

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ

**KURUMSAL KAYNAK PLANLAMASI VE
WEB SERVİSLERİ İLE BİR ERP UYGULAMASI**
(Yüksek Lisans Tezi)

Mustafa DÜLGERLER

İSTANBUL, 2007

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ

**KURUMSAL KAYNAK PLANLAMASI VE
WEB SERVİSLERİ İLE BİR ERP UYGULAMASI**
(Yüksek Lisans Tezi)

Hazırlayan;
Mustafa DÜLGERLER
BM2351-002

Danışman;
Yard. Doç. Dr. Turhan KARAGÜLER

İSTANBUL, 2007

YEMİN METNİ

Sunduđum Yüksek Lisans Tezimi Akademik Etik İlkelerine bađlı kalarak, hiç kimseden akademik ilkelere aykırı bir yardım almaksızın bizzat kendimin hazırladıđına and ierim. 29.03.2006

Mustafa DÜLGERLER

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

Enstitümüz Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği bilim dalı yüksek lisans öğrencilerinden BM2351-002 numaralı Mustafa DÜLGERLER “Beykent Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği’nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği “Kurumsal Kaynak Planlaması ve Web Servisleri ile Bir ERP Uygulaması” başlıklı Tezini, Yönetim Kurulumuzun Tarih ve sayılı toplantısında seçilen ve Fakülte binasında toplanan biz jüri/izleme komitesi üyeleri huzurunda, ilgili yönetmeliğin (c) bendi gereğince (.....) dakika süre ile aday tarafından takdim edilmiş ve sonuçta adayın tezi hakkında Oybirliği/Oy Çokluğu ile Kabul/Red kararı verilmiştir. İşbu tutanak, 4 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü’ne sunulmak üzere tarafımızdan düzenlenmiştir...../...../.....

JÜRİ ÜYELERİ

Danışman
Yard. Doç. Dr. Turhan KARAGÜLER

Üye
Yard. Doç. Dr. Rıfat ÇÖLKESEN

Üye
Prof. Dr. Ali OKATAN

KURUMSAL KAYNAK PLANLAMASI VE WEB SERVİSLERİ İLE BİR ERP UYGULAMASI

Mustafa DÜLGERLER

ÖZET

ERP sistemler, iş sürecindeki tüm adımların belli bir sistematığe bağlanması, organizasyon içerisindeki iş dağılımlarının otomatize edilmesi gibi işlevleri yerine getirmesi nedeniyle, çok karmaşık sistemler olarak bilinmektedirler.

Bu tez ERP sistemlerinin yapısını, bilgiyi işleme yöntemi nedeniyle yapısı belli kurallara bağlanmış ERP sistemlerinin çözümlenmesi için olası yöntemleri ve bunların sonuçlarını incelemektedir.

ERP sistemlerin geliştirilmesi ve kapasitelerinin artırılması için kullanılan yeni teknoloji Web Servisleridir.Bu sistemler web yoluyla erişilebilen, yapılacak işlemler için standart yordamlar içeren uygulama fonksiyonlarıdır.Burada ERP sistemleri ile bütünleşik Web Servislerinin kullanımı, ve getirdiği yenilikler tartışılacaktır.

Anahtar Sözcükler : ERP, Web Servisleri,XML, SOAP, e-business modellemesi

**ENTERPRISE RESOURCE PLANNING AND
A SAMPLE ERP PROGRAM USING WEB SERVICES**

Mustafa DÜLGERLER

ABSTRACT

It is well known that ERP systems (Enterprise Resource Planning) rather aim at automating business processes and codifying organisational procedures and routines.

This thesis examines the nature, consequences and some possibilities for solving ERP system “rigidity” and the relative “organisational freezing” caused by information process and flow coding.

In order to give ERP systems a greater capacity of adaptation and innovation, the characteristics of a new technology, Web Services, have been suggested which virtually allow the functions of any compatible application to be extended by using a standard component software accessible through the Web.

Key Words : ERP, Web Services, XML, SOAP, e-business modeling

İÇİNDEKİLER

Yemin Metni	
Jüri Sayfası	
Türkçe Özet ve Anahtar Kelimeler	
İngilizce Özet ve Anahtar Kelimeler (Abstract)	
Tablolar Listesi	III
Şekiller Listesi	IV
1. GİRİŞ	1
2. KURUMSAL KAYNAK PLANLAMA (ERP)	2
2.1 ERP Sistemlerinin Temel Özellikleri Nelerdir?	3
2.2 ERP'nin Gelişim Süreci	6
2.2.1 Malzeme İhtiyaç Planlaması (MRP)	7
2.2.2 Kapalı Çevrim MRP	9
2.2.3 Kapasite İhtiyaç Planlaması (CRP)	11
2.2.4 Üretim Kaynakları Planlaması	13
2.2.5 Dağıtım Kaynakları Planlaması	15
2.2.6 ERP'nin Ortaya Çıkışı	15
2.3 ERP Sistemlerinin Kullanım Nedenleri	16
2.4 Kurumları ERP Kurmaya Götüren Sebepler	20
3. ELEKTRONİK TİCARET VE WEB SERVİSLERİ	22
3.1 E-Busines (E-İş)	22
3.2 E-Commerce	23
3.4 Web Servisleri	24
3.4 Web Servislerinin Teknik Açından İncelemesi	26
3.4.1 XML Nedir?	27
3.4.2 SOAP (Simple Object Access Protokol)	29
3.4.3 UDDI (Universal Description Discovery and Integration)	32
3.4.6 Web Servislerinin Çalışma İlkesi	33
3.5 ERP ve Web Servisleri	34
3.5.1 Entegrasyon Kolaylığı	36
3.5.2 Uygulama Modeli ile Maliyetlerin Azaltılması	37
3.5.3 Web Servisleri ile ERP Uygulamalarının İyileştirilmesi	38
3.5.4 Günümüzdeki ERP Uygulamaları	38
3.5.5 ERP Sistemleri ile Web Servisli Çözümlerin Karşılaştırılması	40
i. Ölçeklenebilirlik	40
ii. Kurulum ve Uygulama Zamanı	41
iii. Devam Edebilirlik	41
iv. Güvenlik	41
v. Taşınabilirlik	42
vi. Başlangıç Maliyeti	42
vii. Devam ettirme Maliyeti ve Toplam Maliyet	42
3.6.6 ERP Web Servislerinin Ekonomik Açıdan Değerlendirilmesi	43
3.7 ERP ve Mobil Sistemler	44

4	ÖRNEK UYGULAMA İLE ERP VE WEB SERVİSLERİNİN KULLANIMININ İNCELENMESİ	46
4.1	Örnek Uygulama	49
4.1.1	Veri tabanı Yapısının İncelenmesi	50
4.1.2	Program incelemesi	53
4.2	Web servisleri dışındaki çözümlerle ERP uygulamalarının karşılaştırılması	67
5.	SONUÇ	68
	KAYNAKLAR	70
	EKLER	72
	EK A - KODLAMALAR	72
1.	Reference.vb dosyasının özeti	72
2.	_LoginAdmin metodunun SOAP Mesajı hali	77
3.	_LoginAdmin fonksiyonunun Service. vb dosyasındaki içeriğini	78
4.	_LoginAdmin Metodunun kullanımı	79
5.	_LoginUser Fonksiyonuna ait referans.vb dosyasında yer alan imzası	80
6.	_LoginUser Fonksiyonunun web servisi içerisinde yer alan tanımlaması	82
7.	_LoginUser fonksiyonuna ait SOAP Mesajları	83
8.	_GetValues Sınıfı	86

TABLO LİSTESİ

Tablo 1	B2B ve B2C Karşılaştırması	24
Tablo 2	Ana Üreticilerin web Servisi tabanlı yapılarının karşılaştırılması	39
Tablo 3	ERP sistemleri ve Web Servisleri ile desteklenmiş ERP Modellerinin karşılaştırılması	40

ŞEKİL LİSTESİ

Şekil 1	ERP Temel Özellikleri . Kavramsal Grafik (Hagman, 2000)	6
Şekil 2	MRP Sistemi (Acar, 1991)	10
Şekil 3	Kapalı Çevrim MRP Sistemi (Durmuşoğlu, 1993)	12
Şekil 4	MRP II Sistemi (Sümen, 1994)	14
Şekil 5	ERP Sisteminin Kronolojik Gelişimi (Altınkeser, 1994)	16
Şekil 6	Değişen rekabet unsurları (Altınkeser 1999)	17
Şekil 7	Kapsam Bakımında ERP'nin Gelişimi (Altınkeser, 1999)	19
Şekil 8	UDDI' da Servis Bulma Senaryosu	33
Şekil 9	Web Servislerinin çalışma mantığı	34
Şekil 10	Web Servisleri Yazılım Çevresi	45
Şekil 11	Proxy Nesnesini ve SOAP' ın Xml Web Servisi Mimarisindeki yeri.	48
Şekil 12	Sipariş Takip Programı İşlem Zinciri	50
Şekil 13	Sipariş Takip Programı Veri Tabanı Yapısı	52
Şekil 14	Sipariş Takip Programı Yönetici Ekranı	53
Şekil 15	Sipariş Takip Programı Şube Ekranı	54
Şekil 16	Sipariş Takip Programı Mesaj Alanı	55
Şekil 17	Sipariş Takip Programı Sisteme Giriş Ekranı	56
Şekil 18	Uygulama Fonksiyon Çağırım Zinciri	58
Şekil 19	Sipariş Takip Programı Web Servisi	59
Şekil 20	Sipariş Takip Programı Web Servisi Fonksiyonları Listesi	60
Şekil 21	Sipariş Takip Programı Web Servisi Sisteme Giriş Fonksiyonu Test Ekranı	60
Şekil 22	Sipariş Takip Programı Yeni Sipariş Giriş Ekranı	62
Şekil 23	Sipariş Takip Programı Yeni Sipariş Ürün Listesi	63
Şekil 24	Sipariş Takip Programı Yönetici Sipariş Görüntüleme Ekranı	64
Şekil 25	Sipariş Takip Programı Yönetici Sipariş Kabul Ekranı	65
Şekil 26	Sipariş Takip Programı Yönetici Onaylanacak Sipariş Seçim Ekranı	65
Şekil 27	Sipariş Takip Programı Yönetici Sipariş Onaylama/İptal Ekranı	66

1. GİRİŞ

Son yıllarda, firmalar bilgi ve iletişim teknolojisinde yaşanan gelişmelere paralel olarak yeni yönetim ve iş yapma yaklaşımları geliştirmişlerdir. Bu gelişmelerin vardığı son noktalardan birisi de kurumların varolan kaynaklarını daha verimli kullanmalarını amaçlayan kurumsal kaynak planlama yazılımlarıdır.

Uluslararası literatürde Enterprise Resource Planning, kısaca ERP, olarak adlandırılan ve bu tezin konusu olan sistemler, Türkçe kaynaklarda genelde *İşletme Kaynakları Planlaması*, ve *Kurumsal Kaynak Planlama* olarak iki farklı adla anılmaktadır. Kar amacı güden veya gütmeyen her türlü organizasyon için kapsamına bakılmaksızın bu tezde ERP kısaltması kullanılacaktır.

ERP, kurumların tedarikten, dağıtımına kadar tüm iş süreçlerini bütünlük bir veri-bilgi yönetim sistemi desteğiyle yönetmesini sağlayan geniş kapsamlı, modüler yapıya sahip olan yazılım paketleridir. Dünya çapında oldukça büyük bir pazar oluşturan bu yazılım paketinin, üreticisi konumunda olan irili ufaklı yüze yakın firma faaliyet göstermektedir.

Bu tezde, öncelikle ERP'nin işleyişi incelenmiş olup, web tabanlı sistemler ile entegrasyonu üzerinde durularak, örnek bir çalışma tanıtılmıştır.

2. KURUMSAL KAYNAK PLANLAMA (ERP)

Kurumsal Kaynak Planlaması'nın ne olduđu konusuna dair genel kavramlar bulunmasına karşın, tanımı üzerinde tartışmalar devam etmektedir. Yaygın olarak, bir şirkette süregelen tüm bilgi akışının entegrasyonunu sağlayan ticari yazılım paketleri tanımı kullanılmaktadır. (Davenport, 1998).

Kurumsal Kaynak Planlama kavramına 3 farklı şekilde bakmak mümkündür: (1) ERP, bilgisayar yazılımı şeklinde alınıp satılabilen ticari bir maldır, (2) ERP, bir kurumun tüm süreç ve verilerini tek bir geniş kapsamlı ve bütünlük yapı altında toplayan bir gelişim amacıdır, (3) İş süreçlerine çözümler sunan bir altyapının anahtar ögesidir (Klaus ve arkadaşları, 2000).

Bu sistemler adlandırılırken "*Kurumsal*" kelimesinin kullanılmasının sebebi, kapsamlarının belirli bir hizmet veya ürün üretmeye yönelik faaliyet gösteren kurumların tüm fonksiyonlarını içermesidir. ERP sistemleri bütünü bu bütünü oluşturan parçalardan daha büyük olduđu felsefesi üzerine kurulmuştur. Bu felsefeden yola çıkılarak meydana getirilen ERP sistemleri, kurumlarda daha önceleri ayrı ayrı ele alınan işlevleri birbirine bağı bir şekilde kurumun amaçlarını yerine getirmek için çalışan parçalar olarak ele alır ve bundan faydalanarak kurumlardaki her türlü kaynağın (İşçilik, Malzeme, Para, Makine) verimliliğini en üst düzeye ulaştırmayı amaçlar. Başka bir bakış açısıyla, ERP sistemleri şirketin ortak bir yerde saklanan verilerinden elde edilen bilgilerin doğru olarak, doğru makamlara iletilmesini sağlar. Kurumsal Kaynak Planlaması Sistemlerinde yer alan en temel fonksiyonlar içinde Üretim, Finans, Dağıtım, İnsan Kaynakları, Satış&Pazarlama, Envanter Yönetimi, Satın Alma, Kalite ve Proje Yönetimi sayılabilir. Bu işlevlerin açıklamaları ilerleyen bölümlerde verilecektir. Bu genel kurumsal işlevlerin yanında ERP sistemleri, hastanelerde hasta takip yönetimi, üniversitelerde öğrenci yönetimi ya da perakendecilikte yüksek hacimli ambar yönetimi gibi sektöre özel işlevleri de desteklemektedir.

ERP Sistemleri, daha özet bir tanımla, orta veya büyük ölçekli şirketler için çok karmaşık ve standart yönetim yazılım uygulamalarıdır. Bu yazılım paketleri bir birine entegre edilerek, daha geniş ve büyük sistemler kurulabilmektedir. Bu özellik,

ERP sistemlerin miras yolu ile veri paylaşabilme özelliğinden ileri gelir. Genellikle, ERP sistemleri tüm ayırık işlem süreçlerini bütünleştirerek, merkezi olarak yönetilmesini sağlar. Bununla beraber fonksiyonel ve hiyerarşik mantık olarak birbirine sıkıca bağlanmış, bütünleştirilmiş tüm yönetilebilir sistemlerin tekrar bir merkezden yönetilebilir hale getirilmesini sağlar (Baretta, Polo, 2002).

2.1 ERP Sistemlerinin Temel Özellikleri

ERP sistemleri için bilinmesi gereken önemli nokta, bu sistemlerin kurulumunun sınırlı ve birbirine uyum sağlaması için gereken maliyetin oldukça yüksek olması gerçeğidir. ERP sistemlerinin bir organizasyona kurulumu demek, üzerine düşünülmesi gereken ciddi bir kaynak, zaman ve organizasyon içi maliyetlerden oluşan yatırımdır. ERP yazılımı kurulumunda, organizasyon, sistemin kuruma entegre edilmesi durumunda, ERP sisteminin organizasyona yenilik katıp, katmayacağını yoksa bir engel teşkil edip etmeyeceği ve organizasyonun karar verme süreçleri üzerine olacak etkisini düşünmelidir. Eğer Simon tarafından geliştirilen ve Scott-Morton tarafından güncellenen karar verme süreci sınıflandırılması kullanılırsa, (Simon 1970), ERP sistemlerin varlığına doğrudan karar verilmiş veya koşullara bağlanmış, yapısal veya yarı yapısal önemli veri paylaşımlarını onaylamak mümkündür.

Bir ERP sisteminin kullanımının, organizasyonun karar verme süreçlerinde önemli görevleri vardır ve tipik olarak bir organizasyonla ilgili proje düzeyinde önemli noktalarda, özellikle tekrar gerektiren süreçlerde mühendislikte, devreye girer. Diğer taraftan, bunların hepsi, daha çok veya daha az standart prosedürlerin ve adapte edilen organizasyonun özel karakteristik özelliklerinin veya özel ihtiyaçlarıyla uğraşmak için birleştirilen modellerin adaptasyonunu sağlar. Diğer taraftan o, bir ERP sisteminde gömülen yeni modellerle organizasyonun uyumu çerçevesinde bir tekrar düzenleme ve süreçlerin yeniden yapılandırması, prosedürler ve rolleri gerektirir. Ortak adaptasyonun bu süreci, oldukça pahalıdır, ve sistemin adaptasyon ve uyumluluk evresinde konsantre olmayı gerektirir, bir kere başarılı şekilde uygulandığında uyum sağlamış form zamanla sabit hale gelir(Ravagnani, 2000) bu geleneksel bilgi sistemlerinin geliştirilmesinde olur (Truex, Baskerville ve Klein 1999; Bello, Sorrentino ve Virili 2002).

Bu araştırma karar verme süreçlerindeki ‘Donma’ın, organizasyon içerisindeki varlığının özellikle organizasyonun kurulumundaki yerini inceler. Asıl konu uygulama geliştirme ve konsolidasyon evresini oluşturan Web Servisleri olarak bilinen yeni ICT teknolojisinin kullanımının dezavantajlarından kaçınmanın olasılıklarını önemli noktalarını ve karakteristik özelliklerini incelemektir. Bu yeni teknoloji farklı sistemlerin entegrasyonunu ortak bir platformda daha kolay gerçekleştirmenizi sağlar. Bu, organizasyon yenileme sürecini, ERP sistemlerinin geleneksel çevreye uyum ihtiyaçlarının daha az olmasını sağlar.

ERP yazılımları farklı sektörlerin farklı ihtiyaçlarına uyum sağlayabilecek seviyede özelleştirilebilirler. Bu sebepten dolayı ERP yazılımları 3 farklı biçimde ortaya çıkmaktadır:

(1) Yazılımın en kapsamlı ve en genel halidir, pek çok sektörü hedef alır ve kullanılmadan önce yapılandırılmalıdır.

(2) Yazılımın kapsamlı halinden önceden yapılandırılmış şablonlar oluşturulur. Bu şablonlar sektöre ve firma büyüklüğüne göre özelleştirilir.

(3) Yazılım, birinci ve ikinci şekilde yüklendikten sonra firmanın kendi yapısına göre özelleştirilir.

Sektöre, firma büyüklüğüne ya da firmanın kendisine göre özelleştirilmiş ERP sistemlerinin genel özelliklerinden bahsetmek anlamlı olmayacağından ancak bu sistemlerin en kapsamlı ve genel hallerinin ortak özelliklerinden bahsedilebilir. Buradan hareketle, ERP sistemlerinin tanımlayıcı özellikleri hakkındaki genel kanılar aşağıdaki gibi özetlenmiştir (Klaus ve arkadaşları, 2000)

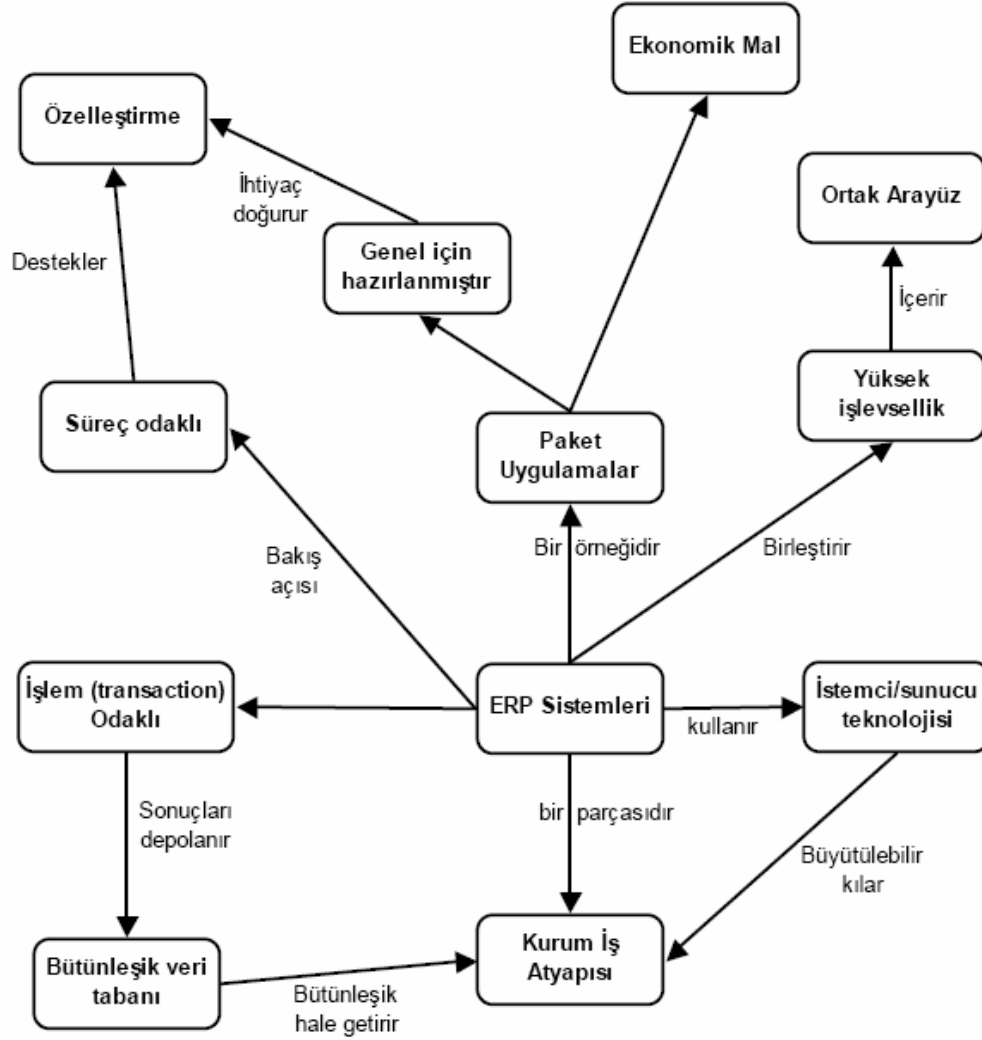
- Tüm sektörleri hedef alan ve kurulumu esnasında özelleştirilebilen standart yazılım paketidir.
- Diğer paketlere kıyasla özelleştirmeye çok daha müsait yapıya sahiptir. Çünkü, hedef sektörü tanımlanmamış olan bu standart paketler kurulum esnasında kurumun özel ihtiyaçlarına göre özelleştirilebilmelidirler.

- Bir veri tabanı yönetimi yazılımı, ara katman yazılımı (middleware) ya da bir işletim sisteminden ziyade ERP bir uygulama yazılımıdır.
- Hem ana verileri hem de iş süreçlerine ait verileri tutan bütünleşik Bir veri tabanıdır.
- Temel iş süreçleri hakkında çözüm önerileri sunar.
- Birçok kurumsal işlevi desteklemeyi hedeflemesinden dolayı yüksek oranda işlevsel bir yapıya sahiptir.
- ERP ürün paketleri dünya genelinde, ülkelerden ve bölgelerden bağımsız çözümler sunmak üzere tasarlanmıştır. ERP paketleri, ülkeden ülkeye farklılık gösteren muhasebe işlemleri, özel biçimli belgeler oluşturulması (teklifler, faturalar vs) ve insan kaynakları yönetimi gibi işlevleri ülkesel gereksinimlere uygun bir şekilde yerine getirirler.
- Temel ERP ürün paketi dünya ölçeğinde kullanımı sağlamaya yeterli işlevselliği içermesi sayesinde bazı sektörleri değil tüm sektörleri hedefler.
- ERP yazılımlarını diğerlerinden ayıran bir özellik de ERP paketlerinin tedarik yönetimi, sipariş yönetimi ve ödeme işlemleri gibi, tekrar eden ve sürekli olan iş süreçlerini destekliyor olmalarıdır. Bu paketler sadece pazarlama, ürün geliştirme ve proje yönetimi gibi düşük seviyede yapılandırılmış ve düzensiz olan işlevler üzerinde yoğunlaşmazlar.

ERP'nin temel teknik özellikleri ise şunlardır:

- Tüm uygulama alanlarında birbiriyle tutarlı grafik arayüzleri.
- Uygulama, veri tabanı ve sunum olmak üzere üç katmandan oluşan bir istemci-sunucu mimarisi.
- İşletim sistemi ve donanımdan bağımsızdır, ERP paketleri Solaris, Windows NT ya da Linux gibi farklı sistemler üzerine kurulabilir.
- Yönetimin karmaşık olması sadece ERP'nin özelliği olmamakla birlikte, bu sistemler kadar kritik öneme haiz sistem sayısı azdır.

ERP'nin sayılan bu ortak özellikleri Şekil 1'de verilen kavramsal grafikte görülebilir.



Şekil 1 - ERP Temel Özellikleri . Kavramsal Grafik (Hagman, 2000)

2.2 ERP'nin Gelişim Süreci

Kurumsal Kaynak Planlama sistemlerinin tarihsel gelişimini incelemek için iş entegrasyonu kavramlarının gelişim sürecine bakmak gerekmektedir. 1960'lı yıllarda bilgisayarların ticari işletmelerde yaygınlaşmaya başlaması ile ilk kurumsal üretim yönetim sistemi, malzeme ihtiyaç planlaması MRP (Manufacturing Requirement Planning) yazılımları kullanılmaya başlamıştır. İlk MRP yazılımı IBM tarafından geliştirilmiştir. İlk aşamada malzeme ağaçları ve stokları kapsayan sistem gittikçe gelişerek 1980'li yıllarda üretim işletmelerinin üretim ile doğrudan ilintili faaliyetleri (satın alma, üretim planlama ve kontrol, muhasebe, stok yönetimi....) kapsar hale

gelmiş ve üretim kaynakları planlaması: MRP II (Manufacturing Resource Planning) adını almıştır. MRP II o zamanlar, etkin imalat planlama için bir sonraki adım olarak görülmekle birlikte; firmalar, karlılık ve müşteri memnuniyeti gibi amaçların sadece üretim değil tüm işletmeyi ilgilendiren kavramlar olduğunu anlamakta gecikmemiş ve finans, satış, dağıtım ve insan kaynakları işlevlerinin de dahil olduğu sistemlere ihtiyaç duyulmaya başlanmıştır. Bu süreçte, ürün geliştirme safhasının teknik işlevleri ile üretim sürecini bütünleştiren Bilgisayar Bütünleşik İmalat (CIM - Computer Integrated Manufacturing) sistemleri ile firmaların ürün dağıtım kanallarını ve ürün dağıtımlarını planlamalarını ve yönetmelerini sağlayan dağıtım kaynakları planlama (DRP - Distribution Resource Planning) sistemleri ortaya çıkmıştır. 1990'lı yıllarda yönetim sistemleri yalnız üretim sektörünü değil tüm sektörleri (telekomünikasyon, perakende, medya, sağlık, kamu...) tüm faaliyet birimlerini (satış sonrası servis, bakım onarım, insan kaynakları, duran varlık yönetimi. . .) kapsar hale geldi ve kurumsal kaynak planlaması ERP (Enterprise Resource Planning) adını aldı. 2000'li yılların başında özellikle internet ve çağrı merkezleri kanallarını kullanarak işletme dışı unsurlarla da bütünleşen ERP sistemleri, müşteri ilişkileri yönetimi (CRM), tedarik zinciri yönetimi (SCM) ve işletme zekası (BI) kavramlarını da kapsayarak ERP II konseptine genişlediler.

İlk aşamada yalnızca malzeme tedariki fonksiyonuna sahip olan yazılımlar günümüzde işletmelerin tüm bölümlerinin birbiri ile entegrasyon içerisinde çalışmasını hedefler olmuştur. Kapsam bu kadar genişleyince ERP sistemleri artık yazılım olarak tanımlanmamaktadır. Başarılı ERP kurulumlarında işletmeler tüm iş süreçlerini sistem üzerinde yeniden yapılandırmaktadırlar. Özetle ERP sistemleri bilgisayara yüklendikten sonra çalışmaya başlayan ve işletmelerin tüm problemlerini kendi kendine çözecek yazılımlar olarak düşünülmemelidir.

2.2.1 Malzeme İhtiyaç Planlaması (MRP)

Sınai işletmelerde hammadde gereksinimlerinin en uygun biçimde sağlanmasında bugüne kadar iki temel yaklaşım görülmüştür. Bunlardan ilki geleneksel diyebileceğimiz, *İstatistik Envanter Kontrolü* yöntemidir. Bu yöntemde, talep tahminleri sonrasında karşılaşılan minimum stok düzeyi, sipariş sayısı, sipariş miktarı gibi değerler yer almaktadır ve stoklardaki değişimlere göre davranılmaya

çalışılarak hammadde gereksinimleri karşılanır. İkinci yöntem ise, Malzeme İhtiyaç Planlaması (Material Requirement Planning, MRP) olarak anılan ve önceki yönteme kıyasla daha düşük stok düzeyleri ile daha etkin stok yönetimi, sağlayan çağdaş bir yaklaşımdır.

MRP ilk olarak 1960'ların başlarında ABD. de malzeme tedarikinde ve üretiminde bilgisayara dayalı bir yaklaşım olarak ortaya çıktı. Bu tekniği tanımlayıcı kitap Orlicky tarafından 1975'te yayınlandı. Bu tekniğin ikinci dünya savaşı sonrasında Avrupa'da birkaç yerde bilgisayar olmaksızın kullanıldığı yönünde kayıtlar bulunmaktadır. Ancak Orlicky bu tekniğin bilgisayar kullanımıyla imalat stoklarını yönetmede daha detaylı uygulamaların yapılmasını sağladığını fark etti.

MRP faaliyetlerinde kullanılan ilk bilgisayar yazılımları sadece hesap yükünü hafifleten uygulamalar şekline geliştirilmiştir. Bu uygulamalar, bugünkü modern MRP II ve ERP sistemlerinin temelini oluşturan ve ana üretim çizelgesine dayanarak yapılan en basit sipariş planlamalarıdır.

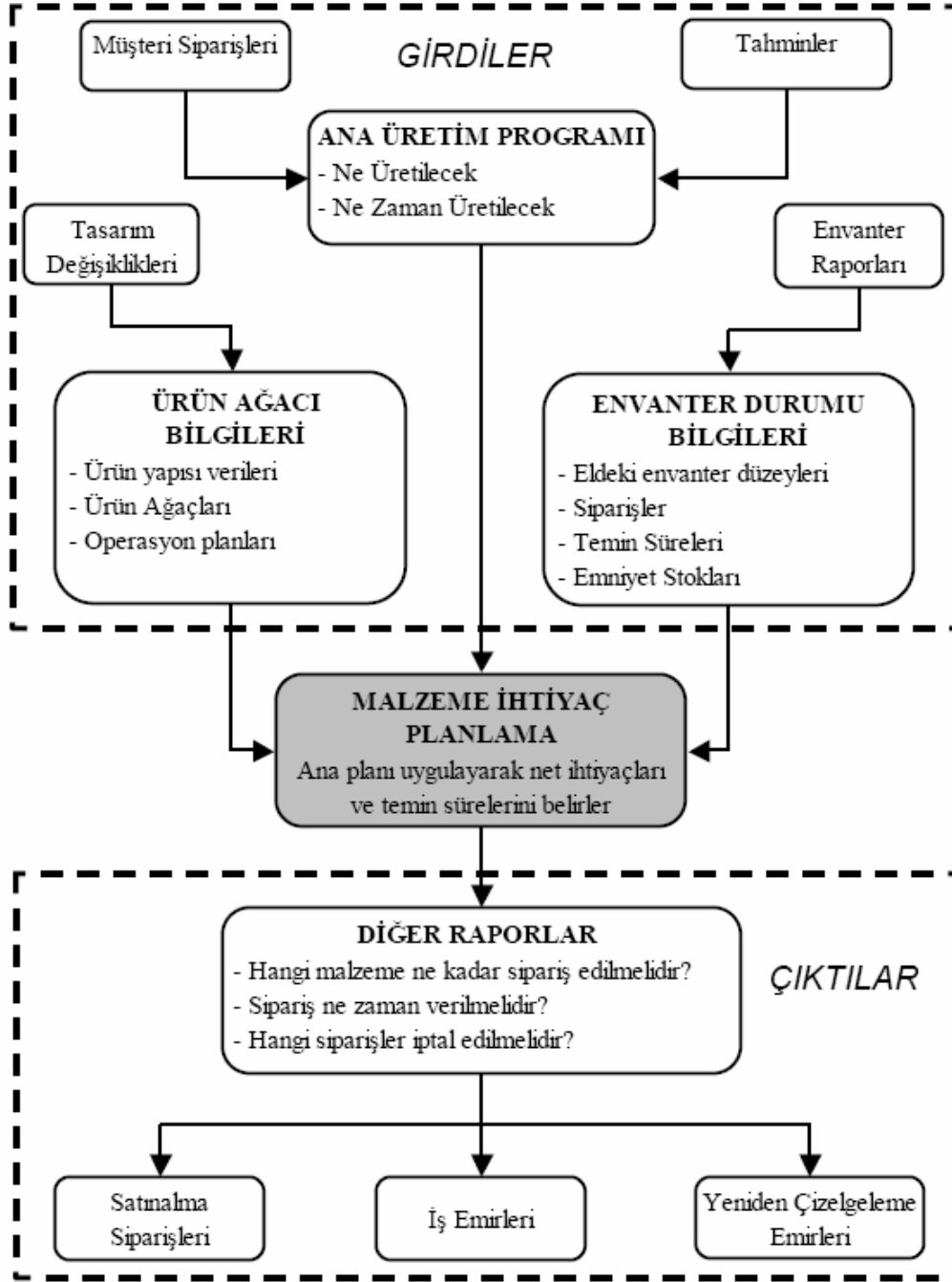
MRP'nin popülaritesi 1970'lerin başlarında Amerikan Üretim ve Stok Kontrol Topluluğu (APICS)'nın bu yöndeki teşvik edici çalışmalarıyla artmıştır. APICS, insanları MRP'nin tüm üretim prosesinin yönetiminde entegre iletişim ve karar destek sistemi olarak çözüm olduğu konusunda ikna etmeye çalışmıştır. Tekniğin optimize edilmesi için sistem analizinin ve yönetim biliminin gerekliliği üzerinde durulmuştur. En önemli sorunlar olarak disiplin, eğitim, anlayış ve iletişim olarak gösterildi. Bu teşvik sonraları bilgisayar endüstri tarafından sürdürülmüştür.

MRP basitçe, son ürün için hazırlanan ana üretim çizelgesini ürün ağacı bilgisi yardımıyla gerekli parça ve malzeme çizelgesine çevirerek satın alma ve imalat emirleri hazırlayan bir envanter yönetim tekniği olarak tanımlanabilir. MRP sistemi ana üretim çizelgesinden hangi son ürünlerin ne zaman ve hangi miktarlarda üretilmesi gerektiğini öğrenir; ürün ağacı bilgilerinden yararlanarak bir son ürün için gerekli olan parçaları ve miktarları hesaplar. Bu bilgileri envanter durumu ile karşılaştırır, üretim ve temin sürelerini de kullanarak parçaların ne zaman ve ne kadar sipariş edileceğini belirler (Şekil 2).

MRP'den önce kullanılan lojistik teknik: ikmal yapmak, kullanılanın yerini doldurmak ilkesi üzerine idi. MRP ileriye bakarak, ileride sadece nelerin gerekeceği tahmin etti. Bu deęişikle . itme'den, çekme'ye dönen yaklaşımla daha yüksek seviyede hizmet sağlanırken, envanter seviyeleri düşmüştür. Bu yaklaşım sadece etkin bir işletimle yararlıdır ve talep tahmini, müşteri siparişlerini değerlendirme gibi destekleme fonksiyonlarına baęlı olarak bir dereceye kadar etkindir.

2.2.2 Kapalı Çevrim MRP

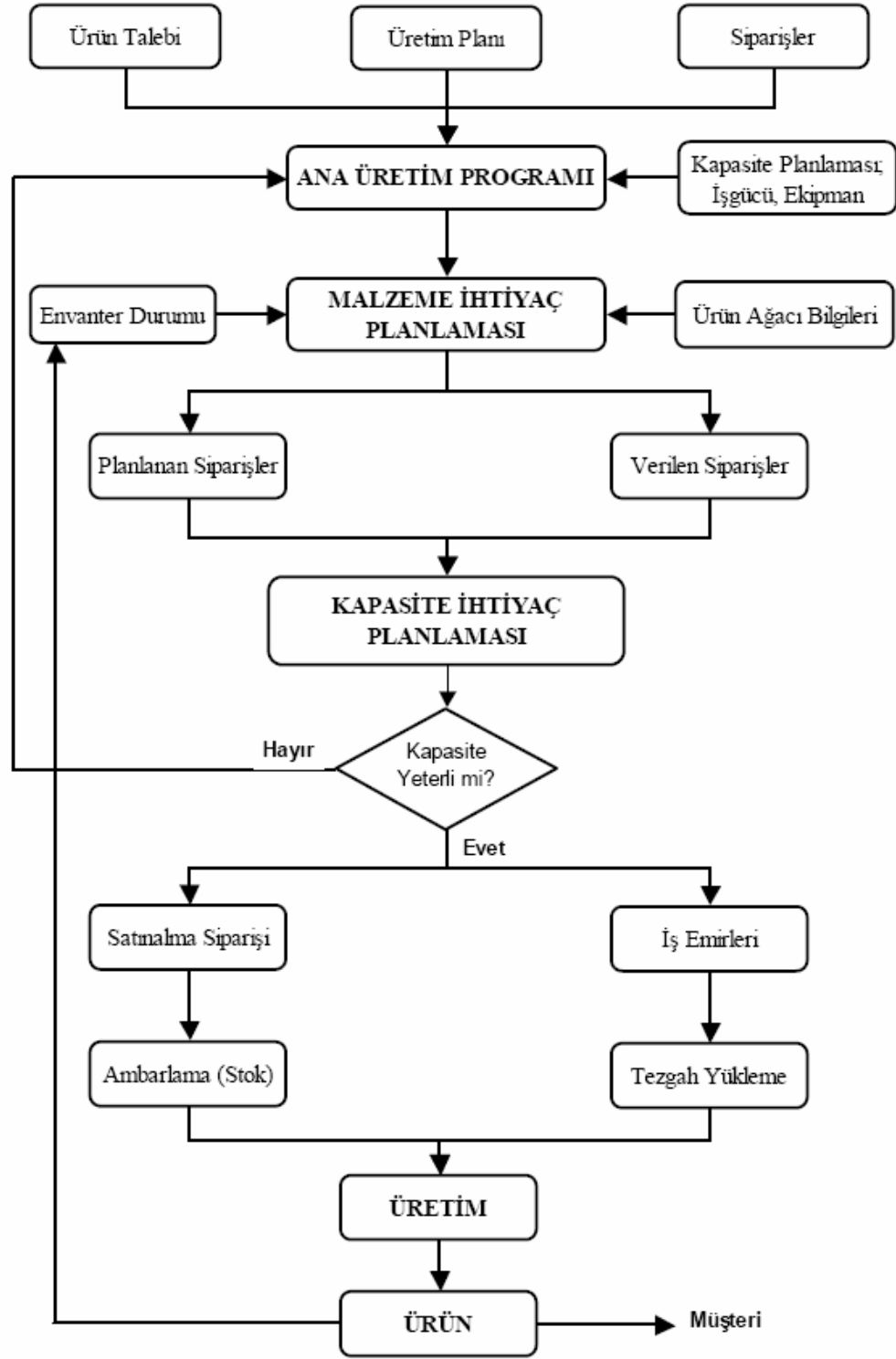
Kapalı çevrim (Closed Loop) MRP, malzeme ihtiyaç planlamasının ana üretim çizelgesinde hedeflenen üretim miktarları ile işletmenin imalat kapasitesi arasındaki ilişkiyi kontrol etmemesi gibi bir sakıncalı özellięi nedeniyle geliştirilmiş bir sistemdir. Bu sistemde, MRP kapasite ile karşılaştırılmakta ve mevcut kullanılabilir kapasitenin yeterli olmadığı durumlarda ana üretim çizelgesi bir geri besleme ile uyarılmaktadır. Şekil 3'te kapalı çevrim MRP sisteminin algoritması gösterilmiştir. Bu sistemle MRP sadece siparişleri planlayan malzeme yönetim aracı olmaktan çıkarak üretim kontrolüne katkıda bulunmaya başlamış olmaktadır.



Şekil 2 - MRP Sistemi (Acar, 1991)

2.2.3 Kapasite İhtiyaç Planlaması (CRP)

Kapasite İhtiyaç Planlaması (CRP . Capacity Requirement Planning) MRP'nin Çıktılarını, kapasite kısıtları ile karşılaştırır ve Ana Üretim Çizelgesinin yapılabilirliğini kontrol eder. APICS tarafından yapılan tanımı şudur: Kapasite seviyelerini veya sınırlarını belirleme, ölçme ve ayarlama fonksiyonudur, ayrıca üretim gereklerini yerine getirebilmek için gereken makine ve işgücü miktarını belirler. MRP'deki açık atölye emirleri ve planlanmış siparişler bu siparişleri zaman periyodunda iş saatleri olarak iş merkezlerine yükleyen CRP için birer girdidir. CRP kısa veya orta dönemde MRP ile üretilen malzeme planını gerçekleştirmek için gerekli olan spesifik işgücü ve teçhizat kaynaklarını miktarsal olarak belirler. Daha sonra gerekli kapasite, potansiyel aşırı veya az yüklemeleri belirlemek için mevcut kapasite ile karşılaştırır. (Yetiş, 1993)



Şekil 3 - Kapalı Çevrim MRP Sistemi (Durmuşoğlu, 1993)

2.2.4 Üretim Kaynakları Planlaması

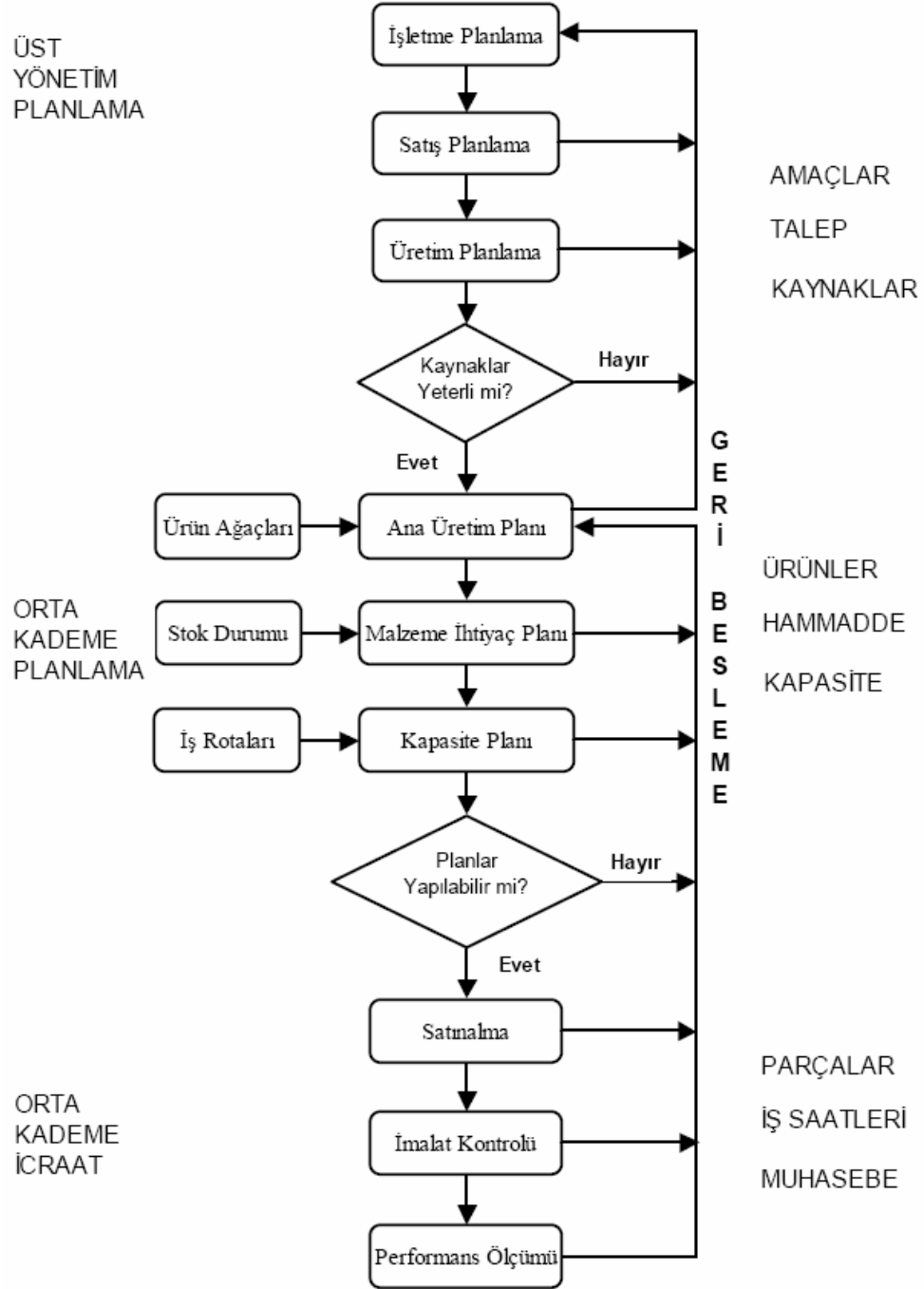
Üretim Kaynakları Planlaması (MRP II - Manufacturing Resource Planning) üretim işletmelerindeki tüm veri çalışmalarının entegrasyonu ile ilgilidir. Bu entegrasyonla, işletme içindeki tüm malzeme hareketleri sürekli ve düzenli olarak bilgisayar kaydına alınarak bütün işletme birimlerince ortaklaşa kullanılan bir veri tabanında saklanmaktadır. Böylece işletme yönetimine, geleceği daha etkili planlayabilme ve alınan kararların sonuçlarını süratle irdeleyebilme gücü kazandırılmış olur.

MRP II, bir firma işletim sistemi ve bazen de işletmenin bilgisayar modeli olarak adlandırılmaktadır. Başka bir deyişle, MRP II, gerçek imalat işletmesini, her faaliyetin etkisini test etmek için benzetebilen (simulate) standart, mantıklı bir sistemdir. Üst yönetime, alternatifler arasında daha sağlam karar vermeyi sağlayan bir yoldur.

İşletmeler malzeme kaynağının yanı sıra işgücü, makine ve para kaynaklarını da en etkin bir şekilde planlamak ve kontrol etmek zorundadır. Üretim Kaynakları Planlaması MRP sistematiğine bağlı olarak söz konusu kaynakların da eşgüdümlü olarak planlanması ve kontrolünü gerçekleştiren bir yaklaşımdır. Esas itibariyle MRP II malzeme ihtiyaç planlamasının yanı sıra, makine ve işçilik kaynağına yönelik olarak da kapasite planlaması çalışmalarını içerir. (Tanyaş, 1994)

MRP II kavramının entegrasyondan sonra gelen anahtar özelliği geri besleme olgusudur. Planlama ve üretimin her aşamasında ortaya çıkabilecek sorunlar ya da yeni oluşumlar karşısında daha önceki seviyelere geri dönerek sistemi yeni şartlara uydurabilme olanağı her zaman vardır.

MRP II, bir imalat işletmesinin tüm kaynaklarının etkin olarak planlanması yöntemidir. Sonuç olarak, MRP II entegrasyon ve geri besleme faktörlerini bilgisayar teknolojisi yardımıyla etkin bir şekilde kullanarak işletmedeki planlama, üretim, finansman sürecini modelleyen ve verim artışını hedefleyen bir araçtır. MRP II sisteminin genel işleyiş şeması Şekil 4'te verilmiştir.



Şekil 4 - MRP II Sistemi (Sümen, 1994)

2.2.5 Dağıtım Kaynakları Planlaması

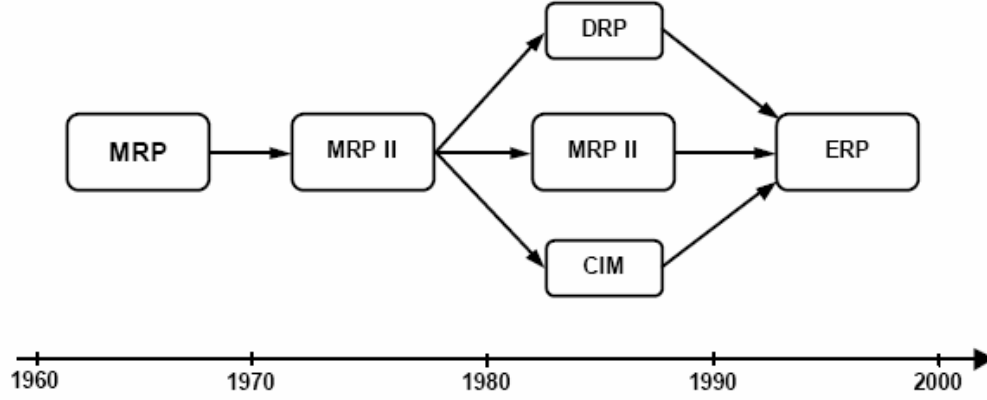
Dağıtım Kaynakları Planlaması (DRP - Distribution Resource Planning) MRP'den esinlenerek envanterin dağıtımında optimizasyon sağlamaya çalışan bir yöntemdir. Literatürde ilk kez 1975 yılında Kanada'da bulunan Abbott laboratuvarlarında kullanılmıştır (Greene, 1987).

Dağıtım Kaynak Planlaması, sistemi şu kriterleri dikkate alarak çalışır; taşıma araçları ve teçhizatları, yükleme/indirme alanı, depolama alanı ve hacmi, ürünlerin birbirine göre taşıma ve depolama özellikleri, taşımadaki tonaj ve zaman kısıtları. DRP, ihtiyaçlar oluştuğunda ilk planlamayı yapar ve bununla yetinmeyerek her değişiklik için de planları yeniler. Dağıtım kaynakları planlamasında bir merkezi depo ve ona bağlı dağıtım depoları söz konusudur. Talep gerek ara depolara gerekse merkezi depolara olabilir. Merkezi depo hem tali depolardan gelen hem de doğrudan kendisine gelen talepleri karşılamak zorundadır. Bunları karşılayabilmek için daha fazla miktarda emniyet stoku bulundurur.

DRP, periyotlar boyunca dağıtım depolarının gereksinimlerinin projeksiyonunu yapar ve ana depodan planlanmış siparişler oluşturur. DRP; üretim kapasitesinin ve stokların etkin bir şekilde tahsis edilmesini sağlamak, müşteri servis düzeyini yükseltmek ve stok yatırımlarını düşürmek için, üretim ve dağıtım yöneticileri tarafından ihtiyaç duyulan bilgi akışını sağlar.

2.2.6 ERP'nin Ortaya Çıkışı

MRP ve MRP II sistemlerinin devamı olarak bünyesine CIM ve DRP sistemlerini de katarak gelişen ve tüm işletme kaynaklarının modüler yapıdan oluşan tek bir bütünleşik sistemle planlanıp yönetilmesini amaçlayan ERP sistemi fikri 1990'lı yılların hemen başında ortaya çıkmıştır. Şekil 5'te ERP'nin ortaya çıkışının kronolojik bir özeti verilmiştir.



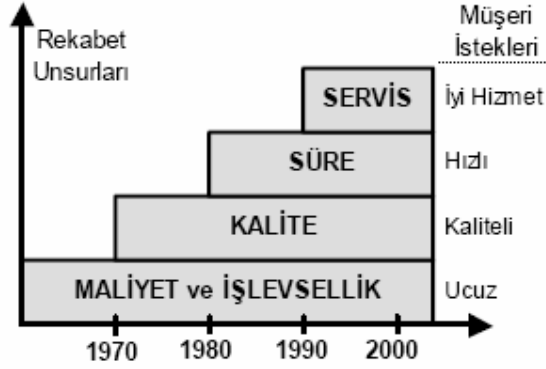
Şekil 5 - ERP Sisteminin Kronolojik Gelişimi (Altınkeser, 1994)

1960'lı yıllarda bilgisayarların imalat yönetiminde kullanılmaya başlamasıyla MRP sistemleri popüler olmaya başlamış, ardından 1970'li yıllarda kapasite planlama, satış gibi işlevleri de içeren MRP II sistemleri hızla yayılmaya başlamıştır. 1980'li yıllarda ise bilgisayarların ürün tasarımı ve imalatı alanında önemli gelişme kaydetmesi ile birlikte CIM devreye girmiştir. Aynı zaman aralığında, birden fazla dağıtım kanalına sahip büyük işletmelerin, ürün dağıtım kanallarını ve dağıtımın kendisini en iyi şekilde yönetmelerini sağlamak için gene bilgisayarların kullanıldığı DRP geliştirilmiş ve MRP II, CIM ve DRP'nin birbirinden bağımsız olarak kullanıldığı melez sistemler ortaya çıkmıştır. Bu sistemlerin birbirleriyle bütünleşik bir şekilde uyumlu çalışmasını sağlamak ihtiyacı ile insan kaynakları, kalite yönetimi gibi yeni işlevlere olan ihtiyaç doğrultusunda 1990'lı yılların başından itibaren tüm bu işlevleri modüler fakat aynı zamanda bütünleşik bir sistem altında toplayan ERP yazılım paketleri görülmeye başlanmıştır.

2.3 ERP Sistemlerinin Kullanım Nedenleri

Organizasyonlar bugün hayati önemi olan iki unsurla karşı karşıyadır. Küreselleşme ve kısalmış Ürün Pazar Ömrü. Küreselleşme, rekabeti şimdiye kadar görülmemiş boyutlara çıkarmış durumdadır. Rekabetin değişen unsurları Şekil 6'da görülmektedir. Hayatta kalabilmek ve gelişebilmek için işletmeler zaman içinde ortaya çıkan yeni rekabet unsurlarına uyum sağlamak zorundadırlar. Böyle bir

rekabet ortamında şirketler başarılı olmak için endüstrideki en iyi uygulamaları takip etmek zorundadır.



Şekil 6 - Değişen rekabet unsurları (Altınkeser 1999)

Kısalmış ürün pazar ömrü, sürekli geliştirme, ürün esnekliği, süper etkin lojistik kontrol ve daha iyi tedarik zinciri yönetimi gerektirir. Bütün bunlar organizasyon içi ve dışı tüm tedarik zincirinde bilgilerin daha hızlı ve hassas girilmesine bağlıdır.

Finans, pazarlama, üretim, insan kaynakları gibi organizasyonel bölümler esnekliklerini kaybetmeden daha yüksek seviyede entegrasyon ile çalışmaya ihtiyaç duyarlar. Organizasyon çapında bir ERP sistemi ile bu ihtiyaçlar karşılanabilir. Bilgisayar ve iletişim teknolojisindeki büyük ilerlemeler organizasyonun birimlerini aralarında daha sıkı bir entegrasyon oluşturacak şekilde güvenli iletişim ağları ile birbirine bağlanabilir hale getirmiştir. Bilgi sistemleri teknolojisi günümüzde makul fiyata yüksek güvenilirlikte bol miktarda veri girişi mümkün kılmaktadır.

Açık sistem (Open System), İstemci/Sunucu Mimarisi (Client/Server Architecture), yüksek performanslı işletim sistemleri, hızlı uygulama geliştirme araçları organizasyon bütünlüğünde böyle bir sistemin çalışmasını sağlamaktadır. Bilgi teknolojilerindeki bu gelişmeler çağdaş bir sistem olan Kurumsal Kaynak Planlama Sistemlerinin gündeme gelmesini sağlamıştır.

Daha önce de ifade edildiği gibi ERP sistemleri, MRP ve MRP II sistemlerinin evriminden ortaya çıkmıştır. MRP sistemleri tek bir görev olan malzeme gereksinim planlaması işini yapmışlardır. MRP II tüm imalat fonksiyonlarını kapsayacak şekilde

genişletilmiştir. İmalat sanayi bilgisayar kullanımı için iyi bir ortamdır. Mühendislerden oluşan ilk imalat komisyonlarında bilgisayar korkusu yoktu. İkinci geniş kullanım alanı olarak CAD ve CAM bilgisayarların çok iyi kullanım alanı bulduğu imalat fonksiyonları olmuşlardır. Aslında imalat mühendisleri önemli ölçüde grafik, bilgisayarlı geometri, bilimsel görüntüleme gibi teorik bilgisayar bilimleri ile ilgilenmişlerdir.

GM, Ford, IBM, HP&Digital gibi büyük şirketler 1980'lere kadar kendilerini daha ziyade imalat şirketleri olarak gösterdiler. Doğal olarak PICS, MAMAN gibi komplike MRP sistemleri en son kurumsal bilgi sistemi olarak düşünülmüştü. Böyle karmaşık imalat çözümlerini başarmak için yazılım ve donanım yatırımları endüstride bu sistemlere görülmemiş bir ayrıcalık tanındığı açıkça ortaya koymuştur. Bu sistemlere göre muhasebe finansman ve personel bilgi sistemleri organizasyon için daha az önemlidir.

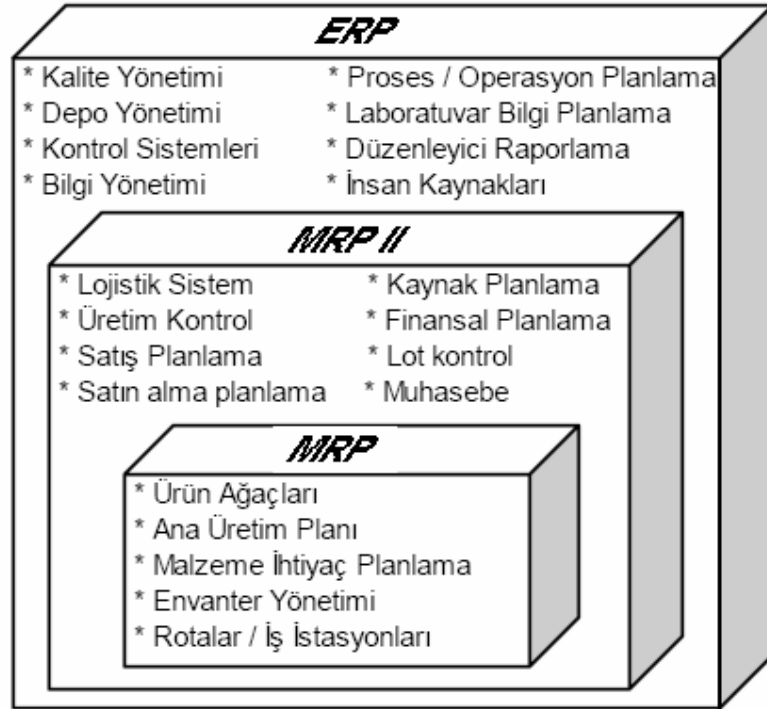
İşletmelerin küreselleşmesi ve bilgisayar ağlarının hızla yayılmasıyla imalat organizasyonlarının bilgi sistemlerini tedarik zincirleri boyunca genişletmeleri dikkat edilecek bir gelişme oldu. Kıtalara yayılmış karmaşık yazılım ve donanım kombinasyonlarıyla tedarikçi bilgi sistemleri entegre edilebilmelidir. Aynı şekilde satıcı-dağıtıcı ağı da imalat bilgi sistemi ile entegre olmalıdır. Ürünlerin Pazar ömürlerinin çok kısalmış olması pazarı kontrol eden ve hızlı yanıt veren imalat sistemlerini mecburi hale getirmiştir. Bu, imalat bilgi sistemlerini pazarlama bilgi sistemleri ile daha sıkı bir entegrasyona zorlamıştır. Esnek imalat sistemleri özelleştirilmiş kitlesel imalata dönüşmek zorunda kalmıştır ki, bu da daha ileri bir bilgi sistemleri entegrasyonu gerektirmektedir.

Çin ve Hindistan gibi büyük Asya devletlerini de içeren dünya ekonomilerinin açılmaları, Avrupa Topluluğu, NAFTA gibi konsolide Pazar ve ticari blokların ortaya çıkışı muhasebe ve finans fonksiyonlarının imalat fonksiyonları ile daha iyi bir entegrasyonunu gerekli kılan gereksinimler zinciri oluşturmuştur. Üretmek ve satmak yetersiz kalmış, organizasyonların finans sistemlerini karmaşık ticaret sınırları, bariyer ve kotalara göre düzenlemek durumunda kalmışlardır. Bilançolar çok döviz kurlu, çok ithalat-ihracat yasalı ve yönetmelikli, çok muhasebe kodlu,

uygulamalı ve dönemli sistemlere uymalıdır. Bu durum muhasebe ve finansman bilgi sistemlerini imalat sistemleri daha ileri bir entegrasyonu gerektirmiştir.

Bütün dünyada ve özellikle Asya ülkelerinde oluşan geniş iş imkanları sayesinde kontrat ve ihracat amaçlı imalat uygulanabilir olmuştur. Bu durumda aniden ortaya imalat fonksiyonunun ötesinde bağımlı ve bağımsız lojistik, malzeme yönetimi, proje yönetimi, finans, satışlar ve personel yönetimini içeren bir Kurumsal Bilgi Sistemi ihtiyacı ortaya çıkmıştır. Münferit bilgi sistem modüllerini entegre etmek neredeyse imkansızdır. Gerekli olan kurumsal gereksinimleri tasarım safhasında dikkate alan bir sistemdir. Kurumsal Kaynak Planlama sistemleri bu değişim senaryosunun doğal bir sonucudur.

Yukarıda anlatıldığı şekilde ERP sistemlerinin kapsam bakımından gelişimi Şekil 7’de gösterilmiştir.



Şekil 7 - Kapsam Bakımında ERP'nin Gelişimi (Altınkeser, 1999)

ERP Sistemlerinin ortaya çıkış nedenleri aşağıda özetlenmiştir

- Küreselleşme ve uluslar arası rekabet
- Bilgi teknolojisinin sağladığı yeni olanaklar
- Uluslar arası dağıtım zincirlerinin yaygın ve daha etkin kullanılabilir hale gelmesi
- Çok tesisli organizasyonların iyi idare ve kontrol edilmesi ihtiyacı
- Ürün ve üretim politikalarındaki rekabete bağlı değişimler

2.4 Kurumları ERP Kurmaya Götüren Sebepler

ERP (Kurumsal Kaynak Planlama) bir çok işletmenin ilgisini çekmektedir. ERP Kurulumu öncesinde, organizasyonun ERP yatırımına ihtiyacının gerçekliği, seçilebilecek yazılımların uygunluğu iyi test edilmelidir. Kararın mali boyutu ve stratejik boyutu yanı sıra çözüm sunan firma çokluğu konuyu karmaşıklaştırmaktadır.

Kurumsal çözüm sunan, ERP yazılımları; işletmelerin tüm birimlerini (insan kaynakları, üretim, planlama, satış sonrası servis, satış/dağıtım, malzeme yönetimi, muhasebe, finans, proje yönetimi, kalite güvencesi, yatırım yönetimi.) ve tüm faaliyet kollarını (telekomünikasyon, ilaç, kimya, paketli tüketim ürünleri, gıda, içecek, çimento, seramik, petrol ürünleri, gaz, metal, kağıt, elektronik, telekomünikasyon, savunma sanayi, beyaz eşya, otomotiv yan sanayi, inşaat, savunma organizasyonları, hastane, sağlık, havacılık tekstil, kablo, mağazacılık, medya) yönetilebilmektedir. Bu kapsamda tüm işletmelerin iş süreçleri ERP yazılımları ile yönetilebilirlik gerçeği ortaya çıkmaktadır. Denge ihtiyaçlar ve maliyetler ile kurulacaktır.

ERP yatırımı maliyetleri aşağıdaki başlıklarda toplanabilir: Yazılım lisansı (1 birim), yazılım yıllık bakım gideri (0.15 birim), implementasyon için danışmanlık giderleri (1-2 birim), implementasyon süresince çalışacak firma içi kalifiye kaynaklar (1 birim), son kullanıcı eğitimleri (0.10 birim), yazılım hayata geçtikten sonra firma içi verilecek destek faaliyeti ve yapılacak iyileştirmeler (0-10 birim).

Maliyetler satıcı firma, kullanıcı sayısı ve kurulacak yapının kompleksliği ile doğru orantılıdır.

ERP yatırımında firmanın ölçeği en önemli unsurdur. Genelde eşzamanlı kullanıcı sayısı ölçeklemede kriter olarak kabul edilmiştir. Küçük ölçekli şirketler: 0-20 kullanıcı, orta ölçekli şirketler: 20- 200 kullanıcı, büyük ölçekli şirketler: 200 ve üstü eşzamanlı kullanıcıya sahip olduğu düşünülür. ERP yatırımları orta ve büyük ölçekli şirketler için uygun görülmektedir.

Yukarıda ERP sisteminin kurulması kararında dengenin maliyet ile kurulabileceği belirtilmişti. Özetle terazinin bir ucunda ihtiyaçların karşılanamamasının maliyeti ve diğer bir ucunda ise ERP yatırımının maliyeti bulunmaktadır. ERP yatırımın maliyeti yaklaşık olarak çıkartılabilmektedir. İhtiyaçların karşılanamamasının maliyetinin hesaplanması için genel-geçer kurallı bir yöntem bulunmamaktadır.

3. ELEKTRONİK TİCARET VE WEB SERVİSLERİ

Günümüzde her şeyin elektronikleştiği bir ortamda ticaretin elektronik ortama taşınmaması kabul edilemezdi. Bu bölümde Elektronik Ticaret kavramı ve bu kavramın kullandığı sistemlerden web servisleri ile olan ilişkisi anlatılacaktır.

3.1 E-Business (E-İş)

E-İş için verilebilecek tek bir tanım yoktur. Bazı görüşlere göre, e-ticaret (E-Commerce), her türlü malın ve servisin bilgisayar teknolojisi, elektronik iletişim kanalları ve ilgili teknolojiler (akıllı kart-smart card-, elektronik fon transferi -EFT-, POS terminalleri, faks gibi) kullanarak satılması ve satın alınmasını kapsayan bir kavramdır. Başka bir görüşe göre ise e-ticaret, ödeme işleminin internet üzerinden yapıldığı alış-satışları içermektedir.

E-İş yoluyla oluşan ekonomi de, dijital ekonomi, elektronik ekonomi (e-ekonomi) olarak tanımlanmaktadır.

Elektronik iletişim teknolojileri ticari hayatta aslında uzun yıllardır (1980'lerden beri) kullanılmaktadır. Ama, internetin e-ticaret için kullanılması çok yenidir (1997'lerden beri). Zaten internet asıl gelişimini (patlamasını) ticari kullanımı artmaya başladıktan sonra yaşamaya başlamıştır. Sonuçta, internet'in yaygınlaşması ile birlikte, web ve e-posta uygulamalarının e-ticaretin doğal mekanı haline geldiğini söyleyebiliriz.

Ülkemizde de, 1998'den sonra, bazı büyük alışveriş merkezleri internet üzerinde satış mağazaları açmışlar, ayrıca kurumlara ve bireysel girişimcilere elektronik dükkan (e-dükkan) kiralayan servis sağlayıcılar ortaya çıkmaya başlamıştır. Basın ve bankacılık alanlarında, konularında öncü niteliği olan bazı kuruluşlar, e-iş alanında da yatırımlarını (2000'lerin başlarında) hızlandırmışlardır.

E-iş, müşteri, tedarikçi ve iş ortakları ile internet üzerinden bağlanarak çalışmaktadır. Bu çalışma biçiminin amacı ise mevcut iş yapma yöntemlerini değiştirerek daha hızlı ve verimli bir ortam yaratmaktır. Bu hedefe ulaşabilmek için şirketler, kendi bilgisayar sistemlerinde bulunan bilgileri paylaşımına açmalı, elektronik ortamda evrak

alış-verişi yapabilmeli, işlerini yürütmek için gerekiyorsa web sitelerinden satış veya internet üzerinden satın alma işlemlerini gerçekleştirmelidir. Kısacası E-Business ortamına geçmek isteyenler daha açık ve şeffaf olabilmeyi göze almalı, bilgisayar sistemlerini bu hedefi destekleyecek noktaya taşınmalıdır.

3.2 E-Commerce

Eğer e-business terimini bir adım öteye taşırsak karşımıza e-commerce terimi çıkar. E-Commerce iki tarafın her türlü bilgi paylaşımını da içerir. Örneğin; üretim yapan bir firmanın müşterisi olarak size ait siparişleri, bunların üretim hattı içindeki durumunu veya o ana kadar yapılmış kalite test sonuçlarını görmek isteyebilirsiniz. Aynı firmanın web üzerinden satış yapan bir bölümü var ise bu kısımdan da bazı malzemeleri satın alıyor olabilirsiniz.

İş Ticaret tarafına getirdiğiniz de bu seferde ticaretin şeklini seçmeniz gerekebilir. Bunlar ;

B2B : Business To Business

B2C : Business To Consumer

Bu iki terimin arasındaki fark, müşterinin kim olduğudur. Genel olarak müşteriler şirket olduğunda *B2B*, müşteriler şahıs olduğunda ise *B2C* terimi kullanılır. Bazen bu iki terim iç içe girerek yeni bir kapsam yaratırlar.

Örneğin, bireylere kitap satan bir internet sitesi *B2C* tarzında, sanayi şirketlerine kimyasal malzeme satan bir şirket ise *B2B* tarzında çalışmalıdır.

Müşterinin kim olduğu görüldüğü kadar basit bir ayrıntı değildir. Çünkü müşteri bir başka şirket olduğunda ortada çözülmesi gereken 2 konu vardır :

	B2B	B2C
Pazarlık	Ürün spesifikasyonu, teslim zamanı, fiyat ve ödeme koşulları her defasında görüşmeye açıktır.	Fiyatlar ve ürünler yayınlanmıştır, teslim koşulları ve fiyat üzerinde müşterinin herhangi bir pazarlık şansı yoktur
Entegrasyon	Müşterinin sistemi ile konuşabilme imkanı geliştirilmeli, iki sistem arasında bilgi paylaşımı mümkün olmalıdır.	Müşterinin web sitesi veya bilgisayar sistemi ile herhangi bir entegrasyona ihtiyaç yoktur.

Tablo 1 - B2B ve B2C Karşılaştırması

3.3 Web Servisleri

Günümüzde Internet sadece statik bilgi sunan web sayfaları olmaktan çıkıp, programlanabilir bir ağ halini almıştır. Bu yüzden uygulama, cihaz ve firmalar ortak bir iletişime gitmek zorunda kalmıştır. Bu programlanabilir zeki altyapıya özetle web servisleri diyebiliriz. Internete ulaşmak için kullanılan araçların artması ve e-business konseptinin hızlı gelişimi, Web Servisleri ihtiyacının artmasına yol açmıştır.

Bugün UDDI ve yalın XML depolaması, Web Servislerinin uygulanması için bütün bileşenleri sağlar. Böylece firmalar, mevcut uygulamalarını Web ortamına kolaylıkla ve uygun maliyetle aktarabilirler, IT ortamlarında yeni Web Servisleri oluşturur.

Microsoft, VisualStudio. NET yazılım geliştirme aracının içerisinde web servisleri geliştirmeye imkan veren yenilikler gerçekleştirilmiştir. Şu an kullanılmakta olan en yaygın web servisi örneği olarak Microsoft un kullandığı "*Passport authenticate*" sistemini gösterebiliriz.

Web Servislerinde kullanılmak üzere geliştirilen teknoloji ve yeni protokoller;

- SOAP (Simple Object Access Protokol): Sistemler arasında XML tabanlı mesajlaşmayı sağlayan bir protokoldür.

- WSDL (Web Service Description Language): WSDL Web Servisi Açıklama Dili anlamına gelir. Yazmış olduğumuz web servis ile ilgili tüm içerik bu dosyanın içerisinde XML formatında bulunmaktadır.
- DISCO (Discovery): Web servisindeki mevcut, kullanılabilir servislerin listesini verir.
- UDDI (Universal Description Discovery and Integration): Web servislerinin listesinin yer aldığı bir çeşit sarı sayfalar olarak düşünebiliriz. (<http://www.uddi.org>)

Bu terminolojilerin arasındaki iletişim ve bir web servisinin çalışma şekli şöyle özetlenebilir. İlk olarak Web Servisini kullanmak isteyen kullanıcı UDDI'dan (<http://www.uddi.org>) istediği servisi arar. Eğer UDDI'dan aranan servisin cevabı geliyor ise UDDI istenen servisin bağlantılarını XML veya HTML formatında gönderir. İstedığımız servisi seçtikten sonra bu isteğin bilgisi Web Servisine gider. Bunun ardından Web servisi yapısını anlattığı WSDL dosyasını XML formatında gönderir. Ardından SOAP Protokolü aracılığı ile konuşurlar.

Web servisleri bileşen tabanlı yazılımların bir evrimi olarak görülebilir. Bu yazılımlar bir kaç standart yazılım parçalarından oluşabilir. Bir yazılım bileşeni kara bir kutu gibidir veri ve mesajları giriş olarak kabul eder çıkış olarak bir sonuç verir genellikle bileşenin nasıl çalıştığını anlamak zordur. Dolayısıyla genellikle tekli parçaları bitleştiren yazılım uygulamaları oluşturmak kolay değildir. WEB servisleri geleneksel birleşen tabanlı yazılımlarının birbiriyle olan bu kopukluğunu standartlaştırarak yok eder yaygın olarak kullanılmasının faydalarından biri yazılım bileşenlerinin online olarak erişilebilir hale getirilebilmesidir. Bu üç temele dayanır.

- 1) Online dizin
- 2) Yazılım bileşenini açıklayan standart dokümantasyon
- 3) Web aracılığıyla yazılım bileşenlerini erişimin kuralları(WEB servisleri kullanılarak uygulamanın belirli bir parçası uygulamanın dışındaymış gibi konumlandırılabilir gerek duyulduğunda uygulamaya uzaktan erişim yöntemleriyle veri ve mesajlarıyla erişilebilir)

ERP sistemleri çok yaygın olarak kullanılması bir firmanın bilgi sisteminin kalbi olarak görülmesine rağmen bir kısım çevre bunu kurmanın ve devam ettirmenin maliyetli olmasından yakınmaktadır özellikle küçük ve orta ölçekli firmalar ERP sistemlerine geçerken üzerinde ciddi olarak bir maliyet planlaması yapmak zorundadırlar. ERP sistemi üreticileri bunun önüne geçebilmek için hazır paket çözümler sunmaktadırlar. Bu çözümler önceden kurgulanmış organizasyon süreçlerini ve bilgi sistem, al yapısını içerir. Web servisleri bu yazılımları birbiriyle uyumlu olarak çalışmasında aracılık yaparlar. Bu yazılımlar diğer sistemleri beslemek için gerekli verileri işlerler ve sağlarlar

3.4 Web Servislerinin Teknik Açından İncelemesi

XML Web servisleri, daha öncede açıklandığı üzere, servisleri iş,uygulama ve sistem servisleri için ağ üzerinden güvenli bir şekilde bilgi paylaşımını sağlayan XML tabanlı ara yüzlerdir. Bugüne kadar kullandığımız COM, COM+ yapılarının internet versiyonudur diyebiliriz. Arka planda sürekli çalışan bir dll dosyası, talep geldikçe, belirlenen sınırlar içerisinde bilgiyi ağ üzerinden istemciye sunar.

Dili standarttır, dolayısı ile herhangi bir uygulama bu talebi yapabilir. E-business çözümlerindeki gibi talebi oluşturan bir kullanıcı ve web tarayıcı olmak zorunda değildir. İstemci uygulama, web servisine isteğini iletir, web servisi istenen doğrultusunda verilen kriterlere göre sonucu üretir ve standart bir dil olan XML formatında istemciye iletir. Bu bilginin iletimi de farklı özelliklerde ki sistemler birbiri ile iletişim kurabilsin diye standartlaştırılmıştır. Bu iletişim yöntemi SOAP mesajlarıdır.

Aşağıda bir web servisi oluşturmak için gerekli temel adımlar listelenmiştir.

- Bir sağlayıcı kendi istediği bir programlama dili,middleware ve platform kullanarak bir web servisi oluşturur,birleştirir ve konuşlandırır.
- Sağlayıcı web servisini WSDL (Web Servisi Tanımlama Dili) kullanarak tanımlar.WSDL dokümanı web servisini diğer birimlere tarif eder.

- Sağlayıcı servisi UDDI(Global Tanım,Keşif ve Entegrasyon) kayıtlarına kayıt eder.UDDI geliştiricilerin web servislerini yayınlamasını ve kendi yazılımının diğer servislerden haberdar olmasını mümkün kılar.
- Bir kullanıcı servisi UDDI kayıtlarını arayarak bulur.
- Kullanıcın uygulaması web servisine bağlanır ve servisin işlevlerini SOAP(Basit Nesne Erişim Protokolü) kullanarak çağırır.SOAP , HTTP üzerinden parametrelerin ve dönüş değerlerinin temsili için bir XML düzeni sunar.Bu tüm web servislerinin kullandığı bir iletişim protokolüdür.

Yukarıdaki teknolojiler sadece basit web servisleri için yeterlidir.Gelişmiş iş tanımları iş hareketleri,çoklu istek hareketleri,şemalar, ve doküman tanımları için uzlaşma sağlanmış yapılar gerektirmektedir.Bu uygulama gereksinimleri saf bir SOAP tabanlı gösterimin limitlerini zorlamaktadır.Bu, ebXML kavramının doğmasına yol açmıştır.ebXML B2B uyumu ve entegrasyonu için bir e-altyapı niteliği taşıyan XML spesifikasyonudur.

Yukarıdaki yaklaşım web servislerinin çalışması için kullanılan bir yoldur.Bunun için şüphesiz ki başka yollarda bulunmaktadır ancak bu teknolojiler en önemlileri ve en çok kabul görenleridir.

- Sun ve Microsoft SOAP,WSDL,ve UDDI terimlerinin ve türevlerinin gelecekte kullanılacağını söylemektedir
- Tüm üretici firmalar bir web servisi standardı oluşturmak için beraber çalışmaktadır.

3.4.1 XML

Bağımsız bir kuruluş olan W3C (World Wide Web Consortium) organizasyonu tarafından tasarlanan ve herhangi bir kurumun tekelinde bulunmayan XML (eXtensible Markup Language), kişilerin kendi sistemlerini oluşturabilecekleri, kendi etiketlerini tanımlayarak çok daha rahat ve etkin programlama yapabilecekleri ve bu belirlenen etiketleri kendi yapıları içerisinde standardize edebilecekleri esnek, genişleyebilir ve kolay uygulanabilir bir meta dildir.

Çok farklı tipteki verileri orijinal formatlarında tek bir havuzda tutabilen XML, bilgiye hızlı, kolay ve ortamdan bağımsız olarak erişebilme imkanı sunar. Günlük yaşantımızda kullanmakta olduğumuz verilerin %80'ini oluşturan ve "*unstructured*" olma özellikleri nedeniyle kendi buldukları medya dışında veri özelliklerini koruyamayan (kelime işlem, elektronik tablo çıktıları, PDF dokümanları, ses, resim vb) farklı tipteki verilerin, oryantasyona gerek duymadan hiyerarşik bir yapıda kullanılabilmelerine olanak vermekte ve bu verilerin hızlı bir şekilde sorgulanabilmelerini sağlamaktadır. Öncelikle veri transferinin kolaylaşmasını ve verinin içerik bilgisiyle saklanabilmesini hedefleyen XML, içerik ve sunum bilgilerini birbirinden ayırır. Bu özelliği ile de HTML'den farklılaşır.

XML'in özelliklerine gruplaşan göre farklı tanımları aşağıda verilmiştir.

- XML bir dokümanın yapısını ve görünümünü tanımlamak için kullanılan uluslararası bir standarttır.
- XML (Extensible Markup Language - Genişletilebilir İşaretleme Dili) yapılandırılmış belge ve verilerin evrensel formatıdır
- XML text tabanlı markup dilidir ve data alış verişinde kullanılan bir standarttır.
- XML bilginin yapısını tanımlamak için kullanılan bir teknolojidir.
- XML bilgiyi tanımlayan ve web'te bilgi alış verişi için kullanılan standart bir biçimdir.
- XML markup dillerini tanımlayan bir metadildir.
- XML verinin yapılandırılması ve tanımlanması için kullanılan bir teknolojidir.
- XML herhangi bir verinin biçimlenmesi, tanımlanması için kullanılan bir teknolojidir.

Yukarıdaki tanımlar birbirine benzemektedir. Bazı kelimeler veri, tanımlama, standart vs.. gibi çok fazla kullanılmış. Tanımlardaki bir farklılık XML'in hem bir teknoloji hem de bir dil olmasından doğmaktadır. Bazı tanımlar sadece XML teknolojisini tanımlarken bazıları dil olarak XML'i tanımlamışlar.

Bütün bu tanımlardan şu sonuçlar çıkarılabilir

- XML hem bir teknolojidir hem de bir dildir
- XML dil olarak markup dil'leri yaratmaya yarar.
- XML verileri tanımlamak için kullanılan bir teknolojidir.
- XML verileri tanımlamak için bir standart oluşturmak için yaratılmıştır.
- XML verileri standart bir şekilde tanımladığından web'te veya herhangi iki program arasında veri alış verişi kolaylaştırmaktadır.

3.4.2 SOAP (Simple Object Access Protokol)

Dağıtık, tek merkezli olmayan uygulamaların veya çeşitli noktalardaki nesnelere, kendi barındıkları platformla aynı veya farklı platformlardaki nesnelere aynı dili konuşması mevcut nesne iletişim ara yüzlerinin (COM, CORBA, EJB gibi) kullandığı transfer protokollerinin gerek uygulamayla bağdaşmaması, yani kullanılan programlama diline veya işletim sistemine uyumsuzluğu, gerekse protokolün mevcut diğer protokollerle olan uyumsuzluğundan dolayı hep sorun teşkil etmiştir ve ortak standart bir protokole ulaşamamıştır. Bu noktada SOAP devreye girerek, web servisleri ve dağıtık uygulamalar arasındaki iletişim karmaşasına bir son verip standartlaşmaya gitmeyi hedeflemektedir.

SOAP (Simple Object Access Protokol - Basit Nesne Erişim Protokolü), distributed (dağıtık) uygulamalarda ve web servislerinin haberleşmesinde kullanılmak üzere tasarlanan, RPC (Remote Procedure Call) modelini kullanan, istemci/sunucu mantığına dayalı bir protokoldür. Dağıtık uygulamalarda ve Remote messaging için yeni bir teknoloji olmasına karşılık, alt yapısı sayesinde web üzerinde kullanmak için en uygun protokol olması nedeniyle Web Servislerinin adı SOAP la birlikte anılır olmuştur.

Internet üzerinden programların birbirleri ile haberleşmesini uygulama geliştirirken sağlaması önemlidir. Uygulama bütünleştirme için RPC (Remote Procedure Calls) gibi çözümler kullanılmaktadır fakat HTTP bunun için tasarlanmamıştır. RPC uyum ve güvenlik problemi göstermektedir. Firewall, Proxy gibi sunucular bu çeşit veri alışverişi için engel oluşturmaktadır. Örneğin birçok firewall 80. port (HTTP)

dışındaki portlara erişimi engeller. SOAP'ta mesajlar HTTP üzerinden XML verisi olarak taşındığından güvenlik duvarına takılmazlar.

HTTP üzerinden uygulamalar arasında bilgi iletişimini sağlamak iyi bir yoldur, çünkü HTTP'yi bütün internet browser'leri ve sunucuları destekler. SOAP bunu sağlamak(başarmak) için yaratılmıştır. SOAP farklı platformlarda (Birbirinden farklı İşletim Sistemlerinde çalışan , farklı programlama dillerinde hazırlanmış teknolojiler. .) çalışan uygulamalar arasında veri iletişimi sağlamayı desteklemektedir.

SOAP'ın avantajlarını irdelemeden önce benzer protokoller, yapıları ve getirileri hakkında bilgi sahibi olmak, SOAP'ın avantajlarını daha iyi anlamak için uygun olacaktır. SOAP ile aynı kategori de sayılabilecek teknolojilerden en yaygın olarak bilinen ve kullanılanları DCOM, CORBA/IIOP ve Java EJB/RMI dir. Bunlardan kesmenin sol tarafında kalanlar, nesne iletişim teknolojileridir, sağ tarafındakiler ise kullandıkları iletişim protokolleri yani SOAP alternatifleridir.

COM+, Microsoft'un COM (Common Object Model - Ortak Nesne Modeli) teknolojisinin gelişmiş olan, farklı nesnelerin konuşabilmesi için ortak ara yüz sağlayan bir teknolojidir. İçinde kullanıldığı programlama dilinden bağımsızdır ve teoride işletim sisteminden de bağımsızdır. Teoride dememin sebebi, COM'un değişik platformlarda çalışabilecek şekilde tasarlanmış olmasına rağmen, pratikteki kullanım alanının MS Windows işletim sistemleri dışına çıkmamış olmasıdır. COM objelerinin diğer işletim sistemleri altındaki nesnelerle iletişim kurabilen uyarlanmış versiyonu DCOM (Distributed COM) dur.

CORBA (Common Object Request Broker Architecture), yine dilden bağımsızdır ve platformlar arası iletişim için daha çok tercih edilir. İletişim protokolü olarak IIOP kullanır. Çeşitli üreticiler tarafından geliştirildiği için farklı sürümleri arasında uyumsuzluklar yaşanabilir.

Java EJB (Enterprise Java Beans) için dil bağımsızlığı bakımından durum farklıdır. EJB objeleri Java dilinde yazılmıştır ve yine sadece Java teknolojisine ait bir protokol olan RMI (Remote Method Invocation) kullanırlar.

SOAP'ı benzeri protokollerden ayıran en belirgin ve üstün özelliği yapısının (daha doğru bir ifade ile mesaj formatının) XML üzerine kurulu olmasıdır. Dolayısıyla SOAP, XML in sağladığı esneklik, kolaylık ve platform bağımsızlığı özelliklerini içerir. XML'in bütün bilgisayar dünyası tarafından kabul görmüş sağlam bir standart olması ve uygulamalarda yaygın olarak kullanılmasından dolayı, text dosyası okuyup, XML işleyebilen bütün platform ve uygulamalar, kolaylıkla SOAP bilgisine de erişip işleyebilirler. SOAP mesaj formatı olarak XML, transfer protokolü olarak ta HTTP kullanır. SOAP'a alternatif teknolojiler, kullanıldıkları mimariler aynı olsa bile, firewall tarafından aralarında haberleşecekleri portlardaki veri akışına izin verilmedikçe, firewalllar arkasında varlık gösteremezler. Diğerlerinin aksine SOAP, HTTP protokolünü kullandığı ve taşıdığı mesajlar düz yazı dosyası halinde olduğu için firewall dostudur.

Bütün bu artılarının yanı sıra, SOAP komple bir dağıtık ortam haberleşme yapısı olmadığı için taşınan bilginin güvenliği ayrı olarak değerlendirilmelidir. Hız bakımından SOAP zarfı (Mesajın içinde ne olduğu ve nasıl işlenmesi gerektiğini belirten kısım) alındığında gerekli objelerin oluşturulması için, SOAP zarfının açılması, XML'in işlenmesi ve parametrelerin dönüştürülmesi işlemleri zaman aldığından alternatiflerine görece yavaş kalabilir.

Sonuç olarak, SOAP, XML gibi sağlam, kolay kullanılabilir ve gelişmeye açık bir temele dayandığı için kısa zamanda alternatiflerinin yerini alarak iletişimde standartlaşmaya doğru atılan yeni bir adım olacaktır.

Eğer SOAP 'nün özelliklerini maddeler halinde verirsek;

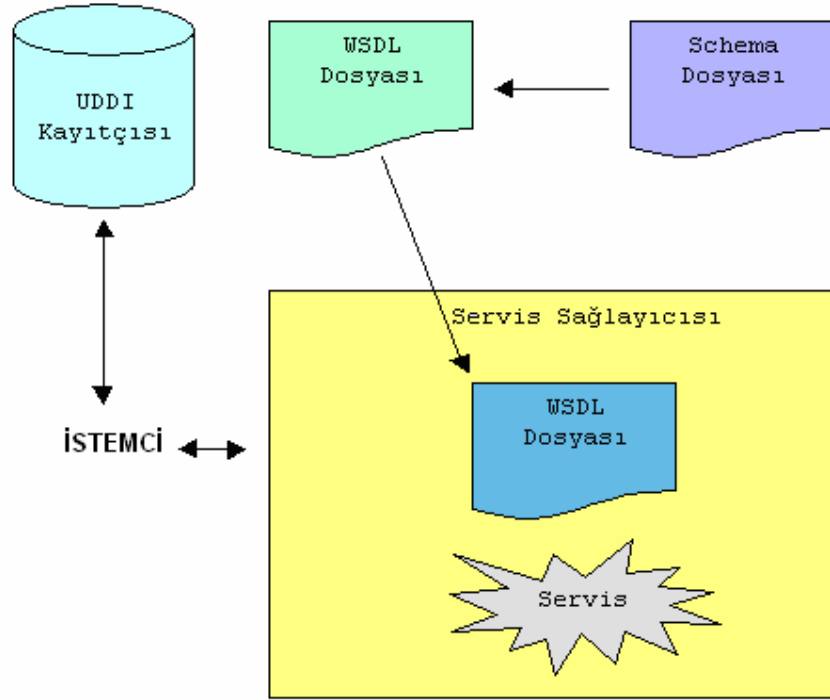
- SOAP, iletişim protokolüdür.
- Uygulamalar arasındaki iletişimi sağlamak içindir.
- Mesaj göndermek için bir format(standart bir yapı)dır.
- İnternet üzerinden veri alışverişini gerçekleştirmek için tasarlanmıştır.
- Platform ve Dil'den bağımsızdır.
- XML tabanlıdır.
- Basit ve Geliştirilebilirdir.

- Firewall gibi sunucularda kolay bir şekilde çalışmaktadır.
- W3C standardı olarak geliştirilmektedir.

3.4.3 UDDI (Universal Description Discovery and Integration)

Web servislerinin altında yatan temel mantık, insanların web servisleri yaratıp, diğer kişilerin bu servisleri web üzerinden kullanmasıdır. Bu yüzden web servislerinin bilgilerinin bulunduğu, insanların ihtiyaçlarını karşılayacak servisleri arayarak nasıl kullanacaklarını öğrenebildikleri bir yapıya ihtiyaç duyulmuştur. Bu noktada, bu düşünceyi gerçekleştirmek için iki yöntem üzerinde çalışılmıştır.

Halen geliştirilmeye devam edilen ilk yöntem UDDI (Universal Description Discovery and Integration) projesi, halen Microsoft, IBM ve diğer şirketler tarafından desteklenen bir kayıtçı sistemidir. UDDI, web servislerinin sarı ve yeşil sayfaları olarak düşünülebilir. Bir kullanıcı ihtiyaçlarına uygun servisi UDDI' da arayarak uygun servisleri bulabilir, bu servisleri nerede bulabileceğini ve nasıl kullanacağını öğrenebilir ve ihtiyaçlarını karşılar. Genel olarak böyle çalışan sistem, tam olarak bitirilmiş değildir. UDDI, daha çok yeni olduğu için, sürekli yeni sürümleri çıkmaktadır ve eksikleri tamamlanmaya çalışılmaktadır. UDDI' ı kullanmak için yaratılan API' ler de tam olarak geliştirilmemiştir. Örnek olarak, bu konuda en büyük çabayı sarf eden şirketlerin başında gelen Sun Microsystems, geliştirmekte olduğu JAXR API'nin üzerinde çalışmaların devam etmekte ve yeni sürümlerini çıkarmaktadır.

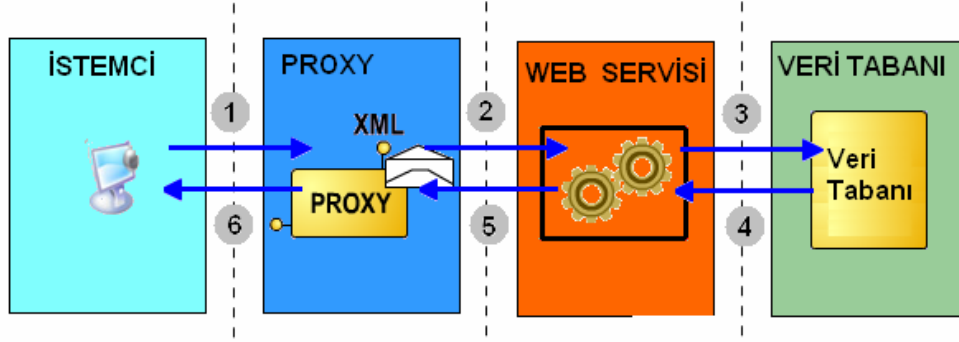


Şekil 8 – UDDI’ da Servis Bulma Senaryosu

UDDI’ a yakından bakacak olursak, bazı eksiklikler görürüz. UDDI’ da her servis için tanımlı meta data bilgisi sabittir. Ama farklı kriterlerdeki servislerinin, kriterlerine uygun olarak kendilerine özgü meta data’larının olması gerekmektedir. Şu anda bu ekstra meta datayı UDDI’ da birkaç yöntemle tanımlayabilmekteyiz (categoryBag, keyedReference gibi) ama bu yöntemler yeterli olmamakla birlikte, UDDI tarafından da desteklenmemektedir. Kullanılan API’ ler ile ekstra meta data ekleme yapmak mümkün olsa bile UDDI henüz desteklemediği için, bu işlemler yapılamamaktadır.

3.4.6 Web Servislerinin Çalışma İlkesi

Web Servisi uygulamaları 4 ana kısımdan oluşur. İstemciler ,Proxy (Referans), Web Servisi, Veri Tabanı. Referans kısmı istemci yazılımının içerisine gömülü olsa da bu arka planda çalışan temel ana birim olarak ele alınmalıdır. Aşağıdaki grafik web servis uygulamalarını oluşturan bu 4 temel ana unsurun birbiri ile olan iletişimini anlatmaktadır.



Şekil 9 – Web Servislerinin çalışma mantığı

İstemciler masaüstü yazılımlar, mobil yazılımlar, İnternet veya internet yazılımları olabilir. İstemci öncelikle Proxy’i tanımalıdır. İstemci isteğini proxy’e iletir.(Adım 1) Proxy’i isteği yorumlar aldığı meta veriyi XML formatına çevirir. Ve bu talebi kendinde tanımlı olan yol vasıtası ile, SOAP mesajlarını kullanarak Web servisine iletir. (Adım 2)

Web Servisi Soap mesajını algıladığında, XML veriyi alır ve gerekli işlemleri gerçekleştirip, veri tabanına uygun sorguyu iletir. (Adım 3) Veri tabanı sorgu sonucunu üretip, kendi dilinde Web servisine sonucu iletir. (Adım 4) web servisi aldığı veri tablosunu, XML’e çevirir, Sonucu ona isteği ileten proxy’e soap mesajları yardımı ile internet yada intranet bulutu üzerinden iletir. (Adım 5) Proxy veriyi aldıktan sonra üzerindeki soap taglarını ayırır, XML veriyi istemcinin anlayacağı dile çevirir ve istemciye iletir. (Adım 6) İstemci yazılım aldığı ham veriyi işleyerek istemci kullanıcıya gerekli bilgiyi görüntüler.

3.5 ERP ve Web Servisleri

Bilginin statik olmaması gibi, yazılım ve erişim yöntemleri de statik değildir. Bir ERP sistemi sürekli olarak, kontrol altında geliştirilmeli, bilginin stabil durumu sağlanmalı, güvenliği garanti altına alınmalıdır. Bir ERP sisteminin, tüm yeni özelliklerini, istemci bilgisayarlar üzerinde herhangi bir geliştirme işlemi gerçekleştirilmeden yüklenebileceğini düşünürsek; sistem işleyişi bu rutinde devam ettiğinde, herhangi bir ekstra geliştirme işlemi olmadan, bilgi işlem bölümü çalışanlarına ihtiyaç duymadan, ekstra uyum sağlama veya geliştirme kodları

yüklenmeden sistemin devam ettirilebilirliği Web Servisleri ile sağlanabilir. ERP uygulamalarında, gerek web servislerinin kullanılması iki kat avantaj getirmektedir. Birincisi entegrasyonu kolaylaştırır, diğeri ise uygulama modelinin maliyetlerinin azaltır.

Yüzeysel olarak, web-erişimli API sunan, web servisleri, o uygulamanın web üzerinden programatik olarak çağrılabilceği anlamına gelmektedir. Web servisleri birlikte-çalışabilir dağıtık uygulamalar geliştirmek için yeni bir platformdur. Bu platform, uygulamaların web aracılığıyla birlikte-çalışabilirliği elde etmek için izledikleri standartlar kümesidir. Web servisleri herhangi bir dilde ve herhangi bir platformda yazılabilir. Çünkü bunlara web servisleri standartlarına göre erişilir.

IBM'e göre web servisleri, web üzerinde tanımlanabilen, yayınlanabilen, bulunabilen ve çağrılabilen modüler uygulamalardır. Microsoft'a göre, bir web servisi, standart Internet protokolleri kullanılarak erişilebilen programlanabilir uygulama mantığıdır. Gartner Group'a göre bir web servisi, genel kullanıma açık bir ağ üzerinden diğere uygulamalar (bir istemci, sunucu ya da başka bir web servisi) tarafından erişilebilen ve bir iş fonksiyonunu temsil eden yazılım bileşenidir. Hepsinin ortak tarafı ise, Internet üzerinden HTTP protokolü aracılığıyla işaretleme dili bileşenlerinin aktarılmasıdır.

Bir web servisi, standart XML mesajları aracılığıyla ağ üzerinden erişilebilir işlemler topluluğunu tanımlayan bir ara yüzdür. Bir web servisi, standart ve biçimsel XML tasarımı kullanılarak tanımlanır ve buna web servisinin servis tanımı denir. Bu, servisle etkileşimde bulunmak için gerek duyulan mesaj formatları, aktarım protokolleri ve konum gibi tüm detayları içerir. ara yüz, servisin gerçekleştirim detaylarını gizler. Böylece, servisin gerçekleştirildiği yazılım ya da donanım platformundan ve yazıldığı programlama dilinden bağımsız olarak kullanılabilmesine olanak verir. Bu da, web servisleri tabanlı uygulamaların zayıf bağı, bileşen tabanlı ve çapraz teknoloji gerçekleştirmeleri olmalarını sağlar ve teşvik eder. Web servisleri belirli bir görev ya da görev kümesini yerine getirir. Bunlar, tek başlarına ya da karmaşık iş bütünlerini başarmak için diğere web servisleriyle birlikte kullanılabilirler.

Bir web servisi herhangi bir işletim sistemi üzerinde yazılmış olabilir, farklı bir işletim sistemi üzerinde çalışabilir, yazılım dili bağımsız olarak oluşturulabilir (ASP.NET bir web servisi oluşturabilmek için bir çok gelişmiş yeteneğe sahiptir.) Çünkü web servislerinin başarısı, onların dil bağımsız, platform bağımsız, protokol bağımsız olarak oluşturulabilmesidir. Diğer bileşen teknolojileri gibi, web servisleri DCOM(Distributed Object Model) kullanmaz, yerine tüm internete bağlı olan her cihaz için standart olan XML (Extensible Markup Language) veri formatını, HTTP (HyperText Transfer Protokol) ve SOAP(Simple Object Access Protokol) protokollerini kullanırlar. Diğer bir deyişle, web standartlarını destekleyen herhangi bir sistem web servislerini de desteklemektedir.

Bu gelişimle, ERP'in şirket içerisinde yapacağı işi, dış dünyaya belli kısıtlamalar ve güvenlik mekanizmaları ile açılmış olur. Ortak bir dil olan XML ve soap tabanlı olması, verilerin masaüstü veya diz üstü bilgisayarlar haricinde ki cihazlarda, tablet Pc ler Smart Phone'lar veya El Bilgisayarları arasında da paylaşılabilmesine izin verilebilmektedir. Bu cihazların hangi işletim sistemini veya yazılımı kullandığı değerlendirilmeden sadece verinin elde edileceği, sunucu ya ait adres bilgisi ve cihazın sunucuya erişim hakkının olmasıdır.

3.5.1 Entegrasyon Kolaylığı

Uygulamaları entegre etmek için web servisleri kullanılarak, bir şirketteki iş süreçleri daha iyi otomatize edilebilir. Birçok işi ilgilendiren iş süreçlerinin entegrasyonu, B2B entegrasyon olarak ifade edilir. Web servisleri, B2B entegrasyonuna imkan veren bir teknolojidir. Web servisleri kullanılarak, hayati iş süreçleri yetkili tedarikçi ve müşterilere sunulmaktadır.

B2B entegrasyonu için web servislerinin kullanılmasının esas avantajı ucuz birlikte çalışabilirliktir. İş fonksiyonlarının web servisleri olarak sunulması, yetkili kesimlerin kullandıkları platform ve programlama diline bakılmaksızın bu servisleri çağırabilmelerine olanak verir. Bu, şirket sınırlarını aşan iş süreçlerinin entegrasyonu için gereken zamanı ve çabayı azaltır ve para tasarrufu sağlar.

Entegrasyon, firmalar için ana masraf kaynaklarından biridir. Uluslararası 2000

firma, bu ihtiyaçlarına çözüm getirmişler ve Bilgi İşlem için ayırdıkları bütçelerinin %33 ünü birbiri ile konuşmayan sistemler arasında iletişim kurmak için harcamışlardır.

ERP genel itibari ile karmaşıktır ve genel kullanım için önerilmez. Bugün, bununla beraber, istemciler ve dış kaynaklı satıcılar, ERP sistemlerinden elde edilecek aynı verilere erişmek istemektedirler. Bu noktada web servisleri, doğru zamanda, doğrulanmış kullanıcılara uygun seviyede ve miktarda veriyi kaynağı belirsiz URL çağruları ile sunarak, bize çözüm sunmaktadır.

Web servisleri sayesinde, güvenilirlik, güvenlik, yönetilebilirlik, yönlendirilebilirlik, keşfedilebilirlik, test edilebilirlik, ve geçerlilik için verilebilecek hizmetin en üst kalitesinde verim alarak sistemler arası entegrasyonu sağlayabiliriz. Web servisleri, farklı uygulamalar tarafından erişilen web servisi yöntemlerindeki programlama elemanları ve veriyi uygun formata getirmek için, temel olarak nesne tabanlı programlama teknolojisi kullanırlar. Bir yazılım köprüsü SAP R/3 sistemi ile PeopleSoft İnsan Kaynakları yazılımını konuşturabilirken aynı köprü diğer İnsan Kaynakları paketlerini SAP'e bağlayıp konuşturamayabilir.

3.5.2 Uygulama Modeli ile Maliyetlerin Azaltılması

Geleneksel ERP sisteminin kurulması, önemli iş süreçlerinin analizini, çalışanların yeniden eğitilmesini ve yeni iş sistem süreçlerini beraberinde getirir. ERP uygulaması ve geliştirilmesi için Web Servislerinin kullanılması, yasal olarak ERP uygulamalarına yapılan harcamaların azaltılması avantajını alır ve yeni bir kullanım ortamı sağlar.

Web Servisleri, tescilli uygulamaların web üzerinden birbiri ile iletişim kurmasını olanaklı kılar. Ana Satıcıların amacı, java veya uygun herhangi bir programı web servisine dönüştüren yüksek seviyeli araçlara erişimi, ara yüz olarak saklayarak sadece işlevselliği sunan ara yüzler oluşturmaktır. Tescilli ERP uygulamaları ve web servisleri bu yüksek seviyeli araçları kullanarak konuşabilirler (HP'nin E-Speak Araçları;IBM'in Dinamik e-business modellemesi ve Sun One).

ERP; entegre çalışan, farklı işlevsellikleri bulunan, çoklu uygulama yazılımları için fonksiyonlar sunar. Bu, istemciye özel olarak yazılmış yazılımların yerine paket yazılımların kullanımını içerir.

3.5.3 Web Servisleri ile ERP Uygulamalarının İyileştirilmesi

Firmalar dahili sistemlerini daha etkili kullanmak için ERP uygulamalarını tercih ederler. Web servisleri, firmaların, etkin olarak istenilen bilgiye ulaşmasını sağlar, hatta bazı durumlarda birbiri ile sıkı konuşan uygulamalar bile gerekli olmayabilir.

ERP sistemler; entegre bir internet bilgi çözümü geliştirerek, bilgileri tüm sistemler üzerinde çalışabilecek, okunabilecek genel bilgi haline getirirler. Bu şekilde oluşturulmuş marketler çok daha etkilidirler. Çünkü, onlar müşteri servisine yoğunlaşırlar. Bu yeni teknoloji iş çevresinden destek aldığı gibi, bir çok satıcı da bu web servislerini destekleyen ürünler geliştirmek zorunda kalacaklardır. Geleneksel ERP tedarikçileri için ürettikleri yazılımlar ile web servislerini konuşturmaları, aynı ortamda buluşturmaları çok zor olacaktır. Buna çözüm olarak aradaki boşlukları kapatacak ekstra yazılımlar üretebilirler.

Web Servislerinin sayısı ve fonksiyonelliği, gittikçe artmaktadır. ERP ve finansman çözümleri sunan üreticiler, web servisi ile ara yüzler geliştirerek geçici çözümler üreterek, bu işlevsellikten faydalanmaya çalışıyorlar. Bu ara yüzler, müşterilerin istediklerini işleyebilecekleri, web servisi hizmetini çalışmasını izleyebilecekleri ara programlardır.

Finansman ve ERP satıcılarının sayısı gittikçe artması, bu ara yüzlerin daha da geliştirilmesi gerekliliğini ortaya çıkarıyor. SAP ve Oracle firmaları kendi yazılımları için mySAP ve Oracle E-Business Suite diye isimlendirilen çözümler sunuyor.

3.5.4 Günümüzdeki ERP Uygulamaları

Ana ERP uygulama yazılımı üreticileri web servislerini programlarının içerisine entegre etmeye başlamışlardır. Oracle, PeopleSoft, SAP şu an hali hazırda çalışan

web servisleri içeren projelere sahiptirler. SAP ve PeopleSoft Web Servislerini uygulama entegrasyonunu kolaylaştırmak için kullanıyorlar.

Oracle, SAP, ve PeopleSoft , Web Servislerini en iyi uygulamalar arası entegrasyon sağlama yöntemi olarak görürlerken; diğerleri web servislerini ve iş-mantık entegrasyonunu sağlamada bir endüstri standardı olarak görüyorlar.

Aşağıdaki tablo ana üreticilerin sundukları web servisi tabanlı yapılarının özelliklerini ve çıkış tarihlerini sunmaktadır.

Üretici	Ortam	Çıkış Tarihi	Pazar Payı	Önerilen
Microsoft	.NET and Passport	2001	1	Bcentral
Oracle	Java	2002	8	11i Suite, Java APIs
Peoplesoft	Tools for SOAP and UDDI	2002	8	Peoplesoft8
SAP	J2EE and ABAP (aynı kernel içinde)	2002	22	R3, mySAP. com
Siebel	Business Services	2001	1	Siebel7

Tablo 2 – Ana Üreticilerin web Servisi tabanlı yapılarının karşılaştırılması

Gartner verilerine göre ERP dünya pazarında 2002 itibarı ile SAP % 22, Oracle % 8, PeopleSoft % 8, Sage % 4, J. D. Edwards % 3, Lawson % 3, Navision % 3, Great Plains/Solomon % 3, Invensys (BaaN) % 2, IFS % 2 , Microsoft %1 , Siebel % 1 pazar payına sahiptir.

3.5.5 ERP Sistemleri ile Web Servisli Çözümlerin Karşılaştırılması

Aşağıda günümüzde kullanılan ERP sistemleri ile bu sistemlerin web servisleri ile desteklenmiş modelleri arasındaki karşılaştırma tablosu Tablo 3'te verilmiştir.

Parametre	Geleneksel Çözüm	Web Servisi Çözümü
Ölçeklenebilirlik	Düşük	Çok Yüksek
Kurulum ve Uygulama Zamanı	Çok Yüksek	Değişken
Devam Edebilirlik	Düşük	Çok Yüksek
Güvenlik	Değişken	Yüksek
Taşınabilirlik	Düşük	Çok Yüksek
Başlangıç maliyeti	Yüksek	Değişken
Devam Ettirme Maliyeti	Yüksek	Düşük
Toplam Maliyet	Yüksek	Düşük
ROI (Yatırımın Geri Dönüşümü)	Değişken	Çok Yüksek

Tablo 3 – ERP sistemleri ve Web Servisleri ile desteklenmiş ERP Modellerinin karşılaştırılması

Bu tablo aynı zamanda bu çalışmanın özeti niteliğini de taşımaktadır. Web Servisleri ile oluşturulmuş bir ERP çözümünün çeşitli yönlerden değerlendirmesini içerir.

i. Ölçeklenebilirlik

Klasik ERP sistemlerine sonradan yeni bir modül eklemek ve bunun uyumlu bir şekilde diğer tüm bölümlerle çalışmasını sağlamak en büyük problemdir. Web Servisleri ile bu sorun ortadan kalkmaktadır. Web servisleri interfaceler ve proxyler ile varolan veriye ulaştıklarından, veriye nereden erişilmek istendiği (veriye ulaşmak isteyen kullanıcıların, erişim hakları yeterli ise) yada metodolojisi sorgulanmamaktadır. Dolayısı ile her modül arasında bulunan, iki modülün bir biri ile veri alışverişinde bulunmasını sağlayan ara yüzler sayesinde web servislerini kullanan ERP sistemlerinin sonraki zamanlarda ölçeklendirilmesi çok daha kolaydır.

ii. Kurulum ve Uygulama Zamanı

Klasik ERP sistemlerini kurmak kesinlikle güçtür. Başlı başına çalışmasını sağlamak, diğer modüllerle uyum problemlerini gidermek, veri havuzunu doğru olarak kullanmasını sağlamak başlıca problemlerdir. Web Servislerinde ise bu durum sistemin yapısına göre değişkenlik göstermektedir. Sistemin büyüklüğünden öte, burada sistemin nasıl işlediği önemlidir. Basit bir anlatımla, bir ERP yazılımını kurmak, bir internete açık veya intranette çalışan web servisini kurmaya göre çok daha zaman alıcı ve bazen daha maliyetli bir iştir.

iii. Devam Edebilirlik

Bir web servisinde oluşan arıza nedeniyle tüm ERP sistemi devre dışı kalmaz: Yönlendirme işlemi ile geçici çözümler çok hızlı şekilde üretilebilir. Yönlendirmenin sınırı yoktur. Fakat klasik ERP sistemlerinde durum daha farklıdır. Bir Veri ambarına erişen ara yüzde oluşan problemi gidermek için kullanabileceğiniz yönlendirme sayısı, sizin donanımınızla sınırlıdır.

iv. Güvenlik

Web servislerinin güvenliği için web servisleri spesifikasyonları geliştiriliyor. Bunlar, Extensible Access Control Markup Language (XACML), Security Assertions Markup Language (SAML), XML Key Management (XKMS), XML Encryption, Web Services Flow Language, XML Digital Signature, Business Transaction Protokol ve Simple Object Access Protokol (SOAP) için uzantılar. Bunların yanı sıra intranet sistemlerinde güvenlik duvarları da bu güvenilirliği desteklemektedir.

Diğer yandan, web servisleri ile, herhangi bir konumdan, genellikle mobil sistemlerde, alınan gerçek zamanlı veri, verinin güvenilirliğinde arttırmaktadır.

v. Taşınabilirlik

ERP sistemleri sadece bir lokasyondaki sistemler değildir. En az iki konumda ERP sunucunuz olması gerekir. Bu konumlardan birincisi verinin işleneceği ortam, (ofis, fabrika) diğeri ise veri ambarının yedeğini tutan sunucunu bulunduğu ortam (afet gibi durumlarda veri kaybının önlenmesi için kullanılan sistemler). IBM ve Microsoft gibi şirketler bu ikincil sunucularını başka şehirlerde özel yapılmış binalarda muhafaza etmektedirler. Bu iki sistemin birbiri ile olan iletişimini sağlamak, bu lokasyonların değişim sonrası ayarlamalarını yapmak klasik ERP sistemlerinde oldukça zor olabilir. Web Servislerini kullanan sistemlerde ise yapılacak işlem basit bir proxy ayarlamasıdır. Yada verilerin bir sunucudan, diğer bir sunucuya aktarımı, üzerindeki tüm kullanıcı erişim haklarının, IP konfigürasyonunun yapılması zaman alıcı ve maliyetli bir işlemdir. Dolayısı ile taşınabilirlik çok düşüktür. Diğer Web tabanlı çözümlerde ise taşınabilirlik sadece bir web servis yönlendirilmesi ile kolayca halledilmektedir.

vi. Başlangıç Maliyeti

Başlangıç maliyeti web servisleri için basitçe web servisini sunacak bir sunucudur. Bu bazen dışarıda olan bir sunucu da olabilir. Eğer istemcilerde web tabanlı ara yüzler ile sisteme dahil oluyorsa sistem maliyeti oldukça düşüktür. Klasik ERP sistemlerinde ise kullanacağınız sunucu özellikleri de aynıdır Yedekleme sistemlerini düşündüğünüzde, bu sistemlerin konuşabilmesi için harcanacak zamanın ve maliyetin boyutlarını düşünürsek, klasik ERP sistemleri çok daha maliyetlidir. İki sistem arası kiralanacak hattın maliyeti oldukça fazladır. Web Servisleri ise varolan internet bulutunu kullandığından bu ücret göz ardı edilebilir.

vii. Devam ettirme Maliyeti ve Toplam Maliyet

Web Servisleri bir kere kurulduktan sonra, lego-vari yapısı nedeniyle devam ettirmek çok kolaydır. Yeni bir modülü eklemek ve adapte etmek, oluşan hatalarda müdahale etme ve sistemi devam ettirme zamanı ve maliyeti oldukça düşüktür.

3.5.6 ERP Web Servislerinin Ekonomik Açıdan Değerlendirilmesi

Web servisleri ile geliştirilmiş bir ERP çözümünün maliyeti, kesinlikle, geleneksel bir ERP çözümünün yüklenmesi durumunda oluşacak maliyete göre ciddi anlamda daha düşük bir değere sahip olacaktır. Bunun sebebi aşağıda listelenen maliyet değerlerinin geleneksel çözümlerinkine göre daha düşük olmasıdır.

- Kurulum ve yaygınlaştırılma maliyeti.
- Danışmanlık maliyeti.
- Gelecekte oluşacak yeni modül ekleme ve ölçeklenebilirlik maliyeti.
- Eğitim maliyeti.
- Entegrasyon ve test maliyetleri.
- Veri Dönüşümü maliyetleri.
- Veri Analiz Maliyetleri.

Web servisi çözümleri kısa zamanda, yüksek verimlilik, ROI(Yatırımın Geri Dönüşümü) düşük yatırım ve hızlı dayanma noktası analizi sağlar

Bir çok küçük ve orta ölçekli firma ERP çözümlerini kullanmaktadı. Geçmişte bu firmaların yıllık kazancı 500 milyon dolardan daha azdı ve büyük pahalı karmaşık ERP paketlerini kurma ve kullanmak için yeterli zamana ve bütçeye sahip değillerdi. Fakat bir dizi ERP üreticisi, Oracle, PeopleSoft, SAP, firmalara web tabanlı modüler, ölçeklenebilir, daha az maliyetli, ERP yazılımları sundular, böylece firmalar ERP kullanma yönündeki kararlarını tekrar düşünmek durumunda kaldılar.

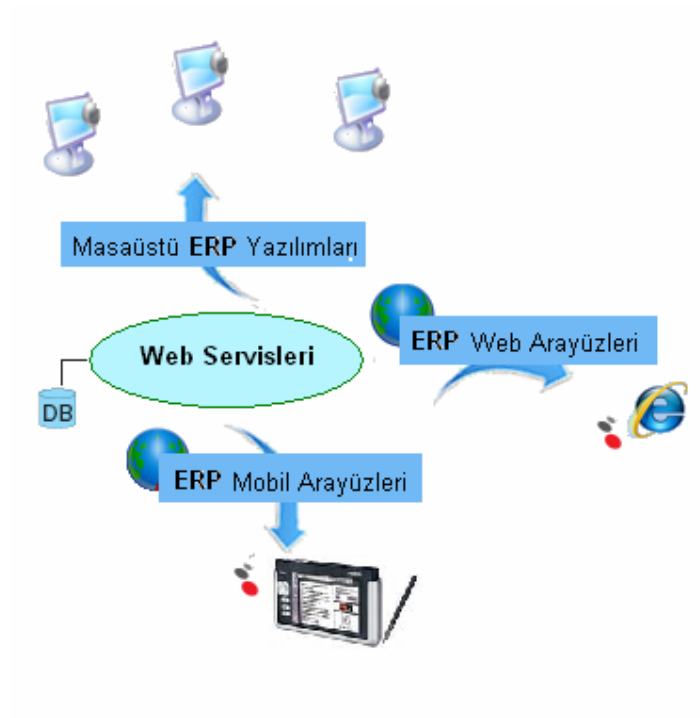
Gelecek iki yıl içerisinde; ERP, tüm dünyada kullanılacak Web Servislerinden ibaret bir platform olarak yeniden tanımlanacaktır. Önceden sadece şirket içi işlemlerin yürütülmesini sağlayan projeler, artık müşteri- tedarikçi zincirini de içine alabilecek şekilde genişleyecek. ERP Web Servisleri, ön işlemler, iş mantığı, e-commerce ve tedarik zincirini de içine alan evrensel iş uygulamaları olarak öne çıkacaktır.

3.7 ERP ve Mobil Sistemler

Bugünün tüketicileri artık mağazalara, kişisel bilgisayarlara veya kara hatlarına bağımlı değiller. Tüketiciler PDA'lar, WAP tabanlı telefonlar ve yeni, etkileşimli mobil ticaret kanalları ile dünyanın her yerinde, günün her saatinde online alışveriş yapabilirler.

Yeni ERP yazılımları bu yeni müşteri tipini memnun etmenize yardımcı olarak mobil ticaret konusunda yeni entegre bir yaklaşım sunar. Sözgelimi, çözüm mobil telefonları ve diğer mobil cihazları kullanarak müşteriler için özel bir Web sitesi oluşturmanızı sağlar. Mağazalara çok nadir olarak giden veya PC üzerinden internet erişimi olmayan tüketiciler için özel olarak tasarlanmış olan bu Web siteleri mobil cihazlara uyarlanmış içerik ve araçlar sunmaktadır. Bu cihazlar üzerine yüklenen yazılımlar, web servislerine ortak ara yüzden erişerek erp ile entegre internet sitesi üzerinden veya masa üstü erp yazılımınıza erişerek istenilen işlemi aynı güvenlik çerçevesinde ve standardında yerine getirmektedirler.

Web servislerini kullanmayan Mobil sistemler, çevrim dışı olarak çalışmaktadır. Dolayısı ile üzerinde taşıdıkları bilgi gerçek zamanlı veri değildir. Kullanıcılar, önce verileri mobil cihazlarına yükler, sonra sunucu ile cihaz bağlantısını koparırlar. Gün boyu veri üzerinde değişiklik yaparlar ve bir zaman sonra sunucuya güncellenmiş verileri aktarmak isterler. GPRS üzerinden çalışan mobil cihazların ERP Web servislerine erişebilmesi ve veriyi eş zamanlı olarak değiştirebilmesi ile bu problem ortadan kalkar. Kullanıcı çevrim dışı değil, çevrim içi çalışmaktadır. Gerektiğinde sunucuya bağlanır, gerekli veriyi alır, işler ve sunucuya geri gönderir.



Şekil 10 – Web Servisleri Yazılım Çevresi

4 ÖRNEK UYGULAMA İLE ERP VE WEB SERVİSLERİNİN KULLANIMININ İNCELENMESİ

Bu tez kapsamında, yapılacak uygulama ERP uygulamalarının bir pilot uygulaması olacaktır. Yapılan uygulama web servisleri ile zenginleştirilerek farklı lokasyonlardaki uygulamaların internet üzerinde bir biri ile bağlantı kurabilmesi sağlanacaktır.

Bir web servisinin, istemci uygulamalar tarafından nasıl kullanılabildiğini anlamak için, web servislerinin mimarisi iyi bilinmelidir. Web Servislerinin Mimarisini kolay bir şekilde anlayabilmek için, web servisi ve istemci uygulamaları üzerinde ele alacağız. Her şeyden önce geliştirdiğimiz web servisi local olarak test edilebilen ve internet tarayıcısı üzerinden çalıştırılabilen bir asmx dosyasından ve buna bağlı çalışan Code-Behind Page olarak adlandırılan kod dosyasından oluşmaktadır. Web servisini test etmek için, web servisinin bulunduğu adresteki asmx uzantılı dosyayı, tarayıcı penceresinden çalıştırmak yeterlidir. Bunun sonucunda, tarayıcı penceresinde bu web servisi hakkındaki bilgilere ulaşabilir ve içerdiği metotları görebiliriz.

Ancak burada yer alan Service Description bağlantısı bize başka bir olanak daha sağlamaktadır. Bu bağlantı yardımıyla web servisimizin tüm içeriğini anlatan bir WSDL dökümanına erişebiliriz. WSDL dökümanının en önemli yanı, XML tabanlı bir içeriğe sahip olmasıdır. Diğer yandan bu döküman, web servisinde kullanılabilecek tüm metotlara, parametrelere ve dönüş değerlerine ilişkin bilgileri içermektedir.

İstemci uygulamanın web servisini kullanabilmesi için, ilk önce web servisinin bulunduğu adrese başvurması gerekir. Bu başvurunun ardından web servisine ait referansı istemci uygulamaya eklediğimizde, bir takım yeni dosyalarında uygulamaya eklendiğini görürüz. Bu dosyalardan belki de en önemli olanı Reference.vb isimli dosyadır. Geliştirdiğimiz uygulama ele alındığında *Reference.vb* dosyasının içeriği Ek B – Kodlamalar -1'deki gibi olacaktır.

Visual Studio .NET tarafından otomatik olarak oluşturulan bu dosyada dikkat çekici noktalar vardır. Her şeyden önce, karşımızda, web servisimizin bir görüntüsü yer

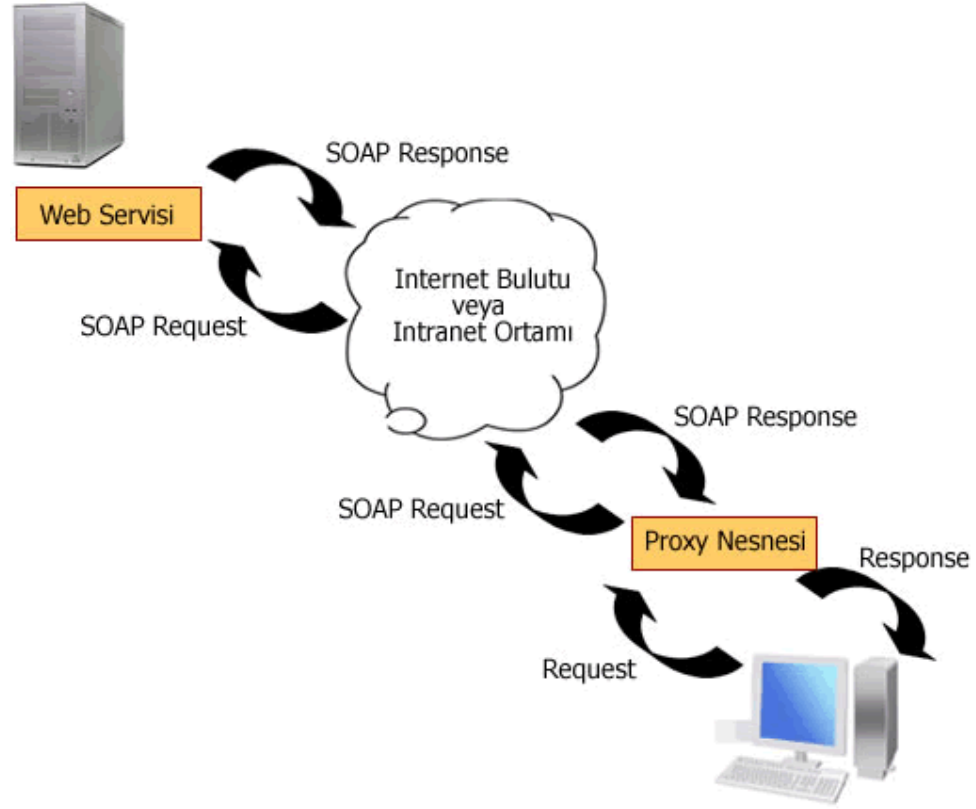
almaktadır. Web servisimizde yazdığımız metotlar kullandıkları parametreler ve daha başka bilgiler. Ancak önemli olan bu dosya sayesinde, istemci uygulamanın artık web servisine ait bir nesne örneğini oluşturup kullanabilecek olmasıdır. Dolayısıyla, istemci uygulamamıza web servisimize ait referansı eklediğimizde, istemci uygulamada bu servise ait bir nesne yapısı oluşturulabilmiştir. Bu sayede aşağıdaki gibi bir bildirim geçerli hale gelir.

```
Dim x As New DadeWebService. Service
```

Dahası, bu nesne üzerinden, web servisindeki metotları aynı isimler ile kullanabiliriz.

```
state = x. _LoginAdmin(Password)
```

Reference. vb' dosyası aslında, istemci ve web servisi arasındaki haberleşmeyi sağlayacak bir Proxy nesnesini oluşturmak amacıyla kullanılır. Dolayısıyla, istemci uygulama web servisini kullanmak istediğinde, yani bu servis üzerinden bir metodu çağırarak istediğinde, bu talebi proxy nesnesinden ister. Proxy nesnesi ise bu talebi , web servisine iletir. Web servisi gelen talebi değerlendirir ve ürettiği cevabı yine istemci uygulamadaki proxy nesnesine gönderir. Proxy nesnesi de sonuçları, uygulama ortamına iletir.



Şekil 11 – Proxy Nesnesini ve SOAP’ ın Xml Web Servisi Mimarisindeki yeri.

Diğer bir sorun, proxy nesnesine gelen taleplerin, web servisine giderken hiç bir engel ile karşılaşmadan nasıl hareket edeceği ve geri geleceğidir. İstemci uygulama, proxy nesnesinde normal olarak talepte bulunur ve cevapları alır. Buradaki ilişki normal olarak bir nesne. metod ilişkisidir. Ancak proxy nesnesi bu mesajları, esnek, kolay okunabilir, herhangi bir engele takılmayacak bir hale getirmek durumundadır.

Web tarayıcı üzerinden service1.asmx dosyasından SOAP mesajlarını incelersek yerel makinede çalıştığımız ve proxy nesnemizi de güncellediğimiz için, proxy sınıfındaki localhost tanımlamasını tekrardan localhost:8080 olarak değiştirmemiz, SOAP mesajlarını Trace Utility aracılığıyla yakalayabilmemiz açısından gerekli olabilir. SOAP Toolkit Trace Utility aracıyla, SOAP mesajlarına baktığımızda, istemciden web servisine `LoginAdmin` metodu için giden XML tabanlı bilgileri Ekler B-Kodlamalar 2’de bulunmaktadır..

_LoginAdmin fonksiyonunun Service. vb dosyasındaki içeriğini Ekler B-Kodlamalar 3’de inceleyebiliriz.

Gördüğünüz gibi, bir desktop uygulama sınıfı içerisinde yer alan fonksiyon ile bir web servisinde yer alan sınıftaki fonksiyon arasındaki tek fark WebMethod tagıdır. Bu tag, oluşturulan fonksiyonun internet üzerinden çağırılabilmesini garanti altına alır.

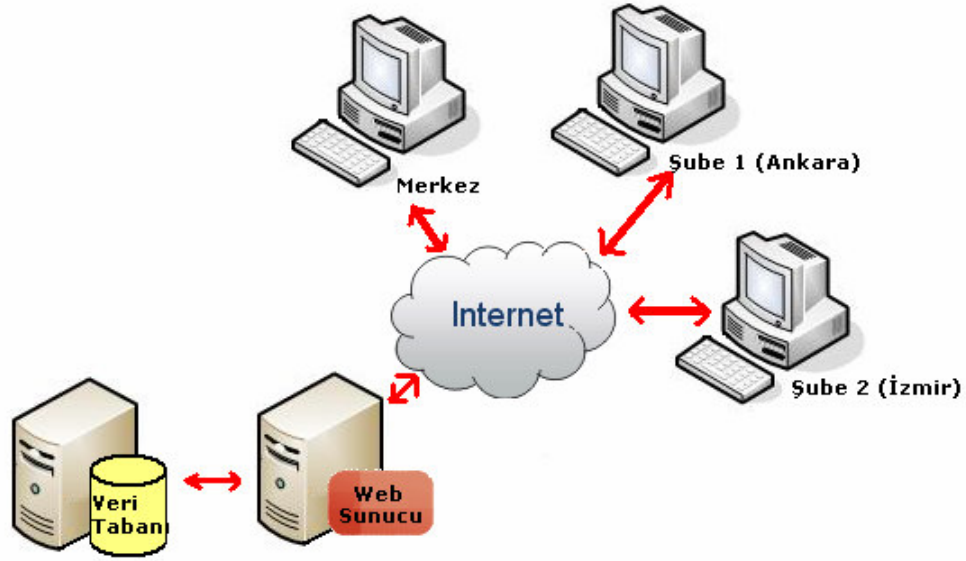
Yazdığımız web servisindeki bir metodun nasıl kullanıldığını incelemek için Ekler B-Kodlamalar 4’deki kod bloğunu inceleyebilirsiniz.

Bunu yönetici giriş formundaki Giriş düğmesinin altında bu fonksiyonu çağırır. Eğer giriş yapılabildiyse, fonksiyon etkin kullanıcı Administrator haklarını devredip, program çalışmaya devam etmektedir..

4.1 Örnek Uygulama

Örnek uygulamada, bir ilaç firmasının karşılaştığı probleme çözüm üretildi. İlaç firması, bir merkezi ve diğer şehirlerde 2 şubeye sahip. Merkez, şubeler ile, merkezde bulunan ürün stoklarını paylaşmak istiyor. Firma, ülke çapında sürekli seyahat halinde olan elemanlara sahip. Firma bu gezici elemanların, istediklerinde güncel ürün stok bilgilerini görüntüleyebilmesini ve siparişlerini anında geçebilmelerini istiyor. Ayrıca firma eczanelerle ortak çalışmakta. Anlaşmalı eczanelerin ürün stoklarından haberdar olması, ve elektronik ortamdan siparişlerini geçmelerini ve takiplerini kendilerinin de yapabilmelerini arzuluyor. Öncelikle problemle ilgili alternatif olabilecek çözümler incelendi. İlk olarak, şubeler ve eczaneler arasında merkezle direk kiralık hat üzerinden bağlantı kurulması düşünüldü. Maliyetli bir çözüm olduğu için ve gezici elemanlarla iletişim kurulamayacağından firmanın ihtiyaçlarını karşılamayacağı görüldü. Web tabanlı arayüzlerle oluşturulmuş bir uygulama çözümü ise web’in açık olduğu tüm tehlikelere açık olacağından, ve raporlama sistemi şirket için yetersiz kalacağından kabul görmedi. Bu bağlamda, en uygun yaklaşım, web servisleri ile tasarlanacak bir program oldu. Sonuçta, firmanın gereksinimlerine de uygun olarak, web servisleri ile zenginleştirilmiş bir ERP uygulamasını örneklemek için bir ilaç firması ve şubeleri arasında veri iletişimi sağlayacak 2 farklı masa üstü uygulama yazıldı. Şubelerin web

servisleri aracılığı ile , merkez sunucuya bağlanıp siparişlerini iletebilecekleri, bu siparişlerin stoklarda var olup olmadığını görüntüleyebilecekleri, siparişlerinin durumunu sürekli izleyebilecekleri bir uygulama kodlandı. Merkezin gelen siparişleri anında görüntüleyebilmesi, bu siparişlerin hazırlanması ve gönderilmesi sağlandı. Aynı zamanda merkez ve şubeler arasında bir mesajlaşma sistemini de içeren web servisi metodu sisteme eklenmiştir. Merkezin veya şubelerin veri tabanı ile olan tüm bağlantısı güvenli bir şekilde web servisleri aracılığı ile yapılmıştır.



Şekil 12 - Sipariş Takip Programı İşlem Zinciri

4.1.1 Veri Tabanı Yapısının İncelemesi

Uygulamamızda, web servis tarafından işlenecek tüm veriler, Microsoft SQL Server 2000 üzerinde saklanacak ve devam ettirilecektir. Kullanıcı tarafında bulunan veriler ise MS-Access tabanlı mdb dosyası içerisinde aynı veri yapısı ile saklanacaktır. Access veri tabanında saklanan tablolarda sadece kullanıcıya ait sipariş bilgileri içermektedir. Gerçek veriler sunucu üzerinde saklanan verilerdir. Herhangi bir veri farklılığı durumunda sunucu üzerinde saklanan veriler daima doğru kabul edilecektir.

Veri Tabanı temelde 3 farklı gruptaki tablolardan oluşur. Firma ve kullanıcı girişleri için gerekli bilgileri içeren bir tablolar, Siparişler için gerekli bilgileri içeren tablolar ve diğer tablolar.

Veri tabanı, şube kullanıcı girişlerinin güvenli bir biçimde yapılabilmesi için, kullanıcı isimlerini ve şifrelenmiş biçimde kaydedilen şube kullanıcı giriş şifrelerini içeren Companies isimli tabloyu içermektedir. Bu tabloda, hem şube kayıtları hem de firma çalışanlarına ait kayıtlar tutulmaktadır. StockCardState tablosu her firmanın kendi yanlarında tuttukları StockCards tablosunun içeriğinin güncel tutulmasını garantiler. Bu iki tablo arasındaki bağ her iki tabloda da C_ID isimi ile adlandırılan firma ID'leri arasında kurulan ikincil anahtar (foreign key) tanımlaması ile kurulmuştur.

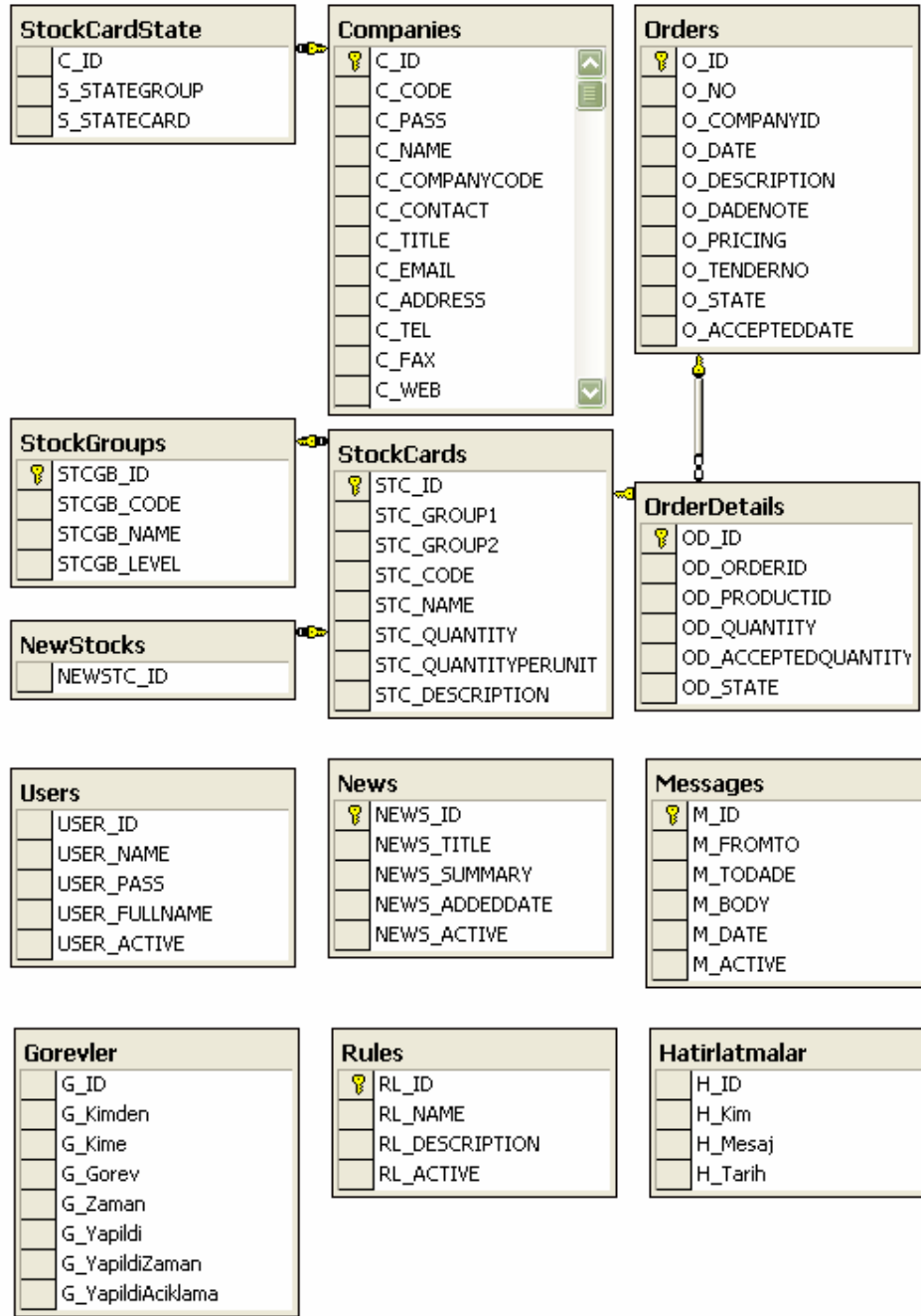
Yönetici ve merkez çalışanlarına ait kullanıcı bilgileri ise Users isimli tabloda muhafaza edilmektedir.

Siparişlere ait bilgiler Orders ve OrderDetails tablolarında saklanmaktadır. Orders tablosu sipariş no, siparişi veren firmanın id'si, sipariş tarihi, tanımlamalar, merkeze ait özel not, toplam sipariş fiyatlandırması, özelleştirilmiş sipariş numarası, sipariş durumu ve siparişin kabul edilme tarihini içeren alanlardan oluşmaktadır. Bir siparişe ait detay bilgiler ise, sipariş id'sini, ürün id'sini, talep edilen, ürün miktarını,ürünün kabul edilebilir miktarını ve durumunu içeren OrderDetails tablosunda tutulmaktadır. Ürünler ve ürünlerin detay bilgilerine, OrderDetails ile StockCards tablosu arasında, OD_ProductID ile STC_ID alanlarının ilişkilendirilmesi ile ulaşılmaktadır.

NewStocks ve News tabloları yardımcı tablolar olarak adlandırdığımız yardımcı bilgileri içeren tablolardır. Programın açılış aşamasında listelenecek, yeni ürün ve haber bilgilerini içermektedirler.

Bu tabloların yanı sıra kullanıcıların mesajlarını içeren Messages, firma haberlerini içeren News, firma ürün alışveriş kurallarını içeren Rules, kullanıcı hatırlatma mesajlarını içeren Hatırlatmalar, firma içi görev yönetimini tamamlayan işlevsellikte

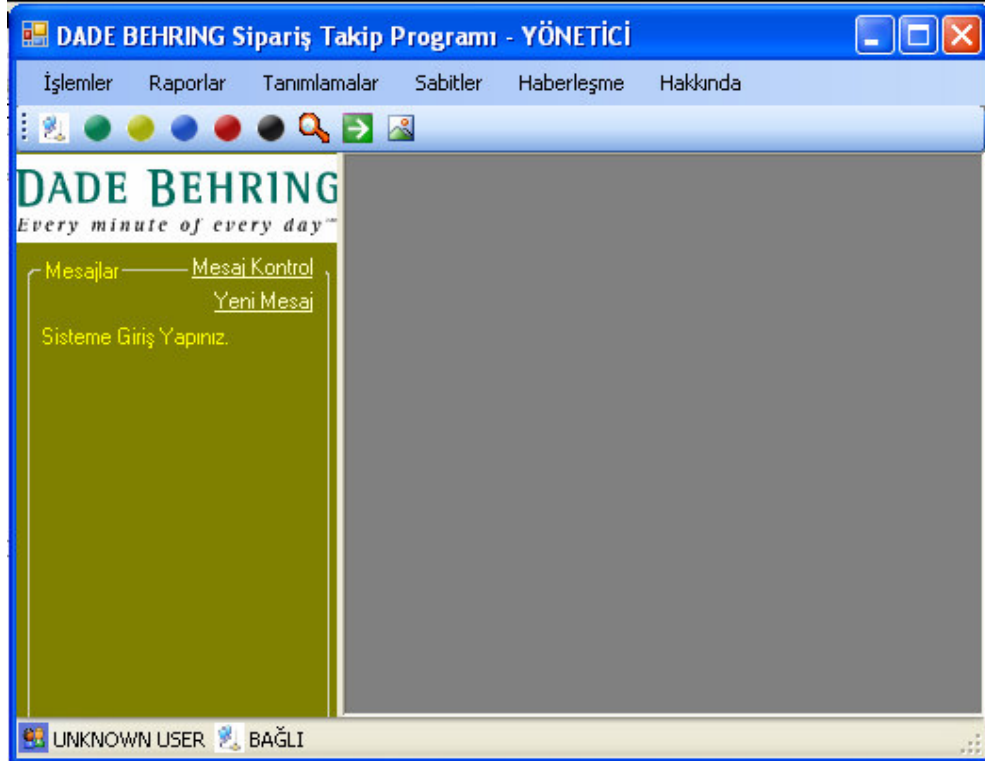
kullanılan Görevler tablosu da yine aynı veri tabanında bulunmaktadır. Tablolar arası her türlü bağlantı ID alanları ile sağlanmış böylece veri tekrarı önlenmiştir.



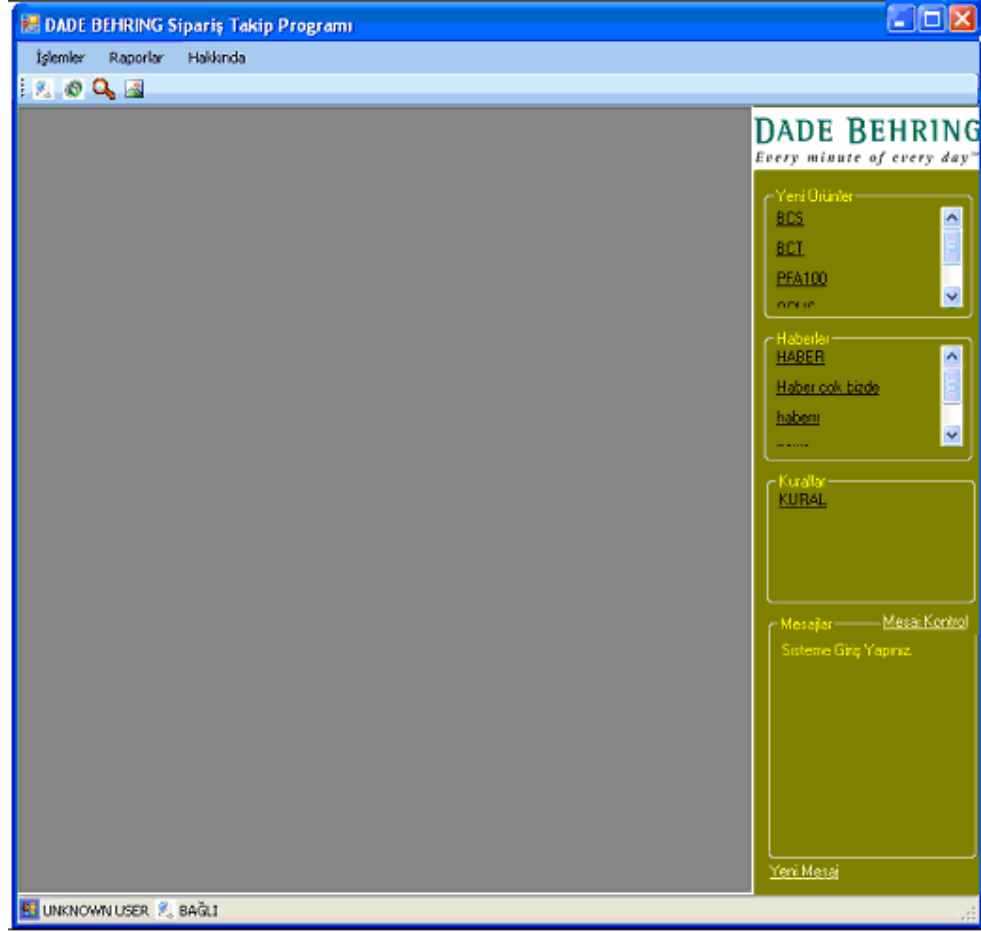
Şekil 13 - Sipariş Takip Programı Veri Tabanı Yapısı

4.1.2 Program İncelemesi

Yazdığımız programda, web servislerine doğrudan erişebilen, yöneticiler ve şubeler için ayrı menüler tasarlanması gerektiğinden, 2 farklı giriş ekranı tasarlanmıştır. Yöneticiler yada şubeler bu programlar arayüzü ile web sunucuna bağlanırlar ve gerekli olan verilere ulaşırlar.



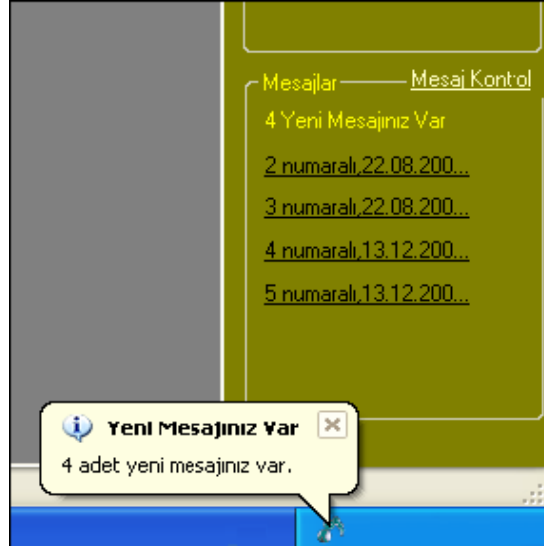
Şekil 14 - Sipariş Takip Programı Yönetici Ekranı



Şekil 15- Sipariş Takip Programı Şube Ekranı

Program ilk çalıştırıldığında kullanıcının web servisine erişip erişemediği kontrol edilir. Bağlantı durumuna göre, durum çubuğunda kullanıcı bağlantı durumu, **BAĞLI** veya **BAĞLI DEĞİL** mesajları ile ikaz verilmektedir.

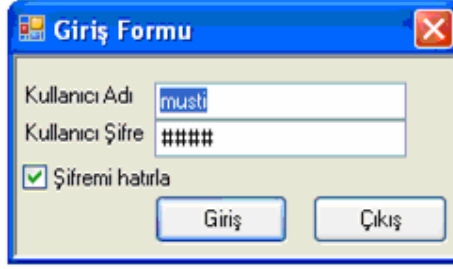
Şubelerin kullandığı ekranın sağ tarafındaki bölümde, yeni ürünler, yeni firma haberleri listelenmektedir. Bu bilgiler web servisinden XML veri olarak alınıp, formatlanarak programda sunulmaktadır. Sisteme giriş yapıldıktan sonra varolan mesajlar yine bu bölümde yer almaktadır.



Şekil 16 - Sipariş Takip Programı Mesaj Alanı

Şimdi bu uygulamadaki işlevsel bölümleri adım adım ele alalım. Öncelikle, program üzerinde incelemeyi yaparken şu sırayı gözeteceğiz, şube sisteme giriş yapacak, siparişini verecek, ardından yönetici siparişi değerlendirecek sonrasında da şube varolan siparişinin durumunu takip edecektir.

Şube programı çalıştırdıktan sonra, siparişini sisteme bağlı iken veya sisteme bağlı değilken olmak üzere 2 farklı konumda oluşturabilir. Bu online ve offline sipariş metodları ile programın sürekli olarak web sunucusuna bağlı kalma zorunluluğu ortadan kalkar. Şubeler, ya da istemciler sadece gerektiği anlarda sunucuya bağlanarak, işlemlerini gerçekleştirerek, web servisi ni de barındıran web sunucusu üzerinde bir işlem yükü oluşturmazlar. Sisteme bağlı olmak ve olmamak arasındaki en önemli fark, ürün listelerinin güncel olup olmamasıdır. Burada örnek uygulamayı incelerken, biz istemcinin sunucuya program açılış aşamasında sunucuya bağlanmak isteyebileceğini düşünerek, online moda sipariş verdiği zamanki gerçekleşecek adımları inceleyeceğiz. Bunun için öncelikle sunucuya bağlanılmalıdır. Sunucuya bağlanmak için **Bağlan** düğmesine tıkladığımızda, kullanıcı girişini yapmamızı sağlayan, kullanıcı adı ve şifresini sorgulayan ekran gelmektedir.



Şekil 17 - Sipariş Takip Programı Sisteme Giriş Ekranı

Bu bölümünde istemci, kullanıcı adını ve şifresini girip *Giriş* düğmesine bastığında, web sunucusu ve program arasında yer alan bir aracı sınıf konumundaki *GetValues* Sınıfı içerisindeki, kullanıcı girişlerini sağlayan fonksiyon olan *_LoginUser* fonksiyonu çağırılacaktır.

```
GetValues._LoginUser(txtKullaniciAdi.Text.Trim, sPassword)
```

_LoginUser fonksiyonu yukarıdaki kodlamamda da görüldüğü üzere, kullanıcı adı ve şifresinden oluşan string tipinde iki parametre almaktadır. Bu fonksiyon, kendi içinde web servisini ve web servisindeki kullanıcı yetkilendirme fonksiyonu olan *_LoginUser* fonksiyonunu çağırır.

```
Dim x As New DadeWebService.Service
xStr = x._LoginUser(UserName, Password)
```

Web Servisini çağırabilmek için, uygulamaya web servisi referansının önceden eklenmiş olması gerekmektedir. Uygulama içerisinde önce web servisinin örneği oluşturulmuştur. Yukarıdaki kodlama örneğinde web servisinin referans ismi *DadeWebService* olarak belirlenmiştir. “Service” *DadeWebService* isimli web servisinde yer alan Servis sınıflarından biridir. *Service* ismi otomatik olarak oluşturulmuştur.

Web Servisinin örneği oluşturulduktan sonra, web servisi içerisindeki tüm özellikler(property), öznelikler(attributes), Veri yapıları (Structures), değişkenler, ve fonksiyonlara bu örnek sınıf vasıtası ile ulaşılabilir. Bu niteliklere ulaşabilmek

için referans dosyası içerisinde oluşan nitelik imzalarından yararlanılır. Yukarıdaki kodlama örneğinde x isimli DadeWebService web servisi sınıfı üzerinden, _LoginUser isimli fonksiyona ulaşılmak istenmektedir.

_LoginUser Fonksiyonuna ait, referans bilgileri Referance.vb dosyası içerisinde yer almaktadır. Siz Web servisini referans olarak uygulamanıza eklediğinizde, o web servisinde kullanıma açık tüm metotların, imzaları bu dosyaya işlenir. Örneğin _LoginUser Fonksiyonuna ait referans imzası Ekler B- Kodlamalar 5’de yer almaktadır.

_LoginUser fonksiyonunun, web servisi içerisindeki tanımı ise Ekler B- Kodlamalar 6’da yer almaktadır.

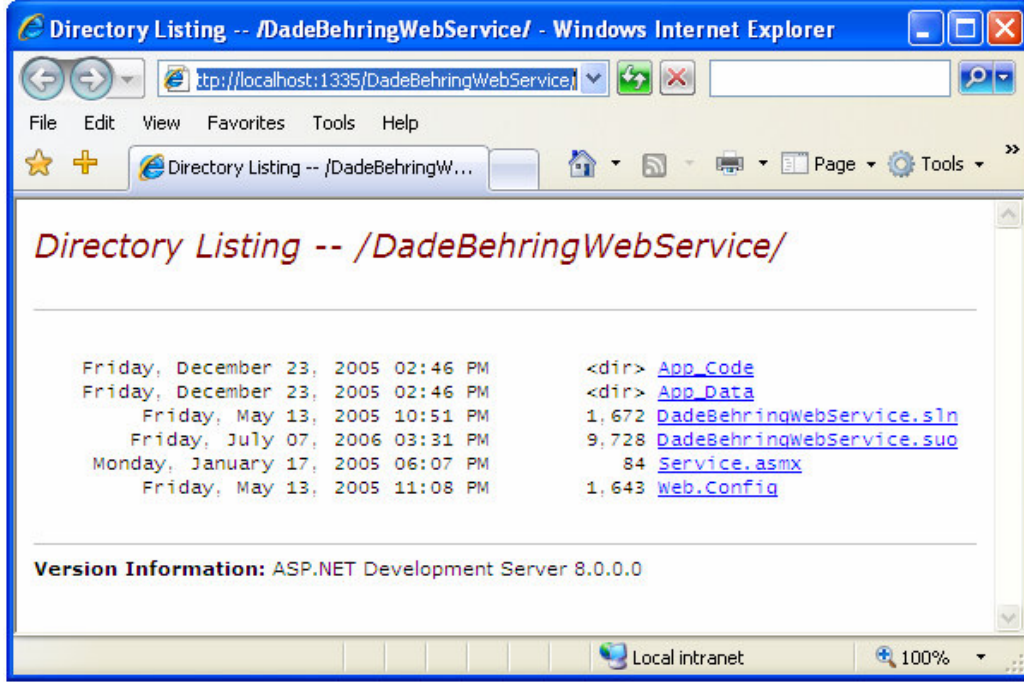
Kullanıcı, kullanıcı adı ve kullanıcı şifresini girip Giriş düğmesine bastığında, bilgiler bu iki parametre değeri _GetValues sınıfındaki _LoginUser fonksiyonuna aktarılır. (İşlem adımı 1). _GetValues sınıfı referans dosyası yardımı ile web servisine ulaşır ve bu iki parametreyi iletir. (İşlem adımı 2). Web Servisi içerisindeki _LoginUser Fonksiyonu içerisindeki kod bloğu çalışır, kullanıcının verdiği bilgiler ile veri tabanındaki kullanıcı bilgileri karşılaştırılır. (İşlem adımı 3). eğer bir uyum varsa, geriye kullanıcı numarası döndürülür. Eğer verilen bilgiler veri tabanında mevcut değilse yani bir uyumsuzluk varsa geriye kullanıcı numarası olarak 0 değeri döndürülür ki, bunun anlamı kullanıcı giriş yapamamıştır. (İşlem adımı 4). Kullanıcı numarası değeri izlediği yoldan geriye döner. (İşlem adımı 5 ve 6)



Şekil 18 – Uygulama Fonksiyon Çağırım Zinciri

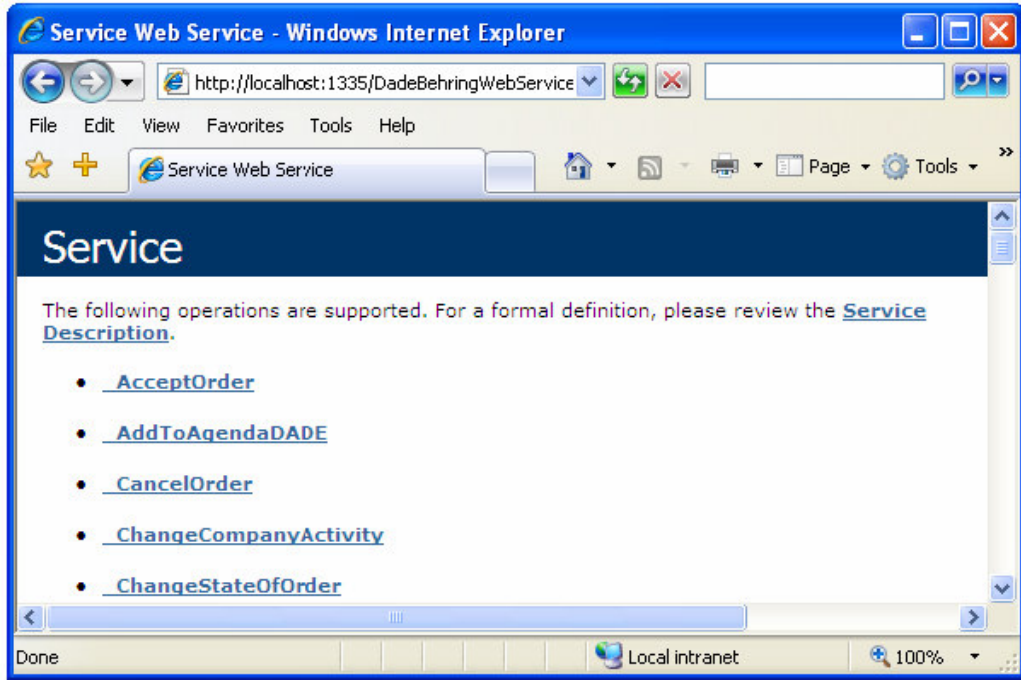
Eğer doğru kullanıcı adı ve şifre girilmişse, Kullanıcı ID sini ve adını içeren bir karakter dizisi, fonksiyonu çağırın metoda sonuç olarak gönderilir.

Web Servisleri içerisindeki metotları aynı zamanda, internet tarayıcı üzerinden de denenebilir. Tarayıcınızın adres satırına <http://localhost:1335/DadeBehringWebService/> yazdığınızda, web servisi projemizdeki dosyalar listelenmektedir.(Şekil 19)



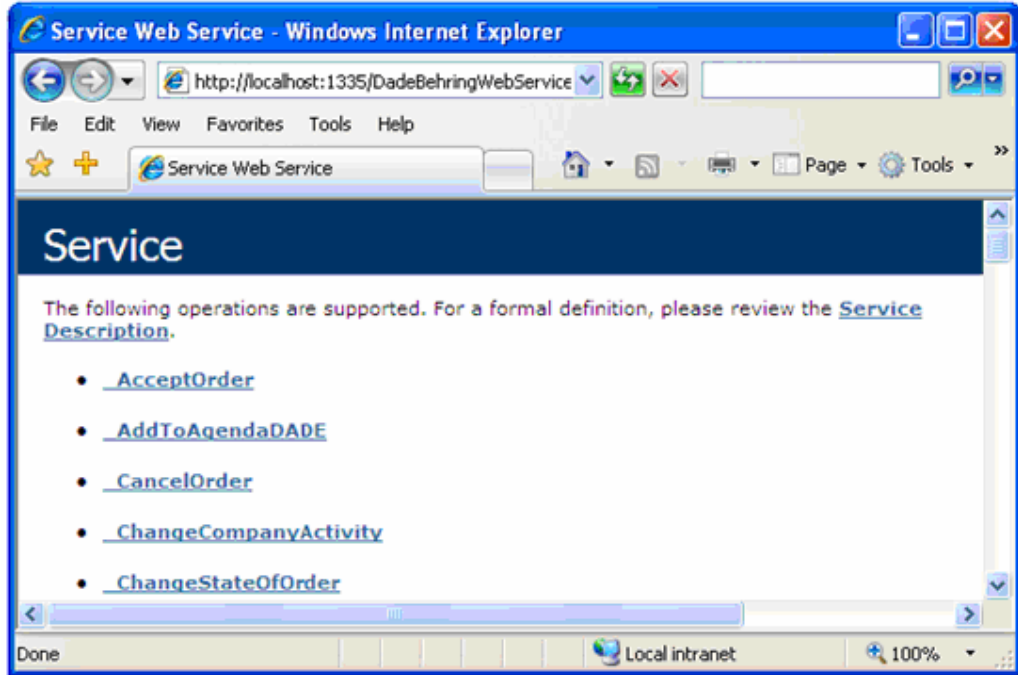
Şekil 19- Sipariş Takip Programı Web Servisi

Service.asmx bizim tüm kodlarımızı içeren bölümdür, Web.config, web servisine ait konfigürasyon ayarlarını içerir. Sln ve suo uzantılı dosyalar, web servislerine ait özel proje dosyalarıdır. Biz web servisini görüntülemek için Service.asmx linkine tıklamalıyız.



Şekil 20- Sipariş Takip Programı Web Servisi Fonksiyonları Listesi

Burada web servisi içerisinde yer alan, erişilebilir tüm metotlar listelenmektedir. Şimdi [_Login User](#) Metoduna tıklayalım.



Şekil 21 - Sipariş Takip Programı Web Servisi Sisteme Giriş Fonksiyonu Test Ekranı

Görüldüğü üzere, _LoginUser isimli fonksiyonu demek için bir ekran açıldı, burada 2 adet parametre alanı görünüyor, bunun anlamı, bu fonksiyonun dışarıdan 2 tane parola kabul ettiğidir. Bu iki değeri girip “Invoke” düğmesine basınca metod çalışacak ve bize sonuç görüntülenecektir. Eğer doğru değerleri yazarsak kullanıcı için fonksiyonun geri döndürdüğü değere bağlı olarak aşağıdaki gibi bir sonuç almamız gerekmektedir.

```
<?xml version="1.0" encoding="utf-8" ?>  
<string xmlns="http://tempuri.org/">1|kullanıcı adı</string>
```

Birde dikkat etmemiz gereken, Web servisi içerisindeki _LoginUser fonksiyonunu denediğimiz bölümün devamında, .net tarafından otomatik olarak oluşturulmuş, fonksiyonun soap mesajları hali yer almaktadır. Bu soap mesajı kodlarını Ekler B – Kodlamalar 7’de incelenebilir.

Şimdi programın içerisindeki sipariş bölümünü inceleyelim. Öncelikle kullanıcı yeni sipariş düğmesine tıklar ve aşağıdaki pencere açılır.

Yeni Sipariş(YENİ SİPARİŞ)

Siparişte Yer Alan Ürünler [Sipariş Listesini Temizle](#)

ÜRÜN KODU	ÜRÜN TANIMI	ADET
EMIT	Syva® Emit® d.a.u.®	10
OPUSP	OPUS	10
PFA 100	PFA100	30

Bayi Açıklaması

Fiyatlandırma
 Normal
 Bedelsiz
 Konsinye
 Tender No

Şekil 22 - Sipariş Takip Programı Yeni Sipariş Giriş Ekranı

Kullanıcı, siparişe yeni ürün talebi eklemek için, “*Ürün Ekle*” düğmesine basar, ardından Şekil 25’de gösterilen ekran görüntülenir.

The screenshot shows a window titled "ÜRÜN SEÇ" with a table of products and a form for selection. The table has columns for ÜRÜN KODU, ÜRÜN TANIMI, ANA GRUP, and ALT GRUP. The product "EMIT" is selected. The form on the right includes fields for "Siparişteki ürün sayısı : 0", "Ürün Kodu" (EMIT), "Ürün Tanımı" (Syva® Emit® d.a.u.®), "Miktar" (0), and "Birim" (x1). There is also an "Açıklama" (Description) field with text about Syva® Emit® products. Buttons for "Ekle" (Add) and "Kapat" (Close) are at the bottom.

ÜRÜN KODU	ÜRÜN TANIMI	ANA GRUP	ALT GRUP
ELISA	ELISA	Hemostasis	Sarf Malzem...
BCS	BCS	Hemostasis	Cihaz
BCT	BCT	Hemostasis	Cihaz
kod	ad	Hemostasis	Yedek Parça
kodum	adm	Hemostasis	Yedek Parça
DPUSP	DPUS	Cardiac	Diğer
PFA 100	PFA100	Drug Monitoring	Sarf Malzem...
EMIT	Syva® Emit® d.a.u.®	Drug Monitoring	Diğer

Şekil 23 - Sipariş Takip Programı Yeni Sipariş Ürün Listesi

Soldaki ürün listesinden kullanıcı ürünü seçer. Seçili ürün sağ taraftaki ürün detay bölümünde görüntülenmektedir. Kullanıcı ürünün adedini girer ve üzerinde Ekle yazan düğmeye basar.

Sipariş hazırlandıktan sonra Kapat düğmesine basılmak suretiyle ürün listeleme ekranı kapatılır. Gerekli notlar yazıldıktan sonra, eğer sipariş daha sonraki bir zamanda, kullanıcı online olma durumunda gönderilecek ise, sipariş daha sonra gönderilmek üzere kaydedilebilir yada doğrudan gönderilebilir.

Sipariş gönderme bölümünde çalışan kod aşağıdaki gibidir.

```
Dim xResult As String = GetValues._SendOrder(str, xSelectedID,
txtDescription.Text, xYurtDisi)
```

Bu projede **3 katmanlı** bir yapı kullanılmıştır, Formlar Web servisine doğrudan değil de bir ara yüz ile erişirler, bu ara yüz "GetValues" isimli ara yüzüdür. Bu ara yüz içerisinde web servisinde erişilecek her fonksiyona ait bir imza bilgisi içerilir. Böylece gelip giden verilerin güvenliği, doğruluğu tek bir noktadan yönetilebilir. Bu arayüze ait kodlama bilgisi Ekler B- Kodlamalar 8'de mevcuttur.

Program, online ve offline olarak, çift yönlü çalışabilmektedir. Eğer kullanıcı offline ise, kendi yanında yer alan MS-Access tabanlı veri tabanını kullanır, buradan kendi siparişleri ve durumları hakkında bilgi alabilir, raporlama yapabilir, ama bu sonuçlar güncel olmayabilir. Bunun avantajı, kullanıcının sürekli web servisine bağımlı kalmamasıdır. Böylece program daha hızlı sonuç üretir, raporlama yapar. Eğer program online moda kullanılırsa, offline moda göre bir hız kaybı olacaktır, Bu hız kaybı, internetten verilerin aktarımı sırasında oluşan zaman kadardır. Online çalışmanın avantajı ise alınan her rapordaki bilginin bilgi güncel olmasıdır.

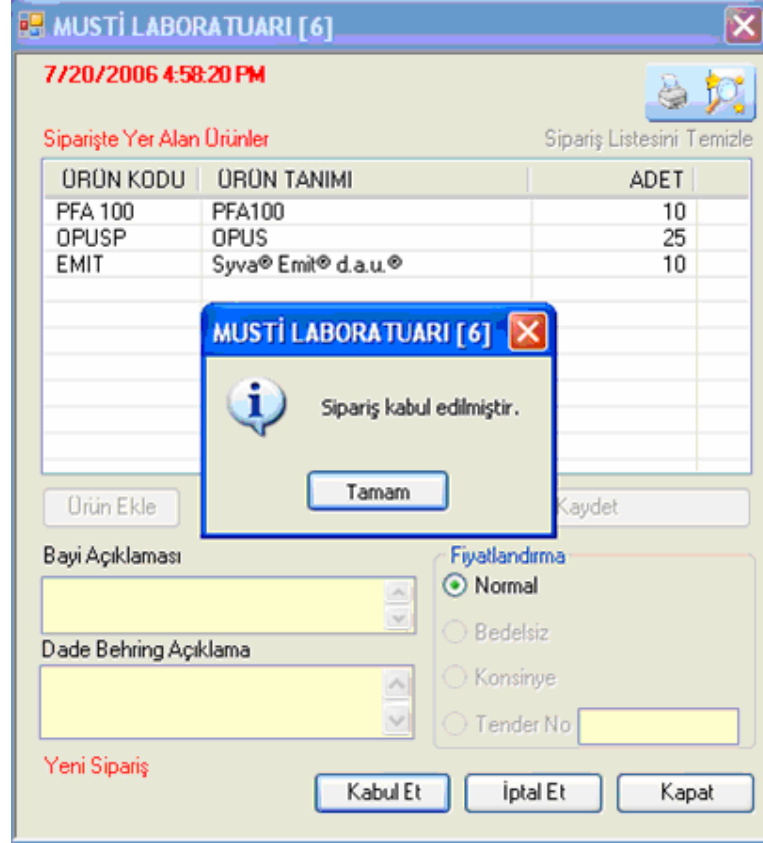
Şimdi Yönetim Ekranında, Siparişin görüntülenmesini inceleyelim.



SİPARİŞ NO	SİPARİŞ TARİHİ	BAYI ADI
6	7/20/2006 4:58:20...	MUSTI LABORATUARI

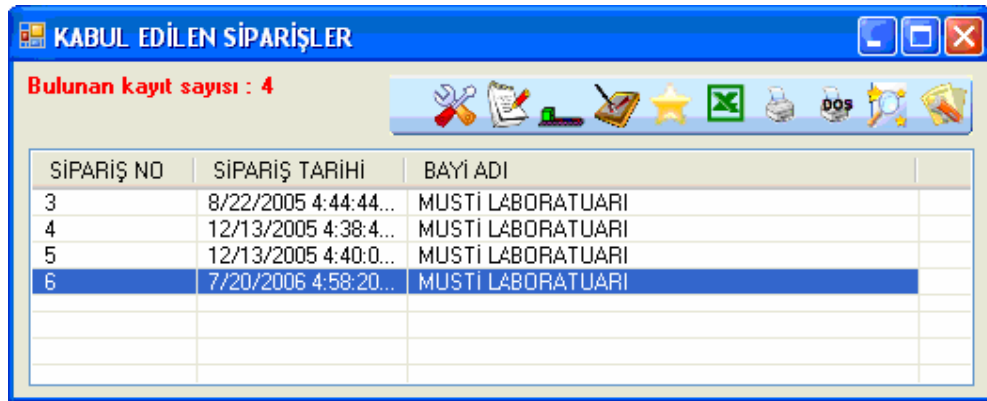
Şekil 24 - Sipariş Takip Programı Yönetici Sipariş Görüntüleme Ekranı

Yönetici görüntülediği siparişlerden onaylayacaklarını ekrana getirmek için, siparişe tıklar.



Şekil 25 - Sipariş Takip Programı Yönetici Sipariş Kabul Ekranı

Bu ekranda, yönetici eğer siparişi kabul ediyorsa *Kabul Et* isimli düğmeye tıklar. Kabul edilen siparişler hazırlanmak için depoya sevk edilir.



Şekil 26 - Sipariş Takip Programı Yönetici Onaylanacak Sipariş Seçim Ekranı

Kabul edilen siparişler depoda listelenir. Hazırlanacak olan sipariş çift tıklanarak çağırılır.

The screenshot shows a web application window titled "MUSTİ LABORATUARI [6] Sipariş Onay/İptali". The window contains the following information:

Sipariş Bilgileri

Sipariş No: 6 Sipariş Tarihi: 7/20/2006 4:58:20 PM

Müşteri: MUSTİ LABORATUARI

Fiyatlandırma: Normal

Sipariş Durum: Kabul Edilmiş.

[Açıklama](#)

ÜRÜN KO	ÜRÜN TANIMI	ADET	KALAN	KABUL	BACK ORDEF
PFA 100	PFA100	10	10	10	0
OPUSP	OPUS	25	25	25	0
EMIT	Syva® Emit® d.	10	10	10	0
*					

Buttons: Onayla, İptal Et, Kapat

Şekil 27 - Sipariş Takip Programı Yönetici Sipariş Onaylama/İptal Ekranı

İletilecek ürün adetleri girildikten sonra Onayla ya basılır böylece siparişin müşteriye sevki gerçekleştirilmiş olur.

Raporlamalarda, istenilen verilere uygun SQL cümlecikleri yazılmış ve bunlar ortak bir metot olan `GetDataTable` ile web servisinden alınmıştır. Bu fonksiyon geriye içerisinde "X" isimli bir veri tablosu içeren veri seti döndürür.

```
Public Shared Function _GetDataTable(ByVal sql As String) As Data.DataSet
```

Bu veri setinden tabloyu elde etmek için aşağıdaki yazım şekli kullanılır.

```
Dim dtDade As Data.DataTable
dtDade = GetValues._GetDataTable(str).Tables("X").Copy
```

Ardından tablo içeriği, rapor tipine göre işlenir. *Sipariş Onayı/İptali* formunda siparişin onaylanması demek siparişin, şubeye gönderilmesi ve ardından kapatılması demektir. "*İptal Et*" ise siparişin dikkate alınmadığını gösterir. Böylece sipariş adımları sonlanmış olur.

4.2 Web servisleri dışındaki çözümlerle ERP uygulamalarının karşılaştırılması

Bu tezde ürettiğimiz çözümü Web servisleri yerine farklı yöntemlerle üretebilirdik. Bu yöntemlerden biri ücretsiz olan internet ortamı yerine kiralık hatlar kullanımı yoluyla elde edilecek bir çözümdür. Fakat buradaki problemimiz kiralık hatların maliyeti ve hat üzerinde herhangi bir problem olduğunda veri hattının çökmesi, veri iletiminin aksama olasılığıdır. Diğer yandan kiralık bir hat olduğu için verileriniz güvenli kanallardan karşı tarafa iletilecektir. Fakat Ülke ya da dünya çapında 1000 bayisi ile sürekli online görüşmek iletişim kurmak isteyen bir firma için, her şube ile kiralık hat kurmak oldukça maliyetli olacaktır. Bunun yerine web servislerini kullanmak güvenilir ve maliyetsiz bir çözümdür.

İkinci olarak, web servisleri yerine asp,aspx ya da php tabanlı bir web sayfası üzerinden talepleri göndererek de çözüm üretilebilirdi. Bu web sayfaları bildiğimiz anlamda görsel sayfalar değil, bir IP üzerinde host edilen gizli sadece mesaj alıp HTML cevap gönderme yeteneğine sahip olan sayfalardır. Buradaki problem ise, veri yapınızın özelleştirilmiş olması gerekliliği ve güvenlik eksikliğidir. QueryString üzerinden yada ViewState nesnelere ile açık internet ortamında bilgi taşımak güvenli değildir bunun yanı sıra, veri yapınızı, veri gönderim mesajlarınızı kesinlikle uyguladığınız sayfaya uygun olarak tasarlamamız ve her işlem için gerekirse ayrı web sayfası tasarlamamız gerekecektir. Bu sayfalar arası karşılaşılabilecek problemlerden diğeri her sayfada geriye dönen veriyi HTML olarak yeniden formatlamamız gerektiğidir. Ayrıca dil destekleri de unutulmamalıdır. Programın genişletilebilirliği ve mobil sistemler için uygulanması mümkün değildir. Mobil uygulamalar için ayrı wap sayfaları tasarlanmalıdır.

5. SONUÇ

Bilgisayarların üretim süreçlerinde kullanılmasından sonra ortaya çıkan üretim, satış ve dağıtımdaki verileri ortak bir havuzda biriktirip işleme talebi zamanında MRP sistemlerini doğurmuş, bu sistemler evrimleşerek günümüzdeki ERP hatta ERP'den bağımsız olarak ortaya çıkan SCM ve CRM gibi sistemlerin de ERP'ye dahil edilmesiyle Genişletilmiş ERP olarak adlandırılan ERP II'nin temellerini oluşturmuşlardır .

Etkin rekabet ortamı içerisinde ERP sistemleri internet üzerinden de işlem yapabilecek yeteneklere kavuşturulduktan sonra bu bir adım öteye taşınarak, “*ERP-web services enabled* ” sistemleri oluşturmuşlardır. ERP yazılımları artık web servisleri olmadan düşünülemez. Web Servisleri, genişletilebilirliği, ve farklı fikirlerin uygulanabilirliği için sunduğu hazır ortamlarla, iş sürecinin daha etkin bir şekilde sürdürülmesine olanak tanır. Gelecekteki ERP uygulamaları, üretim veya verilen hizmetler üzerinde olmaktan çok; üretimin, servislerin ve diğer uygulamaların bir arada çalıştığı sistemler olacaktır. Bu hibrit bir yapıyı ortaya koyar. Şirket içinde çalışan üretim fonksiyonları, ve internet üzerinden elektronik olarak sunulan dağıtılmış hizmet fonksiyonları.

Bu hibrit modeller, internet üzerinden uzaktan bağlantı ile erişilebilecek şekilde yapılandırılmış lokal olarak yüklenmiş üretim fonksiyonları ile hızlı işlem yapmanızı sağlar.

ERP sistemleri son birkaç yıldır akademik çevrelerin de ilgi alanına sıklıkla girmeye başlamıştır. Akademisyenler ERP'yi çeşitli yönleriyle inceleyerek çeşitli sonuçlara varmakta ve değişik yöntemler geliştirmektedirler. Gene gelişmiş ülkelerdeki akademisyenler tarafından yapılan istatistiksel çalışmalarla da ERP sistemlerinin neler getirip neler götürdüğü yavaş yavaş tespit edilmeye çalışılmaktadır.

Bu tezde tamamen web servisleri üzerinde çalışan bir ERP yazılım örneği tasarladık. İstemciler veri tabanı üzerinde yapacakları tüm işlemlere ERP Web Servisleri ile ulaşip, gerçekleştirdiler.

Yazılımı uygularken 3 katmanlı bir yapı kullandık. Uygulama ve Web Servisi arasında gerekli güvenliđi sađlamak, gerekirse Őifreleme iŐlemleri kullanılmasına olanak sađlayacak, web servisi referanslarını tek noktadan yönetecek, web servisinin veri getirmede yetersiz kaldıđı noktalarda, kendi veri yapısını tanımlayabilecek bir ara bir sınıf yerleŐtirdik Tüm veri yönetimini -istemci bazında- bu sınıf içerisinde gerŐekleŐtirdik. Böylece yazılımsal anlamda, web servislerinin nasıl kullanılması gerektiđine alternatif bir yöntem geliŐtirme fırsatı bulduk.

KAYNAKLAR

1. Web Services Business Strategies and Architectures (ISBN 1590591798)
2. Identifying and Assessing Opportunities for the Introduction of Web Services through Business Process Modelling (University of Karlsruhe (AIFB & Research Centre Computer Science (FZI)))
3. Content-Integration for E-Business (11/4/2004 By Michael Stonebraker and Joseph M. Hellerstein for University of California, Berkeley)
4. Semantic Description of Parameters in Web Service Annotations (Jochen Gruber)
5. JClarens: A Java Framework for Developing and Deploying Web Services for Grid Computing (California Institute of Technology, Pasadena, CA, USA)
6. Cooking the Web-ERP: A practical recipe to stir-up enterprise information systems using DOC- and XML-standards 2/18/2004 by Michael H. Gillmann, Joachim Hertel, Christoph G. Jung, Gunther Kaufmann and Michael Wolber University of Southern California)
7. E-Business and Supply Chain Integration (2/19/2003 by Hau L. Lee and Seungjin Whang - Stanford University)
8. Enterprise Data Modeling (1/29/2002 by Anne Marie Smith asmith@manieri.com)
9. ERP II Making Deliver On Its Promise to the EnterPrise (exact software www.Exactsoftware.com)
10. ERP Nedir? (www.microsoft.com/turkiye/mbs/erp/erp_nedir.msp)
11. XML Web Servisleri (Mustafa Dülgerler www.yazgelistir.com)
12. ERP ile E-Business arasındaki ilişki nedir ? (http://www.diyalog.com/html/erp_ebusiness.htm)
13. Quality Attributes for Web Services: An Evaluation of Application Servers (Technical University of Vienna Reinhard Kniebeiß 8093 St. Peter a. O. Matr. Nr. 9625878)
14. Ontology-Based Web Services for Advanced Information Access (Andreas Kopetz- Technische Universität Wien)
15. Innovation and flexibility in second generation ERP Systems (Lapo Mola, Department of Economics and Marketing, IULM University, Via Carlo Bo, 1 - Milan)

16. Study and applications of the MDA Approach for the Web Service Platform
(Denivaldo Lopes- à l'UFR Sciences et Techniques, Université de Nantes,France)
17. What is SOAP (<http://www.soapuser.com/basics1.html>)
18. SOAP forms global integration
(<http://www.adtmag.com/article.aspx?id=9741>)
19. Web Services Interoperability
(<http://www.xml.com/pub/a/2002/01/30/soap.html>)
20. Microsoft Solution Developer Network Makaleleri
([http://msdn2.microsoft.com/en-us/ms978463\(msdn.10\).aspx](http://msdn2.microsoft.com/en-us/ms978463(msdn.10).aspx)) -
[http://msdn2.microsoft.com/en-us/ms978460\(msdn.10\).aspx](http://msdn2.microsoft.com/en-us/ms978460(msdn.10).aspx) -
[http://msdn2.microsoft.com/en-us/aa479079\(msdn.10\).aspx](http://msdn2.microsoft.com/en-us/aa479079(msdn.10).aspx))
21. Web Servis Kayıtçıları(www.C#Nedir.com)
22. www.Uddi.org
23. www.mobilnedir.com

EK A- KODLAMALAR**1 Reference.vb dosyasının özeti**

```

'-----
-----
' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:2. 0. 50215. 44
'
'     Changes to this file may cause incorrect behavior and will be
lost if
'     the code is regenerated.
' </auto-generated>
'-----
-----

Option Strict Off
Option Explicit On

Imports System
Imports System. ComponentModel
Imports System. Data
Imports System. Diagnostics
Imports System. Web. Services
Imports System. Web. Services. Protocols
Imports System. Xml. Serialization

'
'This source code was auto-generated by Microsoft. VSDesigner,
Version 2. 0. 50215. 44.
'
Namespace DadeWebService

    '''<remarks/>
    <System. Diagnostics. DebuggerStepThroughAttribute(), _
        System. ComponentModel. DesignerCategoryAttribute("code"), _
        System. Web. Services.
WebServiceBindingAttribute(Name:="ServiceSoap",

```

```
[Namespace]:= "http://tempuri.org/"> _
    Partial Public Class Service
        Inherits System.Web.Services.Protocols.
        SoapHttpClientProtocol

        Private _LoginAdminOperationCompleted As System.Threading.
        SendOrPostCallback

        Private useDefaultCredentialsSetExplicitly As Boolean

        '''<remarks/>
        Public Sub New()
            MyBase.New
            Me.Url = DADE_BEHRING_ORDER.Settings.Default.
            DADE_ORDER_DadeWebService_Service
            If (Me.IsLocalFileSystemWebService(Me.Url) = true)
            Then
                Me.UseDefaultCredentials = true
                Me.useDefaultCredentialsSetExplicitly = false
            Else
                Me.useDefaultCredentialsSetExplicitly = true
            End If
        End Sub

        Public Shadows Property Url() As String
            Get
                Return MyBase.Url
            End Get
            Set
                If (((Me.IsLocalFileSystemWebService(MyBase.Url) =
                true) _
                    AndAlso (Me.
                useDefaultCredentialsSetExplicitly = false)) _
                    AndAlso (Me.
                IsLocalFileSystemWebService(value) = false)) Then
                    MyBase.UseDefaultCredentials = false
                End If
                MyBase.Url = value
            End Set
        End Set
    End Class
End Namespace
```

```

End Property

Public Shadows Property UseDefaultCredentials() As Boolean
    Get
        Return MyBase. UseDefaultCredentials
    End Get
    Set
        MyBase. UseDefaultCredentials = value
        Me. useDefaultCredentialsSetExplicitly = true
    End Set
End Property

'''<remarks/>
Public          Event          _LoginAdminCompleted          As
_LoginAdminCompletedEventHandler

    <System.          Web.          Services.          Protocols.
SoapDocumentMethodAttribute("http://tempuri.          org/_TestConnection",
RequestNamespace:="http://tempuri.          org/",
ResponseNamespace:="http://tempuri.          org/", Use:=System. Web.
Services.          Description.          SoapBindingUse.          Literal,
ParameterStyle:=System.          Web.          Services.          Protocols.
SoapParameterStyle. Wrapped)> _

    '''<remarks/>
    <System.          Web.          Services.          Protocols.
SoapDocumentMethodAttribute("http://tempuri.          org/_LoginAdmin",
RequestNamespace:="http://tempuri.          org/",
ResponseNamespace:="http://tempuri.          org/", Use:=System. Web.
Services.          Description.          SoapBindingUse.          Literal,
ParameterStyle:=System.          Web.          Services.          Protocols.
SoapParameterStyle. Wrapped)> _

    Public Function _LoginAdmin(ByVal Password As String) As
Boolean

        Dim results() As Object = Me. Invoke("_LoginAdmin", New
Object() {Password})

        Return CType(results(0), Boolean)
    End Function

    '''<remarks/>

```

Kurumsal Kaynak Planlaması ve Web Servisleri

```
Public Overloads Sub _LoginAdminAsync(ByVal Password As
String)
    Me. _LoginAdminAsync(Password, Nothing)
End Sub

'''<remarks/>
Public Overloads Sub _LoginAdminAsync(ByVal Password As
String, ByVal userState As Object)
    If (Me. _LoginAdminOperationCompleted Is Nothing) Then
        Me. _LoginAdminOperationCompleted = AddressOf Me.
On_LoginAdminOperationCompleted
    End If
    Me. InvokeAsync("_LoginAdmin", New Object() {Password},
Me. _LoginAdminOperationCompleted, userState)
End Sub

Private Sub On_LoginAdminOperationCompleted(ByVal arg As
Object)
    If (Not (Me. _LoginAdminCompletedEvent) Is Nothing) Then
        Dim invokeArgs As System. Web. Services. Protocols.
InvokeCompletedEventArgs = CType(arg, System. Web. Services.
Protocols. InvokeCompletedEventArgs)
        RaiseEvent _LoginAdminCompleted(Me, New
_LoginAdminCompletedEventArgs(invokeArgs. Results, invokeArgs.
Error, invokeArgs. Cancelled, invokeArgs. UserState))
    End If
End Sub

'''<remarks/>
Public Shadows Sub CancelAsync(ByVal userState As Object)
    MyBase. CancelAsync(userState)
End Sub

Private Function IsLocalFileSystemWebService(ByVal url As
String) As Boolean
    If ((url Is Nothing) _
        OrElse (url Is String. Empty)) Then
        Return false
    End If
End Function
```

```

End If
Dim wsUri As System.Uri = New System.Uri(url)
If ((wsUri.Port >= 1024) _
    AndAlso (wsUri.Port <= 5000)) _
    AndAlso (String.Compare(wsUri.Host,
"localhost", true) = 0) Then
    Return true
End If
Return false
End Function
End Class

'''<remarks/>
Public Delegate Sub _LoginAdminCompletedEventHandler(ByVal
sender As Object, ByVal e As _LoginAdminCompletedEventArgs)

'''<remarks/>
Partial Public Class _LoginAdminCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

    Private results() As Object

    Friend Sub New(ByVal results() As Object, ByVal exception As
System.Exception, ByVal cancelled As Boolean, ByVal userState As
Object)
        MyBase.New(exception, cancelled, userState)
        Me.results = results
    End Sub

    '''<remarks/>
    Public ReadOnly Property Result() As Boolean
        Get
            Me.RaiseExceptionIfNecessary
            Return CType(Me.results(0), Boolean)
        End Get
    End Property
End Class

End Namespace

```

2 LoginAdmin metodunun SOAP Mesajı hali

```
POST /DadeBehringWebService/Service.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <_LoginAdmin xmlns="http://tempuri.org/">
      <Password>string</Password>
    </_LoginAdmin>
  </soap12:Body>
</soap12:Envelope>
```

3_LoginAdmin fonksiyonunun Service. vb dosyasındaki içeriğini

```
<WebMethod()> _
    Public Function _LoginAdmin(ByVal Password As String) As
Boolean
    Dim state As Boolean = False
    Dim cmd As New SqlCommand("", cn)
    Try
        cn. Open()
        cmd. CommandText = "SELECT COUNT (*) FROM ADMINPASS
WHERE A_PASS='" & Password & "'"
        Dim i As Short = CType(cmd. ExecuteScalar, Short)
        If i = 1 Then state = True
    Catch ex As Exception
        HATA(ex)
    Finally
        cn. Close()
    End Try
    Return state
End Function
```


4. **_LoginAdmin** Metodunun kullanımı

```
Public Shared Sub _LoginAdmin(ByVal Password As String)
    MyUser. _IsConnected = False
    Dim state As Boolean = False
    Try
        Dim x As New DadeWebService. Service
        state = x. _LoginAdmin(Password)
        If state Then
            MyUser. _IsConnected = True
            MyUser. _IsConnectedtoWeb = True
            MyUser. _ID = 0
            MyUser. _Name = "ADMINISTRATOR"
            MyUser. _IsManager = True
        End If
    Catch ex As Exception
        HATA(ex)
    End Try
End Sub
```

5 _LoginUser Fonksiyonuna ait referans.vb doyasında yer alan imzası

```

'''<remarks/>
<System.Web.Services.Protocols.SoapDocumentMethodAttribute("http://t
empuri.org/_LoginUser"
RequestNamespace:="http://tempuri.org/",
ResponseNamespace:="http://tempuri.org/"
Use:=System.Web.Services.Description.SoapBindingUse.Literal,
ParameterStyle:=System.Web.Services.Protocols.SoapParameterStyle.Wra
pped)> _
Public Function _LoginUser(ByVal UserName As String, ByVal Password
As String
As String
Dim results() As Object = Me.Invoke("_LoginUser", New Object()
{UserName
Password})
Return CType(results(0),String)
End Function
'''<remarks/>
Public Overloads Sub _LoginUserAsync(ByVal UserName As String, ByVal
Passwor
As String)
Me._LoginUserAsync(UserName, Password, Nothing)
End Sub
'''<remarks/>
Public Overloads Sub _LoginUserAsync(ByVal UserName As String, ByVal
Passwor
As String, ByVal userState As Object)
If (Me._LoginUserOperationCompleted Is Nothing) Then
Me._LoginUserOperationCompleted = AddressO
Me.On_LoginUserOperationCompleted
End If
Me.InvokeAsync("_LoginUser", New Object() {UserName, Password}
Me._LoginUserOperationCompleted, userState)
End Sub
Private Sub On_LoginUserOperationCompleted(ByVal arg As Object)
If (Not (Me._LoginUserCompletedEvent) Is Nothing) Then
Dim invokeArgs A
System.Web.Services.Protocols.InvokeCompletedEventArgs
CType(arg, System.Web.Services.Protocols.InvokeCompletedEventArgs)

```

```
RaiseEvent _LoginUserCompleted(Me, Ne  
_LoginUserCompletedEventArgs (invokeArgs.Results, invokeArgs.Error,  
invokeArgs.Cancelled  
invokeArgs.UserState))  
End If  
End Sub
```

6 _LoginUser Fonksiyonunun web servisi içerisinde yer alan tanımlaması

```
<WebMethod()> _  
Public Function _LoginUser(ByVal UserName As String, ByVal Password  
As String) As String  
Dim xStr As String = ""  
Dim cmd As New SqlCommand("", cn)  
Try  
cn.Open()  
cmd.CommandText = "SELECT (CONVERT(nvarchar(5),USER_ID) + '|' +  
USER_FULLNAME) FROM Users WHERE USER_NAME='" & UserName & "' AND  
USER_PASS='" &  
Password & "'"<br>  
xStr = CType(cmd.ExecuteScalar, String)  
Catch ex As Exception  
HATA(ex)  
Finally  
cn.Close()  
End Try  
Return xStr  
End Function
```

7 _LoginUser fonksiyonuna ait SOAP Mesajları

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /DadeBehringWebService/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/_LoginUser"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<_LoginUser xmlns="http://tempuri.org/">
<UserName>string</UserName>
<Password>string</Password>
</_LoginUser>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<_LoginUserResponse xmlns="http://tempuri.org/">
<_LoginUserResult>string</_LoginUserResult>
</_LoginUserResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to

be replaced with actual values.

```
POST /DadeBehringWebService/Service.asmx HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
<soap12:Body>
<_LoginUser xmlns="http://tempuri.org/">
<UserName>string</UserName>
<Password>string</Password>
</_LoginUser>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
<soap12:Body>
<_LoginUserResponse xmlns="http://tempuri.org/">
<_LoginUserResult>string</_LoginUserResult>
</_LoginUserResponse>
</soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to

be replaced with actual values.

```
POST /DadeBehringWebService/Service.asmx/_LoginUser HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/x-www-form-urlencoded
Content-Length: length
UserName=string&Password=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

8 - _GetValues Sınıfı

```

Public Class GetValues
Public Shared _StockCards() As
DADE_BEHRING_ORDER.DadeWebService.GeneralStockCardProperties
Public Shared _StockGroups() As
DADE_BEHRING_ORDER.DadeWebService.GeneralStockGroups
Public Shared ConnectionString As String =
"Provider=Microsoft.Jet.oledb.4.0;Data           Source="           &
Application.StartupPath &

"\DB\DadeB.mdb;Persist Security Info=False"
Public Enum OrderState
NewOrder
PrintedForPrepare
OrderAccepted
OrderRefused
SomeAccepted
SomeHasBackOrder
SomeTenderLimitOver
SomeRefused
OrderClosed
End Enum
Public Shared Function _saveOrdersAccepted(ByVal Str As String) As
Boolean
Try
Dim x As New DadeWebService.Service
Return x._saveOrdersAccepted(Str)
Catch ex As Exception
HATA(ex)
End Try
End Function
...
End Class

```


ÖZGEÇMİŞ

04 Ağustos 1980 tarihi, İstanbul İli Fatih ilçesi doğumluyum. İlk, Orta ve Liseyi yine aynı ilçede tamamladıktan sonra, İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümüne kaydoldum. Bu bölümden 2002 yılında birincilikle mezun olduktan sonra, 2004 yılında, Beykent Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği yüksek lisans eğitimine başladım.