttur. Veri tabanı ve uygulama sunucularının her iki grubunda da hem donanıma dayalı hem de uygulamaya dayalı veri güvenliği sağlanmaktadır (Raid5, Clustering). Sistem üzerinde kullanıcı işlemlerinin yoğunlaşması durumuna karşı çözüm olarak sunucularda yük dağılımı yapılmıştır (Load Balancing).

Farklı üniversiteler üzerinde yapılan incelemeler sonucunda kayıt sistemleri üzerinde gruplama yapılmıştır. Kayıt yenileme aşamasında ön koşul olan harç bilgilerinin banka ile üniversite arasında aktarımı incelenen üniversitelerde ortalama günde iki kez yapılmaktadır. Gün içerisinde gecikmeli harç bilgisi aktarımı kayıt yenilemede gecikmeyi ortaya koymaktadır. Harç kontrolünü kayıt tarihi sonunda yapan üniversitelerde ( Dicle Üniversitesi ve Osman Gâzi Üniversitesi), harç yatırmayan öğrenciler de kayıt yenileme veya ders kaydı yapabilmektedirler. Bu üniversiteler, kayıt yenileme döneminin sonunda bankadan toplu liste alıp sistemlerine işlemektedirler. Kayıt yenileme tarihi sonunda İnternet üzerinden kayıt yapan ve harç yatırmayan öğrenciler tespit edilerek ders kayıtları ve kayıt yenileme işlemleri iptal edilmektedir.

Bazı üniversitelerde, İnternet üzerinden kayıt sistemi sadece belirli fakülte veya yüksekokullara açılmıştır. Tüm üniversitede aktif olmayan sistem, derslerdeki kontenjan kontrolünün yapılamamasına yol açmaktadır. MiKaS'ta

fakülte veya yüksekokul ayırımı olmaksızın tüm öğrenciler kayıt yapabilmektedirler.

MiKaS'ta kayıt aşamalarında öğrenciye yol gösteren sanal danışman bulunmaktadır. Bunun yanında yönetmeliklerdeki kurallar ve kontroller gerçekleştirilmektedir. Diğer üniversitelerde böyle bir yapı bulunmamakta, sadece yönetmeliklerde belirlenen kuralların kontrolleri sağlanmaktadır.

Üniversitelerde kullanılan kayıt sistemlerinin çoğunda yapılan kayıt işlemleri geçici yapılarda tutularak kayıt süresi sonunda gerçek sistem tablolarına aktarım yapılmaktadır. Bu yolu izleyen sistemlerde, yapılan kayıtlarda kontenjan kontrolü yapılamamaktadır. Danışmanlar da İnternet üzerinden yapılan kayıtları izleyememektedirler. MiKaS'ta yapılan işlemler gerçek yapılar üzerinde saklanmaktadır.

MiKaS'ta teknik alt yapıda veri güvenliği için clustering ve kullanıcı yoğunluğu artışına karşı load balancing özellikleri kullanılmıştır. İncelenen üniversitelerde bu yapılar kullanılmamaktadır (Adar N.,2005; Akgöbek Ö.,2005; Yiğit B.,2005; Bingöl A.,2005; Çakar F.,2005; Güven A.S.,2005; Kılıçarslan H.G.,2005; Öncü Z.,2005; Özdemir A.,2005; Yıldız b.,2005 ) . Sadece el ile veri güvenliği için yedekleme yapılmaktadır. Kullanıcı yoğunluğuna karşı hiçbir önlem alınmamıştır.

MiKaS, 2003-2004 güz döneminden buyana Muğla Üniversitesinde kullanılmaktadır. Her dönem, öğrencilerin bilgilendirilmesi sayesinde kayıt sisteminin kullanım oranı artmıştır. 2004-2005 Güz döneminde Muğla Üniversitesindeki toplam öğrenci sayısı 14864 iken, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 506, ders kaydı sayısı 1710'dur. 2004-2005 Bahar döneminde, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 1106'ya, ders kaydı sayısı da 3870'e çıkmıştır. Bu rakamlar bir önceki dönemin yaklaşık iki katıdır.

MiKaS teknik alt yapısında iki sunucu türü bulunmaktadır: Veri tabanı sunucuları ve Uygulama sunucuları. Veri tabanı sunucusu olarak iki sunucu makine görev yapmaktadır. Bu sunucularda Microsoft SQL Server 2000 kullanılmaktadır. İki sunucu küme mimarisinde (clustered) çalışmaktadır. Her iki sunucuda ayrıca, depolama ünitesinde Raid 5 erişim şekli ile donanımsal veri

kayıbı güvenliđi sađlanmıřtır. Uygulama sunucusu olarak da yine kümeleme mimarisinde iki sunucu görev yapmaktadır. Uygulama sunucularında Bea Weblogic Application Server 8.1 kullanılmaktadır. Bea Weblogic Application Server, üç katmanlı (three tiered) yazılım uygulama geliştirme modeli içinde orta katmanda görev yapmaktadır. İlk katman veri tabanı sunucuları, üçüncü katman son kullanıcı uygulamalarıdır. Bu sunucular da depolama ünitesinde Raid 5 erişim şekli ile donanımsal veri kaybı güvenliđi kullanmaktadırlar. Ek olarak, sunucularda her türlü donanımsal arızaların, ađ bağlantısının kesilmesi, güç kaynađının arızalanması vb. arızalara karşı donanımın her hayati parçası paralel çiftler halinde kullanılmıřtır.

Tez akışı içerisinde öncelikle programlamada kullanılan Java ve JavaScript teknolojisi tanıtılmıřtır. Java ile ilişkili kullanılan alt teknolojiler ve özellikleri açıklanmıřtır. Materyal ve Yöntem bölümünde MiKaS geliştirilirken izlenen yol ve kullanılan araçlara değinilmıřtir. Dördüncü bölümde de geliştirilen kayıt sisteminin işleyiş aşamaları açıklanmıřtır.

## 2. KAYNAK ÖZETLERİ

MiKaS geliştirilmesi sırasında, öğrenci kayıt sistemi kullanan tüm üniversitelerin ilgili birimleri ile telefon veya elektronik posta aracılığı ile bağlantıya geçilmeye çalışılmıştır. Bağlantı kurulabilen üniversitelerle kullandıkları kayıt sistemleri hakkında bilgi edinilmiştir (Adar N.,2005; Akgöbek Ö.,2005; Yiğit B.,2005; Bingöl A.,2005; Çakar F.,2005; Güven A.S.,2005; Kılıçarslan H.G.,2005; Öncü Z.,2005; Özdemir A.,2005; Yıldız b.,2005 ). Elde edilen bulgular ve analizler sonucu MiKaS altyapısı geliştirilmiştir.

Program geliştirme aşamasında kullanılan Java teknolojisi, üreticisi olan Sun Microsystems'in Java web sayfası (Java Technology,2005) başta olmak üzere diğer Java teknoloji web sayfaları kaynak olarak kullanılmıştır (BEA Product Documentation, 2005;Chan P.,2005; Public CERN Java Infrastructure, 2005).

Veritabanı geliştirme ve kurulumu aşamasında kullanılan SQL Sever'in üreticisi Microsoft Corporation'un SQL Server web adresi kullanılmıştır (SQL Server Home,2005) .

### 3. MATERYAL VE YÖNTEM

Kayıt sistemin geliştirilmesinde öncelikle Muğla Üniversitesi'nde iki dönemde ders kayıtları incelenerek kayıt başlamadan önce ve kayıt sırasında yapılan işlemler belirlenmiştir. Danışmanlar ve öğrenci işleri birimleri ile bağlantı kurularak el ile işleyen kayıt sistemindeki aksaklıklar tespit edilmeye çalışılmış ve meselelere çözüm önerileri alınmıştır.

Kayıt yapısını sayısal ortamda gerçekleştiren (kaynaklarda belirtilen) üniversitelerin ilgili birimleriyle bağlantı kurularak kullanılan sistemler incelenmiştir. Kullanılan öğrenci internet kayıt sistemlerindeki kayıt yenileme ve ders kaydı aşamaları ve işleyişleri, varsa eksiklikleri hakkında bilgi alınmıştır.

El ile yapılan kayıtlardaki aşamalar tespit edilerek "MiKaS kayıt işleyiş aşamaları" ortaya konmuştur. Kayıt aşamaları, üniversite öğrenci işleri ile koordinasyonlu bir şekilde bağlantı kurularak sayısal ortama aktarılacak şekilde adımlara bölünmüştür.

İnceleme aşaması tamamlandıktan sonra İnternet tabanlı öğrenci kayıt sistemi geliştirilmeye başlanmıştır. Geliştirme sırasında mevcut problemlerin çözümü ve işlemlerin hızlandırılabilmesi esas alınarak ilerlenmiştir.

İnternet tabanlı kayıt sistemi yapısı Java dili ile geliştirilmiştir. Geliştirme sırasında Java kodların yazımı ve kontrolü için Borland Jbuilder Java geliştirme programı kullanılmıştır. Jbuilder mevcut Java teknolojilerinin tamamının kullanılabilmesine olanak veren bir ara programdır. Kodların denetim ve hata kontrolü yine jbuilder ile yapılmıştır.

Web sayfalarının tasarım ve biçimlendirilmesinde Macromedia Dreamweaver MX kullanılmıştır. Dreamweaver'da ayrıca kod düzeyinde hazır veritabanı bağlantı desteği kullanılmıştır.

Sunucuda kullanılan veritabanı yönetim sistemi SQL Server 2000'dir. SQL Server 2000'in avantajlarından birisi sunduğu ölçeklenebilirlik ve güvenilirlik özellikleridir. SQL Server 2000, bu özellikleri sayesinde e-ticaret ve iş uygulamalarının gerektirdiği esnekliği sağlayabilmektedir. SQL Server 2000,

simetrik çok işlemcili sistemlerin sunduğu avantajları da kullanarak 32 işlemciye ve 64 GB RAM'e kadar ölçeklenebilmektedir. Ayrıca veri tabanının ve veri yükünün farklı sunuculara dağıtılması da SQL Server 2000'in önemli bir özelliğidir. Bunun yanında, SQL Server 2000 kümeleme, kayıt ve yedekleme stratejileri ile yüksek oranda süreklilik sağlamaktadır. SQL Server 2000, eşzamanlı yedekleme, tamamen bütünleşmiş kayıt ve gelişmiş kümeleme özellikleri ile oldukça yüksek seviyede süreklilik sağlamaktadır. SQL Server 2000, veri tabanı eşzamanlı halde iken ve erişime açıkken yedekleme yapmayı mümkün kılmaktadır. SQL Server 2000, yüksek yoğunluktaki veritabanı işlemlerini yönetebilecek güçte olduğu için seçim yapılmıştır. Bunun yanında farklı veritabanı sunucuları da kayıt sistemi yapısında kullanılabilir.

Tablo 3.1 ve 3.2'de kullanılan sunucuların teknik özellikleri verilmiştir. Bu donanım yapısına sahip sunucularla aynı anda 60 kişinin isteğine cevap verilebilecek bir sistem ortaya konmuştur. Kullanıcı sayısının artması durumunda donanım yapılarında değişikliklerle bu hizmet kapasitesi artırılabilir.

Tablo 3.1 Web sunucusu donanım özellikleri

Bileşen	Özellikleri
Sistem platformu	Windows 2003 Enterprise Server
Hard disk alanı	2x36 MB Hard Disk
Hafıza	1 GB RAM
İşlemci	2 x 3.2 Intel Xeon CPU

Tablo 3.2 Veri Tabanı sunucusu donanım özellikleri

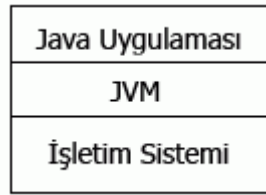
Bileşen	Özellikleri
Sistem platformu	Windows 2003 Enterprise Server
Hard disk alanı	2x36 MB Hard Disk
Hafıza	2 GB RAM
İşlemci	2 x 3.2 Intel Xeon CPU



## 4.JAVA TEKNOLOJİSİ

### 4.1 Java Teknolojisi Yapısı

Java, Sun Microsystems firması tarafından 1991 yılında geliştirilmiş nesneye yönelik bir programlama dilidir. En önemli özelliği platform bağımsız olmasıdır. Java ile yazılmış programlar, işletim sistemi ne olursa olsun farklı bilgisayar sistemlerinde kullanılabilir. Bu yapıyı sağlayan ise farklı platformlara göre yazılmış olan Java sanal makinesi (Java Virtual Machine - JVM) yapısıdır. JVM, yazılmış olan uygulamayı işletim sistemi farkını kullanıcıya hissettirmeden çalıştırma işlemini gerçekleştiren yapıdır. Şekil 4.1’de Java uygulaması ile işletim sistemi ilişkisi verilmiştir.



Şekil 4.1 İşletim Sistemleri ve JVM konumu

Java, şu anda kullanıldığı sektörler açısından çok geniş bir kapsama sahiptir. Grafikten telefona, şifrelemeden web yazılımlarına kadar birçok alanda kabul gören teknoloji, bu geniş çerçevede içerisinde en başarılı atılımlarından birini büyük şirket yazılımları alanında yapmıştır. Java şirket yazılım platformu J2EE (Java 2 Enterprise Edition)’dir.

Java teknolojisinin kaynak kodu açıktır. Bu sayede gelişen teknolojiye rahatlıkla ayak uydurabilmektedir. Belirli kurallara uyulmak kaydıyla Java teknolojisini isteyen kişi veya kurumlar geliştirip değişiklik yapabilirler.

İnternet tabanlı olup veri tabanlarına bağlantılar yapabilen Java, birçok diğer programlama dilinden daha uygun özelliklere sahiptir. Servlet, JSP, JDBC, EJB gibi Java teknolojileri bu tür uygulamalar yazmada büyük kolaylıklar sağlamaktadır.

Gerek Java programlama dili, gerekse bu dile bağılı alt teknolojiler, Visual Basic veya Borland Delphi gibi sadece belirli bir firma tarafından geliştirilmiş ürünler değildir. Sun Microsystems, JVM veya Java programlama diline bağılı alt teknolojiler yazmak için belirli kurallar koymuştur; bu kurallar topluluğuna “belirtiler” denir. Alt teknoloji geliştirecek olan her kurum bu belirtiler çerçevesinde hareket etmek zorundadır. Bu belirtilere sadık kalan her yazılım firması Java sanal makinesi veya Java programlama diline bağılı alt teknolojiler yazabilir (örneğin Application Server - Uygulama Sunucusu). Eğer bu belirtilere sadık kalınmayıp standart dışı bir JVM veya Java programlama diline bağılı alt teknolojiler yazılmaya kalkılırsa hukuki bir suç işlenmiş olur. Java bu özelliğiyle de gelişmeye açık bir teknolojidir (Java Technology, 2005).

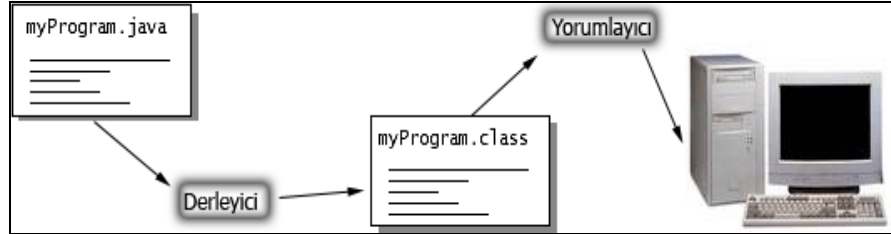
Java teknolojisinin diğler önemli bir özelliği de “çöp toplama sistemi (garbage collector)”dir. Bu yapı da belirtiler arasındadır. Bu yapı sayesinde programcı program kodları arasında çok fazla yer tutan kontrol ve düzenleme işlemlerinin çoğunu takip etmekten kurtulur. Çöp toplama sistemi daha önceden oluşturulmuş, ancak şu an için kullanılmayan ve bellekte boşu boşuna yer işgal eden nesnelere belirleyerek otomatik olarak siler. Böylece bellek yönetim (memory management) yükü tasarımcıdan JVM’e geçmiş olur. Diğler dillerde, örneğin C++ programlama dilinde, oluşturulan nesnelere yok edilme sorumluluğu programcıya aittir. Unutulduğunda veya doğru yapılmadığında ise programın kilitlenmesi hatta sistem kilitlenmesi ile karşılaşılabilir (Java Technology, 2005).

Çöp toplayıcının ne zaman ortaya çıkıp temizleme yapacağı belirli değildir. Eğer bellekte JVM için ayrılan kısım dolmaya başlamışsa çöp toplayıcı devreye girerek kullanılmayan nesnelere bellekten siler.

#### **4.2 Java Kodlarının Çalışma Şekli**

Klasik programlama dillerinde yazılan kod ya derlenir, ya da yorumlayıcı aracılığı ile bilgisayar ortamında çalıştırılır. Java programları ise her iki işlemi de içeren aşamalardan sonra çalıştırılır. Yazılan kodlar, öncelikle derleyici aracılığı ile derlenir ve JVM’in anlayabileceği kodlara dönüştürülür. Bu kodlara

“*bytecode*” adı verilir. Bytecode yapısını anlayan ise yorumlayıcıdır. Her farklı platforma göre ayrı Java yorumlayıcısı olduğu için yazılan kod her platformda çalışabilir. Şekil 4.2’de Java kodları çalışma aşaması verilmiştir.



Şekil 4.2 Java kodlarının çalışma aşaması

Java bir programlama dilinin ötesinde bir sistem platformudur. Sistem platformu donanımsal veya programlanabilir yapılardır. Sistem platformu yapısı Java teknolojisinin işletim sistemi ile kullanıcı arasında durması ile sağlanmaktadır (Eckel B., 1998). Böylece işletim sistemi farkından kullanıcı veya yazılmış program etkilenmemektedir. Java Platformunun iki önemli bileşeni vardır:

- Java Sanal makinesi (Java Virtual Machine - JVM)
- Java Uygulama Yazılım Arabirimi (Java Application Programming Interface- Java API)

JVM Java platformunun temelidir. Farklı donanım yapılarına karşı farklı JVM’ler bulunur. Java API ise içinde çok geniş hazır fonksiyon ve bileşenleri içeren bir yapıdır. İçinde ilişkili fonksiyon ve sınıfları barındıran birçok paket (package) barındırmaktadır. Derlemeden sonra oluşan yeni yapıya “*Native Code*” denir. Bu kodlar her platformda çalışabilir haldedir. Java uygulamaları birçok farklı platform üzerinde çalışabildikleri gibi birçok farklı uygulama alanında da görülebilmektedir (Java Technology, 2005).

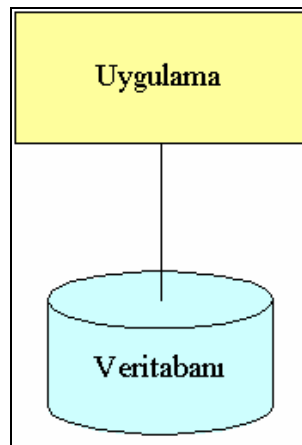
Java sanal makinesi sıcak nokta (Hotspot) adı verilen özel bir teknolojiyi içinde barındırır. Hotspot, bir yazılımda sürekli olarak tekrarlanan ve üzerinden geçilen kod bölümlerine verilen bir isimdir. Java sanal makinesi bir kod çalışmaya başladıktan sonra sıklıkla kullanılan kod bloklarını gözler ve bir süre

sonra bu bytecode bloklarının çalışılan sistemdeki gerçek işlemci komut karşılıklarını bir tür cep belleğe yazar ve zaman ilerledikçe artık byte kod üzerinden değil doğrudan sistemin öz komutlarını kullanarak yazılımın o bölümlerini işletmeye başlar. Bu şekilde ciddi performans avantajı sağlandığı belirtilmektedir (Java Technology, 2005). Yakın zamana kadar, şirket bazındaki yazılımlarda en çok kullanılan teknolojiler istemci-sunucu (client-server) alanında iken; bu eğilim her geçen gün, dağıtık sistemlerin sundukları avantajlar sebebiyle çok katlı (n-tier ya da multi-tier) mimarilere doğru kaymaktadır.

### 4.3 Yazılım Mimarileri ve Java'nın yeri

#### 4.3.1 İstemci-sunucu mimarisi

Klasik istemci-sunucu sistemler iki katlı (2-tier) mimariler üzerine kurulmuşlardır. Bu sistemlerde, uygulama doğrudan veritabanı sunucusuyla bağlantı içindedir. Gerçekleştirilen işin büyük bir bölümü istemci tarafından yapılır, çoğu zaman sunucu sadece veritabanı sunucusu görevi yapar. Bu, uygulamanın hızla çalışması için istemci donanımı önem kazanmaktadır ve ağ olanaklarının yoğun kullanımı ortaya çıkar. Çünkü iş mantığının işlenmesinin büyük bir bölümü istemci tarafından yapılmaktadır. Uygulama, her gerekli veri parçası için, veritabanına bağlanmaktadır. İş mantığının istemci yerine veritabanına aktarılması ise veritabanına özel, yeniden kullanılması zor olan bir uygulama ortaya koyar. Şekil 4.3'te iki katlı mimari yapısı görülmektedir.



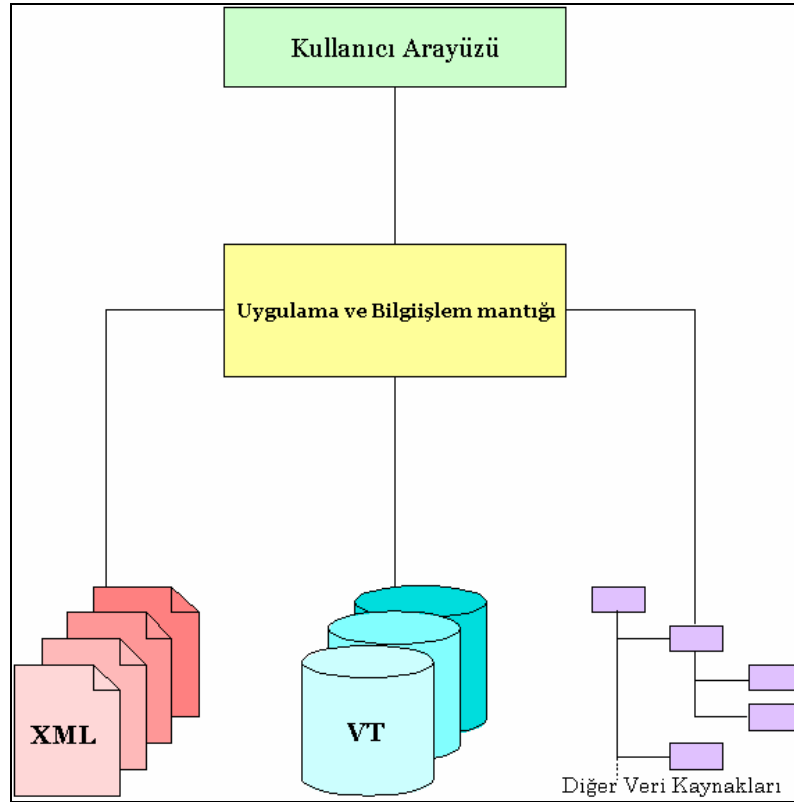
Şekil 4.3 İki katlı mimari (Erkan Y., Ocak 2005)

İki katlı sistemlerde ortaya sıkça çıkan bir problem de, uygulamanın bakımı ve yapılan değişikliklerdir. İş mantığında yapılmak istenen küçük bir değişiklik bile tüm istemci bilgisayarlardaki programların yeniden kurulmasını gerektirir. Bu işlemin otomatik hale getirilmesi bile her bilgisayarın güncelleştirilmiş programı alabilmesini sağlamaz. Yeni uygulamalara alışmak hem zaman hem de ek çalışma gerektirdiğinden, genelde kullanıcılar yeni uygulamaları hoş karşılamazlar ya da işleri gereği buna hazır olmayabilirler. Ayrıca değişiklikler sonucu ortaya çıkabilecek sistemdeki ayarların yapılması ve değişikliklerin uygulanması da kullanıcılar açısından zor ve ek bir işlemdir. Fakat bu işlemler yapılmadan da güncellenen programların çalışabilmesi söz konusu değildir.

İstemci-sunucu sistemlerinin büyük bir dezavantajı da, uygulamanın kullanımının artması halinde (kullanıcı sayısının artması halinde), sistemin ölçeklenirliğinin artmamasıdır. Veritabanı sunucusunun donanım kapasitesi dolduğu zaman tek çözüm, mevcut sunucuyu daha güçlü bir sunucuyla değiştirmektir. Bu da maliyetli bir işlemdir.

#### **4.3.2 Üç katlı mimari**

İki katlı mimarilerdeki problemlere karşı, yazılım endüstrisi üç katlı mimari kavramını geliştirmiştir. Şekil 4.4'de üç katlı mimari verilmiştir.



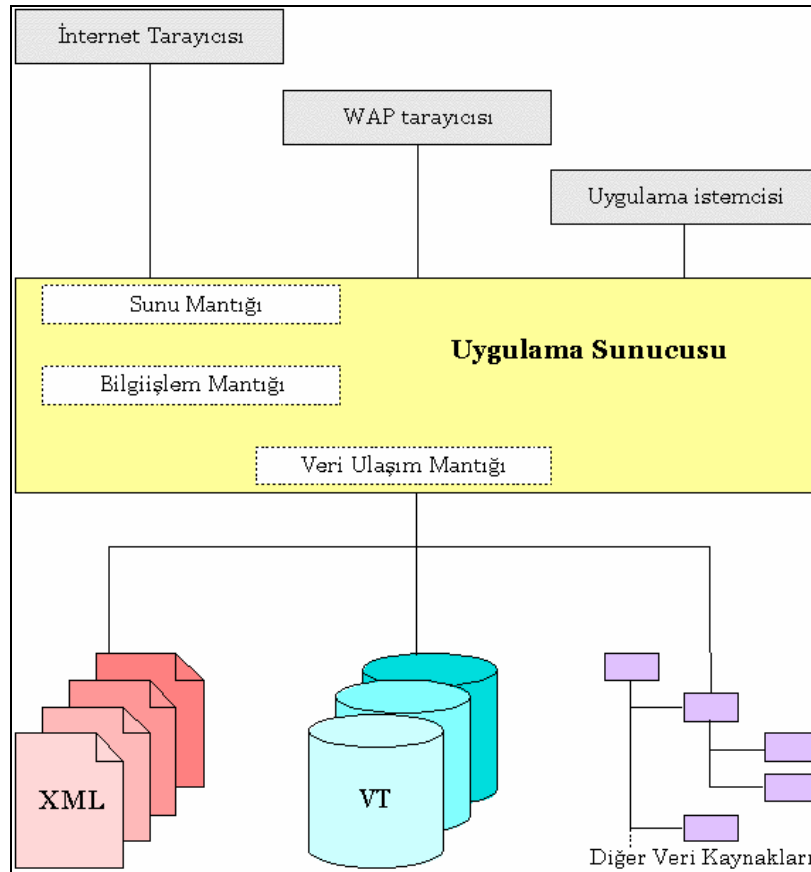
Şekil 4.4 Üç katlı mimari (Erkan Y., Ocak 2005)

Üç katlı mimariler uygulama görevlerine ve sorumluluklarına göre üç bölüme ayrılır. Genelde bu ayırım fiziksel, somut bir ayırımdır. İlk bölüm sunuş katı, kullanıcı arayüzünden oluşur. İkinci bölüm iş katı ya da uygulama katı, uygulamanın işlem mantığını içerir. Son bölüm veri katı, uygulamanın ihtiyacı olan veriyi sunan kattır.

Sunuş katı ile uygulama katının ayrılması ile verinin işlenişi sunuluşundan ayırır. Böylece veri işlem mantığı değiştirilmeden uygulamaya değişik kullanıcı arabirimleri eklenebilir. İnternet kullanımını buna örnek verebiliriz. Bilgisayarlardaki İnternet tarayıcılarıyla ulaşılabilen milyonlarca sayfa, artık taşınabilir küçük elektronik araçlarla ulaşılabılır hale gelmiştir. Bunun yanında şirket elemanlarının şirket dışı kullanımı için geliştirilen ve genelde diz üstü bilgisayarlarda çalışan uygulamalar artık elde taşınabilen farklı elektronik araçlarla da ulaşılabılır durumdadır.

### 4.3.3 Çok katlı mimari

Uygulamalarda görev ve sorumluluk ayırımı biraz daha geliştirilirse, çok katlı (n-tier ya da multi-tier) mimari adı verilen kavram elde edilir. J2EE teknolojisi bu tür mimariler üzerine kurulmuştur. Bu tür mimarilerin avantajı, farklı teknolojilerden ve farklı ortamlardan yararlanan yazılımlar kullanan şirketlerde hissedilir. Bu şirketler, çoğu zaman uygulama sunucularının yardımıyla, mevcut uygulamalarını sorunsuz olarak yeni geliştirilenlerle beraber kullanma olanağı bulurlar. Böylece büyük yatırımlar yapılmadan talepler gerçekleştirilebilir. Şekil 4.5'te çok katlı mimari yapısı görülmektedir.



Şekil 4.5 Çok katlı mimari (Erkan Y., Ocak 2005)

Çok katlı mimariler birden fazla bölümden oluşabilirler. İki ve üç katlı mimarilerde katlar genel olarak somut katları gösterirken, çok katlı mimarilerde bunlar soyut katlar haline gelir ve genelde bu dağılım katları değişik tür görevlere göre yapılır. Genel olarak, J2EE uygulama mimarileri beş ana kattan oluşur (Erkan Y., Ocak 2005) :

- *İstemci Katı (Client Tier)*: Bu kat, geliştirilen uygulamaya ya da sisteme bağlanan diğer uygulamalar ya da cihazlardan oluşur. Örneğin: İnternet tarayıcısı, Java applet, WAP telefon...
- *Sunuş Katı (Presentation Tier)*: Bu kat, istemciler için gerekli olan her türlü sunuş mantığını içerir. Uygulamaya bağlanan istemcilerin taleplerini kaydeder, gerekli iş mantığının uygulanmasını sağlar, talebin işlenmesi sonucu ortaya çıkan veriyi sunulur hale getirip istemciye cevap yollar. J2EE'yi oluşturan teknolojilerden ikisi JSP (JavaServer Pages) ve Java Servlet bu katta bulunur.
- *Uygulama ya da İş Katı (Application/Business Tier)*: İşe dayalı tüm bilgi işlem bu katta toplanır.
- *Entegrasyon Katı (Integration Tier)*: Bu kat, uygulamanın görevini yerine getirmesi için gerekli olan sistem dışı yazılımlara, sistemlere ya da veri tabanlarına bağlantıları sağlamakla yükümlüdür. J2EE uygulamalarının bu bölümleri, genelde, JDBC (Java Data Base Connectivity), J2EE Bağlayıcı (Connector) ya da bağlantı kurulan yazılımlara özel arayüzleri kullanırlar.
- *Kaynak Katı (Resource Tier)*: Uygulama için gerekli veriler ve dış servisler bu katı oluşturur.

Çok katlı bir uygulamayı oluşturan katlar soyut kavramlardır. Bu katları uygulamanın ihtiyaçlarına göre düzenlemek ve daha da detaylı hale getirmek mümkündür. J2EE, son olarak açıklanan "çok katlı" mimarı ve onunla beraber gelen tasarımlar üzerine kurulmuş yeni, güçlü ve çabuk kabul gören bir teknolojidir.



#### 4.4 Servletler

Java Servlet teknolojisi sunucu katında çok kullanılan bir teknolojidir. Kullanıcı tarafından sunucuya gönderilen her istem için yeni bir süreç yaratmak yerine, her istemi aynı JVM içinde hafif bir şekilde işlemek çok daha hızlı bir yoldur. Java Servlet'ler HTTP sunucularının durağan doğasını dinamikleştirmek için geliştirilen sunucu ekleridir. Veri tabanından istem anında verileri çekerek kullanıcıya dinamik bir içerik sunarlar. Servletler, Java sınıfının özelliklerine sahip olduklarından hizmetleri sadece İnternet teknolojileriyle sınırlı değildir. Java Servlet'ler sayesinde elektronik ileti göndermek, veri tabanı uygulamaları gerçekleştirmek ya da çoklu ortam işlemleri yapmak mümkündür.

#### 4.5 Java İsimlendirme ve Rehber Arayüzü (Java Naming And Directory Interface-JNDI)

JNDI, rehberlerin verilerine hızlı erişim sağlayan özel bir veritabanı çeşididir. Farklı sunucularda çalışan J2EE uygulamalarının kaynak paylaşımı ve haberleşmesi bu teknoloji sayesinde yapılır. Ortak kaynakların düzenlenmesi ve kullanım şekli bu yapı sayesinde kontrol edilir.

#### 4.6 Java Veritabanı Bağı (Java Database Connectivity-JDBC)

JDBC (Java Database Connectivity), ilişkisel veritabanlarıyla standart bir şekilde bağlantıyı sağlayarak SQL komutlarını uygulamayı sağlar. JDBC sürücüleri esneklik sağlarlar. Bu sürücüler sayesinde dikkatli yazılan Java kodları farklı veritabanı yönetim sistemlerinde kurulmuş aynı modelle değişiklik gerektirmeden çalışabilir. Kullanılan dört farklı JDBC sürücüsü vardır (Types of JDBC technology drivers, 2005):

- *JDBC-ODBC köprüsü*: Windows platformunda en çok kullanılan ve Microsoft tarafından geliştirilmiş olan ODBC (Open Database Connectivity) standart SQL işlemleri için kullanılan bir API'dir. ODBC daha sonra Visigenic ve Intersol tarafından Windows dışındaki bazı platformlar içinde geliştirilmiştir. Bu tür sürücüler, gelen JDBC istemlerini ODBC komutlarına çevirirler ve geri gelen cevapları tekrar JDBC ve Java nesnelere çevirerek

istemciye sunarlar. Bu tür sürücülerde her istemciye platforma bağlı parça için kurulum yapmak gerekir.

- *Java ve Platforma bağlı teknoloji karışımı:* Bu tür sürücüler Java, veritabanı ve platforma özel teknolojiler kullanırlar. Bu tür sürücülerde de her istemciye bir kurulum yapmak gerekir.
- *Arakat Veritabanı Erişim Sunucusu:* En esnek mimariye sahip olan 3. tür JDBC sürücüler, ortakat (middleware) yazılımlarından destek alırlar. Bu tür sürücüler birden fazla istemciyi birden fazla veritabanı sunucusuna bağlayabilirler.
- *Saf Java JDBC sürücüsü:* İkinci tür sürücülere alternatif olarak geliştirilmişlerdir. En büyük avantajları tamamen Java yazılımları olup istemci tarafında bir kurulum yapmayı gerektirmemeleridir. Bu sürücüler, veritabanına açılan doğrudan soketler aracılığıyla, veritabanına özel protokolleri kullanarak veri iletişimi sağlarlar.

#### **4.7 Java Elektronik Posta (Java Mail)**

Javamail, elektronik ileti alışverişini Java aracılığı ile programlamayı sağlar. Elektronik ileti konusunda İnternet üzerinde kullanılan birden fazla protokol ve teknoloji vardır. Javamail bu teknolojilerden bağımsız olarak hazırlanmış, değişik protokollere aynı şekilde erişimi sağlayan bir kurallar topluluğudur. Javamail aşağıda protokolleri desteklemektedir:

- SMTP (Simple Mail Trasfer Protocol)
- IMAP (Internet Message Access Protocol)
- MIME (Multipurpose Internet Mail Extensions)
- POP3 (Post Office Protocol 3)

#### 4.8 Java İleti Hizmeti (Java Messaging Service-JMS)

MOM (Message Oriented Middleware - Mesaja Yönelik Ortakat Yazılımları) 'lar farklı sistemlerin ve uygulamaların birbirleriyle eşzamansız mesajlar aracılığıyla iletişim kurmasında kullanılırlar. Farklı yapılar MOM uygulamaları kullanmışlardır. Her mesajın bir üreticisi ve bir veya birden fazla kullanıcısı vardır. Böyle farklı sistemler aynı ortamlarda çalışmasalar bile birbirleri ile haberleşebilmektedirler (Java Technology, 2005).

#### 4.9 EJB (Enterprise Java Beans)

J2EE'nin en çok söz ettiren teknolojisi EJB (Enterprise JavaBeans)'dir. Bağımsız yapılacak her farklı iş için EJB oluşturulup farklı hizmetlerde kullanılabilirler. Bu hizmetler uygulama geliştirme zamanını kısalttığı gibi, yazılımı da kolaylaştırır. Bu hizmetlere örnek olarak güvenlik (security), işlem bütünlüğü (transaction), eş zamanlılık (concurrency), ölçeklenebilirlik (scalability), veri sürekliliği (persistence) vb. verilebilir.

EJB teknolojisi ve Javabeans teknolojilerinin benzerliği sadece isimlerinden kaynaklanmaktadır. Aralarındaki tek ortak nokta ikisinin de bileşen belirten teknolojiler olmasıdır. Javabeans mimarisi yazılım alanında genel kullanım için oluşturulmuş bileşen mimarisidir. Enterprise Javabeans bileşenleri ise çok özel kullanımı olan ve iş mantığı taşıyan bileşenlerdir. Bir EJB bileşeni, özel kurallara göre düzenlenmiş Java sınıf dosyalarından ve bileşenin özelliklerini ve kullanım kurallarını içeren bir XML dosyasından oluşur (Public CERN Java Infrastructure, 2005).

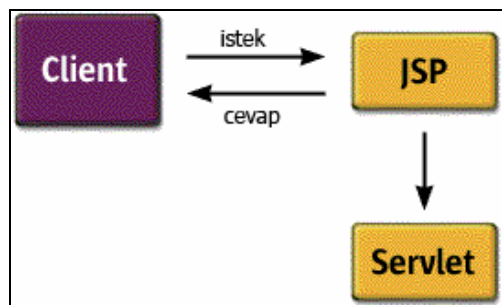
#### 4.10 Java Applet

Applet, uzaktaki sistem üzerinden indirilip İnternet tarayıcı üzerinde çalıştırılabilen Java uygulamalarına verilen isimdir. Appletler web sayfaları içinde kullanılabilen Java uygulamalarıdır. Appletler gerekli parametrelerle birlikte html yapısı içine eklenebilirler ve web sayfası içinde başlı başına bir program gibi çalıştırılırlar. Çalıştırılabilmeleri için kullanılan tarayıcıda applet

çalıştırma izninin açık olması gereklidir. Javanın son kullanıcılar tarafından tanınması applet sayesinde olmuştur. İçerisinde applet olan bir sayfayı açmaya çalışıldığında tarayıcı otomatik olarak Java sanal makinesini çalıştırıp ekranın applete ayrılan bölümünde uygulamanın çalışmasını sağlar.

#### 4.11 Java Sunucu Sayfaları (Java Server Pages-JSP)

Java Server Page (JSP) teknolojisi farklı web-sunucularda çalışabilen ve dinamik içerikli web sayfaları oluşturabilmeye olanak sağlayan bir Java teknolojisidir. JSP normal HTML sayfalarını temel alan bir yapısı vardır ve içinde JSP script kodları bulunan uzantıları “jsp” olan HTML sayfalarıdır. JSP script kodları genelde Java ile yazılmakla birlikte gerektiğinde Java script ile de yazılabilirler. JSP teknolojisi, kullanıcıdan gelecek olan bilgilere göre dinamik olarak HTML kodları üretip tekrar kullanıcıya yönlendirir. Şekil 2.6’da JSP Teknolojisinin çalışma prensibi görülmektedir.

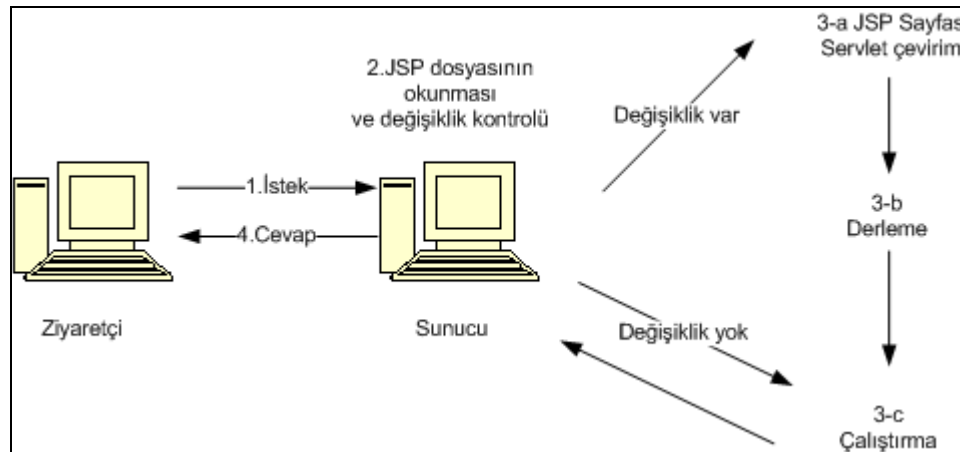


Şekil 4.6 JSP Teknolojisi çalışma prensibi

##### 4.11.1 JSP sayfalarının çalışma şekli

JSP sayfaların yazımı tamamlandıktan sonra, bir JSP sunucusuna sayfalar kopyalanıp belirli tanımlamalar yapıldıktan sonra hazır hale getirilir. Bundan sonra ziyaretçi, jsp uzantılı sayfaya istekte bulunduğu anda, sunucu sayfa içindeki “<%” ve “%>” takıları arasındaki Java kodlarını yorumlayarak jsp sayfasını servlet karşılığına çevirir. *Java* uzantılı *servlet* kaynak dosya derlenip *class*

uzantılı *Java* dosyasına dönüştürülür. Bu dosya çalıştırılarak elde edilen HTML kodu ziyaretçiye gönderilir. Böylece ziyaretçi bilgisayarında, saf HTML kodlarından oluşan *jsp* uzantılı web sayfası görüntülenmiş olur. JSP sayfalarının oluşturulma aşamaları Şekil 4.7’de görülmektedir



Şekil 4.7 JSP kodlarının çalıştırılma işlemi

Web sunucuda bir jsp sayfasının servlete çevrilip derlenmesi işlemi sadece bir kez gerçekleşir. Bu sayede web sunucuya aynı JSP sayfası için istek geldiğinde jsp sayfası tekrar servlete çevrilmek yerine daha önceden çevrilmiş hazır servlet kullanılır. Bu yüzden sayfa için yapılan daha sonraki isteklerde sayfanın çalıştırılması ilkinde göre daha hızlı olur. Jsp sayfasında yapılan herhangi bir değişiklikten sonraki ilk istekte yine servlete çevirme işlemi gerçekleştirilir.

#### 4.11.2 JSP'nin avantajları

JSP'de statik HTML kodlarla dinamik içeriği oluşturan kodlar iç içedir. Web sayfalarının, bilindiği gibi görsel yapı ile kod yapılarını barındıran içerikleri vardır. JSP ile yapılan büyük ölçekli projelerde web tasarımcıları ile web geliştiricileri birbirlerinden bağımsız olarak çalışabilmektedirler. Bu sayede her geliştirici kendi alanı ile ilgili işlemlere odaklanabilmektedir. Web sayfalarının üç önemli geliştirme katmanı vardır:

- Görsel tasarım
- İstemci taraflı programlama (Client-side programming)
- Sunucu taraflı programlama (Server-side programming)

Bu üç katman birbirinden bağımsız olarak farklı geliştirme ortamlarında geliştirilir ve tek bir HTML dosyası elde edilir. Web tasarımcısı tasarımını, kullanıcı-taraflı web geliştirici ilgili scriptlerini, sunucu taraflı uygulama yazılımcısı da JSP kodlarını rahatlıkla ve bağımsız bir şekilde oluşturabilir.

JSP, Java teknolojisinin bir alt teknolojisi olduğu için Java'nın tüm önemli özelliklerinden faydalanabilir. Yani JSP teknolojisi ile nesneye dayalı programlama yapılabilir, farklı platformlarda çalışılabilir, güvenilir ve geliştirilebilir bir teknolojidir. JSP derin programlama bilgisi gerektirmez. Temel Java yapıları ile etkin kodlamalar yapılabilir.

Esneklik özelliğine sahip olan JSP sayesinde, daha sonraki uygulamalarda da kullanılacak bileşen (component) tabanlı yapılar oluşturulabilir. Bu özelliği "Java bean" ya da "servlet" yapıları sağlar.

JSP özgün takıları destekler. Tekrar tekrar aynı kodları kullanmak gerekiyorsa bu kodlar "özgün takı"lar olarak tanımlanabilir. Bu sayede programcı, sayfa içinde özgün takıları kullanılarak uzun kodlar yazmaktan kurtulur.

#### **4.11.3 Java yazılım geliştirme araçları (Java software development kit-JSDK)**

JSDK, Sun Microsystems tarafından program geliştiriciler için hazırlanmış yazılım paketine verilen addır. Java programlarını yazmak, derlemek ve çalıştırabilmek için JSDK'ya ihtiyaç vardır. Bu paket iki nedenden dolayı kurulmalıdır. Birincisi, Java web sunucularının çalışmasını sağlar. İkinci neden ise JSDK, ilerleyen bölümlerde değinilecek olan Javabean ve servlet

bileşenlerinin derlenmesini sağlamaktadır. JSDK, Sun Microsystems'in web sitesinden ücretsiz olarak indirilebilmektedir(<http://java.sun.com/j2se/>).

Farklı Java yazılım geliştirme araçlarının tamamı JSDK'yı kullanmakla birlikte işleyiş olarak ve sundukları kolaylıklar bakımından farklılaşmışlardır. Kaliteli uygulama sunucularının birçoğu bir tümleşik geliştirme ortamı (Integrated Development Environment-IDE) ile doğrudan bağlantılı olarak çalışabilir. Örneğin, Weblogic Workshop veya Jbuilder Weblogic Edition kullanarak, Weblogic için hızlı yazılım geliştirmek mümkündür. Bunun yanında, hemen hemen tüm uygulama sunucuları, yazılım geliştirme sürecini kolaylaştıran çok kullanışlı araçlarla beraber sunulmaktadırlar. Bunlar, çoğu zaman kurulum (deployment), geliştirme ve hata ayıklama alanında çalışmaktadırlar. Bu araçlar, yazılım mühendislerine önemli zaman kazandırmaktadırlar. Yazılım geliştirme araçları kodların yazımı sırasında hata kontrolü yapabilmekte ve kod düzeltme önerileri sunabilmektedirler. Bu sayede programcıya kodlama esnasında büyük kolaylık sağlanmış olur. Kodların çalıştırılması sırasında da hata kontrolü ve hata ayıklama işlemleri bu programlar sayesinde rahatlıkla yapılabilmektedir. Bu amaçla en çok kullanılan Java yazılım geliştirme programları JBuilder, Weblogic Workshop ve Eclipse'dir

#### **4.11.4 Java uygulama sunucuları**

JSP kodlarını yorumlayıp kullanıcıya çıktı olarak aktarma işlemini gerçekleştiren sunuculara JSP Web Sunucu denir. Farklı firmaların JSP sunucuları bulunmaktadır. İhtiyaca göre, var olan JSP sunucularından herhangi biri seçilebilir. Bu sunucuların farklılıkları, yönetim modülleri ve sağladıkları ek fonksiyonlardan ileri gelir.

J2EE kapsamında ele alınınca, uygulama sunucusu, J2EE teknolojilerini kullanarak geliştirilen uygulamaları, standartlara (J2EE belirtimine) uygun olarak çalıştıran yazılımdır. Kullanıcı arayüzü ile veri tabanı (veya bu görevi gören sistemler) arasında yer alırlar. Çok katlı bir mimari düşünülürse, uygulama sunucusu, mimarinin orta katlarını içinde bulundurur. İşte bu yüzden, uygulama sunucuları, ortakat yazılımı (middleware) sayılırlar ve ortakat yazılım

teknolojilerini kullanırlar. Genellikle, kullanıcı arayüzüyle bilgi işlem servisleri arasında ya da veri işlem katıyla veri tabanı arasında ya da dağıtık nesnelere ölçeklenirlik servisleri arasında olup bu teknolojilerden yararlanırlar.

#### **4.12 Java Teknolojisinde Güvenlik**

Java İnternette birçok alanda karşımıza çıkmaktadır. Milyonlarca kişinin bulunduğu bu ortamda güvenlik konusu önem kazanmaktadır. Java güvenlik alanında iddialı bir dildir. Güvenlik iki başlık altında incelenebilir. Birincisi programlama sırasındaki kod denetimi ve düzenidir. İkincisi, programlama tamamlanıp kullanıma açıldıktan sonra korunması gereken sistem güvenliğidir.

##### **4.12.1 Java programlama dili yapısındaki kod güvenliği**

Kod bazında denetimler Java programlama dilinin önemli bir ayırt edici özelliğidir. Bu özellik diğer programlama dillerinde bulunmayan bir özelliktir. Denetimler:

- Diziler (Array), program kodu çalışırken (run-time) sınırlarını aşıp aşmadıkları kontrol edilir.
- Değişkenlerin dilin yapısına uygun olup olmadıkları ve belirsiz tiplerin kullanılmasında hataya olanak vermemek için değişkenler kontrol altında tutulurlar.
- Hafıza yönetimi otomatik olarak yapılmaktadır. Bu düzenleme hafıza boşluklarının (memory leaks) oluşmasına fırsat vermez. Hafıza boşlukları, hafızanın kullanıldıktan sonra tekrar sisteme iade edilmemesi sonucu ortaya çıkmaktadır.
- Aynı zamanda işaretçilerin (pointer) boşaltılmasına da denetim getirmektedir. Java, programcıya işaretçilere doğrudan müdahale imkânı vermemektedir.

Java programlama dilinde nesne ve sınıflar, tanımlanan türlerine uygun olup olmadıkları kontrol edilir. Diğer bir güvenlik önlemi temel kütüphanelerdeki sınıfların sadece ilişki kurmaları gereken, önceden belirlenmiş sistem kaynaklarından başka kaynaklarla işlem yapmalarını sağlamaktır. Herhangi



bir işlemde bu sınıfların görevleri doğrultusunda işleyip işlemedikleri kontrol edilir. Bu kısım, uygulamaların kontrol altına tutulmasında önemli bir rolü üstlenmektedir. Bu işlemleri yapan birimler güvenlik yönetim birimi (security manager) ve sınıf yükleyiciler (class loaders) dir.

#### **4.12.2 Java sistem güvenliği**

İnternet hayatın vazgeçilmezlerinden biri olmuştur. Bu sebeple İnternet üzerindeki bilgi alışverişlerinde güvenlik büyük önem kazanmıştır. Finansal, kişisel ve iş hayatındaki bilgilerin paylaşımında kiminle haberleşildiğinin bilinmesi, gönderilen bilginin karşı tarafa doğru olarak iletilmesi ve bilgi transferi sırasında bilginin başkaları tarafından izlenememesi, güvenliğin sağlanması açısından gerekmektedir. Bu gereksinimleri yerine getirebilmek amacıyla güvenli yuva tabakası (Secure Sockets Layer-SSL) geliştirilmiştir.

Sistem güvenliği dâhilindeki bilgi akışının güvenliği dışında programlama bazında kullanıcı bilgilerinin denetimi ve kontrolü gerekmektedir. Kullanıcıların güvenliği için, sistemde kullanıcıların ve kullanıcı kaynaklarını tanımlanır. Bu tanımlamalar Java yapısındaki kullanıcı ve kullanıcı grupları tanımlaması ve kaynaklara erişim denetimleri ile yapılabilir. Bir diğer kullanıcı tanımlama ve denetim mekanizması da programlama ile sağlanabilir (Oaks S.,2001).

#### ***Güvenli Yuva Tabakası (Secure Sockets Layer-SSL) :***

SSL network üzerindeki bilgi transferi sırasında güvenlik ve gizliliğin sağlanması amacıyla Netscape firması tarafından geliştirilmiş bir güvenlik protokolüdür. 1996 yılında 3.0 sürümünün çıkarılmasıyla hemen bütün İnternet tarayıcılarının (Microsoft Explorer, Netscape Navigator vb.) desteklediği bir standart haline gelmiş ve çok geniş uygulama alanları bulmuştur.

SSL İnternet Explorer 3.0 ve Netscape Navigator 3.0 tarayıcılar ve üst sürümlerinde çalışabilir. SSL güvenlik alanında gönderilen bilgiler, çok sıradan görünen, fakat çözülmesi gerçekte imkânsız olan bir kodlama sistemiyle şifrelenir. Girilen bilgiler önce şifrelenmiş bir bilgiye dönüştürülür. Bu bilgi

ancak diğer uçtaki web sunucusu tarafından çözümlenip anlamlı bir bilgi haline getirilebilir (Java Security, 2005).

#### 4.13 Java Script

Script basit tanımıyla HTML sayfalarına ek işlevler katabilen ve HTML kodları arasına yerleştirilebilen küçük programcıklara verilen addır. Scriptler tek başlarına çalışamazlar. Mutlaka HTML kodları içerisine yerleştirilirler. Tüm scriptlerin ortak özelliği web sayfalarında dinamik içerik sağlamak ya da kullanıcıyla iletişim kurmak için kullanılır. Scriptler içerisinde Javascript en yaygın ve etkili olarak kullanılan dildir.

İlk başta yalnızca Netscape tarayıcıları için tasarlanan Javascript, daha sonra diğer tarayıcılar tarafından da desteklenen genel bir scripting dili haline gelmiştir. Javascript, uzun fonksiyonlar ve kodlar içeriyorsa tek başına .js uzantılı dosyalara da yazılabilir. Kodlar yazılmış olan dosya HTML kodları içerisinde çağrılabilir ve kullanılabilir.

Javascript, HTML kodları arasında yer aldığından tarayıcı tarafından yorumlanır. Yazılması ve kavranması basit ve kolaydır. Web sayfalarındaki tüm nesnelere çalışabilir. Bu sayede web sayfalarına dinamizm sağlar ve kullanıcı tarafında (client side) denetimi sağlar ve işlevliliği artırır.

#### 4.14 Java Teknolojisi Seçim Nedenleri

Java teknolojileri birçok satıcı tarafından desteklenmektedir. Böylece kod değiştirmeden, istenen satıcının farklı ürünleri üzerinde geliştirme yapılabilir. Java teknolojisinin jsp alt teknolojisi ile oluşturulan sayfaları kullanabilmek için yüksek sistem gereksinimine ihtiyaç duyulmaz, sadece tarayıcı olması yeterlidir.

Java teknolojisini kullanmak için gerek duyulan donanım özellikleri oldukça düşüktür. Java uygulamalarını çalıştırmak için gerekli olan JRE'in (Java Runtime Environment) boyutu yaklaşık 8 MB'dır.

Ticari kalitedeki IDE'lerden (Integrated Development Environment ) Eclipse ve Netbeans IDE'leri ne tür yazılım (ticari/ akademik) geliştirilirse

geliştirilsin ücretsizdir. Jbuilder ve Jdeveloper IDE'leri ise akademik yazılım projeleri için ücretsizdir.

Java gelecek sürümlere uyumluluk ve gelecek güvencesi verir. Örneğin, 1997 yılında yazılan bir program, bugün tüm JVM'lerde sorunsuz bir şekilde çalışır. Fakat aynı güvence diğer teknolojileri için söylenemez. Örnek olarak Microsoft firması Visual Basic 6'ya olan desteği kestiğini açıklamıştır (Altıntaş, A.B, 2004). Java'nın eski sürümleri (Java 1.0.2 ye kadar) Sun Microsystems'in sitesinden indirilebilir.

Java teknolojisi, hakkında en fazla kaynak bulunan teknolojidir (Tablo 4.1). Geliştirilmek istenen projelerde rahatlıkla kaynak bulunabilmektedir.

Tablo 4.1 Google arama motorundaki programlama dillerinin arama sonucu (www.google.com, Ocak 2005)

<b>Programlama Dili</b>	<b>Kaynak sayısı</b>
Java	148.000.000
Perl	51.400.000
C++	29.100.000
Visual Basic	19.800.000
Delphi	15.000.000
C#	11.000.000

Java, diğer programlama dilleri ile uyumlu bir şekilde çalışır. JNI (Java Native Interface) dışında JSR-223 belirtimi de diğer diller ile ortak konuşmayı sağlamaya yöneliktir. Java, küçük cihazlardan büyük sistemlere kadar birçok ortamda çalışır.

Avrupa finansal enstitüsünün yaptığı araştırmada büyük yazılım projelerinin %75 inin Java teknolojileri kullanılarak geliştirildiği ortaya çıkmıştır. Software Development Times adlı programlama haber sitesinin anketi bu sonuçları doğrulamaktadır (Software Development Times, Ocak 2005).

Java teknolojileri sadece üniversitelerde öğretilen bir ders değildir. Aynı zamanda birçok araştırma ve geliştirme çalışmalarında yoğun olarak kullanılmaktadır ve desteklenmektedir. UC Berkeley, Matt Welsh tarafından yapılan çalışmalar sonucu yeni giriş çıkış (Input Output- I/O) sistemi, Java 1.4

sürümüne dahil edilmiştir. Güney Avustralya Üniversitesi tarafından Java 1.5' de gelecek olan "Generic Java Language Extension" özelliği geliştirilmiştir. Bunun yanı sıra CERN ve HEP gibi araştırma organizasyonları kullanıcılar için Java projeleri geliştirip sunmaktadırlar.

Paralel sistemlerde ucuz fiyat avantajı sağlar. Google'ın başarılı olmasındaki sebeplerden biri, çok da maliyetli olmayan on bin adet sunucunun paralel bir şekilde çalışmasıdır. Yeni uygulama, tek ve güçlü bir sunucu almaktansa, ucuz ama birçok sunucu alıp bunları paralel bir şekilde çalıştırmak şeklindedir. Java, bu imkânı vermektedir.

Java, programlama dilleri içinde kullanımı gün geçtikçe artan bir dildir. Tablo 4.2'de 2004 yılının ilk yıllarında yapılan bir araştırma sonucu programlama dillerinin kullanım oranları verilmiştir.

Tablo 4.2 Programlama dillerine göre proje sayıları (www.sourceforge.net, Ocak 2005)

<b>Programlama Dili</b>	<b>Proje Sayısı</b>
C++	12,765
C	12,762
Java	11,203
PHP	8,437
Perl	5,317
Python	2,999
Javascript	1,612
C#	1,259
TCL	778
Ruby	283
Ada	77
Eiffel	67

## **5.ÖĞRENCİ KAYIT SİSTEMİ TANIMI VE YAPISI**

Bu bölümde, üniversitelerdeki el ile kayıt sistemi ve İnternet üzerinden kayıt sistemleri incelenmiştir. İnceleme sonunda mesele ve aksaklıklar tespit edilerek MiKaS geliştirilmiştir.

### **5.1 Klasik Kayıt Aşamalarının İncelenmesi**

Kayıt aşamaları, Muğla Üniversitesinde güz ve bahar dönemi içinde incelenmiştir; kayıt başlamadan önce ve kayıt sırasında yapılan işlemler ortaya konmuştur. Daha sonra el ile kayıt yapılan sistemdeki eksiklikler tamamlanmış, meselelere çözüm üretilerek yeni kayıt sistemi yapısı ortaya konmuştur.

#### **5.1.1 Kontrol ve ön hazırlık aşaması**

Öğrencinin kayda başlaması için ön şart olan harç miktarı belirlenmekte ve sistemde saklanmaktadır. Kontrollerden sonra anlaşmalı bankaya harç bilgileri yollanmaktadır. Harç bilgileri oluşturulduktan sonra eğitim öğretim programları belirlenmekte ve ilgili bölüm/programlara yazılı olarak bildirilmektedir.

#### **5.1.2 Kayıt yenileme ve ders kaydı**

Kayıt yenileme için, öğrencinin harç miktarını banka şubesine yatırması ve dekontu danışmana getirmesi gerekmektedir. Öğrenci daha sonra, ders kaydı için danışmanla derslerin seçimi ve kaydını gerçekleştirmektedir. Ders kaydı sırasında danışmanın öğrencinin geçmiş dönem bilgilerine ulaşması veya öğrencinin durum kontrolünü mutlaka yapması gerekmektedir.

### **5.2 El İle Kayıt Sisteminde Karşılaşılan Meseleler**

Klasik kayıt sisteminde karşılaşılan ilk problem, kayıt yenileme için öğrencinin danışmana ödeme dekontunu getirmesidir. Öğrenci bulunduğu şehirden harç yatırıyorrsa dekontun danışmana ulaştırılması zaman kaybına yol açmaktadır.

Ders kaydı sırasında danışman, öğrenciye ait geçmiş dönem bilgilerini gözden geçirmek ve not ortalamasını kontrol etmek zorundadır. Ancak bu bilgilere dayanarak yeni dönemden ders seçimi ve kaydı yapılabilir. Bu kontroller ve bilgilerin gözden geçirilmesi işlemleri her öğrenci için yapıldığında önemli bir zaman kaybını ortaya çıkarmaktadır. Aynı zamanda danışmanın dikkatinden kaçabilen ders veya öğrenci durumları da söz konusu olabilmektedir. Bu da yanlış ders seçimlerini ortaya çıkarmakta ve daha sonra düzeltilebilmesi güç durumlara yol açabilmektedir.

Bölüm ve programlara göre ders seçimi bazı durumlarda farklılık gösterebilmektedir. Bu durumda danışmanın ayrıca bölüm kararlarını da iyi takip etmesi ve kayıt sırasında dikkate alması gerekmektedir.

Öğrenci, farklı bölümlerden ders seçmek istediğinde danışman farklı bölümdeki dersle ilgili kontenjan kontrolünü yapamadığından öğrenciyi ilgili bölümlere yönlendirmek zorunda kalmaktadır. Bu durum öğrenci ve danışman için vakit kaybına neden olmaktadır.

### **5.3 MiKaS Aşamaları**

Klasik kayıt sisteminin incelenmesi ve problemlerinin belirlenmesinden sonra öğrenci kayıt sistemi aşamaları yeniden düzenlenmiştir. MiKaS'ta aşamalar, zaman kaybı ve hata oranını azaltacak şekilde oluşturulmuştur.

#### **5.3.1 Kontrol ve ön hazırlık aşaması**

Öğrencilere ait tanımlayıcı bilgiler, öğrenci sisteme ilk giriş yaptığında kontrol edilir. Bu bilgiler kayıt sırasında kontrol için kullanılacaktır:

*Öğrenci Dönem ve Yıl Bilgileri:* Öğrencilere ait yıl ve dönem bilgileri veri tabanı yapısında saklanır. Böylece danışmanın yapması gereken öğrenci dönem ve not bilgisi kontrolleri kayıt esnasında yapılabilmektedir.

*Harç ve Kredi Bilgileri:* Öğrencinin kayda başlaması için ön şart olan harç miktarının doğru olarak belirlenmesi ve sistemde saklanması gerekmektedir. Bu bilgiler belirlendikten sonra anlaşma yapılan banka ile harç ödeme bilgilerinin eşzamanlı olarak aktarımı konusunda anlaşma sağlanır. Böylece öğrencinin harç

yatırma bilgisi bilgisayar ortamında gecikmesiz olarak alınabilir ve danışmanın dekontu daha sonra edinmesi sağlanır.

*Fakülte ve Yüksekokul Eğitim Öğretim Programlarının Hazırlanması ve Kontrolü:* Yeni döneme ait eğitim-öğretim programları oluşturulur. Böylece öğrenci İnternet üzerinden kendi bölümü dışındaki diğer bölümlerden de ders seçebilir ve seçim sırasında kontenjan kontrolü otomatik olarak yapılabilir.

### **5.3.2 Kayıt yenileme ve ders kaydı**

Geliştirilen sistemde kayıt yenileme işlemi için öğrencinin harç miktarını yatırması ve İnternet üzerinden kayıt sistemine girmesi yeterlidir. Giriş yaptıktan sonra sistem banka kontrollerini gerçekleştirip kayıt yenilemeyi otomatik olarak gerçekleştirir. Ders kaydı işlemi için ilerleyerek gelen listelerden ders seçimini yapması yeterli olacaktır. Yapılan işlemler, danışmanla daha sonra değiştirilebilir veya iptal edilebilir.

### **5.4 Kayıt Sistemi Tablo Yapıları**

Öğrenci bilgilerinin saklanması için veri tabanı yapısı Microsoft SQL Server 2000 seçilmiştir. SQL Server kullanımı ve yönetimi kolay bir veri tabanı yönetim sistemi sunmaktadır. İsteğe göre bu yapı farklı veri tabanı sunucularına çevrilebilir. Veri tabanı yapısındaki tablolar Bölüm 5.5.1-5.5.7’de incelenmektedir.

#### **5.4.1 Öğrenci genel bilgileri tablo grubu**

Öğrenciye ait genel bilgilerin tutularak sistemin öğrenci tanımı ve takibini yapması sağlanır. Bu tablo grubunda üniversite içerisinde öğrenciye ait tanımlayıcı tüm bilgiler tutulabilir.

Sunulan sistem için örnek tablo “Stud01” adlı tabloda tutulmaktadır. Tabloya ait tanımlayıcı bilgiler Tablo 5.1 ‘de yer almaktadır.

Tablo 5.1 Stud01 tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
StudentNumber	Varchar	15
PreName	Varchar	15
FirstName	Varchar	15
LastName	Varchar	15

Tablo 5.1’de Stud01 tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- StudentNumber: Öğrenci numarası.
- PreName, FirstName, LastName: Öğrenci isim bilgilerini kapsar

#### 5.4.2 Öğrenci dönem bilgileri tablo grubu

Öğrenci sistemde farklı dönemlerde farklı durumlarda bulunabilir. Bu durumlar kayıt dondurma, ceza veya hazırlık gibi kayıt sistemini etkileyen durumlar olabilir. Bu yüzden öğrencinin dönemsel bilgileri mutlaka tutulmalıdır

Dönem bilgileri sistemde “Stud01Status” tablosunda tutulur. Tabloya ait tanımlayıcı bilgiler Tablo 5.2.’de bulunmaktadır.

Tablo 5.2 Stud01Status tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
Stud01ProgramRecordNo	İnt	4
EduYearRecordNo	İnt	4
EduSemRecordNo	İnt	4
Status	varchar	15

Tablo 5.2’de Stud01Status tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- Stud01ProgramRecordNo: Öğrencinin kayıtlı bulunduğu bölüm anahtarı.
- EduYearRecordNo, EduSemRecordNo: Dönem anahtarları
- Status: Öğrenci dönem durumunu kapsar.



### 5.4.3 Öğrenci ders bilgileri tablo grubu

Öğrencinin aldığı derslerin takibi için derslere ait bilgiler sistemde saklanmalıdır. Bu bilgiler dersin kredisi, öğrenci sınav notları ve öğrenci başarı durumu gibi bilgilerdir.

Ders bilgilerinin saklanması örnek sistemde “Stud01Course” ve “Stud01CourseOverload” tablolarında tutulur. “Stud01Course” tablosunda alınan dersler, “Stud01CourseOverload” tablosunda da ders yerine saydırma bilgileri tutulur.

Tablo 5.3 Stud01Course tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
Stud01StatusRecordNo	İnt	4
CurriculumRecordNo	İnt	4
TakeType	varchar	10
FirstExam	Real	4
FirstExamMakeUp	Real	4
SecondExam	Real	4
SecondExamMakeUp	Real	4
HomeWork	Real	4
Attendance	Real	4
Project	Real	4
Final	Real	4
FinalMakeUp	Real	4
Average	Real	4
Grade	Real	4
WeightedGrade	Real	4
Letter	varchar	2
Status	varchar	10
Muaf	smallint	2
CancelDeptCourseOverride	İnt	4
BolumDisiSecmeli	smallint	2

Tablo 5.3’de Stud01Course tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- Stud01StatusRecordNo: Öğrenci dönem durumu tablo anahtarı.
- CurriculumRecordNo: Ders anahtar alanı.
- TakeType: Dersi alış tipi.

- FirstExam- BolumDisiSecmeli: Bu alanlar isteğe bağlı olarak dersle ilgili tutulacak bilgileri saklar. (Ara sınavlar, ders tipi, muafiyet vb..).

Tablo 5.4 Stud01CourseOverload tablosu

Alan Adı	Türü	Uzunluk
RecordNo	int	4
OverloadedStud01Course	int	4
OverloadingStud01Course	İnt	4

Tablo 5.4’de Stud01CourseOverload tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- OverloadedStud01Course: Yerine saydırılan ders anahtarı.
- OverloadingStud01Course: Yerine sayılacak ders anahtarını kapsar.

#### 5.4.4 Öğrenci bölüm bilgileri tablo grubu

Öğrencinin kayıtlı olduğu bölüm ve bölüme ait bilgiler, kayıt için farklı tablo gruplarında saklanır, bu durumda anahtar alanla diğer tablolar bağlantı kurulabilir.

Örnek sistemde öğrencinin kayıtlı olduğu bölüme ait bilgiler “Stud01\_Program” tablosunda tutulmaktadır. Bu tabloya ait tanımlayıcı bilgiler Tablo 5.5’te görülmektedir.

Tablo 5.5 Stud01\_Program tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
Stud01RecordNo	İnt	4
EducationTypeRecordNo	İnt	4
Status	Varchar	15
StudentYear	Smallint	2

Tablo 5.5’de Stud01\_Program tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- Stud01RecordNo: Öğrenci anahtar alanı.
- EducationTypeRecordNo: Öğretim türü anahtar alanı.
- Status: Öğrenci durum bilgisi.
- StudentYear: Öğrenci yılını kapsar.

#### 5.4.5 Öğrenci harç bilgileri tablo grubu

Kayıt yenilemeye geçmeden önce öğrencinin yatırması gereken harç miktarlarının kontrolü ve denetimi için bu gruptaki tablolar oluşturulmalıdır. Bu tablolar, farklı üniversitelerdeki farklı harç ve kayıt gideri yapısına göre değişiklik arz edebilir.

Örnek sistemde öğrenciye ait harç ödeme bilgileri “Stud01\_Payments” tablosunda tutulur (Tablo 5.6). Farklı ödeme türleri için ödeme türüne ait tanımlayıcı bilgiler “PaymentTypes” tablosunda saklanır (Tablo 5.7).

Tablo 5.6 Stud01\_Payments tablosu

Alan Adı	Türü	Uzunluk
RecordNo	int	4
Stud01ProgramRecordNo	int	4
StudentNumber	varchar	10
StudentName	varchar	60
EduYearRecordNo	int	4
EduSemRecordNo	int	4
EducationType	varchar	15
PaymentTypeCount	smallint	2
PaymentTypeRecordNo	int	4
Amount	float	8
Interest	float	8
PaymentDate	datetime	8
Sent	smallint	2
Paid	smallint	2

Tablo 5.6’da Stud01\_Payments tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- Stud01ProgramRecordNo: Öğrencinin kayıtlı olduğu bölüm anahtarı.
- EduYearRecordNo, EduSemRecordNo: Dönem anahtarları.
- EducationType: Öğretim türü.
- PaymentTypeCount: Borç sayısı.
- PaymentTypeRecordNo: Borç türü.
- Amount: Borç miktarı.
- Interest: Faiz.
- PaymentDate: Ödeme tarihi.
- Sent: Bankaya gönderildi bilgisi.
- Paid: Ödeme onayını kapsar.

Tablo 5.7 PaymentTypes tablosu

<b>Alan Adı</b>	<b>Türü</b>	<b>Uzunluk</b>
RecordNo	İnt	4
PaymentType	varchar	50

Tablo 5.7’de Payment\_Types tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar.
- PaymentType: Borç türünü kapsar (Harç, Kayıt gideri vb.).

#### 5.4.6 Sistem dönem bilgileri tablo grubu

Kayıt sırasında, aşamalarda kullanılan ve kontrol edilen değişkenler saklanır. Kayıt başlangıç ve bitiş tarih aralıkları, gerekli ise farklı bölüm/programlara göre tarih aralıkları tanımları yapılır. MiKaS'ta dönem bilgileri üç tabloda tutulur (Tablo 5.8-5.9-5.10), "EduSem" tablosunda yarıyıl bilgisi "EduYear" tablosunda yıl bilgileri ve "RegistrationParameters" tablosunda kayıt tarihleri tutulur.

Tablo 5.8 RegistrationParameters tablosu

Alan Adı	Türü	Uzunluk
RecordNo	int	4
EduYearRecordNo	int	4
EduSemRecordNo	int	4
Reg1BeginDate	datetime	8
Reg1EndDate	datetime	8
Reg2BeginDate	datetime	8
Reg2EndDate	datetime	8

Tablo 5.8'de RegistrationParameters tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- EduYearRecordNo, EduSemRecordNo:Dönem bilgileri.
- Reg1BeginDate, Reg1EndDate: Kayıt yenileme başlangıç ve bitiş tarihleri.
- Reg2BeginDate, Reg2EndDate: Ders kaydı başlangıç ve bitiş tarihlerini kapsar.

Tablo 5.9 EduSem tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
Name	varchar	20

Şekil 5.9’da EduSem tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar Alan.
- Name: Dönem adını kapsar.

Tablo 5.10 EduYear tablosu

Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
Name	Varchar	20

Tablo 5.10’da EduYear tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar Alan.
- Name: Yıl adını kapsar.

#### 5.4.7 Öğrenci şifre bilgileri tablo grubu

Kayıt sistemine giriş için öğrencilere ait kullanıcı adı ve şifre bilgilerinin saklanması gerekmektedir. Sistemde öğrenci bilgileri “StudentLogins” tablosunda saklanır (Tablo 5.11).

Tablo 5.11 StudentLogins tablosu

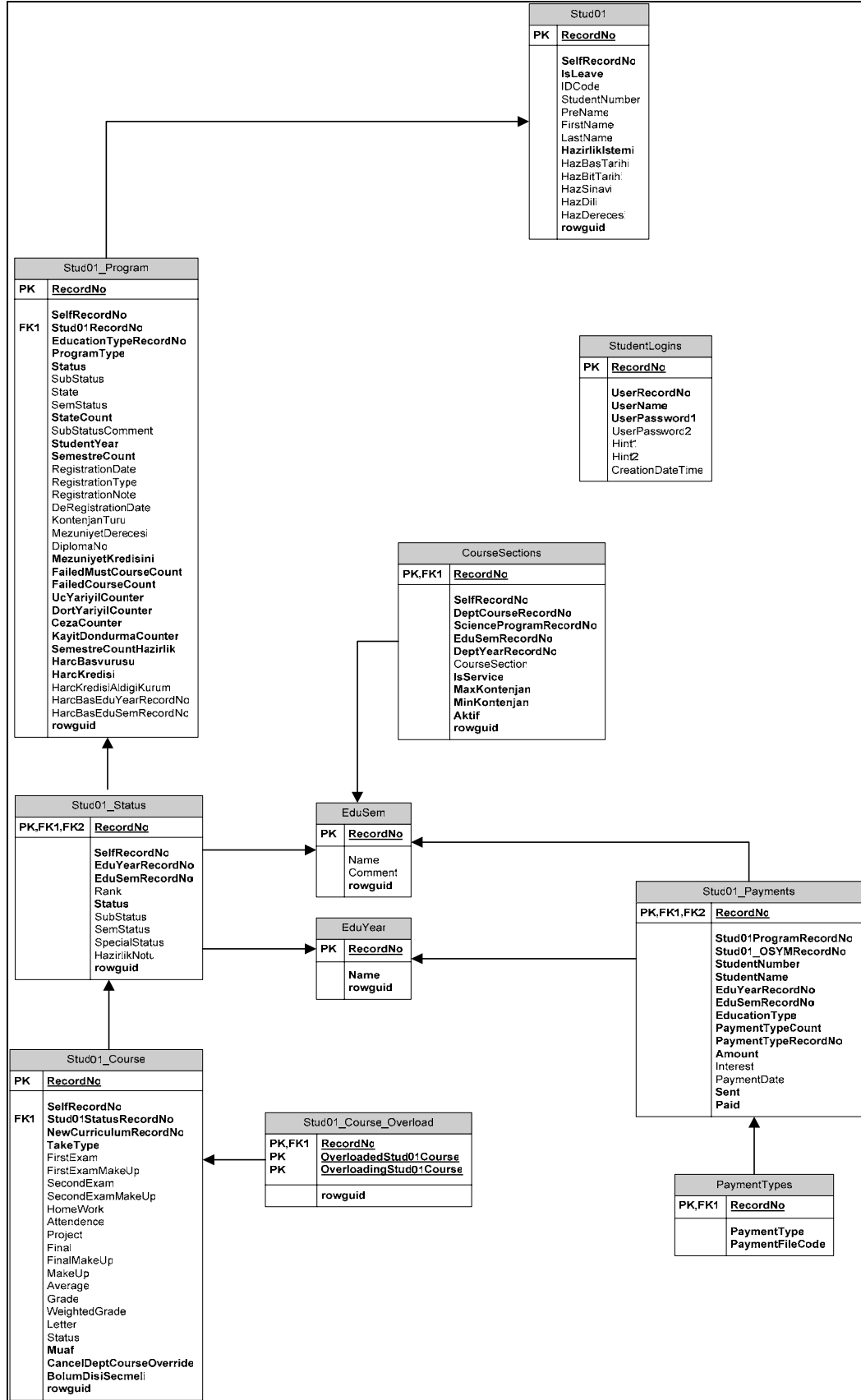
Alan Adı	Türü	Uzunluk
RecordNo	İnt	4
UserRecordNo	İnt	4
UserName	Varchar	10
UserPassword	Varchar	10

Tablo 5.11’de StudentLogins tablo alanları verilmiştir. Burada;

- RecordNo: Anahtar alan.
- UserRecordNo: Öğrenci tablosu yabancı anahtarı.
- UserName: Kullanıcı adı.
- UserPassword: Kullanıcı şifresini kapsar.

### **5.5 Öğrenci Bilgileri Tablo İlişkileri**

Kayıt yapısı içerisinde oluşturulan tabloların birbirleriyle veri tabanı içindeki ilişkileri Şekil 5.1’de görülmektedir. Bu yapıya gerektiği takdirde yeni tablolar eklenerek genişleme yapılabilir.



Şekil 5.1 Tablo ilişkileri



## 5.6 Kayıt Sistemi Tanımı

Öğrenci Kayıt Sistemi üç ana aşamadan oluşur. Bu aşamaların her birinde danışmana gerek duyulmadan yapılabilecek işlemler, öğrenciye sunulur ve ilgili seçimler sisteme kaydedilir. Öğrenci işlemlerini gerçekleştirirken her aşamada, sistem tarafından yapılan işlemlerin kaydı tutulur. Bu sayede sistem yöneticisi veya danışman istediği zaman sistemdeki hareketleri detaylı olarak görebilir, gerektiğinde sonradan bu işlemleri iptal edebilir veya değiştirebilir.

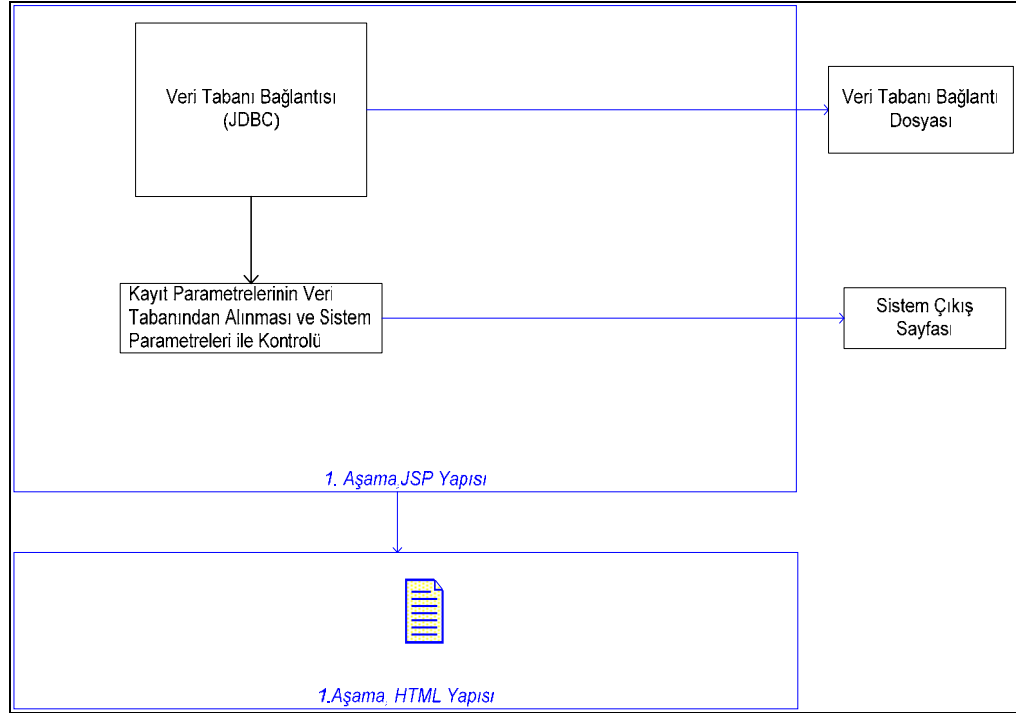
Kayıt sistemi yapısında iki tür kontrol mekanizması mevcuttur. Sunucu kısmında, sayfalar oluşturulmadan önce uygulanması gereken kurallar kontrol edilir ve öğrenciye bu kuralları içeren sayfa içeriği yansıtılır. Kullanıcı tarafında ise, kullanıcı işlemlerinin kontrolü ve yönlendirilmesi için denetim mekanizmaları mevcuttur.

### 5.6.1 Birinci aşama: açıklama ve sistem bilgilerinin doğrulanması

Sisteme giriş yapmış olan öğrencinin giriş kontrolleri yapılır. Şekil 5.2’de birinci aşama sayfa yapısı verilmiştir. Girişte kontrol edilen bilgiler:

- *Öğrenci Giriş Bilgilerinin Doğrulanması:* Sisteme girerken kullanılan kullanıcı adı ve şifre bilgileri kontrolden geçirilir. Kullanıcı adı ve şifre bilgileri veri tabanından okunarak doğrulanma işlemi gerçekleştirilir.
- *Kayıt Tarihi Bilgileri:* Veri tabanına girilmiş olan kayıt tarihi aralığı kontrolü yapılır. Eğer kayıt periyodu dışında ise öğrenci uyarılarak kayıt mekanizması sonlandırılır.

Kontrol işlemlerinden sonra öğrenciye kayıt sistem yapısı ile ilgili açıklamaları içeren HTML kodları görüntülenir ve kayda devam etmesi için veya çıkış için seçenek verilir.

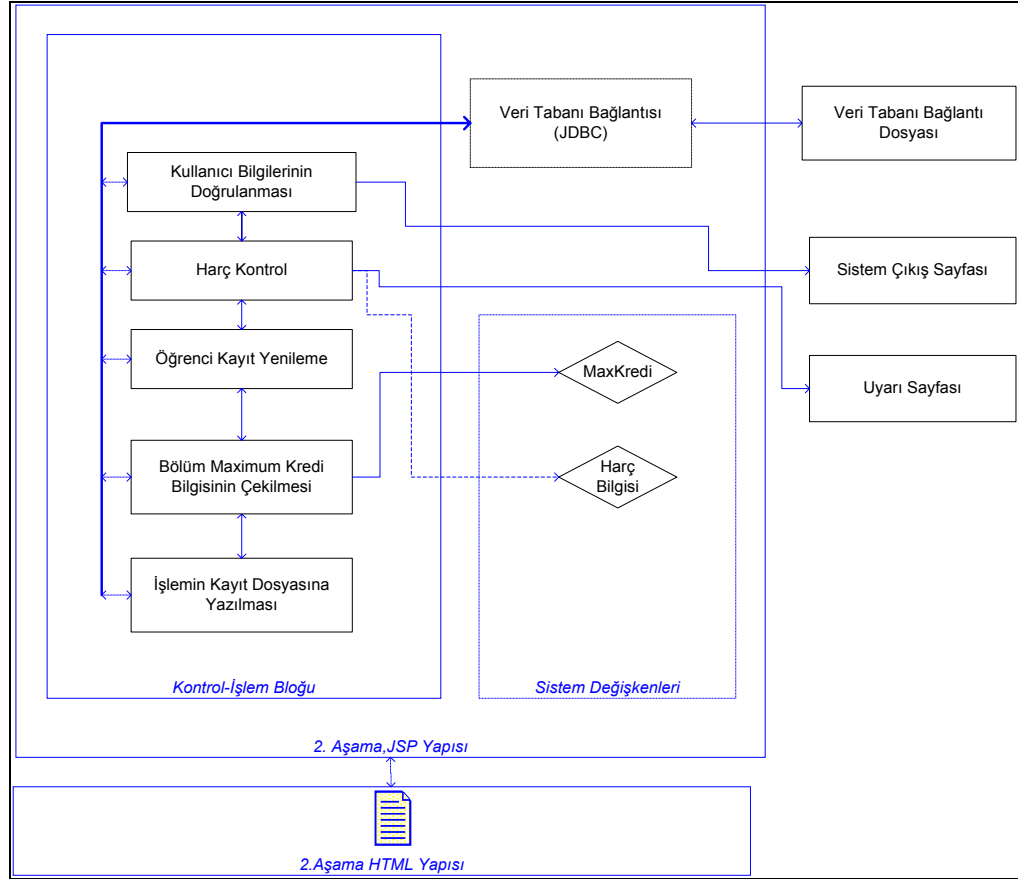


Şekil 5.2 Birinci aşama sayfa yapısı

### 5.6.2 İkinci aşama: harç denetimi ve kayıt yenileme

Her aşama başlangıcında olduğu gibi önce öğrenci şifre bilgileri daha sonra harç bilgileri kontrol edilir. Eğer ilgili miktarlar yatırılmışsa öğrencinin kaydı yenilenir ve bu işlem mesaj olarak öğrenciye yansıtılır. Harç miktarı yatırılmamışsa, sistem tarafından durum öğrenciye bildirilir ve kayıt işlemi sonlandırılır. Daha sonra öğrenci harcını yatırıp tekrar kayıt işlemlerine devam edebilir.

Sistemde daha sonraki aşamalarda kullanılacak olan maksimum kredi ve harç durum bilgisi kullanıcı oturumundaki değişkenlere aktarılırlar. Şekil 4.3'te ikinci aşama sayfa yapısı verilmiştir.



Şekil 5.3 İkinci aşama sayfa yapısı

### 5.6.3 Üçüncü aşama (1.ana aşama): önceki dönemlerden başarısız olunan zorunlu derslerin seçimi

Ders seçimleri ve durum kontrol mekanizmaları bu aşamadan itibaren başlamaktadır. Aşamada yer alan adımlar aşağıda yer almaktadır:

- Standart kontrollerin (harç, şifre vb..) yapılması,
- Öğrencinin kaldığı zorunlu ders listesinin uygun bir SQL fonksiyonu veya sorgusu ile bulunması,
- Elde edilen liste html kodlarla web sayfasında görüntülenmesi,
- Öğrenci not ortalaması sisteme uygun yazılmış bir fonksiyonla alınıp, kullanıcı oturumundaki genel not ortalaması değişkenine aktarılması,
- Öğrenciye ait mevcut dönem ders listesi veri tabanından alınarak bilgi amacıyla sayfada listelenmesi işlemi gerçekleştirilir.

Yukarıdaki işlemler sonucunda elde edilen html sayfası kullanıcıya yollanarak tarayıcısında görüntülenir. Bu sayfa içinde ayrıca kullanıcı bilgisayarında kontrolleri yapacak olan JavaScript kodları da oluşturulur. Bu kodlar kredi kontrolü yaparak öğrencinin ders seçim esnasında maksimum kredi miktarını aşmamasını sağlar. Birinci ana aşama olan bu aşamadan itibaren standart olan bir ekran yapısı öğrencinin karşısına çıkmaktadır. Şekil 5.4’de üçüncü aşamanın ekran görünümü görülmektedir.

Muğla Üniversitesi  
Bilgi İşlem Daire Başkanlığı

Genel Not Ortalaması : 0.55  
Alınan Toplam Kredi : 18  
Alınabilecek Max. Kredi : 18

Ders Kaydı - Adım 1

Liste 1 - 2003-2004 Bahar dönemi öğrenci ders kayıt listesi (Şu ana kadar aldığınız dersler)

Kredi	Kod	Kısım	Ders	Türü
1	0	TDB104	22 TÜRK DİLİ II	Zorunlu
2	0	YDB134	1 İNGİLİZCE II--	Seçmeli
3	3	İKT102	2 İKTİSADA GİRİŞ-II	Zorunlu
4	3	KAY122	2 ANAYASA HUKUKU	Zorunlu
5	3	İŞL142	2 BORÇLAR HUKUKU	Zorunlu
6	3	MAT118	2 MATEMATİK II	Zorunlu
7	3	KAY114	2 SİYASAL DÜŞÜNCELER TARİHİ	Zorunlu
8	3	İKT110	2 İKTİSAT TARİHİ	Zorunlu
Toplam: 18				

Liste 2 - Önceki dönemlerden başarısız olunan zorunlu dersler:

Kredi Kod	Kısım	Ders	Türü
Toplam: 0			

Alınamayacak dersler

Liste 1’de şu ana kadar almış olduğunuz ders listesi, Liste 2’de daha önceki dönemlerden kaldığınız zorunlu dersler listesi yer almaktadır.

Önceki dönemlerden almanız gereken zorunlu ders görünmemektedir 2. adıma geçmek için "Devam" butonunu kullanınız.

Devam

Şekil 5.4 Üçüncü aşama ekranı

Şekil 5.4’te verilen ekranda farklı kısımlar, öğrenciye yol gösterir ve öğrenciyi bilgilendirir. Bu alanlar şekil üzerinde incelenirse:

1. Alan: Öğrencinin genel not ortalaması ve aldığı kredi miktarı görüntülenir.

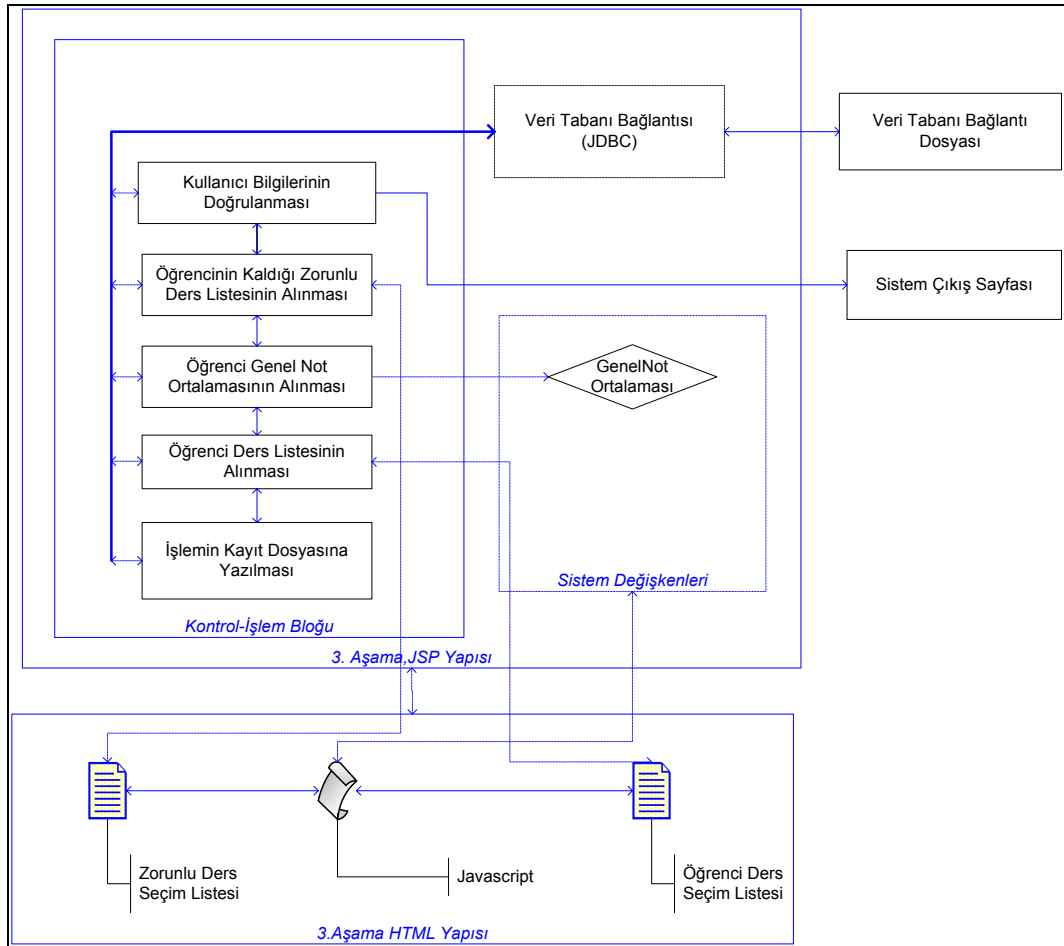
2. Alan: Ders kaydında bulunulan adım görüntülenir.

3. Alan: Öğrenci ders listesi bu alanda listelenir. Her aşamada seçim yaptıkça dinamik olarak güncellenen listedir.

4. Alan: Öğrencinin seçim yapması için ders listesinin verildiği kısımdır. Bu listede derslerle ilgili kredi, kod, kısım ve tür bilgileri yansıtılır.

5. Alan: Bu alanda sisteme daha önceden girilmiş kurallar ve bilgiler doğrultusunda öğrenciye açıklamalar yapan ve yol gösteren bir “Sanal Danışman” yer almaktadır.

Şekil 5.5’de üçüncü aşama sayfa yapısı görülmektedir.



Şekil 5.5 Üçüncü aşama sayfa yapısı

#### **5.6.4 Dördüncü aşama (ikinci ana aşama): mevcut dönemden zorunlu derslerin seçimi**

İkinci ana aşama, bir önceki aşamada yapılan seçimlerin veri tabanına yazıldığı ve mevcut dönemde tanımlanmış olan eğitim-öğretim programında var olan zorunlu derslerin listelendiği aşamadır. Yine bu aşamada da standart ekran alanları gelir ve öğrencinin durumuna göre seçim yapması sağlanır.

- Öğrencinin daha önceki aşamada varsa aldığı derslerin kredisi oturum değişkenlerine işlenir ve maksimum kredi hesabı bu sayıya göre kontrol edilir.
- Bir önceki aşamada yapılan işlemler izleme tablosuna yazılır.
- Kullanıcı bilgilerinin kontrolü yapılır.
- Mevcut dönem zorunlu ders listesinin alınması ve listelenmesi veri tabanı bağlantısı ile gerçekleştirilir.
- Öğrencinin gelinen aşamaya kadar aldığı dersler listelenir.
- Sanal danışman alanında, öğrencinin durumuna göre alabileceği kredi sayısı ve açıklamalar görüntülenir.

Genel Not Ortalaması : 0.55  
 Alınan Toplam Kredi : 18  
 Alınabilecek Max. Kredi : 18

Adım 2

Liste 1 - 2003-2004,Bahar Ders Kayıt Listesi (Şu ana kadar aldığınız dersler)

Kredi Kod	Kısım	Ders	Türü
1 0 TDB104	22	TÜRK DİLİ II	Zorunlu
2 0 YDB134	1	İNGİLİZCE II-	Seçmeli
3 3 İKT102	2	İKTİSADA GİRİŞ-II-	Zorunlu
4 3 KAY122	2	ANAYASA HUKUKU	Zorunlu
5 3 İŞL142	2	BORÇLAR HUKUKU	Zorunlu
6 3 MAT118	2	MATEMATİK II	Zorunlu
7 3 KAY114	2	SİYASAL DÜŞÜNCELER TARİHİ	Zorunlu
8 3 İKT110	2	İKTİSAT TARİHİ	Zorunlu
Toplam			18

Liste 2 - 2003-2004,Bahar Eğitim-Öğretim Programındaki Zorunlu Dersler

Kredi Kod	Kısım	Ders	Sınıf	Türü
1 3 İKT104	1	<input type="checkbox"/> DOĞAL KAY. VE ÇEV. EK.	1	Zorunlu
2 3 İKT106	1	<input type="checkbox"/> İKTİSAT SOSYOLOJİSİ	1	Zorunlu
3 2 TDB104	29	<input type="checkbox"/> TÜRK DİLİ II	1	Zorunlu
Toplam			0	

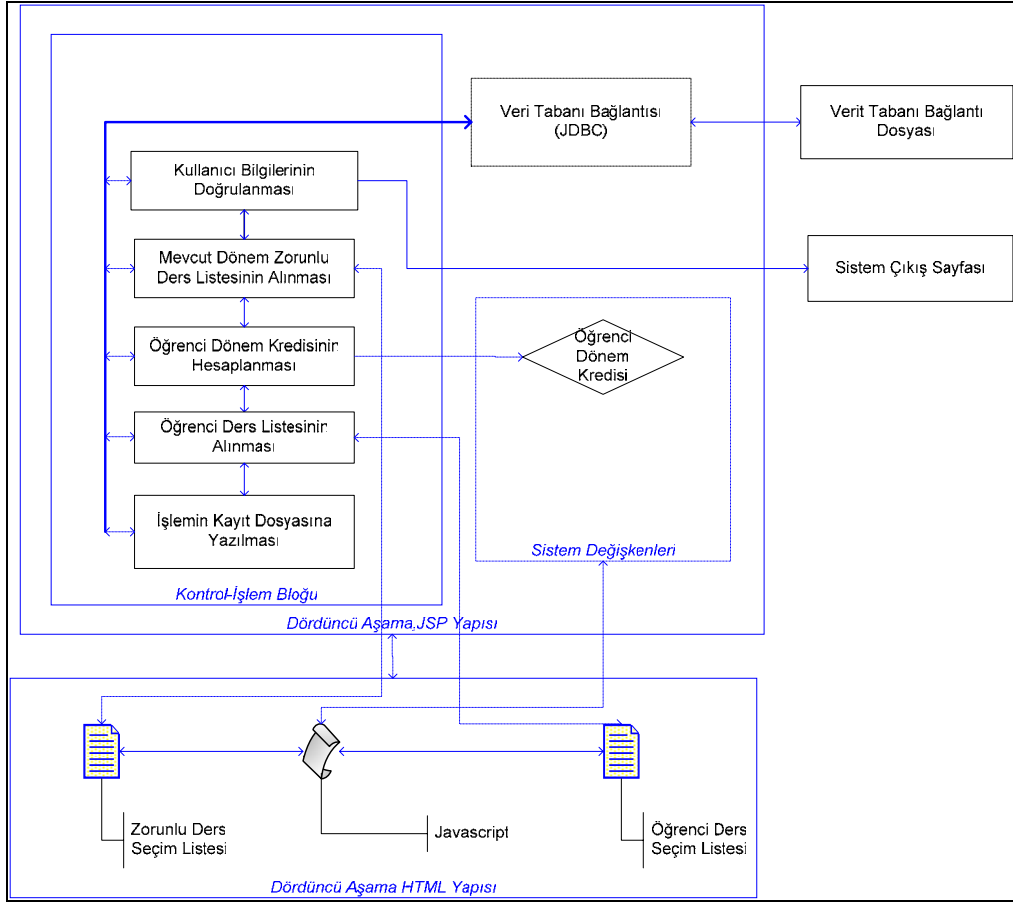
Liste 1'de şu ana kadar almış olduğunuz ders listesi, Liste 2'de bu dönem açılmış olan zorunlu ders listesi yer almaktadır. Liste 1'deki kredi toplamı dahil olmak üzere en fazla bu listeden 18 kredilik ders alabilirsiniz.

Sistem zorunlu dersleri otomatik olarak işaretleyecektir. Birden fazla kısma ayrılmış derslerin 1. kısmı, sistem tarafından otomatik olarak verilir. Daha sonra danışmanınızla bu kısmı değiştirebilirsiniz.

Eğer daha önceden aldığınız dersler listede zorunlu olarak işaretli geliyorsa bunun nedeni sistemdeki yerine saydırma veya muafiyet işlemlerinin doğru yapılmamış olmasından kaynaklanır. Bu durumda kayda devam etmeyiniz ve işlemlerinizi danışmanınızla birlikte yapınız.

Şekil 5.6 Dördüncü Aşama Ekranı

Şekil 5.6'da dördüncü aşama ekranı ve Şekil 5.7'de dördüncü aşama sayfa yapısı görülmektedir.



5.7 Dördüncü aşama sayfa yapısı

### 5.6.5 Beşinci aşama (üçüncü ana aşama): mevcut dönemden seçmeli derslerin seçimi

Son ana aşama olan bu aşamada öğrenci, kredi durumuna göre seçmeli derslerini seçer. Seçim sırasında yine daha önceden belirlenmiş kriterler öğrenci tarafında kontrol edilir.

- Kullanıcı bilgileri kontrol edilir.
- Öğrencinin daha önceki aşamada varsa aldığı dersler dönem kredisine eklenir.
- Bir önceki adımda yapılan işlemler izleme dosyasına yazılır.
- Mevcut döneme ait seçmeli ders listesi veri tabanından alınır.
- Öğrencinin gelinen aşamaya kadar aldığı ders listesi görüntülenir.



- Kredi kontrolü ve ders seçimi için JavaScript kodları oluşturulur. Böylelikle öğrenci tarayıcısında maksimum kredisini aşmadan ders seçebilir.

Şekil 5.8’de beşinci aşama ekranı ve Şekil 5.9’da beşinci aşama sayfa yapısı görülmektedir.

Genel Not Ortalaması	: 0.55	<b>Adım 3</b>
Alınan Toplam Kredi	: 18	
Alınabilecek Max. Kredi	: 18	

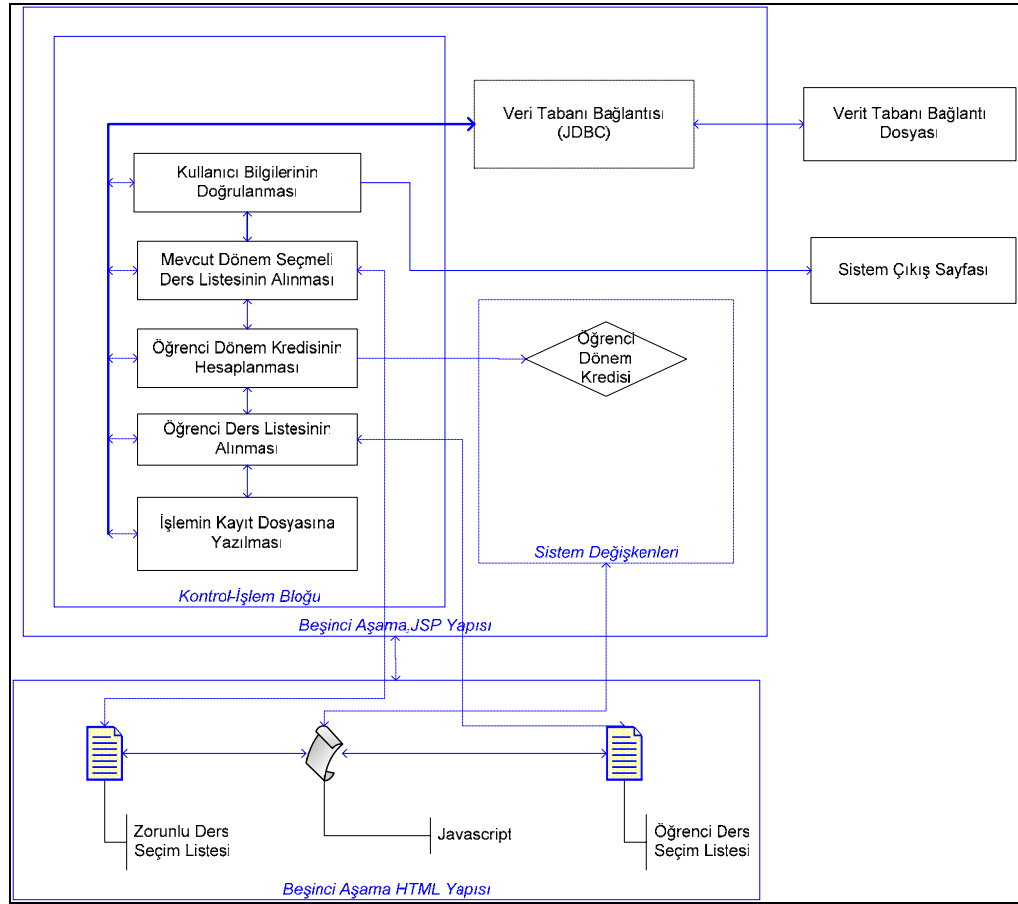
  

Liste 1 - 2003-2004,Bahar Ders Kayıt Listesi (Şu ana kadar aldığınız dersler)				
Kredi Kod	Kısım	Ders	Türü	
1	0	TDB104 22	TURK DILI II	Zorunlu
2	0	YDB134 1	İNGİLİZCE II--	Seçmeli
3	3	İKT102 2	İKTİSADA GİRİŞ-II-	Zorunlu
4	3	KAY122 2	ANAYASA HUKUKU	Zorunlu
5	3	İŞL142 2	BORÇLAR HUKUKU	Zorunlu
6	3	MAT118 2	MATEMATİK II	Zorunlu
7	3	KAY114 2	SİYASAL DÜŞÜNCELER TARİHİ	Zorunlu
8	3	İKT110 2	İKTİSAT TARİHİ	Zorunlu
Toplam 18				

Liste 2 - 2003-2004,Bahar Eğitim-Öğretim Programındaki Seçmeli Dersler					
Kredi Kod	Kısım	Ders	Kontenjan	Sınıf	Türü
1	3	GSR190 1	0	1	Seçmeli
2	0	YDB144 1	0	1	Seçmeli
3	3	GSM130 1	0	1	Seçmeli
4	3	BSD170 1	0	1	Seçmeli
5	3	GSR200 1	0	1	Seçmeli

Şekil 5.8 Beşinci aşama ekranı



Şekil 5.9 Beşinci aşama sayfa yapısı

### 5.6.6 Altıncı aşama: alınan ders listesinin görüntülenmesi

Bu ekran, bilgilendirme amaçlıdır ve öğrencinin en son ders listesinin görüntülediği kısımdır. Son aşama olan altıncı aşama ekranı Şekil 4.10'da, sayfa yapısı da Şekil 4.11'de verilmiştir.

Genel Not Ortalaması	: 0.55
Alınan Toplam Kredi	: 18
Alınabilecek Max. Kredi	: 18

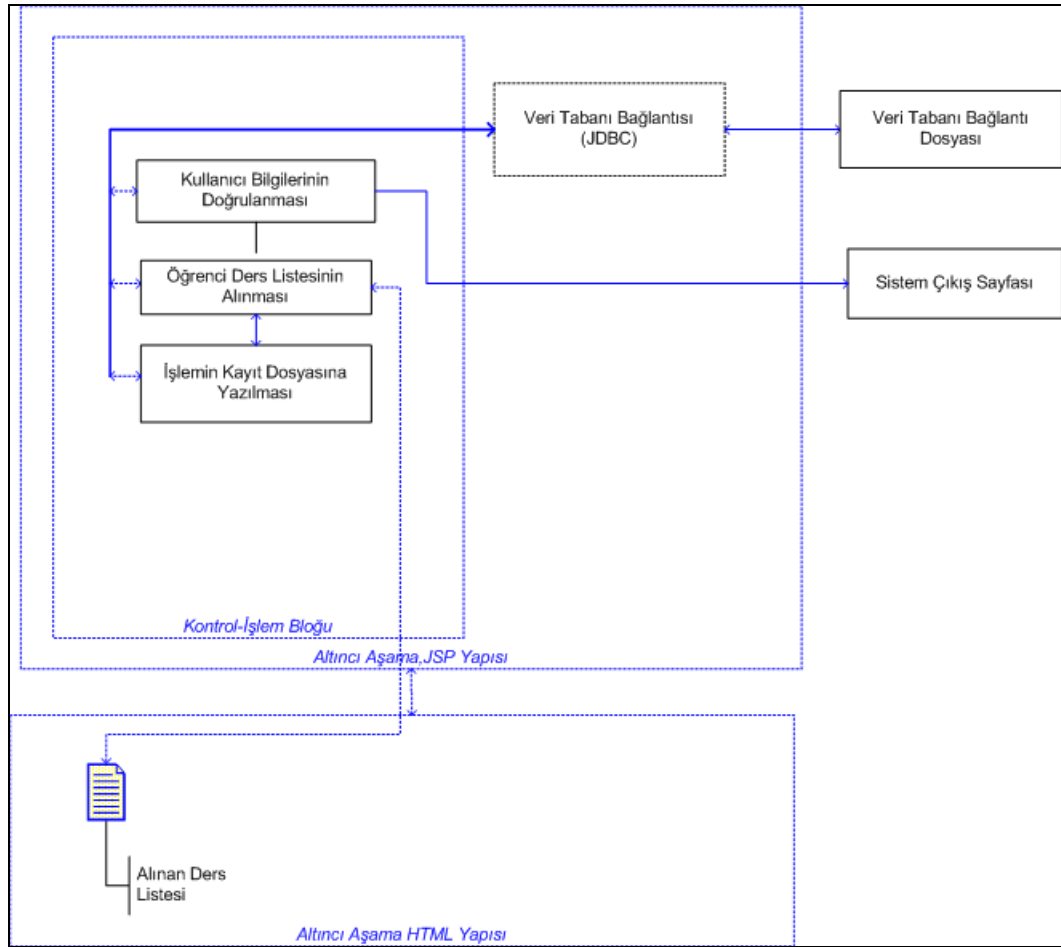
Liste 1 - 2003-2004, Bahar dönemi öğrenci ders kayıt listesi (Şu ana kadar aldığınız dersler)

Kredi	Kod	Ders
1	0	TDB104 TÜRK DİLİ II
2	0	YDB134 İNGİLİZCE II--
3	0	YDB194 INTERMEDIATE ENGLISH II--
4	3	İKT102 İKTİSADA GİRİŞ-II-
5	3	KAY122 ANAYASA HUKUKU
6	3	İŞL142 BORÇLAR HUKUKU
7	3	MAT118 MATEMATİK II
8	3	KAY114 SİYASAL DÜŞÜNCELER TARİHİ
9	3	İKT110 İKTİSAT TARİHİ
<b>Toplam</b>		

Derslerle ilgili kayıt işleminiz tamamlanmıştır. Almış olduğunuz derslerin kontrolünü danışmanınızla birlikte yapmayı unutmayın!

[Devam](#)

Şekil 5.10 Altıncı aşama ekranı



Şekil 5.11 Altıncı aşama sayfa yapısı

## 5.7 Uygulama Sunucusu

Yazılmış olan JSP sayfaları yorumlayıp kullanıcılara aktaran birçok JSP Sunucu bulunmaktadır. Bu sunucuların pek çok ortak noktası vardır. Öğrenci kayıt sistemi için kullanılan Web Sunucu, BEA firmasına ait olan Weblogic 8.1 sunucusudur.

Weblogic Server birçok farklı işletim sistemine kurulabilen bir sunucudur. Tablo 5.12’de bir önceki sürümü olan 7.0 ve güncel sürüm olan 8.1.x ile ilgili donanım ve sistem gereksinimleri görülmektedir.

Tablo 5.12 Weblogic Server kurulumu için donanım ve işletim sistemi gereksinimleri (BEA Product Documentation, 2005)

<b>İşletim sistemleri ve Donanımları</b>	<b>7.0.x</b>	<b>8.1.x</b>
HP NonStop™ G06.20 on MIPS	x	x
HP OpenVMS 7.3, 7.3-1 on Alpha	x	x
HP Tru64 UNIX 5.1 on Alpha	x	x
HP-UX 11.0, 11i on PA-RISC	x	x
HP-UX 11i on Itanium	x	x
IBM AIX 4.3.3 on PowerPC, POWER 3 and POWER 4	x	x
IBM AIX 5.1 on PowerPC, POWER 3 and POWER 4	x	x
IBM AIX 5.2 on PowerPC, POWER 3 and POWER 4	x	x
Microsoft Windows 2000 Server, Advanced Server on Pentium	x	x
Microsoft Windows 2000 Professional on Pentium	x	x
Microsoft Windows Server 2003 Standard, Enterprise, and Datacenter on Pentium	x	x
Microsoft Windows Server 2003 Enterprise and Datacenter on Itanium		x
Microsoft Windows NT 4.0 on Pentium	x	
Microsoft Windows XP on Pentium	x	x
Red Hat Enterprise Linux 2.1 AS, ES on Pentium	x	x
Red Hat Enterprise Linux 2.1 WS on Pentium	x	x
Red Hat Enterprise Linux 2.1 AS, ES on Itanium		x
Red Hat Enterprise Linux 3.0-1 AS, ES on Pentium		x
Red Hat Linux Advanced Server 2.1 on eGenera BladeFrame System	x	x
Red Hat Linux 7.2 on Pentium	x	

Tablo 5.12 (devamı)

İşletim sistemleri ve Donanımları	7.0.x	8.1.x
Red Hat Linux 7.2 on IBM zSeries/S390	x	
SCO UnixWare 7.1.3 on Pentium	x	
SCO UnixWare 7.1.4 on Pentium		x
Sun Solaris 7 on SPARC	x	
Sun Solaris 8 on SPARC	x	x
Sun Solaris 9 on SPARC	x	x
Sun Solaris 9 on x86		x
SuSE Linux® Enterprise Server 7 on Pentium	x	
SuSE Linux® Enterprise Server 8 on Pentium		x
SuSE Linux® Enterprise Server 7 on IBM zSeries/S390®	x	
SuSE Linux® Enterprise Server 8 on IBM zSeries/S390®	x	x
SuSE Linux® Enterprise Server 8 on PowerPC, POWER4		x

Tablo 5.13’de de görüldüğü üzere Weblogic Server’in çalışması için gereken sistem bileşenleri de minimum düzeydedir.

Tablo 5.13 Weblogic Server sistemi için minimum sistem gereksinimleri (BEA Product Documentation, 2005)

Bileşen	Gereksinimler
Sistem platformu	Tablo 4.12’de belirtilmiş olan sistemlerden bir tanesi yeterlidir.
Hard disk alanı	Windows işletim sisteminde ana kurulum için 445 MB boş alan ve kurulum için 275 MB geçici boş alan gerekmektedir. UNIX işletim sistemi grubunda ana kurulum için 250 MB ve kurulum için yaklaşık 275 MB boş alan gerekmektedir.
Hafıza	Minimum 512 MB RAM’a ihtiyaç duyulur.
Renk	Grafik kurulum için 256 renk yeterlidir. Konsol kurulumu için renk gereksinimi yoktur.
Java 2 SDK	Kurulumla birlikte gelmektedir.

### 5.7.1 Weblogic sunucusu kurulum aşamaları

Weblogic sunucu kurulumu basit ve kullanıcı dostu bir arabirime sahiptir. İlk aşama “hoş geldiniz” ekranıdır. İkinci aşama lisans sözleşmesi anlaşmasının bulunduğu aşamadır. Kabul edilip ileri butonuyla diğer aşamaya geçilir. Üçüncü aşama kurulum klasörünün belirlendiği aşamadır. Standart olarak “C:\BEA” klasörü kullanılır. Dördüncü aşama kurulum türü seçiminin yapıldığı aşamadır. Burada kullanıcıya “Complete” ve “Custom” seçenekleri sunulur. Eğer tüm örnekler ve yardım dosyaları ile birlikte kurulum yapmak istenirse “Complete” seçilir. Program grubundaki her bir nesne ayrı ayrı kurulumu seçilecekse “Custom” seçeneği ile kurulum yapılır. Beşinci aşamada dosyalar kopyalandıktan sonra sunucunun otomatik başlayıp başlamamasına karar verilir.

Kurulumun tamamlanmasından sonra sunucu başlatılıp yönetici modülüne geçilerek sunucu sistemi tanımlamaları yapılır. Sunucu sistem ayarlarını yapmak için sunucu başlatıldıktan sonra programın kurulduğu gruptaki “Configuration Wizard” programı yardımı ile istenen özellikte konfigürasyon tanımı yapılır. Tanımlamalar bittikten sonra yönetici konsolu tarayıcıdan açılıp web modül yüklenebilir.

### 5.7.2 Weblogic sunucusuna SSL kurulumu

SSL kurulumu için Java platformunun “keytool” adlı programından yararlanır. SSL için kullanılacak anahtarları içeren dosyalar keytool yardımcı programı ile oluşturulur.

- Keytool ile sunucu anahtarı ve sertifika oluşturulur.
- Weblogic yönetim konsolundan oluşturulan sertifika tanımları yapılır. Sertifika tanımı yapılırken oluşturulan sertifika türü ve şifreleme yapısı yönetim konsolunda belirtilmelidir.
- Değişiklikler Weblogic konsolundan uygulanır ve konfigürasyon kaydedilir.

Bu aşamalar gerçekleştirildikten sonra sunucu yeniden başlatıldığında artık SSL aktif durumda olacak şekilde sunucu çalışacaktır.

## **5.8 Güvenlik Tanımı**

Aşamalar içerisindeki her sayfada kullanıcı bilgilerinin doğruluk kontrolü yapılmaktadır. Kontrolde daha önceden veri tabanı yapısında saklanan kullanıcı bilgileri ile giriş yapan kullanıcıya ait bilgiler karşılaştırılır. Bunun dışında Javanın kendi güvenlik tanımlarından yararlanılarak güvenlik pekiştirilmiştir. Bu güvenlik katmanı, kullanıcı tanımı ve kullanıcıya ait kaynak tanımına dayanmaktadır. Bu işlem için web modülüne ait “web.inf” tanımlayıcı dosyasında kaynak tanımı ve kaynağa izin verilecek kullanıcı grubu tanımı yapılır.

## **5.9 Kayıt Sistemi Yapısının Düzenlenmesi ve Farklı Kayıt İşleyişlerine Uyarlanması**

MiKaS'ta kayıt işleyişi ve denetim mekanizmaları esnek ve değiştirilebilen bir yapıya sahiptir. Bu özelliğiyle farklı sistemlere uyarlanabilir. Bunun için yapılması gereken, aşamaları içeren sayfa yapılarında kullanılan SQL komutlarının ve denetim mekanizmalarının, kullanılan kayıt sistemi özelliğine göre değiştirilmesidir.

## 6. ARAŞTIRMA BULGULARI

MiKaS, mevcut öğrenci internet kayıt sistemlerinin incelenerek eksiklik ve meseleleri çözmeye yönelik geliştirilmiş bir kayıt sistemidir. Hâlihazırda Muğla Üniversitesi'nde kullanılmakta olup, kullanım oranı öğrencilerin bilgilendirmesine ve kullanım kolaylığının anlatılmasına bağlı olarak her dönem artmaktadır. Öğrenci kayıtları ile ilgilenen danışmanların, kayıt dönemi içerisinde yapılan kayıtları takibi sayesinde sistemdeki kayıtlar anlık olarak takip edilebilmektedir.

MiKaS, 2003-2004 güz döneminden buyana Muğla Üniversitesinde kullanılmaktadır. 2004-2005 Güz döneminde Muğla Üniversitesindeki toplam öğrenci sayısı 14864 iken, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 506, ders kaydı sayısı 1710'dur. 2004-2005 Bahar döneminde, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 1106'ya, ders kaydı sayısı da 3870'e çıkmıştır. Bu rakamlar bir önceki dönemin yaklaşık iki katıdır. Sistemin kullanım oranının artmasıyla kayıt dönemindeki yoğunluğun giderek azaldığı gözlenmiştir. Ayrıca el ile yapılan kayıtlardaki hataların (yanlış ders kısmı seçme veya kontenjanı dolu ders seçme gibi...) önüne geçilmiştir. İlerleyen dönemlerde kullanım oranının artırılması ile yoğunluk ve hataların azalması amaçlanmaktadır.

MiKaS değişik üniversitelerdeki kayıt sistemlerine uyarlanarak kullanım alanı bulabilir. Bunun için, geliştirilen modüldeki veritabanı bağlantı yapısı ve kontrol yapısının uygulanacak kayıt sistemine göre değiştirilmesi yeterli olacaktır.



## 7. SONUÇLAR ve TARTIŞMA

Sunulan kayıt sistemi sayesinde öğrenciler buldukları yerden kayıt yenileme ve ders kaydı işlemlerini yapabilmektedirler. Sistem üzerinden yapılan işlemler bir kayıt dosyasında tutulmaktadır. Bu kayıt dosyasının incelenmesi sonucunda hata oranının çok aza indirildiği görülmüştür. Benzer internet kayıt sistemlerinin incelenmesi sonucunda daha önce değinilen eksiklikler giderilmiştir. Ders kontenjanlarının düzenli kontrolü sağlanmıştır. İnternet üzerinden kayıt işlemi için öğrencilerin eğitimi ve teşviki ile kayıt yoğunluğu İnternete aktarıldığında kayıt dönemindeki hataların azaldığı ve danışman iş yükünün hafifletildiği gözlenmiştir.

MiKaS, dört dönemdir Muğla Üniversitesinde kullanılmaktadır. Her dönem öğrencilerin de bilgilendirilmesi sayesinde kayıt sisteminin kullanım oranı artmıştır. 2004-2005 Güz döneminde Muğla Üniversitesindeki toplam öğrenci sayısı 14864 iken, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 506, ders kaydı sayısı 1710'dur. 2004-2005 Bahar döneminde, MiKaS üzerinden kayıt yenileyen öğrenci sayısı 1106'ya, ders kaydı sayısı da 3870'e çıkmıştır. Bu rakamlar bir önceki dönemin yaklaşık iki katıdır. İnternet üzerinden kayıtlar incelendiğinde, normal kayıt tarihleri içerisinde karşılaşılan hata oranından çok daha az kullanıcı kaynaklı hatalara rastlanmıştır. Ayrıca kayıtlarla ilgilenen danışmanların ve öğrencilerin büyük zaman kazandıkları ortaya çıkmıştır. Öğrencilerin kayıtlarını İnternet üzerinden yapmalarına ağırlık verildiği takdirde kayıt haftasındaki yoğunluklar azalacak ve hata ve kontrol denetimleri sağlanmıştır olacaktır.

Saniyede yaklaşık 500 ders kaydı yapılabilmesi sistemde mümkündür. Bu sayı Bea Weblogic Server'ın yönetim modülündeki izleme ekranından elde edilmiştir. Anlık yapılabilecek işlem kapasitesi artırılmak istenirse sunucuların kapasitelerinin artırımına gidilebilir veya kümeleme yapısındaki sunucu sayısı artırılabilir.

Öğrenci bilgileri dönem içerisinde doğru işlenir ve takibi doğru yapılırsa MiKaS'ta işleyiş bakımından sorunla karşılaşılmamaktadır. MiKaS teknik altyapısında kullanılan veri güvenliği ve kümeleme teknolojileri sayesinde de teknik problemlerden kaynaklı aksaklıkların olabildiğince önüne geçilmiştir.

## KAYNAKLAR

Adar N., Osman Gazi Üniversitesi Öğrenci İşleri Daire Başkanlığı.

Akgöbek Ö., Harran Üniversitesi Bilgi İşlem Daire Başkanı.

Altıntaş, A.B., Java Neden İyidir?, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bahar Etkinlikleri, 22 Mayıs 2004, Kocaeli.

Yiğit B., Ondokuz Mayıs Üniversitesi Öğrenci İşleri Daire Başkanlığı.

BEA Product Documentation, <http://e-docs.bea.com>, Ocak 2005.

Bingöl A., Fırat Üniversitesi Bilgi İşlem Dairesi.

Chan P., Java Developers Almanac, <http://www.javaalmanac.com>; Ocak 2005.

Çakar F., Dicle Üniversitesi Bilgi İşlem Dairesi.

Eckel B., 1998, Thinking in Java, Newjersey, 860p.

Güven A. S., Osmangazi Üniversitesi Öğrenci İşleri Daire Başkanlığı.

Java 2 Platform Security, <http://java.sun.com/j2se/1.3/docs/guide/security>, Ocak 2005.

Java 2 SDK, Enterprise Edition Documentation Bundle, [http://java.sun.com/j2ee/sdk\\_1.2.1/techdocs](http://java.sun.com/j2ee/sdk_1.2.1/techdocs), Ocak 2005.

Java API Specifications, <http://java.sun.com/reference/api/index.html>, Ocak 2005.

Java Database Connectivity, <http://java.sun.com/products/jdbc/>, 2005.

Java Documentation, <http://docs.sun.com/app/docs>, Ocak 2005

Java Reference Documentation, <http://java.sun.com/reference/docs/index.html>, Ocak 2005.

Java Security, <http://java.sun.com/j2se/1.3/docs/guide/security>, Ocak 2005.

Java Technical Articles and Tips, <http://java.sun.com/reference/techart/index.html>, Ocak 2005.

Erkan Y., J2EE İmparatorluğu – Teknolojiler, <http://www.teknoturk.org/docking/yazilar/tt000051-yazi.htm>, Ocak 2005.

Hunter, J., Crawford W., 2001, Java Servlet Programming 2nd Edition, Sebastopol, California, 780p.

Kılıçarslan, H.G., ÜNİPA, ÜNİPA Proje Koordinatörü.

Kutlu N., Muğla Üniversitesi Öğrenci İşleri Daire Başkanlığı, Ocak 2005.

Oaks S., 2001, Java Security 2nd Edition, California, 618p.

Öncü Z., Süleyman Demirel Üniversitesi Bilgi İşlem Dairesi.

Özcan E.,2003, Türkiye'deki Üniversitelerde İnternet Tabanlı Akademik Kayıt ve Not Takip Sistemleri.

Özdemir A., Cumhuriyet Üniversitesi Bilgi İşlem Dairesi.

Public CERN Java Infrastructure, <http://java.web.cern.ch/java/>, Ocak 2005.

Software Development Times, <http://www.sdtimes.com/news/092/story5.htm>, Ocak 2005

SQL Server Home, <http://www.microsoft.com/sql>, Ocak 2004

Types of JDBC technology drivers ,  
<http://java.sun.com/products/jdbc/driverdesc.html>, Ocak 2005.

Yıldız B., Atatürk Üniversitesi Bilgi İşlem Dairesi.

Yükseköğretim Kurulu Başkanlığı, <http://www.yok.gov.tr>, Ocak 2005

## **ÖZGEÇMİŞ**

### **KİŞİSEL BİLGİLER**

Adı Soyadı : Kürşat Kurt

Doğum Yeri : YOZGAT

Doğum Tarihi : 1978

Medeni Hali : Bekâr

### **EĞİTİM VE AKADEMİK BİLGİLER**

Lise : 1991-1994 Mehmet Akif Ersoy Lisesi/KIRŞEHİR

Lisans : 1995-2000 Muğla Üniversitesi Fen-Edebiyat Fakültesi İstatistik ve  
Bilgisayar Bilimleri

Yabancı Dili : İngilizce

### **MESLEKİ BİLGİLER**

Mezuniyetimin ardından Muğla Üniversitesi Enformatik Bölümü'nde Okutman olarak göreve başladım. 2001 yılında yüksek lisans programına başladım. Aynı zamanda Muğla Üniversitesi Bilgi İşlem Daire Başkanlığı'nda görevlendirilerek öğrenci kayıt sistemi projesi olan Dijital Üniversite Projesi'nin yürütülmesi görevine başladım. Muğla Üniversitesi Beden Eğitimi ve Spor Yüksekokulu Sınav Değerlendirme Yazılımı'nın geliştirilmesinde yer aldım. Halen Enformatik Bölümü'nde bilgisayar derslerine girmekteyim ve Muğla Üniversitesi Bilgi İşlem Daire Başkanlığı'nda, Dijital Üniversite Projesinin yürütülmesi görevlerimi sürdürmekteyim.