

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SUNUCU TARAFLI PROGRAMLAMA DİLLERİNDE
FONKSİYONEL DİL İLE NESNE TABANLI
PROGRAMLAMA DİLLERİNİN KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Özgür ÖRNEK

Enstitü Anabilim Dalı : ELEKTRONİK BİLGİSAYAR EĞİTİMİ

Tez Danışmanı : Yrd. Doç. Dr. A. Turan ÖZCERİT

Eylül 2006

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SUNUCU TARAFLI PROGRAMLAMA DİLLERİNDE
FONKSİYONEL DİL İLE NESNE TABANLI
PROGRAMLAMA DİLLERİNİN KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Özgür ÖRNEK

Enstitü Anabilim Dalı : ELEKTRONİK BİLGİSAYAR EĞİTİMİ

Bu tez .. / .. /2006 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

**Prof. Dr. Hüseyin
EKİZ
Jüri Başkanı**

**Yrd. Doç. Dr. A. Turan
ÖZCERİT
Üye**

**Yrd. Doç. Dr. İbrahim
ÖZÇELİK
Üye**

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Yrd. Dç. Dr. Ahmet Turan Özcerit'e teőekkürü bir borç bilirim.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vii
TABLOLAR LİSTESİ.....	viii
ŞEKİLLER LİSTESİ.....	ix
ÖZET.....	xi
SUMMARY.....	xii

BÖLÜM 1.

GİRİŞ.....	1
------------	---

BÖLÜM 2.

JAVA WEB UYGULAMA GELİŞTİRME	6
2.1. Web Uygulama Terimleri.....	6
2.2. Web Uygulamaları Nasıl Çalışır?.....	7
2.3. Durağan Web Sayfalarının Çalışması.....	7
2.4. Dinamik Web Sayfalarının Çalışması.....	8
2.5. Java Web Uygulamalarının Çalışması.....	9
2.6. JSP Teknolojisi.....	10
2.6.1. JSP mimarisi.....	11
2.6.2. JSP dilinin hayat döngüsü.....	13
2.7. Servlet Teknolojisi.....	14
2.7.1. Etkin.....	14
2.7.2. Uygunluk.....	14
2.7.3. Güçlü.....	14
2.7.4. Taşınabilir.....	14
2.8. Servlet Mimarisi.....	15

2.9. Servlet'in Çalıştırılması.....	15
2.10. Servlet Metodları.....	16
2.11. JSP ve Servlet'in Birarada Kullanılması.....	16
2.12. Servlet ve JSP Model Yapıları.....	17
2.12.1. Model 1 Mimarisi.....	17
2.12.2. Model 2 Mimarisi.....	18

BÖLÜM 3.

PHP VE MYSQL İLE WEB YAZILIMI GELİŞTİRME.....	19
3.1. PHP Nedir ve Nasıl Çalışır?	19
3.2. PHP Dilinin Avantajları.....	20
3.2.1. Ücretsiz lisans	20
3.2.2. Kolay ve hızlı kullanım.....	20
3.2.3. Güncel veritabanları ile uyumlu ve etkili çalışma.....	21
3.2.4. Performans.....	21
3.2.5. Güvenli ve güncel.....	22
3.2.6. Yaygın işletim sistemlerine uyumluluk.....	22
3.2.7. Başarılı açık kod projesi.....	23
3.3. PHP' nin Temelleri.....	23
3.4. MySql ve PHP.....	24
3.5. PHP ile SQL kullanımı.....	26
3.6. MySQL Yönetimi.....	27

BÖLÜM 4.

ONLINE MALZEME STOK TAKİP PROGRAMI.....	28
4.1. Programın Tanımı.....	28
4.2. Kullanıcı Modülü.....	28
4.2.1. Login olma.....	29
4.2.2. Malzeme sayfası.....	29
4.2.3. Malzeme seçimi.....	30
4.2.4. Malzeme detayı.....	31
4.2.5. Malzemeyi seçme.....	31
4.2.6. Seçim işleminin bitirilmesi.....	31
4.2.7. Malzeme arama.....	32

4.2.8. Yeni kullanıcı ekleme.....	33
4.3. Admin Modülü.....	33
4.3.1. Admin arayüzü.....	33
4.3.2. İşlem onaylama.....	34
4.3.3. İşlem detayı.....	35
4.3.4. Kullanıcı onaylama.....	35
4.3.5. Malzeme işlemleri.....	36
4.3.6. Onaylanmış işlemler.....	36
4.3.7. Onaylanmış mullanıcılar.....	37
4.3.8. Şifre değiştirme.....	38
BÖLÜM 5.	
PROGRAMIN JSP VE PHP DİLİNDE GELİŞTİRİLMESİ.....	39
5.1. Veritabanı Tasarımı ve Tabloları.....	39
5.1.1. Program varlık-ilişki (ER) modeli.....	39
5.1.2. Programda kullanılan tablolar.....	40
5.1.3. Malzeme database SQL scripti.....	41
5.2. Programın JSP Dilinde Geliştirilmesi.....	43
5.2.1. JSP mimarisi.....	43
5.2.2. Programın sınıf (class) diyagramları.....	44
5.2.3. Programın sınıf kodları.....	47
5.2.3.1. Malzeme sınıfı.....	47
5.2.3.2. İşlem sınıfı.....	49
5.2.3.3. Cart sınıfı.....	50
5.2.3.4. Kullanıcı sınıfı.....	52
5.2.3.5. Satır sınıfı.....	55
5.2.3.6. Admin sınıfı.....	55
5.2.4. Program altyapısı.....	56
5.2.4.1. GirişServlet (Controller).....	58
5.2.4.1.1. GirişServlet sınıf kodları.....	58
5.2.4.1.2. GirişServlet sınıfının çalışması.....	61
5.2.4.2. Kullanıcı sınıfı (Model).....	64
5.2.4.3. Sepet.jsp (Görünüm).....	64

5.2.4.3.1. TLD dosyası.....	66
5.2.4.3.2. Tag sınıfı.....	67
5.3. Uygulamanın PHP Dilinde Geliştirilmesi.....	69
5.3.1. PHP kodunun HTML imleri ile birlikte kullanılması.....	70
5.3.2. PHP dilinde fonksiyon kullanımı.....	72
5.3.3. MySQL veritabanına bağlanma.....	75
5.3.4. Session kullanımı.....	77
BÖLÜM 6.	
TEZ ANALİZ.....	81
6.1. Webserver Stress Tool Programının Tanıtımı.....	81
6.1.1. Test bölümü.....	82
6.1.1.1. Test türü.....	82
6.1.1.2. Kullanıcı simülasyonu.....	82
6.1.2. URL seçimi.....	83
6.1.3. Test sonuçları bölümü.....	84
6.1.3.1. Log files.....	84
6.1.3.2. Graphs.....	85
6.2. Sonuçların Analizi.....	86
6.2.1. 100 kullanıcıya kadar sayfanın yüklenmesi.....	87
6.2.2. 500 kullanıcıya kadar sayfanın yüklenmesi.....	88
BÖLÜM 7.	
SONUÇLAR VE ÖNERİLER.....	89
KAYNAKLAR.....	91
ÖZGEÇMİŞ.....	93

SİMGELER VE KISALTMALAR LİSTESİ

API	: Application Programming Interface
CGI	: Common Gateway Interface
COM	: Component Object Model
DBMS	: Database Management System
EJB	: Entity Java Bean
ER	: Entity Relationship
IIS	: Internet Information Services
JDBC	: Java Database Connectivity
J2EE	: Java 2 Enterprise Edition
J2SE	: Java 2 Standart Edition
JSP	: Java Server Page
JVM	: Java Virtual Machine
MVC	: Modem View Controlller
PHP	: Personal Home Page
SDK	: Standart Development Kit
SQL	: Structured Query Language
UML	: Unified Modelling Language
VTYS	: Veritabanı Yönetim Sistemi
W3	: World Wide Web

TABLULAR LİSTESİ

Tablo 1.1. Lomerson ve diğ. araştırma sonuçları.....	3
Tablo 1.2. Birim zamandaki oturum sayısı.....	4
Tablo 3.1. 2006 Ağustos ayında web sunucu kullanım sayısı ve oranı.....	21
Tablo 3.2. ASP ve PHP nin karşılaştırılması.....	22

ŞEKİLLER LİSTESİ

Şekil 2.1. Web Uygulama Çalışma Şekli.....	7
Şekil 2.2. Durağan Web Sayfa Çalışması	8
Şekil 2.3. Dinamik web sayfası çalışma prensibi.....	9
Şekil 2.4. Java Web Uygulama Çalışma Yapısı.....	10
Şekil 2.5. JSP örnek üye gösterme sayfası.....	11
Şekil 2.6. uyeGoster.jsp sayfası kod bloğu.....	13
Şekil 2.7. JSP Hayat Döngüsü.....	14
Şekil 2.8. Servlet ve JSP çalışma yapısı.....	15
Şekil 2.9. Servlet Çalışması.....	16
Şekil 2.10. Model 1 mimari.....	17
Şekil 2.11. MVC (Model 2) mimari.....	18
Şekil 3.1. PHP script dilinin çalışma şeması	20
Şekil 3.2. 2006 Ağustos tarihine kadar olan PHP kullanım istatistiği.....	20
Şekil 4.1. Online malzeme stok takip programı kullanıcı girişi arayüzü.....	29
Şekil 4.2. Online malzeme stok takip programı malzemeler sayfası arayüzü	30
Şekil 4.3. Online malzeme stok takip programı malzeme grupları arayüzü....	30
Şekil 4.4. Malzeme detay arayüzü.....	31
Şekil 4.5. Malzeme seçme işleminin bitirilmesi.....	32
Şekil 4.6. Malzeme arama.....	32
Şekil 4.7. Yeni kullanıcı ekleme.....	33
Şekil 4.8. Admin arayüzü.....	34
Şekil 4.9. İşlem onaylama.....	34
Şekil 4.10. İşlem onay detayı.....	35
Şekil 4.11. Kullanıcı onaylama.....	36
Şekil 4.12. Admin malzeme işlemleri menüsü ve yeni malzeme ekleme.....	36
Şekil 4.13 Onaylanmış işlemler.....	37
Şekil 4.14. Onaylanmış kullanıcılar.....	37

Şekil 4.15. Onaylanmış kullanıcı detayı.....	38
Şekil 4.16. Şifre değiştirme.....	38
Şekil 5.1. ER şemasına göre yeni tablolar ve alanları.....	39
Şekil 5.2. Kullanici Tablosu.....	40
Şekil 5.3. Malzeme Tablosu.....	40
Şekil 5.4. İşlem Tablosu.....	40
Şekil 5.5. IslemDetay Tablosu.....	41
Şekil 5.6. Admin Tablosu.....	41
Şekil 5.7. Malzeme veritabanı ve tablo oluşturma scripti.....	42
Şekil 5.8. Kullanıcı mimarisi.....	43
Şekil 5.9. Admin Mimarisi.....	44
Şekil 5.10. Kullanici sınıf diyagramı.....	45
Şekil 5.11. Malzeme Sınıf Diyagramı.....	45
Şekil 5.12. Islem sınıf diyagramı.....	46
Şekil 5.13. Cart sınıf diyagramı.....	46
Şekil 5.14. Satir sınıf diyagramı.....	47
Şekil 5.15. Programda MVC modeli kullanımı.....	64
Şekil 6.1. Webserver stress tool programı arayüzü.....	81
Şekil 6.2. Webserver stress tool programı test türü bölümü.....	82
Şekil 6.3. Webserver stress tool programı kullanıcı simülasyonu bölümü...	83
Şekil 6.4. Webserver stress tool programı URL seçme ve kaydetme.....	83
Şekil 6.5. Webserver stress tool programı URL recorder (kaydedici).....	84
Şekil 6.6. Webserver stress tool programı Log files bölümü.....	85
Şekil 6.7. Webserver stress tool programı Graphs bölümü.....	86
Şekil 6.8. Birim zamanda 100 kullanıcının JSP ve PHP sayfalarına istekte bulunduğu test sonucu.....	87
Şekil 6.9. Birim zamanda 500 kullanıcının JSP ve PHP sayfasına istekte bulunduğu test sonucu.....	88

ÖZET

Anahtar Sözcükler: JSP, Servlet, PHP, Sunucu Tarafli Programlama

Internet'in günümüzde neredeyse her alanda kullanılması kaçınılmaz olarak herkesin bu teknolojiye ilgi duymasını sağlamıştır. Özellikle online alışveriş, bankacılık, online rezervasyonlar gibi işlemler çok yaygın kullanılmaktadır. Bu işlemler ise güvenlik gerektiren sunucu tarafli programlama teknolojinin kullanılması ile gerçekleştirilmektedir.

Bu çalışmada, yaygın kullanılmakta olan sunucu tarafli programlama dillerinin karşılaştırılması amaçlanmıştır. Fonksiyonel ve nesne tabanlı dillerinin yaygın kullanımı nedeniyle bu iki teknolojinin karşılaştırılması tercih edilmiştir. Bu amaçla bir online malzeme stok takip projesi hem PHP hem de JSP teknolojisi ile tasarlanmıştır. Her iki ayrı projede yapılan testler sonucunda fonksiyonel ve nesne tabanlı dillerin sunucu üzerinde çalışma performanslarının ölçülmesi gerçekleştirilmiştir.

COMPARING FUNCTIONAL PROGRAMMING AND OBJECT ORIENTED PROGRAMMING LANGUAGES ON THE SERVER SIDE PROGRAMMING

SUMMARY

KEYWORDS: JSP, Servlet, PHP, Server Side Programming, MVC

Nowadays, everybody is interested in Internet technology because of Internet is being used almost every field. Especially, online purchase, banking, online reservations etc. processes is commonly used. To use these processes is required server side programming.

In this area, server side programming languages which are commonly used are aimed to compare them. Functional languages and object oriented languages are compared because they are being used in common.

An online material stock tracing project was developed using PHP and JSP technologies. Each project was tested and the performance of the object oriented and functional programming languages have been simulated.

BÖLÜM 1. GİRİŞ

Internet kullanımının yaygınlaştığı ilk yıllarda, web sayfaları durağan olarak hazırlanmaktaydı. Web sunucusundan, istemci (client) tarayıcılarına (browser) gönderilen içerik, tüm istemciler için aynıydı. Durağan web sayfalarında, web sitesi ziyaretçileri hakkında bilgi toplamak, ziyaretçi tercihlerini öğrenmek gibi sunucu ile ziyaretçi arasındaki etkileşim mümkün değildi. Bu ihtiyacı çözmek için 1990'lı yıllardan itibaren sunucu taraflı programlama(server side programming) dilleri geliştirilmeye başlandı. Bu amaçla ilk olarak CGI (Common Gateway Interface) programlama dili geliştirildi. CGI dilinin kullanımının zorluğu nedeniyle daha kolay ve anlaşılır diller olan Perl, ASP, PHP, Cold Fusion gibi farklı firmaların geliştirdiği fonksiyonel diller kullanılmaya başlandı. Sonraki yıllarda nesne tabanlı programlama (Object Oriented Programming) dilinin gücü ve kolaylıkları ile birlikte Sun firması tarafından Java Servlet ve JSP teknolojisi ile Microsoft firması tarafından ASP.NET teknolojisi geliştirildi. Günümüzde fonksiyonel dil olarak PHP ve ASP, nesne tabanlı dil olarak da Java Servlet ve JSP ile ASP.NET teknolojisi yaygın bir şekilde kullanılmaktadır.

Sunucu taraflı çalışan programlama dilleri temelde aynı işlevi görür. Ancak performans, hız, kolaylık ve ekonomiklik bu dilleri tercih etmede önemli faktörlerdir. Bu nedenle sunucu taraflı bir teknolojiyi öğrenmek ve kullanmaya karar vermek işgücü ve maddi açıdan bir yatırım demektir. Birbirine alternatif olan bu teknolojilere yatırıma karar vermeden önce iyi bir plan yapılmalıdır.

Bu tez çalışmasında, sunucu taraflı programlama dilleri içinde fonksiyonel programlama ve nesne tabanlı programlama dillerinin çeşitli açılardan karşılaştırılması yapılmıştır. Böylece sunucu taraflı programlama dilleri içinde fonksiyonel veya nesne tabanlı dil seçmek isteyenlere yardımcı olmak amaçlanmıştır.

Bu tez çalışmasının oluşturulması sırasında fonksiyonel dil olarak PHP programlama dili kullanılmıştır. Sunucu taraflı uygulama geliştirmede kullanılan diller arasında PHP dilinin yaygın kullanımı, giderek yükselen grafiği ile dikkat çekmekte ve gelecek vaat etmektedir. Küçük ve orta ölçekli uygulamalar geliştirmedeki pratikliği, sözdizimi ve öğrenilmesindeki kolaylığı, çoğu platformda çalışabilmesi ve ücretsiz olmasından dolayı çoğu hosting firması tarafından desteklenmektedir.

Nesne tabanlı dil olarak Java Servlet ve JSP kullanılmıştır. Java sanal makinası (JVM) ile her platformda çalışabilmesi, güvenli, esnek ve kararlı bir dil olması ve hızla artan kullanımı nedeniyle Java Servlet ve JSP dili tez çalışmasında tercih edilmiştir.

Nesne tabanlı dil ile nesne tabanlı olmayan dillerin kullanımının karşılaştırılması amacıyla Johnson[1] tarafından 2000 yılında her iki dili kullananların araştırması yapılmış ve her iki dilin üstünlük ve zayıflıkları ortaya konmuştur. Benzer şekilde Henderson-Sellers[2] kompleks uygulamalarda nesne tabanlı dillerin daha başarılı olduğunu ortaya koymuştur.

Roberts[3] tarafından yapılan başka bir çalışmada, PHP, ASP, ASP.NET, ColdFusion teknolojileri incelenmiş ve bu teknolojilerin avantaj ve dezavantajları ortaya konmuştur. Bu teknolojilerin tercih edilmesinde, sunucu taraflı dillerin çalıştığı işletim sistemine uyumluluk, ekonomiklik, gelecek vaad edip etmemesi ve programın kolay yazılıp yazılmaması faktörlerinin etkili olduğu sonucu çıkarılmıştır.

Lomerson ve diğ.[4] ise e-ticaret sitelerinin geliştirilmesinde kullanılan sunucu tabanlı dilleri incelemiştir. Bunun için yazılım sektöründeki önde gelen 423 yazılım geliştirme firmasına anket düzenlenmiştir. Bu anket neticesinde Tablo 1.1.'deki sonuçlara ulaşılmıştır.

Tablo 1.1. Lomerson ve diğ. araştırma sonuçları

Sıra	Teknoloji
1	JSP
2	ASP
3	Perl
4	PHP
5	ColdFusion

Bu bilgilerin analizine göre sunucu taraflı dillerin seçiminde kullanılan işletim sistemi önemli bir etken olduğu sonucunu çıkarmışlardır. Microsoft işletim sistemlerinde ASP ve ADO, linux ve unix işletim sisteminde ise sunucu pazarını elinde bulunduran Apache sunucuyu kullanan JSP ve Perl dillerinin kullanılmasının daha iyi seçim olacağını belirtmişlerdir.

Vervae[5] ise Java dilinin yavaş ve uyumsuz olmasından dolayı kötü bir dil olduğunu düşünenlere karşı Java dilinin güçlü bir dil olduğu ve kolay bir yazım kuralına sahip olduğunu belirtmiştir. Java dilinin, platformdan bağımsız bir dil olmasının da aynı kodun bütün işletim sistemlerinde çalıştırılmasına imkan tanıdığını belirtmiş, Java sanal makinası (Java virtual machine) ile bellek kullanımında büyük tasarruflar sağlamasıyla diğer dillere üstünlüğü olduğunu savunmuştur. Java'nın asıl üstün olduğu alanın ise web teknolojileri olduğunu ve Java dilinin web programlama için tercih edilmesi gerektiğini önermiştir.

Morrison[6] ise web tabanlı dilleri çeşitli kategorileri ayırmıştır. Morrison'a göre Web tabanlı uygulamalarda kullanılacak birçok dil bulunmaktadır. Bu dillerden hangisinin seçileceği istekte bulunandan çok yazılım geliştiricilerin düşünmesi gereken bir durumdur. Kullanılacak yazılım ortamlarından biri ASP.Net teknolojisidir. Diğer taraftan bu tür uygulamalarda oldukça yoğun bir şekilde JSP kullanımına da rastlanmaktadır. Bununla beraber web uygulamalarında kullanılan kaynaklar genel olarak Java, C++, Visual Basic dilleri kullanılarak geliştirilmektedir.

Sunucu tabanlı uygulamalarda kullanılan protokollere bağlı olarak bazı durumlarda CGI kullanılabilir. CGI kullanımının en büyük nedeni CGI'nin sunucu bağımsız çalışabiliyor olmasıdır.

Sunucunun hizmet alanı çok geniş değilse, kısa süre içinde çok fazla işlemin gerçekleştirilmesi gerekmiyorsa sunucu taraflı çalışan betiklerin (script) kullanılması düşünülmelidir. Bu durumlar için kullanımı ve anlaşılması en kolay ortamlar olarak Cold Fusion ve ASP betikleri düşünülmektedir.

Web programcılığında kullanılan dillerin hangisinin daha hızlı ve uyumlu olduğu konusunda Apte, Hansen ve Reeser[7] de bir çalışma yapmıştır. Bu çalışmada, CGI ve Java Servlet-JSP dillerinin performans karşılaştırmaları yapılmıştır. Elde edilen sonuçlara göre CGI'in Java Servlet-JSP'e göre daha hızlı olduğuna karar verilmiştir. Bu kararın alınmasında kullanılan analizlerin bir tanesinin sonucu Tablo 1.2'deki gibidir.

Tablo 1.2. Birim zamandaki oturum sayısı

Teknoloji	Oturum/saniye
Java Servlet	0.6
JSP	0.8
CGI	2.0
Fast CGI	2.5

Ayrıca web sunucularını performanslarının karşılaştırılması amacıyla, Rubinstein[8] tarafından Apache ve Microsoft IIS testleri yapılmış ve Apache'nin %40 Microsoft IIS'den daha hızlı olduğu sonucunu ortaya koymuştur.

PHP ve JSP dilleri kullanılarak çeşitli uygulamalar üzerinde testler ölçümler yapılmış olmasına rağmen nesne tabanlı ve nesne tabanlı olmayan dillerin karşılaştırılması yapılmamıştır.

Bu tez çalışmasında nesne tabanlı dil olan Java Servlet ve JSP dilleri ile nesne tabanlı dil olmayan PHP 4.0 dilinin performanslarının karşılaştırılması yapılmıştır. Bu amaçla malzeme stok programı her iki dilde yapılmıştır. Hız, kolaylık, güvenlik, ekonomiklik, taşınabilirlik gibi faktörler açısından her iki dilin performansları karşılaştırılmıştır.

Her iki dilde veritabanı işlemlerini gerçekleştirmek için yaygın kullanımı ve ücretsiz lisansı nedeniyle MYSQL veritabanı seçilmiştir. Web sunucu olarak yaygın kullanımı ve hızlı performansı nedeniyle PHP dili için Apache server, JSP dili için ise Apache Tomcat kullanılmıştır.

BÖLÜM 2. JAVA WEB UYGULAMA GELİŞTİRME

2.1. Web Uygulama Terimleri

Bir Web uygulamasının çalışmasının daha iyi anlaşılabilmesi için önce web teknolojisi terimlerinin bilinmesi gerekmektedir. Bu terimlerden bazıları şunlardır:

WWW: Web ya da W3 (World Wide Web), yazı, grafik, ses, film gibi pek çok farklı yapıdaki verilere kompakt ve etkileşimli bir şekilde ulaşmayı sağlayan çoklu bir ortam sistemidir. Hiper ortam, bir dokumandan başka bir dökümanın çağrılmasına olanak sağlar (iç içe dökümanlar). Bütün bu farklı yapıdaki veriler uygun bir standart ile bir arada kullanılıp bir WWW tarayıcısında (WWW Browser) görüntülenir.

URL: Uniform Resource Locator kelimelerinin kısaltmasıdır. Web adreslerinin resmi ismidir [9]. Ziyaretçi tarayıcısı üzerinden web servislerine yönlendirme yapılabilmesini sağlayan bir komut formatıdır. Her Web sayfası sadece kendine ait bir URL'ye sahiptir.

URL satırlarının genel formatı şu şekildedir:

<Protokol>://<alan adı>[:port_numarası]/<dizin>/dosya_adi
şeklindedir.

Örnek:

<http://www.sakarya.edu.tr:80/akademik/fbe/duyurular.htm>

http:Protokol

www.sakarya.edu.tr: Alan adı

80: Port numarası: Port numarası 80 kullanan URL'lerin port numarası kullanılmadan URL adresi kullanılabilir.

/akademik/fbe/: dizin

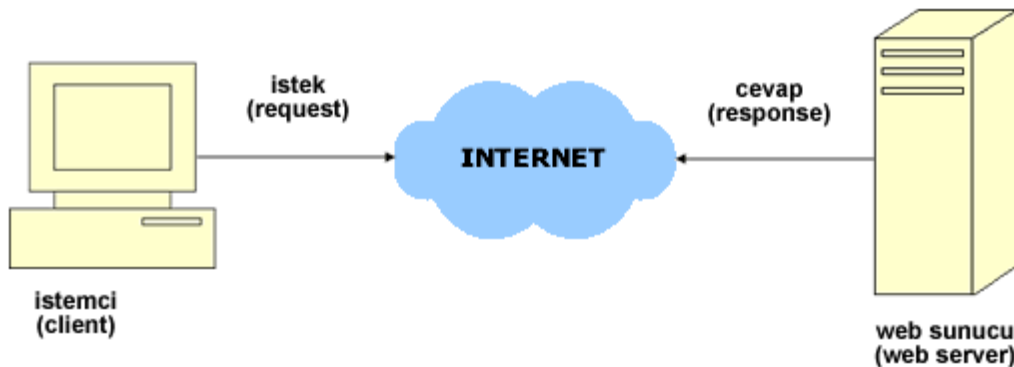
duyurular.htm: Dosya adı

HTTP: Hypertext Transfer Protocol kelimesinin kısaltmasıdır. Web sunucu ile istemci tarayıcısı arasındaki iletişimi sağlayan bir protokoldür. İstemci tarayıcısı üzerinden web sunucusundan istekte bulunur. Web sunucusu da bu isteğe cevap olarak dosyaları istemci tarayıcısına gönderir. Bu işlem HTTP protokolü ile gerçekleştirilir[10]. HTTP protokolü sunucu ve istemci arasındaki veri alışverişini, Internet protokolü olan TCP protokolünü kullanarak yapar[11].

HTML: Hypertext Markup Language kelimesinin kısaltmasıdır. Web sunucusundan istemci tarayıcısına gönderilen verileri web sayfası formatına dönüştüren içerik formatlama dilidir. Bu dil, daha çok, yazılı bir dökümanı formatlamak ve bir objeden başka bir objeye linkler sağlamak ile ilgili komutlar içerir. HTML, HTTP ve ilgili diğer protokolleri kullanabilmek için kullanıcı arayüzleri hazırlamayı sağlar.

2.2. Web Uygulamaları Nasıl Çalışır?

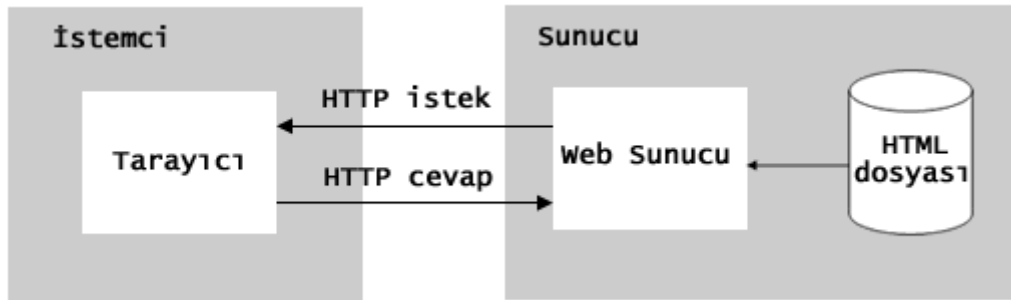
Bir Web uygulaması istemci ve sunucu bilgisayarından oluşmaktadır. İstemcinin tarayıcısı ile web sunucusundan HTTP protokolü ile istemde bulunur. Sunucu bilgisayarda istemcinin isteğine göre dosyaları istemcinin tarayıcısına gönderir. Şekil 2.1’de bir web uygulamasının çalışma prensibi gösterilmektedir.



Şekil 2.1 Web Uygulama Çalışma Şekli

2.3. Durađan Web Sayfalarının alıřması

Statik web sayfaları, Web sunucusundan istemci tarayıcılarına gnderilen belgelerin ieriđi deđiřmez. Őekil 2.2’de statik web sayfalarının alıřma prensibi gsterilmektedir.



Őekil 2.2. Durađan web sayfa alıřması

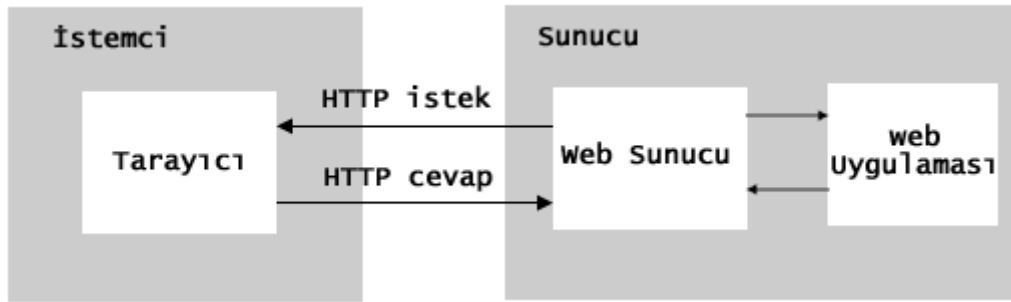
İstemci tarayıcısı ile bir web sayfası isteđinde bulunur. Bu iřlem, istemci tarayıcıda web sayfasının adresini girmesiyle ya da bařka bir sayfadan linke tıklamasıyla bařlar. Aynı anda tarayıcıdan web site sunucusuna HTTP isteđi gnderilir.

Web sunucusu istemci tarayıcısından gelen HTTP isteđini aldıđında, diskindeki ilgili HTML dosyasını HTTP cevabı olarak istemci tarayıcısına gnderir.

Web sunucusundan istemci tarayıcısına gelen HTTP cevabı, tarayıcı tarafından HTML belgesi olarak grntlenir.

2.4. Dinamik Web Sayfalarının alıřması

Durađan web sayfalarının aksine, dinamik web sayfalarının ieriđi web sunucusuna gnderilen parametreye gre deđiřir. Őekil 2.3’de dinamik web sayfalarının alıřma prensibi gsterilmektedir.



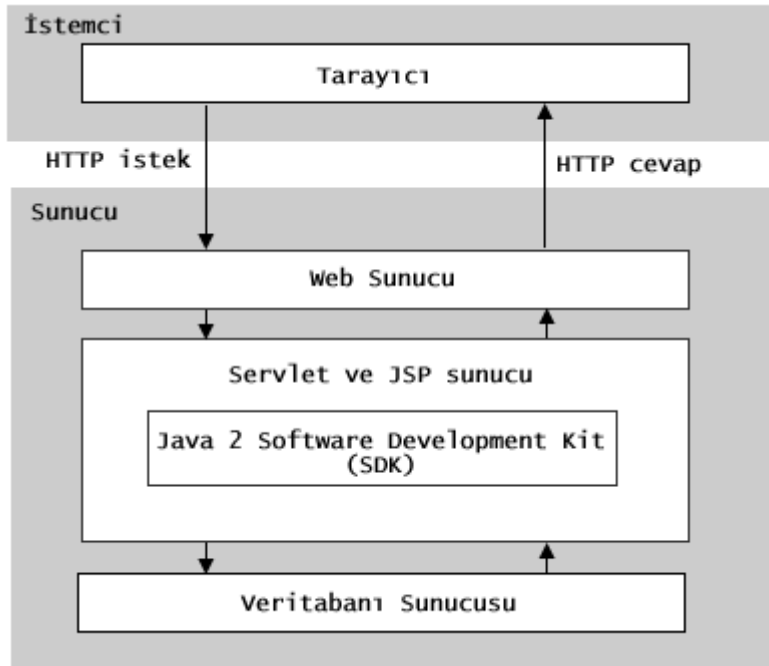
Şekil 2.3. Dinamik web sayfası çalışma prensibi

İstemci tarayıcısından girilen bir web adresi veya herhangi bir sayfadan tıklanan bir link ile tarayıcı tarafından web sunucusuna http isteği gönderilir. Web sunucusuna gelen http isteği, dinamik web sayfası isteği şeklinde ise web sunucusu arkasında çalışan web uygulayıcısına gönderilir.

Web uygulamayıcısına gelen istek işlenir. HTML belgesine dönüştürülür. Bu HTML belgesi web sunucusuna gönderilir. Web sunucusuna gönderilen HTML belgesi http cevabı olarak istemci tarayıcısına gönderilir ve tarayıcı ekranında görüntülenir.

2.5. Java Web Uygulamalarının Çalışması

Java Web uygulamaları Java Server Page (JSP) ve servletlerden oluşur. Java uygulamalarının çalışabilmesi için web sunucu ile birlikte servlet ve JSP sunucusunun çalışması gerekir. Tomcat sunucusu, piyasada en çok kullanılan JSP sunucusudur. Bu nedenle, tez projesinde tasarlanan java uygulamaları Tomcat sunucusunda çalıştırılmıştır. Tomcat sunucusunun çalışması için Java yazılımı yüklü olması gerekir. Sun Microsystems firmasına ait siteden ücretsiz indirilebilen SDK(Standart Development Kit) veya J2SE (Java 2 Platform, Standard Edition) versiyonlarının sunucu tarafında yüklü olması gerekir. SDK, Java dosyalarını derleme (compile) ve standart Java sınıflarını(class) kullanabilme işlemlerini gerçekleştirir. Şekil 2.4'de Java web uygulamasının çalışma prensibi gösterilmektedir [12].



Şekil 2.4. Java web uygulama çalışma yapısı

2.6. JSP Teknolojisi

Java Server Pages (JSP) teknolojisi, hızlı ve kolay dinamik web sayfaları oluşturmak için Sun firması tarafından geliştirilmiştir. JSP teknolojisinin platform bağımsızlığı yani herhangi bir işletim sisteminde sorunsuzca çalışabilmesi nedeniyle web uygulamalarındaki kullanımı hızla artmaktadır. JSP teknolojisi şu kolaylıkları sağlamaktadır.

- Web sitesi tasarımcılarının java dilini öğrenme zorunluluğu olmadan JSP teknolojisini kullanabilmesi.
- JSP teknolojisinde imler(tag) kullanılarak, karmaşık Java işlemlerini yapan basit imler kullanılarak programın daha basit ve kolay yapılabilmesi.
- Java Server Pages Standard Tag Library(JSTL) kütüphanesinin JSP diline entegre edilmesi ile bu teknolojinin fonksiyonlarının kullanılabilmesi. (13)

JSP teknolojisinde geliştirilen uygulamalarda, program satırlarının Java programlama diliyle yazılabilmesi, karmaşık ve büyük uygulamaların geliştirilmesinde JSP teknolojisinin daha çok tercih edilmesini sağlamaktadır.

JSP nin bu özellikleri bir liste halinde aşağıdaki gibi şekilde sıralanabilir:

- Web sunucu desteği: Apache, Netscape ve Microsoft IIS gibi web sunucular JSP ile uyumlu çalışabilir.
- Platform desteği: Java SDK kurulu herhangi bir işletim sisteminde çalışabilir.
- Bileşen modeli: Farklı platformlarda JavaBeans, Enterprise JavaBeans gibi tekrar kullanılabilir modeller içerir.
- Veritabanı erişimi: JDBC (Java Database Connectivity) kullanılır.
- Güvenlik: Java güvenlik modeli içinde çalışır.
- Yeni imler oluşturabilme [14]

2.6.1. JSP mimarisi

JSP dosyaları uzantıları .jsp olarak kaydedilir. Şekil 2.5 deki tarayıcının adres çubuğundaki jsp uzantısından sonraki ? karakteri ise jsp dosyasına gönderilen parametreleri belirtir.



Şekil 2.5. JSP örnek üye gösterme sayfası

JSP kodunun büyük kısmı HTML kodlarından oluşur. Program satırları ise çok az Java kodu içerir. Kompleks java kodları kullanılması gerektiğinde servletler içinde kod yazılarak JSP sayfasına yönlendirilme yapılır. Şekil 2.5'in JSP kodlarını gösteren Şekil 2.6 da <body> tag'i altında "<%>" ile başlayıp "%>" ile biten java kodları bulunmaktadır. Bu kodlar formdan gelen bilgileri değişkenlere aktarmaktadır. Sayfanın alt kısmındaki "<%= %>" tag leri arasında ise değişken değerleri gösterilmektedir.

```

<html>
<head>
<title>Uye Kayıt Ekleme</title>
</head>
<body>
<%
String adi = request.getParameter("adi");
String soyadi= request.getParameter("soyadi");
String email = request.getParameter("email");
%>

<h1>Uye listemize katıldığınız için teşekkürler.</h1>
<p>Girmiş olduğunuz bilgiler </p>
<table width="200" border="1" cellspacing="0" cellpadding="5">
<tr>
<td>Adiniz</td>
<td><%=adi%></td>
</tr>
<tr>
<td>Soyadiniz</td>
<td><%=soyadi%></td>
</tr>
<tr>
<td>Emailiniz</td>
<td><%=email%></td>

```

```

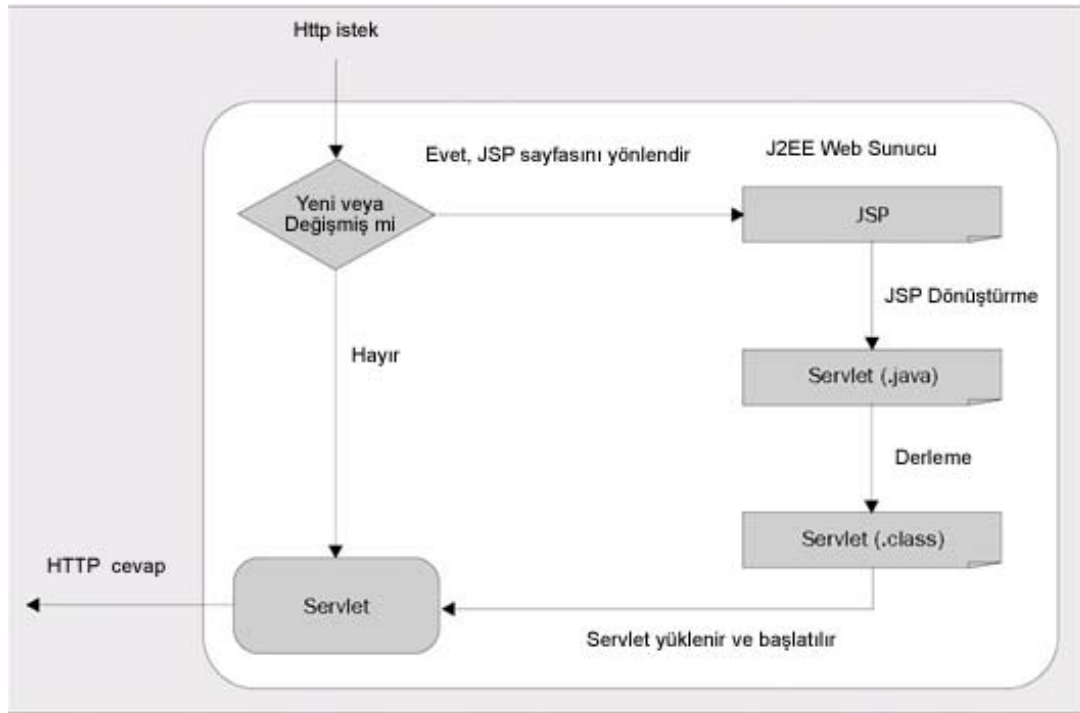
</tr>
</table>
</body>
</html>

```

Şekil 2.6. uyeGoster.jsp sayfası kod bloğu

2.6.2. JSP dilinin hayat döngüsü

Bir JSP sayfası çağrıldığında, ilk olarak istek web sunucusuna iletilir. Web sunucusu sayfanın JSP uzantılı olması durumunda, JSP sayfasını JSP sunucusuna (Tomcat) gönderir. JSP sunucusu da JSP sayfasını Java servlet kaynak koduna dönüştürür. Daha sonra java kaynak kodunu, java class dosyasına dönüştürür. Bu işlemler ilk jsp sayfası istemi yapıldığında yavaş olur. Daha sonraki jsp sayfasına yapılan isteklerde JSP sunucusu jsp sayfasının servlet kodu olup olmadığını kontrol eder, servlet kodu varsa herhangi bir dönüştürme işlemi yapmadan ilgili servleti çalıştırır ve böylece sonraki isteklerin çalışması daha hızlı olur. (Şekil 2.7)



Şekil 2.7. JSP Hayat Döngüsü [15]

2.7. Servlet Teknolojisi

Java servlet, servlet sunucuya (servlet container) dinamik olarak yüklenen ve çalıştırılan java sınıflarıdır. Servlet'ler, benzer teknoloji olan CGI diline göre üstünlükleri ile hızla kullanımı artmaktadır. Servlet'lerin avantajları şunlardır:

2.7.1. Etkin

Servlet'lerin derleme işlemini Java Virtual Machine(JVM) kullanılarak yapıldığından daha ağır çalışan işletim sistemi kullanılmaz. Servlet'ler hafızaya bir kere yüklenir. Servlet'lere yapılan her bir istek için yeni bir servlet hafızaya yüklenmez. RAM'in daha az kullanılması ile hız artar.

2.7.2. Uygunluk

Servlet'ler, HTML form bilgilerini değerlendirmek, http başlıkları okumak ve yerleştirmek, çerezler(cookie) oluşturmak, oturum(session) takip etmek gibi üst seviye uygulamalar geliştirmek için yeterli ve gerekli alt yapıya sahiptir. Java dilini bilen bir kişinin başka bir dil öğrenmesine gerek yoktur.

2.7.3. Güçlü

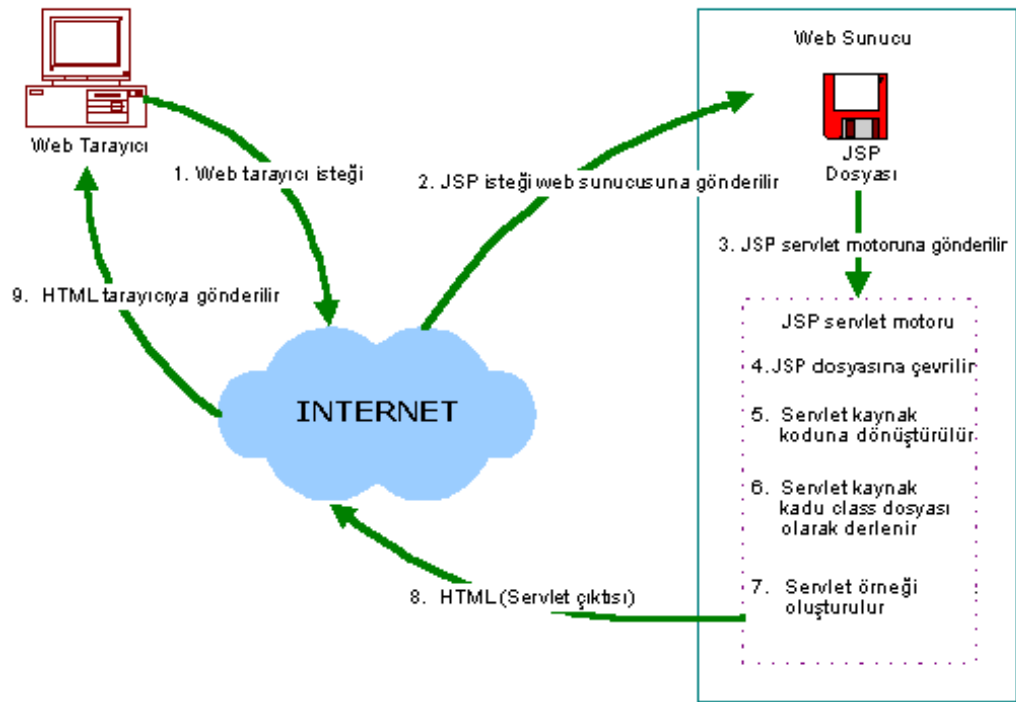
Servlet'ler geleneksel CGI dili ile yapması çok güç işlemleri yerine getirebilecek özelliklere sahiptir.

2.7.4. Taşınabilir

Servlet'ler Java programlama dilinde yazılırlar ve belirli bir API'ı (Applications Programming Interface) takip ederler. Bu özelliği ile Apache, Microsoft Internet Information Services(IIS), IBM WebSphere gibi web sunucularında hiç değiştirilmeden çalıştırılırlar[16].

2.8. Servlet Mimarisi

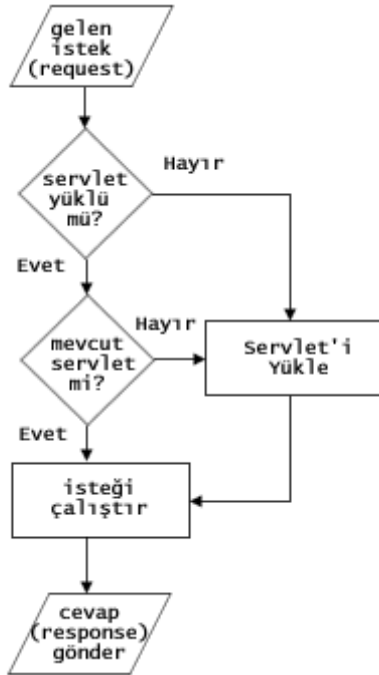
JSP sayfaları, JSP sunucu tarafından servlet'lere dönüştürülürken, servlet ise hazır java kodu derlenerek sınıflara dönüştürülür. Bu sınıflar çalıştırılarak html belgesi olarak Web sunucusuna gönderilir. Ve web sunucu da, kullanıcının tarayıcısına bu html belgesini gönderir. Şekil 2.8'de JSP ve servlet sayfalarının mimarisi gösterilmektedir:



Şekil 2.8. Servlet ve JSP çalışma yapısı [17]

2.9. Servlet'in Çalıştırılması

Servlet'ler ilk oluşturulduğunda, init metodu ile çağrılır. Bu aşamada servlet'in bir örneği(instance) oluşturulur. Ve bunu takiben, service metodu çağrılır. Service metodu ile gelen isteğe göre doGet ve doPost metodlarına yönlendirme yapılır. Servlet kodları gelen isteğe göre doGet veya doPost metodlarında üzerlerine yazılırlar (overriding). Servlet uzun bir süre çağrılmadığı ya da servlet sunucu kapatıldığında, servlet sunucu servlet örneğini kaldırır. Servlet kaldırılmadan önce destroy metodu çağrılır. Şekil 2.9



Şekil 2.9. Servlet Çalışması

2.10. Servlet Metodları

- `public void init() throws ServletException {}`
- `public void service(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {}`
- `public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {}`
- `public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {}`
- `public void destroy() {}`

2.11. JSP ve Servlet'in Birarada Kullanılması

Servlet 'ler aslında java sınıfları olduğu için uygulama içindeki web sayfalarının kod satırları servlet'lerde yazılmalıdır. Çünkü servlet'ler karar verme işlemleri, veritabanı sorgulama, ve kaynaklara erişim gibi karmaşık program kontrolü isteyen işlemlerin yapılmasında JSP ye göre daha hızlı performans verir. Oysaki bu karmaşık program kontrolü JSP de yapılmış olsaydı sadece bu işlemler için birçok JSP sayfası kullanılması gerekir. Bu durumda JSP sayfalarının yönetimi oldukça zor olmaktadır.

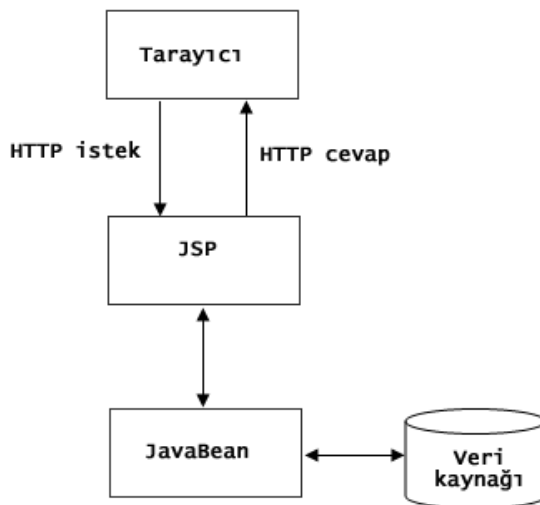
Ayrıca web uygulamasının programlama kısımlarının JSP’de yapıyor olması durumunda, bu kodların diğer JSP sayfalarında tekrar kullanımı mümkün olmayacaktır. Çünkü JSP sayfalarında kodlar scriptlet içinde gömülüdür. Bu şekilde aynı fonksiyonlara erişim mümkün değildir. JSP sayfaları, HTML kodlarından oluşan bir dil olduğu için web uygulama içindeki web sayfalarının tasarımı JSP içinde yapılmalıdır. Bu işlemin gerçekleştirilmesi için servlet’lerde işlenen kod bloğu istek (request) ve cevap (response) nesnesi olarak JSP sayfalarını yönlendirilir. Böylece servlet tüm program kodlarını işlemiş olur, JSP de tüm HTML belgelerini işlemiş olur.

2.12. Servlet ve JSP Model Yapıları

İki çeşit model yapısı vardır. Model 1 ve Model 2.

2.12.1 Model 1 mimarisi

JSP, web uygulamasına gelen isteklerin ve gönderilen cevapların yürütülmesinden sorumludur. Şekil 2.10 da görüldüğü gibi JSP JavaBean ile iletişim kurarak, veritabanına erişmeyi gerçekleştirir. JavaBean verilerinin kaydedildiği ortama da veri kaynağı adı verilir. Genel olarak sınırlı ve küçük uygulamalarda model 1 mimarisi iyi çalışır. Ancak JSP içindeki kod sayısı büyüdüğü zaman model 2 mimarisinin kullanılması daha faydalı olacaktır.

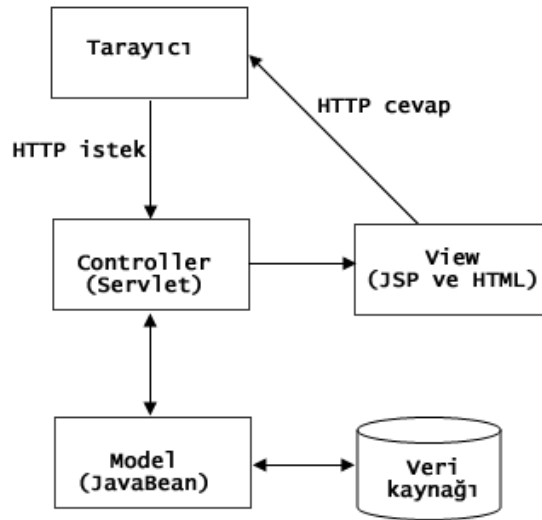


Şekil 2.10. Model 1 mimari

2.11.2. Model 2 mimari (Model View Controller -MVC)

Programcılarının yaygın şekilde kullandığı mimaridir. Model 2 ya da MVC mimarisi, model, görünüm(view), kontrolör(controller) katmanlarından oluşmaktadır. Model 2 mimarisi, model 1 mimarisinden daha iyi performansta çalışır.[12]

MVC mimarisinde, model katmanını uygulamadaki JavaBean temsil eder. Görünüm(view) katmanını sunum katmanı olarak kabul edilir. Web uygulamasındaki HTML ve JSP uzantılı sayfalar tarayıcıdaki görünüm işini gerçekleştirirler. Kontrolör(Controller), web uygulamasının akışını kontrol eder. Bu işlemleri ise servlet'ler gerçekleştirir. (Şekil 2.11)



Şekil 2.11. MVC (Model 2) mimari

BÖLÜM 3. PHP VE MYSQL İLE WEB YAZILIMI GELİŞTİRME

3.1. PHP Nedir ve Nasıl Çalışır?

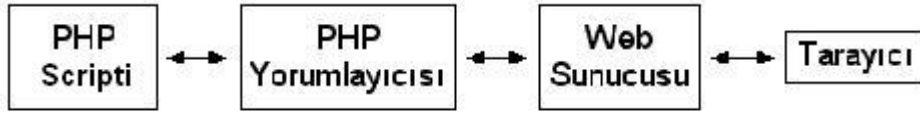
PHP, sunucu taraflı çalışan web script(betik) dilidir. 1994 yılında Rasmus Lerdorf tarafından geliştirilmiştir. PHP kısaltmasını, Personel Home Page kelimelerinin baş harflerinden almıştır. 1994 ten günümüze kadar sırasıyla PHP/FI, PHP 3.0, PHP 4.0 ve son olarak nesne tabanlı yazılımı destekleyen PHP 5.0 versiyonları geliştirilmiştir.

PHP script dili ile yazılan sayfaları, PHP veya (kullanılan sürüme göre) PHP, PHP3 gibi uzantılı dosya olarak kaydedilir. PHP ile yazılan dosyalar derlenmezler (compile edilmezler). Sadece web sunucuda bu dilde yazılmış scriptleri yorumlayabilecek bir PHP yorumlayıcı program mevcuttur. Bu yorumlayıcı yazılmış PHP scriptlerini web sunucunun anlayabileceği bir biçime dönüştürür ve gönderir

PHP ile yazılmış bir web sayfasına bağlanıldığında temel olarak aşağıdaki işlemler yapılmaktadır.

- a) İstemci(client) tarafından PHP dosyası bir tarayıcı (browser) ile çağrılır.
- b) Web sunucu dosya uzantısından bu isteğin bir PHP dosyası olduğunu algılar ve PHP yorumlayıcısına yollar.
- c) PHP yorumlayıcısı ilgili dosya içindeki scriptleri çalıştırarak geriye döndürdüğü sonucu web sunucuya tekrar gönderir.
- d) Web Server' a ulaştırılan sonuç istemci tarafına HTML dosya olarak yollanır.

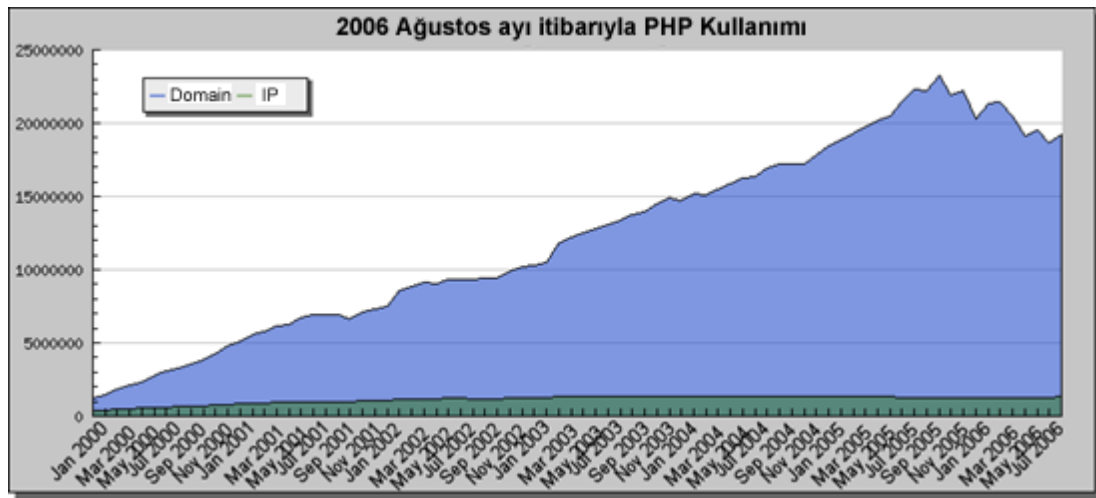
(Şekil 3.1)



Şekil 3.1. PHP script dilinin çalışma şeması [18]

3.2. PHP Dilinin Avantajları

PHP sitelerinin sayısı her geçen gün artmaktadır. Günümüzde yaklaşık 94 milyon web sitesininin 20 milyonu PHP sitelerinden oluşmaktadır.[19] (Şekil 3.2)



Şekil 3.2. 2006 Ağustos tarihine kadar olan PHP kullanım istatistiği [19]

PHP kullanımının yaygın olmasının nedenleri aşağıdaki başlıklar altında toplanabilir

3.2.1. Ücretsiz lisans

PHP ücretsizdir. Herkes <http://www.php.net> sitesinden PHP dosyalarını ücretsiz indirebilir.

3.2.2. Kolay ve hızlı kullanım

PHP dili, öğrenilmesi ve yazılımı kolay bir programlama dilidir. Web sunucuları içinde en çok tercih edilen Apache web sunucu ile tümleşik çalışabilmektedir. (Tablo 3.1)

Tablo 3.1. 2006 Ağustos ayında web sunucu kullanım sayısı ve oranı [20]

Web Sunucusu	Ağustos 2006 sayısı	Oran
Apache	57906817	62.52
Microsoft	27905439	30.13
Zeus	521619	0.56
Sun	344862	0.37

C syntax'ı (yazılım biçimi) ile okunması ve takip edilmesi kolaydır. Aynı zamanda C özelliklerini taşımasına rağmen C kadar katı kuralları olan bir dil değildir. Ayrıca sürekli büyüyen kütüphanesinin her bilgi kaynağına erişmesi, yazılım geliştirme sürecini kısaltmaktadır. Aynı zamanda .NET, JAVA veya COM kütüphaneleri ile uyumlu çalışıp, kaynaklarından fayda sağlanabilmektedir.

3.2.3. Güncel veritabanları ile uyumlu ve etkili çalışma

PHP yaygın kullanılan her türlü veritabanını destekler. Özellikle MySQL ve Oracle veritabanları ile daha uyumlu ve etkili çalışır. Çünkü, bu veritabanları ile bağlantı kurmak birkaç PHP komutuyla kolayca gerçekleştirilir.

3.2.4. Performans

PHP dili, hızlı bir web programlama dilidir. Günümüzde, PHP'nin web uygulamalarında rakibi sanal makineler olan .NET ve JAVA'dır. Sanal makineler kendileri için sistem kaynaklarını yoğun kullandığı için uygulamanın çalışması yavaşlar. Sanal makinelerin iddiası platform bağımsız olmalarıdır. Fakat PHP farklı bir yaklaşım ile kaynakları en az kullanarak platform bağımsızlık konusunu çözümlenmiştir. Sanal makineler, ara derlenmeden geçmiş kodları, bulunduğu ortama göre son derlenmeden geçirirler. Bu şekilde platform bağımsız olurlar. PHP ise her platform için PHP grubu tarafından uyumlu olacak şekilde geliştirilmiştir, bu sayede arada bir sanal makinaya ihtiyaç duymaz. Ayrıca PHP'nin modüler yapısı sayesinde kullanılmayacak modüller sistemden çıkarılıp ortam optimize edilebilir. Bu özellik sanal makinelerde mevcut değildir [21]. PHP'yi sürekli olarak dünyanın dört bir yanından Internet üzerinden katılarak geliştiren PHP geliştiricileri Zend adı altında Java'ya alternatif bir oluşum geliştirmektedirler.

PHP 4.0 versiyonu ile Microsoft firmasının geliştirdiği ASP teknolojisi arasında yapılan test sonuçlarına göre PHP dilinin daha hızlı olduğu görülmektedir. Bu test de, QuickSort algoritması çalıştıran özdeş ASP ve PHP scriptleri, 128 MB RAM'li P2-233 sistemi üzerinde Windows NT/Service Pack4 ve web server olarak IIS 4.0 kullanılarak çalıştırılmıştır. Burada PHP yorumlayıcısı ISAPI modülü olarak çalışmakta ve işletim sisteminin NT ve web serverin IIS olması ASP için bir avantaj olması gerekirken PHP nin daha hızlı çalıştığı görülmektedir. (Tablo 3.2)

Tablo 3.2. ASP ve PHP nin karşılaştırılması[22]

	ASP	PHP/Zend		PHP/Zend+optimizer	
	Süre	Süre	Performans Artışı	Süre	Performans Artışı
Basit Döngü (10 Milyon iterasyon)	44sn.	43sn.	2%	12.5sn	352%
QuickSort algoritması (200 öge)	10sn.	44sn	-440%	6sn.	66%

3.2.5. Güvenli ve güncel

PHP grubu en az iki ayda bir sürüm günceller. Bu güncellemeler varsa bug (açık)ların telafisini ayrıca yeni özellikleri ve optimizasyonları içermektedir. Bu nedenle güvenli ve güncel programlama dilidir.

3.2.6. Yaygın işletim sistemlerine uyumluluk

PHP; Unix, Linux, Windows, Macintosh OS X gibi birçok işletim sisteminde Apache web sunucu ile birlikte çalışabilmektedir. Bununla birlikte, Microsoft firmasının geliştirdiği Internet Information Server(IIS) adındaki web sunucusuyla da uyumlu çalışabilmektedir. Geliştirilen web uygulaması, harici kaynaklardan bağımsız ise bu özelliği ile PHP uygulaması tercih sebebi olmaktadır. .NET framework her ne kadar Microsoft tarafından platform bağımsız olarak tanıtılsa da bu Microsoft'un Windows'dan başka platformu tanımamasından kaynaklanmaktadır; ama bugün web

sunucuları tarafında bağımsız organizasyonların yaptığı araştırma çoğunlukla Windows olmayan işletim sistemlerinin kullanıldığını göstermektedir.

3.2.7. Başarılı açık kod projesi

Bir çok açık kod projesi organizasyonel hatalar sonucu başarısız olmaktadır. Fakat PHP hiç bir zaman bu tarz sorunlarla karşılaşmamıştır. Çünkü, arkasında Zend adında PHP'yi sürekli geliştiren bir firma bulunmaktadır.

3.3. PHP' nin Temelleri

PHP dosyaları temelde web sunucuda yer alan metin dosyalarından farklı birşey değildir. Web sunucu ayarına bağlı olarak .PHP3, .phtml ya da .PHP uzantısıyla tanımlanırlar. Bu metin dosyasında PHP komutları ve HTML yer alır. Web sunucusu PHP olarak tanımlanmış örneğin .PHP3 uzantılı bir dosya çağrıldığı zaman bu dosyayı önce PHP yorumlayıcısı gönderir. PHP yorumlayıcısı aslında metin dosyası olan .PHP3 dosyasındaki komutları çalıştırarak HTML'e dönüştürür. Ortaya çıkan HTML kodu da web sunucu aracılığıyla isteği gönderen kullanıcıya iletilerek işlem tamamlanmış olur.

PHP yorumlayıcısı kendisine ulaşan metin dosyasında "<?" ya da "<?PHP" kelimelerine rastlayana kadar olan tüm HTML kodu web sunucuya aynen gönderir. PHP komutlarının başladığını belirten "<?" ya da "<?PHP"den sonra ">"ya kadar olan bölümü işleterek çıkan sonuçları, web sunucuya iletir. Bir PHP dosyası sadece HTML'den, hem HTML hem PHP komutlarından ya da sadece PHP komutlarından oluşabilir.

PHP temel olarak C dilinin notasyonu üzerine inşa edilmiştir. Tüm PHP komutları ";" (noktalı virgül) işaretiyle sonlandırılmalıdır. PHP dosyalarına açıklayıcı bilgiler eklemek için tek satırlık açıklamalarda //, birden çok satırı kapsayacak açıklamalarda /* ve */ işaretleri kullanılır. Ekranı basılacak olan metin ya da HTML kodlarında C'de olduğu gibi \n belirteci yeni bir satıra geçmek için, \t belirteci tab boşluğu vermek için kullanılır. PHP'de esnek kodlama yapılabilir. Metin değişkenler ister tek

ister çift tırnakla kullanılabilir. PHP'de değişken ve komutlar küçük büyük harf duyarlıdır. Değişken tanımlarken metin ya da sayı olmasının çok fazla önemi yoktur. Değişken adlarının başına \$ karakteri eklenir. Metin olarak tanımlanan ama sayı içeren bir değişkende daha sonra matematiksel işlem yapabilmek mümkündür. Tüm diğer dillerde olduğu gibi PHP'de de sayı, uzun sayı, metin, array ve obje tipi değişken tanımları yapılabilir. Bu değişkenlerin tiplerinin önceden tanımlanarak PHP'ye bildirilmesi gerekmemektedir.

PHP'de yaygın olarak ekrana bilgi basmak için echo komutu kullanılır. Aynı işlem için print ya da printf de kullanılabilir. Ekrana basılacak bilgi metni içinde tekli ya da çiftli tırnak olması durumlarında öncesinde \ işareti kullanılmalıdır. Aksi takdirde echo komutunun metin ayırıcı olan çift ya da tekli tırnaklı basılacak bilgideki tırnaklar karışabilir.[23]

```
<?
echo "Yüksek Lisans Teziiiii<br>\n"; // Tek satır açıklamalarında çift slash
yeterlidir.
/* Çoklu satır açıklamalarında
tek slash ve yıldız işareti kullanılır*/
echo "\t<br>Satır sonlarında noktalı virgül olmalıdır. Özel işaretlerde \
kullanılmalıdır.";
echo ("1");
echo "1";
echo '1';
// Yukardaki üç satır da aynı işlevi görmektedir.
?>
```

3.4. MySql ve PHP

PHP yaygın kullanılan her türlü veritabanı kullanıcılarına veritabanına, özel PHP komutlarıyla kullanım imkanı sağlar. PHP kullanım klavuzu incelendiğinde veritabanları arasında en çok PHP komutunun MySQL, daha sonra Oracle için

olduğu görülecektir. MySQL komutlarından birçoğu özel durumlarda gerekli olurken 10 kadar komut her türlü MySQL işleminde daha yaygın olarak kullanılırlar.

Bir PHP dosyasında MySQL veritabanı kullanılmak istendiğinde mutlaka veritabanı sunucusuna bağlanılmalı ve bir veritabanı seçilmelidir. Genellikle bir PHP dosyasında bir veritabanına bağlantı yapılacağı için veritabanı sunucusuna bağlandıktan sonra bir veritabanı ismi verilerek tüm diğer MySQL işlemleri o veritabanı sunucusu ve o veritabanı üzerinde yapılabilir. Eğer istenirse birden çok veritabanı sunucusundaki birden çok veritabanına da bağlantı yapılabilir.

Aşağıdaki komutlarda `mysql_connect()` komutuyla veritabanı sunucusuna ip, kullanıcı adı ve şifre verilerek bağlantı sağlanır. Daha sonraki `mysql_select_db()` komutuyla da dosyadaki tüm MySQL işlemlerinde kullanılacak olan veritabanı tercihi belirtilmiş olur.

```
mysql_connect("localhost","root","sifre") or die("Sunucuya ulaşamadım");
mysql_select_db("potdb") or die("Veritabanına ulaşamadım...");
```

PHP dosyasında tüm veritabanı işlemlerinden önce veritabanı sunucusuyla bağlantı kurulmalıdır. PHP'de MySQL'e sorgu göndermek için `mysql_query()` komutu kullanılır. Parantezler içindeki alana SQL dilinde MySQL'in anlayacağı veritabanı sorgulama komutları girilmelidir. MySQL'de oluşabilecek hataları kullanıcının PHP içinde farkedebilmesi için PHP'nin özel komutları vardır. Bu komutlar sayesinde veritabanı sorgulamasında oluşan sonuçlar hata numarasıyla birlikte farkedilebilir. Hata gösterge komutları kullanılmadan PHP, MySQL'e dair hiçbir hata mesajı vermeden işleme devam eder. MySQL'in sorgulanmasından geriye dönen sonuç tablosundaki veriler de `mysql_result()` komutuyla alınır. Aşağıdaki satırlarda `mysql` veritabanı programından `ogrenciler` isimli tablodaki kayıtlar çağrılmaktadır.

```
$isimler = mysql_query(select * from ogrenciler");
if (mysql_error()) {echo ("MySQL hatası oluştu. Hata no :"); echo mysql_error();}
$ad = mysql_result($isimler,$i,"ad");
$soyad = mysql_result($isimler,$i,"soyad");
```

Veritabanından dönen tablodaki ilk satırın satır numarası 0'dır. Her sonucun bir satır olduğu bu tabloda tablo alanının ismiyle satır numarası verilerek sonuçlar çağırılabilir. Öğrenci numarası, adı ve soyadının bulunduğu "ogrenciler" isimli tabloya yapılacak olan genel sorguda geriye dönen sonuç tablosunun her satırında bir öğrenci numarası, ad ve soyad bulunacaktır. Bu sonuç tablosundan sonuçların birer birer alınarak kullanılması için genellikle while döngüsü kullanılır. Sonuç tablosundaki satır sayısını öğrenmek için mysql_numrows() komutu kullanılır. Aşağıdaki satırlarda bahsedilen konu ile ilgili satırlar yer almaktadır.

```
$sisimler = mysql_query(select * from ogrenciler");
if (mysql_error()) {echo ("MySQL hatası oluştu. Hata no :");,
echo mysql_error();
$sayi = mysql_numrows($sisimler)
$i = 0;
while ( $i < $sayi):
$ad = mysql_result($sisimler,$i,"ad");
$soyad = mysql_result($sisimler,$i,"soyad");
echo ("Ad : $ad Soyad : $soyad");
$i++;
end while;
```

Sonuç tablosunun PHP'de kullanılmasını sağlayan mysql_query() komutuyla tüm tablodaki değerler bir sayı kullanılarak tanımlanır. Sonuçlar ise mysql_result() kullanılarak metin değişkenine atanırlar. Eğer sonuçların bir dizi değişkeninde (array) toplanması isteniyorsa mysql_fetch_array() komutu kullanılır.

3.5. PHP ile SQL kullanımı

PHP ile herhangi bir veritabanını kullanırken bilinmesi gereken en önemli ayrıntı SQL (Structured Query Language) adı verilen veritabanı yönetim ve sorgulama dilidir. Bu dilin detayları hakkında bilgi sahibi olunarak veritabanını performanslı ve hızlı şekilde kullanabilmek mümkün olur. Çoğu zaman orta derecede SQL kullanılan

sitelerde performans kaybı ve gereksiz sorgularla karşılaşılır. Veritabanından sonucu tek sorguda çekmekle iki sorguda çekmek arasında oldukça büyük performans kaybı oluşacaktır. Ayrıca SQL kullanılarak yapılan işlemleri PHP ile yapmak güç kaybına da yol açar.

Öğrenci sınav sonuçlarının bulunduğu bir tablo göz önünde bulundurulduğunda birinci, ikinci ve üçüncü sınav sonuçlarının PHP'de toplanmasıyla performans ve memory kaybı vardır. SQL'de sınav sonuçlarının toplanması her zaman için daha hızlı olacaktır.

3.6. MySQL Yönetimi

Tüm SQL işlemleri için PHP kullanılabilceği gibi veritabanı sunucusunda veritabanı, tablo oluşturmak, değiştirmek ya da diğer işlemler için farklı arayüzler kullanmak zaman kazandırıcı olabilir. PHP kullanılarak geliştirilen PHPMyAdmin sayesinde web sunucu aracılığıyla tüm MySQL özelliklerini yönetebilmek mümkündür. Hem Unix hem de Windows ortamları için PHPMyAdmin tavsiye edilir. Windows'da da MySQL'i yönetmek için myAdmin adındaki yazılım da oldukça kullanışlıdır. MySQL'e ve PHP'ye yeni başlayan kullanıcılar için veritabanını tanımak amacıyla ayrı bir yazılım kullanmaları tavsiye edilebilir[24].

BÖLÜM 4. ONLINE MALZEME STOK TAKİP PROGRAMI

4.1. Programın Tanımı

Bu tezde, sunucu taraflı programlama dillerinde nesne tabanlı programlama ile fonksiyonel programlama dillerinin performanslarının karşılaştırılması amaçlanmıştır. Karşılaştırma işlemi için hem nesne tabanlı hem de fonksiyonel programlama dillerinde online malzeme stok takip projesi yazılmıştır. Nesne tabanlı dil olarak JSP, fonksiyonel dil olarak PHP programlama dili seçilmiştir. Dolayısıyla online malzeme stok takip programı iki ayrı dilde yazılmıştır.

Online malzeme stok takip programında, onaylı kullanıcılar sayfaya online erişerek, malzeme sayfasında bulunan malzemeleri miktarı ile birlikte seçimini yaparak işlem siparişi yapmaktadırlar. Malzeme siparişi, onaylanmamış işlemler olarak veritabanına kaydedilir. Admin kullanıcısı işlemi onayladığında, malzeme siparişi veren kullanıcının e-posta adresine siparişinin onaylandığına dair mesaj yollanmaktadır. Ayrıca yeni kullanıcılar, kullanıcı bilgilerini girerek siteye yeni üye olabilmektedir. Ancak kullanıcının üyeliğinin tamamlanabilmesi için admin kullanıcısının üyelerin onaylaması gerekmektedir. Üye kayıt bilgileri admin kullanıcısına onaylanmamış kullanıcılar olarak iletilmektedir. Admin kullanıcısı aynı zamanda stok takip programına yeni ürünler girebilmekte, üyelerin siparişlerini kontrol edebilmekte ve admin kullanıcısının şifre değiştirebilme işlemlerini gerçekleştirebilmektedir.

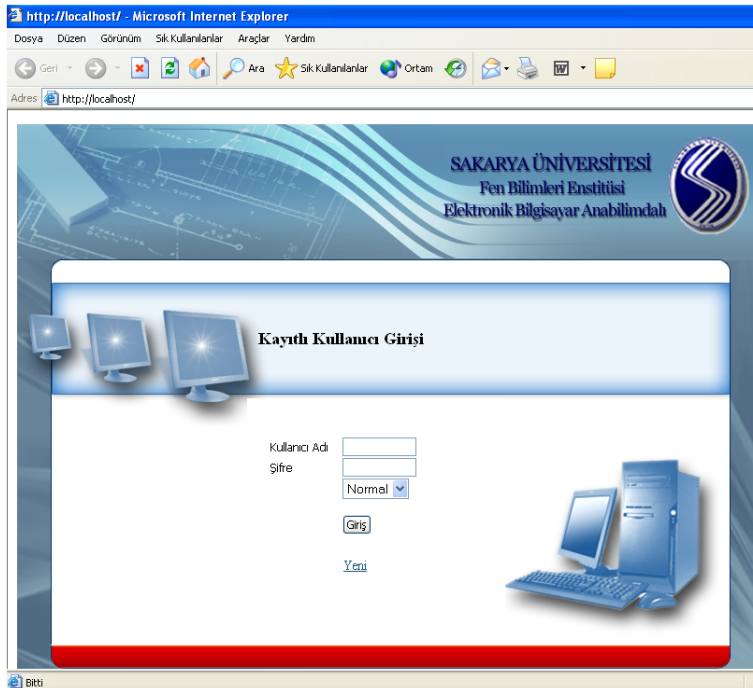
Program iki temel modülden oluşmaktadır: Kullanıcı modülü ve admin modülü.

4.2. Kullanıcı Modülü

Malzeme seçim işlemleri burada yapılır. Bu işlemin yapılabilmesi için önce onaylı kullanıcının kullanıcı ad ve şifresini girdiği login işlemini yapması gerekir.

4.2.1. Login olma

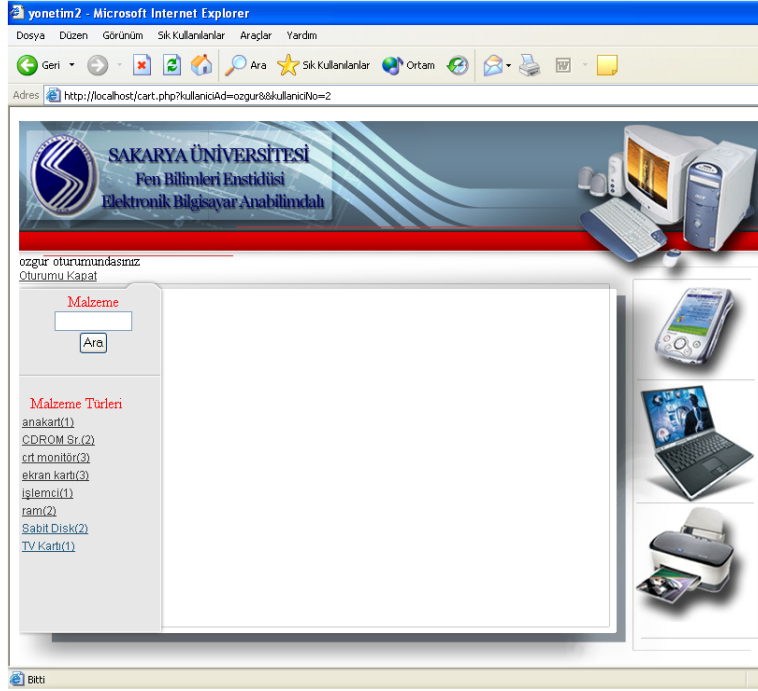
Web tarayıcısının adres kutusuna <http://localhost> yazılarak index veya menü sayfası yüklenir. Malzemeler sayfasına kullanıcının ulaşabilmesi için, kullanıcı adı ve şifresini girmesi gerekmektedir. Ayrıca oturum türünün de normal oturum olması gerekmektedir. (Şekil 4.1)



Şekil 4.1. Online malzeme stok takip programı kullanıcı girişi arayüzü

4.2.2. Malzeme sayfası

Kullanıcı adı ve şifresi doğru girilmesi durumunda malzeme sayfasına erişilir. Sayfanın sol tarafında malzeme tür listesi, sol üst köşesinde ise malzeme arama bölümleri bulunmaktadır. İlgili malzeme türünden kaç tane stokta olduğu parantez içinde belirtilmiştir. Sayfanın üstünde ise login olan ya da oturum açan kullanıcının adı görülmektedir. (Şekil 4.2)



Şekil 4.2. Online malzeme stok takip programı malzemeler sayfası arayüzü

4.2.3. Malzeme seçimi

Malzeme sayfasında ilgili malzeme türlerine tıklandığında, sayfanın ortasında ilgili türdeki malzemeler listelenir. (Şekil 4.3)

KOD	TÜR	MARKA	MODEL	Detay
bm-11	crt monitör	philips	107E	Detay
bm-12	crt monitör	Philips	107T60	Detay
BML-1	crt monitör	LG	710S	Detay

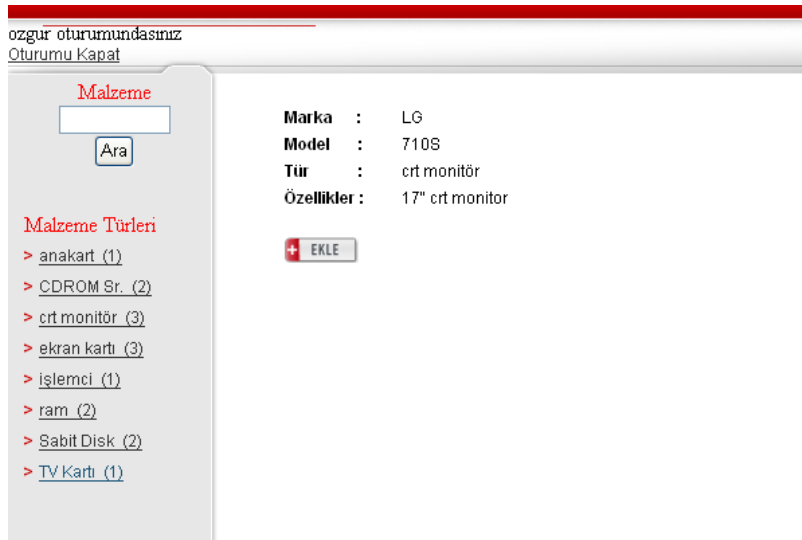
Şekil 4.3. Online malzeme stok takip programı malzeme grupları arayüzü

4.2.4. Malzeme detayı

Malzeme listesinde herbir malzemenin detaylı bilgisine ulaşmak için detay butonuna tıklanır. İlgili malzemenin detaylı bilgisi listelenir.

4.2.5. Malzemeyi seçme

Malzeme seçilmek istenirse, malzeme detay sayfası altındaki ekle butonuna tıklanır. Seçim listesine ilgili malzeme eklenir. İstenirse ilgili malzemenin sayısı değiştirilebilir. Kullanıcı isterse başka malzemeler seçebilir. Ancak seçtiği malzemeler seçim listesinde silinmezler. Malzemeyi seçim listesinden çıkarmak için adet kısmına 0 yazılarak değiştir butonuna tıklanır. Kullanıcı malzeme seçimini bitirdiğinde onayla butonuna tıklayarak işlemi bitirir. (Şekil 4.4)

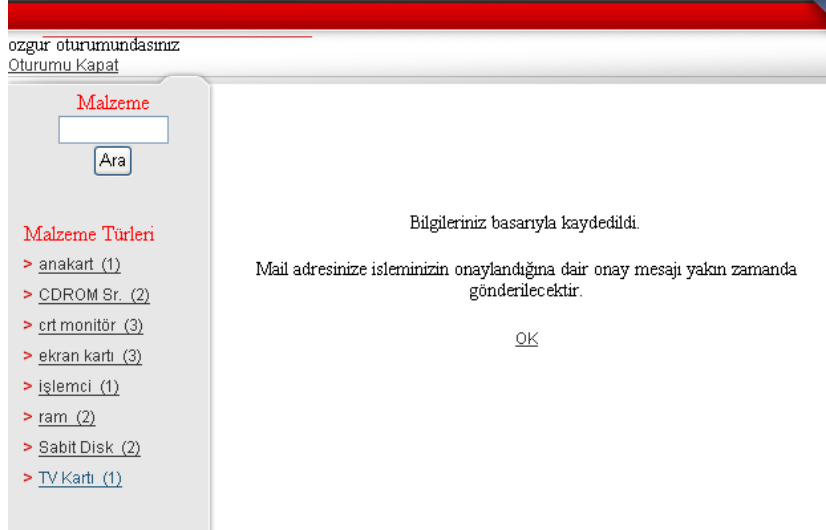


Şekil 4.4. Malzeme detay arayüzü

4.2.6. Seçim işleminin bitirilmesi

Seçim listesinde onayla butonuna tıklanarak, seçim işlemi bitirilir. Ancak bu seçim işleminin geçerli olabilmesi için, yönetici (admin) tarafından yaptığı işlemin onaylanması gerekir. Bu nedenle kullanıcının e-posta adresine yaptığı işlemin onaylandığına dair e-posta ulaşması gerekir. OK butonuna tıklanarak seçim işlemi

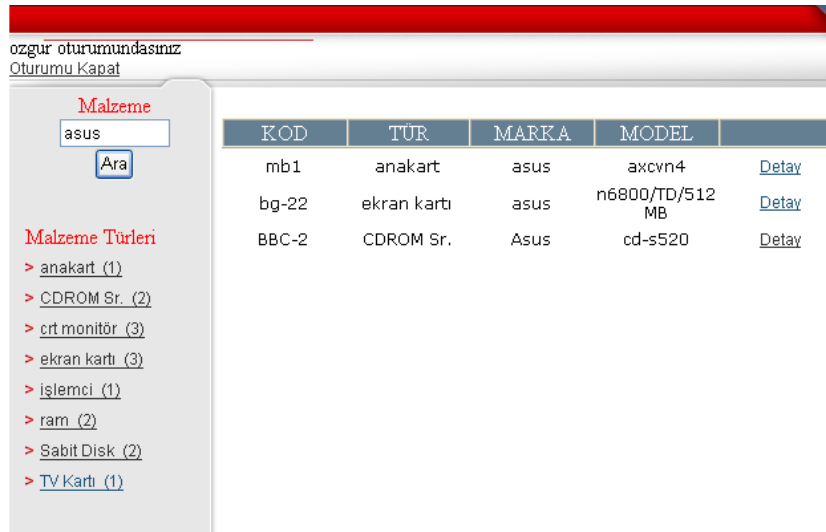
bitirilir ve malzemeler menüsüne yeni bir seçim yapmak için dönülür. Seçim işlemi sonucunda oturum (session) nesnesinde tutulan malzemeler listesi silinir. (Şekil 4.5)



Şekil 4.5. Malzeme seçme işleminin bitirilmesi

4.2.7. Malzeme arama

Malzeme sayfasında, aranılan özellikteki bilgiler arama kutusuna yazılarak arama yapılabilir. Malzemenin türü, modeli, marka ve özellikleri bilgileri girilerek ilgili malzemeler listelenir. Daha sonra malzeme seçim işlemine devam edilebilir. Örneğin programda arama kutusuna Asus yazılırsa, tür, marka ve özellik bilgilerine göre arama yapılarak ilgili bütün kayıtlar listelenecektir. (Şekil 4.6)



Şekil 4.6. Malzeme arama

4.2.8. Yeni kullanıcı ekleme

Malzeme sayfasına üyelik sistemi ile girilebildiği için, üye olmayanların malzeme sayfasına ulaşması mümkün değildir. Ancak index sayfasının altında yeni linkine tıklanarak yeni kullanıcı ekleme sayfasına girilebilir. Bu sayfada ilgili kutucuklara bilgiler girilerek kullanıcı üye olabilmektedir. Aynı kullanıcı adı olan başka bir kullanıcı adı girildiğinde kullanıcı uyarılarak başka bir kullanıcı adı girilmesi istenir. Kutucukların sağ tarafındaki * işareti ise, girilmesi zorunlu olan bilgi demektir. (Şekil 4.7)

Şekil 4.7. Yeni kullanıcı ekleme

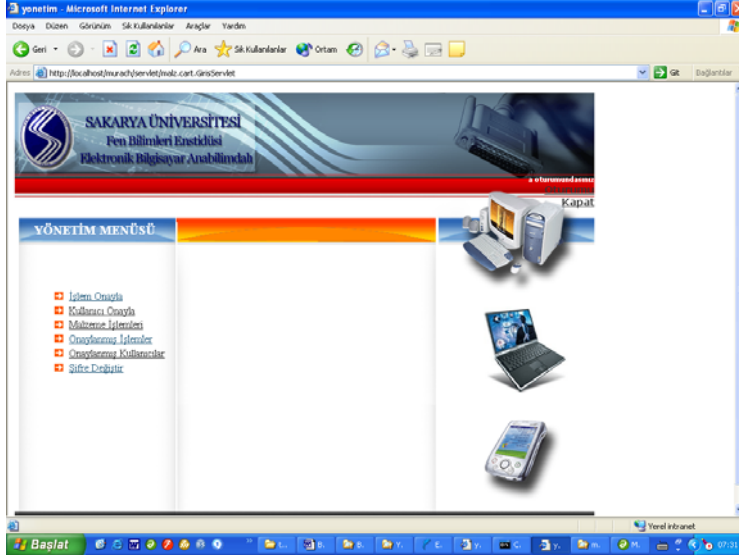
4.3. Yönetici (Admin) Modülü

Bu modülde, kullanıcının yaptığı işlem onaylanarak seçim işlemi tamamlanmış olur. Aynı zamanda yeni üye olan kullanıcılar da onaylanarak malzeme sayfasına giriş yapabilmesine izin verilir. Stoğa yeni malzeme ekleme, stok miktarını değiştirme gibi stok işlemleri de bu modülde yapılır.

4.3.1. Admin arayüzü

Index sayfasında oturum türü admin seçilerek admin kullanıcı adı ve şifresi girilerek admin arayüzüne girilir. Admin arayüzünde işlem onayla, kullanıcı onayla, malzeme

işlemleri, onaylanmış işlemler, onaylanmış kullanıcılar ve şifre değiştir menüleri bulunmaktadır. (Şekil 4.8)



Şekil 4.8. Admin arayüzü

4.3.2. İşlem onaylama

Kullanıcı tarafından yapılan malzeme sipariş işlemleri listelenir. Bu işlemlerin onayı yönetici tarafından yapılır. (Şekil 4.9)



Şekil 4.9. İşlem onaylama

4.3.3. İşlem detayı

İlgili işlemin “Detay için” linki tıklanarak detaylı bilgilerine ulaşılır. İlgili işlemde hangi malzemeden kaç adet seçildiği görülür. Yönetici tarafından kullanıcının yaptığı işlem onaylanırsa “işlemi onayla” linkine tıklanır. (Şekil 4.10)

YÖNETİM MENÜSÜ

- [İşlem Onayla](#)
- [Kullanıcı Onayla](#)
- [Malzeme İşlemleri](#)
- [Onaylanmış İşlemler](#)
- [Onaylanmış Kullanıcılar](#)
- [Şifre Değiştir](#)

İşlem No : 12
 İşlem Tarihi : 2006-09-07
 İstekte Bulunamın : mithat
 Adı :
 Soyadı : percin
 Email : mpercinc@mynet.com

Tür	Marka	Model	Özellikler	Adet
crt monitör	philips	107E		1

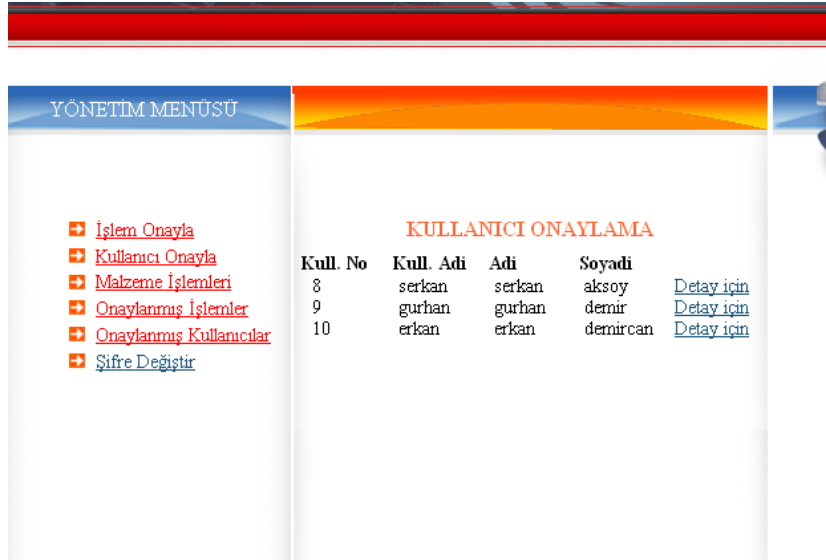
Toplam: 1 adet

[İşlemi Onayla](#) [İşlemlere Geri Dön](#)

Şekil 4.10. İşlem onaylama

4.3.4. Kullanıcı onaylama

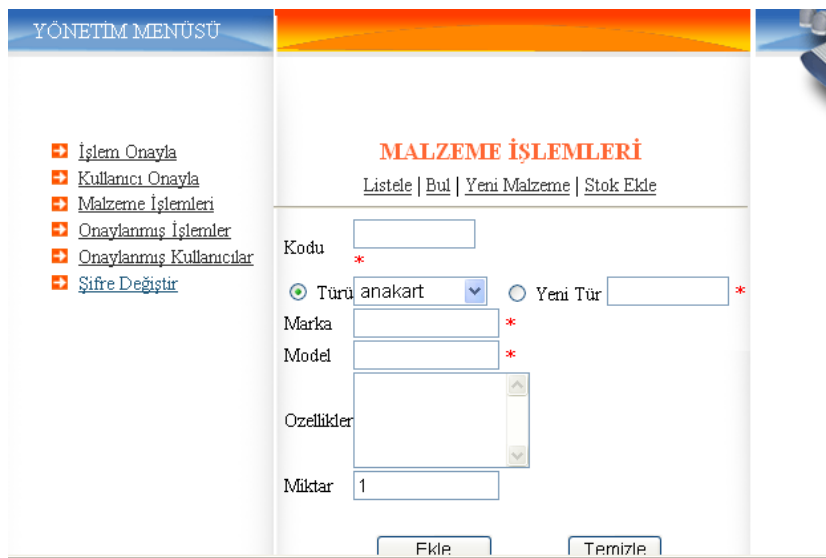
Üye olan kullanıcıların yönetici tarafından onaylanmadan malzeme sayfasına login olması mümkün değildir. Bu nedenle üye kullanıcıların yönetici tarafından onaylanması gerekmektedir. Kullanıcı Onayla linkine tıklanarak yönetici tarafından onayı bekleyen kullanıcılar listelenir. İlgili kullanıcının detay linkine tıklanarak kullanıcının detaylı bilgileri görüntülenir. “Kullanıcı onayla” linkine tıklanarak kullanıcı onaylanmış olur. (Şekil 4.11)



Şekil 4.11. Kullanıcı onaylama

4.3.5. Malzeme işlemleri

Malzeme stok işlemlerinin yapıldığı yerdir. Burada, yeni malzeme ekleme, listeleme, stok girişi, malzeme bulma işlemlerinin yapılabilir. (Şekil 4.12)



Şekil 4.12. Admin malzeme işlemleri menüsü ve yeni malzeme ekleme

4.3.6. Onaylanmış işlemler

Onaylanan işlemlerin listesi yapılır. (Şekil 4.13)

YÖNETİM MENÜSÜ		ONAYLANMIŞ İŞLEMLER		
<ul style="list-style-type: none"> ➤ İşlem Onayla ➤ Kullanıcı Onayla ➤ Malzeme İşlemleri ➤ Onaylanmış İşlemler ➤ Onaylanmış Kullanıcılar ➤ Şifre Değiştir 	İşlem No	Adı	Soyadı	İşlem Tarihi
	14	isa		Detay için...
	15	isa		Detay için...
	16	isa		Detay için...

Şekil 4.13 Onaylanmış işlemler

Detay için butonuna tıklanarak ilgili işlemin detaylı bilgilerine ulaşılır.

4.3.7. Onaylanmış kullanıcılar

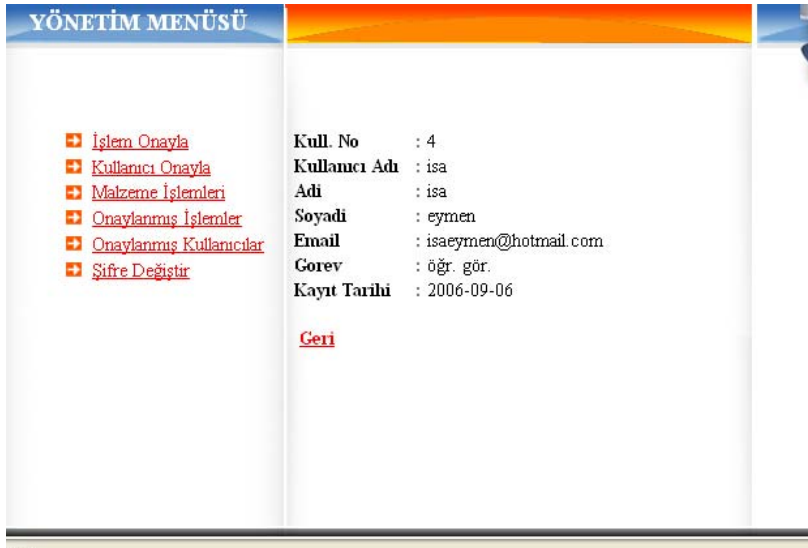
Yönetici tarafından onayı yapılan kullanıcılar listelenir. (Şekil 4.14)

İlgili kullanıcının detaylı bilgilerine “Detay için” linkine tıklanarak ulaşılır. (Şekil 4.15)

YÖNETİM MENÜSÜ		ONAYLANMIŞ KULLANICILAR			
<ul style="list-style-type: none"> ➤ İşlem Onayla ➤ Kullanıcı Onayla ➤ Malzeme İşlemleri ➤ Onaylanmış İşlemler ➤ Onaylanmış Kullanıcılar ➤ Şifre Değiştir 	Kull. No	Kull. Adı	Adı	Soyadı	
	3	ozgur	ozgur	ozgur	Detay için
	4	isa	isa	eymen	Detay için
	5	edp	EDİP	ŞEN	Detay için
	6	cerin	cerin	gul	Detay için
	7	mithat	mithat	percin	Detay için
	1	ozi	ozi	ozi	Detay için

Şekil 4.14. Onaylanmış kullanıcılar

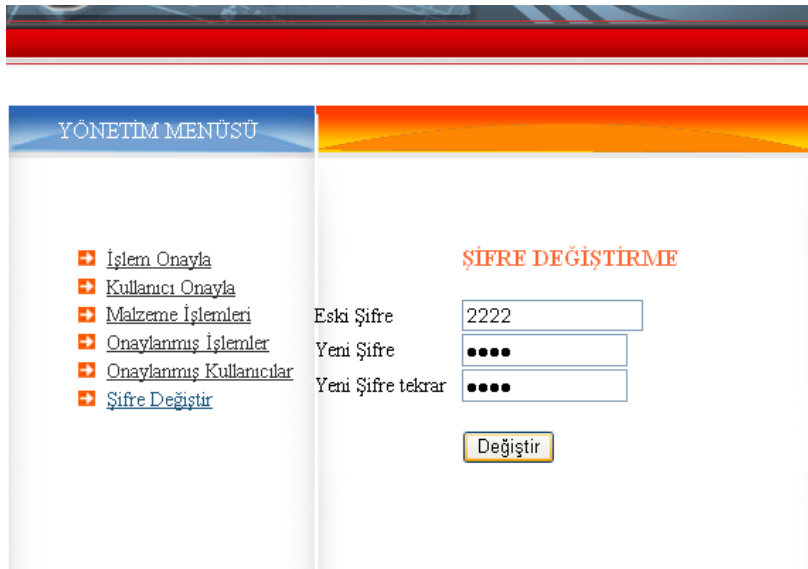
Detay için butonuna tıklanarak ilgili kullanıcının detaylı bilgilerine ulaşılır.



Şekil 4.15. Onaylanmış kullanıcı detayı

4.3.8. Şifre değiştirme

Yönetici (admin) şifresinin değiştirildiği yerdir. Mevcut şifre, yeni şifre bilgileri girilerek şifre değiştirilmesi yapılır. Mevcut şifre alanı doğru girilmeden şifre değiştirme yapılamaz. (Şekil 4.16)



Şekil 4.16. Şifre değiştirme

BÖLÜM 5. PROGRAMIN JSP VE PHP DİLİNDE GELİŞTİRİLMESİ

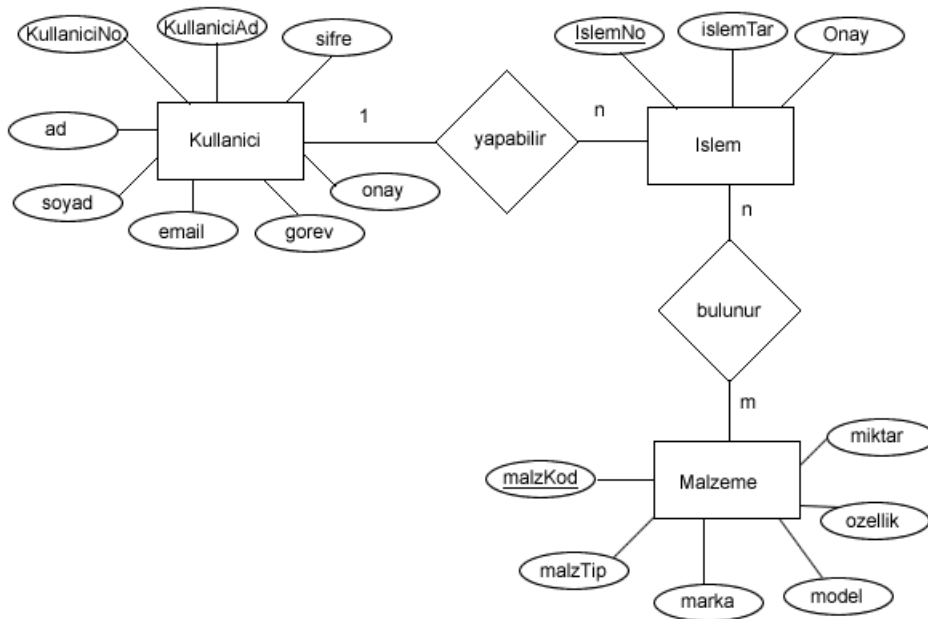
Programda her iki dil için veritabanı olarak MySQL Servers and Clients 3.23 versiyonu kullanılmıştır. Web sunucu olarak PHP dili için Apache 2.0.58 versiyonu, JSP dili için ise Apache Tomcat 4.0.1 versiyonu kullanılmıştır.

5.1. Veritabanı tasarımı ve tablolar

Programda kullanılan tablolar, varlık-ilişki (entitiy-relationship) modeline göre bağlantılar tasarlanmıştır.

5.1.1. Program varlık- ilişki (Entity-Relationship) modeli

Programın analiz aşamasında kullanıcı, malzeme, işlem varlıkları tespit edilmiş ve bu varlıkların ER şeması yapılmıştır. Şekil 5.1 de malzeme ER şeması görülmektedir.



Şekil 5.1. ER şemasına göre yeni tablolar ve alanları

5.1.2. Programda kullanılan tablolar

Kullanici Tablosu	
Alan adları	Alan Türü
kullaniciNo(Key)	İnt
kullaniciAd	varchar(20)
Sifre	varchar(20)
Ad	varchar(30)
Soyad	varchar(30)
Email	varchar(20)
Gorev	varchar(30)
Onay	Enum('y','n')

Şekil 5.2. Kullanici Tablosu

Malzeme Tablosu	
Alan Adları	Alan Türü
malzKod	varchar(20)
malzTip	varchar(30)
Marka	varchar(30)
Model	varchar(30)
Ozellik	varchar(50)
Miktar	İnt

Şekil 5.3. Malzeme Tablosu

Islem Tablosu	
Alan Adları	Alan Türü
islemNo	int
islemTarih	datetime
kulaniciNo	int

Şekil 5.4. Islem Tablosu

IslemDetay Tablosu	
Alan Adları	Alan Türü
islemDetayNo	int

islemNo	int
malzKod	varchar(20)
Adet	

Şekil 5.5. IslemDetay Tablosu

Admin Tablosu	
Alan Adları	Alan Türü
User	Varchar(16)
Password	Varchar(30)

Şekil 5.6. Admin Tablosu

5.1.3. Malzeme database SQL scripti

Programda kayıt işlemleri MYSQL veritabanı kullanılarak yapılmıştır. Programda kullanılan veritabanına malzeme adı verilmiştir. Bu veritabanı altında tablolar oluşturulmuştur. (Şekil 5.7)

```

DROP DATABASE IF EXISTS malzeme;
CREATE DATABASE malzeme;
USE malzeme;
CREATE TABLE Kullanici(
kullaniciNo INT NOT NULL AUTO_INCREMENT,
kullaniciAd VARCHAR(20) NOT NULL DEFAULT "",
sifre VARCHAR(20) NOT NULL DEFAULT "",
ad VARCHAR(30) NOT NULL DEFAULT "",
soyad VARCHAR(30) NOT NULL DEFAULT "",
email VARCHAR(20) NOT NULL DEFAULT "",
gorev VARCHAR(30),
onay enum('y','n') DEFAULT NULL,
PRIMARY KEY (kullaniciNo)
);
CREATE TABLE Malzeme (
malzKod VARCHAR(20) NOT NULL DEFAULT "",
malzTip VARCHAR(30) NOT NULL DEFAULT "",

```

```
marka VARCHAR(30) NOT NULL DEFAULT "",
model VARCHAR(30) NOT NULL DEFAULT "",
ozellik VARCHAR(50),
miktar INT NOT NULL DEFAULT '0',
PRIMARY KEY (malzKod)
);
CREATE TABLE Islem(
islemNo INT NOT NULL AUTO_INCREMENT,
islemTarih DATETIME NOT NULL DEFAULT '00-00-0000 00:00:00',
onay enum('y','n') DEFAULT NULL,
kullaniciNo INT NOT NULL,
PRIMARY KEY (islemNo),
FOREIGN KEY(kullaniciNo) REFERENCES Kullanici (kullaniciNo)
);
CREATE TABLE IslemDetay(
islemDetayNo INT NOT NULL AUTO_INCREMENT,
islemNo INT NOT NULL,
malzKod VARCHAR(20) NOT NULL,
adet INT NOT NULL DEFAULT '0',
PRIMARY KEY (islemDetayNo),
FOREIGN KEY(islemNo) REFERENCES Islem(islemNo),
FOREIGN KEY(malzKod) REFERENCES Malzeme(malzKod)
);
CREATE TABLE Admin (
user VARCHAR(20),
password VARCHAR(20)
);
```

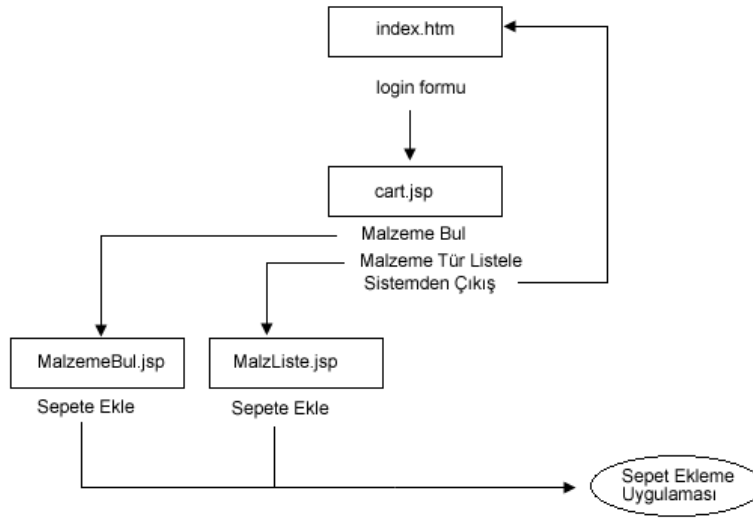
Şekil 5.7. Malzeme veritabanı ve tablo oluşturma script'i.

5.2. Programın JSP Dilinde Geliştirilmesi

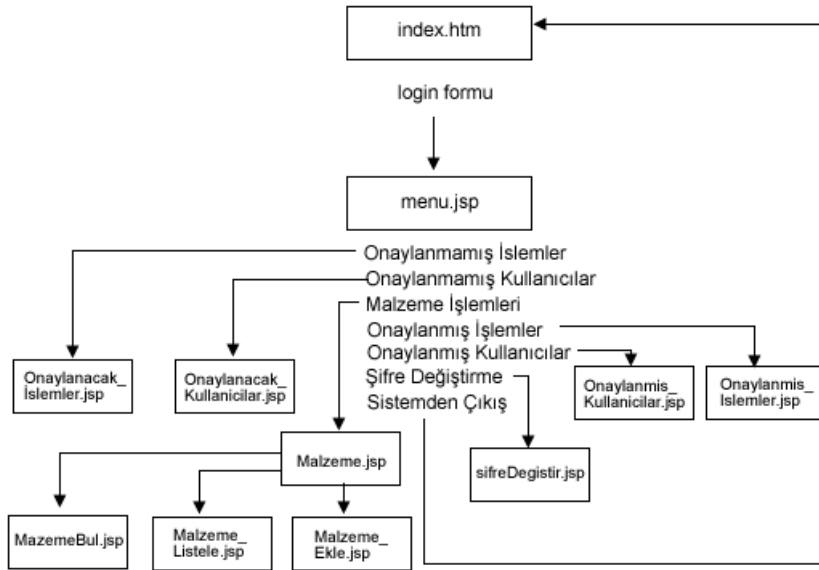
JSP dilinde programın yazılımı, JSP mimarisi, sınıf diyagramları ve sınıf kodları aşamalarından oluşmaktadır.

5.2.1. JSP mimarisi

Programın hem kullanıcı arayüzü hem de admin arayüzü ayrıdır. Her bir işlem için ayrı (oturum) açıldığı için iki ayrı yapı kullanılmıştır. Kullanıcı yapısı Şekil 5.8' ve Şekil 5.9'daki gibidir.



Şekil 5.8. Kullanıcı mimarisi



Şekil 5.9. Admin Mimarisi

5.2.2. Programın sınıf (class) diyagramları

Programda, kullanıcı sınıfı(Şekil 5.10), malzeme sınıfı(Şekil 5.11), işlem sınıfı(Şekil 5.12), cart sınıfı(Şekil 5.13) ve satır sınıfları(Şekil 5.14) kullanılmıştır.

Kullanici
- kullanıcıAd:String - sifre: String - ad:String - soyad: String - email:String - görev:String
+ Kullanici() + Kullanici(kullanıcıAd :String, sifre :String, ad :String, soyad: String, email: String, görev :String) + setKullanıcıAd(kullanıcıAd String) + getKullanıcıAd():String + setSifre(sifre String) + getSifre():String + setAd(ad String)

```

+ getAd(): String
+ setSoyad(soyad :String)
+ getSoyad():String
+ setEmail(email :String)
+ getEmail():String
+ setGorev(gorev :String)
+ getGorev():String

```

Şekil 5.10. Kullanici sınıf diyagramı

Malzeme

```

- malzKod:String
- malzTip:String
- marka: String
- model:String
- ozellik:String
- miktar:int

```

```

+ Malzeme()
+ Malzeme(malzemeKod :String, malzTip: String,
           marka :String, model :String,
           ozellik :String, miktar: int)
+ setMalzKod(malzKod: String)
+ getMalzKod():String
+ setMalzTip(malzTip: String)
+ getMalzTipe():String
+ setMarka(marka String)
+ getMarka(): String
+ setModel(model :String)
+ getModel():String
+ setOzellik(ozellik :String)
+ getOzellik():String
+ setMiktar(miktar :int)
+ getMiktar():int

```

Şekil 5.11. Malzeme Sınıf Diyagramı

Islem
- islemNo: Islem - kullanıcı: Kullanici - satirlar: Vector - islemTar:Date
+ Islem() + Islem(islemNo: int, kullanıcı: Kullanici, satirlar: Vector, islemtar: Date) + setIslemNo(islemNo: int) + getIslemNo():int + setKullanici(kullanici: Kullanici) + getKullanici():Kullanici + setSatirlar(Vector satirlar) + getSatirlar():Vector + setIslemTar(islemTar:Date) + getIslemTar():Date

Şekil 5.12. Islem sınıf diyagramı

Cart
- items:Vector
+ Cart() + Cart(satirlar:Vector) + setItems(satirlar:Vector) + getItems():Vector + addItem(item:Satir) + removeItem(item:Satir)

Şekil 5.13. Cart sınıf diyagramı

Satir
- malzeme:Malzeme - adet: int
+ Satir() + Satir(malzeme:Malzeme)

```

+ setMalzeme(malzeme: Malzeme)
+ getMalzeme():Vector
+ setMiktar(miktar:int)
+ getMiktar():int
+ getToplamMiktar():int

```

Şekil 5.14. Satir sınıf diyagramı

5.2.3. Programın sınıf kodları

Tüm sınıflar malz/base dizinine kaydedilmiştir. Bu nedenle malz.base şeklinde sınıf yolları belirtilmiştir. Sınıf değişkenleri private tanımlanmıştır. Bu nedenle herbir sınıf değişkeni için get ve set metodları tanımlanmıştır. Ayrıca sınıf yapılandırıcı (constructor) bileşeni herbir sınıf için tanımlanmıştır.

5.2.3.1. Malzeme sınıfı

```

package malz.base;
public class Malzeme {
private String malzKod;
private String malzTip;
private String marka;
private String model;
private String ozellik;
private int miktar;
public Malzeme(){
public Malzeme(String malzKod, String malzTip, String marka, String model,
                String ozellik, int miktar) {
this.malzKod=malzKod;
this.malzTip=malzTip;
this.marka=marka;
this.model=model;
this.ozellik=ozellik;
this.miktar=miktar;

```

```
}  
public String getMalzKod(){  
    return malzKod;  
}  
public void setMalzKod(String malzKod){  
    this.malzKod=malzKod;  
}  
public String getMalzTip(){  
    return malzTip;  
}  
public void setMalzTip(String malzTip) {  
    this.malzTip=malzTip;  
}  
public String getMarka(){  
    return marka;  
}  
public void setMarka(String marka){  
    this.marka=marka;  
}  
public String getModel(){  
    return model;  
}  
public void setModel(String model){  
    this.model=model;  
}  
public String getOzellik(){  
    return ozellik;  
}  
public void setOzellik(String ozellik){  
    this.ozellik=ozellik;  
}  
public int getMiktar(){  
    return miktar;
```

```
}  
public void setMiktar(int miktar){  
    this.miktar=miktar;  
}  
}
```

5.2.3.2. Islem sınıfı

```
package malz.base;  
import java.util.*;  
public class Islem {  
    private int islemNo;  
    private Kullanici kullanici;  
    private Date tarih;  
    private ArrayList satirlar;  
    public Islem() { }  
    public Islem(Kullanici kullanici,Date tarih, ArrayList satirlar) {  
        this.islemNo=0;  
        this.kullanici=kullanici;  
        this.tarih=tarih;  
        this.satirlar=satirlar;  
    }  
    public void setIslemNo(int islemNo) {  
        this.islemNo=islemNo;  
    }  
    public int getIslemNo(){  
        return islemNo;  
    }  
    public void setKullanici(Kullanici kullanici) {  
        this.kullanici=kullanici;  
    }  
    public Kullanici getKullanici(){  
        return kullanici;  
    }  
}
```

```
}  
public void setTarih(Date tarih) {  
    this.tarih=tarih;  
}  
public Date getTarih(){  
    return tarih;  
}  
public void setSatirlar(ArrayList satirlar) {  
    this.satirlar=satirlar;  
}  
public ArrayList getSatirlar(){  
    return satirlar;  
}  
public int getToplamAdet(){  
    int toplam=0;  
    for(int i=0; i<satirlar.size();i++)  
    {  
        Satir satir=(Satir)satirlar.get(i);  
        toplam+=satir.getAdet();  
    }  
    return toplam;  
}  
}
```

5.2.3.3. Cart sınıfı

```
package malz.base;  
import java.sql.*;  
import java.util.ArrayList;  
public class Cart{  
    private ArrayList items;  
    public Cart(){  
        items=new ArrayList();
```

```
}  
public Cart(ArrayList items){  
    this.items=items;  
}  
public void setItems(ArrayList items){  
this.items=items;  
}  
public ArrayList getItems(){  
return items;  
}  
public synchronized void addItem(Satir satir)  
{  
String kod=satir.getMalzeme().getMalzKod();  
int adet=satir.getAdet();  
for(int i=0;i<items.size();i++){  
    Satir satirim=(Satir)items.get(i);  
    if (satirim.getMalzeme().getMalzKod().equals(kod)) {  
        satirim.setAdet(adet);  
        return;  
    }  
}  
items.add(satir);  
}  
public synchronized void removeItem(Satir satir)  
{  
    String kod=satir.getMalzeme().getMalzKod();  
    for (int i=0;i<items.size();i++)  
    {  
        Satir satirim=(Satir)items.get(i);  
        if (satirim.getMalzeme().getMalzKod().equals(kod))  
        {  
            items.remove(satirim);  
            return;  
        }  
    }  
}
```



```

        }
    }
}
public synchronized void addItem2(Malzeme malzeme)
{
items.add(malzeme);
}
}

```

5.2.3.4. Kullanici simfi

```

package malz.base;
import java.util.*;
public class Kullanici {
private int kullaniciNo;
private String kullaniciAd;
private String sifre;
private String ad;
private String soyad;
private String email;
private String gorev;
private Date tarih;
public Kullanici(){
this.kullaniciNo=0;
}
public Kullanici(String kullaniciAd,String sifre,String ad,String soyad,String
email,String gorev)
{
this.kullaniciAd=kullaniciAd;
this.sifre=sifre;
this.ad=ad;
this.soyad=soyad;
this.email=email;
}
}

```

```
this.gorev=gorev;
}
public void setKullaniciNo(int kullaniciNo)
{
this.kullaniciNo=kullaniciNo;
}
public int getKullaniciNo()
{
return kullaniciNo;
}
public void setKullaniciAd(String kullaniciAd)
{
this.kullaniciAd=kullaniciAd;
}
public String getKullaniciAd()
{
return kullaniciAd;
}
public void setSifre(String sifre)
{
    this.sifre=sifre;
}

public String getSifre()
{
    return sifre;
}
public void setAd(String ad)
{
this.ad=ad;
}
public String getAd()
{
```

```
return ad;
}
public void setSoyad(String soyad)
{
this.soyad=soyad;
}
public String getSoyad()
{
return soyad;
}
public void setEmail(String email)
{
this.email=email;
}
public String getEmail()
{
return email;
}
public void setTarih(Date tarih)
{
this.tarih=tarih;
}
public Date getTarih()
{
return tarih;
}
public void setGorev(String gorev)
{
this.gorev=gorev;
}
public String getGorev()
{
return gorev;
```

```
}  
}
```

5.2.3.5. Satir sınıfı

```
package malz.base;  
public class Satir{  
    private Malzeme malzeme;  
    private int adet;  
    public Satir(){ }  
    public Satir(Malzeme m, int a){  
        malzeme = m;  
        adet = a;  
    }  
    public void setMalzeme(Malzeme m){  
        malzeme = m;  
    }  
    public Malzeme getMalzeme(){ return malzeme; }  
    public void setAdet(int a){  
        adet = a;  
    }  
    public int getAdet() { return adet; }  
}
```

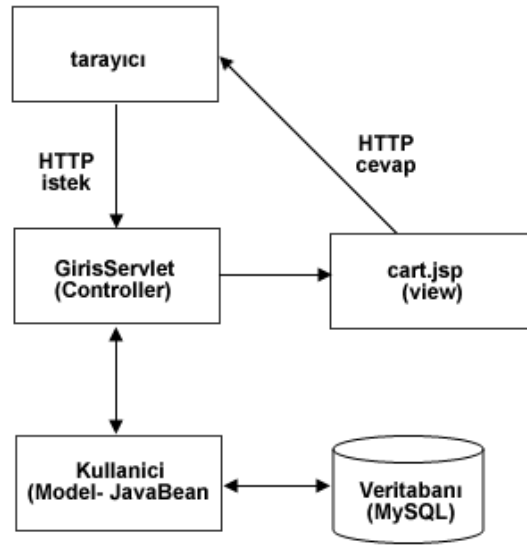
5.2.3.6. Admin sınıfı

```
package malz.base;  
public class Admin{  
    private String user;  
    private String password;  
    public Admin(){ }  
    public Admin(String user, String password){  
        this.user=user;
```

```
this.password=password;
}
public void setUser(String user){
this.user=user;
}
public String getUser()
{
return user;
}
public void setPassword(String password){
this.password=password;
}
public String getPassword(){
return password;
}
}
```

5.2.4. Program altyapısı

Bu programda, MVC (Model View Controller) modeli ve Java dili kullanılmış olup J2EE mimarisi üzerinde program geliştirilmiştir. Kullanıcı arayüzü tasarımı, Java Server Pages (JSP) ile yapılmaktadır. Kullanıcı ile uygulama arasındaki veri akışı için oturum (session) nesneleri kullanılmıştır. Oturum (session) işlemleri servlet sunucu (servlet container) tarafından gerçekleştirilmektedir. Projedeki login işleminin MVC modeline göre akışı şekil 5.15'deki gibidir.



Şekil 5.15. Programda MVC modeli kullanımı

Kullanıcı tarafından girilen kullanıcı adı ve şifresi bilgileri GirişServlet java dosyasına gönderilir. Değişken değerlerinin gönderilmesi için form tanımlanarak, girilen değişken değerleri post metodu ile malz/cart klasörü altındaki GirişServlet adındaki servlet dosyasına gönderilir. Bu işlem için index.jsp sayfasında aşağıdaki satırlar kullanılmıştır.

```
<form method="post" action="/murach/servlet/malz.cart.GirişServlet">
```

```
<tr>iş
```

```
<td width="37%">Kullanıcı Adı</td>
```

```
<td width="63%"><input name="kullaniciAd" type="text" size="11"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Şifre</td>
```

```
<td><input name="sifre" type="password" size="11"></td>
```

```
</tr>
```

```
<tr>
```

```
<td height="24">&nbsp;&nbsp;&nbsp;</td>
```

```

<td><select name="oturumTip" class="style4" id="oturumTip">
  <option value="normal" selected>Normal</option>
  <option value="admin">Admin</option>
</select></td>
</tr>
<tr>
  <td height="15" colspan="2" ><p align="center">&nbsp;</p></td>
  <td height="15" ><input type="submit" value="Giriş"></td>
</tr>
</form>

```

5.2.4.1. GirişServlet (Controller)

GirişServlet dosyasına gönderilen veriler database bağlantısı yapılarak verilerin uygunluğu kontrol edilir. Eğer veriler doğru ise cart.jsp dosyasına hatalı ise login sayfasına hata mesajı gönderilir. Bu işlem için aşağıdaki kod satırları kullanılmıştır.

5.2.4.1.1 GirişServlet sınıf kodları

```

package malz.cart;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import malz.base.*;
import malz.data.*;
import java.util.ArrayList;
public class GirişServlet extends HttpServlet{
  private Connection connection=null;
  public void init() throws ServletException{
    try{
      Class.forName("org.gjt.mm.mysql.Driver");
      String yol="jdbc:mysql://localhost/malzeme";

```

```

        String username="root";
        String password="";
        connection=DriverManager.getConnection(yol,username,password);
    }
    catch(ClassNotFoundException e) {
        System.out.println("Database surucusu bulunamadi!");
    }
    catch(SQLException e) {
        System.out.println("Database baglantisi yapilamadi!");
    }
}
public void destroy() {
    try {
        connection.close();
    }
    catch(SQLException e) {
        System.out.println("Database baglantisi kapatilamadi. Hata
mesaji"+e.getMessage());
    }
}
public void doGet(HttpServletRequest request,HttpServletResponse response)
        throws IOException,ServletException
    {
        String oturumTip=request.getParameter("oturumTip");
        System.out.println(oturumTip);
        if(oturumTip.equals("normal")) this.normal(request,response);
        if(oturumTip.equals("admin")) this.admin(request,response);
    }
private void normal(HttpServletRequest request,HttpServletResponse response)
throws IOException, ServletException
    {
        HttpSession session=request.getSession(true);
        String kullanicAd=request.getParameter("kullanicAd");

```



```

session.setAttribute("kullaniciAd",kullaniciAd);
String sifre=request.getParameter("sifre");
String url="";
Kullanici kullanici=new Kullanici();
try
{
kullanici=KullaniciDB.kayitOku(connection,kullaniciAd,sifre);
if (kullanici.getKullaniciNo()==0)
{
url="/index.jsp?mesaj=Giris hatasi.<br> Kullanici adi ve sifresini yeniden
giriniz";
}
else
{
Cart cart=new Cart();
Cart cart2=new Cart();
session.setAttribute("cart",cart);
session.setAttribute("kullanici",kullanici);
url="/cart.jsp?islem=cart";
}
}
catch(SQLException e){}
RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher(url);
dispatcher.forward(request,response);
}
private void admin(HttpServletRequest request,HttpServletResponse
response) throws IOException, ServletException
{
String user=request.getParameter("kullaniciAd");
String password=request.getParameter("sifre");
HttpSession session=request.getSession(true);
String url="";

```

```

try
{
boolean isLogin=AdminDB.login(connection,user,password);
if (!isLogin)
{
url="/index.jsp?mesaj=Admin Giriş hatası.<br> Kullanıcı adı ve şifresini
yeniden giriniz";
}
else
{
session.setAttribute("admin",user);
session.setAttribute("user",user);
url="/yonetim.jsp?islem=islemOnay";
}
}
catch(SQLException e){}

RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher(url);
dispatcher.forward(request,response);
}

public void doPost (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
doGet(request,response);
}
}

```

5.2.4.1.2. GirişServlet sınıfının çalışması

a) Kullanılan sınıflar import metodu ile çağrılır. Bu işlem aşağıdaki satırlar ile yapılmaktadır.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import malz.base.*;
import malz.data.*;
import java.util.ArrayList;
```

b) init metodunda ise, veritabanı bağlantısı yapılır. Bu işlem için, mysql veritabanı tanımlanır. Kullanıcı adı root ve şifre alanı boş bırakılır. Bu işlem aşağıdaki satırlar ile yapılmaktadır.

```
public void init() throws ServletException {
    try {
        Class.forName("org.gjt.mm.mysql.Driver");
        String yol="jdbc:mysql://localhost/malzeme";
        String username="root";
        String password="";

        connection=DriverManager.getConnection(yol,username,password);
    }
    catch(ClassNotFoundException e) {
        System.out.println("Database surucusu bulunamadi!");
    }
    catch(SQLException e) {
        System.out.println("Database baglantisi yapilamadi!");
    }
}
```

c) destroy metodunda ise, dosya ile işlem bittiğinde bağlantının kesilmesi sağlanır. Bu işlem connection.close() satırı ile gerçekleştirilir.

d) doGet metodunda uygulamanın asıl programlama işlemleri gerçekleştirilir. Yeni bir session nesnesi tanımlanır. Değişken değerleri sınıf değişkenlerine atanır. OturumTip değişkeninin değeri normal ise normal metoduna, admin ise metoduna yönlendirilir. Bu işlem aşağıdaki satırlar ile yapılmaktadır.

```
HttpSession session=request.getSession(true);
String kullanıcıAd=request.getParameter("kullanıcıAd");
String sifre=request.getParameter("sifre");
```

Daha sonra kullanıcı nesnesi tanımlanır. Try bloğu arasında KullanıcıDB sınıfından kayıtOku metodu çağrısı yapılarak kullanıcıAd ve sifre değerlerinin veritabanında olup olmadığı kontrol edilir.

```
Kullanıcı kullanıcı=new Kullanıcı();
try
{
    kullanıcı=KullanıcıDB.kayıtOku(connection,kullanıcıAd,sifre);
    if (kullanıcı.getKullanıcıNo()==0)
    {
```

Kullanıcı kontrolü ise KullanıcıDB sınıfının kayıtOku metodu ile yapılmaktadır. Aşağıda kayıtOku metodunun kodları görülmektedir.

```
public static Kullanıcı kayıtOku(Connection connection, String kullanıcıAd, String
sifre) throws SQLException{
Kullanıcı kullanıcı=new Kullanıcı();
String query="select * from kullanıcı where kullanıcıAd='"+kullanıcıAd+"' and
sifre='"+sifre+"' and onay='y'";
Statement st=connection.createStatement();
ResultSet rs=st.executeQuery(query);
if (rs.next())
{
```

```

int kullanıcıNo=rs.getInt("kullanıcıNo");
String ad = rs.getString("ad");
String soyad = rs.getString("soyad");
String email = rs.getString("email");
String gorev = rs.getString("gorev");
kullanıcı=new Kullanıcı(kullanıcıAd,sifre,ad,soyad,email,gorev);
kullanıcı.setKullanıcıNo(kullanıcıNo);
}
rs.close();
st.close();
return kullanıcı;
}

```

KullanıcıOku metodundan kullanıcı nesnesi döndürülürse RequestDispatcher nesnesi ile cart.jsp dosyasına yönlendirme yapılır. Bu işlemin kod satırları aşağıdaki gibidir.

```

String url= "/cart.jsp?islem=cart";
RequestDispatcher dispatcher = getServletContext().getRequestDispatcher(url);
dispatcher.forward(request,response);

```

5.2.4.2. Kullanıcı sınıfı (model)

MVC mimaride, javabean yani java sınıfları model katmanını oluştururlar. Herhangi bir java sınıfının javabean olabilmesi üç özelliğinin bulunması gerekir Bunlar;

- a) Parametere değeri olmayan yapılandırıcının (constructor) olması.
- b) Sınıf değişkenlerinin public tipte olmaması.
- c) Değişken değerlerine değeri atama ve değişkenleri kullanabilmesi amacıyla set ve get metodlarının bulunması.

5.2.4.3. Sepet.jsp (View)

Programda GirişServlet dosyasında session nesnesine atanan kullanıcı nesnesinin değişken değerlerinin gösterildiği jsp dosyasıdır. Web uygulamalarında programcılar

ile tasarımcıların yaptıkları işlerin kolaylaştırılması amacıyla kullanıcı tanımlı özel im(tag) ler geliştirilmiştir. Tag Library Descriptor kısaca TLD olarak adlandırılır. TLD kullanımının bazı avantajları vardır. Bu avantajlardan bir tanesi, TLD'ler java kodları içermezler. JSP sayfalarının daha az kod ile yazılmamı sağlarlar. Ayrıca bu tag'ler JSP sayfalarında birden çok yerde kullanılabilir.

Geliştirilen programda, cart.jsp sayfası MVC mimaride görünüm kısmını oluşturur. Cart.jsp dosyasında malzeme.jsp sayfası gömülü kullanılır. Özel tag'lerin kullanıldığı yer malzeme.jsp dosyasıdır. Bu işlem aşağıdaki kod satırları ile gerçekleştirilir.

```
<%@ taglib uri="/web-inf/tlds/malzeme.tld" prefix="mlz" %>
<table width="467" border="1" cellpadding="0" cellspacing="0">
  <tr valign="middle" align="center" bgcolor="#647E92">
    <td width="93"><span class="style7">KOD</span></td>
    <td width="93" class="style7">TÜR </td>
    <td width="94" class="style7">MARKA</td>
    <td width="94" class="style7">MODEL</td>
    <td width="93">&nbsp;</td>
  </tr>
</table>
<table width="467" cellpadding="0" cellspacing="0">
  <mlz:cart>
    <tr valign="middle" align="center" height="27">
      <td width="93" class="style4"><%=malzKod%></td>
      <td width="93" class="style4"><%=malzTip%></td>
      <td width="94" class="style4"><%=marka%></td>
      <td width="94" class="style4"><%=model%></td>
      <td width="93" class="style4">
        <% String url="/murach/servlet/malz.cart.DetayServlet?malzKod="+malzKod; %>
        <a href="<%=response.encodeURL(url)%>" Detay</a> </td>
      </tr>
    </mlz:cart>
  </table>
```

Yukarıdaki satırlarda `<mlz:cart>` ve `</mlz:cart>` kodları jsp kullanıcı tanımlı özel tag'leridir. Cart nesnesinin değişkenleri `<%= %>` tag'leri arasında belirtilerek değerler ekranda gösterilir. Ancak özel tag'lerin kullanılabilmesi için TLD dosyasının adı ve yolu belirtilmesi gerekir. Bunun için sayfanın başında tld dosyası belirtilir. TLD dosyasının yolunun belirtilmesi aşağıdaki gibidir.

```
<%@ taglib uri="/web-inf/tlds/malzeme.tld" prefix="mlz" %>
```

5.2.4.3.1. TLD dosyası

TLD dosyası XML türünde bir belgedir. Jsp tag'lerinin ve ilgili tag sınıflarının tanımlandığı yerdir. Aşağıdaki satırlar ile TLD belgesi tanımlanmıştır.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library
1.1//EN"
"http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">
<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.2</jspversion>
  <shortname>mlz</shortname>
  <uri></uri>
  <info>tag library for malzeme application</info>
  <tag>
    <name>cart</name>
    <tagclass>malz.tags.CartTag</tagclass>
    <teiclass>malz.tags.CartTEI</teiclass>
    <bodycontent>JSP</bodycontent>
  </tag>
  <tag>
    <name>sepet</name>
    <tagclass>malz.tags.SepetTag</tagclass>
```

```

    <teiclass>malz.tags.SepetTEI</teiclass>
    <bodycontent>JSP</bodycontent>
</tag>
<tag>
    <name>islem</name>
    <tagclass>malz.tags.IslemTag</tagclass>
    <teiclass>malz.tags.IslemTEI</teiclass>
    <bodycontent>JSP</bodycontent>
</tag>
<tag>
    <name>malzeme</name>
    <tagclass>malz.tags.MalzemeTag</tagclass>
    <teiclass>malz.tags.MalzemeTEI</teiclass>
    <bodycontent>JSP</bodycontent>
</tag>
<tag>
    <name>onaylananIs</name>
    <tagclass>malz.tags.OnaylananIslemTag</tagclass>
    <teiclass>malz.tags.OnaylananIslemTEI</teiclass>
    <bodycontent>JSP</bodycontent>
</tag>
</taglib>

```

5.2.4.3.2. Tag Sınıfı

JSP kullanıcı tanımlı imlerin temel özellikleri TLD dosyasında tutulurlar. Ancak yaptıkları işlevler, TLD dosyasında yolu belirtilen im (tag) sınıfları tarafından gerçekleştirilirler. Cart tag'ının(im) tld dosyasında belirtilen CartTag sınıfının kodları aşağıdaki gibidir.

```

package malz.tags;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;

```



```
import java.util.*;
import java.io.Writer;
import java.io.IOException;
import java.sql.*;
import java.text.NumberFormat;
import malz.base.*;
public class CartTag extends BodyTagSupport{
private ArrayList items;
private Iterator iterator;
private Malzeme item;
public int doStartTag(){
Cart cart=(Cart)pageContext.findAttribute("cart");
items=cart.getItems();
if (items.size()<=0) {
return SKIP_BODY;
}
else {
return EVAL_BODY_BUFFERED;
}
}
public void doInitBody() throws JspException
{
iterator=items.iterator();
if (iterator.hasNext()) {
item=(Malzeme)iterator.next();
pageContext.setAttribute("malzKod",item.getMalzKod());
pageContext.setAttribute("malzTip",item.getMalzTip());
pageContext.setAttribute("marka",item.getMarka());
pageContext.setAttribute("model",item.getModel());
pageContext.setAttribute("ozellik",item.getOzellik());
pageContext.setAttribute("adet",new Integer(item.getMiktar()));
}
}
```

```
public int doAfterBody() throws JspException {
    try
    {
        if(!iterator.hasNext()){
            JspWriter out=bodyContent.getEnclosingWriter();
            bodyContent.writeOut(out);
            return SKIP_BODY;
        }
        else
        {
            item=(Malzeme)iterator.next();
            pageContext.setAttribute("malzKod",item.getMalzKod());
            pageContext.setAttribute("malzTip",item.getMalzTip());
            pageContext.setAttribute("marka",item.getMarka());
            pageContext.setAttribute("model",item.getModel());
            pageContext.setAttribute("ozellik",item.getOzellik());
            pageContext.setAttribute("miktar",new Integer(item.getMiktar()));
            return EVAL_BODY_AGAIN;
        }
    }
    catch(IOException e){
        return SKIP_BODY;
    }
}
```

5.3. Uygulamanın PHP Dilinde Geliştirilmesi

Uygulamanın PHP dilinde yazılımı fonksiyonel olduğu için herhangi bir sınıf tanımlaması yapılmamıştır. Projenin tüm işlemleri php dosyası içinde yapılmıştır. Bu nedenle PHP dosyalarının içerikleri JSP dosyalarına göre daha karışıktır. Çünkü veritabanına bağlanma, oturum nesnesi oluşturma, karşılaştırma ve döngü gibi

işlemlerin tümü PHP dosyasında yapılmıştır. Ancak PHP dosyalarındaki bu karışıklığı gidermek amacıyla program kodları fonksiyonlara bölünerek yazılmıştır.

5.3.1. PHP kodunun HTML imleri ile birlikte kullanılması

PHP kodları HTML imleri ile birlikte içiçe kullanılır. PHP kodlarını ayırmak için `<?php` ile `?>` imleri kullanılır. Formdan gönderilen verileri almak için `$_POST[degisken_adi]` ya da `$_GET[degisken_adi]` komutu kullanılır. Formdan veri gönderilmesinin kontrolü `isset` komutu ile yapılır. Header komutu ile bir sayfadan başka bir sayfaya yönlendirme yapılabilir. `index.php` sayfasında bu komutlar kullanılmıştır. Aşağıda bu sayfanın kodları görülmektedir.

```
<?php
require ('malzeme_fns.php');
if (isset($_POST[durum]))
{
$kullaniciAd=$_POST[kullaniciAd];
$sifre=$_POST[sifre];
$oturumTip=$_POST[oturumTip];
if ($oturumTip=='normal')
{
$durum=kullaniciOku($kullaniciAd,$sifre);
if (!$durum)
$mesaj="Kullanici adi ve sifre hatasi. Lütfen tekrar deneyin";
else
{
$kullaniciNo=kullaniciNoGetir($kullaniciAd);
Header("Location:cart.php?kullaniciAd=$kullaniciAd&&kullaniciNo=$kullaniciNo"
);
}
}
}
?>
```

```

...
...
...
<form method="post" action="index.php">
  <tr>
    <td width="9%"><p class="style3">&nbsp;</p></td>
    <td width="28%">Kullanıcı Adı</td>
    <td width="63%"><input name="kullaniciAd" type="text" class="style3"
id="kullaniciAd" size="11"></td>
  </tr>
  <tr>
    <td><p class="style3">&nbsp;</p></td>
    <td>Şifre</td>
    <td><input name="sifre" type="password" class="style3" id="sifre"
size="11">
    <input name="durum" type="hidden" value="true" >
  </td>
</tr>
  <tr>
    <td height="24"><p>&nbsp;</p></td>
    <td height="24">&nbsp;</td>
    <td><select name="oturumTip" class="style4" id="oturumTip">
      <option value="normal" selected>Normal</option>
      <option value="admin">Admin</option>
    </select>
  </td>
</tr>
  <tr>
    <td height="8" colspan="2" class="style4"> <p
align="center">&nbsp;</p></td>
    <td height="8" class="style4"><p>&nbsp;</p></td>
  </tr>
  <tr>

```

```

        <td height="15" colspan="2" class="style4"><p
align="center">&nbsp;</p></td>
        <td height="15" class="style4">
        <input name="Submit" type="submit" class="style3" value="Giriş"></td>
    </tr>
</form>
</table>

```

5.3.2. PHP dilinde fonksiyon kullanımı

PHP dilinde bütün program kodları ve HTML tag'leri aynı sayfada toplandığı için bu karışıklığı gidermek için fonksiyon kullanımı geliştirilmiştir. Örneğin veritabanına bağlanma, veritabanında sorgulama, karşılaştırma gibi işlemleri başka bir dosya içerisindeki fonksiyon ile kullanılırsa, sayfadaki kod karışıklığı azalacaktır. Hatırlanacağı gibi JSP de bu tür programlama işlemleri MVC modeline göre java sınıfı olan servlet'lerde yapılmaktadır. PHP'de fonksiyon kullanımı için ilgili fonksiyonun bulunduğu dosya require komutu ile çağrılır.

```
require ('malzeme_fns.php');
```

Yukarıdaki kod satırı ile malzeme_fns.php sayfası, Java'daki import koduna benzer şekilde malzeme_fns.php dosyası çağrılır. Malzeme_fns.php dosyasında çeşitli işlemleri yerine getiren fonksiyonlar bulunmaktadır. Bu fonksiyonlar aşağıdaki gibidir.

```

<?php
function display_cart($cart)
{
echo '<table border=0 width=100%>
<tr bgcolor=cccccc>
<th>Malzeme Adi</th><th>Marka</th><th>Model</th><th
width=20%>Miktar</th><th>&nbsp;</th>
</tr>';

```

```

foreach($_SESSION[cart] as $malzKod => $miktar)
{
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="select * from malzeme where malzKod='$malzKod'";
$result=mysql_query($sorgu);
$malzTip=mysql_result($result,0,'malzTip');
$ozellik=mysql_result($result,0,'ozellik');
$marka=mysql_result($result,0,'marka');
$model=mysql_result($result,0,'model');
echo "<tr>
        <form action=\"sepete_ekle2.php\" action=\"post\">
        <td>$malzTip</td><td>$marka</td><td>$model</td>
        <td><input type=\"text\" name=\"$malzKod\" value=\"$miktar\"></td>
        <td><input type=\"hidden\" name=\"save\" value=\"true\">
        <input type=\"hidden\" name=\"malzKod\" value=\"$malzKod\">
        <input type=\"submit\" name=\"submit\" value=\"Degistir\"></td></tr>
        </form>";
}
echo "<tr><th colspan=3 align=right>&nbsp;  </th><th align=left>toplama
$_SESSION[malzeme] var</th></tr>";
echo " </table>";
}
function kull_cakisma($kullaniciAd)
{
$sonuc=false;
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="select * from kullaniciler where kullanicilerAd='$kullaniciAd'";
$result=mysql_query($sorgu);
$num_rows=mysql_num_rows($result);
if ($num_rows>0) $sonuc=true;
return $sonuc;
}

```

```

}
function kull_ekle($kullaniciAd,$sifre,$ad,$soyad,$email,$gorev)
{
$sonuc=false;
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="insert into kullanici(kullaniciAd,sifre,ad,soyad,email,gorev)
values('$kullaniciAd','$sifre',
'$ad','$soyad','$email','$gorev')";
$result=mysql_query($sorgu);
if($result)
$sonuc=true;
mysql_close($conn);
return $sonuc;
}
function kullaniciOku($kullaniciAd,$sifre)
{
$sonuc=false;
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="select * from kullanici where kullaniciAd='$kullaniciAd' and sifre='$sifre'";
$result=mysql_query($sorgu);
$num_rows=mysql_num_rows($result);
if ($num_rows>0) $sonuc=true;
mysql_close($conn);
return $sonuc;
}
function kullaniciNoGetir($kullaniciAd)
{
$sonuc=false;
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="select kullaniciNo from kullanici where kullaniciAd='$kullaniciAd'";

```

```

$result=mysql_query($sorgu);
$kullaniciNo=mysql_result($result,0,'kullaniciNo');
mysql_close($conn);
return $kullaniciNo;
}
function islemDetayEkle($malzKod,$miktar,$islemNo)
{
$durum=false;
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="insert into islemdetay(islemNo,malzKod,adet)
values('$islemNo','$malzKod','$miktar)";
$result=mysql_query($sorgu);
}
function malzemelerBul($ara)
{
$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
$sorgu="select * from malzeme where marka like '%$ara%' or malzTip like
'%$ara%";
$result=mysql_query($sorgu);
$res_array=array();
for($count=0;$row=mysql_fetch_array($result);$count++)
$res_array[$count]=$row;
return $res_array;
}
?>

```

5.3.3. MySQL veritabanına bağlanma

Php dilinde MySQL veritabanına bağlanmak kolaydır. Bunun için hazırlanmış birkaç mySQL komutu kullanmak yeterlidir. Örneğin, projede kayıtlı kullanıcıların oturum açabilmesi amacıyla kullanıcı adı ve şifresinin kontrolünün yapıldığı

malzeme_fns.php dosyası içindeki kullanıcıOku fonksiyonunda bu işlem birkaç kod satırı ile yapılmıştır. Bu işlem aşağıdaki satırlarda gösterilmiştir. MySQL veritabanına bağlanamama ya da sorgunun yanlış olmasından dolayı hatalı durumlarda programın kesilmemesi için ilgili satırların sonuna “die” kelimesi eklenir.

```
function kullanıcıOku($kullanıcıAd,$sifre)
{
$sonuc=false;
$conn=mysql_connect("localhost","root","") or die ("Veritabanına bağlanma
hatası!");
mysql_select_db("malzeme",$conn) or die ("Sorgu hatası!");
$sorgu="select * from kullanıcı where kullanıcıAd='$kullanıcıAd' and sifre='$sifre'";
$result=mysql_query($sorgu);
$num_rows=mysql_num_rows($result);
if ($num_rows>0) $sonuc=true;
mysql_close($conn);
return $sonuc;
}
```

Yukarıdaki kodlarda ilk önce mysql_connect(“localhost”,”root”) satırı ile “localhost” yani aynı yerde tanımlı MySQL veritabanına “root” şifresi ile bağlanılır. Bu işlem sonucunda true/false türünde değer döndürülür. Bu değer \$conn değişkenine atanır. Daha sonra MySQL veritabanında “malzeme” isimli veritabanına mysql_select_db("malzeme",\$conn) komut ile bağlanılır. İlgili veritabanında hangi tablo ve alanı kullanılacağını belirtmek için \$sorgu değişkenine atama yapılır. İlgili işlemleri çalıştırmak için mysql_query(\$sorgu) komutu kullanılır. Mysql_query komutu ile döndürülen dizi değerleri \$result değişkenine atanır. \$result değişkeni ile istenen işlemler yapıldıktan sonra mysql_close() komutu ile veritabanına olan bağlantı kesilir.

5.3.4. Session kullanımı

Kullanıcının sunucuya bağlanmasından itibaren sayfadan çıkıncaya kadar yaptığı tüm işlemler bir oturum(session) olarak isimlendirilir. Tanımlanmış normal bir değişken, sadece tanımlandığı sayfada geçerli olup kullanılabilirken, bir session değişkeni sitenin tüm sayfalarında geçerli olur ve herhangi bir sayfada iken session değişkenine aktarılan değer, sitenin diğer sayfalarında da işleme sokulabilir.

PHP dilinde session değişkeni atamak için `$_SESSION[değişken_adı]` komutu kullanılır.

Yapılan projede birçok sayfada session kullanılmıştır. Session değişkenine atama yapabilmek için ilk olarak php kodlarına `session_start()` komutu ile başlanması gerekir. Cart isimli bir session tanımlamak için aşağıdaki satır yazılır.

```
$_SESSION[cart]=array()
```

satırında olduğu gibi cart isimli dizi değişkeni session değişkeni olarak tanımlanmış olur. Daha sonra cart dizisindeki değişken değerlerin gösterebilmek amacıyla

```
foreach($_SESSION[cart] as $malzKod => $miktar)
```

döngüsü ile tüm değerlere ulaşılması sağlanır. Aşağıda bu işlemlerin yapıldığı satırlar gösterilmektedir.

```
<?php
session_start();
$yeni=$_GET[malzKod];
if ($yeni)
{
if(!isset($_SESSION[cart]))
{
```

```

$_SESSION[cart]=array();
$_SESSION[malzeme]=0;
$_SESSION[tutar]=0.00;
}
if(isset($_SESSION[cart][$yeni]))
{
$_SESSION[cart][$yeni]++;
}
else
{
$_SESSION[cart][$yeni]=1;
}
$_SESSION[malzeme]=array_sum($_SESSION[cart]);
}
if(isset($_GET[durum]))
{
foreach($_SESSION[cart] as $malzKod => $miktar)
{
if($_GET[malzKod]==$malzKod)
{
if($_GET[$malzKod]==0)
unset($_SESSION[cart][$malzKod]);
else
$_SESSION[cart][$malzKod]=$_GET[$malzKod];
}
}
$_SESSION[malzeme]=array_sum($_SESSION[cart]);
}
if($_SESSION[cart])
{
echo '
<table width="465" border="0" cellpadding="0" cellspacing="0">
    <tr align="left" bgcolor="#647E92" class="style3">

```

```

        <td width="2">&nbsp;</td>
            <td width="65">KOD </td>
        <td width="65">T&Uuml;R</td>
        <td width="66">MARKA</td>
        <td width="66">MODEL</td>
        <td width="66" align="left">ADET</td>
        <td width="135" colspan="2"></td>
    </tr>;

$conn=mysql_connect("localhost","root","");
mysql_select_db("malzeme",$conn);
foreach($_SESSION[cart] as $malzKod => $miktar)
{
    $sorgu="select * from malzeme where malzKod='$malzKod'";
    $result=mysql_query($sorgu);
    $malzTip=mysql_result($result,0,'malzTip');
    $ozellik=mysql_result($result,0,'ozellik');
    $marka=mysql_result($result,0,'marka');
    $model=mysql_result($result,0,'model');
    echo " <tr align=left valign=middle class=style9 height=27>
        <td>&nbsp;</td>
        <td><p>$malzKod</p></td>
        <td><p>$malzTip</p></td>
        <td><p>$marka</p></td>
        <td><p>$model</p></td>";
    echo "
        <form action=\"cart.php\" method=\"get\">
            <td align=left><p><input name=\"$malzKod\" type=text value=$miktar
size=4></p></td>
            <td width=68>
                <p><input type=submit name=Submit value=Degistir>
                <input name=malzKod type=hidden value=$malzKod>
                <input type=\"hidden\" name=\"durum\" value=\"true\">
                <input type=\"hidden\" name=\"islem\" value=\"sepet\">

```

```

        </p></td>
    </form>
    </tr>";
}
echo " <tr><th colspan=3 align=right>&nbsp;</th><th align=left>toplam
$_SESSION[malzeme] malzeme sectiniz.</th></tr>";
echo " </table>";
echo "
    <table width=465 border=0 cellpadding=0 cellspacing=0>
    <tr>
    <td>&nbsp;</td> <td>&nbsp;</td>
    </tr>
    <tr>
    <td>&nbsp;</td> <td>&nbsp;</td>
    </tr>
    <tr>
    <td>&nbsp;&nbsp;&nbsp;<a href=\"islem.php\"><img
src=\"images/onayla.png\" border=0 /></a></td>
    <td>&nbsp;</td>
    </tr>
    </table>";
}
else
echo "Sepetinizde ürün bulunmamaktadır";
?>

```

Görüldüğü üzere PHP programlama dili ile JSP diline göre aynı işlemi daha az kod ile kolayca yapılabilir. Ancak PHP dili ile geliştirilen sayfalar, JSP sayfalarına göre daha karmaşık olmaktadır. Çünkü bütün programcılık işlemleri sadece PHP kodu ile HTML kodunun iç içe kullanıldığı sayfalarda yapılmaktadır.

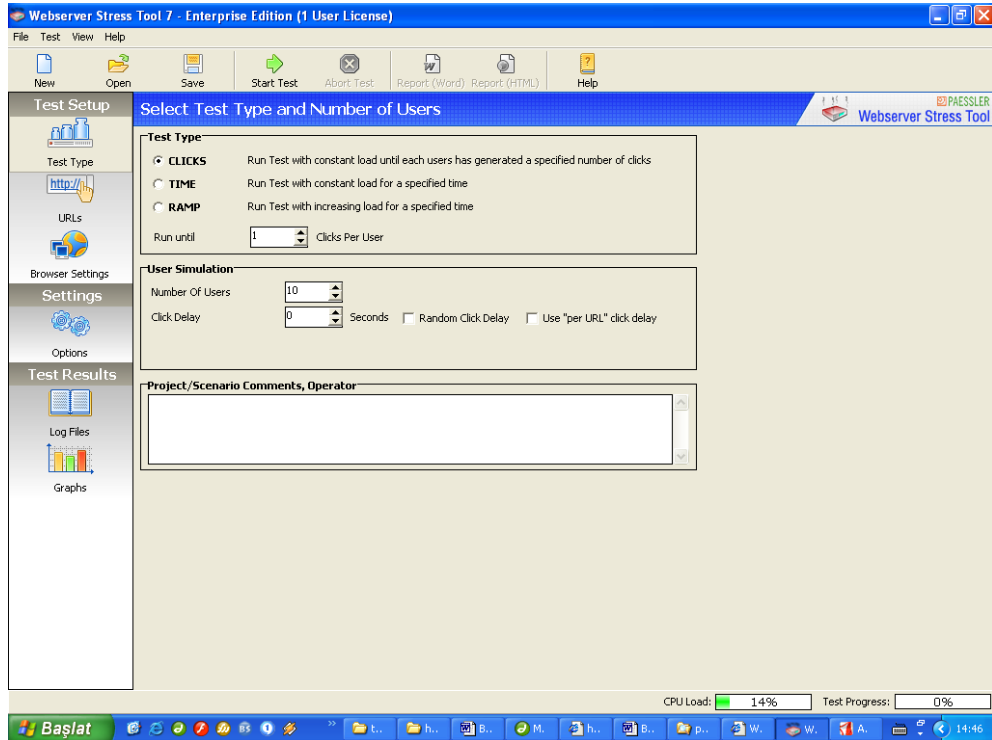
BÖLÜM 6. PERFORMANS ANALİZİ

Bu tezde, JSP ile PHP programlama dillerinin performanslarının analizi yapılarak sonuçların ortaya konulması amaçlanmıştır. Bunun için online malzeme stok takip projesi her iki dilde geliştirilmiştir.

Web uygulamalarında performans analizi için sanal test araçları kullanılmaktadır. Bu projede performans analizi için Webserver Stress Tool programı kullanılmıştır. Webserver Stress Tool ile sanal kullanıcıların sayısı ve herbir kullanıcının sayfa istek sayısı belirtilerek sınırlı sürede test yapılabilmektedir.

6.1. Webserver Stress Tool Programının Tanıtımı

Program penceresi test tipi, ayarlar ve sonuçlar bölümünden oluşur. (Şekil 6.1)



Şekil 6.1. Webserver stres tool programı arayüzü

6.1.1. Test bölümü

Bu bölümde yapılacak olan testlerle ilgi düzenlemeler yapılır. Test türü, kullanıcı simülasyonu düzenlemeleri yapılır.

6.1.1.1. Test türü

Üç türde test yapılabilmektedir.

1. Clicks: Öncesinde belirtilen tıklama sayısı bitinceye kadar test yapılır. Clicks testi bazı web sayfa dizisinin testi için uygun seçimdir.
2. Time: Belirli bir sürede, belirli kullanıcının tıklama sayısınca test yapılır. Bu test türü ile web sunucusunun tam yükleme ile test yapılır.
3. Ramp: Belirli bir sürede, 1 kullanıcıdan belirlenen kullanıcıya kadar artan şekilde test yapılır. (Şekil 6.2)

Test Type

CLICKS Run Test with constant load until each users has generated a specified number of clicks

TIME Run Test with constant load for a specified time

RAMP Run Test with increasing load for a specified time

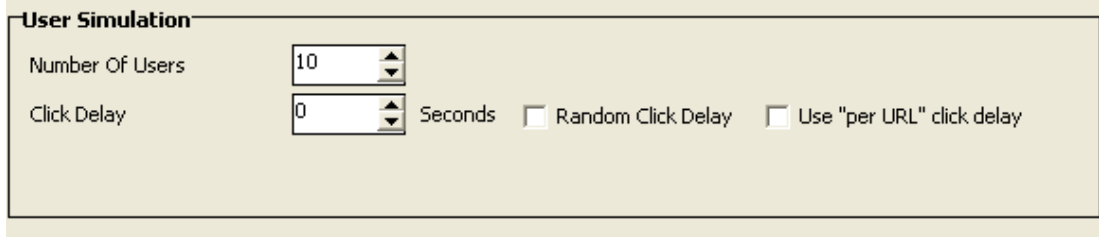
Run until Clicks Per User

Şekil 6.2. Webserver test tool programı test türü bölümü

6.1.1.2. Kullanıcı simülasyonu

Number of Users: Web sitesini aynı anda kullanabilecek kullanıcı sayısı belirtilir.

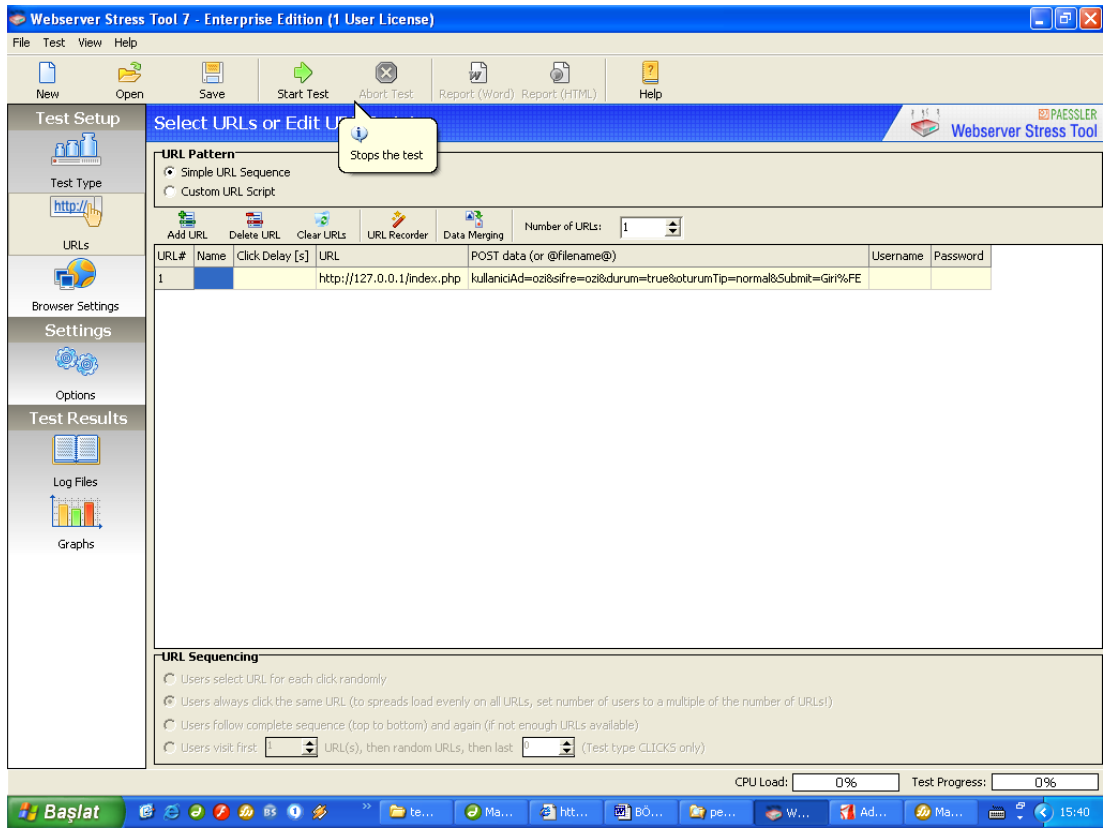
Click Delay: Herbir kullanıcının kaç saniye aralıklarla tıklama yapacağı belirtilir. (Şekil 6.3)



Şekil 6.3. Kullanıcı simülasyonu

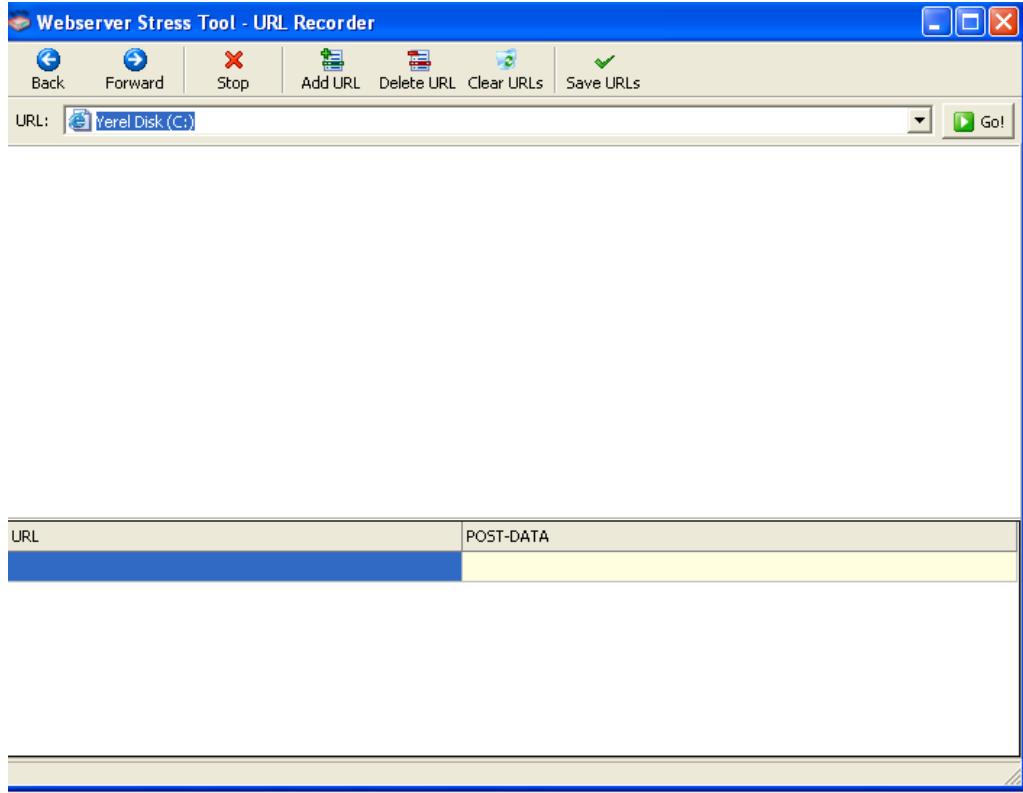
6.1.2. URL seçimi

Bu bölümde testi yapılacak web sayfaları seçilir. Gerekirse web sayfası ile ilgili post ya da get metodu ile gönderilen veriler de belirtilebilir. (Şekil 6.4)



Şekil 6.4. URL seçme ve kaydetme

Ayrıca URL Recorder kısayolu ile bir web sitesinde yapılan her türlü hareket kaydedilerek, ilgili web sayfalarının testlerde kullanılması sağlanır. (Şekil 6.5)



Şekil 6.5. URL recorder (kaydedici)

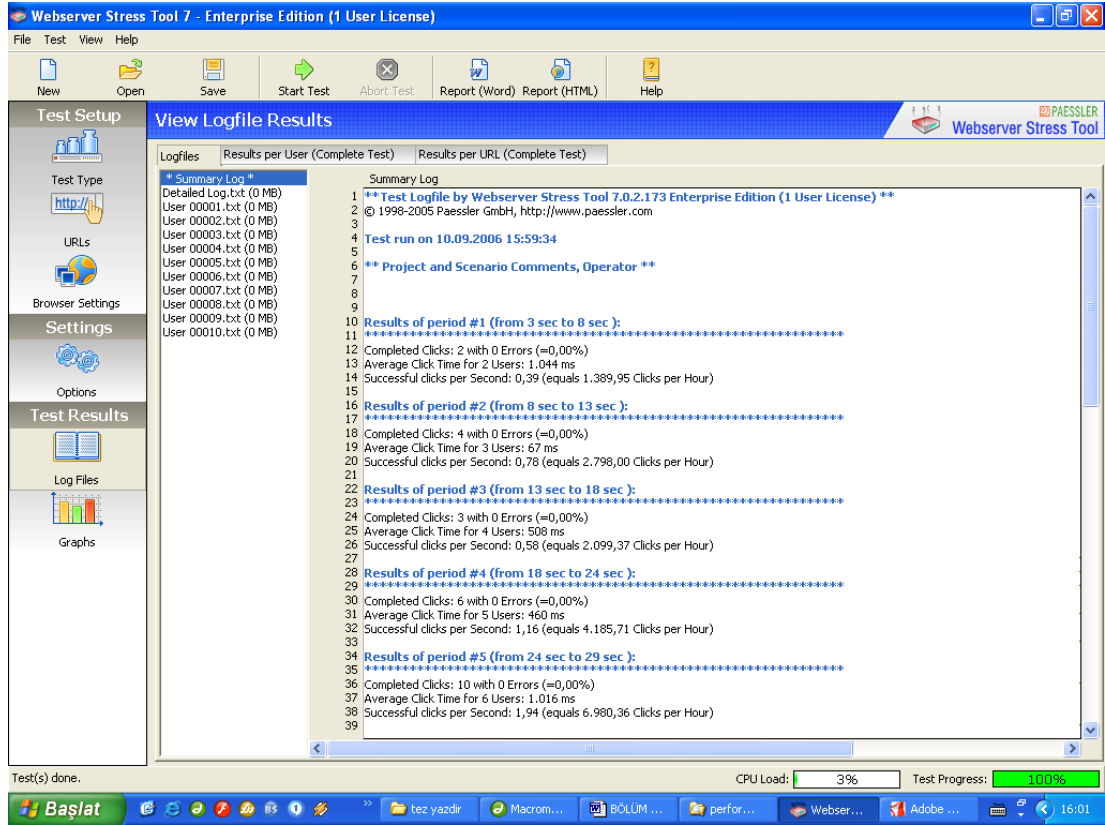
6.1.3. Test sonuçları bölümü

İlgili url seçilerek ve gerekli parametreler yapıldıktan sonra Start Test kısayoluna tıklanarak test başlatılır. Test süresince sayfa altında CPU Load kısmında işlemcinin performansı gösterilir. Sonuçların daha gerçekçi olması için test süresince işlemcinin %100 kullanım sınırına ulaşmaması gerekir.

Test bittikten sonra test results kısmında test sonuçları görüntülenir. Test sonuçları 2 şekilde görüntülenir:

6.1.3.1. Log files

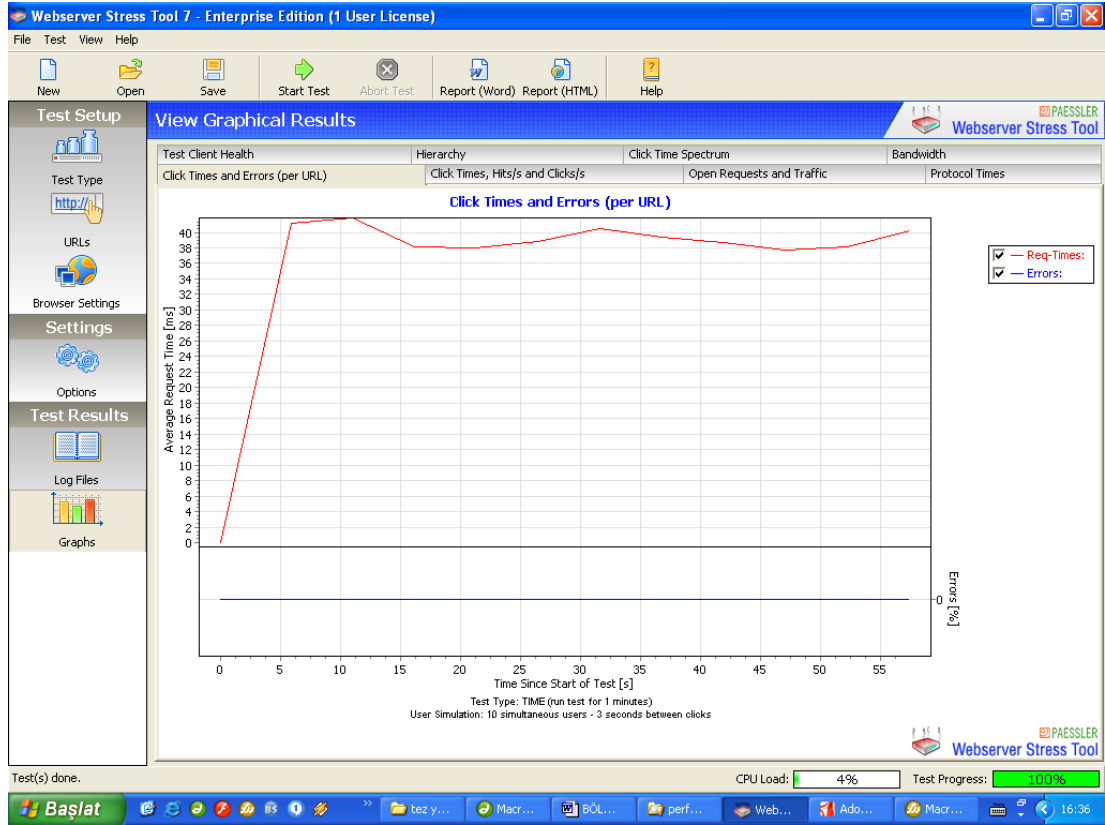
Test süresince oluşan her türlü değişimin kullanıcı sayısına göre görüntülediği yerdir. (Şekil 6.6)



Şekil 6.6. Log files

6.1.3.2. Graphs

Bu kısımda, belirlenen sürede kullanıcının tıklama süresi ve varsa hatalar grafiği, network trafiği, işlemcinin performansı, band genişliği grafiği gibi çeşitli yönden analizler görüntülenmektedir. (Şekil 6.7)



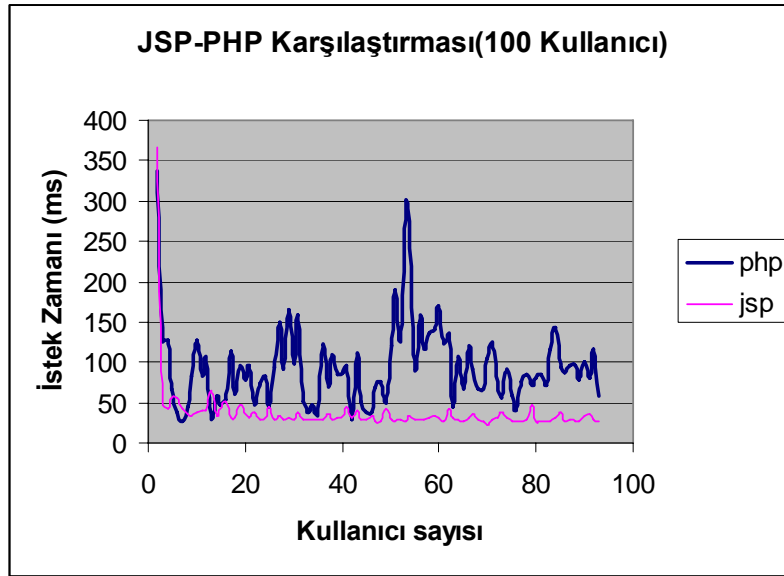
Şekil 6.7. Graphs bölümü

6.2. Sonuçların Analizi

Webservers Stress Tool programı kullanılarak PHP ve JSP dilinin her ikisinde ayrı ayrı geliştirilen online malzeme stok takip programının MySQL veritabanında kayıt sorgulama ve kayıt ekleme işlemleri test edilmiştir. Bunun için 100 ve 500 aktif kullanıcının belirli süre içinde kullanıcı sayısı artarak belirli saniye aralıklarda sayfaları tıklaması senaryoları yapılmıştır. Bu test işleminde kullanılan bilgisayarda, Pentium M 1.4 MHz işlemci kullanılmıştır. İşlemcinin L1 cache bellek kapasitesi 64 kB, L2 cache bellek kapasitesi ise 2048 kB'dır. Bellek kapasitesi 256 MB, türü DDR ve hızı ise 333 Mhz'dir. Kullanılan diskin kapasitesi 30 GB, dönüş hızı ise 4200 RPM'dir. Anakart veriyolu hızı ise 400 MHz'dir. İşletim sistemi olarak Windows XP Pro kullanılmıştır. Herbir test işleminden önce Apache ve Tomcat web sunucular kapatılıp yeniden başlatılmıştır.

6.2.1. 100 kullanıcıya kadar sayfanın yüklenmesi

10 dakikada 1 kullanıcıdan 100 kullanıcıya kadar sayfanın test işlemi yapılmıştır. Her bir kullanıcı 30 saniye aralıklarla sayfayı yeniden tıklama yapmıştır. Her iki sayfanın grafiği aşağıdaki gibidir. (Şekil 6.1)



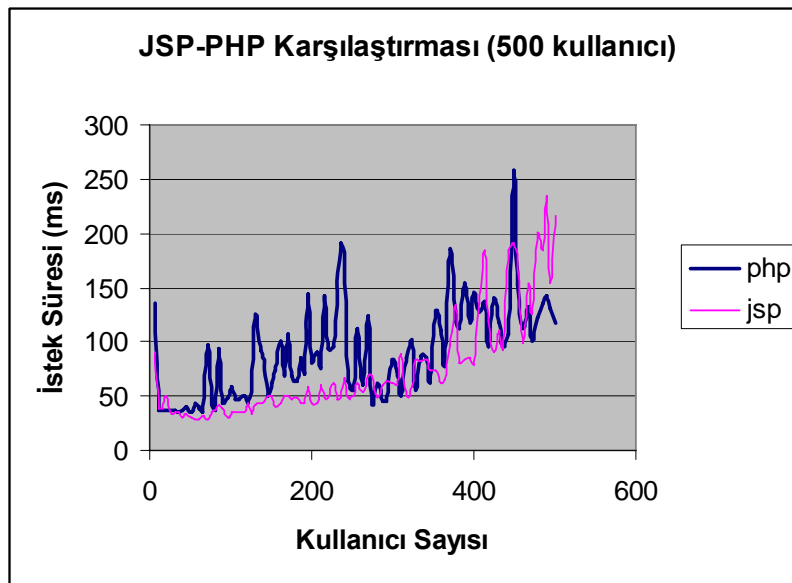
Şekil 6.8. Birim zamanda 100 kullanıcının jsp ve php sayfalarına istekte bulunduğu test sonucu

Şekil 6.8’de görüldüğü üzere, JSP sayfasının istek süresinin daha az olduğu görülmektedir. PHP sayfalarının istek süresi ise yüksek değerlerdedir. Her iki sayfanın ilk anda yüksek değer göstermesinin nedeni sayfaların ilk anda yorumlanma ve derlenme işleminden kaynaklanmaktadır. Daha sonraki kullanıcı isteklerinde ise yorumlama ve derleme işlemi yapılmadığı için istek süresi düşük kalmıştır. JSP sayfasına olan isteklerin kullanıcı sayısı arttıkça sabit bir eğri oluşturduğu görülmektedir. Bu durumda JSP sayfasının güçlü ve kararlı bir dil olduğu sonucuna varılabilir. PHP sayfasının ise, JSP sayfasına göre daha yüksek değerlerde istek süresi oluşmaktadır. PHP’nin JSP’ye göre daha yavaş olduğu söylenebilir. Ayrıca, PHP sayfasının 50 kullanıcıya kadar istek süresinin dalgalı değişim değerlerinin daha az olmasına karşın, 50 kullanıcıdan sonra PHP sayfasının istek zamanının daha büyük dalgalı değişim gösterdiği görülmektedir. Bu durum PHP sayfasının istek süresinin 50 kullanıcıyı geçtiğinde düzensiz çalıştığını göstermektedir. Buna göre

PHP dilinin kullanıcı sayısı arttıkça daha kararsız bir durum arzettiği sonucuna varılabilir. Bu durumda JSP sayfasının daha kararlı olduğu söylenebilir.

6.2.2 500 kullanıcıya kadar sayfanın yüklenmesi

10 dakikada 1 kullanıcıdan 500 kullanıcıya kadar sayfanın test işlemi yapılmıştır. Her iki sayfanın grafiği aşağıdaki gibidir. Her bir kullanıcı 30 saniye aralıklarla sayfayı tıklama yapmıştır.(Şekil 6.2)



Şekil 6.9. Birim zamanda 500 kullanıcının jsp ve php sayfalarına istekte bulunduğu test sonucu

Yukarıdaki şekilde, kullanıcı sayısı 400'e varıncaya kadar, JSP sayfası istek süresinin PHP sayfası istek süresine göre daha kısa sürdüğü görülmektedir. 400 kullanıcıdan sonra JSP istek süresinde dalgalı değişimler görülmektedir. PHP sayfası istek süresi ise 50 kullanıcıya kadar sabit bir eğri çizerken, 50 kullanıcıdan sonra yükselen bir yönde büyük değişimler gösterdiği görülmektedir. Ancak kullanıcı sayısı 50 oluncaya kadar iki dil arasında büyük farklılıklar olmadığı görülmektedir. Genel olarak JSP sayfası, PHP sayfasından daha hızlı olmakla beraber 50 kullanıcıya kadar PHP ve JSP sayfalarının birbirine yakın bir performansı olduğu söylenebilir. Bu durumda kullanıcı sayısının 50'yi geçmediği web sitelerinde PHP ve JSP dili kullanımında büyük fark yoktur ancak kullanıcı sayısının 50'den fazla olduğu web sitelerinde JSP dilinin kullanılması daha doğru tercih olabilir.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Yapılmış olan testlerde elde edilen sonuçlara göre:

1. Performans açısından: JSP sayfaları, PHP sayfalarından daha hızlıdır.
2. Kolaylık açısından: JSP sayfaları, java kodu olan servlet ve XML kodu olan özel TLD imlerini kullandığından daha kompleks bir dildir. Küçük bir projenin geliştirilmesi PHP'ye nazaran daha uzun ve daha zahmetlidir.
3. Kod gizliliği: JSP servlet sayfaları class uzantılı bytecode'a dönüştürüldüğü için sayfa kodlarının gizliliği sağlanabilmektedir. Ancak PHP sayfalarındaki PHP kodlarında böyle bir dönüştürme yoktur. Bu nedenle JSP dili ile yazılan kodlar gizlenebilir.
4. Verimlilik açısından: Projenin geliştirilmesi süresi, sayfanın hızlı çalıştırılması açısından değerlendirildiğinde, küçük projelerin geliştirilmesinde PHP, büyük ve kompleks projelerin geliştirilmesinde ise JSP dili kullanılması önerilir.

Bu tezde JSP ve servlet teknolojisi ile geliştirilen uygulamanın kompleks olmasından dolayı programın yazılım geliştirme süresi uzamıştır. Ancak JSP ve servlet dilinde son zamanlarda geliştirilen framework teknolojisi kullanılması durumunda programın yazılım geliştirme süresi kısaltılabilir.

Ayrıca programın performans testi için kullanılan uygun aracın bulunması da programın gelişim süresini etkilemiştir. Çünkü yaygın kullanılan çok sayıdaki test aracından PHP ve JSP dilinde geliştirilen uygulamaya uygun olanını bulmak için ayrı bir araştırma safhası gerçekleştirilmiştir.

Test işlemine başlamadan önce herbir web sunucunun kapatılıp yeniden başlatılması gereklidir. Sunucuların zamanla performansı düştüğünden sunucuların ilk defa çalıştırılmadığı durumlarda test işlemi hatalı olacaktır.

Web uygulamalarında teknoloji, nesne tabanlı programlamaya doğru ilerlemektedir. Microsoft, ASP teknolojisini nesne tabanlı ASP.NET teknolojisine yükseltmiştir. Uygulamada kullanılan PHP 4.0 dili nesne tabanlı teknolojiyi desteklememesine karşın, PHP grubu nesne tabanlı teknolojiyi destekleyen PHP 5.0 versiyonunu geliştirmiştir. JSP dilinin zaten nesne tabanlı bir dil olduğu bilinmektedir. Bundan sonraki çalışmalarda, web uygulamalarında nesne tabanlı dillerin çeşitli açılardan performans analizlerinin yapılmasının, web programcılarının teknoloji tercihinde önemli bir rehber olacağı inancındayım.

KAYNAKLAR

- [1] JOHNSON, R. The ups and downs of object oriented systems developments. Communication of ACM 43(10), 68-73
- [2] HENDERSON-SELLER, B. & UNHELKA. Open modeling UML. Boston, Addison Wesley, 2000
- [3] ROBERTS, Gary. Computers in Libraries, Learning Server-Side Scripting, Sep 2005. 25. 8, ProQuest Computing, sayfa 37
- [4] LOMERSON, William L., JONES, Christopher G., SCHAWAGER, Paul H., Core Web Technologies For New E-Commerce Employees, The Journal of Computer Information Systems, Winter 2004/2005, 45, 2, ABI/INFORM Global, sayfa 44
- [5] VERVAET, Erwin, JavaWorld, San Francisco, Şubat 2001, sayfa 1.
- [6] MORRISON, Mike., MORRISON, Joline., KEYS, Anthony., ACM, Eylül 2002/Vol. 45, No. 9
- [7] APTE, Varsha., HANSEN, Tony., REESER, Paul., Java: It's a good thing; Why Java isn't slow, ugly, or irrelevant, AT&T Labs, 200 Laurel Ave, Middletown, NJ 07748, USA, Ağustos 2002.
- [8] RUBINSTEIN, D., Feather in Apache's cap. [http://www.sdtimes.com/news/054/story3 .htm](http://www.sdtimes.com/news/054/story3.htm), 2002
- [9] http://www.bilisimterimleri.com/bilgisayar_bilgisi/bilgi/24.html
- [10] KURNIAWAN, B., Java For the Web with Servlets, JSP and EJB: A Developer's Guide to J2EE Solutions, New Riders Publishing, 2002
- [11] PATZER, A., Jsp Examples And Best Practices, Apress, New York, 2002
- [12] STEELMAN, A., MURACH, J., Murach's Java Servlet's and JSP, Murach, USA, 2003
- [13] <http://java.sun.com/products/jsp/overview.html>

- [14] MORKOÇ, M., Java Server Pages ve Servlet, Alfa, İstanbul, 2003
- [15] HAAN, P.D., LAVANDOWSKA L., PANDURAGA S. N., PERRUMAL K., Beginning JSP 2: From Novice to Professional, Apress, New York, 2004
- [16] MORKOÇ, M., Java Server Pages ve Servlet, S. 2-3, Alfa, İstanbul, 2003
- [17] SHASOR, G., CHACE A., RYDIN M., JSP Tag Libraries, Manning Publications, Greenwich, 2001
- [18] <http://www.phpbul.com/php-nedir.php>
- [19] <http://www.php.net/source.php?url=/usage.php>
- [20] http://news.netcraft.com/archives/2006/08/01/web_server_survey.html
- [21] <http://www.csharpnadir.com/makalegoster.asp?MIId=433>
- [22] <http://www.ulakbim.gov.tr/dokumanlar/programlama/2000php/performance.html>
- [23] <http://www.phpbuilder.com/>
- [24] <http://www.php.net>

ÖZGEÇMİŞ

16.01.1976 tarihinde Samsun'da doğan Özgür Örnek, ilk orta ve lise tahsilini aynı şehirde tamamladı. 1994 tarihinde Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Bilgisayar Sistemleri Öğretmenlik programına girdi ve 1999 yılında bitirdi. 1999 yılından itibaren eğitim sektöründe çalışma hayatına devam etmektedir.