

T.C  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
İŞLETME ANABİLİM DALI  
SAYISAL YÖNTEMLER VE YÖNETİM BİLİMİ PROGRAMI  
YÜKSEK LİSANS TEZİ

**WEB TABANLI KALİTE YÖNETİM  
SİSTEMİ UYGULAMASI**

**Kadir KIRDA**

Danışman  
**Doç. Dr. Onur ÖZVERİ**

2008

## **YEMİN METNİ**

Yüksek Lisans Tezi olarak sunduğum “Web Tabanlı Kalite Yönetim Sistemi Uygulaması” adlı çalışmanın, tarafımdan, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Tarih

...../...../.....

Kadir KIRDA

# YÜKSEK LİSANS TEZ SINAV TUTANAĞI

## Öğrencinin

Adı ve Soyadı : Kadir KIRDA  
Anabilim Dalı : İşletme  
Programı : Sayısal Yöntemler ve Yönetim Bilimi  
Tez Konusu : Web Tabanlı Kalite Yönetim Sistemi Uygulaması  
Sınav Tarihi ve Saati :

Yukarıdaki kimlik bilgileri belirtilen öğrenci Sosyal Bilimler Enstitüsü'nün ..... tarih ve ..... sayılı toplantısında oluşturulan jürimiz tarafından Lisansüstü Yönetmeliği'nin 18. maddesi gereğince yüksek lisans tez sınavına alınmıştır.

Adayın kişisel çalışmaya dayanan tezini ..... dakikalık süre içinde savunmasından sonra jüri üyelerince gerek tez konusu gerekse tezin dayanağı olan Anabilim dallarından sorulan sorulara verdiği cevaplar değerlendirilerek tezin,

BAŞARILI OLDUĞUNA	<input type="radio"/>	OY BİRLİĞİ	<input type="radio"/>
DÜZELTİLMESİNE	<input type="radio"/>	OY ÇOKUĞU	<input type="radio"/>
REDDİNE	<input type="radio"/>		

ile karar verilmiştir.

Jüri teşkil edilmediği için sınav yapılamamıştır. \*\*\*  
Öğrenci sınava gelmemiştir. \*\*

\*Bu halde adaya 3 ay süre verilir.  
\*\*Bu halde adayın kaydı silinir.  
\*\*\*Bu halde sınav için yeni bir tarih belirlenir.

Tez burs, ödül veya teşvik programlarına (Tüba, Fulbright vb) aday olabilir.	<input type="radio"/>	Evet
Tez mevcut hali ile basılabilir.	<input type="radio"/>	
Tez gözden geçirildikten sonra basılabilir.	<input type="radio"/>	
Tezin basımı gerekliliği yoktur.	<input type="radio"/>	

## JÜRİ ÜYELERİ

.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....
.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....
.....	<input type="checkbox"/>	Başarılı	<input type="checkbox"/>	Düzeltilme	<input type="checkbox"/>	Red	.....

İMZA

# ÖZET

Yüksek Lisans Tezi  
Web Tabanlı Kalite Yönetim Sistemi Uygulaması  
Kadir Kırdı

Dokuz Eylül Üniversitesi  
Sosyal Bilimler Enstitüsü  
İşletme Anabilim Dalı  
Sayısal Yöntemler Ve Yönetim Bilimi Programı

Küreselleşen dünyanın yeni şartlarına ayak uydurmak için insanlar kadar kuruluşların da çaba sarf etmesi ve yenikleri yakından takip edip, yeni teknolojilerden yararlanması gerekir. Yararlanılması gereken teknolojilerin en önemlilerinden olan bilgi ve iletişim teknolojileri, özellikle de internet, dünyanın dört bir yanını birleştirebilen, adeta dünyayı küçülten bir araçtır. İnternet sayesinde uluslar arası çalışma fırsatları doğmakta, farklı ülkedeki kuruluşların rekabeti her geçen gün artmaktadır. Kuruluşun bünyesinde bulunan teknolojiler, sadece e-posta, faks, kelime işlemci gibi kullanım amaçlarıyla sınırlı kalmamalıdır. Böyle olduğu durumlarda, kullanılmayan kaynağın israfı söz konusudur. Donanımlar, yazılım olmadan işe yaramayan cihazlardır. Donanımları anlamlı kılan, yazılımlardır. Eğer gereksinim duyulan yazılım bulunmuyorsa, alınan donanıma yatırılan para boşa gitmiş demektir.

Kuruluşların, rekabette geride kalmamak için yaptıkları girişimlerden biri, kalite yönetim sistemi standardını uygulamaktır. Kalite yönetim sistemi, süreçlere dayalı, sürekli gelişmeyi benimseyen, tüm çalışanların katılımını destekleyen bir yönetim sistemidir. Ayrıca, bilgi odaklı olması nedeniyle, yapılan bütün işlerin dokümantasyon üzerinden yürütülmesi, verilere dayalı yani gerçekçi kararlar alınmasını sağlar. Dokümantasyonun elle yapıldığı durumlarda, özellikle kalite yönetim temsilcisi olmak üzere, çalışanlara ek yük getirmektedir. Dokümanların oluşturulması, onaylanması, dağıtım listelerinin hazırlanması, revizyon talebinde bulunulması, revize edilip onaylandıktan sonra eski sürümlerin toplanıp yenilerinin dağıtılması gibi işlemlerin otomatik olarak gerçekleştirilebilmesi imkanı varken bireysel olarak yapılması sıkıntı yaratmaktadır.

Bu tezde, ISO-9001 Kalite Yönetim Standardı'na sahip bir kuruluşun kalite yönetim sistemi dokümantasyonunda yapılan incelemede ortaya çıkan gereksinimler doğrultusunda web tabanlı program tasarlanmış, uygulama örneği sunulmuştur. Bu, nesne tabanlı programlama ve buna dayalı modelleme dili açıklanarak yapılmıştır. Bu tez, aynı zamanda web tabanlı yazılım geliştirme ve Microsoft tarafından geliştirilen .NET teknolojileri hakkında da bilgiler içermektedir.

**Anahtar Kelimeler:** Kalite Yönetim Sistemi, Web Tabanlı Yazılım, Microsoft .Net Framework, İçerik Yönetim Sistemi

# **ABSTRACT**

**Master Thesis**

**The Execution of Web Based Quality Management System**

**Kadir Kırda**

**Dokuz Eylül University**

**Institute of Social Sciences**

**Department of Business**

**Numerical Methods and Management Science Program**

The organizations, also, have to do their best to follow closely the new conditions of the globalization world and make use of the new technologies like people. Information and the knowledge technologies, which must be utilized, and especially internet, are the instruments that can connect all sites of the world and make small the world as well. Owing to internet, international opportunities occur and the competition of companies from different countries increases day by day. The technologies, exist in the organization, shouldn't be limited themselves to e-mail, fax, word processor. In this situation, the waste of unused sources occurs. Hardware is not useful when software doesn't exist. If required software does not exist, the investment of hardware can be called as wrong thing.

One of the attempts of organizations is implement quality management system standard, to avoid of staying backward in the competition. Quality management system is a system that is based upon processes and consistent development, supporting full accession. Additionally, implementation of documentation of all works in the organization, secures to take good decisions which depend on real data. When documentation is done manually, this creates additional tasks to workers, especially representative of the quality management. There are opportunities of doing works automatically: like establishing acceptance, setting up distribution lists, request of revision of documents, and after acceptance, taking back of old versions, and distribute of new documents set hardships. If the organization doesn't use this automation people will be in trouble.

In this thesis, a documentation of quality management system of an organization, which has ISO-9001 Quality Management System standard, has been analyzed. A web based program has been developed according to requirements, and then one sample has been demonstrated. This execution has been introduced by explaining object oriented programming and software modeling language. And this thesis includes knowledge about web based software development and .NET technologies that Microsoft developed.

**Key Words:** Quality Management System, Web Based Software, Microsoft .Net Framework, Content Management System

# İÇİNDEKİLER

YEMİN METNİ.....	ii
YÜKSEK LİSANS TEZ SINAV TUTANAĞI.....	iii
ÖZET .....	iv
ABSTRACT.....	v
İÇİNDEKİLER .....	vi
KISALTMALAR.....	ix
ŞEKİLLER LİSTESİ .....	x
EKLER LİSTESİ .....	xi
GİRİŞ .....	1
1 NESNE TABANLI PROGRAMLAMA.....	2
1.1 Yazılım Tarihi ve Nesne Tabanlı Programlama .....	2
1.2 Sınıf ve Nesne Kavramları.....	4
1.3 Nesne Tabanlı Programlama Dillerinin Ortak Özellikleri .....	6
1.3.1 Sarmalama.....	6
1.3.2 Çok Biçimlilik.....	7
1.3.3 Kalıtım .....	7
1.4 UML (Birleşik Modelleme Dili).....	8
1.4.1 Sınıf Şemaları.....	10
1.4.2 Kullanım Durumu Şemaları .....	13
1.4.3 Durum Şemaları .....	16
1.4.4 Etkinlik Şemaları.....	19
1.5 Veritabanları.....	21
1.5.1 Veritabanı Kavramı.....	21
1.5.2 Veritabanı Yönetim Sistemleri.....	23
1.5.3 İlişkisel Veritabanı Yönetim Sistemleri.....	23
2 WEB TABANLI YAZILIM .....	26
2.1 Web Ortamı ve Web Tabanlı Yazılım .....	26
2.2 Web Geliştirmede Yazılım Mühendisliğinden Faydalanmak .....	26
2.3 Microsoft .NET Framework.....	27
2.4 ASP.NET.....	28

2.5	Yazılım Geliştirme Platformu: Visual Studio.NET .....	30
2.6	Yararlanılan Diller ve Teknolojiler .....	31
2.6.1	HTML .....	31
2.6.2	CSS.....	32
2.6.3	JavaScript .....	32
2.6.4	XML.....	33
2.6.5	C#.....	34
2.6.6	SQL ve Saklı Yordamlar .....	34
3	KALİTE YÖNETİM SİSTEMİ .....	36
3.1	Kalite İle İlgili Kavramlar .....	36
3.1.1	Kalite Kavramı .....	36
3.1.2	Kalite Kontrol .....	37
3.1.3	Kalite Güvence.....	38
3.1.4	Toplam Kalite Yönetimi .....	38
3.2	TSE EN ISO 9001:2000 Kalite Yönetim Sistemi Standardı.....	39
3.2.1	9001:2000 Standardının Temel Prensipleri.....	40
3.2.1.1	Sürekli İyileştirme.....	40
3.2.1.2	Müşteri Odaklılık .....	41
3.2.1.3	Karar Alma Mekanizmasında Gerçekçi Yaklaşım.....	43
3.2.1.4	Çalışanların Katılımı .....	44
3.2.1.5	Karşılıklı Fayda Sağlayan Tedarikçi İlişkileri .....	46
3.2.1.6	Süreç Yaklaşımı .....	47
3.2.1.7	Yönetim Sistem Yaklaşımı.....	50
3.2.1.8	Liderlik.....	51
3.2.2	9001:2000 Standardının Maddeleri.....	53
3.2.2.1	Giriş (Madde 0).....	53
3.2.2.2	Kapsam (Madde 1).....	54
3.2.2.3	Atıf Yapılan Standartlar (Madde 2) .....	55
3.2.2.4	Terimler Ve Tarifler (Madde 3) .....	55
3.2.2.5	Kalite Yönetim Sistemi (Madde 4) .....	55
3.2.2.6	Yönetim Sorumluluğu (Madde 5) .....	58
3.2.2.7	Kaynak Yönetimi (Madde 6) .....	62

3.2.2.8	Ürün Gerçekleştirme (Madde 7) .....	65
3.2.2.9	Ölçme, Analiz Ve İyileştirme (Madde 8).....	72
4	WEB TABANLI KALİTE YÖNETİM SİSTEMİ UYGULAMASI.....	75
4.1	Uygulamanın Amacı Ve Kapsamı .....	75
4.2	Gereksinimlerin Belirlenmesi .....	76
4.3	Analiz Ve Tasarım .....	77
4.3.1	Temel Yapı.....	77
4.3.2	Üyelik, Yetkilendirme ve Profil Yapılandırması .....	81
4.3.2.1	ASP.NET 2.0 Üyelik ve Profil Desteği.....	81
4.3.2.2	Üyelik ve Profil Ayarları.....	84
4.3.2.3	Kullanıcı Arabiriminin Tasarımı.....	86
4.3.3	Dokümantasyon Modülü.....	87
4.3.3.1	Kullanım Durumları ve Kullanım Durumu Şemaları.....	88
4.3.3.2	Durum ve Etkinlik Şemaları .....	92
4.3.3.3	Sınıf Şemaları.....	93
4.3.3.4	Veritabanının Tasarlanması .....	96
4.4	Tasarım ve Programlamanın Tamamlanması.....	100
	SONUÇ VE ÖNERİLER .....	104
	KAYNAKLAR .....	112
	EKLER.....	115



## KISALTMALAR

<b>B2C</b>	İngilizce açılımı "Business to Customer" olan ve "İşletmeden Tüketicie" anlamına gelen bir e-ticaret modelidir.
<b>CSS</b>	Stil Şablonları (Cascading Style Sheets)
<b>ERP</b>	Kurumsal Kaynak Planlaması
<b>HTML</b>	Zengin Metin İşaret Dili (Hyper Text Markup Language)
<b>HTTP</b>	Yardımlı Metin Aktarma İletişim Kuralı (Hypertext Transfer Protocol)
<b>IDE</b>	Bütünleşik Geliştirme Ortamı (Integrated Development Environment)
<b>ISO</b>	Uluslar arası Standardizasyon Kuruluşu (International Organization for Standardization)
<b>İVTYS</b>	İlişkisel Veritabanı Yönetim Sistemi
<b>KYS</b>	Kalite Yönetim Sistemi
<b>PUKÖ</b>	Planla, Uygula, Kontrol Et, Önlem Al
<b>SQL</b>	Yapısal Sorgulama Dili (Structured Query Language)
<b>UML</b>	Birleşik Modelleme Dili (Unified Modeling Language)
<b>VTYS</b>	Veritabanı Yönetim Sistemi
<b>VWD</b>	Görsel Web Tasarım Aracı (Visual Web Developer)
<b>W3C</b>	Dünya Çapında Ağ Birliği (World Wide Web Consortium)
<b>XML</b>	Genişletilebilir Yapılandırma Dili (eXtensible Markup Language)
<b>YBS</b>	Yönetim Bilişim Sistemi

## ŞEKİLLER LİSTESİ

Şekil 1: Nesne Tabanlı Programlamada Gerçek Nesnelerin Karşılıkları.....	5
Şekil 2: UML Sınıf Şeması Örneği.....	12
Şekil 3: UML Nesne Şeması Örneği.....	13
Şekil 4: Kullanım Durumu Örneği.....	15
Şekil 5: Kullanım Durumu Şemalarının Standart Gösterimi .....	16
Şekil 6: Bir Asansörün Durum Şeması .....	18
Şekil 7: Etkinlik Şeması Örneği.....	20
Şekil 8: İleri Düzey Etkinlik Şeması Bileşenleri.....	21
Şekil 9: İlişkisel Veritabanı Tasarımı Görüntüsü.....	25
Şekil 10: Süreç Tabanlı Kalite Yönetim Sistemi Modeli.....	50
Şekil 11: Yeni Bir Web Sitesi Projesi Oluşturma Sayfasının Ekran Görünümü .....	78
Şekil 12: Projenin Temel Klasör Yapısı.....	79
Şekil 13: Üyelikle İlgili Nesnelerin Bulunduğu Menü .....	85
Şekil 14: Web Tabanlı Kalite Yönetim Sisteminin Kullanım Durumu Şeması.....	89
Şekil 15: Yeni Doküman Ekleme Kullanım Durumu .....	90
Şekil 16: Bir Dokümanı Revize Etme Etme Kullanım Durumu.....	91
Şekil 17: Doküman Onaylama Kullanım Durumu.....	91
Şekil 18: Doküman Nesnesinin Durum Şeması.....	92
Şekil 19: Doküman Onaylama Etkinlik Şeması.....	93
Şekil 20: Dokümantasyon Modülünün Sınıf Şemaları.....	96
Şekil 21: Dokümantasyon Modülünün Veritabanı Tabloları ve İlişkiler.....	98
Şekil 22: Veritabanı Sorgu Tasarım Aracı.....	100

## **EKLER LİSTESİ**

- EK 1** KULLANICI GİRİŞ PANELİ
- EK 2** YENİ KULLANICI EKLEME PANELİ
- EK 3** YENİ KULLANICIYA PROFİL EKLEME
- EK 4** OLUŞTURULAN YENİ KULLANICININ PROFİL SAYFASI - 1
- EK 5** OLUŞTURULAN YENİ KULLANICININ PROFİL SAYFASI - 2
- EK 6** UNVANLARIN YÖNETİMİ
- EK 7** KİŞİLERE UNVANLARIN ATANMASI
- EK 8** KULLANICILARIN YÖNETİMİ
- EK 9** KATEGORİ YÖNETİMİ
- EK 10** YENİ DOKÜMAN EKLEME PANELİ - 1
- EK 11** YENİ DOKÜMAN EKLEME PANELİ - 2
- EK 12** DOKÜMAN GEZGİNİ VE BEKLEYEN GÖREVLER
- EK 13** ONAY BEKLEYEN DOKÜMANLAR SAYFASI
- EK 14** ONAY BEKLEYEN YENİ DOKÜMAN
- EK 15** ONAYLANAN DOKÜMANIN BİLGİLERİ
- EK 16** DOKÜMAN GEZGİNİNDE GÖRÜNÜM
- EK 17** REVİZYON TALEP ETME
- EK 18** BEKLEYEN REVİZYON TALEPLERİ
- EK 19** BEKLEYEN REVİZYON TALEPLERİ
- EK 20** ONAY BEKLEYEN REVİZYON
- EK 21** REVİZYONU ONAYLANMIŞ DOKÜMAN
- EK 22** REVİZYON LİSTESİ

# GİRİŞ

Kuruluşlar, her geçen zamanda artan şiddetli rekabet koşullarına ayak uydurmak zorundadırlar. Bunun için yapılması gereken en önemli şeylerden biri, bilişim çağının gerekliliklerini yerine getirmek, bir diğeri de kaliteyi sağlamaktır. Bilişim çağının gerekliliği, teknolojinin etkin kullanılması ve yeniliklerin takip edilmesidir. Kalite ise müşterilerin beklentilerini karşılayabilecek düzeyde olmalıdır.

Kuruluşlar, kaliteyi ve kalitenin sürekliliğini sağlayabilmek için ISO-9001 Kalite Yönetim Sistemi Standardı'nı uygulamak isteyebilirler. Bu standart, kaliteyi sağlamak ve sürdürmek için gerekli olan minimum gereksinimleri içerir. Çok esnek yapısı olması dolayısıyla, her türlü kuruma uygulanabilmektedir. ISO-9001'de sistematik çalışma ve gerçekçi verilere dayalı karar vermenin sağlanabilmesi için, yapılan her işe sistem mantığıyla yaklaşılır ve bütün işler dokümanlar üzerinden yürütülür.

Yapılan her işin dokümanlar üzerinden yürütülmesi, beraberinde ek yük getirmekte, özellikle de kalite yönetim temsilcileri için sıkıntı yaratmaktadır. Her bir dokümanların hazırlanması, onaylanması, dağıtılması, revizyon talep edildiğinde revize edilip onaylatılması, eski sürümlerinin geri toplanıp dağıtım listelerine göre yenilerinin dağıtılması işlemleri, rutin işlemler olup, otomasyonunun sağlanması gerektiği düşünülmektedir.

Bu tezde, kalite yönetim sistemi için program tasarımı amaçlanmıştır. Birinci bölümde yazılım modellemeyle ilgili nesne tabanlı programlama, birleşik modelleme dili ve veritabanlarına değinilmiştir. İkinci bölümde web tabanlı yazılım oluşturmak için yararlanılan teknoloji ve diller açıklanmıştır. Üçüncü bölümde kalite yönetim sistemi açıklandıktan sonra dördüncü bölümde uygulamaya geçilmiştir. Uygulamada, bir organizasyonun kalite yönetim sisteminin dokümantasyonu için gereklilikler ortaya konulmuş, buna uygun şekilde tasarım ve programlama yapılmıştır. Programın son halinde bir uygulamanın ekran görüntüleri, ekler bölümünde yer almaktadır.

# 1 NESNE TABANLI PROGRAMLAMA

## 1.1 Yazılım Tarihi ve Nesne Tabanlı Programlama

1950'lerde bilgisayar yazılımlarında öncelik arzu edilen işin yapılmasıydı. Bu da genellikle özel hesap makineleri ile yapılan ağır matematiksel işlemlerin bilgisayar ile yapılmasının sağlanmasıydı. İşler toplu olarak verilir ekrandan veya yazıcıdan çıktı alınır. Yazılımlar ürün tarzında değil de kendisi için yapılan kuruluşa özel biçimde geliştirilmekteydi. 1960'lı yıllarda özellikle askeri amaçlı ve gerçek zamanlı sistemler için yazılım geliştirilmiş ve veri tabanı yönetim sistemlerinin ilk temelleri atılmıştır (Koçer, 2007, s:8).

1970'li yıllarda UNIX'in yaygınlaşması ve bilgisayarların ucuzlamaya başlaması ile üniversiteler, araştırma kuruluşları, büyük şirketler, kamu kurumları kendi bilgi işlem birimlerini kurmaya başladılar. Bu birimlerin ana görevi var olan merkezi bilgisayarlar üzerinde çalışacak kendilerine özel programları yazmak ve merkezi bilgisayarın çalışmasını sağlamaktı. Bu nedenle, 1970'lerin programcılık çalışmasının öz nitelikleri de tamamen ihtiyaçtan dolayı, aşağıdaki gibi programların yazılmasına yönelik olarak gelişmiştir (Güngören, 2005 s:17),

- Sadece bir kullanıcısı olan,
- Güncellenmesi ve yeni özellikler eklenmesi önemsenmeyen,
- Çok az sayıda kişinin, genelde tek kişinin yazdığı,
- Yüzlerce programlama dilinden rasgele birisinde yazılan,
- Çok ciddi bellek ve işlemci gücü yetersizliğine göre yazılmış.

Bu programları geliştirirken çok az sayıda kişinin çok az sayıda kullanıcıya yönelik olarak yazılım geliştirmesi ve yazılımın kodlarının her an el altında bulunması bazı önemli esneklikler sağlamıştı. Örneğin, çeşitli program hataları bulunduğu anda düzeltilebilirdi. Çünkü yazılıma ihtiyaç duyan ve onun ne yapması gerektiğini bilen kişi ile geliştiren kişi ya aynı kişiydi ya da aynı çatı altında çalışan ve yakın iletişimde olan kişilerdi. İşin o zamanlardaki doğası gereği, bu programlar

geliştirilirken aşağıda sıralanan ölçütler sıklıkla göz ardı edildi (Güngören, 2005 s:17),

- Yazılımın kolay kullanılması,
- Proje yönetim yöntem bilimleri,
- Modüler mimariler,
- Kalite,
- Kodların yeniden kullanılması.

1970'lerin sonunda bilgisayar sektöründe yeni bir bilgisayar türü ortaya çıktı. Kişisel bilgisayarlar o kadar ucuzdu ki, insanlar evlerine alıyorlardı. Birdenbire eskiden bilgisayar almaya ve bilgisayarlar ile yapılan işleri yapmaya parası yetmeyen kalabalık bir kitlenin yazılım gereksinimi ortaya çıktı. Kitleler için yazılım yazmak yepyeni bir kavram olarak ortaya çıktı (Güngören, 2005 s:18).

Yazılımı kullanan kişilerin sayısı artıp, coğrafi olarak dağılınca; program yazan insanlar ile kullanıcılar arasında bir farklılaşma yaşandı. Artık yazılımda bir değişiklik gerektiğinde dahili telefonda birisini arayıp, derdini anlatıp, iki gün sonra da sorunu çözülen yöneticiler yoktu. Önce bunun yazılımdan kaynaklanıp kaynaklanmadığı tespit edilmeliydi. Sonra paket programı yazan kişilerden bunun bir sonraki sürümünde çözülmesi istenecekti. Bu da en erken aylar, bazen yıllar sürecek bir bekleme demekti (Güngören, 2005 s:19).

Paket programların birden fazla müşteri için yazılması asgari müşterekte anlaşmayı gerektirdi. Artık firmaların kendi özel gereksinimleri için özel programlar yerine, genele hitap eden gereksinimler için hazırlanmış paket programlar olduğundan; firmalar işlerini programlara uydurmaya çalışmaya başladı. Prensipte müşterinin kendisini programa uydurmak için değişmesi doğru bir şey değildir. Ancak pratikte bu olabilir ve 1980'lerde çokça oldu. Bir kurumun satın alabileceği en pahalı yazılım ürünlerinden birisi olan kurumsal kaynak planlama (ERP) ürünlerinde başarısızlık oranının %85'in üzerinde olduğu bu dönemi bazı müşteriler acı ile anar.

Ancak başarılı projelerin getirileri o kadar büyük oldu ki, başarısızlıklar unutuldu (Güngören, 2005 s:19).

Kitleler için paket programların yazılması gereksinimi, çoğunlukla proje gruplarının oluşturulmasını gerektirdi. Yazılım projeleri, kişilere bağlı olduğu için herhangi bir şahsi problemde projenin süresi uzayabiliyordu. Projeye yeni katılan kişilerin uyum sağlaması da zaman alıyordu. Artık yeni bir yazılım türüne ihtiyaç gitgide artıyordu. Bu yazılım dili sistem mantığıyla çalışmalı, sistemi alt sistemlere bölerek analiz ve tasarım yapabilme imkanı olmalıydı. Nesne tabanlı programlama dilleri, böyle bir ihtiyacın sonucunda ortaya çıkmıştır (Güngören, 2005). Nesne tabanlı programlama, yazılım tarihi açısından devrim niteliğinde bir buluştur. Sonraki bölümlerde, nesne tabanlı programlama kavramını temel alınacaktır.

## **1.2 Sınıf ve Nesne Kavramları**

Nesne tabanlı programlama, gerçek hayatta varolan nesnelerin yazılımdaki karşılıklarının oluşturulmasını sağlayan programlama türüdür (Şekil 1). Bu karşılıkların özellikleri ve ayrıntıları, yazılımın amacına göre değişebilir. Örneğin bir öğrenci açısından çanta, sadece eşya ve kitap taşıma amaçlı kullanılıyorken, bir çanta satıcısı açısından alınıp satılan, kar getiren bir nesnedir. Yazılım oluşturulurken çanta nesnesinin öğrenciye göre mi, satıcıya göre mi kodlanacağı belirleyicidir. Nesne tabanlı programlama, sistem anlayışına göre çalışır. Nesnelere sistem olarak alt ve üst sistemleri temsil edebilir. Örneğin bir kitap nesnesi, kütüphane nesnesinin alt sistemi, sayfa nesnesinin ise üst sistemidir. Sistem yapısı sayesinde yazılımda sistem analizi mümkün olabilmektedir.

Nesneler, onları tanımlayan ve şekillendiren kalıplara göre oluşturulur. Aynı kalıbı kullanarak aynı türde birden çok nesne oluşturulabilir. Nesnelerin oluşturulmasını sağlayan bu kalıplar, sınıflardır. Sınıflar bir kere oluşturulurken, nesnelere sınıflar sayesinde istenilen sayıda oluşturulabilir. Yani nesnelere sınıfların örneğidir. Programda bir nesne oluşturulduğunda, bellekte ona uygun miktarda yer ayrılır.

## Şekil 1: Nesne Tabanlı Programlamada Gerçek Nesnelerin Karşılıkları



Nesne tabanlı programlamada nesne ve sınıf kavramları matbaa makinesi örneğiyle daha iyi anlaşılabilir. Bir matbaa, kitap basılması için kullanılan kalıptır. Matbaa yardımı ile ders kitabı, roman, bilimsel yayın ve bunun gibi birbirinden farklı baskı yapılabilir. Örneğin yazılımda ders kitabı oluşturulmak istendiğinde; ortak özelliklerin bulunduğu bir kalıp olan “Kitap” sınıfı oluşturulur. Bu sınıf, kitapların genel özelliklerini içermelidir. Kitabın yazarı, basım yeri, tarihi gibi. Ders kitabı nesnesini oluştururken de, belirlenen bu kısımlar kendine özel bilgilerle doldurulur. Örneğin,

Yazar: “Kadir Kırda”,  
Basım Yeri: “İzmir”,  
Tarih: “2008” gibi.

İlerleyen bölümde sınıf ve nesne kavramları UML (birleşik modelleme dili) ile daha ayrıntılı şekilde açıklanacaktır.



### **1.3 Nesne Tabanlı Programlama Dillerinin Ortak Özellikleri**

Nesne tabanlı programların ortaya çıkış nedenlerine bağlı olarak, bütün nesne tabalı programlama dillerinin sahip olduğu ortak temel özellikler vardır. Nesne tabanlı programlama dillerinin kimliği niteliğini taşıyan üç ortak özellik şöyledir,

1. Sarmalama,
2. Çok biçimlilik,
3. Kalıtım.

#### **1.3.1 Sarmalama**

Sarmalama, kodun bir sistem halinde sahip olduğu öğelere erişimi denetleyen, hatalı kullanımlara karşı önleme niteliği olan ve karmaşıklığı en düşük seviyeye getiren bir özelliktir. Sarmalamada program kodu içerisinde yer alan öğelere nesnenin dışından erişilip erişilemeyeceği ayarlanabilir. Modelde oluşturulan nesnelerin çeşitli öğeleri vardır. Bu öğelerden bazıları değişkenlerdir. Bir nesnenin sahip olduğu değişkenlerin bir kısmının sadece nesnenin sahip olduğu program kodu tarafından değiştirilmesi gereklidir. Örneğin 11 karakterlik TC kimlik numarasının, ilgili metotlar ile doğrulanıp veritabanına kaydedilmesi gerektiği durumda, programcıya değişkene doğrudan erişim izni verildiyse, programcı iyi niyetli de olsa ilgili veriye doğrudan müdahale ettiğinde ileride tahmin edilemeyecek sorunlar ortaya çıkabilir.

Sarmalamada nesnenin öğelerine erişim public ve private gibi erişimci ifadeleri ile sağlanabilir. Public ile belirtilen öğeler dışarıdan erişilebilir, private ise öğenin sadece sınıf üyeleri tarafından erişilmesini mümkün kılar. Sarmalamanın en önemli özelliği, nesneyi kullanan programcının müdahale etmesi gerekmeyen, yani nesnenin kendi içinde bulunan programa özgü ayrıntıları görmemesidir. Bununla ilgili, cep telefonlarının çalışma tekniği örnek olarak gösterilebilir. Cep telefonu yazılımlarında yüzlerce veya binlerce kod vardır. Bu kodların çoğu, diğer fonksiyonların çalışmasını sağlamak için mevcuttur. Telefonu kullanan kişilerin bu

kodları bilmesinin gereği yoktur. Bu nedenle, erişilmesi istenmeyen kodlar gizlenerek hem karmaşa hem de olası hatalar engellenmiş olur.

### 1.3.2 Çok Biçimlilik

Çok biçimlilik, genel bir etkinlikler sınıfına erişmek için bir arabirime imkan veren bir özelliktir (Schilt, 2006, s:13). Çok biçimlilik, bir nesnenin gerçekleştirmesi istenen eylemleri uygulama garantisi vermesi demektir. Bunu arabirimler vasıtasıyla gerçekleştirir. Örneğin, C# programlama dilinde yerleşik olarak bulunan, varolan diziyi sıralamaya yarayan “Sort” fonksiyonu tamsayı, metin gibi temel değişkenlerde herhangi bir ön işlem yapmaksızın kullanılabilir. Ancak yeni bir nesne tanımlandığında ve bu nesnelerin oluşturduğu dizinin sıralanması istendiğinde, neye göre sıralama yapılacağı tanımlanmadığında program hata verecektir. Bu fonksiyonu uygulayabilmek için, tanımlanan sınıfa “IComparable” adında bir arabirim tanımlanması gerekir. Bu arabirim, sınıf içerisinde sınıfın iki üyesinin karşılanması gerektiğinde hangisinin daha büyük, hangisinin daha küçük olduğuna karar vermesini sağlayan “CompareTo” fonksiyonunu tanımlar. Söz konusu arabirimi ve dolayısıyla fonksiyonu uygulayan nesne türünden oluşan dizide böylece sıralanabilir hale gelmektedir. Çok biçimlilik sayesinde, birden çok arabirimi kullanan nesnelere, sayısız işlevi yerine getirebilir. Bunun sonucunda da programın esnekliği ve güvenilirliği artmış bulunmaktadır.

### 1.3.3 Kalıtım

Kalıtım, bir nesnenin diğer bir nesnenin özelliklerini sahiplenebilmesini sağlayan bir yöntemdir. Bu önemlidir, çünkü kalıtım, hiyerarşik sınıflandırma kavramını destekler. Bilgilerin büyük bölümü hiyerarşik (yani, yukarıdan aşağıya doğru) sınıflandırma sayesinde yönetilebilir kılınır. Örneğin, kırmızı tatlı elma, elma sınıflandırmasının bir parçasıdır. Elma, meyve sınıfının bir parçasıdır. Meyve ise daha büyük bir sınıf olan yiyecek sınıfının altında yer alır. Yani, yiyecek sınıfı, aynı zamanda mantıksal olarak kendisinin bir alt sınıfı olan meyve sınıfına da uygulanabilecek belirli özellikleri (yenilebilir, besleyici vs.) sahiptir. Bu özellikleri

ek olarak, meyve sınıfının da kendisini diğer sınıflardan ayırt eden, kendisine özgü özellikleri (sulu, tatlı vs.) vardır. Elma sınıfı bu tür özelliklerden elmaya özgü olanları (ağaçta yetişir, tropik meyve değildir vs.) tanımlar. Kırmızı Tatlı elma, bu durumda, kendisinden önce gelen tüm sınıfların tüm özelliklerini kalıtım yoluyla devralır ve bu özelliklerden sadece kendisini eşsiz kılanları tanımlar (Schilt, 2006, s:14).

Kalıtım kullanılsaydı, her nesne kendi özelliklerinin tümünü açıkça tanımlamak zorunda kalacaktı. Kalıtım sayesinde bir nesnenin kendi sınıfı içinde sadece kendisini eşsiz kılan özellikleri tanımlaması yeterlidir. Nesne, genel özelliklerini atalarından kalıtım yoluyla devralabilir. Böylece, bir nesnenin daha genel bir durumun spesifik bir örneği olmasını mümkün kılan, kalıtım mekanizmasıdır (Schilt, 2006, s:14).

#### **1.4 UML (Birleşik Modelleme Dili)**

Yazılım teknolojisindeki gelişmeler, nesne tabanlı programlama dillerinin ortaya çıkışı ve büyük çaplı projelerde birden çok kişinin birlikte çalışmak zorunda olması, birçok zorluğu da beraberinde getirmiştir. İnternet'in aradaki mesafeleri ortadan kaldırmasıyla, birçok yazılım projesinin birbirinden uzak mesafelerde bulunan kişiler tarafından oluşturulma ihtiyacı ortaya çıkmıştır. Bunun dışında, tasarlanan programların devasa hacimli olması, aynı anda onlarca ve hatta yüzlerce kişi tarafından yazılması, yazılım sektöründe herkesin aynı şekilde anlayacağı bir modelleme dili ihtiyacını iyice artırmıştır. "Unified Modelling Language" (Birleşik Modelleme Dili) kelimelerinin baş harflerinden oluşan UML, böyle bir ihtiyacın sonucunda oluşturulmuştur.

1990'lı yıllarda "Rational Software Corporation" isimli şirkette çalışan üç kişi tarafından, bu sorunlara bir çözüm getirmek üzere UML modelleme dili geliştirildi. Kısa bir süre sonra (17 Kasım 1997'de) UML, bir komite tarafından standardize edildi (Taşdelen, 2005, s:2).

Yazılım oluřturmada programlamadan önceki ařama olan tasarım ařamasında ne kadar doęru modelleme yapılırsa, daha sonra herhangi bir sorun çıkması olasılıęı o kadar azalır. Doęru modellemeyi yapabilmek için harcanan zaman ilk bařta dezavantaj gibi görünse de ileride çıkabilecek sorunları bertaraf ettięi için en büyük getirisi, yazılımın başarılı bir řekilde sonuçlanmasıdır. Başarısız sonuçlanan yazılım projelerinin altında yatan en önemli neden, programlamaya geçilmeden önce yetersiz ön çalışma yapılması veya hiç yapılmamasıdır.

UML ile hazırlanmış bir yazılım hem daha az maliyetli hem daha etkili ve daha uzun ömürlü olur. UML ile dokümantasyonu yapılmış bir programın sonradan düzenlenmesi daha kolay olur. UML'nin faydalarını maddeler halinde sıralanacak olursa (<http://www.csharpnedir.com/makalegoster.asp?MIId=6>, 17/06/08),

1. Öncelikle programın kodlanmaya başlamadan önce geniş bir analizi ve tasarımı yapılmış olacaęından kodlama işleminin daha kolay olur. Çünkü programdan ne beklendięi ve programlama ile neler yapılacaęı profesyonel bir řekilde belirlenebilir.
2. Programda beklenmedik bir takım mantıksal hatalar en aza indirgenebilir.
3. Tasarım ařaması düzgün yapıldıysa tekrar kullanılabilen kodların sayısı artacaktır. Bu da program geliştirme maliyetini büyük ölçüde düşürecektir.
4. Programımızın kararlılıęı artacaktır. UML ile dokümente edilmiş kodları düzenlemek daha az zaman alacaktır.
5. Ortak çalışılan projelerde programcıların iletişimi daha kolay hale gelir. Çünkü UML ile program parçalarına ayrılır ve parçalar arasında ilişkiler kurulur.

UML'de modelleme yapılabilmesi için yararlanılan bileşenler řunlardır,

1. Sınıf Şemaları
2. Kullanım Durumu Şemaları
3. Durum Şemaları
4. Etkinlik Şemaları

5. İşbirliği Şemaları
6. Paket Şemaları
7. Bileşen Şemaları
8. Dağıtım Şemaları

Uygulama kısmında yararlanılacak olan ilk dört bileşen ayrıntılarıyla açıklanacaktır.

#### **1.4.1 Sınıf Şemaları**

Sınıf şemaları; sistemin statik yapısını ortaya çıkartmak amacı ile tasarlanırlar. Sınıf şemalarının içerdiği unsurlar; sınıflar ve aralarındaki ilişkileri temsil eden, şekilleri standardize edilmiş simgelerdir. Sınıflar; yatay çizgilerle üçe ayrılmış dikdörtgen simgesi ile temsil edilirler (Taşdelen, 2005, s:13).

Dikdörtgenin en üst kısmında sınıfın ismi yazılır. İkinci kısımda sınıfın özellikleri yer alır. Özellikler, sınıfın değişkenleridir. Örneğin “Gezegen” sınıfının bir nesnesi olan “Dünya”, çap, dönüş hızı, çekim kuvveti gibi “Gezegen” sınıfında tanımlanan özelliklere sahip olabilir. Dikdörtgenin en alt bölümünde ise işlevler yer alır. İşlevler, sınıfın oluşturduğu nesnelerin gerçekleştirebileceği faaliyetleri temsil eder. Bu faaliyetler çeşitli değişkenleri parametre olarak alabilirler. İşlevlere “cep telefonu” sınıfı örnek gösterilebilir. Bu sınıftan türemiş nesnelerin modeli ve kendine has özellikleri ne olursa olsun, sınıfın işlevlerini yerine getirmesi gerekmektedir. Herhangi bir cep telefonundan beklenen en temel işlevler, arama yapabilmesi, mesaj gönderip alabilmesi, arama yapıldığı zaman sesli, titreşimli veya görüntülü uyarı vermesidir. Cep telefonu sınıfından oluşturulan nesneler, boy, renk, marka, model olarak birbirinden farklı olsa da, belirlenen temel işlevleri yerine getirmek zorundadırlar.

Bir sınıfın birden fazla özelliği olabileceği gibi, hiç özelliği de bulunmayabilir. Aynı şekilde, birden çok işlevi olacağı gibi, hiç işlevi olmayabilir. Sınıfın işlevi veya özellikleri bulunmuyorsa ilgili bölüm boş bırakılır. Ancak bu

şemaların daha farklı bir kullanımı daha vardır. Anlatımı yapılan sistemde şemanın içerisindeki bölümlerden bazılarının görünmesi çok önemli değilse, o bölüme “...” (üç nokta) konulabilir. Üç nokta görülen yerlerde anlaşılması gerekir ki, bu bölümde özellikler (veya işlevler) var fakat ilgili gösterimde gerek duyulmadığı için belirtilmemiştir.

Özellik ve işlevlerde kullanılan +, - ve # sembolleri vardır. Bu semboller, verilerin erişilebilirliğinin belirlenmesini sağlar. Eğer bir ögenin solunda + işareti varsa, sınıf dışarisından erişim gerçekleştirilebilir. “-” işareti, ilgili üyenin sınıf dışından veya türetilmiş sınıflardan erişilemeyeceğini ifade eder. “#” (diyez) görülen yerlerde ise, ilgili sınıftan yeni bir sınıf türetilirse, bu üyeye erişilebileceğini belirtir. Bu sembollerin programlamadaki karşılıkları da kullanabilir. Bu sembollerin ifade karşılıkları şöyledir,

<u>Sembol</u>	<u>İfade</u>
+	public
-	private
#	protected

Sınıf şemalarında yer alan bir diğer öge, değişken türüdür. Değişken türleri verilerin oluşturulma amaçlarına uyumlu olmasını sağlar. Örneğin mesafe değişkeni tamsayı türünde olması gerekirken, varılacak hedefin adı, metin türündedir. Türlerin belirlenmesinin en önemli faydalarından biri, bellekte değişken türüne göre yer tutulacağından, bellek israfını önler. Bir diğer önemli faydası ise, hatalı işlemlerden korumaktır. Örneğin, mesafe değişkeninde saklanan değer, kilometreden mile dönüştürülmek istendiğinde, bu değişkenin tamsayı olduğunun garanti edilmesi zorunludur. En sık kullanılan değişken türleri şunlardır,

- string = Metin,
- integer(int) = Tamsayı,
- datetime = Zaman birimi,
- boolean = İkili (true/false).

Bu bilgilere dayanarak, özelliklerin ve işlevlerin gösterim standardı aşağıdaki biçimde gösterilebilir,

### Özellik

<b><u>Erişilebilirlik</u></b>	<b><u>Değişken İsmi</u></b>	<b>:</b>	<b><u>Tür İsmi</u></b>	<b><u>[=ilk değer]</u></b>
+ (veya public)	AdiSoyadi	:	string	(seçeneğe bağlı)

### İşlev

<b><u>Erişilebilirlik</u></b>	<b><u>Metot İsmi</u></b>	<b><u>([Parametre Listesi])</u></b>	<b>:</b>	<b><u>Geri Dönüş Değer Türü</u></b>
+ (veya public)	ParaCekme	()	:	int

Özelliklerde istendiği takdirde ilk değerın ataması “=” işareti ile yapılabilir. İlk değerin atanması zorunlu değildir. Ancak bazı durumlarda ilk değerin atanması işlerliği artırmakta, bazı durumlarda da olası hataları önleyebilmektedir.

İşlevlerde parametre listesi bölümü boş olabileceği gibi, çeşitli türlerde birden çok parametre ve yazılımcı tarafından oluşturulan nesnelere, parametre olarak yer alabilir. İşlevlerin herhangi bir değeri döndürmesi beklenmediği durumlarda, ilgili kısma “void” yazılır. Örneğin sitenin ziyaretçilerini sayan “SayiArttir()” metodunun herhangi bir değeri döndürmesi beklenmeyebilir. Bu durumda geri dönüş değeri türü kısmı “void” olmalıdır. Bu bilgilere dayanarak, “Calisan” isimli bir sınıf oluşturmak istendiğinde Şekil 2’deki gibi bir şema elde edilir.

### Şekil 2: UML Sınıf Şeması Örneği

<b>Calisan</b>
<b>+ AdiSoyadi : string</b> <b># TelNo : int</b> <b>- Maas : int</b>
<b>+ Calisma() : void</b> <b>+ MaasCekme() : int</b>

Nesne şemaları, sınıf şemalarına benzer ancak isimlerinin altı çizilidir ve değerlerin atanması “=” ile yapılmıştır. Sınıf isminin yanında “:” işaretinden sonra örneği alınan sınıfın adı yazılır. Çalışan isimli sınıftan bir nesne türettiğimizde Şekil 3’deki gibi olacaktır.

### Şekil 3: UML Nesne Şeması Örneği

<b><u>Kadir:Çalışan</u></b>
<b>AdiSoyadi : string = “Kadir Kırdı”</b>
<b>TelNo : int = 5555555</b>
<b>Maas : int = 1000</b>

#### 1.4.2 Kullanım Durumu Şemaları

Bir kullanım durumu bir sistemin, sistem dışındaki bir aktörün gözlemleyebileceği bir faydası olan bir sonuç verecek şekilde çalıştığı bir dizi eylemi ve varsa bu eylemdeki çeşitli değişik akışları tanımlayan bir belgedir (Güngören, 2005, s:117). Kullanım durumlarının belirlenmesinde kullanılacak yöntem; "Sistemde yapılan işlemler nelerdir?" sorusunun sorulması ve bu soruya uygun cevabın bulunmasıdır (Taşdelen, 2005, s:3).

Kullanım durumları ile beraber, sistemin kullanıcılarının kimler veya neler olduğunun belirlenmesi gerekir. Bu durumda sistemin aktörleri ortaya çıkar. Aktörler, sistemin içinde yer almayan, ancak sistemde bir değişikliğin meydana gelmesine neden olan kişi, donanım veya başka bir yazılım olabilir. Aktörler kullanım durumlarını harekete geçirirler.

Aktörler genellikle rollere göre tanımlanmalıdır. Kişi adı yerine rollere göre oluşturulan sistem, bireylerin değişmesine göre yeniden düzenlenmek zorunda kalmaz. Bir kişi birden çok role de sahip olabilir. Örneğin, bir mağaza çalışanı, alışveriş yapmak istediğinde çalıştığı mağazanın müşterisi olabilir. Roller ve kullanıcılar kim ve ne olursa olsun, aktörler açısından değişmeyen tek kural, sistemin dışarısında bulunmasıdır.



Kullanım durumunun özellikleri aşağıdaki gibi sıralanabilir (Güngören, 2005 s:117),

- Kullanım durumu, sadece ve sadece bir tanımdır. Herhangi bir notasyon ya da programlama diline ait bir anlatım değildir. Sadece metin olabileceği gibi UML şemaları ve tablolarla desteklenebilir de. Ama esas hitap ettiği kitle her zaman için insanlardır. Ayrıca bu tanımın hemen herkesin anlayacağı derecede basit bir dilde yazılmış olması gerekir.
- Bir kullanım durumu, bir aktör için, o aktör tarafından gözlemlenebilir bir faydanın elde edildiği bir dizi olayı anlatır. Yani kullanım durumunun hitap ettiği bir ana aktör ve bu kullanım durumu sonunda bu ana aktörün elde ettiği bir fayda (sonuç) olmalıdır. Bunların olmadığı ya da açık olarak belirlenmediği herhangi bir belgeleme, bir kullanım durumu sayılmaz.
- Her koşulda kullanım durumları ardışık olarak olan olayları, etkinlikleri listeler. Yani bir kullanım durumundaki adımlar takip edilerek bir sonuca ulaşılabılır,
- Ayrıca birden fazla akış da olabilir. Bu akışların her biri farklı sonuçlar verebileceği gibi aynı sonuca daha farklı şekillerde ulaşılmasını da sağlayabilirler.

Kullanım durumu şemaları, modellenmeye çalışılan sistemin davranış ve özelliklerini, görsel olarak ifade etmeye yarayan çizimlerdir. Kullanım durumu şemaları; aktörler, kullanım durumları ve bu unsurlar arasında tanımlı ilişkilerden yola çıkılarak oluşturulurlar (Taşdelen, 2005, s:5).

Kullanım durumu ile kullanım durumu şemaları birbirinden farklıdır. Kullanım durumları sistemde gerçekleştirilen eylemin basamaklar halinde akışını gösterir. Kullanım durumlarıyla ilgili örnek Şekil 4'te görülebilir. Şekildeki kullanım durumu, aktör, kapsam, adımlar, alternatif akışlar ve akış sonucunu bilgilerini içermektedir. Alternatif akışlar, diğer kullanım durumlarına atıf şeklinde olabileceği gibi, çok kapsamlı olmadığı durumlarda aynı kullanım durumu içerisinde belirtilebilir. Oluşturulan kullanım durumları, kullanım durumu şemalarında elips

şeklinde gösterilir. Eylemi gerçekleştiren aktör ise sistem sınırları dışında kalan çöp adamlar olarak çizilir. Eylemi gerçekleştiren aktörlerden, ilgili kullanım durumlarına doğru giden bir ok, hangi aktörün hangi kullanım durumuyla ilgili olduğunun açıkça görülebilmesini sağlar.

#### Şekil 4: Kullanım Durumu Örneği

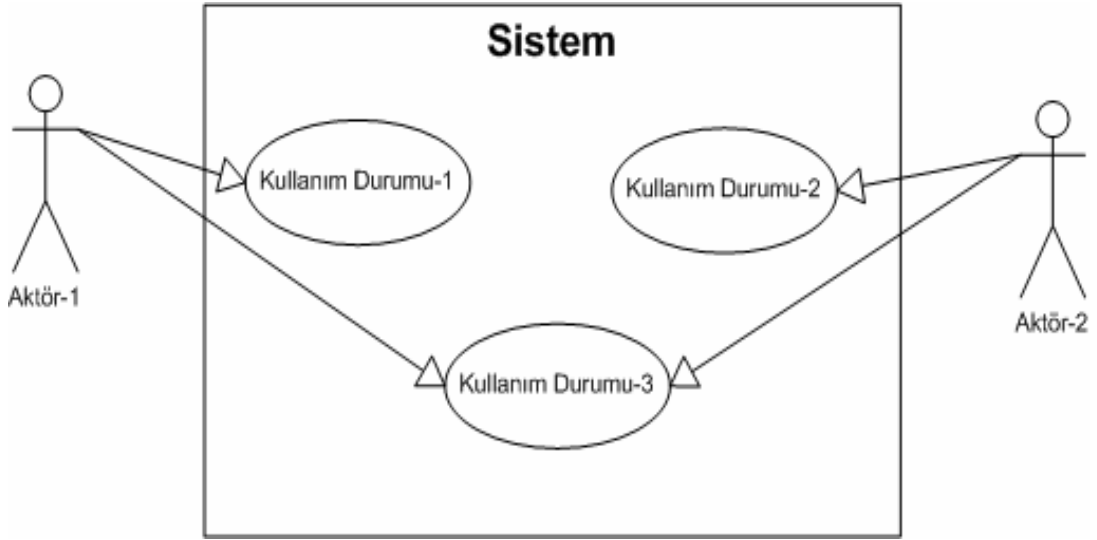
<b>Kullanım Durumu: 108</b> Müşteri Bankamatikten Para Çeker	
Aktör	<b>Herhangi bir Müşteri (M) tipi</b>
Kapsam	Müşteri, bankamatığı kullanarak hesabından bir miktar para çekmek ister
Adımlar	Müşteri bankamatik cihazına bankamatik kartını sokar Sistem müşterinin müşteri numarasını karttan okur ve müşterinin ad soyadını merkezden edinir. Ardından adını ve soyadını göstererek şifresini sorar. Müşteri şifresini girer. Sistem, Müşterinin şifresini Merkezden onaylar ve menüyü gösterir. Müşteri menüden para çekmeyi seçer. Sistem Müşterinin hesaplarını listeler ve para çekmek için bir hesap seçmesini ister. Müşteri bir hesabı seçer. Sistem müşterinin çekmek istediği miktarı sorar. Müşteri çekilmesini istediği miktarı girer. Sistem parayı hesaptan düşer. Bunu merkeze kaydeder. Ardından parayı teslim eder. Müşteri parasını, bankamatik kartını ve dekontu alır.
Alternatif Akış	Müşteri şifresini yanlış girer. (Kullanım Durumu: 125) Müşterinin hesabında yeterli para yoktur. (Kullanım Durumu: 126) Müşterinin istediği miktarda para bankamatik tarafından verilemiyordur. (Kullanım Durumu: 127)
Sonuç	Müşteri para çekmiştir. Müşterinin para çektiği miktar hesabından düşülmüştür.

**Kaynak: Güngören, 2005, s:120**

Kullanım durumları hazırlanıp şemaları oluşturulduktan sonra, kullanım durumlarında yer alan öğelerden sınıf tasarımında yararlanır. Kullanılan isimler ve fiiller, programcı için sınıfın başlangıç noktasını oluşturur. Bu şekilde tasarlanan bir sınıfın hata payı düşük olacaktır.

Kullanım durumlarında tanımlanmış isimler sınıfların özelliklerini, fiiller ise işlevlerin oluşturulmasında katkıda bulunur. Örneğin “Müşteri bankamatikten para çeker.” cümlesinde “Müşteri”, “Bankamatik” ve “Para” isimleri birer sınıf, “Çekme” eylemi de işlev için çıkış noktası olacaktır (Güngören, 2005).

**Şekil 5: Kullanım Durumu Şemalarının Standart Gösterimi**



### 1.4.3 Durum Şemaları

Durum şemaları, bir nesnenin ömrü boyunca sahip olacağı durumları belirleyen şemalardır. Durum şemalarında durumlar dışında, durumlar arasında geçişe neden olan olaylar ve şartlar da yer alır. Bir nesnenin, çeşitli nedenlerle farklı durumlarda bulunması gerekebilir. Bu, belirli bir zaman aralığında olmak zorunda değildir. Durum şemalarında zaman dikkate alınmaz. Yalnızca nesnenin sahip olacağı durumlarla ilgilidir. Bir çamaşır makinesinin durumları, durum şemasına örnek olarak gösterilebilir. Çamaşır makinesinin çeşitli durumları sayılacak olursa,

- Başlangıç durumu (makine duruyor),
- Deterjanla yıkama durumu,
- Durulama durumu,
- Sıkma durumu.

Durumlar arasındaki geiř, bazı řartların yerine getirilmesi ile gerekleřebilir. rneęin makinenin sıkma durumuna gemesi, zamanlama ayarının ilgili noktaya gelmesiyle veya dıřarıdan bir mdahale ile ilgili noktaya getirilmesiyle gerekleřebilir.

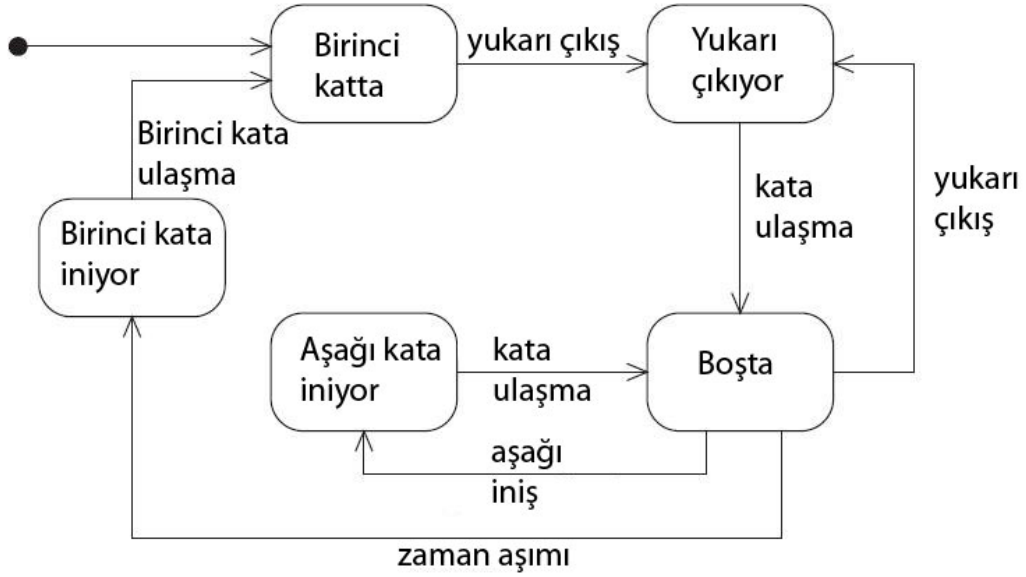
Durum řemalarının temel kavramları řoyledir (<http://www.csharpnedir.com/makalegoster.asp?MId=86>, 17/06/2008; Tařdelen, 2005),

- State (Durum): İlgili nesnenin belirli bir zamandaki durumunu ifade eder. Křeleri yuvarlatılmıř drtgenler řeklinde ifade edilir. Drtgenin iine durumun adı yazılır.
- Transition (Geiř): Bir durumdan dięerine gemeyi tanımlar. Geiři gstermek iin iki durum grafięi arasında nceki durumdan sonraki duruma doęru bir ok izilir.
- Event (Olay): İki durum arasında geiře neden olan řeydir. Televizyonun kapalı durumundan aık durumuna geiři iin dęmesine basılması bir olaydır. Olaylar, geiři ifade eden okların yanlarına yazılır.
- Action (Eylem): İlgili nesnenin bir durumdan dięer bir duruma getięinde gerekleřtirdięi iřlere eylem denir.
- Initial State (İlk Durum): Nesnenin yařam dngsndeki ilk durumunu belirtir. İi dolu yuvarlak ile gsterilir.
- Final State (Son Durum): Nesnenin yařam dngsndeki son durumu gstermek iin kullanılan semboldr. ember iinde ii dolu yuvarlak ile gsterilir.
- Substate (Alt durum): Sistemin sahip olduęu durumlar, bazen alt durumları da ierebilir.

Durum řemalarıyla ilgili olarak belirtilmesi gereken en nemli řey, řemanın sadece bir nesneye iliřkin olduęudur. řemada bir nesnenin btn durumları yer almaktadır. řekil 6'da bir asansrn durumları gsterilmektedir. İlk durum, asansrn birinci katta olduęu durumdur. Yukarı ıkıř eylemi tetiklendięinde, asansrn yukarı ıkma durumu ortaya ıkmaktadır. Kata ulařtıęında asansrn

duruyor olduğunu ifade eden boşta durumu ortaya çıkmaktadır. Bu durumda üç eylem gerçekleşebilir. Zaman aşımı nedeniyle tekrar birinci kata dönmesi, aşağı ve yukarı katlardan birine gitmesi komutu verilmesi sonucu ortaya çıkan durumlardır.

**Şekil 6: Bir Asansörün Durum Şeması**



**Kaynak: Eriksson, Penker, Lyons, David, 2004, s:27**

Durum şemalarının kullanılmasının birçok yararları vardır. Bu yararlar şöyle sıralanabilir (Güngören, 2005, s:128),

- Var olan durumların sistemi ifade etmekte yeterli olup olmadığı kolayca tartışılabilir. Acaba Bir talep için olan durumlar burada yeterli midir? Örneğin teklif istendiği zaman tedarikçi elinde mal olmadığını belirtirse ne olacaktır? Talebin durumu değişecek midir? Yoksa bir şekilde aynı duruma geri mi dönecektir?
- İkinci olarak iki durum arasında geçişi bir etkinliğe bağlamakta zorlanılabilir. Bir nesne bir durumdan diğerine geçmişse bunun bir nedeni olmalıdır. Bir varlığı durumunda değişiklik olması için, o varlığa etki eden bir kuvvetin olması gerekir.

- Bazen bir duruma geçilmesine neden olan etkinlik ile o durumdan çıkılmasına neden olan etkinlik karıştırılabilir. UML durum şeması bunun takip edilmesi için de çok yararlı bir araçtır.
- Zaman zaman bazı durumlara aynı etkinliğin sonucunda ulaşılabilir ya da durumlar çok yakın biçimde tanımlanabilirler. UML durum şemalarının ileri bir özelliği olan bileşke durum ve alt durum tanımları ile bu durumlar yönetilebilir.
- Bazen iki durum arasındaki geçiş bir etkinlik dışında bir koşulun sağlanmasına da bağlıdır. Geçişlerde koşullara yer vererek ihtiyaç duyulan gösterim yapılabilir.

#### 1.4.4 Etkinlik Şemaları

Durum şemalarının bir alt kümesi olarak da nitelenen etkinlik şemaları genelde iş akışlarının grafiksel gösteriminde kullanılır. Örneğin aşağıda bir B2C\* uygulamasına ait sipariş alımında kullanılan senaryo görülmektedir (Taşdelen, 2005, s:23),

- Müşteri sipariş vermek için gerekli bağlantıya tıklar,
- Seçilen kategori altındaki ürünler ekranda listelenir,
- Müşteri sipariş etmek istediği ürünü seçer,
- Uygulama seçilen ürünle ilgili stok bilgilerini veritabanından alır. Stoktaki miktar sıfırdan farklı ise ürünle ilgili bilgiler müşteriye ekranda gösterilir ve müşteriden sipariş adedi girmesi istenir,
- Aksi taktirde müşteriye ürünün elde kalmadığına ilişkin bilgi verilir,
- Uygulama sipariş edilen ürünün fiyatı ile sipariş adedini çarparak toplam fiyatı hesaplar,
- Müşteriden isim, adres, ödeme türü gibi bilgiler istenir. Ödeme türünü kredi kartı olarak seçen müşteriden kart bilgileri istenir,

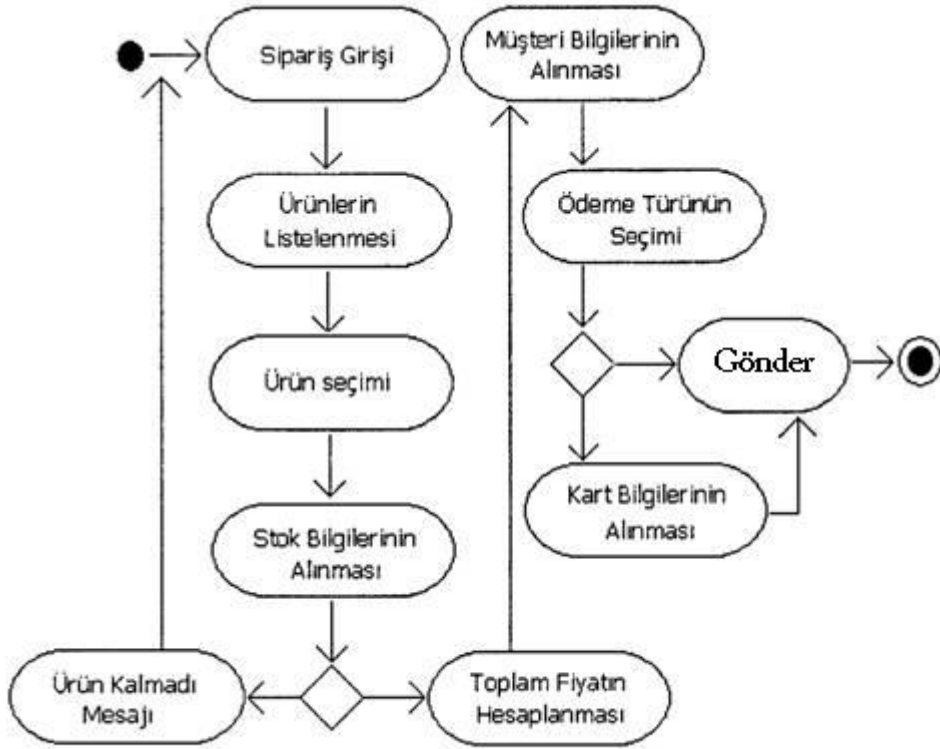
---

\* İngilizce açılımı "Business to Customer" olan ve "İşletmeden Tüketicie" anlamına gelen bir e-ticaret modelidir.

- Müşteri gerekli bilgileri doldurarak formu ibraz eder.

Bu senaryoya ilişkin iş akışı ve etkinlik şeması Şekil 7’deki gibidir.

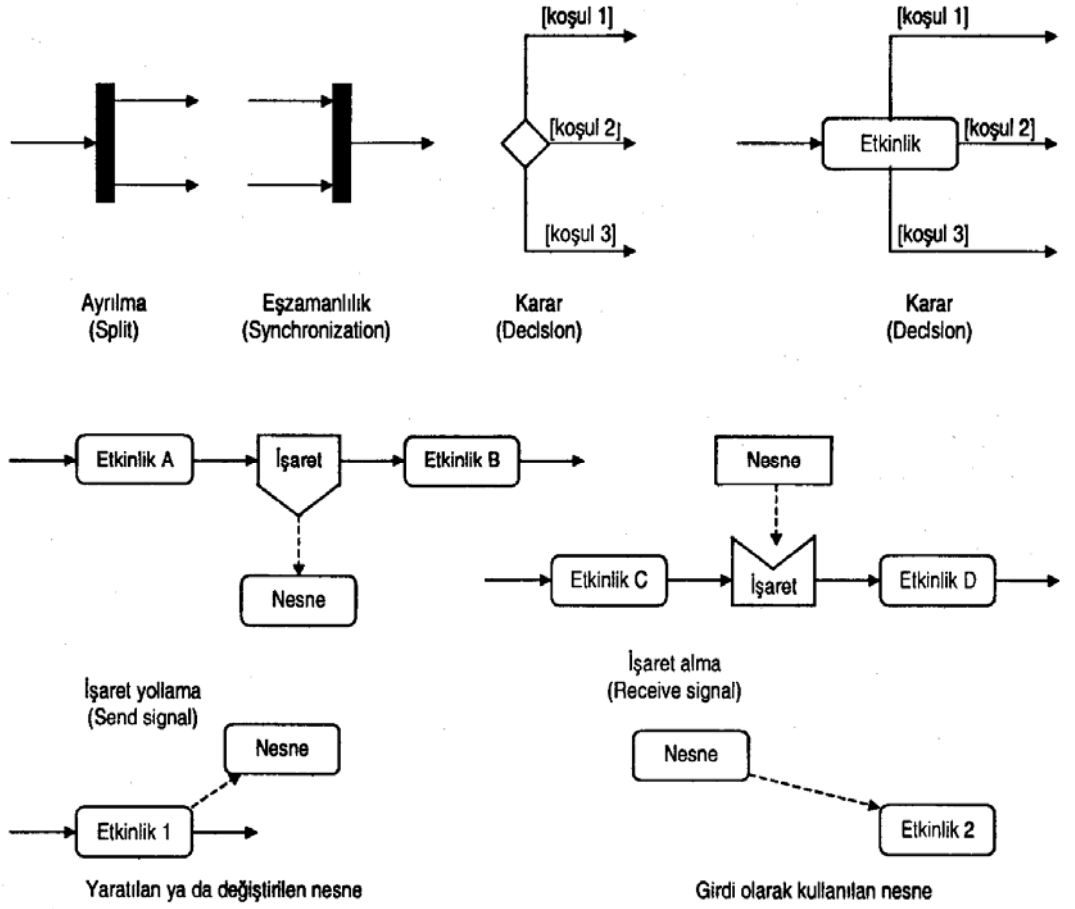
**Şekil 7: Etkinlik Şeması Örneği**



**Kaynak: Taşdelen, 2005, s:23**

UML etkinlik şemaları, klasik akış şemalarına benzemektedir. Ancak programcılık açısından etkinlik şemaları, akış şemalarına oranla daha üstündür. Akış şemaları basit algoritmalar için yeterlidir. Fakat etkinlik şemalarında daha ileri düzey özellikler mevcuttur (Şekil 8). Klasik akış şemasında paralel etkinlikler gösterilememektedir. Daima tek yol izlenmektedir. Etkinlik şemaları paralel etkinliklere izin verdiği gibi, bu etkinliklerin ortak başlama veya bitiş zamanlarının da gösterilmesini mümkün kılar. Akış şemasında ortaya çıkan ürünün nerede ve ne şekilde kullanılacağı belirtilmemektedir. Böyle bir ihtiyaç durumunda etkinlik şemaları tercih edilmelidir.

Şekil 8: İleri Düzey Etkinlik Şeması Bileşenleri



Kaynak: Güngören, 2005, s:134

## 1.5 Veritabanları

### 1.5.1 Veritabanı Kavramı

Veritabanı, birbirleriyle bütünleşmiş veri topluluğu olarak ifade edilebilir. Bir başka deyişle veritabanı, birbiriyle ilişkili ve bütünleşik veri topluluğudur. Bir veritabanı, sadece verilerin toplanması, biriktirilmesi değil, aynı zamanda içerdiği veri topluluğunu oluşturan elemanların birbiriyle olan anlamlı ilişkilerinin düzenlenmesini de içerir. Örneğin, bir üniversitenin elemanlarından (öğelerinden) olan öğrenciler ve dersler ele alındığında bu iki eleman arasında mantıksal bir ilişki



ortaya çıkar. Çünkü öğrenciler belli birtakım derslere girmektedirler. Buradan hareketle öğrenciler ve derslere ilişkin birden fazla ilişki kurulabilir. Bir üniversitenin öğrencilere ve derslere ilişkin veritabanında, hangi öğrencilerin, hangi dersler, hangi öğretim üyesinden, hangi derslikte, hangi gün ve saatte, ara sınav sonuçları, devam durumu, ders başarı durumu ve benzer konular yer alır (Sürmeli, Erdoğan, Erdoğan, Banar, Kaya, Sevim, 2006, s:71). Veritabanı kullanmanın yararları şunlardır (Hoşcan, Şahin, 2003, 118-119),

1. Veritabanında bilgiler yinelenmez. Her kayıttan sadece bir tane vardır. Bunu açıklanmak istenirse, herhangi bir YBS’de müşterilere ait adresleri içeren bir veritabanı varsa her müşteri için tek adres bilgisi vardır.
2. Veritabanında bulunan bilgiler standart hale getirilir. Yine aynı örnek üzerinde açıklamak gerekirse her müşterinin adres bilgileri standart olur. Her müşterinin bulunduğu il Eskişehir ise hepsinde Eskişehir’dir. Bilgi girişinde yanlışlık yapılarak bazı müşterilerde “Eskişehir”, bazılarında “Eskisehir”, bazılarında da “Eşkisehir” gibi yazım hatalarına fırsat verilmez.
3. Veritabanında bulunan bilgiler çeşitli kullanıcılar tarafından farklı işlemler için paylaşılabilir. YBS içinde kurulmuş bulunan ve YBS’nin gerçekten en önemli parçasını oluşturan veritabanında bulunan bilgiler kullanıcıların amaçları doğrultusunda farklı bakış açıları ile aynı anda kullanılabilir. YBS’deki veritabanında yine müşterilere ait bilgiler varsa bir kullanıcı müşterilere ait adresleri kağıt üzerine çıktı alırken, bir başka kullanıcı müşterilere ait hesap işlemlerini yapabilir.
4. Veritabanı üzerinde güvenlik sınırları oluşturulur. Tüm bilgiler aynı yerde saklandığı için giriş ve çıkışlar kontrol altında tutulur. İstenmeyen, güvenlik açısından sakıncalı kullanıcıların veritabanına erişimi engellenir.
5. Veritabanındaki bilgilerin periyodik olarak yedekleri alınır. Böylece istenmeyen herhangi bir durumda bilgi kaybı en aza indirgenir.

Yapı olarak bütün veritabanları aynı tipte değildir. Veritabanları, verileri saklama ve verilere erişim açısından farklılık gösterebilir. Veritabanlarının,

- İlişkisel Veritabanları,
- Hiyerarşik Veritabanları,
- Ağ Veritabanları,
- Nesneye Yönelik Veritabanları

olmak üzere farklı türleri vardır. Uygulama bölümünde ilişkisel veritabanı kullanılacağı için sadece ilişkisel veritabanı üzerinde durulacaktır.

### **1.5.2 Veritabanı Yönetim Sistemleri**

Veritabanı Yönetim Sistemi (VTYS), veritabanının oluşturulması, yönetilmesi ve korunmasını amaçlayan yazılım programlarıdır. VTYS ile amaçlanan, veri ve bilgi dosyalarının (veri topluluklarının) oluşturulması ve güncellenmesi, verilerin seçilmesi ve tekrar kullanılması (sorgulanması) ve çeşitli çıktı ve raporların alınmasıdır. Veritabanında yapılacak olan ekleme, silme güncelleme gibi işlemler VTYS aracılığı ile yapılır. Dolayısıyla veritabanlarıyla ilgili her türlü işletimsel gereksinimi karşılamak amacıyla kullanılan yazılım sistemine Veritabanı Yönetim Sistemi (VTYS) denir. Dolayısıyla, VTYS, bir işletmenin veriyi merkezileştirmesine, onu etkin bir şekilde yönetmesine ve saklanmış olan verilere belli bir amacı gerçekleştirmek için üretilmiş yazılımlar olan (Word, Excel, çeşitli muhasebe yazılımları vb) uygulama programlarına erişmesine olanak sağlayan bir yazılım olmaktadır (Sürmeli, Erdoğan, Erdoğan, Banar, Kaya, Sevim, 2006, s:77).

### **1.5.3 İlişkisel Veritabanı Yönetim Sistemleri**

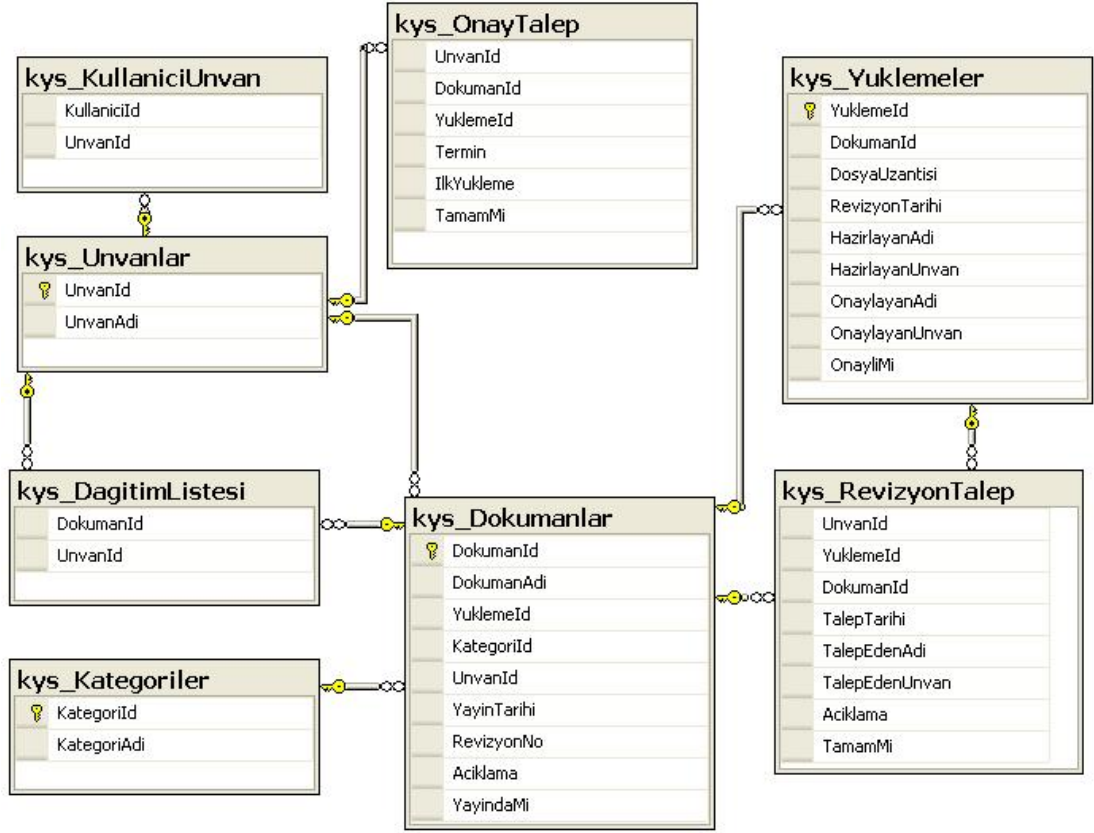
İlişkisel veritabanı, günümüzde en yaygın kullanılan veritabanı sistemlerinden biridir. İlişkisel veritabanları satır ve sütunların meydana getirdiği tablolardan (listelerden) oluşur. Bu tablolar birbiri ile ilişkileri olan tablolardır. Dolayısıyla bir veri tabanında ilişkiden söz edebilmek için en az iki tablonun yer alması ve bu iki tablodaki verilerin birbiri ile bir şekilde ilişkilendiriliyor olması gerekir. Bu şekilde ilişkisel veritabanları, veritabanı denilen büyük dosyalardan oluşur. Her bir tablo, belli yapıya uygun verileri saklamak üzere tasarlanır (Sürmeli, Erdoğan, Erdoğan, Banar, Kaya, Sevim, 2006, s:82).

İlişkisel veritabanında anahtar, birbiriyle eş kayıtları ayırmak için kullanılan bileşenlerdir. Birincil anahtar ise tablodaki bir kaydı diğer tüm kayıtlardan ayırmak için kullanılan bileşendir. Tüm ilişkisel veritabanlarında her bir ilişkide mutlaka birincil anahtar olmalıdır. Aynı konu ile ilgili olan bilgiler belirlenmelidir. Dizin olarak kullanılacak alanlar, zaman içinde değiştirilebilecek şekilde belirlenir ([www.gunduz.org/download.php?dlid=18](http://www.gunduz.org/download.php?dlid=18), 18/03/07). Bir İlişkisel Veritabanı Yönetim Sistemi tablolar üstünde şu üç işlevi yerine getirmek zorundadır (<http://www.verivizyon.com/kitapoku.asp?catid=69>, 18/03/07),

1. SEÇME: Herhangi bir tabloda (listede) yer alan tüm bilgileri gösterebilmelidir,
2. İZDÜŞÜRME: Herhangi bir tablodan sadece belli sütunların yer aldığı seçme işlevlerini yerine getirebilmelidir,
3. BİRLEŞTİRME: Birden fazla tabloda yer alan bilgileri, yeri geldiğinde tek bir tabloymuş gibi sunabilmelidir.

İlişkisel veritabanıyla ilgili bir örnek Şekil 9’da görülebilir. Şekilde sekiz adet veritabanı tablosunun kolonlarının satır şekilde tanımlandığı tablolar görülmektedir. Her tabloda kayıtların girileceği kolonların isimleri vardır. Tablolar arası ilişkiler, aynı verileri içeren kolonların birleştirilmesi ile olmaktadır. Birincil anahtar olan kolonun (şekilde satır ile temsil edilmektedir) solunda anahtar simgesi bulunmaktadır. Tablolar arasındaki ilişkilerde anahtarın bulunduğu tablodaki veri bir kere girilmek zorundayken, diğer tabloda defalarca yer alabilir. İlişkilerde sonsuz işareti olan taraftaki veri, diğer tablonun birincil anahtara sahip verilerinden oluşur.

Şekil 9: İlişkisel Veritabanı Tasarımı Görüntüsü



## **2 WEB TABANLI YAZILIM**

### **2.1 Web Ortamı ve Web Tabanlı Yazılım**

İnternet'in yaygınlaşması ile, bireyler ve kurumlar, birçok işi web üzerinden gerçekleştirir oldular. Kişisel eğlence aracından ziyade, alışverişten sınav başvurularına, e-devlet uygulamalarından kütüphane taramalarına kadar neredeyse hayatın her aşamasında yerini aldı.

Eskiden web, sadece statik sayfaların yayınlanması ile gerçekleştiriliyordu. Bu durum, sayfaları görüntüleyen kişilerin etkileşimli işlem yapmalarına fırsat vermiyordu. Web 2.0 olarak tanımlanan yeni nesil teknoloji sayesinde, Internet Explorer, Firefox, Opera gibi ağ tarayıcıları vasıtasıyla içerikleri değiştirilebilen sayfalar oluşturulmaya başlandı. Bu sayede, forumlar, arama motorları, içerik yönetim sistemleri gibi içerikleri ağ tarayıcısı aracılığıyla yönetilebilen, yeni sektörlerin oluşmasını sağlayan uygulamalar ortaya çıkmaya başladı.

Günümüzde bir çok kişi web ile profesyonel anlamda uğraşmaktadır. Kimi tasarım, kimi programlama, kimi de hazırlanan sistemleri kullanmak vasıtasıyla bir şekilde uğraşmakta. Web siteleri, birkaç statik sayfadan oluşabileceği gibi, karmaşık iş uygulamalarından da oluşabilir. Eğer karmaşık iş uygulamaları tasarlanmak isteniyorsa, iyi bir analiz ve ön çalışma yapılmalıdır.

### **2.2 Web Geliştirmede Yazılım Mühendisliğinden Faydalanmak**

Yazılım mühendisliği; yazılım geliştirmeye uygulanan sistematik ve ölçülebilir bir yaklaşımdır. Yani, mühendislik prensiplerinin yazılım geliştirmeye uygulanmasıdır. Bu, birçok web projesinde eksik olan bir yaklaşımdır. Bunun iki ana nedeni vardır: İlk neden, web geliştirme işleminin çoğunlukla yazılı raporların geliştirilmesine benzemesidir. Belge yapısı, grafik tasarımı ve üretim üzerine bir uygulamadır. Bu, belge yönelimli bir örnektir. Tüm bunlar küçük ve orta ölçekli statik siteler için iyi gibi görünse de web sitelerindeki dinamik içeriği, web siteleri

artık belgeler değil de servislerin sunduğu noktaya çıkarıldığında bu örnek artık uygun olamaz. Birçok insan bir proje için yazılım mühendisliği işleri yapmayı aklına bile getirmez (Welling, Thomson, 2005, s:496).

Yazılım mühendisliği işlerinin kullanılmamasının ikinci nedeni de, web uygulaması geliştirme işleminin, normal uygulama geliştirmeden birçok yönden farklı olmasıdır. Geliştiriciler, verilen hazırlık sürelerinin çok kısa olmasıyla ve siteyi hemen yapmaları gerektiği için sürekli bir baskıyla başa çıkmak zorundadır. Yazılım mühendisliği; işleri sırayla, planlı bir şekilde ve planlamaya zaman ayırarak yapmak demektir. Ancak web projelerinde genel yaklaşım planlamaya çok fazla zaman ayırmadan uygulamaya geçmek şeklindedir (Welling, Thomson, 2005, s:496-497).

Web projeleri üzerinde plan yapılmadığında, herhangi bir yazılım projesi üzerinde plan yapılmadığı zaman karşılaşılan sorunlar ortaya çıkar: Hatalı uygulamalar, gecikmeler ve okunamayan kodlar. O zaman yapılması gereken, web uygulamalarının bu yeni disiplini içinde, yazılım mühendisliğinden faydalanmaktır (Welling, Thomson, 2005, s:497).

### **2.3 Microsoft .NET Framework**

Microsoft .NET Framework, internetin dağıtılmış ortamında uygulama geliştirmeyi kolaylaştıran yeni bir hesaplama platformudur. .NET Framework, kodun güvenli çalışmasını garantileyen ve performans sorunlarını gideren, nesneye yönelik bir programlama ortamı sağlamak için tasarlanmıştır (<http://technet2.microsoft.com/windowsserver/tr/library/1737693b-931d-440b-acd0-e0e6f6b6a2f51055.mspx?mfr=true>, 27/06/2008).

.NET Framework, çok dağınık (highly-distributed), bileşen yönelimli uygulamaların geliştirilmesini ve yürütülmesini destekleyen bir ortam tanımlar. .NET Framework, farklılık gösteren bilgisayar dillerinin birlikte çalışmasını mümkün kılar ve Windows platformu için güvenlik, taşınabilirlik (programlar açısından) ve ortak bir programlama modeli sağlar. .NET Framework çok önemli iki unsur tanımlar.

Bunlardan ilki Common Language Runtime'dır (CLR – Ortak Dil Çalışma Zamanı). Bu, programınızın çalışmasını idare eden sistemdir. Diğer avantajlarının yanı sıra CLR, .NET Framework'ün programların taşınabilirliğini mümkün kılan, karışık dilde programlamayı destekleyen ve güvenliği sağlayan bir parçasıdır (Schildt, 2005, s:9).

İkinci unsur ise .NET sınıf kütüphanesidir. Bu kütüphane, programınızın çalışma zamanı ortamına erişmesine imkan verir (Schildt, 2005, s:9). .NET Framework sınıf kitaplığı CLR ile sıkı bir şekilde tümleşik olan yeniden kullanılabilir türler koleksiyonudur. Sınıf kitaplığı nesneye yöneliktir ve uygulamaların ek işlevler türetebileceği türler sağlar. Bu nesne yönelimi, .NET Framework türlerinin kullanımını kolaylaştırır. Ayrıca, diğer bileşenler, .NET Framework içindeki sınıflarla sorunsuz bir şekilde tümleşebilir. .NET Framework türleri, dizi yönetimi, veri koleksiyonu, veritabanı bağlantılabirliği ve dosya erişimi gibi birçok programlama görevinin gerçekleştirilmesini sağlar. Ek olarak, sınıf kütüphanesi, çeşitli özel geliştirme senaryolarını destekleyen türler içerir (<http://technet2.microsoft.com/windowsserver/tr/library/1737693b-931d-440b-acd0-e0e6f6b6a2f51055.msp?mfr=true>, 27/06/2008).

## 2.4 ASP.NET

ASP.NET, kuruluş sınıfında web uygulamaları oluşturmak ve dağıtmak için gereken hizmetleri sağlayan birleştirilmiş bir web uygulaması platformudur. ASP.NET, her tarayıcı veya aygıtı hedefleyebilen daha güvenli, ölçeklenebilir ve kararlı uygulamalar için yeni bir programlama modeli ve altyapısı sunar (<http://technet2.microsoft.com/WindowsServer/tr/Library/1fc6f54f-5131-4fbc-a402-4b7e2859ca361055.msp?mfr=true>, 27/06/2008).

ASP.NET, internetin büyük ölçüde dağıtılmış ortamında uygulama geliştirme işlemini basitleştiren bir bilgi işlem ortamı olan Microsoft .NET Framework'ün parçasıdır. .NET Framework sınıf kitaplığı, geliştiricilerin uygulama oluşturmak için kullanabilecekleri, kapsamlı, nesneye yönelik türlerden oluşan bir koleksiyondur. ASP.NET'in avantajları şunlardır (<http://technet2.microsoft.com>

[/WindowsServer/tr/Library/1fc6f54f-5131-4fbc-a402-4b7e2859ca361055.msp?mfr=true](#), 27/06/2008),

- **Yönetilebilirlik:** ASP.NET, ayarları web uygulamalarına ve sunucu ortamına uygulamayı kolaylaştıran, metne dayalı, hiyerarşik bir yapılandırma sistemi kullanır. Yapılandırma bilgileri düz metin biçiminde saklandığı için yeni ayarlar yerel denetim araçları kullanılmadan uygulanabilir. Yapılandırma dosyalarında yapılan değişiklikler otomatik olarak algılanır ve uygulamaya uygulanır,
- **Güvenlik:** ASP.NET, web uygulamaları için varsayılan yetkilendirme ve kimlik doğrulama düzenleri sağlar. Geliştiriciler uygulamanın gereksinimlerine bağlı olarak kolayca yeni düzen ekleyebilir, var olan düzenleri kaldırabilir veya değiştirebilir,
- **Dağıtım Kolaylığı:** ASP.NET uygulamasını sunucuya dağıtmak için gerekli dosyaları sunucuya kopyalamak yeterlidir. Çalışan derlenmiş kodu dağıtmak veya değiştirmek için bile sunucunun yeniden başlatılması gerekmez,
- **Gelişmiş Performans:** ASP.NET sunucuda çalışan derlenmiş koddur. ASP.NET, geleneksel Active Server Pages (ASP) uygulamalarından farklı olarak kullanıma hazır, geliştirilmiş bir performans sunmak için erken bağlama, zamanında (JIT) derleme, doğal en iyi duruma getirme ve önbelleğe alma hizmetlerinden yararlanır,
- **Esnek Çıkış Önbelleği:** ASP.NET, uygulamanın gereksinimlerine bağlı olarak sayfa verilerini, sayfaların bir bölümünü veya tamamını önbelleğe alabilir. Önbelleğe alınmış öğeler önbellekteki dosyalara veya diğer öğelere bağımlı olabileceği gibi, bir süre sonu ilkesine göre de yenilenebilir,
- **Uluslararası desteği:** ASP.NET, istek ve yanıt verilerini göstermek için kendi içinde Unicode kullanır. Uluslararası desteği bilgisayar, izin ve sayfa düzeyinde yapılandırılabilir,
- **Taşınabilir Aygıt desteği:** ASP.NET her tür aygıtta her tür tarayıcıyı destekler. Geliştiriciler geleneksel masaüstü tarayıcıları için kullandıkları programlama tekniklerini yeni taşınabilir aygıtlar için de kullanabilir,



- Ölçeklenebilirlik ve Kullanılabilirlik: ASP.NET özellikle kümelenmiş ve çok işlemcili ortamlarda performansı artıracak biçimde uyarlanmış özellikleriyle ölçeklenebilirlik sunmak üzere tasarlanmıştır,
- İzleme ve Hata Ayıklama: ASP.NET, hata ayıklama işlemi sırasında hem uygulama, hem sayfa düzeyinde etkinleştirilebilen izleme hizmetleri sunar. ASP.NET, .NET Framework hata ayıklama araçları sayesinde hem geliştirme sürecinde, hem uygulama üretime sokulduktan sonra yerel ve uzaktan hata ayıklama işlemini destekler,
- .NET Framework ile Tümüleşiklik: ASP.NET .NET Framework'ün parçası olduğundan web uygulamalarında tüm platformun gücünden ve esnekliğinden yararlanır. Web'den .NET sınıf kitaplığının yanı sıra ileti ve veri erişimi çözümlerine de sorunsuz olarak erişilebilir. ASP.NET dilden bağımsız olduğu için geliştiriciler uygulamaya en uygun dili seçmekte özgürdür,
- Varolan ASP Uygulamalarıyla Uyumluluk: ASP ve ASP.NET bir IIS web sunucusunda birbirini etkilemeden yan yana çalışabilir; ASP.NET'i yüklemek, var olan ASP uygulamalarına kesinlikle zarar vermez. Yalnızca .aspx dosya adı uzantısına sahip dosyalar ASP.NET tarafından işlenmeye başlar. Dosya adı uzantısı .asp olan dosyalar ASP altyapısı tarafından işlenmeye devam eder,
- Web Formları teknolojisi, form tabanlı güçlü web sayfaları oluşturulmasını sağlar. Web Formları sayfaları sayfanın kodunu basitleştirmek için yeniden kullanılabilir yerleşik veya özel bileşenler kullanır,

## 2.5 Yazılım Geliştirme Platformu: Visual Studio.NET

Visual Studio.NET, .NET platformu için geliştirilmiş uygulama geliştirme platformudur. En büyük özelliklerinden birisi IDE (Integrated Development Environment) denilen ortak bir uygulama geliştirme platformu sunmasıdır (<http://www.msakademik.net/vsnet.aspx>, 02/07/08). Visual Studio.NET, .NET Framework sınıf kitaplığında bulunan nesnelere kullanılmalarını sağlar. Bu nesnelere, sürükle bırak tekniğiyle kolayca eklenip, istenildiği takdirde özelleştirilebilirler. Bu

uygulama, hazırlanan projede modüllerin ve kullanıcı kontrollerinin oluşturulması gibi birçok kolaylık sunar.

Uygulamada, Visual Studio'nun ücretsiz sürümü olan Microsoft Visual Web Developer 2005 Express Edition kullanılmıştır. Bu sürümde, bazı özellikler eksik olmakla birlikte, temel gereksinimleri karşılayacak düzeydedir. Ancak profesyonel uygulamalarda ve proje grupları halinde çalışılırken, Visual Studio'nun farklı hedef kitleler için hazırlanan türlerinden biri tercih edilebilir.

## 2.6 Yararlanılan Diller ve Teknolojiler

### 2.6.1 HTML

Zengin Metin İşaret Dili (İngilizce Hyper Text Markup Language, HTML) günümüzde internet üzerinde veri paylaşımı için kullanılan en yaygın metin tabanlı dildir. HTML, Bir programlama dili olarak tabir edilemez. Zira HTML kodlarıyla, kendi başına çalışan bir program yazılamaz. Ancak bu dili yorumlayabilen programlar aracılığıyla çalışabilen programlar yazılabilir. Programlama dili denilememesinin nedeni tam olarak budur. Temel gereği yazı, görüntü, video gibi değişik verileri ve bunları içeren sayfaları birbirine basitçe bağlamak, buna ek olarak söz konusu sayfaların internet tarayıcı yazılımlar tarafından düzgün olarak görüntülenmesi için gerekli kuralları belirlemektir ([http://tr.wikipedia.org/wiki/Zengin\\_metin\\_i%C5%9Faretleme\\_dili](http://tr.wikipedia.org/wiki/Zengin_metin_i%C5%9Faretleme_dili)).

W3C tarafından standartlaştırılmaktadır. Web kavramı, CERN'de bir bilgisayar programcısı olan Tim Berners-Lee'nin HTML adlı bilgisayar dilini bulup geliştirmesiyle oluşmuştur. Html kodları "<" ve ">" arasına yazılıp "</" ve ">" biterler ([http://tr.wikipedia.org/wiki/Zengin\\_metin\\_i%C5%9Faretleme\\_dili](http://tr.wikipedia.org/wiki/Zengin_metin_i%C5%9Faretleme_dili)). Örnek olarak bir metni paragraf şeklinde görüntülemek için aşağıdaki gibi etiketlenir,

<p>Paragraf şeklinde görülecek yazı</p>

## 2.6.2 CSS

Kademeli Stil Dosyaları anlamına gelen CSS (Cascading Style Sheets), web sayfalarının şekillendirilmesinde kullanılır. HTML kodları ile biçimlendirmeye nazaran önemli miktarda zaman ve emek kazancı sağlayan CSS, bunu yüzlerce sayfa için tek bir standart stil dosyası oluşturulabilmesine borçludur. Örneğin 50 sayfalık internet sitesinin stil dosyasında küçük bir değişiklikle paragrafın iki yana yaslı olması sağlanabilir. Böylece 50 sayfada teker teker değişiklik yapmaya gerek kalmadan, istenen işlem gerçekleştirilebilir.

CSS'nin özellikleri sadece yazı karakterleri ile sınırlı değildir. CSS, görme engellilerin kullandığı ekran okuyucular için problem niteliğindeki tablolardan kaçınarak, siteyi inşa edebilme olanağı sağlar. Tablo kullanılmayan sitelerin sayısı her geçen gün artmaktadır.

## 2.6.3 JavaScript

Netscape Navigator 2.0 ile birlikte Brendan Eich tarafından geliştirilen ve önceleri Mocha daha sonra LiveScript olarak adlandırılan ve en sonunda şu anki adını alan JavaScript dili başlangıçta sadece istemci taraflı yorumlanan bir dildir. Aralık 1995 tarihinde, C dilinin tarayıcılara uyarlanmış hali diyebileceğimiz JavaScript piyasaya sürüldü. Web sayfalarında dinamik içerik sağlamak ya da kullanıcıyla iletişim kurmak için kullanılmaktadır (<http://tr.wikipedia.org/wiki/JavaScript>).

Uygulamada bölümünde JavaScript, girilen bilgilerin doğrulanması, boş bırakılmış alanlar için uyarıda bulunma, açılıp kapanan paneller gibi özellikler için kullanılmıştır. JavaScript'in en önemli özelliği, sunucu taraflı değil de istemci taraflı çalışabilmesidir. Örneğin, sayfaya girilen her bilginin sunucuya gönderilmesine gerek kalmadan işlem görebilmesi sağlanabilir.

## 2.6.4 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 ([http://www.rssnedir.com/xml\\_nedir.php](http://www.rssnedir.com/xml_nedir.php), 06/06/08). XML'in sağladığı yararlar şöyle sıralanabilir (<http://www.bilgiservisim.com/2007/08/21/xml-nedir>, 06/06/2008),

- Açık standartlar,
- Esnek web uygulamalarının geliştirilmesi,
- Farklı kaynaklar ve uygulamalar arasında verilerin bütünleştirilmesi,
- Değişik uygulamaların içindeki verileri tanımlamak için kullanılır. Verinin kendi kendini tanımlayabildiği için (self-describing) gönderen uygulamaya bağlı kalmadan alınıp işlenebilir,
- XML sayesinde yalnızca değişen verilerin güncellenmesi sağlanır. Değişen verinin görüntülenmesi bütün sayfanın yenilenmesine yol açmaz,
- XML içinde kodlanmış veriler web üzerinden masaüstlerine taşınır. XML'in ulaştırılması için de HTTP kullanıldığı için bütün veritabanları ve dosyalarla uyumlu hale gelir,
- XML, metin tabanlı bir formata sahiptir ve uygulama geliştiricilerin uygulamalarla istemciler arasında veri transferini sağlar. XML ayrıca yapısal verilerin sunucular arasında da alışverişini sağlar.

ASP.NET'te XML formatındaki dosyalar, birçok işlevde kullanılabilir. Örneğin, bütün ayarların yapıldığı Web.config dosyası, veritabanı amacıyla kullanılan XML dosyalar veya site ağacı, bunlardan sadece birkaçıdır.

### 2.6.5 C#

C#, .NET uygulamaları geliřtirmek için Microsoft'un öncelikli dilidir. Güncel yeniliklerle birlikte gelen gücü kanıtlanmış özellikleriyle, modern kurumsal bilgi işlem ortamları için kullanışlı ve verimli programlar yazmayı kolaylaştırır. Bu sebeple, her açıdan 21. yüzyılın en önemli programlama dillerinden biridir (Schildt, 2006, s:4). Uygulama bölümünün programlama kısımları C# ile yazılacaktır.

### 2.6.6 SQL ve Saklı Yordamlar

SQL, Structured Query Language, yani Yapılandırılmış Sorgu Dili anlamına gelir. İliřkisel veritabanı yönetim sistemlerine (RDBMS – Relational database management systems) erişim için standart dildir. SQL, bir veritabanında veri depolamak ve veritabanından veri almak için kullanılır. MySQL, Oracle, PostgreSQL, Sybase, Microsoft SQL Server ve daha birçok veritabanı sisteminde kullanılır (Welling, Thomson, 2005, s:236).

Saklı yordamlar, derlenmiş SQL cümlecikleridir. Bunlar birer veritabanı nesnesi oldukları için, doğrudan veritabanı yöneticisi olan programda (örneğin SQL Server) yer alırlar. Bu nedenle veritabanı bir yere taşındığında, saklı yordamlar da otomatik olarak taşınmış olur. Bu, saklı yordamların tercih edilme nedenlerinden sadece birisidir. Diğer yandan, derlenmiş olmaları aslında bu SQL cümleciklerinin doğrudan makine diline dönüřtürüldüğü anlamına gelmez. Aslında, çalıştırılmak istenen SQL cümleciklerini bir saklı yordam içine yerleştirilerek, bunun bir veritabanı nesnesi haline gelmesi ve çalıştırıldığında doğrudan, veritabanı yöneticisini üzerinde barındıran sunucu makinede işlemesi sağlanır. Bu, doğal olarak, istemci makinelerdeki iş yükünü azaltır ve performansı arttırır. Nitekim bir program içinde çalıştırılan SQL cümleleri, saklı yordamlardan çok daha yavaş sonuç döndürür. Dolayısıyla saklı yordamlar özellikle çok katlı mimarinin uygulanması istenen projelerde faydalıdır (<http://www.csharpnedir.com/makalegoster.asp?MId=208>, 01/07/2008).

Uygulama bölümünde, veritabanı sorguları sadece saklı yordamlar vasıtasıyla gerçekleştirilecektir. Saklı yordamlar, program tarafından birkaç kerede süzulebilecek sorguların, birleştirilip zamandan kazanılmasını sağlar. En güzel özelliği, “IF-ELSE” gibi kodların yerleşik olarak barındırılmasıdır. Bu sayede, eğer veritabanı, site sunucusundan uzakta barındırılıyorsa, sorgu sayısını azaltarak, trafiğinin ve her sorguyla artan istemcinin cevaplanması süresinin düşürülmesini sağlar.

## 3 KALİTE YÖNETİM SİSTEMİ

### 3.1 Kalite İle İlgili Kavramlar

#### 3.1.1 Kalite Kavramı

Hızlı ve sürekli gelişen çevre, işletmeleri etkilemekte ve değişikliğe zorlamaktadır. İşletmeler varlıklarını sürdürebilmek için değişiklikleri önceden tahmin etmek ve acımasız rekabet ortamında üstünlük sağlamak zorundadırlar. Sürekli artan küresel şartları, işletmenin önüne yaşamlarını sürdürebilmeleri için en temel özelliği çıkartmıştır: “Müşteri tatmini”. Bu özelliğe sahip olmak için gerekli temel üç şart ise “Müşterinin istediği mal ve hizmeti”, “daha ucuza” ve “kısa sürede” ulaştırmaktır şeklinde karşımıza çıkmaktadır. Günümüzde hızla artan ürün ve hizmet arzında müşterinin en temel tercih faktörü kalitedir (Özalp, Şahin, Berberoğlu, Ceylan, 2003, s:207).

Kalitenin günümüze kadar değişik tanımları yapılmıştır. Geçmiş dönemlerde kalite, ürün sağlamlığını, uzun ömürlülüğü olmasını ifade etmekteydi. Yapılan bir eşyanın ya da aletin sağlamlığı, kaliteli olarak değerlendirilmesine neden olmaktadır. Sanayi devrimi ile birlikte ürünlerde, ölçülere uyum; bir başka ifadeyle, standart kavramı gündeme gelmiştir. Ürünlerin uygun standartlara uygun olarak yapılması sonrasında ise ürünün belirlenen standartlara uygunluğu, kusursuzluğu kalite olarak kabul edilmiştir (Özalp, Şahin, Berberoğlu, Ceylan, 2003 s:207).

Kalitenin özellikleri şunlardır (Doğan, Tütüncü, 2003, 28-30),

1. Performans: Bir ürünün temel işlev özellikleri anlamına gelen performans, örneğin bir otomobil için hız, konfor; bir televizyon için renk, ses, görüntü vb. özellikler olabilmektedir.
2. Özellikler: “Özellik” kelimesi bir ürünün temel fonksiyonlarını tamamlayan kavram olarak nitelendirilebilir.

3. Güvenilirlik: Ürünün kullanım ömrü içerisinde kendisinden beklenen tüm fonksiyonları tam olarak yerine getirip getirmediğinin ölçütüdür.
4. Uygunluk: Uygunluk, ürünün tasarımının ve işleyiş özelliklerinin önceden belirlenmiş standartlara uyup uymama derecesidir.
5. Dayanıklılık: Bir ürün veya hizmetin kullanım ömrünün uzunluğudur.
6. Servis Görme Yeteneği: kalitenin altıncı boyutu hizmet görme yeteneği, yani hız, çabukluk, nezaket, yeterlilik, ehliyet ve tamir edebilme kolaylığı olarak ifade edilmektedir.
7. Estetik: Estetik, tüketicilerin beş duyusuna hitap eden ürün özellikleridir.
8. Algılanan Kalite: Tüketiciler her zaman ürünün tüm özellikleri ile ilgili ayrıntılı bilgi sahibi değildirler ve böyle durumlarda dolaylı bir takım ölçütler karar vermelerinde önemli rol oynamaktadır. Reklam faaliyetlerinde yaratılan ürün imajı, marka imajı gibi faktörler ürün kalitesinin tüketici tarafından olumlu veya olumsuz algılanmasına neden olabilir.

### **3.1.2 Kalite Kontrol**

Son yıllarda kalite kontrolü, faaliyet yönetiminin temel konusu haline gelmiştir. Kontrol, bir eylemi denetlemek ve yönlendirmektir. Kalite kontrol, bir eylemin fiili sonuçlarının kalite standartlarıyla ya da hedefleriyle karşılaştırılmasını, bunlar arasındaki uyumsuzlukların saptanmasını ve bu uyumsuzlukların giderilmesini sağlayacak düzeltici önlemlerin alınmasını gerektirir (Benligiray, 2005, s:184).

Kalite kontrolünün temel amacı, kalite beklentilerini karşılamaktır. Bunun için önceden belirlenen kriterlere, standartlara göre, prosedürlerin, yöntemlerin, koşulların, süreçlerin, kayıtların ve insan unsurunun sürekli gözlemlenerek analiz edilmesi ve uygunluğun saptanması gerekir. Kaliteyle ilgili faaliyetlerin ve ulaşılan sonuçların planlara uyup uymadığı ve amaca ulaşmak için uygun olup olmadıkları sistematik ve tarafsız olarak incelenmelidir. Ayrıca kaliteyi daha da geliştirmek ya da hataları gidermek için neye gereksinim duyulduğu saptanmalıdır. Bütün bu



kontroller, genellikle yöneticiler, incelenecek alanda doğrudan sorumluluk taşımayan görevliler ya da müşteriler tarafından yapılır (Benligiray, 2005, s:184).

### **3.1.3 Kalite Güvence**

Kalite güvencesi, müşteri yararına yürütülen bir faaliyettir ve bu faaliyetin amacı, kalite aracılığıyla müşteri memnuniyetini güvence altına almaktır. Kalite güvencesi sistemi, işletmede çeşitli departmanların gösterdiği kaliteyi, korumayı, iyileştirmeyi, müşterinin tam beğenisini kazanmayı ve en ekonomik düzeyde bir üretim/hizmet sağlamayı amaçlayan çabaların bileşimidir. Çünkü kaliteyi sağlama, onu sürekli güvence altında tutma ve kaliteyi geliştirme rastlantılarla değil, ancak sistemli çabalarla mümkündür (Benligiray, 2005, s:185).

Uzun yıllar boyunca kaliteyi yönetme ve yönetimin kalitesini artırma çalışmaları sonuçta birleştirilmiş ve toplam kalite yönetimi doğmuştur. Çünkü üstün bir kalite tutturabilmek için rakiplerden daha çok müşterilerin taleplerini karşılayacak bir dizi üretim şartnamesi ve hizmet standardı geliştirmek gerekmektedir. Rakiplerle baş edebilmek için gerekli kalite standardı ise, yine belirlenen ürün şartnamesine ve hizmet standardına sıkı sıkıya uymaktan geçmektedir (Benligiray, 2005, s:185).

### **3.1.4 Toplam Kalite Yönetimi**

Toplam Kalite Yönetimi, müşteri beklentilerini her şeyin üzerinde tutan ve müşteri tarafından tanımlanan kaliteyi, tüm faaliyetlerin yürütülmesi sırasında ürün ve hizmet bünyesinde oluşturan bir yönetim biçimidir. Dinamik pazarda, yıkıcı Pazar rekabetin karşısında ayakta kalabilmek ancak bu anlayışı benimsemekle mümkün olacaktır. En basit şekli ile TKY (Doğan, Tütüncü, 2003, s:37),

Toplam = Herkesin katılımı,

Kalite = Müşteri gereksinim ve beklentilerinin tam olarak karşılanması,

Yönetim = Kalitenin ürün ya da hizmet için bütün koşulların sağlanması.

### **3.2 TSE EN ISO 9001:2000 Kalite Yönetim Sistemi Standardı**

Kalite yönetim sistemi, kalite politikasını, hedeflerini ve sorumlulukları belirleyen ve bunları kalite sistemi dahilinde kalite planlama, kalite kontrol, kalite güvence ve sürekli iyileştirme gibi metotlarla uygulayan yönetsel fonksiyonların bütünüdür (ISO 8402:1994). Kuruluşlar kalite yönetim sistemini uygulayarak ürün ve hizmet kalitesini geliştirmeyi, müşteri memnuniyetini arttırmayı ve sürekli iyileşmeyi amaçlar. Bunun için uluslararası bir standart olan ISO 9001:2000 Kalite Yönetim Sistemi standardı mevcuttur.

ISO 9001:2000 Kalite Yönetim Sistemi standardı, kuruluşların etkin kalite yönetim sistemi oluşturabilmeleri için yol gösterici bir standarttır. Bu standart her kuruluşun kendi ihtiyaçlarına özgü kalite yönetim sistemi oluşturabilmesini sağlar. Dolayısıyla tek düze bir kalite yönetim sistemi standardı uygulamasından söz edilemez. Bu da ISO 9001'in esnekliğinin en belirgin özelliğidir.

ISO 9001 standardının karşılanıp sistemin etkin bir şekilde kurulup faaliyete geçebilmesi için ilk ve en önemli koşul üst yönetimin benimsemesidir. Üst yönetimin kabul edemediği veya tam olarak algılayamadığı bir kalite yönetim sistemi büyük olasılıkla başarısız olacaktır.

ISO 9000, bir kuruluşun kaliteyi etkileyen tüm süreçlerinde, kalitesizliğin oluşmasını önleyici bir kontrol düzeni kurarak, bu süreçlerde insan, makine ve bilgi unsurları arasındaki eşgüdümün en uygun kalite-verimlilik-maliyet bileşiminde sağlanabilmesi için gerekli teknik ve yönetsel prosedürlerin oluşturulmasını ve kullanılmasını sağlayan, yani kalite güvencesini sağlayan uluslar arası standartlara uygunluk niteliği taşıyan bir araçtır. ISO 9000 standartları Toplam Kalite ile birlikte uygulanırsa başarısı daha da artacaktır. Bu nedenle ISO 9000 uygulamaları anahtar performans alanlarından biridir (Doğan, Tütüncü, 2003, s:37).

### 3.2.1 9001:2000 Standardının Temel Prensipleri

ISO 9001 Kalite Yönetim Sistemi standardının sekiz prensibi vardır. Bu prensipler,

- 1) Sürekli İyileştirme
- 2) Müşteri Odaklılık
- 3) Karar alma mekanizmasında gerçekçi yaklaşım
- 4) İnsanların katılımı
- 5) Karşılıklı fayda sağlayan tedarikçi ilişkileri
- 6) Proses yaklaşımı
- 7) Yönetim sistem yaklaşımı
- 8) Liderlik

#### 3.2.1.1 Sürekli İyileştirme

Sürekli iyileşmeyi ilke haline getirmemiş bir kuruluşun rekabet yeteneğini koruması, artan müşterileri beklenti ve ihtiyaçlarına cevap vermesi zorlaşır. Piyasa sürekli gelişmekte ve rekabet her geçen sürede daha da artmaktadır. Kuruluşun içinde bulunduğu çevreye ayak uydurabilmesi ve rekabet gücünü koruyabilmesi için sürekli iyileştirmeyi uygulayabilmelidir.

Değişimin ve gelişimin en belirleyici faktörü sürekli iyileştirmedir. Sürekli iyileştirmeye odaklanmış firmaların gelişmesi kaçınılmazdır. Sürekli iyileştirme için firmaların devamlı olarak faaliyetlerini gözden geçirmesi gerekmektedir. Kalite yönetimi sistemi içersinde arzulanan kalite düzeyine ve müşteri tatminine birdenbire erişilemeyeceği, bunun ancak sürekli iyileştirme süreci içinde gerçekleşebileceği açıktır. Firmaların performansı sürekli iyileşme, yaratıcılık ve öğrenme kültürü içinde bilgi paylaşımına dayalı bir yönetim biçimi ile yükselebilir. Bu çerçevede sürekli iyileştirme firmaların kalıcı bir hedefi olmalıdır (Özveri, 2007, s:38).

Sürekli gelişimin oluşturulması için firmalarda yeterli bir alt yapının oluşturulması ve aşağıdaki konuların sağlanması gerekli olur (Özveri, 2007, s:38),

- Gelişmelerin uygulanmaya yansması,
- Potansiyel gelişme alanlarının tanımlanması için belirlenmiş mükemmellik kriterlerini kullanarak periyodik tetkiklerin yapılması,
- Tüm süreçlerin etkinliğinin ve verimliliğinin sürekli iyileştirilmesi,
- Önleme tabanlı aktivitelerin özendirilmesi,
- Firmadaki her kişinin uygun sürekli iyileştirme araç ve yöntemleriyle donatılması için eğitim ve öğretimin verilmesi.

Sürekli gelişme çalışmalarının firmalara sağlayacağı faydalar aşağıdaki gibi sıralanabilir (Özveri 2007, s:38),

- Strateji ve iş planlamasını sürekli iyileştirmeyle birleştirerek daha rekabetçi iş planlarının yaratılması,
- Gerçekçi ve rekabetçi iyileştirme hedefleri ile bunlara ulaşmak için kaynakların ortaya konulması,
- Süreçlerin sürekli iyileştirmesine herkesin katılımının sağlanması,
- Ürün, süreçler ve sistemlerin iyileştirilmesi için firmadaki çalışanları fırsatlara yönlendirmesi ve cesaretlendirmesi.

### **3.2.1.2 Müşteri Odaklılık**

Kuruluşların varlıklarını sürdürebilmesi müşterilere bağlıdır. Kuruluşlar, müşterilerini çok iyi analiz etmeli, sundukları ürün ve hizmetlerle memnun etmeli ve onların fikirlerine önem vermelidir. Müşteri kavramı, iç müşteri ve dış müşteri olarak ikiye ayrılır. İç müşteriler kuruluşun kendi çalışanlarıdır. Kuruluş, çalışanlarına ne kadar değer verirse, onların maddi ve manevi gereksinimlerini ne kadar iyi karşılarsa, getirisi de o kadar çok olacaktır.

Firmalar müşterileri ile var olduklarına göre onların şimdiki ve gelecekteki ihtiyaçlarını anlayarak beklentilerinin de ötesinde hizmet verme anlayışını benimsenmelidirler. (<http://www.kalite.saglik.gov.tr/kalitekavramlari/iso9000.htm>, 27/12/2006). ISO 9000 firmaların müşteri tatmini ile ilgili bilgiyi izlemesini şart koşturmaktadır. Firmaların temel amacı müşteri ihtiyaçlarını karşılamak olduğundan, bu bilgi kalite yönetim sisteminin en önemli performans göstergesi olarak değerlendirilmektedir (Baş, 2004, s:4-5).

Müşteri odaklı çalışmaları savunan bu anlayışta, müşteri kavramı sadece üretilen mal ve hizmetleri alan kişi ve kurumlar olarak tanımlanan dış müşterilerle sınırlı kalmayıp, aynı zamanda firma içi müşterileri de kapsamaktadır. Kalite yönetimi yaklaşımında İç ve Dış müşteri olmak üzere iki müşteri türü belirtilmektedir ([http://www.danismend.com/konular/pazarlamayon/paz\\_musteri\\_odaklilik.htm](http://www.danismend.com/konular/pazarlamayon/paz_musteri_odaklilik.htm), 05/11/2006).

Firmalar müşterilerine bağlı olarak hayatta kalacakları için müşteri ürün veya hizmetin kalitesinin son hakemidir. Müşteri bağlılığı, kalıcılığı ve pazar payı kazanımı firmanın mevcut ve potansiyel müşterilerinin beklentilerine odaklanmasıyla en iyi şekilde optimize edilir. Bu nedenle firmalar müşterilerinin bugünkü ve gelecekteki ihtiyaç ve beklentilerini anlamalı, bu ihtiyaç ve beklentileri karşılamalı ve hatta aşmalıdır. Firmalar bunu karşılayabilmesi için aşağıdaki faaliyetleri gerçekleştirmelidir (Doğan, Tütüncü, 2003, s.55),

- Müşteri istek ve beklentilerinin araştırılması ve anlaşılması,
- Müşteri ve fayda sağlayan tarafların ihtiyaç ve beklentileri arasında dengeli bir yaklaşım sağlanması,
- Müşteri ihtiyaç ve beklentilerin doğru olarak belirlenip ürün ve hizmetlere yansıtılması,
- Müşteri memnuniyeti ölçümü ve sonuçlara göre, müşteri davranışının analiz edilmesi,
- Müşteri ilişkilerinin sistematik bir şekilde yönetilmesi.

Bu yaklaşımın temel yararları aşağıdaki gibidir (Özveri, 2007, s:30),

- Müşteri ihtiyaçlarının firmada anlaşılması ve çalışanların bu yönde motive edilmesi,
- Hedeflerin ve başarının müşteri ihtiyaç ve beklentileriyle beraber değerlendirilmesi,
- Müşteri ihtiyaçlarını karşılama amacıyla firmanın performansının artırılması yoluyla firma içi iletişimsizliğin en aza indirilmesi,
- Müşteri memnuniyeti ile çalışanların bilgi, birikim ve becerilerinin artması.

### **3.2.1.3 Karar Alma Mekanizmasında Gerçekçi Yaklaşım**

Karalar verilirken sezgiler veya duygulardan yararlanılabilir fakat, kararlar ilgili bilgiler, olaylar ve veriler toplanıp analiz edildikten sonra kararların verilmesi daha etkili olur (Seaver, 2002, s:27). Etkili kararlar veri ve bilgilerin analizine dayandırılmalıdır. Firmalar kalite yönetim sistemi kapsamında veri, enformasyon ve bilgi tanımlarını yaparak bunların nasıl elde edilebileceğinin yöntemlerini de oluşturmuş olmalıdır. Böylelikle verilecek kararlarda yararlanılabilir ve güvenilir bilgilerin kullanılması mümkün olabilir. Verilere dayalı karar verme sürecinde (Özveri, 2007, s:39),

- Hedeflerle ilgili ölçümlerin yapılması, veri ve bilgi toplanması,
- Veri ve bilgilerin yeterli doğrulukla, güvenilir ve geçerli metotlarla analizinin yapılması,
- Uygun istatistik tekniklerin kullanılması,
- Sezgi ve deneyimle dengelenmiş mantıksal analiz sonuçlarının karar verme amacıyla kullanılması gerekmektedir.

Bilginin ve elde edilme yollarını tanımlanması firmalar açısından gerek karar verme, gerekse diğer faaliyetlerde önemli yararlar sağlayacaktır. Gerçeklere dayalı karar verme ilkesinin uygulanması ile aşağıdaki faydalar sağlanabilir (Halis, 2004, s:139),

- Politika ve strateji oluşumu: Veri üzerine kurulu stratejiler daha gerçekçidir ve doğrulanması kolaydır,
- Amaç ve hedefler: Gerçekçi ve rekabetçi hedef ve amaçları oluşturmak için verilerin kullanılmasını sağlar,
- Sürekli iyileştirme: İyileştirmeye kılavuz olacak şekilde süreç ve sistem performansını anlamak ve gelecekteki problemlerin önlenmesine yardımcı olabilecek verilerin oluşturulmasını sağlar,
- İnsan kaynakları yönetimi: Veri ve bilgi analizi yaparak insan kaynakları politikalarının oluşturulmasını sağlar.

#### **3.2.1.4 Çalışanların Katılımı**

Bütün çalışanların tam anlamıyla katılımı, her düzeyde çalışana düşüncelerini ve yeteneklerini kuruluş yararına paylaşma fırsatı verir. Bu sayede yeni fırsatlar yakalanabilir, tepe yönetiminin fark edemediği aksaklıklar önlenebilir ve benzeri yararlar sağlanabilir.

Çalışanların potansiyeli; "firmaların değerleri", "güven" ve "yetkilendirmeye dayalı firma kültürü" ile ortaya çıkarılır. "Bir işi en iyi, o işi yapan bilir" temel prensibini esas alan bu anlayışta iş süreçlerinin iyileştirilmesi ve geliştirilmesinde bizzat o işi yapan personelin katılımı çok önemlidir. Çalışanların katılımı, kalite yönetimi felsefesinin temel taşlarından. Çalışanların katılımı, birlikte iş yapmaları ya da dostane ilişkiler içinde bulunmaların yanında, problemlerin çözümünde tüm çalışanların enerjilerinden faydalanmak şeklinde anlaşılmalıdır. Kaliteye ulaşmak, firmada çalışan herkesin sorumluluğunu gerektirmektedir. Birey olarak herkes kalitenin önemini anlamalı ve kaliteye ulaşmak için sorumluluk hissetmelidir (Özveri, 2007, s:32).

Toplam kalite yönetimi, tüm işlem ve aşamalar için top yekun kaliteyi hedeflemektedir. Top yekun kalite ve mükemmelleşme sağlanarak müşteri memnuniyeti sağlanabilir. Kaliteli ürünlerin sunulmasında sistemin önemli bir role sahip olmasına karşın, sistemin iyi bir şekilde işleyip işlememesinde insan faktörü

belirleyici bir rol oynamaktadır. Bu nedenle, kalitenin temini yönünde yapılacak çalışmalarda çalışanlara değer verilmeli, görüş ve önerileri dikkate alınmalıdır (Özveri, 2007, s:33).

Toplam Kalite Yönetiminde çalışanların katılımı üç düzeyde gerçekleşir (Bayrak,1999, s:49),

1. **Bireysel Düzeyde:** Bireysel katılım, öneri sistemleri ile hayata geçirilir. Öneri sistemi ile bireyin yaratıcı gücü firmanın gelişimiyle ilişkilendirilerek önemli bir başlangıç yapılır,
2. **Grup Düzeyinde:** Çalışanların işlerinin iyileştirilmesine yönelik gönüllü katılımı ile sorunlara duyarlılığını artıracaktır. Gruba katılarak öneri getirme ve çözümde yer alma, çalışanların özgüvenlerini yükseltmek ve bireysel gelişmelere imkan sağlamak suretiyle kalite konusunda önemli katkıda bulunmaktadır,
3. **Firma Düzeyinde:** Bu düzeyde katılım, kalite yönetim anlayışının firmadaki tüm işlemlere yayılması şeklinde gerçekleşir. Firmanın kararlarına katılım, açıklık, sorumluluk üstlenme ve sorumluluğu paylaşma gibi unsurlar firma içi katılımı artıracaktır.

Kaliteli hizmet sunulmasında sistemin önemli bir role sahip olmasına karşın, sistemin iyi bir şekilde işleyip işlememesinde insan faktörü belirleyici bir rol oynamaktadır. Bu nedenle, kalitenin temini yönünde yapılacak çalışmalarda bireylere değer verilmeli, görüş ve önerileri dikkate alınmalıdır. Çalışanların kalite çalışmalarına katılımının özendirilmesi, kalite çalışmaları için uygun ortam temin edilmesi ve kalite çalışmalarına her aşamada yardımcı olunması gerekmektedir. (Özveri, 2007, s:33)

Her seviyedeki çalışanlar firmaların özüdür ve tam katılımının sağlanması ile yeteneklerinin firma yararına kullanılması sağlanabilir. Bireylerin kalite yönetim sistemine tam katılımı gerek sistemin benimsenmesi ve uygulamada yaratılacak yararlar, gerekse gelecekte sistemin geliştirilmesi açısından büyük önem



taşımaktadır. Böylesi bir katılımın sağlanabilmesi için firmaların aşağıdaki şartları sağlamaları faydalı olabilmektedir (Özveri, 2007, s:34),

- Problem çözümü için sahiplenme ve sorumluluğu kabul,
- Gelişmeler için aktif olarak fırsat aramak,
- Bilgi ve deneyimi güçlendirmek üzere aktif olarak fırsat aramak,
- Gruplar ile takımlarda bilgi ve deneyimi serbestçe paylaşmak,
- Müşteri için değer yaratma fırsatları yaratmak,
- Firma hedeflerini ilerletmek için yaratıcı ve yenilikçi olmak,
- Yapılan işlerden memnun olmak,
- Firmanın parçası olmaktan heyecan ve gurur duymayı sağlamak.

Etkin bir katılımın sağlanması sistem açısından aşağıdaki faydaları sağlayabilir (Özveri, 2007, s:34),

- Firmaların politika ve stratejilerinin geliştirilmesine destek verme,
- Firma hedeflerinin çalışanlar tarafından sahiplenmesi,
- Çalışanların karar ve süreç geliştirme faaliyetlerine katılması,
- Çalışanların daha fazla tatmin olması ve kişisel gelişimlerini desteklemesi.

### **3.2.1.5 Karşılıklı Fayda Sağlayan Tedarikçi İlişkileri**

Kuruluşun can damarlarından biri olan tedarikçiler de aslında birer kuruluştur. Tedarikçilerin gelişmesi, onlardan ürün ve hizmet tedarik eden kuruluşları da olumlu etkileyecektir.

Firmalar tedarikçileri ile karşılıklı güvene, entegrasyona ve bilgi paylaşımına dayalı bir yarar ilişkisi ile daha etkin çalışır. Firmalar hem kendisi, hem de tedarikçileri için katma değer yaratan karşılıklı ilişkiler içinde olarak yaratıcılığı da teşvik etmiş olurlar. Tedarikçilerin seçiminden başlayarak değerlendirilmesi ve ilişkinin sona erdirilmesine kadar geçen süreç tanımlanmış ve bu ilişkin başarı ile

sürmesini sağlayacak katkılarla donatılmış olmalıdır. Bu ilkenin uygulanması aşağıdaki faaliyetleri içerir (Halis, 2004, s:139),

- Temel tedarikçilerin tanımlanması ve seçimi,
- Firmayı ve toplumu gözeterek kısa dönem kazançları ile uzun dönem itibarını hesaba katarak tedarikçi ilişkilerini oluşturmak,
- Açık ve net iletişim kurmak,
- Süreç ve ürünün, gelişimini ve gerçekleşmesini birleştirmek,
- Müşteri ihtiyaçlarının açıkça anlaşılmasını sağlamak,
- Gelecek planlarının ve bilgilerinin paylaşımını sağlamak,
- Tedarikçi iyileştirme ve başarılarını izlemek.

Tedarikçilerle ilişkilerin tanımlanmasının yararları aşağıdaki gibi sıralanabilir (Özveri, 2007, s:40),

- Tedarikçilerle stratejik iş birlikleri ve ortaklıklar geliştirerek rekabetçi ortamın yaratılması,
- Tedarikçilerin önceden katılımı ile daha rekabetçi amaç ve hedeflerin yerleştirilmesi,
- Güvenilir, zamanında, kusursuz teslimi garanti edecek şekilde tedarikçi ilişkilerinin yaratılması ve yönetilmesi,
- Tedarikçi eğitimleri ve birleşik iyileştirme çabalarıyla tedarikçi yeterliliğinin geliştirilmesi ve kuvvetlendirilmesi.

### **3.2.1.6 Süreç Yaklaşımı**

Süreç yaklaşımı, kuruluştaki bütün faaliyetleri süreç olarak görmek demektir. Bu yaklaşımda işletmenin bütün süreçleri belirlenir ve dokümanite edilir.

Kalite yönetim sisteminin etkili uygulanması için firmalar süreçlerini ve birbirleri ile olan etkileşimlerini tanımlamalı ve yönetmelidir. Bu sistem yalnızca ana süreci değil aynı zamanda destek süreçleri de kapsar. Süreçler, girdileri, kaynakları,

kontrolleri (kontrol kriteri ile kontrol yöntemi) ve çıktıları ile birlikte tanımlanırlar. Kaynakların kullanılması ve girdilerin çıktı şekline dönüştürülmesi için oluşturulan faaliyetler “süreç” olarak tanımlanır. Süreçlerin bir sistem olarak uygulanması, tanımlanması ve birbirleri ile etkileşimlerinin yönetilmesi “ süreç yaklaşımı” olarak tanımlanır (Özveri, 2007, s:34).

ISO 9001:2000 standardında uygulama ön planda tutulmakta olup, süreç yaklaşımı desteklenmektedir. Kaynakları kullanarak, girdileri çıktıya çeviren her faaliyet bir süreçtir. Bir sürecin çıktısı diğerinin girdisini oluşturmaktadır. Süreç yaklaşımı açısından aşağıdaki konular önemlidir (Özveri, 2007, s:34),

- Şartların anlaşılması ve karşılanması,
- Katma değer sağlayan süreçlerin dikkate alınması,
- Performans ve etkinlik sonuçlarının değerlendirilmesi,
- Somut ölçümler temel alınarak sürekli iyileştirmenin gerçekleştirilmesi.

Firmalar, birbirleriyle ilişkili tüm faaliyetlerini anlayıp, sistematik olarak yönetebildiği ölçüde başarılı olur. Süreç yaklaşımı ilkesini uygulayan firmalar ([http://www.polater.com.tr/devam.php?sub\\_page=1&page=bilgi\\_agaci&new\\_page=1](http://www.polater.com.tr/devam.php?sub_page=1&page=bilgi_agaci&new_page=1), 18/11/2006),

- Hedeflenen sonuçlara ulaşmak için süreçlerini tanımlamalı,
- Süreçlerin girdi ve çıktılarını tanımlamalı ve ölçülebilir hale getirmeli,
- Fonksiyon ve süreçler arası ilişkileri ortaya koymalı ve açıklığa kavuşturmalı,
- Süreçlere ilişkin olası riskleri, süreç akışını, sürecin tedarikçisi, müşterisi ve diğer paydaşlarının sürece olan etkisini belirlemeli,
- Sürecin yönetimi için yetki ve sorumlulukları açık, anlaşılır ve dengeli bir biçimde dağıtmalı,
- İç müşteri ve tedarikçileri ile diğer paydaşları ve onların beklentilerini tanımlamalı,

- Beklenen sonuçlara ulaşabilmek için sürecin aşamaları, faaliyetleri, akışı, kontrol ölçümleri, eğitim, ekipman, malzeme, bilgi ve yönetim ihtiyaçlarını saptamalıdır.

Faaliyetler ve ilgili kaynaklar bir süreç olarak yönetildiği zaman daha verimli olabilir. Firmaların bu amaçla süreçlerini tanımlaması ve aralarındaki ilişkileri ortaya koyması gerekmektedir. Böylelikle tüm süreçlerin girdi – çıktı sınırlamaları ile mekanizmaları açıkça belirlenmiş ve karşılıklı ilişkileri yönetilebilir biçimde konumlandırılmış olacaktır. Bu amaçlara ulaşılabilmesi için firmalar aşağıdaki yolları izleyebilir (Baş, 2004, s:2-3),

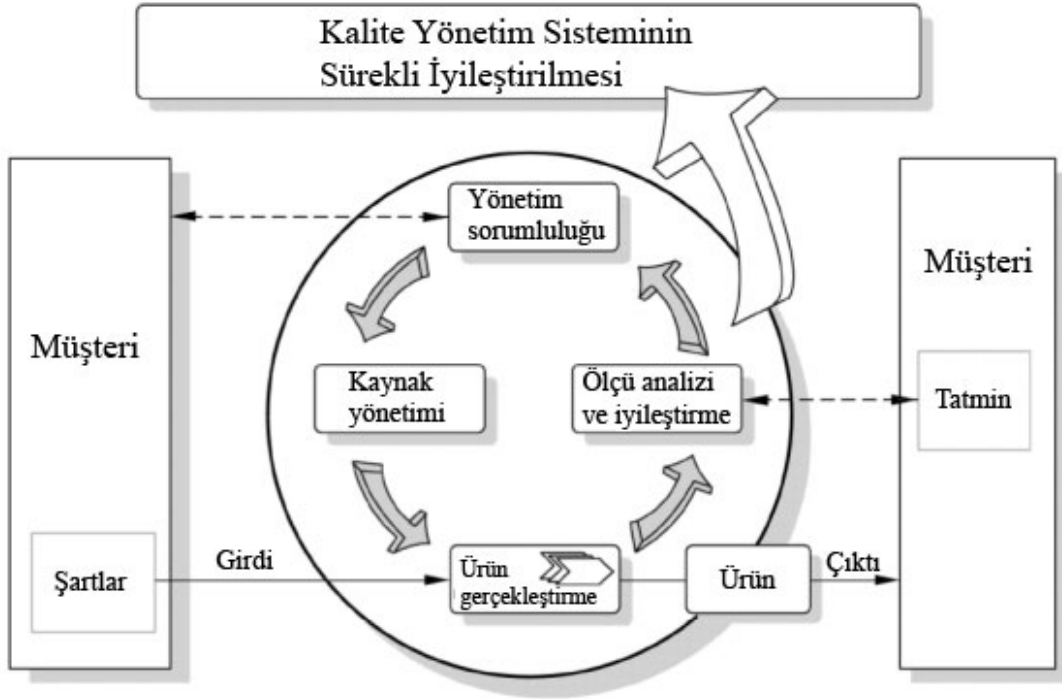
- Kalite yönetim sistemlerinin tümünü akış diyagramları ile ifade etmek,
- Tekrarları ve değer sağlamayan aşamaları iptal etmek,
- Her bir süreci, diğer süreçlerle etkileşimleri ile birlikte tanımlamak,
- Tüm süreçler arasında bir iç müşteri-tedarikçi ilişkisi oluşturmak.

Firmalar süreçleri gerçekleştirdiğinde aşağıdaki faydaları sağlayabilir (Özveri, 2007, s:36),

- Tüm işler için süreç yaklaşımının uygulanması ile daha az maliyet, hataların azalması, değişikliğin daha etkin kontrolü ve daha tahmin edici çıktılar oluşturabilir,
- İnsan kaynaklarının verimli çalışmasına imkan verir.

Kalite yönetim sistemlerini, süreçleri yeterince dikkate almadan oluşturan firmalar, ISO 9000'nin sağlayabileceği yararlardan çok azıyla yetinmek durumunda kalabilirler (Baş, 2004, s.2-3). Süreç tabanlı kalite yönetim sisteminde girdi müşteri şartlarıdır. Şekil 10 müşteri şartlarının karşılanması ile müşteri memnuniyetini arttırmak üzere süreç yaklaşımının uygulanması, geliştirilmesi ve kalite yönetim sisteminin etkinliğinin iyileştirilmesine yönelik PUKÖ döngüsünü göstermektedir.

**Şekil 10: Süreç Tabanlı Kalite Yönetim Sistemi Modeli**



**Kaynak: TSE EN ISO 9001:2000 Standardı, md:0.2**

### 3.2.1.7 Yönetim Sistem Yaklaşımı

Yönetim sistemi, firmaların amaçlarına erişmek için gereken tüm işleri yapabilmesini sağlamakta kullanılan süreçler olarak tanımlanabilir. Bu bakış açısı ile yönetim sistemi kapsamında belirli bir hedef için süreçler arası ilişki sistemini tanımlamak, anlamak ve yönetmek firmaların etkinliğini ve verimliliğini artırır. Süreçlerin bir sistem olarak tanımlanması, anlaşılması ve yönetilmesi; hedeflerin başarılmasına, etkinliğin ve verimliliğin artmasına katkı yapar. ISO 9000 KYS’de üst yönetimin sistem içindeki rolü ve sorumlulukları belirgin bir şekilde öne çıkmaktadır. Üst yönetim müşteri şartlarının belirlenmesi ve karşılanmasında birinci derece sorumludur (Baş, 2004, s:4). Süreçlerin tanımlanması çalışmalarında aşağıdaki konular dikkate alınabilir (Özveri, 2007, s:36),

- Belli bir hedefi etkileyen süreçlerin tanımlanması ve geliştirilmesi,
- Sistemin, hedefe ulaşmak için etkin şekilde yapılandırılması,

- Sistem süreçleri arasındaki bağımlılığın anlaşılması,
- Sistemin ölçüm ve değerlendirme yoluyla geliştirilmesi,
- Kaynak sınırlamalarının faaliyetler öncesi belirlenmesi.

Sistem yaklaşımının sağlayacağı faydalar aşağıdaki gibi sıralanabilir (Özveri, 2007, s:37),

- Süreç tabanlı rekabetçi planların oluşturulması,
- Her sürecin hedef ve amaçlarını firmanın hedefleriyle birleştirmesi,
- Problemlerin sebeplerini ve zamanında iyileştirme aksiyonlarını anlamaya götüren süreç etkinliğinin daha geniş bakış açısıyla görülmesi,
- Çapraz fonksiyonel engelleri azaltmak ve takım çalışmasını ortak hedeflere ulaştırması.

### **3.2.1.8 Liderlik**

Kalite yönetim sistemi uygulamasında yöneticiler anahtar konumda olduğundan, KYS ana kavramlarını ilk önce yöneticiler öğrenmelidirler. Yöneticilerin firmalarda liderlik fonksiyonunu gereği gibi üstlenmeleri beklenmektedir. Bu kapsamda yöneticiler, çalışanları KYS'yi öğrenmeye ikna etmeli ve KYS'nin faydalarını anlamalarına yardımcı olmalıdırlar. Liderlik vasfına sahip yöneticilerin temel sorumluluğu, firmanın hedef ve değerlerinin katılımcı bir şekilde belirlenmesine yardımcı olmak, KYS çalışmalarına görünür bir şekilde destek olmak ve çalışanları bu yönde motive etmektir. Liderler, çalışanlarla sürekli iletişim halinde olmalı, motive etmeli ve verimli bir ortam hazırlamak için gerekli desteği sağlamalıdır (Özveri, 2007, s:31). Liderlik ilkesini uygulayan firmalardaki liderler aşağıdaki konulara önem vermelidir ([http://www.polater.com.tr/devam.php?sub\\_page=1&page=bilgi\\_agaci&new\\_page=1](http://www.polater.com.tr/devam.php?sub_page=1&page=bilgi_agaci&new_page=1), 18/11/2006),

- Önleyici olmalı ve örnek bir liderlik sergilemeli,
- Firma çalışma ortamında bir olumsuzluk meydana geldiğinde anlamalı ve bu ortamı iyileştirecek şekilde çaba göstermeli,

- Tüm paydaşların talep ve beklentilerini göz önünde tutmalı,
- Firmanın geleceği ile ilgili açık ve anlaşılır vizyon ve misyon oluşturmali,
- Firmanın tüm seviyelerinde etik bir model olmalı ve paylaşılan değerler yaratmalı,
- Güven ortamı yaratmalı ve korkuyu ortadan kaldırmalı,
- Çalışanlara kaynak sağlamalı ve kendilerini gösterebilmeleri için sorumluluk vermeli,
- Çalışanlara ilham vermeli, onları yüreklendirmeli ve katkılarını değerlendirebilmeli,
- Açık ve dürüst iletişimi desteklemeli,
- Çalışanların eğitimine önem vermeli ve onlara önderlik etmelidir.

Firmaların amaç ve idare birliğini, çalışanların firma hedeflerinin başarılmasına tam olarak katılımını ve faaliyetlerin sağlıklı olarak sürdürülmesini sağlamak liderlerin görevidir. Firmanın kalite yönetim sistemin başarısında kritik bir rol oynayacak lider aşağıdaki eylemleri gerçekleştirebilmelidir (Özveri, 2007, s:31),

- Proaktif olma ve örneklerle idare,
- Dış çevredeki değişiklikleri anlamak ve tepki vermek,
- Müşteri, hissedarlar, tedarikçiler, yerel halk ve toplumu da içeren tüm fayda sağlayanların ihtiyaçlarını göz önüne almak,
- Firmada her seviyede paylaşılan değerleri ve etik rol modellerini oturtmak,
- Güven kurmak ve korkuyu bertaraf etmek,
- Çalışanlara gerekli kaynakları sağlamak,
- Açık ve dürüst iletişimi desteklemek.

Liderin yukarıda sözü edilen eylemleri gerçekleştirmesi firma açısından aşağıdaki faydaları yaratabilir (Özveri, 2007, s:32),

- Firmanın geleceğine ilişkin vizyon oluşturulması ve bu vizyonun çalışanlara yayılması,
- Vizyonun firmanın ölçülebilir amaç ve hedeflerine dönüştürülmesi,

- Firmanın hedeflerine ulaşabilmesi için çalışanların yetkilendirilmesi ve katılımlarının artırılması,
- Çalışanların kararlı hale getirilmesi, bilgilendirilmesi, yetkilendirilmesi ve motive edilmesi.

### **3.2.2 9001:2000 Standardının Maddeleri**

Standardın temel noktalarından biri süreç yaklaşımıdır. Standard kalite yönetim sisteminin, müşteri şartlarını karşılamak sureti ile müşteri tatminini artırmak için kalite yönetim sisteminin geliştirilmesi, uygulanması ve etkinliğinin iyileştirilmesinde süreç yaklaşımının benimsenmesini teşvik eder. Buradaki “süreç yaklaşımını teşvik eder” ifadesinden süreç yaklaşımının standardın zorunlu bir şartı olmadığı anlaşılmaktadır. Firmalar, standardın şartlarını karşılamak için istediği yaklaşım ve modeli seçmekte tamamen serbesttir ve bu serbestlik ISO 9001:2000 ile birlikte daha da arttırılmıştır (Baş, 2004, s:12). Bu bölümdeki maddeler, TSE'nin orijinal TS-EN-ISO 9001:2000 standardındaki maddelerin derlemesi yapılarak sunulmuştur.

#### **3.2.2.1 Giriş (Madde 0)**

##### **Genel (Madde 0.1)**

Bu başlıkta, standardın genel açıklaması, kullanım amacı, kimler tarafından kullanılabilceği ve kapsamı kısaca açıklanmaktadır. Standardın esnek yapısına da değinilmektedir.

##### **Proses Yaklaşımı (Madde 0.2)**

Bu maddede ISO 9001:2000'in temelini oluşturan proses yaklaşımına değinilmiştir. Proses yaklaşımına göre, işletmedeki bütün faaliyetler proses olarak belirlenir. Bu proseslerin sürekliliği ve iyileştirilmesi için PUKÖ (Planla, Uygula, Kontrol Et, Önlem Al) döngüsü önerilmiştir (Şekil 10).



### **ISO 9004 İle Bağlantılar (Madde 0.3)**

Bu başlıkta, ISO 9001 ve ISO 9004'ün bağlantısından bahsedilmiş ve karşılaştırılması yapılmıştır. ISO 9001'in standart şartlarını belirleme, ISO 9004'ün uygulamada daha geniş bir açıklayıcı, yani rehber niteliğinde olduğuna değinilmiş ve kullanım amaçları anlatılmıştır.

### **Diğer Yönetim Sistemleriyle Uyumluluk (Madde 0.4)**

Bu başlık altında ISO 9001'in ISO 14001:1996 ile aynı çizgiye getirilip uyumlaştırıldığı belirtilmektedir. Ayrıca diğer standartlarla bütünleştirilmesinin mümkün olduğu eklenmektedir.

#### **3.2.2.2 Kapsam (Madde 1)**

##### **Genel (Madde 1.1)**

Genel başlığında, bir kuruluşun bu standarda hangi durumlarda ihtiyaç duyduğu belirtilmiştir. Söz konusu durumlar şunlardır:

- Kuruluşun, müşterinin taleplerini, yürürlükteki mevzuatı ve ürünün yeterliliğini sürekli karşılama yeterliliğini gösterme ihtiyacı olduğu durumda.
- Müşteri memnuniyetinin sağlanması amacıyla, yürürlükteki mevzuatlara uyulduğu güvencesi ve bütün prosesleri kapsayarak sistemin sürekli iyileştirilerek verimliliğinin artırılması amacı olduğu durumda.

Bu başlıkta, standartta kullanılan ürün kelimesinin aynı zamanda hizmet kelimesini de karşıladığı ifade edilmektedir.

## **Uygulama (Madde 1.2)**

Uygulama başlığında, ISO 9001:2000 standardının her türdeki ve her kapasitedeki kuruluşa uygun olduğundan bahsedilmektedir. Standardın bazı şartlarının hariç tutulabileceği ve hariç tutmayla ilgili yerine getirilmesi gereken koşullar yer almaktadır.

### **3.2.2.3 Atıf Yapılan Standartlar (Madde 2)**

ISO 9001 standardında geçen terim ve tarifler için ISO 9000:2000 “Quality Management Systems-Fundamentals and vocabulary” ve çevirisi olan TS EN ISO 9000 “Kalite Yönetim Sistemleri-Temel kavramlar ve sözlük” standartlarına atıf yapılmıştır.

### **3.2.2.4 Terimler Ve Tarifler (Madde 3)**

Bu standartta ISO 9000’de geçen terim ve tariflerin uygulandığından bahsedilmiştir. ISO 9000:1994 baskısındaki “tedarikçi” terimi yerine “kuruluş”, “taşeron” yerine “tedarikçi” terimleri kullanılmıştır. Ayrıca, “ürün”ün aynı zamanda “hizmet”i ifade etmektedir.

### **3.2.2.5 Kalite Yönetim Sistemi (Madde 4)**

#### **Genel Şartlar (Madde 4.1)**

Bu bölümde, kalite yönetim sisteminin oluşturulması, uygulanması, sürekli iyileştirilmesi ve sürekliliğinin sağlanması için yapılması gerekenler maddeler halinde listelenmiştir. Bu maddelere göre kuruluş,

- Bütün prosesleri belirlemeli ve tanımlamalı,
- Proseslerin sırasını ve etkileşimlerini belirlemeli,

- Bu proseslerin kontrolünü ve etkinliğini sağlayacak kriter ve metotları belirlemeli,
- Proseslerin doğru çalışması ve izlenmesi için gerekli bilgi ve kaynağı hazırda bulunmasını sağlamalı,
- Proseslerin ölçüm ve analizini gerçekleştirmeli,
- Planlanan sonuçlara ulaşmak ve sürekli iyileştirme için gereken faaliyetleri uygulamalıdır.

Ayrıca bu maddeye göre, işletme bazı prosesleri dış kaynaklı hale getirebilir. Dış kaynaklı hale getirilen dokümanların kontrolünün sağlanmasının gerekliliğinden bahsedilmiştir.

## **Dokümantasyon Şartları (Madde 4.2)**

### **Genel (Madde 4.2.1)**

Oluşturulması gereken kalite yönetim sisteminin dokümantasyonunda hazırlanırken mutlaka yapılması gerekenler şunlardır,

- Kalite politikası ve kalite hedeflerinin dokümanları,
- Kalite El Kitabı,
- Standardın zorunlu kıldığı dokümante edilmiş prosedürler,
- Proseslerin planlanması, yürütülmesi ve kontrolü için gerekli duyulan dokümanlar,
- Standardın 4.2.4 maddesi gereği zorunlu kılınan kayıtlar.

Bu maddede dokümantasyonun kapasitesinin çeşitli kapasite ve faaliyet türüne göre değişebileceği ve istenilen ortam türünde saklanabileceğine değinilmektedir.

### **Kalite El Kitabı (Madde 4.2.2)**

Bu başlıkta, kuruluşun kalite el kitabı oluşturması şartı yer almaktadır. Oluşturulacak kalite el kitabında aşağıdakilerin mutlaka bulunması gerekir,

- Kalite yönetim sisteminin hangi kapsamda oluşturulduğu ve eğer bazı maddeler hariç tutulduysa, bunlarla ilgili açıklamalar,
- Kalite yönetim sistemi kapsamında dokümante haline getirilen prosedürler ve bunlara yapılan atıflar,
- Belirlenen prosesler arasındaki ilişkilerin açıklanması.

### **Dokümanların Kontrolü (Madde 4.2.3)**

Kalite yönetim sisteminde yer alan dokümanların kontrolü ile ilgili bir prosedür oluşturulmalıdır. Kayıtlar, özel doküman türü olduğu için 4.2.4 Kayıtların Kontrolü maddesine göre ayrı bir prosedür oluşturulmalıdır. Dokümanların kontrolüyle ilgili prosedür aşağıdakileri içermelidir,

- Dokümanların yayınlanmadan önce yeterlilik açısından onaylanması,
- Doküman gözden geçirme, güncelleme ve tekrar onayı,
- Dokümanlardaki değişikliklerin ve güncel revizyon durumlarının tanımlanması,
- Yürürlükte bulunan dokümanlarının, belirlenen kullanım noktalarında bulunabilirliğinin sağlanması,
- Dokümanların okunabilirliğini ve kolay tanımlanabilmesini sağlama,
- Dış kaynaklı dokümanların tanımlanması ve kontrollü dağıtımının yapılması,
- Güncel olmayan dokümanların istenmeyen kullanımlarını önleme ve herhangi bir nedenle saklanması durumunda, belirli bir tanımının bulunması.

### **Kayıtların Kontrolü (Madde 4.2.4)**

Kalite yönetim sisteminin uygunluğu ve etkinliğini sağlamak amacıyla, kayıtların oluşturulması ve muhafaza edilmesi gerekmektedir. Kayıtların okunabilir olarak kalması, kolayca ayırt edilebilir ve tekrar elde edilebilir olarak kalması gerekir. Kayıtların oluşturulması, muhafazası, tekrar elde edilmesi ve saklama süresi sonunda elden çıkarılmasıyla ilgili dokümanite edilmiş prosedür oluşturulmalıdır.

### **3.2.2.6 Yönetim Sorumluluğu (Madde 5)**

#### **Yönetimin Taahhüdü (Madde 5.1)**

Kalite yönetim sisteminin uygulanması, geliştirilmesi ve sürekli iyileştirilmesiyle ilgili, üst yönetimin taahhüdü gerekmektedir. Üst yönetim, taahhüdünü yerine getirmesi için aşağıdakileri yerine getirmelidir,

- Yasal şartlar kadar müşteri şartlarının da yerine getirilmesinin önemli olduğunu kurum içerisinde iletmesi,
- Kalite politikasını oluşturmalı,
- Kalite hedefleri belirlenmiş olmalı,
- Yönetim gözden geçirmeleri yapılmalı,
- Kaynak bulunabilirliğinin sağlanması.

#### **Müşteri Odaklılık (Madde 5.2)**

Bu madde müşteri tatmininin sağlanması amacıyla, müşteri şartlarının belirlenmiş ve yerine getirilmiş olmasının, üst yönetimin görevi olduğunu vurgular. Müşteri odaklılıkla ilgili koşullar 7.2.1 ve 8.2.1 numaralı maddelerde açıklanmıştır.

### **Kalite Politikası (Madde 5.3)**

Üst yönetimin görevlerinden biri de kalite politikasını oluşturmaktır. Kalite politikası ile ilgili üst yönetimin yerine getirmesi gereken şu koşullar bulunmaktadır,

- Kuruluşun amacına uygun olmalıdır.
- Oluşturulan kalite yönetim sisteminin sürekli iyileştirilmesi ve şartlara uygunluğu açısından bir taahhüt içermelidir.
- Oluşturulacak kalite hedefleri için bir çerçeve niteliği taşınmalıdır.
- Kalite politikası belirlendikten sonra, kuruluşun tüm seviyelerine iletmeli ve anlaşılmalıdır.
- Uygunluluğunun sürekliliği sağlanması için gözden geçirilmelidir.

Kalite politikası (Özveri, 2007, s:51),

- Üst yönetimin oluşturduğu vizyon ve strateji ile tutarlı olmalı,
- Kalite hedeflerinin firmada anlaşılması ve izlenmesine izin vermeli,
- Üst yönetimin kalite için taahhüdünü ve hedeflerin başarılması için yeterli kaynakların sağlandığını göstermeli,
- Üst yönetimin açık liderliği ile firmada kalite için bir taahhüdün geliştirilmesini amaçladığını ifade etmeli,
- Müşterilerin ve diğer ilgili tarafların ihtiyaçlarının ve beklentilerinin tatmini ile ilgili olarak sürekli iyileştirmeyi kapsamalıdır.

### **Planlama (Madde 5.4)**

#### **Kalite Hedefleri (Madde 5.4.1)**

Üst yönetim, kuruluşun politikası ile tutarlı olarak, madde 7.1'de belirtilen ürün şartlarını da karşılamak koşuluyla, kuruluşun fonksiyonları ve seviyelerinde kalite hedefleri oluşturmalıdır. Kalite hedefleri ölçülebilir nitelikte olmalıdır.

## **Kalite Yönetim Sisteminin Planlanması (Madde 5.4.2)**

Üst yönetim, kalite hedefleri de dahil olarak, madde 4.1'deki şartları karşılayabilecek bir kalite yönetim sistemini planlamalı ve planlanan bu sistemde değişiklikler planlanıp uygulandığında, sistem tutarlılığının sürdürülmesini sağlamalıdır.

## **Sorumluluk, Yetki Ve İletişim (Madde 5.5)**

### **Sorumluluk Ve Yetki (Madde 5.5.1)**

Kuruluştaki tüm çalışanların sorumlulukları, yetkileri, bu sorumluluk ve yetkiler arasındaki ilişkileri üst yönetim tarafından tanımlanmalıdır. Ayrıca kuruluştaki iletişimi sağlamak da üst yönetimin görevidir.

### **Yönetim Temsilcisi (Madde 5.5.2)**

Üst yönetim, kuruluş içindeki görevine ve sorumluluğuna bakmaksızın bir üyeyi, kalite yönetim temsilcisi olarak atmalıdır. Kalite yönetim temsilcisinin sorumlulukları şunlardır,

- Kalite yönetim sistemi için ihtiyaç duyulan proseslerin oluşturulmasını, uygulanmasını ve sürekliliğini sağlamak
- Kalite yönetim sisteminin performansı ve ihtiyaçları hakkında üst yönetimi bilgilendirmek
- Kuruluştaki tüm çalışanların müşteri şartlarının bilincinde olmasını sağlamak.

### **İç İletişim (Madde 5.5.3)**

Üst yönetim tarafından, kalite yönetim sisteminin etkinliğini sağlayacak şekilde iletişim proseslerinin oluşturulması ve gerçekleştirilmesi gerekmektedir.

Kalite yönetim sistemindeki süreçlerin ve bunların etkililiğinin tüm fonksiyonlarda ve seviyelerde kabul edildiğinden emin olmak için firmalarda iletişimin sağlanması istenmektedir. Kalite yönetim sisteminin etkinliği için firma içerisinde, süreçler arası iletişim kanalları ve iletişim süreçleri oluşturulmalıdır. İletişim faaliyetleri için aşağıdaki örnekler verilebilir (Özveri, 2007, s:53),

- Alanlarında uzman çalışanların iletişimi,
- Takım brifingleri ve diğer toplantılar,
- İlan tahtaları, dahili gazeteler/magazinler,
- Elektronik posta, web sayfaları gibi işitsel/görsel elektronik medya.

## **Yönetimin Gözden Geçirmesi (Madde 5.6)**

### **Genel (Madde 5.6.1)**

Üst yönetim, yeterliliğini, sürekli uygunluğunu ve etkinliğini sağlamak için belirli aralıklarla kalite yönetim sistemini değerlendirmelidir. Değerlendirme, sistemin iyileştirme fırsatlarını, kalite hedeflerini, kalite politikasını ve kalite yönetim sisteminin diğer ihtiyaçlarını kapsamalıdır. Yönetimin gözden geçirmeleri sonucunda elde edilen kayıtlar madde 4.2.4'e göre muhafaza edilmelidir.

### **Gözden Geçirme Girdisi (Madde 5.6.2)**

Yönetimin gözden geçirmesinde bulunması zorunlu girdiler şunlardır,

- Yapılan tetkik sonuçları,
- Müşterilerden sağlanan geri bildirim,
- Proses performansı ve ürün uygunluğu,
- Düzeltici ve önleyici faaliyetlerin durumu,
- Daha önce yapılan yönetimin gözden geçirme faaliyetinden devam eden takip faaliyetleri,
- Kalite yönetim sistemine etki edebilecek değişiklikler,



- İyileştirme önerileri.

### **Gözden Geçirme Çıktısı (Madde 5.6.3)**

Yönetimin gözden geçirme çıktısında mutlaka bulunması gerekenler şunlardır,

- İçerdiği proseslerle birlikte kalite yönetim sisteminin iyileştirilmesi,
- Ürünün müşteri şartlarına bağlı olarak iyileştirilmesi,
- Kaynak gereksinimleri.

### **3.2.2.7 Kaynak Yönetimi (Madde 6)**

#### **Kaynakların Sağlanması (Madde 6.1)**

Kuruluş,

- Kalite yönetim sisteminin uygulanması sürdürülmesi ve etkinliğinin iyileştirilmesi,
- Müşteri isteklerini karşılayarak müşterilerinin memnuniyetini artırmak için gerekli kaynakları saptamalı ve tedarik etmelidir.

Burada üst yönetime düşen görev, kalite yönetim sistemini uygulamak ve iyileştirmek için gerekli faaliyetleri belirlemek ve bu faaliyetlerin gerçekleşmesi için ihtiyaç duyulan kaynakları temin etmektir. Müşteri araştırmaları, kalite kontrol çalışmaları, istatistiksel çalışmalar, düzeltici ve önleyici faaliyetler, satış ve satış sonrası hizmetler, ürün tanıtımları, AR-GE çalışmaları, üretim planlaması, üretim geliştirmesi kaynak ihtiyacı duyulan faaliyetlerden bir kaçıdır (Baş, 2004, s:72).

## **İnsan Kaynakları (Madde 6.2)**

### **Genel (Madde 6.2.1)**

Ürün kalitesini etkileyebilecek işi yapan çalışanlar, uygun eğitim, öğrenim, beceri ve deneyim yönünden yeterli olmalıdır. Çalışanların yeterliliğine karar vermeden önce (Baş, 2004, s:74-75),

- Yapılacak işler tanımlanmalı,
- Bu işleri karşılayabilecek yetenekler listelenmeli,
- Bu bilgiler bir araya getirilerek, her bir işte görevlendirilebilecek çalışanın özellikleri oluşturulmalıdır.

### **Yeterlilik, Farkında Olma (Bilinç) ve Eğitim (Madde 6.2.2)**

Kuruluş, yeterlilik, kalite yönetim sisteminin sürekliliğini ve tutarlılığını sağlamak için şu koşulları yerine getirmelidir,

- Ürünün kalitesini etkileyen personelin sahip olması gereken yeterliliği sağlamalı,
- Eğitim ve benzeri ihtiyaçları sağlamak için önlemler almalı,
- Alınan önlemlerin etkin olup olmadığı değerlendirilmeli,
- Belirlenen kalite hedeflerinin başarılması için personelin nasıl katkıda bulunacaklarını belirlemeli, personelin yaptıklarının bilincinde olması sağlanmalı.
- Eğitim, öğretim, beceri ve deneyimle ilgili kayıtları madde 4.2.4'e göre muhafaza etmelidir.

Yönetim, firmanın etkili çalışması için gereken yetkinliğin var olmasını sağlamalıdır. Yönetim, şimdiki ve gelecekteki yetkinlik gereksinmelerini, firmada var olan yetkinlikle karşılaştırarak analiz etmelidir. Yetkinlik ihtiyacı saptanırken aşağıdaki konular dikkate alınmalıdır (Halis, 2004, s:180),

- Stratejik ve operasyonel planlar ve hedeflere göre gelecekte ortaya çıkabilecek talepler,
- Yönetim ve iş gücü ikame gereksinimlerinin önceden tahmini,
- Firmanın süreçleri, teçhizat ve araçlarındaki değişiklikler,
- Tanımlanan faaliyetleri gerçekleştirecek insanların bireysel yetkinliklerinin değerlendirilmesi,
- Firmayı ve ilgili tarafları etkileyen standartlar, yasa ve tüzük gerekliliklerin yerine getirilmesi.

### **Alt Yapı (Madde 6.3)**

Ürünün gerekli şartlara uygun olmasını sağlamak için, gerekli olan altyapı sağlanmalı ve sürdürülmelidir. Altyapı şunları kapsamalıdır,

- Binalar, çalışma alanları ve tesisler,
- Prosesle alakalı yazılım ve donanım,
- İletişim ve taşıma gibi destek hizmetler.

### **Çalışma Ortamı (Madde 6.4)**

Kuruluştaki ürün şartlarının karşılanmasını sağlamak için gerekli olan çalışma ortamı düzenlenmeli ve sürdürülmelidir. Uygun çalışma ortamı, verimliliği artırdığı gibi iş kazaları gibi olumsuz olayları önleme ve işe odaklanmayı artırma gibi olumlu katkılar sağlar.

Bu maddede ürün şartlarına uygunluğun sağlanması için gerekli çalışma ortamının belirlenmesi ve yönetilmesinden bahsedilmektedir. Çalışma ortamı, doğrudan ürün kalitesini olumsuz etkileyebileceği gibi, işi yapan personeli etkileyerek, ürün uygunluğunun sağlanması için olumsuz bir etki de yaratabilir (Özveri, 2007, s:57).

### **3.2.2.8 Ürün Gerçekleştirme (Madde 7)**

#### **Ürün Gerçekleştirmenin Planlanması (Madde 7.1)**

Kuruluşun ürün ve hizmet gerçekleştirmesini sağlamak amacıyla gerekli olan bütün prosesleri planlamalı ve geliştirmelidir. Ürün gerçekleştirmenin planlanması prosesi, madde 4.1’de bahsedildiği gibi diğer proseslerle uyumlu olmalıdır.

Kuruluş ürün gerçekleştirmeyi planlarken, uygun durumda aşağıdaki şartları sağlamalıdır,

- Ürün ve hizmet için kalite hedefleri ve şartlar,
- Proses ve dokümanların oluşturulması, ürünle ilgili kaynakların temini,
- Ürünün kabulü, geçerli kılma, muayene gibi faaliyetler için gerekli kriterler,
- Gerçekleştirilen ürünün şartları karşıladığına dair kayıtlar.

#### **Müşteri İle İlişkili Prosesler (Madde 7.2)**

##### **Ürüne Bağlı Şartların Belirlenmesi (Madde 7.2.1)**

Kuruluş, ürünle ilgili yasal ve mevzuat şartlarını, teslim öncesi ve sonrası şartları da içeren müşteri tarafından belirlenen şartları, müşteri tarafından belirlenmeyen ancak bilinen kullanım için belirlenen şartları, ayrıca kuruluşun kendi belirlediği şartları belirlemelidir.

##### **Ürüne Bağlı Şartların Gözden Geçirilmesi (Madde 7.2.2)**

Kuruluş, müşteriye ürünü sağlamayı taahhüt etmeden önce, ürüne bağlı şartları gözden geçirmelidir. Ayrıca kuruluş,

- Ürün şartlarının tarifinin yapılmasını,

- Daha önce belirtilen şartlarla sözleşmede veya siparişte belirlenen şartlar arasındaki farkların çözümlenmesini,
- Belirlenen şartları karşılama yeterliliğine sahip olmasını sağlamalıdır.

Gözden geçirme sonuçlarının kayıtları ve gözden geçirmeden kaynaklanan faaliyetler sürdürülmelidir. Müşterinin şartlarının dokümanede edilmiş beyanını sağlamadığı durumlarda müşteri şartları, kabulden önce firma tarafından teyit edilmelidir. Ürün şartları değiştiğinde firma, ilgili dokümanları yenilenmesi ve ilgili personelin bu değişikliklerden haber edilmesi sağlanmalıdır (Özveri, 2007, s:58).

### **Müşteri İle İletişim (Madde 7.2.3)**

Bu madde kuruluşun, ürün bilgisi, tadiller de dahil olarak başvurular, sözleşmeler, sipariş alımları, müşteri şikayetleri de dahil olarak, müşteri geri beslemesi ile ilgili müşteri ile iletişimini etkin biçimde sağlamasını ve uygulamasını şart koşmaktadır.

Müşteriler ile iletişim için aşağıdaki dokümanlar kullanılabilir (Şale, 2004, s:122),

- Gelen/giden evrak listesi,
- Telefon görüşmesi listesi,
- Görüşme kayıtları,
- Müşteri şikayet dokümanları,
- Müşterilere ait bilgi kayıtları.

### **Tasarım Ve Geliştirme (Madde 7.3)**

#### **Tasarım Ve Geliştirme Planlaması (Madde 7.3.1)**

Kuruluş, aşağıdakileri de tayin edecek şekilde ürün tasarım ve geliştirmesini planlamalıdır,

- Tasarım ve geliştirme aşamaları,
- Tasarım ve geliştirmenin bütün aşamalarını doğrulama ve geçerli kılma,
- Tasarım ve geliştirme faaliyetlerinin sorumluluk ve yetkileri.

Kuruluş, ürün tasarım ve geliştirmesine görev alanlarla ilgili sorumluluk ve iletişimi etkin biçimde gerçekleştirmek için gerekli olan arabirimleri yönetmelidir. Planlama çıktısı, uygun olduğunda tasarım ve geliştirme geliştikçe güncellenmelidir.

### **Tasarım Ve Geliştirme Girdileri (Madde 7.3.2)**

Fonksiyon ve performans şartları, uygun olan yasal ve mevzuat şartları, uygun olduğunda önceki benzer tasarımların bilgileri, tasarım geliştirmeyi ilgilendiren diğer şartlardan oluşan ürün şartları girdileri belirlenmeli ve madde 4.2.4'e göre muhafaza edilmelidir. Bu girdilerin yeterliliği gözden geçirilmelidir. Şartlar birbirleri ile tutarlı ve açıkça tanımlanmış olmalıdır.

Dış ve iç girdiler aşağıdaki gibidir (<http://www.kalite.saglik.gov.tr/kalitekavramlari/iso9000.htm>, 27/12/2006),

Dış girdiler,

- Müşteri veya pazar ihtiyaçları ve beklentileri,
- Diğer ilgili tarafların ihtiyaçları ve beklentileri,
- Tedarikçinin katkıları,
- Tasarım ve geliştirmeyi başarmak için kullanıcı girdisi,
- İlgili yasa ve mevzuat şartlarında değişiklikler,
- Uluslar arası veya ulusal standartlar.

İç girdiler,

- Politikalar ve hedefler,

- Süreçlerin çıktıları dahil, firmadaki çalışanların ihtiyaçları ve beklentileri,
- Teknolojik geliştirmeler,
- Geçmiş tecrübelerden gelen bilgi ve geri beslemesi,
- Mevcut süreçler, ürünlerin kayıtlar ve verileri.

### **Tasarım Ve Geliştirme Çıktıları (Madde 7.3.3)**

Tasarım ve geliştirme çıktıları, tasarım ve geliştirme girdisinin doğrulayabilecek bir formda elde edilir ve dağıtımdan önce onaylanır. Tasarım ve geliştirme çıktılarında bulunması gereken özellikler şunlardır,

- Tasarım ve geliştirme girdilerinin şartlarını karşılamak,
- Satın alma, üretim ve hizmet sunulması ile ilgili bilgi sağlamak,
- Ürün kabul kriterlerini içermek ve atıfta bulunmak,
- Ürünün güvenli ve uygun kullanımı amacıyla gerekli olan bütün özellikleri saptamak.

Tasarım çıktıları aşağıdaki gibidir (Şale, 2004, s:133),

- Satın alma bölümünün ihtiyaç duyduğu kalemler,
- Ürünün seri imalatı için gerekli yöntem, bilgi vb. ihtiyaçlar,
- Satış ve sonrası kullanımda ihtiyaç duyulabilecek detayların tespit edilerek ürüne yansıtılabilmesi.

### **Tasarım Ve Geliştirmenin Gözden Geçirilmesi (Madde 7.3.4)**

Madde 7.3.1 tasarım ve geliştirmenin planlanması aşamasında, tasarım ve geliştirmenin gözden geçirilmesi aşağıdaki amaçlara uygun olarak planlı şekilde gerçekleştirilmelidir,

- Tasarım ve geliştirme sonuçlarının şartları karşılamasının değerlendirilmesi,
- Gerekli düzeltici ve önleyici faaliyetlerin gerçekleştirilmesi.

Gözden geçirme sonuçları ve gerekli kayıtlar madde 4.2.4'e göre muhafaza edilmelidir. Gözden geçirme faaliyetlerine, ilgili fonksiyonların temsilcilerini de içermelidir.

#### **Tasarım Ve Geliştirme Doğrulaması (Madde 7.3.5)**

Tasarım ve geliştirme çıktılarının, düzenlenen planlara göre girdi şartlarının karşılanıp karşılanmadığı bakımından doğrulaması gerekir. Doğrulama ile ilgili kayıtlar madde 4.2.4'e göre muhafaza edilmelidir.

#### **Tasarım Ve Geliştirmenin Geçerli Kılınması (geçerliliği) (Madde 7.3.6)**

Planlanan düzenlemelere göre nihai ürünün amaçlanan şartları karşılayacak nitelikte tasarım ve geliştirmesinin geçerliliği yapılmalıdır. Uygun durumda, ürünün tesliminden ve oluşturulmasından önce ürün geçerliliği tanımlanmalı, ilgili kayıtlar madde 4.2.4'e göre muhafaza edilmelidir.

#### **Tasarım Ve Geliştirme Değişikliklerinin Kontrolü (Madde 7.3.7)**

Ürünün tasarım ve geliştirmesinde meydana gelen değişiklikler, kayıt altına alınmalı ve muhafaza edilmelidir. Değişiklikler uygulanmadan önce gözden geçirilmeli, doğrulanmalı ve uygun bulunursa geçerli kılınıp onaylanmalıdır. Tasarım ve geliştirme değişikliklerinin daha önce teslim edilen ürün üzerindeki etkisi değerlendirilmelidir.

#### **Satın Alma (Madde 7.4)**

##### **Satın Alma Prosesi (Madde 7.4.1)**

Satın alma prosesine göre, kuruluş, satın aldığı ürünün şartlara uygunluğunu sağlamalıdır. Bu amaçla kuruluş, satın alınacak ürünün belirlenen şartları



karşılmasını sağlayacak kriterler oluşturmalı ve satın alma prosesinde bunları uygulamalıdır. Seçme, değerlendirme ve tekrar değerlendirme için oluşturulan kriterlere göre yapılan değerlendirmelerin kayıtları madde 4.2.4'e göre muhafaza edilmelidir.

#### **Satın Alma Bilgisi (Madde 7.4.2)**

Satın alma bilgisi, ürünü yeterince açıklamalı, önceden belirlenen satın alma şartlarını içermelidir. Kuruluş ise bu şartların yeterliliğini sağlamalıdır.

#### **Satın Alınan Ürünün Doğrulanması (Madde 7.4.3)**

Bu maddede kuruluşun satın alınacak ürünün belirlenmesi ve satın alma şartlarını karşılamak için çeşitli kontrolleri gerçekleştirmesi ifade edilmektedir.

#### **Üretim Ve Hizmetin Sağlanması (Sunulması) (Madde 7.5)**

##### **Üretim Ve Hizmet Sağlamanın Kontrolü (Madde 7.5.1)**

Standardın bu maddesinde üretim ve hizmet sağlanmasının planlanması ve yürütülmesinin kontrollü koşullar altında yapılması gerektiği anlatılmaktadır. Kontrollü koşullar, ürün için açıklamanın bulunabilirliği, çalışma talimatlarını, uygun donanım kullanımı, izleme ve ölçme faaliyetleri, teslimat öncesi ve sonrası gibi bilgileri kapsamalıdır.

##### **Üretim Ve Hizmet Sağlanması İçin Proseslerin Geçerliliği (Madde 7.5.2)**

Üretim ve hizmetin bir sonraki izleme ve ölçmede doğrulanamadığı yerlerde, üretim ve hizmet sağlama prosesleri geçerli kılınmalıdır. Geçerli kılma prosesinde şu düzenlemelerin yapılması gerekmektedir,

- Söz konusu proseslerin gözden geçirilmesi ve onaylaması için oluşturulmuş kriterler,
- Donanım ve personel yeterliliğinin doğrulanması,
- Çeşitli metot ve prosedürlerin kullanımı,
- Kayıtların kontrolü
- Yeniden geçerli kılma

### **Tanımlama Ve İzlenebilirlik (Madde 7.5.3)**

Tanımlama ve izlenebilirlik, ürün ve hizmetin oluşturulması sırasında hangi süreçlerden geçtiğini, olası hataları ve hata tespitini kolaylaştıran bir yöntemdir. Bu maddede kuruluşun, ürünün gerçekleştirilmesi boyunca uygulanan bütün faaliyetlerin belirlenmesini, takibini ve madde 4.2.4'e göre kayıt altına alınmasını belirtmektedir.

### **Müşteri Malı (varlığı) (Madde 7.5.4)**

Kuruluş, müşterilere ait her türlü mal varlığını kendisinde bulunduğu süreçte özenle saklamalıdır. Müşteri malına zarar gelirse, kaybolursa veya başka herhangi bir problem ortaya çıkarsa, bu durum müşteriye bildirilmeli ve kayıtlar muhafaza edilmelidir. Müşteri varlığı eşya gibi somut varlıklar olabileceği gibi, telif hakkı gibi soyut varlık da olabilir.

### **Ürünün Korunması (Madde 7.5.5)**

Bu maddede, ürünün teslim edilinceye kadar muhafaza edilmesi ve uygunluğunun korunması ifade edilmektedir. Ürünün korunması, ürün ve ürünü teşkil eden bütün parçalar için, taşıma, depolama gibi faaliyetlerin tümünü içerir.

### **İzleme Ve Ölçme Cihazlarının Kontrolü (Madde 7.6)**

Bu maddeye göre kuruluş, taahhüt ettiği takdirde gerekli izleme ölçmeyi ve bununla ilgili teçhizatları sağlamalıdır. İzleme ve ölçmeyle ilgili gerekli şartlara

uygunluęu saęlayacak prosesler oluřturmalıdır. Bu maddede ayrıca, gerekli olduęunda geęerli sonuların saęlanması iin izleme tehizatıyla ilgili yerine getirmesi gereken kořullar yer almaktadır. İzleme ve lme iřlemi iin bilgisayar yazılımı kullanıldıęında, amalanan iřlevi yerine getirme yeteneęi teyit edilmeli ve bu mmknse ilk kullanımdan nce yapılmalıdır.

### **3.2.2.9 lme, Analiz Ve İyileřtirme (Madde 8)**

#### **Genel (Madde 8.1)**

Kuruluř, rnn ve kalite ynetim sisteminin uygunluęunu saęlamak ve kalite ynetim sisteminin srekli iyileřmesini saęlamak amacıyla gerekli olan lme, analiz ve geliřtirme prosesini planlayıp uygulamaya geirmelidir. lme analiz ve iyileřtirme iin belirlenen faaliyetler istatistiksel teknikleri ierir.

#### **İzleme Ve lme (Madde 8.2)**

##### **Mřteri Memnuniyeti (Madde 8.2.1)**

Kuruluř, mřteri řartlarının ne derece karřılanıp karřılanmadıęı ve mřterilerin algılaması gibi mřteriyle ilgili bilgileri takip etmelidir. Bu bilgilerden yararlanmak iin eřitli metotlar belirlemelidir.

##### **İ Tetkik (Madde 8.2.2)**

İ tetkik, kalite ynetim sisteminde planlanan dzenlemelerin doęru iřleyip iřlemedięini ve řartlara uygunluęunu belirlemek amacıyla gerekleřtirilen bir faaliyettir. Bu faaliyet sonucu ortaya ıkacak sonular, st ynetim tarafından yorumlanıp analiz edilir, gerekli nlemler ve iyileřtirmeler yapılır. Tetkikiler kendi iřlerini tetkik etmemelidir. Tetkik sonunda elde edilen kayıtlar madde 4.2.4'e gre muhafaza edilir.

### **Proseslerin İzlenmesi Ve Ölçülmesi (Madde 8.2.3)**

Kuruluş, planlanan sonuçlara ulaşma amacıyla oluşturulan proseslerin yeterliliğini göstermek için kalite yönetim sisteminin ölçüm ve analizini gerçekleştirecek uygun metotları belirlemelidir. Planlanan sonuçlara ulaşamadığı durumlarda ise gerektiğinde düzeltici faaliyetler ve iyileştirmeler yapılmalıdır.

### **Ürünün İzlenmesi Ve Ölçülmesi (Madde 8.2.4)**

Kuruluş, nihai ürünün şartları yerine getirip getirmediğini belirlemek amacıyla kontrolünü yapmalıdır. Doğrulamalar, planlanan düzenlemelere göre gerçekleştirilmeli, sonuçlar madde 4.2.4'e göre muhafaza edilmelidir.

### **Uygun Olmayan Ürünün Kontrolü (Madde 8.3)**

Şartlara uygun olmayan nihai ürünün yanlışlıkla müşteriye teslim edilmesinin veya kullanılmasının önüne geçmek amacıyla, uygun aralıklarla ürün kontrolleri gerçekleştirilmelidir. Kontrollere ilişkin faaliyet ve sorumluluklar için dokümante edilmiş prosedür oluşturulmalıdır. Uygun olmayan ürünün, yanlışlıkla müşteriye teslim edilmesi veya kullanılması durumunda ortaya çıkacak etkilere karşı da önlemler alınmalıdır. Uygunsuzlukla ilgili bütün kayıtlar madde 4.2.4'e göre muhafaza edilmelidir.

### **Veri Analizi (Madde 8.4)**

Standardın bazı maddelerinde, kuruluşun elde ettiği verilerin analiz edilip bilgiye dönüştürülmesi ifade edilmektedir. Veri analizi, kalite yönetim sisteminin iyileştirilmesi ve sürekli gelişimi amacıyla elde edilen bu verilerin analizinin yapılmasıdır. Eldeki verilerin yararlı bilgiler haline getirilmesi, yönetim tarafından yorumlanıp bu sayede daha doğru kararlar almasını sağlar. Veri analizi, gerçeklere dayalı karar verme prensibinin temelini oluşturmaktadır.

## **İyileştirme (Madde 8.5)**

### **Sürekli İyileştirme (Madde 8.5.1)**

Sürekli iyileştirme, kalite yönetim sisteminin temel hedeflerinden biridir. Buna göre kurum, kalite yönetim sisteminin etkinliğini, kalite politikasını, hedeflerini, veri analizi, iç tetkik sonuçları, yönetim gözden geçirme sonuçları ile sürekli iyileştirmeyi sağlamalıdır.

### **Düzeltilici Faaliyet (Madde 8.5.2)**

Düzeltilici faaliyet, tespit edilen bir uygunsuzluğun sebebini ortadan kaldırmayı amaçlar. Kuruluşlar, düzeltilici faaliyetleri gerçekleştirmek için prosedür oluşturmalıdır, bu madde zorunlu prosedürlerden biridir. Düzeltilici faaliyet, uygunsuzluğun nedenlerinin belirlenmesi, ortadan kaldırılması ve tekrarının önlenmesi şartlarını taşımalıdır. Düzeltilici faaliyet sonucu elde edilen kayıtlar madde 4.2.4'e göre muhafaza edilmelidir.

### **Önleyici Faaliyetler (Madde 8.5.3)**

Kuruluş, potansiyel uygunsuzlukları önlemek için çeşitli önlemler almalıdır. Bunlar, önleyici faaliyetlerdir ve zorunlu prosedürler arasında yer almaktadır. Önleyici faaliyetler ile ilgili prosedür, potansiyel uygunsuzlukların belirlenmesi, nedenlerinin saptanması, gerekli önlemlerin belirlenmesi, alınan tedbirlerin sonuçlarının madde 4.2.4'e göre muhafaza edilmesi ve sonuçların gözden geçirilmesini içermelidir.

## 4 WEB TABANLI KALİTE YÖNETİM SİSTEMİ UYGULAMASI

### 4.1 Uygulamanın Amacı Ve Kapsamı

Web tabanlı kalite yönetim sistemi geliştirilmesinin amacı, bu sistemi uygulayan işletmelerde kırtasiye masraflarını azaltmak, işlevselliği artırmak, kalite yönetim temsilcisi üzerindeki yükü azaltmak ve dolayısıyla bu görevi üstlenen çalışanın daha verimli çalışabilmesini sağlamaktır. Sistemin web tabanlı oluşu, çalışanların birbirlerinden uzak mesafelerde olduğu durumlarda zaman ve performans kayıplarını engellemektedir. Dokümanların oluşturulması, revizyonu, onaylaması ve daha sonra dağıtımının yapılması, sıkıntılı ve zaman alıcı bir süreçtir. Bu durum, çalışan sayısı az olan kuruluşlar için sorun gibi görünmese de, daha kalabalık kuruluşlarda her yeni doküman oluşturulmasında, revizyon olduğu durumlarda da geçerliliğini yitirmiş dokümanın toplanıp yenilerinin dağıtılması, kalite yönetim sisteminin kuruluşu hantallaştırdığının düşünülmesine neden olmaktadır. Kalite yönetim sisteminin web ortamında yürütülmesi ile bu işlemler otomatik hale getirilip, doküman toplama ve dağıtım işlerinin otomatik olarak gerçekleşmesi planlanmaktadır.

Web tabanlı kalite yönetim sisteminin amaçlanan faydaları listelenecek olursa,

- Dokümanların etkin kontrolü ile kalite yönetim temsilcisi üzerindeki yükü azaltmak, güncellenen dokümanların toplanıp yenilerinin dağıtılması işleminin, otomatik güncelleme sayesinde tamamen ortadan kalkması,
- Üyelik ve yetkilendirme ile çalışanların rol tabanlı erişimleri sayesinde kişilerin görevlerini daha iyi yapabilmeleri, kalite yönetim temsilcisinin yetki vermesi sayesinde hangi bölümlerde ne kadar yetkileri olduğunu açıkça bilmeleri,

- Kuruluşa yeni katılan veya sahip olduğu unvanları değişen çalışanın, kalite yönetim temsilcisinin yazılımdaki unvanları ayarlaması ile bütün yetki ve sorumluluklarının otomatik güncellenmesi,
- Mevcut dokümanların arşivlerle her an karşılaştırılabilme imkanı sayesinde, yapılan değişiklik geçmişinin görülebilmesi,
- Modüler yapı sayesinde ihtiyaç duyulan diğer modüllerin de sonradan eklenip işleyişe katkıda bulunabilmesi, kuruluşa özgü modüllerin oluşturulabilmesi,
- Kırtasiye masraflarını en aza indirmek gibi birçok fayda sayılabilir.

Bu kapsamda, bir organizasyonun kalite yönetim sisteminin dokümantasyonu incelenmiş, buna uygun şekilde analiz, tasarım ve programlama yapılmıştır.

## 4.2 Gereksinimlerin Belirlenmesi

Web tabanlı kalite yönetim sistemini oluşturulmasında ilk basamak, gereksinimlerin belirlenmesi olacaktır. Gereksinimler; standarda bağlı gereksinimler, yazılımı kullanacak kişilerin talepleri, teknolojinin kullanımının beraberinde getirdiği şartlara yönelik gereksinimlerdir. Bu gereksinimlerden başlıcaları şöyledir,

- Standarda bağlı gereksinimler: Bu gereksinimler, dokümantasyonun sağlanması, kayıtların belirlenen süre kadar muhafaza edilmesi, dokümantasyon üzerinde kullanıcıların yetkilendirilmesi, revizyon talepleri, doküman oluşturma ve onay gibi gereksinimlerdir. Standardın 4.2.3 maddesinde dokümanların kontrolü ile ilgili bilgilere ulaşılabilir.
- Müşteri talepleri: Sistemi kullanacak olan hedef kitle, kullanım kolaylığı sağlayacak taleplerde bulunabilir. Bu talepler, işlevsel, görsel ve güvenlik talepleri olabilir. Sistemin güvenliğini sağlamak, programcı ve donanım uzmanları kadar, sistemin kullanıcılarına da bağlıdır. Örneğin, sistem yöneticisi, herhangi bir nedenle şifresinin öğrenilmesi durumunda, sisteme zarar verici işler yapılmasının önlenmesi için önemli işlevlerde cep telefonlarındaki pin kodu gibi ayrı bir işlevsellik de talep edebilir. Bu pin koduyla, önemli bilgilerin silinmesi sırasında uygulamanın bir adımı olarak

araya eklenebilir. Ya da müşteri odaklılık ilkesine istinaden, iç müşteri olan çalışanlarının, sisteme kayıt olduklarında doğum tarihlerinin depolanmasını, doğum günü olan çalışanların kutlanması için mecburi olmayan fakat faydalı bir talepte de bulunabilir.

- Teknolojik şartlar: Bu şartlar, genel olarak teknolojik gelişmelere, piyasa şartlarına, şirketin alım gücüne ve yapılan işlerin türüne bağlıdır. Kuruluş, eğer teknoloji ağırlıklı çalışıyorsa ve hali hazırda kurumsal yazılım kullanıyorsa, kalite yönetim sistemi yazılımına daha kolay geçebilecektir. Kuruluş, söz konusu anda kurumsal kaynak planlama yazılımı tarzı bir program kullanıyor olabilir. Bu durumda entegrasyon da talep edilebilir.

### **4.3 Analiz Ve Tasarım**

Bu aşamada kullanılacak araç ve tekniklerin belirlenmesi, modüllerin tasarımı ve oluşturulması yapılacaktır.

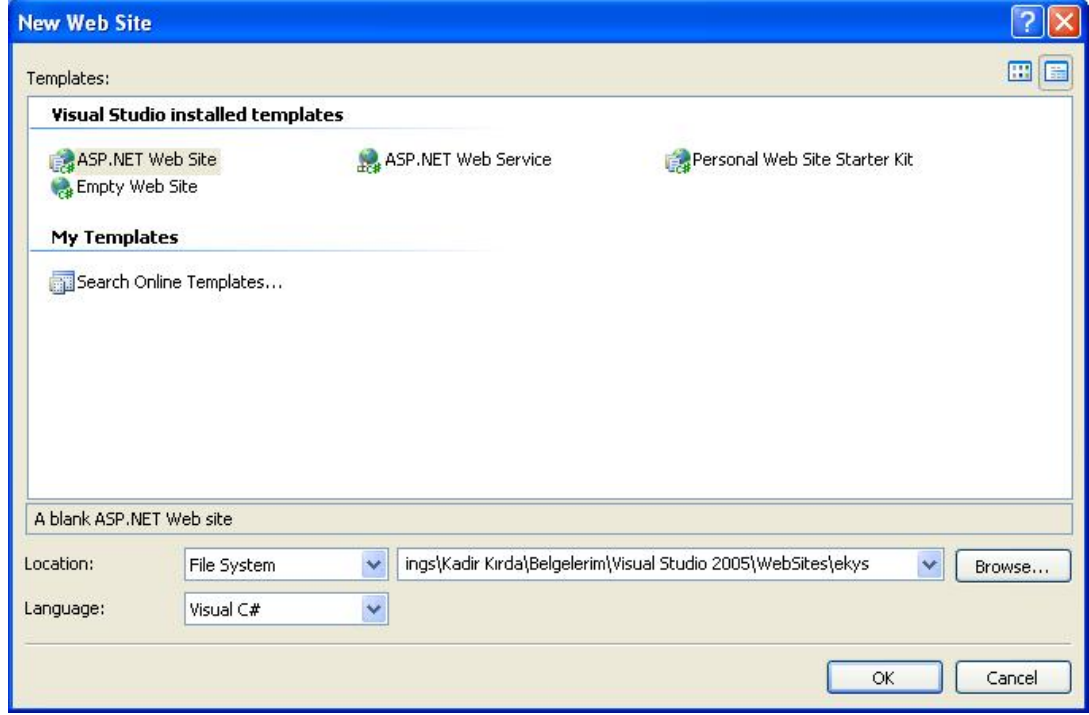
#### **4.3.1 Temel Yapı**

Sistem tasarlanıp oluşturulurken, modüllere göre teker teker ele alınacaktır. Ancak ana çatıyı oluşturan ayarların ilk başta yapılması, diğer tüm modülleri etkileyeceği için gereklidir. Temel olarak yapılacak ayarlar, projede kullanılacak programın ayarları, veritabanı bağlantıları, sınıf tasarım şekli, yani katmanlara göre yazılım oluşturuluyorsa katmanların ve klasör sisteminin belirlenmesi gibi ayarlardır. Tasarım için Visual Studio'nun ücretsiz sürümü olan Visual Web Developer (VWD) 2005 Express Edition, veritabanı yönetimi için de SQL Server Management Studio Express veritabanı yönetim aracı kullanılacaktır. Projenin oluşturulması için VWD'de File menüsünden "New Web Site..." seçilir. Açılan pencerede (Şekil 11) temalardan "ASP.NET Web Site", Location menüsünde dosya sistemi anlamına gelen "File System" seçilip, ilgili proje klasörü belirlenir. Language kısmında dil olarak, yazılımcı hangi dilde daha iyiye onu seçebilir. Farklı dillerde programcıların bir arada program geliştirmelerini sağlayan ortak dil çalışma özelliği desteği bu



aşamada görülebilir. “Ok” düğmesi tıklandığında yeni proje oluşturulmak üzere hazırlanır.

**Şekil 11: Yeni Bir Web Sitesi Projesi Oluşturma Sayfasının Ekran Görünümü**

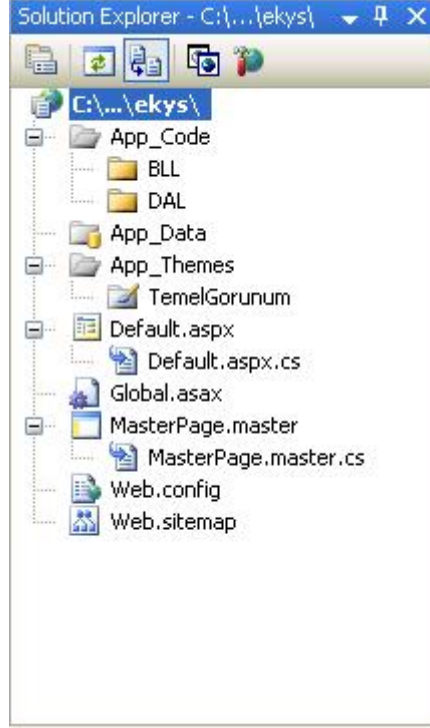


Oluşturulan projenin klasör yapısı “Solution Explorer” penceresinde görülebilir (Şekil 12). Solution Explorer’da belirlenmesi gereken bazı özel klasörler ve temel dosyalar bulunur. Bu klasörler resimde görüldüğü gibi App\_Data, App\_Code, App\_Themes klasörleridir. Eğer harici bir kod kütüphanesi alınacaksa, Bin klasörü de oluşturulur.

Şekil 12’de görülen klasörlerin ve dosyaların açıklamaları aşağıdaki gibidir,

**App\_Code:** ASP.NET 2.0 öncesi sürümlerde, yazılan kodların derlenip bin klasörüne yüklenmesi gerekiyordu. 2.0 sürümüyle gelen yeniliklerden biri derlenmemiş kodların olduğu gibi sunucuya yüklenerek çalıştırılabilmesi imkanidir. Bu da kodların güncellenmesini kolaylaştırmakta ve tasarım aşamasında zaman kazandırmaktadır.

**Şekil 12: Projenin Temel Klasör Yapısı**



**App\_Data:** Verilerin saklandığı klasörü oluşturur. Adres satırından girilerek erişilemeyen özel klasörlerden biridir. Access, XML gibi veritabanları bu klasör altında güvenle saklanabilir. Önemli dosyaları saklamak için de ideal bir klasördür. Dosyalar, adres satırından doğrudan girildiğinde erişilemez. Çünkü özel klasörlere erişim kısıtlıdır. Ancak, oluşturulacak arabirim vasıtasıyla kaydetme ve görüntüleme yapılabilir. Bu sayede rollere göre kısıtlama da getirilebilir.

**App\_Themes:** Sitenin temaları bu klasör altında yer alır. Temalar çok değişik amaçlarla oluşturulabilir. Örneğin, ekran görünümü için ayrı, yazıcı için ayrı, mobil bağlantı için ayrı tema oluşturulabilir. Bu klasör sayesinde temalar arasında değişiklik yapmak programatik olarak mümkün kılınmıştır.

**Default.aspx:** Varsayılan giriş sayfasıdır. Aynı isimde ve cs uzantılı dosyalar ise “CodeBehind” özelliği denen, arka plan kod dosyalarıdır. Dosya uzantısının cs olması sayfanın C# ile, vb olması visual basic ile kodlandığını gösterir.

**Global.asax:** Uygulamanın başlaması, bitişi, oturum başlaması ve bitişi gibi durumlarda site için genel olarak yapılacak işlemlerin belirlenmesini sağlar. Örneğin, bir kullanıcı sisteme giriş yaptığında içerikleri ne kadar görüntülense görüntülesin, sadece bir oturum sayılmaktadır. Bu dosyanın bir diğer özelliği, ortaya çıkan hatalarda ne yapılması gerektiğinin belirlenebilmesidir.

**MasterPage.master:** Görünüm için genel iskeleti oluşturur. Diğer sayfalar için çatı görevi görür. Temalar ise bu iskeleti şekillendirir.

**Web.config:** Site ile ilgili ayarların yapıldığı dosyadır. Veritabanı bağlantısı, üyelik, yetkilendirme ve profil ayarları, oturum süresi, sayfa temaları gibi sayısız ayar bu dosyadan yapılabilir.

**Web.sitemap:** Dosya yapısına uygun şekilde oluşturulduğunda programda bulunan navigation bölümündeki araçlar için temel kaynak görevi görür. Bir başka özelliği de, rollere göre görüntülemenin sınırlandırılabilmesidir. Örneğin “Yönetim” sayfasının bağlantısını sadece yönetici ve editör haklarına sahip olanların görmesi sağlanabilir.

App\_Code klasörünün BLL ve DAL adında iki klasör içerdiği görülmektedir. Oluşturulan projenin çok katmanlı yapıyı desteklenmesi amaçlanmıştır. Çok katmanlı yapı şu bileşenlerden oluşacaktır,

- **Veritabanı katmanı,** verilerin saklandığı birimdir. Veritabanı, Oracle, MySQL, MSSql, Access gibi ilişkisel veritabanı yönetim sistemleri olacağı gibi, düz metin, XML dosyaları şeklinde de olabilir.
- **Veri erişim katmanı,** ilgili nesnelerin verilerle olan iletişimini sağlamaktır. Kayıt ekleme, çıkarma, değiştirme ve kayıtları listeleme gibi metotları sağlar.

Veri erişim katmanı her veritabanı türü için farklı şekilde oluşturulur. İş katmanı açısından yapılan işte bir değişiklik görülmez.

- **İş katmanı**, kullanıcı arabiriminde kullanılacak verileri sağlar. İş katmanı sabittir, veritabanı veya arabirime göre değişmez. Arabirim oluşturulurken kullanılacak nesne metotları, özellikleri, iş katmanı tarafından sağlanır. İş katmanının en önemli özelliği, arabirimde kullanılacak işlemlerin her seferinde tekrar oluşturulmasını engellemektir. Bir kere oluşturulan nesne birçok yerde defalarca kullanılabilir.
- **Arabirim**, kullanıcıların muhatap olduğu görsel araçtır. Nesnelere ilgileri bilgileri görüntüler, iş katmanının sunduğu metotlarla kullanıcıların işlemler yapabilmelerini sağlar.

Çok katmanlı yazılımın temel amacı, kuruluşun farklı tür veritabanı kullanmayı talep ettiğinde, bütün kodların baştan aşağı değişmesini engellemektir. Başka bir neden ise programı satın almak isteyen kişi ve kuruluşların iş katmanındaki kodları görmeden farklı veritabanlarına uyumlu hale getirebilmelerini sağlamaktır. Çok katmanlı yazılım, çok küçük boyutlu projeler için gerekli olmamakla birlikte, devasa projeler için katman türü ve sayısı artırılabilir. Projede veri erişim katmanı DAL (Data Access Layer), iş katmanı ise BLL (Business Logic Layer) olarak belirlenmiştir.

Uygulamada SqlExpress veritabanı kullanılacaktır. SQL Server Management Studio Express programıyla bir veritabanı oluşturulur. Oluşturulan veritabanı bağlantısı Web.config dosyasına eklenir.

### **4.3.2 Üyelik, Yetkilendirme ve Profil Yapılandırması**

#### **4.3.2.1 ASP.NET 2.0 Üyelik ve Profil Desteği**

ASP.NET 2.0 ile birlikte, daha önceki versiyonlarda bulunmayan üyelik ve profil ayarlarının yapılabileceği çeşitli sınıflar çıkmıştır. Hemen hemen bütün projelere rahatlıkla uyarlanabilecek bu sınıflar, kullanıcıların yönetimleriyle ilgili

yazılması gereken kodların çoğundan tasarruf etmeyi sağlar. Üyelikle ilgili sınıflar; Membership ve MembershipUser sınıflarıdır. Membership sınıfı, genel olarak gereksinim duyulan metotları içermektedir. Bu metotlar, açıklamalarıyla birlikte aşağıda listelenmiştir.

### **Membership Sınıfının Metotları**

**CreateUser:** Yeni bir kullanıcı hesabı oluşturur. Bu metodun farklı parametrelerle kullanılabilen türleri vardır.

**DeleteUser:** Belirlenen kullanıcıyı silmeye yarar. Silme işlemi yapılırken kullanıcıya özgü uygulamadaki diğer özelleştirilmiş ayarların silinmesi veya varsayılan ayarlarına getirilmesine dikkat edilmelidir. Örneğin, kullanıcılara ait profil fotoğrafı eklenebiliyorsa, fotoğrafların kullanıcı silme işlemi esnasında kaldırılması gerekmektedir.

**FindUsersByEmail:** Belirtilen e-posta adresine sahip bütün kullanıcılara ait bir liste döndürür.

**FindUsersByName:** Belirli ada sahip bütün kullanıcıların listesini döndürür. Genelde ismin bir bölümünü girip arama yapmak içindir. Örneğin, kullanıcı adının herhangi bir yerinde “ali” olan bütün kullanıcıların listesi istensin. Döndürülen liste şöyle olabilir: { “galip”, “aliriza”, “salih”, “halis” }

**GeneratePassword:** Şifrelerin saklanması için üç seçenek vardır,

- Düz metin,
- İki taraflı şifreleme,
- Tek taraflı şifreleme.

Sadece tek taraflı şifreleme yapıldığında, şifre geri getirilemez. Bu tekniğin işleyişi şöyledir; kullanıcı adını ve şifresini ilgili yerlere girer. Sistem şifreyi bir

algoritmaya göre tekrar şifreleyip veritabanında saklar. Karşılaştırma işlemi de kullanıcının şifresinin tekrar şifrelenmiş hali ile veritabanında ilgili kayıttın karşılaştırması ile mümkündür. Bu nedenle şifre herhangi bir nedenle kaybedildiğinde yeni şifrenin oluşturulması gerekliliği bu nedendir. Bir başka durumda da, kullanıcılar sistem yöneticisi tarafından kaydedildiğinde şifresinin güvenlik açısından otomatik oluşturuluyor olması gerekebilir. Oluşturulan şifre kimse tarafından bilinmeden, doğrudan hesabı oluşturulan kişinin e-posta adresine iletilir.

Bu metotlardan bazıları, kullanıcı listesi döndürür. Döndürülen liste nesnelerinin her biri MembershipUser sınıfının bir örneğidir. MembershipUser sınıfının özellikleri ve metotları açıklamalarıyla birlikte aşağıda belirtilmiştir.

### **MembershipUser Sınıfının Özellikleri**

- **Comment:** Genellikle yöneticinin ekleyebildiği, kullanıcıyla ilgili açıklama bölümü.
- **CreationDate:** Hesabın oluşturulma tarihi.
- **Email:** Kullanıcının e-posta adresi.
- **IsApproved:** Kullanıcının sisteme giriş yapıp yapamayacağını belirleyen “true/false” türünde bir özellik. Eğer bu özellik “false” yapılırsa kullanıcı sisteme giriş yapamaz.
- **IsLockedOut:** Kullanıcının ayarlarda belirtilen sayı kadar başarısız giriş yapıp yapmadığını belirten özellik. Bu özellik eğer “true” ise tekrar “false” oluncaya kadar sisteme giriş yapamaz. Salt okunur bir özelliktir. “False” ayarlanabilmesi için UnlockUser metodundan yararlanılmalıdır.
- **IsOnline:** Kullanıcının çevrimiçi olup olmadığını belirten özellik.
- **LastActivityDate:** Kullanıcının sisteme son giriş yaptığı tarihi gösterir.
- **LastLockoutDate:** Belli sayıda başarısız giriş yapıldığında kullanıcının hesabı kilitlenir. Bu kilitlenmenin en son gerçekleştiği zamanı gösterir.
- **LastLoginDate:** Son giriş yapılan tarih.
- **LastPasswordChangedDate:** Kullanıcının şifresini son değiştirdiği tarih.

- **PasswordQuestion:** Kullanıcının şifresini unuttuğunda yeni şifre oluşturulması için cevaplaması gereken şifre sorusu.
- **UserName:** Sisteme giriş yapmak için kullanılan kullanıcı adı.

### MembershipUser Sınıfının Metotları

- **ChangePassword:** Kullanıcının şifresini değiştirir.
- **ChangePasswordQuestionAndAnswer:** Şifre sorusunu ve cevabını değiştirir.
- **GetPassword:** Üyelik ayarlarına göre şifreyi geri getirmeyi sağlar. Eğer şifreler tek taraflı şifreleniyorsa bu metot işe yaramaz.
- **ResetPassword:** Kullanıcı için yeni bir şifre üretir.
- **UnlockUser:** Giriş yapmak için belirlenen sayının üzerinde deneme yapmış olan kişi kilitlenir. Bu metot, kilidin açılmasını sağlar.

Yerleşik üyelik sınıfları neredeyse ihtiyaç duyulan bütün metotları ve özellikleri içermektedir. Bununla birlikte yazılımın türüne göre kullanıcılar için ek özellikler eklenmek istenebilir. Örneğin çevrimiçi sağlık hizmeti veren bir sitede, müşterilerin sağlıkla ilgili ek bilgilerinin de tutulması istenebilir. Boy, kilo, yaş, cinsiyet gibi özellikler birçok site için gerekli olmasa da, sağlıkla ilgili çok şey ifade eder. ASP.NET 2.0 versiyonu ile özelleştirilebilen profil bilgileri alt yapısı için de çözüm getirilmiştir. Profil bilgileri için gerekli altyapının oluşturulması çok daha pratiktir. Web.config dosyasında istenilen profil özellikleri ve türleri belirlendiğinde, ASP.NET, otomatik olarak bu özelliklere erişimi sağlamaktadır. Sonraki bölümde daha ayrıntılı şekilde ele alınacaktır.

#### 4.3.2.2 Üyelik ve Profil Ayarları

ASP.NET'in üyelik ve profil özelliklerini kullanabilmek için web.config dosyasında bağlantı ayarları ve varsayılan ayarların yapılması gerekir. Üyelik için gereken ayarlar; veritabanı bağlantısının adı, uygulama adı, şifrenin geri döndürülebilir olup olmaması, şifrenin sıfırlanabilme imkanı, gizli soru gerekip

gerekmediği, hatalı şifrenin ardı ardına girilebileceği en yüksek sayı, izin verilen en kısa şifre uzunluğu gibi ayarlardır.

Üyelik ayarları tamamlandıktan sonra, programda üyelik için oluşturulan nesnelerin kullanılabilmesi imkanı oluşur. Bu nesneler, VWD'de ToolBox bölümünün Login alt bölümünde yer almaktadır (Şekil 13).

### Şekil 13: Üyelikle İlgili Nesnelerin Bulunduğu Menü



Bu menüde bulunan nesnelere her biri, web sayfasına eklenerek aktif hale getirilebilir. Nesnelerin işlevleri kısaca şöyledir,

- **Login**, sayfada giriş paneli oluşturulmasını sağlar,
- **LoginView**, kayıtlı kullanıcılar için farklı, anonim kullanıcılar için farklı içerik gösterilmesini sağlar. Kayıtlı kullanıcılar da sahip olduğu rollere göre içerik görüntüleyebilir,
- **PasswordRecovery**, şifresini herhangi bir nedenle kaybeden kullanıcının yeni şifre edinmesini sağlar,
- **LoginStatus**, sisteme giriş yapmış kullanıcılar için çıkış, anonim kullanıcılar için giriş bağlantısı işlevi görür.
- **LoginName**, giriş yapan kullanıcıların kullanıcı adlarını gösterir, diğerleri için ise anonim kullanıcı olarak gösterir.
- **CreateUserWizard**, yeni kullanıcı hesabı oluşturulmasına yarayan nesnedir.
- **ChangePassword**, şifre değiştirme aracıdır.



Profil oluşturmak için web.config dosyasında veritabanı bağlantısı ayarı, uygulama adı ayarları gibi temel ayarların yapılması gerekir. Profilde özelleştirilmiş verileri saklayabilmek için web.config dosyasında formatında arzu edilen özelliklerin isimleri ve veri tipleri XML formatıyla eklenir. Web.config dosyasında profil bölümünün “properties” etiketinin altında şöyle bir görünüm oluşur,

```
<properties>
  <add name="Adi" type="String" />
  <add name="Soyadi" type="String" />
  <add name="WebSitesi" type="String" />
  <add name="TelefonNo" type="String" />
  <add name="Adres" type="String" />
</properties>
```

Bu özellikler eklendiğinde, profil bilgilerine kullanıcı adı vasıtasıyla erişim yapılarak işlemler gerçekleştirilebilir.

Üyelik ve profil ayarlarının yapılmasını sağlayan bütünleşik bir araç mevcuttur. Website menüsünden ASP.NET Configuration seçeneği tıklandığında tarayıcıda “Web Sitesi Yönetim Aracı” açılır. Bu araç vasıtasıyla yeni üyeler, roller ve izin yetkileri eklenebilir. Web tabanlı kalite sistemi uygulamasında sadece yöneticilere erişim izni verilen işlemleri gerçekleştirmek için oluşturulan sayfaların içine bulunduğu klasör “Admin” klasörü olarak belirlenmiştir. Roller web sitesi yönetim aracında “admin” adında bir rol belirlenir. Daha sonra oluşturulan kullanıcılara bir veya birden çok rol atanır. Web sitesi yönetim aracı, klasörlere göre rollerin erişim izinlerini ayarlayabilme olanağı sağlar. Bu şekilde ilk yönetici hesabı oluşturulup arabirim tasarımına geçilir.

#### **4.3.2.3 Kullanıcı Arabiriminin Tasarımı**

Arabirim tasarımında ilk olarak kullanıcıların eklenmesini sağlayacak bileşenin oluşturulması gerekmektedir. Kullanıcı ekleme işlemi sadece sistem yöneticisi tarafından gerçekleştirileceğinden, ilgili sayfa Admin klasörünün altında oluşturulur. Kullanıcı eklemek için CreateUserWizard bileşeni sayfa içine sürüklenip bırakıldığında varsayılan ayarlar ile kullanılmaya hazır durumdadır.

Kullanıcı oluřturma modülünde kullanıcının oluřturulmasının hemen ardından profilinin de ayarlanması istenir. Bu nedenle, CreateUserWizard kontrolünde hesap oluřturma adımıdan hemen sonra yeni bir adım eklenir. Oluřturulan kullanıcı profilinin bu ařamada, oluřturulduktan hemen sonra düzenlenmesi bu sayede saęlanabilir. Kullanıcılar yönetici tarafından oluřturulacaęından řifrelerin otomatik olarak kiřilere gitmesi de isteęe baęlı olarak ayarlanabilir.

Kullanıcı oluřturulması tamamlandıktan sonra, sayfa otomatik olarak kullanıcının profilinin gösterildięi sayfaya yönlendirilir. Bu sayfa da sadece sistem yöneticisinin kontrolünde olduęu için Admin klasörüne eklenir. İlgili sayfada, profili oluřturma veya yeniden düzenleme, sistem yöneticisi olarak atama, fotoęrafı yükleme, kullanıcıya kuruluř içinde sahip olduęu unvanları atama ve kullanıcıyı sistemden tamamen silme gibi özelliklerin bulunması düşünölmektedir. Kullanıcı profilinde yer alan unvanlarla ilgili birim, dokümantasyon modülüyle bütünleřik olacaęından ilerdeki bölümlerde anlatılacaktır.

Kullanıcı oluřturma ve kullanıcı profili düzenleme sayfalarına ek olarak, kullanıcıların toplu olarak görölebileceęi listenin bulunduęu kullanıcı listesi sayfası da eklenir. Bu sayfada kullanıcılar profillerinde yazan bilgilere göre listelenir. Arzu edilirse ada, soyada, e-posta adresine göre arama, sıralama gibi özellikler de eklenebilir.

### **4.3.3 Dokümantasyon Modülü**

Dokümantasyon modülü, programın tam anlamıyla temelini oluřturan ana bölümdür. Bu modül, özel dokümanlar olarak nitelendirilen kayıtları kapsamamaktadır. Kayıtların kontrolü daha farklı bir yapıda olduęundan, kayıtlar için ayrıca bir modülün oluřturulması gerektięi düşünölmüřtür. Bu projenin kapsamı örnek nitelięinde olacaęı için, dokümanların kontrolüyle sınırlıdır. Benzer çalışmalar dięer modüller için de geręekleřtirilebilir.

#### 4.3.3.1 Kullanım Durumları ve Kullanım Durumu Şemaları

Dokümantasyon modülü için ilk önce kullanım durumları ve sistemi kullanacak aktörler belirlenmelidir. “Kullanıcılar kimlerdir, hangi işlevleri gerçekleştirirler, hangi haklara sahiplerdir?” sorularına yanıtların bulunması gerekir.

UML ile tasarıma başlamadan önce sistemdeki aktörlerin ve kullanım durumlarının belirlenmesi gerekir. Kullanım durumları, sistemde yapılan işleri ifade eder. Aktörler ise, sistemin içinde yer almayan ve kullanım durumlarının meydana gelmesine neden olan kişiler veya yazılımlardır. Kullanım durumları ve aktörlerin belirlenmesi, yazılımcıların ve kullanıcıların oluşturulacak sistemi doğru algılamasını, ortaya çıkabilecek mantıksal hataların çoğunun başlangıçta bertaraf edilmesini sağlar.

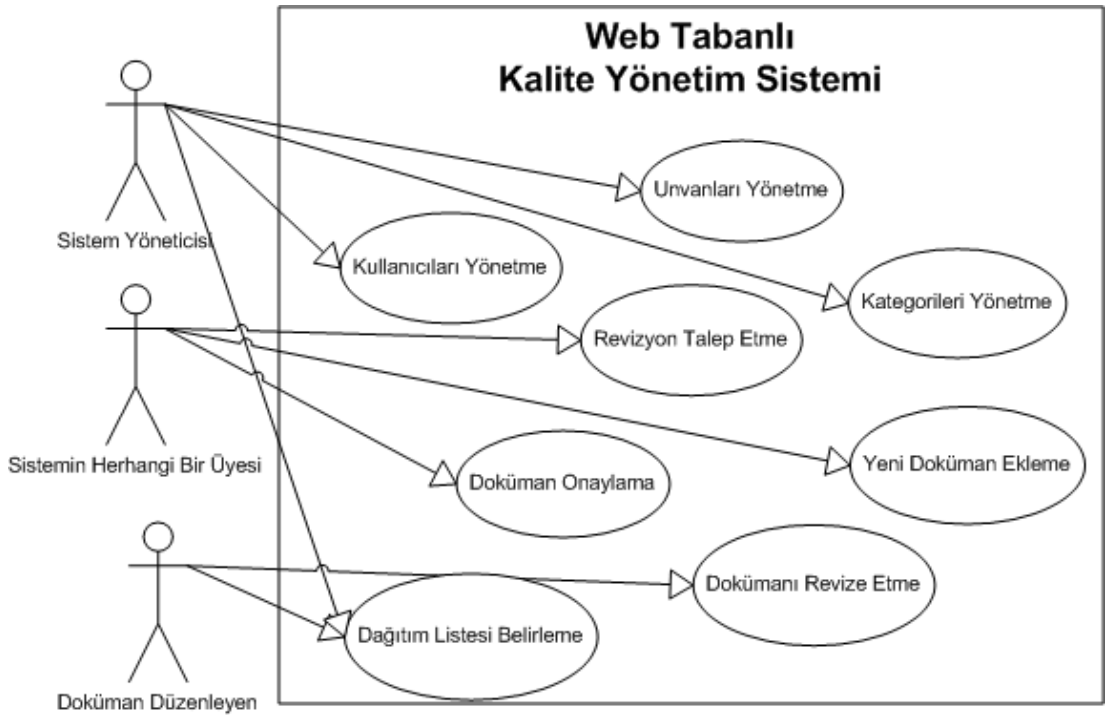
Web tabanlı kalite yönetim sisteminde kullanım durumları, yeni kullanıcı oluşturma, kullanıcı unvanlarını yönetme, kullanıcı silme gibi eylemleri de kapsar. Ancak kullanıcıların yönetimi ile ilgili işlemler ASP.NET'in yerleşik sınıfları kullanılarak tasarlandığı için kullanıcı yönetiminde sadece dokümantasyon modülünde yer alan unvanların yönetimine değinilecektir. Dolayısıyla, kullanım durumu türlerini listeleyecek olursak,

- Unvanlar: Unvan ekleme, çıkarma, güncelleme, kişilere unvan atama gibi kullanım durumları.
- Doküman kategorileri: Kategori ekleme, çıkarma, güncelleme kullanım durumları,
- Yeni doküman ekleme veya varolan dokümanı revize etme kullanım durumları,
- Dağıtım listeleri: İlgili dokümana belirlenen unvanların görme yetkilerini atama ve varolan yetkileri kaldırma kullanım durumları,
- Talepler: Revizyon talebi, yapılan revizyonu onaylama talebi ve yeni dokümanı onaylama talebi kullanım durumları,

- Onaylama: Yeni doküman veya revizyonu onaylama kullanım durumları.

Kullanım durumları, kullanım durumu şemasında gösterilmiştir (Şekil 14). Şekilde, kullanım durumları web tabanlı kalite yönetim sistemi sınırları içerisinde elips şeklinde gösterilmiştir. Aktörler ise sistem sınırları dışında çöp adamlar ile simgelenmiştir. Hangi aktörün hangi kullanım durumu ile ilgili olduğu, aradaki oklarla belirlenir. Unutulmaması gereken şey, “Sistem Yöneticisi”nin aynı zamanda “Sistemin Herhangi Bir Üyesi” de olabileceğidir. Yani bir kişi birden fazla aktör ile simgelenen rollere sahip olabilmektedir. Kullanım durumu şeması, sistemin bütün halinde görülebilmesini sağlar. Bu sayede, sistemin kullanıcılarının beklentileri de iletişim kazasına yol açmayacak derecede açık biçimde belirtilmiş olmaktadır.

**Şekil 14: Web Tabanlı Kalite Yönetim Sisteminin Kullanım Durumu Şeması**



Görüldüğü gibi kullanım durumları, her faaliyet için geçerlidir. Kullanım durumlarının neler olduğunu belirledikten sonra, kullanım durumu kartları hazırlanır. Bu kartlardan en önemli üçü olan “yeni doküman ekleme”, “doküman revizyonu” ve “talep onayları” aşağıda ayrıntılarıyla oluşturulmuştur.

Buna göre, “Yeni Doküman Ekleme” kullanım durumu Şekil 15’te, “Bir Dokümanı Revize Etme” kullanım durumu Şekil 16’da ve “Doküman Onaylama” kullanım durumu ise Şekil 17’de gösterilmiştir.

### Şekil 15: Yeni Doküman Ekleme Kullanım Durumu

Aktör	Sistemdeki herhangi bir “Üye”
Kapsam	Üye, yeni bir doküman eklemek ister
Adımlar	<ol style="list-style-type: none"> <li>1. Üye doküman ekleme sayfasını açar.</li> <li>2. Dokümanın ilgili olduğu kategoriye seçer.</li> <li>3. Dokümanı hangi unvanına istinaden ekliyorsa, listeden onu seçer.</li> <li>4. Takvimden termin zamanını seçer.</li> <li>5. Onaylayacak kişinin unvanını listeden seçer.</li> <li>6. İlgili dokümanı ekleme kısmında belirler.</li> <li>7. Dağıtım listesinden kimlerin göreceğini belirler.</li> <li>8. Gönder’i tıklayarak işlemi tamamlar.</li> </ol>
Alternatif Akış Koşulu	Yeni dokümanın ekleneceği kategori yoktur (Adım 2)
Alternatif Akış	<ol style="list-style-type: none"> <li>2.1 Üye, sistem yöneticisinden yeni kategori eklenmesini talep eder</li> <li>2.2 Sistem yöneticisi yeni kategori ekler ve ana akışa dönlür</li> </ol>
Alternatif Akış Sonrası	Ana akışa dönlür.
Sonuç	<ol style="list-style-type: none"> <li>1. Yeni doküman eklenmiştir.</li> <li>2. İlgili unvana onay talebi iletilmiştir.</li> <li>3. Doküman sunucuya yüklenmiş, yüklemeler tablosuna kayıt girilmiştir.</li> </ol>

Kullanım durumları, aktörler ve kullanım durumu şeması hazırlandıktan sonra, sınıfların oluşturulması, nesne tabanlı programlama yolundaki ilk adımdır. Sınıflar, kullanım durumlarında herhangi bir eksikliğe fırsat vermeyecek şekilde oluşturulmalıdır.

### Şekil 16: Bir Dokümanı Revize Etme Etme Kullanım Durumu

Aktör	Dokümanı düzenleyen kişi olan “Düzenleyici”
Kapsam	Düzenleyici, revizyon talebi olan dokümanı revize etmek ister
Adımlar	<ol style="list-style-type: none"><li>1. Düzenleyici çıkan uyarı neticesinde revizyon talepleri sayfasını açar.</li><li>2. Revizyon talep listesinde ilgili dokümanı seçer.</li><li>3. Dokümanla ilgili bilgiler sayfada listelenir.</li><li>4. Revize etmek için, “Dokümanı revize et” bağlantısını tıklar.</li><li>5. Açılan panelde dokümanı onaylayacak unvanı seçer.</li><li>6. Takvimde onaylama için termin süresini belirler.</li><li>7. Revize edilmiş dokümanı ekler.</li><li>8. Gönderme komutu verilir.</li></ol>
Sonuç	<ol style="list-style-type: none"><li>1. Dokümanın yeni revizyonu eklenmiştir.</li><li>2. İlgili unvana revizyon onayı talebi iletilmiştir.</li><li>3. Doküman sunucuya yüklenmiş, yüklemeler tablosuna kayıt girilmiştir.</li></ol>

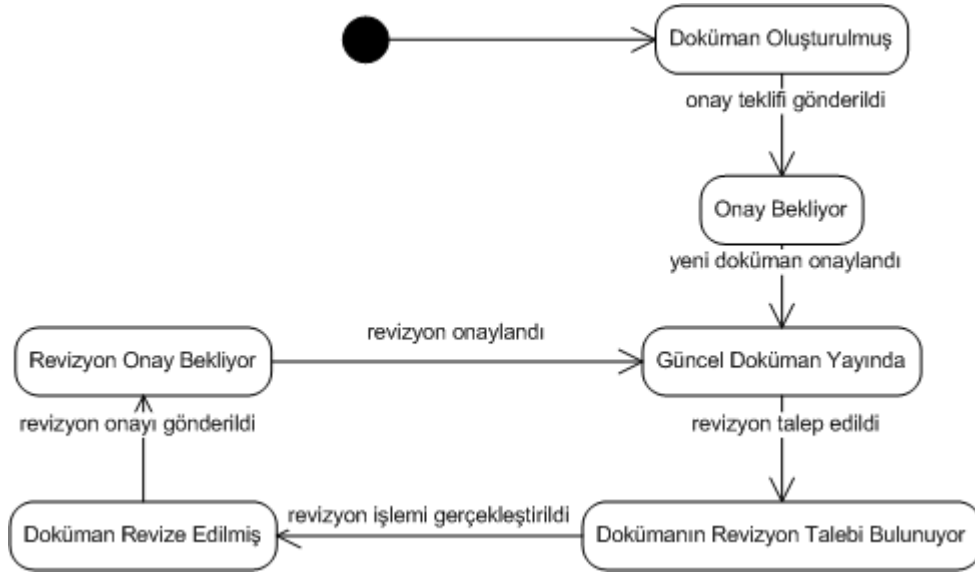
### Şekil 17: Doküman Onaylama Kullanım Durumu

Aktör	Doküman düzenlendiğinde onay talebi giden unvana sahip “Yetkili”
Kapsam	Yetkili, gelen onay talebini inceleyip onaylar
Adımlar	<ol style="list-style-type: none"><li>1. Yetkili, uyarı vasıtasıyla onay taleplerinin bulunduğu sayfayı açar.</li><li>2. Listedeki onaylanması istenen doküman seçilip, onaylamayla ilgili sayfaya erişilir.</li><li>3. Doküman bilgileri doğrulanır, doküman dosyası açılıp incelenir.</li><li>4. Doküman eğer bir revizyonsa karşılaştırma yapılabilmesi için güncel doküman sayfasına bir bağlantı bulunur.</li><li>5. Onaylanması uygun görülürse onaylama gerçekleştirilir.</li><li>6. Uygun görülen yükleme bilgileri, güncel versiyon olarak kaydedilir.</li><li>7. Dokümanın düzenlenme tarihi, onaylayan kişinin adı, unvanı, gibi bilgiler güncellenir, eğer revizyonsa revizyon sayısı bir artırılır.</li></ol>
Alternatif Akış Koşulu	Dokümanın onaylanması uygun görülmez (Adım 5)
Alternatif Akış	<ol style="list-style-type: none"><li>5.1 Onaylanmayan doküman, onay listesinden kaldırılır.</li><li>5.2 Yönetici, nedenini belirterek tekrar düzenlenmesini talep eder.</li><li>5.3 Doküman için revizyon talebi gönderilir.</li></ol>
Alternatif Akış Sonrası	Ana akıştan çıkılır.
Sonuç	<ol style="list-style-type: none"><li>1. İlgili doküman onaylanmış, bilgileri güncellenmiştir.</li><li>2. Dağıtım listesinde bulunanların yeni dokümana erişimi sağlanmıştır.</li></ol>

#### 4.3.3.2 Durum ve Etkinlik Şemaları

Durum şemaları daha önce bahsedildiği gibi, bir nesneyle ilgili bütün durumları gösteren şemalardır. Durum şemaları, kullanım durumlarında yer alan söz konusu nesnelerin bütün durumlarının doğru olarak belirlenip belirlenmediğini görebilmek için kullanılır. Örneğin, herhangi bir kullanım durumunda, bir nesnenin durumları doğru olarak anlatılmayabilir veya ayrıntılar gözden kaçabilir. Durum diyagramı vasıtasıyla bu eksiklik giderilmiş olacaktır. Şekil 18’de “Doküman” nesnesinin durum şeması gösterilmiştir. Durum şeması, sürekli gözden geçirilip eksikler düzeltilmeli, son haline ulaşılmadan tasarım aşamasından kodlamaya geçilmemelidir.

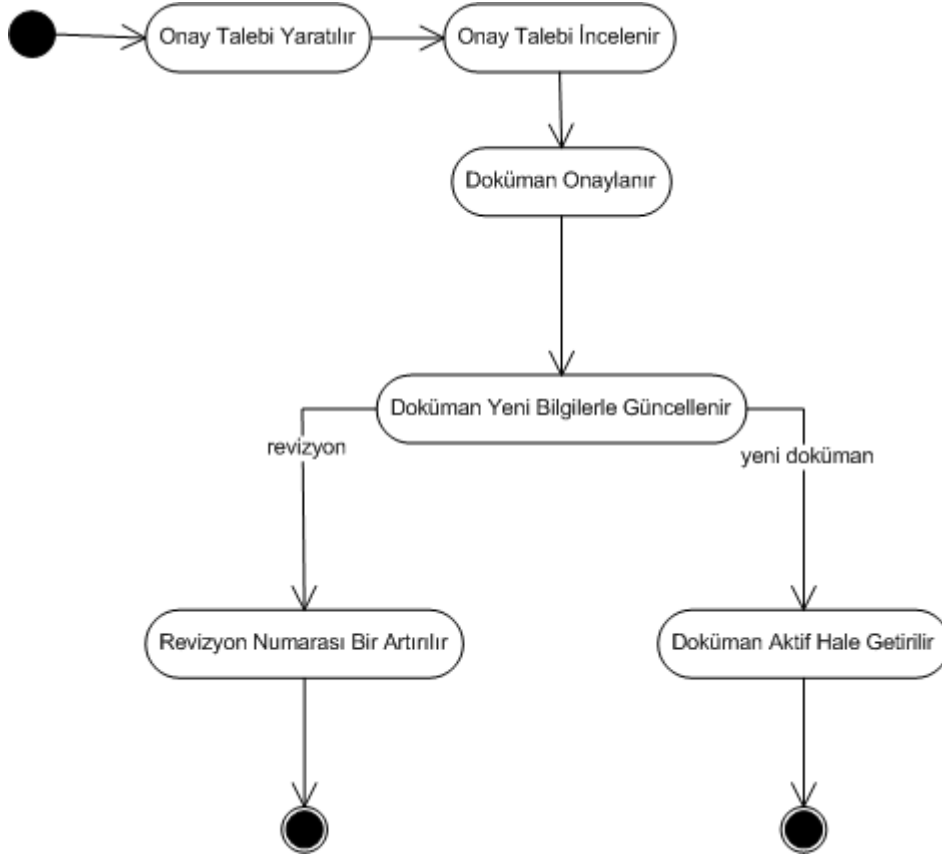
Şekil 18: Doküman Nesnesinin Durum Şeması



Durum şeması, bir nesne ile ilgili bütün adımları içerir. İki durum arasında geçişi sağlayan etkinlikler ise etkinlik şemalarında gösterilir. Etkinlik şeması olarak dokümanın onaylanması etkinliği Şekil 19’da ele alınmıştır. Dokümanın onaylanması, dokümanın durum diyagramında “Güncel Doküman Yayında” durumuna geçmesine neden olmaktadır.

Durum ve etkinlik şemaları, ilgili nesnelerin karmaşık etkinlikler ve durumlara konu olduğu zamanlarda çok yararlıdır. Her nesne için bu şemalar gerekmez. Örneğin sadece iki durumu olan bir nesne için durum ve etkinlik şeması oluşturulmayabilir.

**Şekil 19: Doküman Onaylama Etkinlik Şeması**



#### 4.3.3.3 Sınıf Şemaları

Web tabanlı kalite yönetim sisteminin dokümantasyon modülüne bağlı nesneler, kullanım durumlarına göre belirlenir. Kullanım durumlarında işaret edilen nesneler, programda oluşturulacak sınıflar için çıkış noktasını oluşturur. Bu sınıflar ve açıklamaları şöyledir,

- Unvan: Kullanıcı unvanlarının isimleri ve kimliği yer alır. Yeni bir unvan oluşturulduğunda, unvan adı ve otomatik olarak oluşturulan kimliği bir kayıt



olarak eklenir. Unvanların kullanıcılara atanması için KullaniciUnvan, doküman dağıtım listesinde unvanlara izin verilmesi için DagitimListesi sınıfları kullanılır.

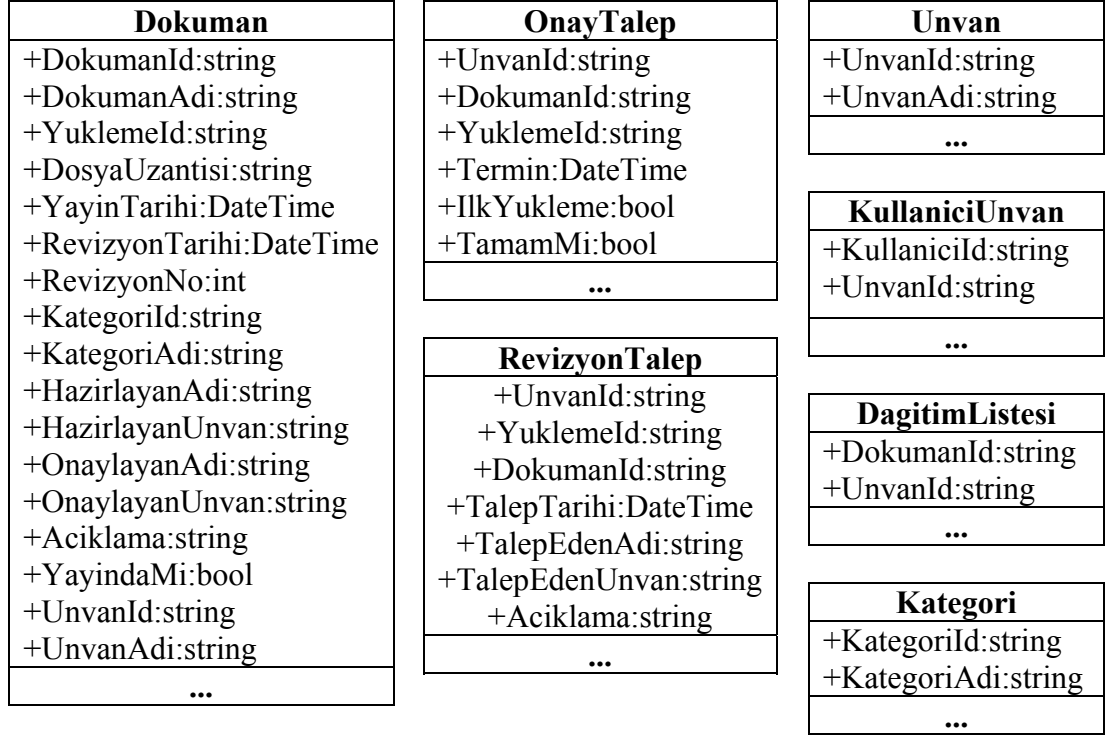
- KullaniciUnvan: Kullanıcılara, sisteme giriş için kullandıkları kullanıcı adlarına göre unvan ataması yapılan sınıftır. Kullanıcının sahip olduğu unvanların listesine ulaşabilmek için, kullanıcı adı ile sorgulama yapılır, unvan kimliklerinin listesi döner.
- DagitimListesi: Dağıtım listesi aslında bir liste sınıfı değildir. Kullanıcı unvanlarına göre görme izni verilen dokümanların her bir bağlantısını ifade eder. Bir dokümanın dağıtım listesinin belirlenmesi, ilgili doküman kimliğinin girilmesi sonucu döndürülen, DagitimListesi sınıfından türemiş nesnelere oluşan listedir. Dokümanın dağıtım listesinde yeni bir unvana izin verilmek istendiğinde, ilgili dokümanın kimliği ve ilgili unvanın kimliğinden oluşan kombinasyon kaydı eklenir.
- Kategori: Dokümanların eklendiği klasörlerden her birini oluşturur. Kategoriler, kategori adı ve kimliğinden oluşur. Dokümanların listeleri, kategorilere göre listelenir. İçeriği bulunan kategori silinemez.
- OnayTalep: Bir doküman onaylanmak için gönderildiğinde, bu sınıftan nesne türetilir. Kişinin onay bekleyen taleplerinin olup olmadığı, sahip olduğu unvanlara göre taranır. Eğer onay bekleyen talep varsa, OnayTalep sınıfından türemiş nesnelere listesi döndürülür. OnayTalep, aynı zamanda termin süresinin de belirlenmesi için bir özellik içerir. Gönderilen onaylar, termin süresine kadar onaylanmadıysa, kullanımdan kalkar. Onaylanıncaya kadar sorgulamalar sırasında görünmeyecektir. Onay talepleri, yeni doküman onayı veya revizyon onayı şeklinde olabilir.
- RevizyonTalep: Bir dokümanla ilgili revizyon talebi bulunduğunda, RevizyonTalep nesnesi oluşturulur. Bekleyen revizyon talepleri, dokümanı düzenleme görevi kime verilmişse o kişi tarafından görülür. Sistem, kişilere değil, görevlere göre çalıştığından görev nakli olduğu durumda göreve yeni atanan kişiye, doküman düzenleme yetkisi başka göreve geçtiğinde ise o kişiye görünür.

- Doküman: Dokümanlarla ilgili bilgilerin saklandığı sınıftır. Sınıftan türetilen bir nesne, sadece bir dokümanı tanımlar. Doküman sınıfından oluşturulan nesnelerin listelenmesi değişik türlerde gerçekleştirilebilir. Revizyonda aynı dokümanın eski sürümleri, güncel dokümanları aramada, ilgili kategorideki dokümanların listesi, yönetici haklarına sahip olanlarda ise dağıtım listesinde olup olmadığına bakılmaksızın tüm dokümanların listesi görüntülenir. Listeler, farklı sorgulara göre oluşturulabilir. Doküman nesnelere tanımlanırken farklı tablolardan karmaşık sorgular gerçekleştirilmesi gerekmektedir. Çünkü, dokümanların tanımları ayrı bir tabloda, yüklenen her dosya ve tanımları ayrı bir tabloda, diğer bilgiler de ayrı tablolarda tutulmaktadır. Eğer bütün bilgiler tek bir tabloda tutulmaya çalışılırsa, ilişkisel veritabanı mantığına aykırı işleyiş ile verilerin tutarlılığının tehlikeye girme ihtimali ortaya çıkar. Sorguların karmaşıklığının yanı sıra, sorgudan döndürülen sonucun türü değişmez, her seferinden Doküman sınıfından türemiş nesnelerin listesi türündedir.

Bu bilgilerin ışığında sınıf şemalarının özellikleri Şekil 20’de görülmektedir. Sınıf şemalarında sadece özelliklere yer verilmiş, metotlar bölümü üç nokta ile belirtilmiştir. Bu da her sınıfın kendi metotları olduğunu, ancak sınıf şemasında gösterilmediğini ifade etmektedir. Metotlar, ilgili nesneye göre veri ekleme, okuma, listeleme gibi temel metotlardır.

Sınıf şemalarında sadece “public” niteliğine sahip (sınıf dışarısından erişilebilir nitelikte) özellikler listelenmiştir. Bununla birlikte, her özellik için veri tutması amacıyla sınıfın “private” (sadece sınıf içerisindeki öğeler tarafından erişilebilen) öğeleri bulunmaktadır. Sınıf diyagramlarının bu şekilde gösterilmesinin amacı, sınıfın sahip olduğu temel özelliklerin karmaşaya fırsat verilmeden sunulmak istenmesidir.

**Şekil 20: Dokümantasyon Modülünün Sınıf Şemaları**



#### 4.3.3.4 Veritabanının Tasarlanması

Gereksinim duyulan sınıflar oluşturulduğunda, ilgili verilerin tutulacağı veritabanlarının tasarlanması gerekmektedir. Veritabanı olarak ilişkisel veritabanı kullanılacaktır. Ancak metin dosyaları veya XML türü dosyalar, veritabanı olarak kullanılmak istenebilir. Bu nedenle sınıfların kimlikleri, veritabanı tarafından otomatik artma ile oluşturulan sayı yerine, GUID (Globally Unique Identifier) denilen program tarafından her seferinde eşsiz olarak üretilen ve 32 karakterden oluşan kimlikler kullanılacaktır. Böylece, veritabanı tutarsızlığı önlenmiş olup, esnek yapılandırmaya da açık olacaktır. Veritabanı olarak MSSQL kullanılacak olup, sorgular için saklı yordamlar (Stored Procedures) kullanılacaktır. Saklı yordamlar ile programla birkaç kerede sorgulanacak veriler, akıllı yapıları sayesinde bir kerede sorgulamaya fırsat tanıyıp zaman ve performans sağlamaktadır. Saklı yordamların bir diğer özelliği, daha güvenli olmasıdır. Sorgular, parametreler vasıtasıyla iletildiği için, kötü niyetli yazılımcıların sisteme doğrudan SQL sorguları göndermesine fırsat

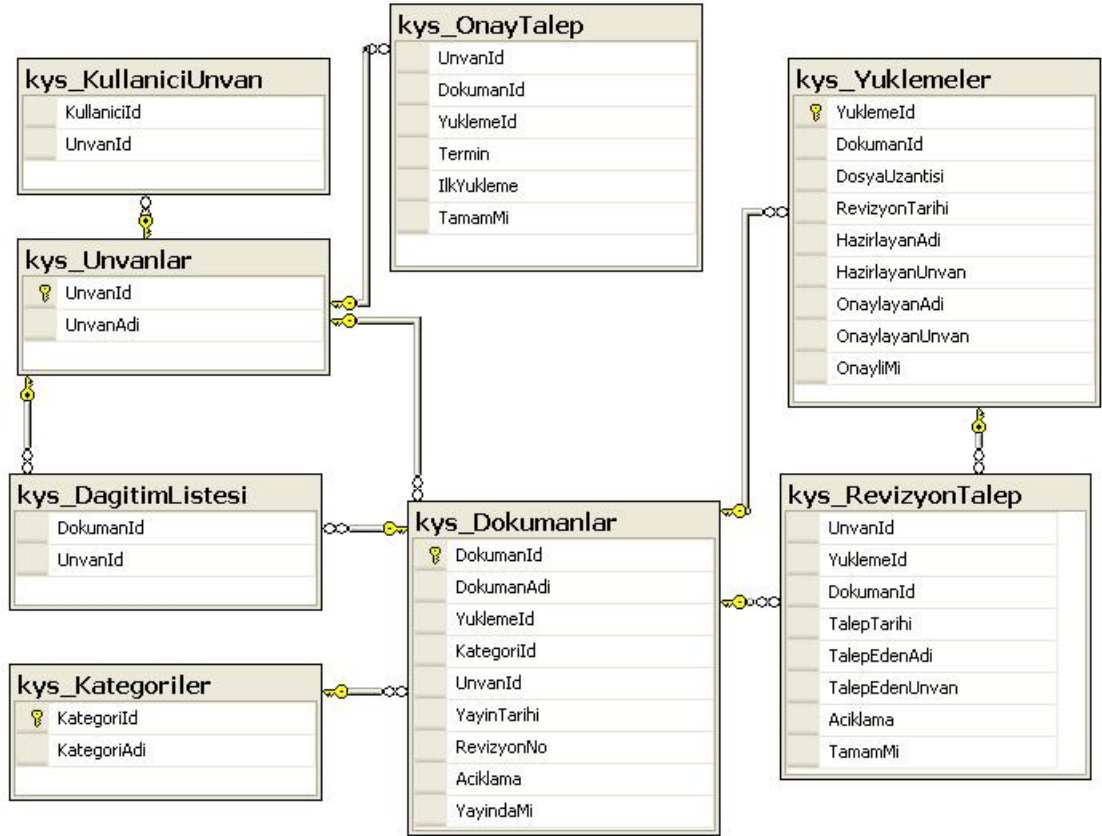
vermemektedir. Gereksinim duyulan veritabanı tablolar ve açıklamaları sırasıyla şöyledir,

1. Unvanlar: Unvanların ve unvan kimliklerinin yer aldığı tablodur.
2. KullaniciUnvan: Kullanıcılara unvan ataması için kullanılan tablolardır.
3. DagitimListesi: Dağıtım listesi kayıtlarının her biri, sadece doküman kimliği ve unvan kimliğini içermektedir.
4. Kategoriler: Kategori isimlerinin ve kimliklerinin bulunduğu tablodur. Bu tabloda yer alan kategori kimliği, dokümanlar tablosunda dokümanın hangi kategoride olduğunu göstermesi için kullanılır.
5. OnayTalep: Düzenlenen her doküman için onay talebi kaydı eklenir. Onay bekleyen taleplerin listesi, unvana göre ve tamamlanıp tamamlanmadığına göre sorgulanır.
6. RevizyonTalep: Revizyonu talep edilen dokümanın kayıtlarını tutan tablodur.
7. Yuklemeler: Yeni oluşturulan ve revize edilen her dokümanda bu tabloda yeni bir kayıt oluşturulur. Her dokümanın düzenleyen ve onaylayıcı değişeceğiinden, bu veriler yüklemeler tablosunda yer alır.
8. Dokumanlar: Dokümanların ilk oluşturulması esnasında kayıt eklenir. Daha sonra her revizyonda, gerekli alanlar güncellenir.

Veritabanı tabloları, ilişkileri de barındırmalıdır. Bu ilişkiler, birbirleri ile bağlantılı bilgilerin değiştirilmesi, silinmesi gibi dalgalık sonucu veri bütünlüğünün bozulmasına neden olabilecek etkenleri ortadan kaldırmayı sağlar. Örneğin, belirli bir kategoride içerik oluşturulduğunda, ilgili kategorinin silinmesi, kategoriye göre listelenen dokümanların görünmesini engelleyecek, telafi edilemeyen problemlere yol açabilecektir. Tablolardaki ilişkilerin belirlenmesi birincil anahtarlara göre yapılır. Birincil anahtar, tabloda eşsiz kayıt girilmesi gereken kolonu tarif eder. Bu tabloda GUID (Evrensel eşsiz tanımlayıcı) ile üretilen kimliklerin saklandığı kolonlar birincil anahtar olarak atanmıştır. Bir ilişki örneği olarak, dokümanlar tablosunda eklenecek kategorinin kimliği yer almaktadır. Bu durumda aradaki ilişkinin çeşitli seçenekleri vardır. Silme ve güncelleme işlemlerinde veritabanının nasıl davranması gerektiği belirlenebilir. Kategori silindiğinde, ilgili dokümanların hepsinin silinmesi

sağlanabilir. Ancak tabii ki böyle bir işlem dokümanlar için arzu edilmez. Bu durumda “hiçbir şey yapma” seçeneği tercih edilir. Kategori silinmeye çalışıldığında, veritabanı aradaki ilişkiye dayanarak bağlantılı kayıtların olup olmadığını kontrol edecek, eğer kategoride doküman kayıtları varsa kategorinin silinmesini engelleyecektir. Anahtar simgesi, ilgili kolonun birincil anahtar, yani eşsiz veri içerdiğini gösterir. Bağlantının diğer ucundaki sonsuz işareti ise diğer tablodaki kaydın, sayısız kez tekrarlanabileceğini temsil eder. Bu bilgilere istinaden veritabanı tablolarının genel görünümü, SQL Server Management Studio Express ile Şekil 21’deki gibi görüntülenir.

**Şekil 21: Dokümantasyon Modülünün Veritabanı Tabloları ve İlişkiler**

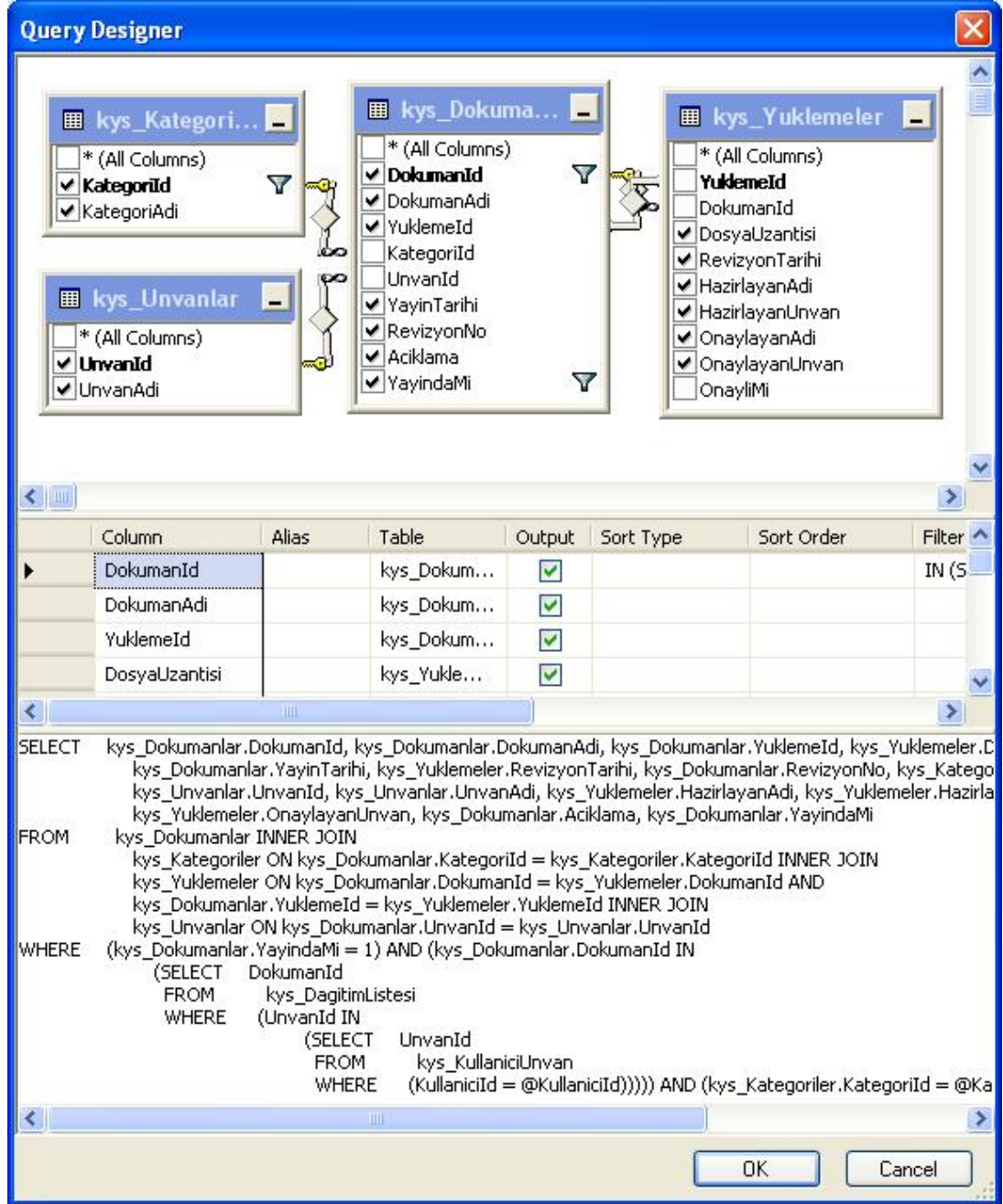


Şekil 21’deki tablolarda her bir satır, tablonun kayıt girilen kolonu niteliindedir. Yani bir kayıt, listedeki bütün kolonlarda veri içermektedir.

Veritabanı tasarımı tamamlandığında, sorguların hazırlanması gerekir. Sorgular daha önce bahsedildiği gibi saklı yordamlar vasıtasıyla gerçekleştirilir. Bunun için SQL Server Management Studio Express’de görsel sorgu tasarım aracı (Şekil 22) kullanılabilir. Normalde kayıt ekleme, değiştirme ve silme sorgularında karmaşık sorgulara ihtiyaç duyulmaz. Ancak, seçme sorguları tahmin edilenden çok daha karmaşık olabilmektedir. Sorgu tasarım aracı, bu zorluğun üstesinden gelmeye yardımcı olmaktadır. Şekil 22’de sistem yöneticisi olmayan bir üyenin, unvanına ve kategoriye göre dokümanları listelemesi için yapılan sorgu, sorgu tasarım aracı ile görüntülenmektedir. Bu sorgu sonucu, doküman sınıfının nesnelere oluşan bir liste döndürülmektedir.

Saklı yordamlar, gerekli bütün faaliyetler için oluşturulur. Bu yordamlar, daha önce bahsedilen çok katmanlı tasarımın veri erişim katmanının bir parçasını teşkil eder. Saklı yordamlara ulaşmak için, Visual Web Developer aracında Solution Explorer’da Şekil 12’deki görüntüsünde bulunan DAL (veri erişim katmanı) klasöründe dokümantasyon modülü için klasör oluşturulur. Saklı prosedürleri kullanması için gereken kodlar bu bölümde yer alır. Farklı türde veritabanı oluşturulmak istenirse, aynı klasörde farklı tür veritabanı (örneğin MySQL) için erişim sınıfları ve metotları yer alır. Her veritabanının erişimi kendine özgü teknikler içermekle birlikte, veri erişim katmanında yapılan değişiklik, iş katmanını etkilemeyecektir.

Şekil 22: Veritabanı Sorgu Tasarım Aracı



#### 4.4 Tasarım ve Programlamanın Tamamlanması

Modellemesi ve tasarımı gerçekleştirilen web uygulamasının ekran görüntüleri ekler bölümünde yer almaktadır. Anlatım sırasında her ekran görüntüsüne referans verilerek açıklanacaktır.

EK 1’de kullanıcı giriş paneli görülmektedir. Bu panelde, geçerli kullanıcı adı ve şifre girilerek sisteme giriş yapılabilir. Anonim kullanıcıların sistemin herhangi bir kısmını görmeleri mümkün değildir. Kullanıcıların giriş yapılabilmesi için önce sisteme tanıtılmaları gerekir. Bunu sadece sistem yöneticisi hakkına sahip kişiler yapabilir. Yeni kullanıcı ekleme paneli de EK 2’de görülmektedir. Yeni kullanıcı ekleme paneli ASP.NET’in yerleşik nesnelere yararlanılarak oluşturulmuştur. Bununla birlikte, yeni eklenen kullanıcının temel bilgilerinin girilebilmesi için bu panelde ikinci bir adım daha oluşturulmuştur (EK 3).

Kullanıcı eklendikten sonra, ilgili kullanıcının profil sayfasına (EK 4 ve EK 5) yönlendirme yapılır. Bu sayfada, kullanıcının bilgileri yeniden düzenlenebilir, fotoğrafı yüklenebilir, sistem yöneticisi olarak atanabilir ve unvanları belirlenebilir. “Sistem yöneticisi mi?” yazısının yanındaki onay kutucuğu onaylanırsa, ilgili kullanıcı sistem yöneticisi olarak atanır ve bütün yönetici haklarına sahip olur.

Kullanıcıya unvanların atanabilmesi için ilk önce unvanların tanımlanması gerekir. Bu, “Unvanların Yönetimi” sayfasından yapılabilir (EK 6). Tanımlanan unvanlar, otomatik olarak profil sayfalarında listelenir ve atama bu şekilde yapılabilir (EK 7). Sistem, birden fazla unvan alabilmeyi desteklemektedir. Yani bir kullanıcı, hem İdari İşler Sorumlusu, hem de Kalite Yönetim Temsilcisi unvanlarına sahip olabilmektedir.

Eklenen tüm kullanıcıların yönetimini gerçekleştirebilmek için, kullanıcıların listelenmesine ihtiyaç vardır. Bu listeye Kullanıcı Yönetimi sayfasından erişilebilir (EK 8). Listede kullanıcıların temel bilgileri yanında, sisteme giriş için kullandıkları kullanıcı adları bulunur. Bunu tıklayarak ilgili kullanıcılar ile ilgili işlemleri yapabilirler.

Dokümanların oluşturulabilmesi için ilk önce doküman kategorilerinin oluşturulmasına ihtiyaç vardır. Kategorilerle ilgili yönetim sayfası EK 9’da görülmektedir. Kategoriler, klasör mantığıyla çalışmaktadır. Her doküman mutlaka



bir kategorinin altında yer almalıdır. Bu sayede sınıflandırma sağlanıp, aranan dokümana daha kısa sürede erişme imkanı oluşmuş olur.

Yeni doküman ekleme paneli, EK 10 ve EK 11’de görülmektedir. Bu panelde, sistemde daha önce yer almayan, yeni bir doküman tanımlanır. Bu sayfada dağıtım listesi dışındaki bütün alanlar zorunludur. Hazırlayan kısmında açılan menüde, sadece o anda dosyayı ekleyen kişinin unvanları listelenmektedir. Bunun amacı, birden fazla unvanı bulunan kullanıcıların, hangi unvanıyla dokümanı ekleyip, revizyon taleplerinin ulaşacağını belirlemesini sağlamaktır. Böylelikle, örneğin unvanlarından birini devreden kullanıcının, devrettiği unvanla ilgili bütün sorumlulukları diğer kullanıcıya devredilmektedir.

Yeni doküman ekleme panelinde, ekleyen kullanıcı, dağıtım listesinden istediği unvanları seçebilir. Böylece doküman, onaylandığı anda seçilen kullanıcılar tarafından görüntülenebilir olacaktır. Dokümanın kullanıcı listesi herhangi bir anda, sistem yöneticisi tarafından tekrar düzenlenebilir.

EK 12’deki ekran görüntüsünde, dokümanların klasörlere göre araştırılabileceği bir sayfa görülmektedir. Bu ekranda kategorinin ilgili satırın sağ tarafından bulunan “İçeriği Listele” bağlantısı tıklandığında, alt tarafta izin verilen bütün dokümanların listesi görülmektedir. Dokümanların listesi, kullanıcının sahip olduğu unvanlara göre ve sistem yöneticisi olup olmadığına göre listelenir. Eğer kullanıcı, sistem yöneticisi haklarına sahipse, istisnasız, varolan bütün dokümanları görüntüleyebilir. EK 12’de ayrıca sol altta bekleyen görevlerle ilgili bir uyarı bulunmaktadır. Bu uyarılar, yeni doküman ve revizyon onay taleplerinde, bir de sorumlusu olunan dokümanlar için revizyon taleplerinde görüntülenir.

Onaylanmayı bekleyen dokümanların listesi EK 13’te, onaylanması beklenen dokümanın sayfasına gidildiğinde oluşan ekran EK 14’te gösterilmiştir. Onay bekleyen doküman sayfasında, onayın revizyon için mi yoksa, yeni doküman için mi olduğu belirtilir. İlgili bağlantı tıklanıp kontrol edildikten sonra onaylanır. Onaylanan dokümanın sayfasında (EK 15) dokümanın bilgileri ve eklenen dosyanın bağlantısı

bulunur. Dokümanların listesi bölümünde, eklendiği kategori seçildiğinde yeni eklenen doküman görünür hale gelir (EK 16).

İlgili dokümanın sayfasında revizyon talep etme bağlantısı bulunmaktadır. Bağlantı tıkladığında revizyon talebiyle ilgili bölüm açılır (EK 17). Bu bölümde, revizyon talebinin nedeni yapıldıktan sonra “Revizyon Talebini Gönder” düğmesi tıkladığında, dokümanı düzenleyen kişinin unvanına revizyon talebi gönderilir.

Dokümanı düzenleme yetkisi olan kişi sisteme girdiğinde, bekleyen görevler kısmında revizyon talepleri uyarısını görür. Uyarıyı tıkladığında revizyon talepleri sayfası açılır (EK 18). Revizyon talepleri listesinde seçilen talebin doküman bilgileri açılan formda görünür. Doküman revize edilmek istendiğinde, Dokümanı revize et bağlantısı tıklanır, revizyon listesinin hemen altında revizyon bağlantısı açılır. Gerekli revizyon, yüklendikten sonra tekrar onaya gönderilir.

Onaylamayla sorumlu kişi sisteme girdiğinde hazırlanmış revizyonun onayıyla ilgili bağlantıyı tıklar. Bu kez onay bekleyen doküman, revizyon olduğu için, ilgili sayfada karşılaştırma yapılabilmesi için dokümanın güncel sürümü bağlantısı da bulunmaktadır. Doküman onaylandığında, hazırlanan revizyon, en son bilgileriyle dokümanlar veritabanında güncellenir. Revizyon numarası da artırılır.

Bir doküman revize edildiğinde, eski sürümlerinin gerektiğinde karşılaştırmalar yapılabilmesi için saklanması gerekmektedir. Doküman listesinde saat simgesi tıkladığında revizyon listesinin görüntülenmiş hali, EK 22’de görülmektedir. Her revizyonun, kim tarafından hazırlanıp onaylandığı, ne zaman yayınlandığı ve dokümanın bağlantısı açılan pencereden erişilebilir.

## SONUÇ VE ÖNERİLER

Son yıllarda teknoloji, hayatın vazgeçilmez bir parçası olmaya başladı. İnsanlar evlerinden çıkarken cep telefonlarını unuttukları zaman, sanki cüzdanlarını unutmuş gibi bir hisse kapılır oldular. Teknolojinin olumlu ve olumsuz sayısız etkileri bulunmaktadır. Olumsuzlar etkileri bir yana, olumlu etkilerinden yararlanmak, teknolojiyi yakından takip etmek, onu akıllıca kullanmak, rekabet avantajı sağlamak ve hatta çoğunlukla bir adım öne geçirmektedir.

İnsanlar kadar, işletmeler de teknolojilerin etkilerinden nasibini almaktadır. Bilgisayar, internet, haberleşme cihazları, yazılımlar ve diğer teknolojik araçlar pek çok işletme için sadece demirbaş niteliği taşımaktadır. Ancak, örneğin bir kurumsal kaynak planlama yazılımı düşünüldüğünde, işletmenin temel iskeletinde yer alan bu yazılım, kuruluşun bütün işlevlerini kontrol edebilir, verileri depolayıp gerektiğinde çıkarabilir, daha da güzeli raporlayabilir ve uyarı sistemleriyle kuruluşu çeşitli risklerden koruyabilir. Dolayısıyla teknolojinin işletmelerin bazıları için etkileri az gibi görünse de, bazıları için hayati öneme sahiptir.

Sektörlerdeki rekabet, her geçen gün iyice kızışmaktadır. Rekabette ayakta kalabilmek için işletmelerin hedefi, buldukları konumu muhafaza etmek değil, sürekli büyümek ve gelişmek olmalıdır. Ancak, teknolojiden yeteri kadar yararlanmayan işletmelerin, bu konuda iddiaları biraz daha zayıf olacaktır.

Günümüzde teknolojiden yararlanmak, sadece kelime işlemci yazılımları, e-posta adresleri, faks, yazıcı ve tarayıcı gibi temel işlevleri karşılayacak amaçlarla sınırlı kalmamalıdır. Sadece bu amaçlar için alınan donanımlar işletmeler için zarar niteliğindedir. Çünkü, akıllıca kullanıldığında, bunlardan başka bir çok işlevi yerine getirebilecek bilgisayarların ve diğer donanımların, sadece temel işlevlerde kullanılması israf niteliğindedir.

ISO-9001 Kalite Yönetim Sistemi standardı, kuruluşların yaptıkları bütün işlere süreç niteliğinde yaklaşmasını gerektiren, müşteri odaklı çalışmanın temel

prensiplerden biri olduđu bir yönetim sistemi standardıdır. Kuruluşlar, genellikle kalite yönetim sistemini, toplam kalite yönetimi için bir başlangıç noktası ve sürekli iyileşmeyi sağlamak için oluştururlar. Kalite Yönetim Sistemine sahip bir kuruluştta, yapılan bütün işlemlerin dokümantasyonu mecbur kılınmaktadır. Bu dokümantasyon sistemi, neredeyse her şeyin açık ve belirgin olmasını sağlamaktadır. Herkesin görevi açıkça belirtilmiş, görevler kişiler üzerine değil, roller üzerine, sürekli iyileştirme ve geliştirme fırsatlarına açık ve tam katılımı sağlayan bir sistemdir.

Kalite yönetim sistemi hakkında en çok rahatsızlık duyulan faktör, dokümantasyon sisteminin getirdiđi yükür. Sistem o kadar ayrıntılı ve belirgin dokümanite edilir ki, dokümanların sayısı ve kontrolü de o kadar zor olur. Bu nedenle, genellikle dokümanların düzenlenmesi, onaylanması, dağıtılması, revize edilmesi, toplanıp tekrar dağıtılması gibi zorunlu işlevler, bilhassa kalite yönetim temsilcilerinin üzerinde ek yük getirmekte, dolayısıyla kalite yönetim temsilcileri yaptıkları bu işlemler nedeniyle çok zaman kaybetmektedir.

Kalite yönetim sisteminin dokümantasyon sisteminin elle yapılmasının ortaya çıkardığı ek yük, bilişim sistemlerinin yardımıyla bir seviyeye kadar indirilebilir. Kalite yönetim sistemi ile ilgili yazılım kullanıldığında, zorunlu işlemler bir kere tanımlanacak, kişiler, roller ve yetkiler ilk başta belirlenecektir. Bu sayede aynı işlemlerin tekrarlanmasına gerek kalmayacaktır. Yapılan işlerin ve yetkilerin sabit olması, böyle bir yazılım kullanma imkanını ortaya çıkarmaktadır. Örneđin, dokümanı oluşturan kişi otomatik olarak tanımlanacak, revizyon talepleri doğrudan bu kullanıcıya iletilecek ve revizyon gerçekleştiğinde, doğrudan onaylama işlemi, yetkili kişinin ekranında belirecek ve onaylanan dokümanın dağıtımı, dağıtım listesinde daha önce belirlenen kişilere otomatik olarak iletilebilecektir. Bütün bu işlemler, kalite yönetim temsilcisinin etkisi olmadan da yürüyebilecektir.

Bir işletme, büyüklüğüne ve bulunduđu sektöre göre farklı çalışan sayısına ve farklı mekan yerleşimlerine sahip olabilir. Bir lojistik firmasının çalışanları, birbirlerinden çok uzak mesafelerde bulunuyorken, başka bir şirketin çalışanlarının hepsi aynı mekanda bulunuyor olabilir. Ancak, gerek müşteriler, gerekse tedarikçi ve

ortaklar, bir şekilde uzak mesafedeki kişilerle ilgili işlem yapılması gerekebilir. Bu nedenle, kullanılacak yazılımın yerleşime bağımsız olmasında yarar vardır. Bu da ancak web tabanlı yazılımlar sayesinde mümkün olan bir işlemdir. Web tabanlı yazılımlar, mekan bağımsızlığına imkan verdiği gibi, platform bağımsızlığı sayesinde program yükleme zorunluluğunu da ortadan kaldırabilir. Platform bağımsızlığından kasıt, Windows, Linux ya da MacOS işletim sistemi olduğuna bakılmaksızın, web tarayıcısı olan bütün bilgisayarlarda, yazılımın çalışır hale gelmesi demektir. Hatta gerekli imkanlar sağlandığında, GPRS desteği olan cep telefonları ya da cep bilgisayarlardan da erişim mümkün olabilmektedir.

Web tabanlı yazılımı elde etmenin birkaç farklı yöntemi vardır. Bunlardan biri, kuruluşun varolan bir paket programının satın alması, diğeri, sistemin kuruluş tarafından tasarlanması ve programlanması, bir diğeri, sistemin kuruluşun kendisinde tasarlanması, programının başka kişi ve kuruluşlar tarafından yapılması gibi. Bu uygulamada, uygulamanın tasarım ve programlanması aşamaları ele alınmış, tasarımı ve/veya programlamanın kuruluş tarafından yapılacağı varsayımı üzerine durulmuştur. Programın kuruluşun kendisi tarafından oluşturulmasının bazı avantajları şöyledir,

- Paket programların maliyetleri, programların, en genel çapta işlerliğini sağlamak için, mümkün olan bütün özelliklerin eklenmesi nedeniyle ve program üreticilerinin kar etme amaçları nedeniyle kuruluşun üreteceği programlardan daha yüksek olabilmektedir. Kuruluş, bu tür maliyetlerden kurtulabilir.
- Yazılımla ilgili temel ihtiyaçlar belirlendiğinde, kuruluşa özel her yeni modül gereksiniminde, paket program üreticilerine ek ücretler ödenmek zorunda kalınmaz.
- Kuruluşun kendi bünyesinde çalışanlar, yazılımlar konusunda kendilerini geliştirme fırsatı bulabilir, deneyim kazanırlar.

Web tabanlı yazılımlar, çeşitli dillerde, farklı veritabanı türleri kullanılarak, seçeneğe bağlı mekanlarda barındırılabilirler. Barındırma ortamı, kuruluşun kendi

sunucusu olabileceği gibi, barındırma hizmeti veren şirketlerin sunucuları da olabilir. Sunucunun şirkette olması, güvenlik açısından faydalı, ancak maliyetler açısından dezavantajlıdır. Barındırma hizmeti veren kuruluşlardan yararlanılması ise, güvenliğin yeteri kadar sağlanamaması durumunda, sunucu taraflı bilgilerin görüntülenmesi ve istenmeyen bilgilerin başkaları tarafından elde edilmesi olasılığını içermektedir. İki seçenek arasında, yazılımın kapsamı, niteliği, kullanım amacı gibi birçok özellik göz önünde bulundurulmalı ve iyi bir analiz sonucu karar verilmelidir.

Web tabanlı uygulamaların veritabanı seçimi de çok önemlidir. Veritabanı olarak metin dosyaları, XML dosyalar, MS Access veritabanı ve daha kapsamlı veritabanları olan MSSQL, MySQL, Oracle da kullanılabilir. Veritabanı seçiminde yazılımın kapsamı çok önemlidir. Örneğin on veya yirmi kullanıcısı olan bir yazılım için metin dosyaları bile yeterli olabilirken, binlerce kullanıcısı olan bir yazılım için, profesyonel veritabanı şirketlerinin ürünleri tercih edilmelidir.

Web tabanlı yazılım geliştirme için kullanılan diller çeşitlilik gösterir. Bu dillerden en popüler olanları yakın zamana kadar, ASP ve PHP idi. ASP, Windows sunucular için geliştirilen, Active Server Pages kelimelerinin baş harflerinden oluşan web programlama dilidir. PHP ise aynı amaçla oluşturulmuş, ancak Linux sunucular için geliştirilen programlama dilidir. Ancak Windows'un “.Net Framework” teknolojisinin ardından ortaya çıkan ASP.NET, diğerlerinden çok büyük farklılıklar göstermektedir.

Windows tarafından geliştirilen .Net Framework, ortak dil çalışma zamanı sayesinde, programlama diline bağımsız olarak çalışmayı, sınıf kütüphanesi sayesinde de hızlı ve güvenilir yazılımlar geliştirmeyi mümkün kılmaktadır. Ortak dil çalışma zamanı özelliği ile, bir proje grubunda, her yazılımcı için hangi programlama dilini kullandığına bağlı olmaksızın ortak yazılım geliştirmesi olanaklıdır. Örneğin, C#'ta uzman bir yazılımcı, uygulamanın iş katmanını programlarken, Visual Basic'de uzman başka bir yazılımcı veritabanı erişim katmanını programlayabilir. Katmanlar nesne tabanlı olacağı için, farklı kodlarla yazılmış sınıflara erişim de

sıkıntı yaratmayacaktır. Sınıfın nesnelere, metotları ve deęişkenleri hangi programlama diliyle yazılırsa yazılsın aynı şekilde kullanılacaktır.

.Net Framework'ün sınıf kütüphanesi, yazılımda ihtiyaç duyulabilecek temel öğelerin çoğunu üzerinde barındırmaktadır. Örneğin bir veri tablosu yapılacağı zaman, PHP ve ASP'deki gibi tablonun en baştan tasarlanması gerekmez. Listelenecek verilerle ilgili bir nesne oluşturduktan sonra, örneğin "GridView" adındaki verilerin tabloda gösterilmesini sağlayan nesne eklendiğinde ve veri kaynağı olarak, oluşturulan liste nesnesi işaret edildiğinde, program, otomatik olarak verileri listeleyebilir. Bu tür nesnelere en önemli özellięi, sınıf kütüphanesinin sahip olduğu hazır sınıfların özelleştirilebilmesidir. Özelleştirilen sınıflar, nesne tabanlı programlamanın, kalıtım özellięi kullanılarak, yani miras alınmış yeni sınıflar oluşturularak özelleştirilebilir. Bu sayede, modüler yapı desteklenmiş olur ve yapılabilecekleri sadece hayal gücüyle sınırlanmaz. .Net Framework sınıf kütüphanesi vasıtasıyla tasarım yapılırken, bilinmesi gereken en önemli şey, nesne tabanlı programlama mantığıdır.

Nesne tabanlı programlama, yazılımın bir sistem olarak ele alındığı, kodlanması istenen varlıkların, nesnelere olarak sistemin içerisinde tanımlandığı bir yapıdır. Çok geniş kapsamlı yazılımlarda, sistem yapısı, sistem analizini olanaklı kılmakta, tasarımı hızlandırıp, ortaya çıkan sorunların en kısa sürede ortadan kaldırılmasına olanak sağlamaktadır. Ayrıca, nesne tabanlı programlama, modüler yapıya destek verdiği için, yeni yazılımların entegrasyonu ve yönetimi, oldukça etkin bir şekilde gerçekleşmektedir.

Bir programın iyice anlaşılmadan kodlanmaya başlanması, sürekli düzeltmeler yapılmasını gerektireceęi ve belki de en baştan tasarlanması gerekeceęi için çok risklidir. Programlama başlamadan önce bilgiler mutlaka iyi analiz edilmeli, yazılımı talep eden kişi ya da kuruluş ve yazılımcılar tam anlamıyla etkin bir iletişime sahip olmalıdır. Yazılımcılar arasında da ortak bir anlayışın var olması gerekir. Yazılımı birden fazla kişi oluşturacaksa, tasarım hakkında görüş birliğini

gösteren simgelere ihtiyaçları olacaktır. Bu gibi nedenlerle ihtiyaç duyulan modelleme dili için bir standart olarak birleşik modelleme dili-UML geliştirilmiştir.

UML, yazılımcıların kendileri arasında ve programı talep edenler arasında ortak bir modelleme dili niteliğindedir. Bu dil ile, aslında soyut olan programın, somutlaştırılmasını sağlar. Modelleme dilinde kullanılan şekiller standart olduğu için, herkes tarafından aynı şekilde anlaşılmakta, kavram kargaşasına yer vermemektedir. UML’de çeşitli bileşenler vardır (Sınıf şemaları, kullanım durumu şemaları, durum ve etkinlik şemaları gibi). Bu bileşenlerden bazıları veya çok kapsamlı yazılımlar için hepsinden yararlanarak, programın son haline kadar yol haritası çizilebilir. Böylece, sonradan çıkabilecek sorunlar, yazılımın kodlama aşamasında meydana gelebilecek talepler en aza indirilebilir. Program yazmak, zaman alan ve yazılım şirketleri için de maliyetli bir işlemdir. Süre ne kadar uzarsa, yazılımcılara verilecek ücretler de o kadar artmaktadır. Başlangıçta iyi tasarlanmamış bir yazılımın sürekli düzeltilmesi, iyi tasarlanmış bir yazılımın başlangıçtaki modellemede harcadığı zamandan kat kat fazla olabilmektedir. Sonuç olarak, UML kullanma zorunluluğu olmamakla birlikte, kullanmanın getirileri göz ardı edilemez.

Uygulama kısmında kalite yönetim sisteminin dokümantasyon yükünün ortaya çıkardığı olumsuzlukları ortadan kaldırmaya yönelik olarak web tabanlı kalite yönetim sistemi örneği oluşturulmaya çalışılmıştır. Bunun için ilk önce uygulamanın amacı açıklanmış, daha sonra analiz ve tasarıma geçilmiştir. Tasarımla ilgili ana yapı oluşturulmuş, dizin içeriği ve çok katmanlı programlama mantığı açıklanmış, temel ayarlar yapılmıştır.

Üyelik, yetkilendirme ve profil ayarları için ASP.NET’in 2.0 versiyonu ile birlikte oluşturulan sınıfları anlatılmış ve nasıl kullanılacağı açıklanmıştır. Bu sınıflar, ihtiyaç duyulan neredeyse bütün metotları içermektedir. Ancak istendiği takdirde, nesne tabanlı programlamanın kalıtım özelliği sayesinde, yeni sınıflar türetilip özelleştirilebilir. Örneğin, kullanıcılarla ilgili bilgiler varsayılan ayar olan MSSQL yerine, Access veritabanında tutulabilecek şekilde yeniden düzenlenebilir. Yazılımcının ilk başta oluşturmak için zaman kaybedeceği, üstelik en temel özellik



olan üyelik sınıflarında yapılan bir hatanın telafi edilemeyebileceği düşünüldüğünde, Microsoft'un oluşturduğu sınıfları kullanmak en doğru seçim olabilir. Bütünleşik üyelik sınıflarını kullanmanın bazı avantajları şunlardır,

- Web.config dosyasından belirtilen ayarları değiştirerek, veri kaynağı, zaman aşımı süresi gibi özellikler kolayca ayarlanabilmektedir.
- Sayfalara ve dizinlere göre yetkilendirme ayarları bütünleşiktir. Örneğin, bir dizin içindeki web.config dosyasında “yonetici” rolüne sahip kullanıcılara izin verip diğerlerini engelleme seçeneği ayarlandığında, ayarlar hemen aktif olacaktır.
- İlk kullanıcı ayarlarının yapılabilmesi için, site yapılandırma aracı özelliği bulunmaktadır.
- Şifrelerin üç farklı türde, yani, açık, tek ve çift taraflı şifreleme özelliklerinde saklanabilme seçenekleri bulunmaktadır. Bu sayede örneğin, güvenliğin çok sıkı olması gerektiği bir uygulamada, tek taraflı, yani geri döndürülemeyen şifreleme özelliği kullanılabilir. Bu özelliğe, kullanıcı şifresini girdiğinde, şifre belirli bir algoritmaya göre tekrar şifrelenir ve veritabanındaki kopyasıyla karşılaştırılarak doğruluğu onaylanır.
- Bütünleşik üyelik sınıfı, bütünleşik profil sınıfının kullanılabilmesi için temel teşkil eder. Profil sınıfında, bir profil özelliğinin eklenmesi için web.config dosyasına bir satır bilgi eklemek yeterli olmaktadır.

Üyelik ve profil sınıfları için gereken ayarlar yapıldıktan sonra, dokümantasyon modülüyle ilgili gereksinimler belirlenmiş ve UML modelleme araçlarından yararlanarak, model şemaları oluşturulmuştur. Model şemaları ile bağlantılı olarak, sınıflar ve veritabanı tasarlanmış, gerekli veritabanı ilişkileri kurulmuştur.

Modelleme ve tasarım bittikten sonra, programlama aşamasına geçilmiş, ortaya çıkan ekran ekte listelenmiştir. Yapılan tasarım, örnek niteliğinde olup bazı eksik yönlere sahiptir. Örneğin, tasarımda yedekleme ve bakım ile ilgili bilgilere yer verilmemiştir. Ayrıca, dokümanın onaylanması aşamasında, onaylanmamasıyla ilgili

bilgi bulunmadığından dolayı, bu aşama, kuruluşlar arasında farklılık gösterebileceğinden, bilinçli olarak boş bırakılmıştır. Kimi kuruluşlar, dokümanın onaylanmadığında tamamen ortadan kalkmasını isteyebileceği gibi, kimi kuruluşlar, tekrar gözden geçirme, başkasına yönlendirme gibi seçeneklerin bulunmasını arzu edebilirler. Bu eylem, kuruluşun ihtiyaçlarına göre, doğru analizler sonucu tasarlanmalıdır. Yapılan uygulama, genel geçer yapıda olmayıp, öneri niteliğindedir. Uygulamanın amaçlarını maddeler halinde sıralayacak olursak,

- Kuruluşların, kalite yönetim sistemleri için yazılım kullanmasının gerekliliğini vurgulamak,
- Nesne tabanlı programlama ve birleşik modelleme dili UML hakkında bilgi vermek. Bu sayede programlama ile ilgili hiç bilgisi olmayanların bile rahatlıkla kavrayabilecekleri ortak noktayı göstermek,
- .Net Framework teknolojilerinin yazılım dünyasına katkılarına değinmek ve yazılım oluşturmanın her geçen zaman daha da kolaylaştığını belirtmek,
- Kuruluşların, mümkün olduğunda kendi bünyelerinde çalışan kişileri bu konuda teşvik etmelerini, uygun görüldüğünde çeşitli eğitimler almalarını sağlamak. Bu sayede, imkan bulunduğu, kar amaçlı girişimlerde bulunmaları imkanını göstermek.

Son olarak, bu çalışmayla ilgili öneriler aşağıdaki gibi listelenebilir,

- Üniversitelerin kendi bünyelerinde bu konuda yapılan akademik çalışmaların artırılması. Akademisyen ve öğrencilerin bulunduğu çeşitli proje gruplarının oluşturulup ortak çalışmalarının sağlanması, dolayısıyla bilgi paylaşımlarının gerçekleşmesi,
- Bölümler arası işbirliklerinin artırılması. Örneğin, Bilgisayar Mühendisliği ile İşletme bölümünün ortak çalışabilmesi için imkanların oluşturulması,
- Sosyal bilimlerde, bilişimle ilgili derslerin son yeniliklere uyumlu hale getirilip güncelleştirilmesi. Gerektiğinde kurs ve seminerlerle desteklenmesi,
- Büyük projelerin TÜBİTAK gibi çeşitli destekleme kuruluşları tarafından destek görmesi için girişimlerde bulunulması.

# KAYNAKLAR

## Basılı Kaynaklar

Baş T., (2004) “**ISO 9000:2000 Kalite Yönetim Sistemi**”, Sistem Yayıncılık 4.Basım, İstanbul.

Bayrak, S. (1999) “**TKY'nin Kamu Kurumlarında Uygulanmasının Pratik Güçlükleri**”, Kamu Yönetiminde Kalite I. Ulusal Kongresi, TODAİE Yayını, Ankara, (49-50)

Benligiray, S. (2005) “**Büro Yönetimi**”, Anadolu Üniversitesi, Eskişehir

Doğan, Ö., Tütüncü, Ö (2003) “**Hizmet İşletmelerinde Toplam Kalite Yönetimi Kapsamında ISO 9001:2000**”, Dokuz Eylül Üniversitesi Rektörlük Matbaası, İzmir

Eriksson, H-E., Penker, M., Lyons, B., Fado, D (2004) “**UML 2 Toolkit**”, Wiley Publishing, Canada

Güngören, B. (2005) “**UML ile Nesne Tabanlı Çözümleme ve Tasarım**”, Seçkin Yayıncılık, Ankara

Halis, M. (2004) “**Toplam Kalite Yönetimi**”, Roma Yayınları, No:13, Ankara

Hoşcan, Y., Şahin, M. (2003), “**Yönetim Bilgi Sistemi**”, Anadolu Üniversitesi, Ankara

Koçer, H. G. (2007), “**XP ve UML'nin Yazılım Geliştirme Sürecine Etkileri**”,Yayınlanmış Yüksek Lisans Tezi. Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Özalp, İ., Şahin, M., Berberoğlu, G., Ceylan, R. (2003) “**Yönetim Organizasyon**”, Anadolu Üniversitesi, Eskişehir

Özveri, O. (2007) “**Kalite Yolculuğu**”, Altın Nokta Yayıncılık, İzmir

Schilt, H. (2006) “**Herkes İçin C# 2.0**”, Alfa Yayınları, İstanbul

Seaver, M., (2002) “**ISO 9000 Focus Moves from Compliance to Achieving Results**”, ISO Management Systems, March-April, (27-31)

Sürmeli, F., Erdoğan, M., Erdoğan, N., Banar, K., Kaya, E, Sevim, A. (2006), “**Muhasebe Bilgi Sistemi**”, Anadolu Üniversitesi, Eskişehir

Şale, İ. (2004) “**ISO 9001:2000 Kalite Yönetim Sistemi ve Uygulamaları**”, Seçkin Yayınevi, Ankara

Taşdelen, A. (2005) “**İleri ve Kurumsal Düzey .Net Uygulamaları**”, Pusula Yayıncılık, İstanbul

TS-EN-ISO 9001:2000 (2001) “**Türk Standardı**”, TSE yayını, Ankara.

Welling, L., Thomson, L. (2005) “**Uzmanlar İçin PHP ve MySQL**”, Alfa Yayınları, İstanbul

## **İnternet Kaynakları**

<http://www.esharpedir.com>

<http://www.gunduz.org>

<http://www.verivizyon.com>

<http://technet2.microsoft.com>

<http://www.msakademik.net>

<http://tr.wikipedia.org>

<http://www.rssnedir.com>

<http://www.bilgiservisim.com>

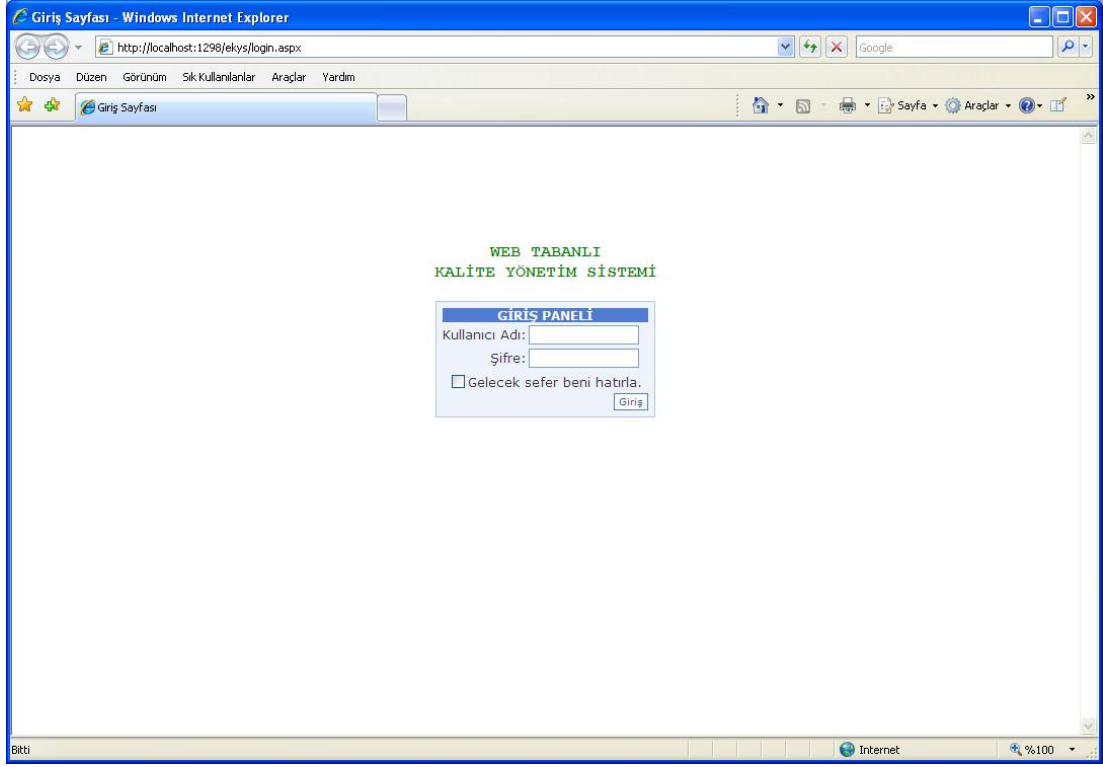
<http://www.danismend.com>

<http://www.polater.com.tr>

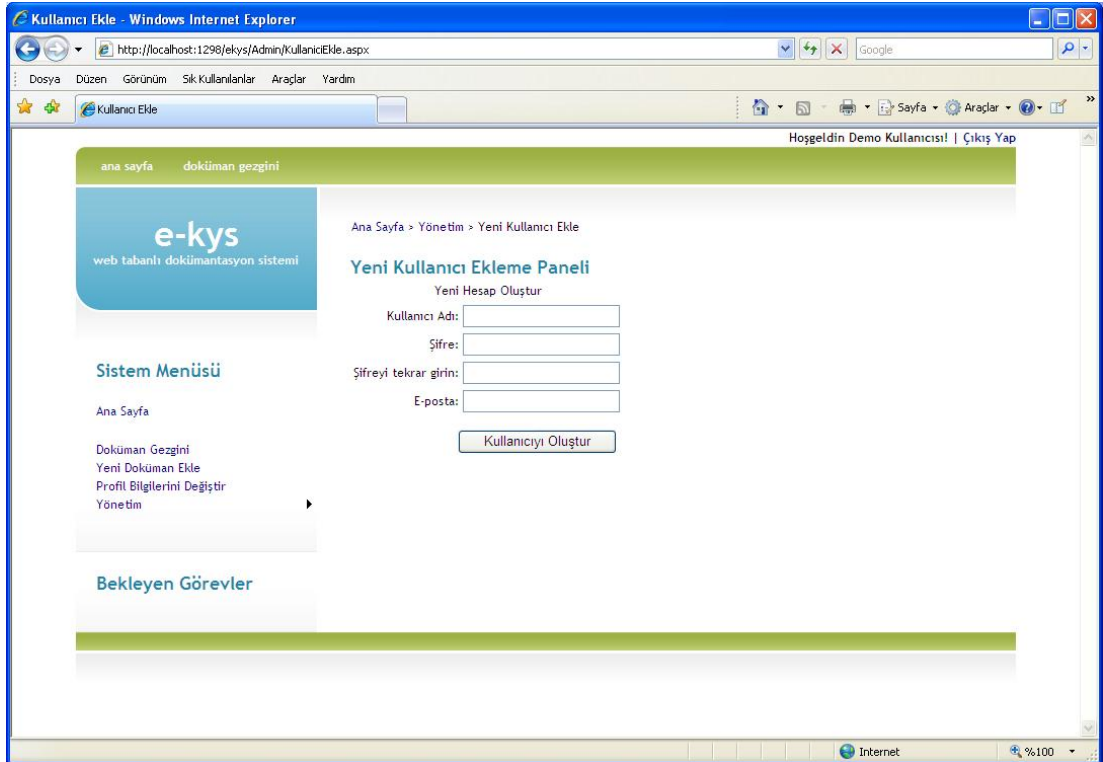
<http://www.kalite.saglik.gov.tr>

# **EKLER**

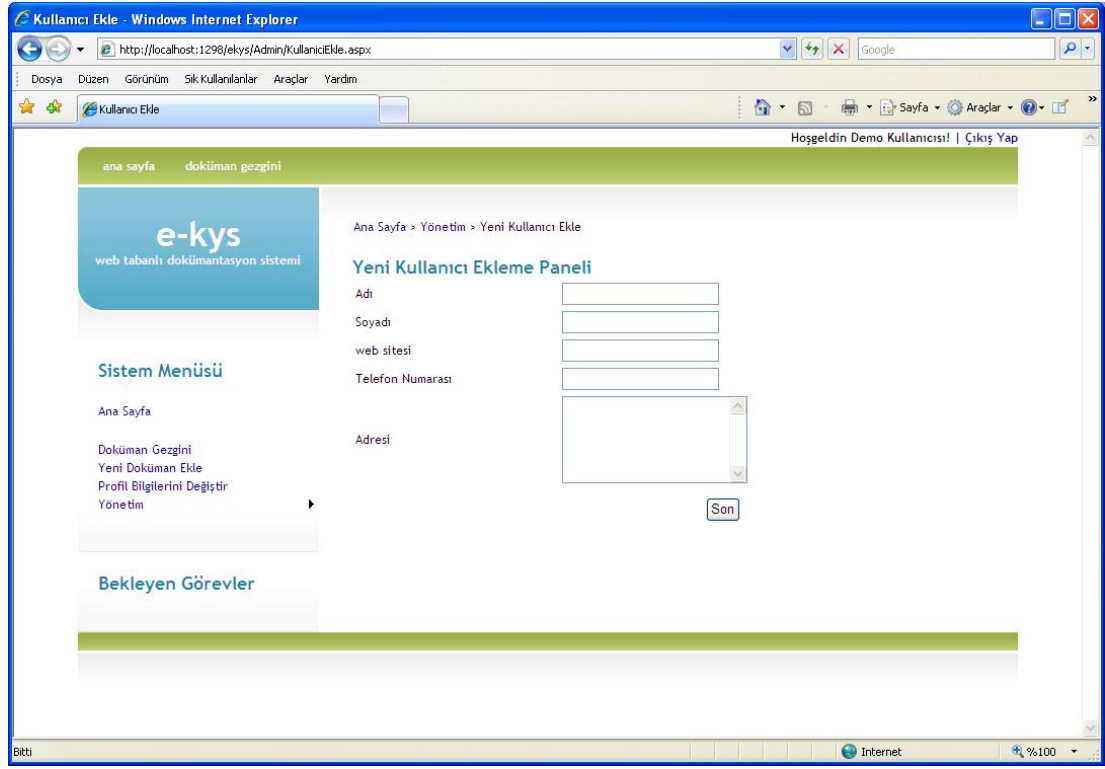
## EK 1: KULLANICI GİRİŞ PANELİ



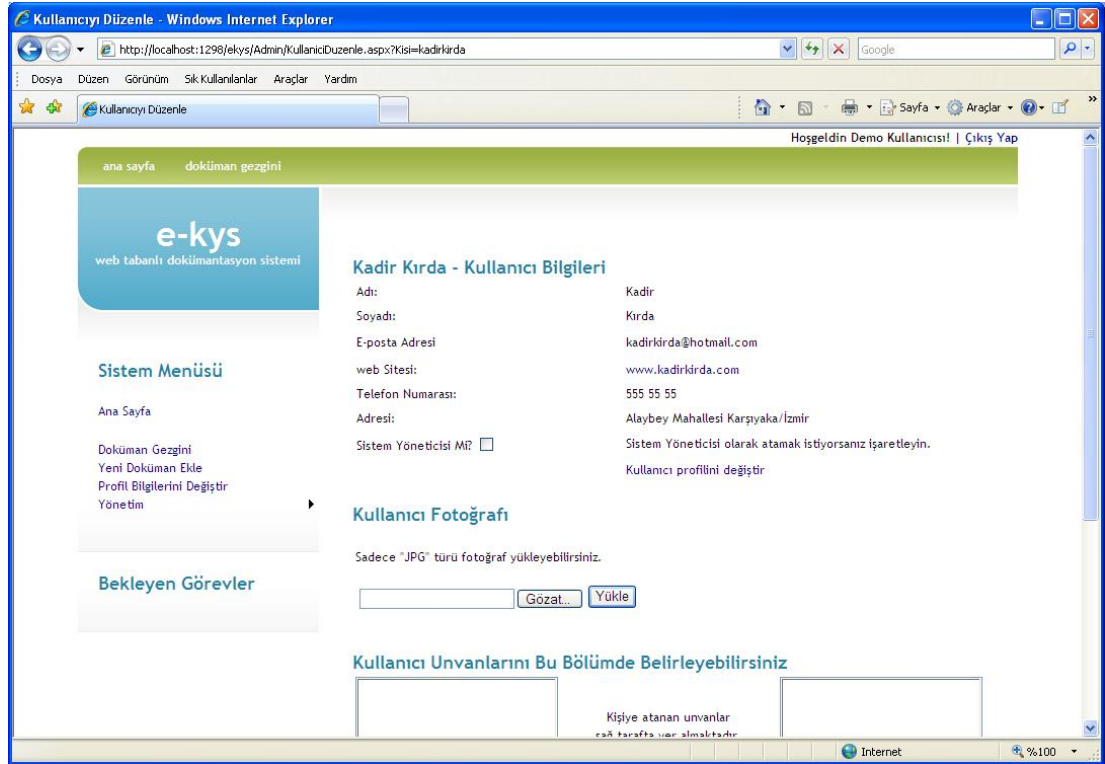
## EK 2: YENİ KULLANICI EKLEME PANELİ



### EK 3: YENİ KULLANICIYA PROFİL EKLEME



### EK 4: OLUŞTURULAN YENİ KULLANICININ PROFİL SAYFASI - 1





## EK 5: OLUŞTURULAN YENİ KULLANICININ PROFİL SAYFASI - 2

Kullanıcı Düzenle - Windows Internet Explorer

http://localhost:1298/ekys/Admin/KullaniciDuzenle.aspx?Kisi=kadirkirca

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Kullanıcı Düzenle

Yönetim

Bekleyen Görevler

**Kullanıcı Fotoğrafi**

Sadece "JPG" türü fotoğraf yükleyebilirsiniz.

Gözet... Yükle

**Kullanıcı Unvanlarını Bu Bölümde Belirleyebilirsiniz**

Kişiyeye atanan unvanlar sağ tarafa yer almaktadır.

Seçili Unvanı Ekle -->>

<<--Seçili Unvanı Kaldır

Yeni unvan eklemek veya varolan unvanları düzenlemek için tıklayın

Kullanıcıyı kalıcı olarak silmek için tıklayın

Internet %100

## EK 6: UNVANLARIN YÖNETİMİ

Unvan Yönetimi - Windows Internet Explorer

http://localhost:1298/ekys/Admin/UnvanYonetimi.aspx

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Unvan Yönetimi

Hosgeldin Demo Kullanıcısı! | Çıkış Yap

ana sayfa doküman gezgini

**e-kys**  
web tabanlı dokümantasyon sistemi

Ana Sayfa > Yönetim > Unvanları Düzenle

**Unvan Yönetimi**

Kalite Yönetim Temsilcisi	Değiştir Sil
Sekreter	Değiştir Sil
Bilgi İşlem Sorumlusu	Değiştir Sil
Genel Müdür	Değiştir Sil
Müdür Yardımcısı	Değiştir Sil
Bina Görevlisi	Değiştir Sil

Yeni Unvan:  Ekle

**Sistem Menüsü**

Ana Sayfa

Doküman Gezgini

Yeni Doküman Ekle

Profil Bilgilerini Değiştir

Yönetim

**Bekleyen Görevler**

Internet %100

## EK 7: KİŞİLERE UNVANLARIN ATANMASI

Kullanıcıyı Düzenle - Windows Internet Explorer

http://localhost:1298/ekys/Admin/KullaniciDuzenle.aspx?Kisi=kadirkirda

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Kullanıcıyı Düzenle

Ana Sayfa Adresi: Alaybey Mahallesi Karşıyaka / İzmir

Doküman Gezgini Sistem Yöneticisi Mi?  Sistem Yöneticisi olarak atamak istiyorsanız işaretleyin.

Yeni Doküman Ekle Kullanıcı profilini değiştir

Profil Bilgilerini Değiştir

Yönetim

**Kullanıcı Fotoğrafi**

Sadece "JPG" türü fotoğraf yükleyebilirsiniz.

Gözet.. Yükle

**Kullanıcı Unvanlarını Bu Bölümde Belirleyebilirsiniz**

Kalite Yönetim Temsilcisi  
Sekreter  
Genel Müdür  
Müdür Yardımcısı  
Bina Görevlisi

Bilgi İşlem Sorumlusu

Kişiyne atanan unvanlar sağ tarafta yer almaktadır.

Seçili Unvanı Ekle -->>

<<--Seçili Unvanı Kaldır

## EK 8: KULLANICILARIN YÖNETİMİ

Kullanıcıları Yönet - Windows Internet Explorer

http://localhost:1298/ekys/Admin/Kullanicilar.aspx

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Kullanıcıları Yönet

Hosgeldin Demo Kullanıcısı! | Çıkış Yap

ana sayfa doküman gezgini

**e-kys**  
web tabanlı dokümantasyon sistemi

Ana Sayfa > Yönetim > Kullanıcıların Yönetimi

**Kullanıcı Yönetimi**

Toplam Kayıtlı Kullanıcılar: 7

Adı	Soyadı	Telefon	Adres	Düzenle
Bilge	İşlemci	111 11 11	Mustafa Kemal Caddesi, Bornova / İzmir	(bilgeislemci)
Kadir	Kırda	555 55 55	Alaybey Mahallesi Karşıyaka / İzmir	(kadirkirda)
Kalime	Temsal	333 33 33	Şirinyer, Buca / İzmir	(kalimelemsal)
Demo	Kullanıcısı	222 22 22	Dokuz Eylül Üniversitesi, Tinaztepe Kampusu	(kalite)
Melih	Yardımcı	444 44 44	Çankaya Konak / İzmir	(melihyardimsever)
Mustafa	Enüst	999 99 99	Atakent Mavişehir / İzmir	(mustafaenust)
Selim	Önparmak	888 88 88	Yeşildere Caddesi, Buca / İzmir	(selimonparmak)

**Sistem Menüü**

Ana Sayfa

Doküman Gezgini

Yeni Doküman Ekle

Profil Bilgilerini Değiştir

Yönetim

**Bekleyen Görevler**

## EK 9: KATEGORİ YÖNETİMİ

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Sistem Menüsü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir  
Yönetim

Bekleyen Görevler

Ana Sayfa > Yönetim > Kategori Yönetimi

### Kategori Yönetimi

PLANLAR	Değiştir Sil
FORMLAR	Değiştir Sil
PROSEDÜRLER	Değiştir Sil
LİSTELER	Değiştir Sil
TALİMATLAR	Değiştir Sil
SÜREÇLER	Değiştir Sil
TABLOLAR	Değiştir Sil

Yeni Kategori:

## EK 10: YENİ DOKÜMAN EKLEME PANELİ - 1

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Sistem Menüsü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir  
Yönetim

Bekleyen Görevler

Ana Sayfa > Yeni Doküman Ekle

### Yeni Doküman Ekleme Paneli

Doküman Adı:

Kategori:

Hazırlayan:

Açıklamalar:

Onaylayacak Kişi:

Termin Zamanı:

Yüklenecek Doküman:

Haz Temmuz 2008 Ağu

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

## EK 11: YENİ DOKÜMAN EKLEME PANELİ - 2

Yeni Doküman Ekle - Windows Internet Explorer

http://localhost:1298/ekys/YeniDokuman.aspx

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Yeni Doküman Ekle

Sistem Menü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir  
Yönetim

Bekleyen Görevler

Doküman Adı: Düzeltilen Faaliyetler Prosedürü

Kategori: PROSEDÜRLER

Hazırlayan: Kalite Yönetim Temsilcisi

Açıklamalar: Düzeltilen faaliyetlerle ilgili prosedürün yer aldığı doküman.

Onaylayacak Kişi: Genel Müdür

Termin Zamanı: 14 Temmuz 2008

Yüklenecek Doküman: C:\Documents and Settings\Kadir Kırdar\Desktop\Gözet...

Dağıtım Listesi (ctrl karakterine basılı olarak birden fazla seçebilirsiniz)

Kalite Yönetim Temsilcisi  
Sekreter  
Bilgi İşlem Sorumlusu  
Genel Müdür  
Müdür Yardımcısı  
Güvenlik Görevlisi

Dokümanı Ekle

## EK 12: DOKÜMAN GEZGİNİ VE BEKLEYEN GÖREVLER

Doküman Gezgini - Windows Internet Explorer

http://localhost:1298/ekys/DokumanGezgini.aspx

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Doküman Gezgini

Hosgeldin Mustafa Enüstü! | Çıkış Yap

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Sistem Menü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler

Bekleyen onayları görmek için tıklayın.

1 adet yeni doküman onaylanmayı bekliyor

Klasörler

PLANLAR İçeriği Listele

FORMLAR İçeriği Listele

PROSEDÜRLER İçeriği Listele

LİSTELER İçeriği Listele

TALİMATLAR İçeriği Listele

SÜREÇLER İçeriği Listele

TABLOLAR İçeriği Listele

## EK 13: ONAY BEKLEYEN DOKÜMANLAR SAYFASI

Onay Bekleyenler - Windows Internet Explorer

http://localhost:1298/ekys/OnayBekleyenler.aspx

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Onay Bekleyen Dokümanlar

Onay bekleyen dokümanın adını tıklayarak ilgili sayfada onaylama işlemini gerçekleştirebilirsiniz.

Doküman	Kategori	Hazırlayan	Açıklama	Revizyon
Düzeltilici Faaliyetler Prosedürü	PROSEDÜRLER	Kalime Temsal Kalite Yönetim Temsilcisi	Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.	<input type="checkbox"/>

Sistem Menüü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler

Bekleyen onayları görmek için tıklayın.

1 adet yeni doküman onaylanmayı bekliyor

## EK 14: ONAY BEKLEYEN YENİ DOKÜMAN

Onaylanamamız İçin Bekliyor - Windows Internet Explorer

http://localhost:1298/ekys/OnayBekleyenDokuman.aspx?id=51831df3-7786-4269-a99a-9cb5872add14

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Onay Bekleyen Doküman (Yeni Doküman)

Doküman Adı: Düzeltilici Faaliyetler Prosedürü

Kategori: PROSEDÜRLER

Yayın Tarihi / Revizyon Tarihi: Onaylanmadı / Revize Edilmedi

Revizyon No: Revize Edilmedi

Hazırlayan Adı / Unvanı: Kalime Temsal / Kalite Yönetim Temsilcisi

Açıklamalar: Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.

Yayın Durumu:

Dosya Bağlantısı: Düzeltilici Faaliyetler Prosedürü

Dokümanı Onayla Reddet

Sistem Menüü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler

Bekleyen onayları görmek için tıklayın.

1 adet yeni doküman onaylanmayı bekliyor

## EK 15: ONAYLANAN DOKÜMANIN BİLGİLERİ

ana sayfa doküman gezgini

**e-kys**  
web tabanlı dokümantasyon sistemi

**Sistem Menüsü**

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

**Bekleyen Görevler**

**Doküman Bilgileri**

Doküman Adı: Düzeltilici Faaliyetler Prosedürü  
Kategori: PROSEDÜRLER  
Yayın Tarihi/Revizyon Tarihi: 04 Temmuz 2008 Cuma/Revize Edilmedi  
Revizyon No: Revize Edilmedi  
Sorumlu Unvan: Kalite Yönetim Temsilcisi  
Hazırlayan Adı/Unvanı: Kalite Temsal / Kalite Yönetim Temsilcisi  
Onaylayan Adı/Unvanı: Mustafa Enüst / Genel Müdür  
Açıklamalar: Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.  
Yayın Durumu:   
Dosya Bağlantısı: Düzeltilici Faaliyetler Prosedürü  
Yönetimsel Araçlar: Revizyon talebi

## EK 16: DOKÜMAN GEZGİNİNDE GÖRÜNÜM

ana sayfa doküman gezgini

**e-kys**  
web tabanlı dokümantasyon sistemi

**Sistem Menüsü**

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

**Bekleyen Görevler**

Ana Sayfa > Doküman Gezgini

**Klasörler**

PLANLAR	İçeriği Listele
FORMLAR	İçeriği Listele
<b>PROSEDÜRLER</b>	<b>İçeriği Listele</b>
LISTELER	İçeriği Listele
TALIMATLAR	İçeriği Listele
SÜREÇLER	İçeriği Listele
TABLOLAR	İçeriği Listele

**Doküman Listesi**

Doküman	Düzenleyen	Son Düzenleme
Düzeltilici Faaliyetler Prosedürü	Kalite Yönetim Temsilcisi	04.07.2008

## EK 17: REVİZYON TALEP ETME

**Doküman Bilgileri**

Doküman Adı: Düzeltilici Faaliyetler Prosedürü  
Kategori: PROSEDÜRLER  
Yayın Tarihi/Revizyon Tarihi: 04 Temmuz 2008 Cuma/Revize Edilmedi  
Revizyon No: Revize Edilmedi  
Sorumlu Unvan: Kalite Yönetim Temsilcisi  
Hazırlayan Adı/Unvanı: Kalime Temsal / Kalite Yönetim Temsilcisi  
Onaylayan Adı/Unvanı: Mustafa Enüst / Genel Müdür  
Açıklamalar: Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.

Yayın Durumu:   
Dosya Bağlantısı: Düzeltilici Faaliyetler Prosedürü  
Yönetimsel Araçlar: Revizyon talebi

**Revizyon Talebi**

Revizyon Talep nedenini yazın.

Dokümanın genel nedenlerden dolayı revize edilmesini talep ediyorum.

## EK 18: BEKLEYEN REVİZYON TALEPLERİ

**Revizyon Talepleri**

Revizyon Talep Eden	Açıklama	Talep Tarihi	
Görüntüle	Selim Onparmak	Dokümanın genel nedenlerden dolayı revize edilmesini talep ediyorum.	04.07.2008

Dokümanı revize etmek için tıklayın.

1 adet revizyon talebi bulunuyor.

**Doküman Bilgileri**

Doküman Adı: Düzeltilici Faaliyetler Prosedürü  
Yayın Tarihi: 04.07.2008  
Revizyon Tarihi: 04.07.2008  
Revizyon No: 0  
Kategori: PROSEDÜRLER  
Hazırlayan Kişi: Kalime Temsal / Kalite Yönetim Temsilcisi  
Onaylayan Kişi: Mustafa Enüst / Genel Müdür  
Açıklama: Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.  
Dosya Bağlantısı: Düzeltilici Faaliyetler Prosedürü

## EK 19: BEKLEYEN REVİZYON TALEPLERİ

Revizyon Talepleri - Windows Internet Explorer

http://localhost:1298/ekys/RevizyonTalepleri.aspx

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Revizyon Talepleri

Sistem Menü

Ana Sayfa

Doküman Gezini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir  
Yönetim

Bekleyen Görevler

Revizyon taleplerini görmek için tıklayın.

1 adet revizyon talebi bulunuyor.

Revizyon Talep Eden

Görüntüle Setim Onparmak Açıklama Talep Tarihi

Dokümanın genel nedenlerden dolayı revize edilmesini talep ediyorum. 04.07.2008

Onaylayacak Unvan Genel Müdür

Haz Temmuz 2008 Ağu

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Termin Zamanı

Yüklenecek Doküman

C:\Documents and Settings\ Gözet...  
Revizyonu Onaya Gönder

Doküman Adı	Düzeltilici Faaliyetler Prosedürü
Yayın Tarihi	04.07.2008
Revizyon Tarihi	04.07.2008
Revizyon No	0
Kategori	PROSEDÜRLER

Internet %100

## EK 20: ONAY BEKLEYEN REVİZYON

Onaylanmaz İçin Bekliyor - Windows Internet Explorer

http://localhost:1298/ekys/OnayBekleyenDokuman.aspx?id=6c4b9eeb-7711-4d8e-91ed-7a12b1114855

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Onaylanmaz İçin Bekliyor

Hosgeldin Mustafa Enüstü! | Çıkış Yap

ana sayfa doküman gezini

e-kys  
web tabanlı dokümantasyon sistemi

Sistem Menü

Ana Sayfa

Doküman Gezini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler

Bekleyen onayları görmek için tıklayın.

1 adet revizyon onaylanmayı bekliyor.

Onay Bekleyen Doküman (Revizyon)

Doküman Adı	Düzeltilici Faaliyetler Prosedürü <<Güncel Sürümü Görüntüle>>
Kategori	PROSEDÜRLER
Yayın Tarihi / Revizyon Tarihi	04 Temmuz 2008 Cuma / Revize Edilmedi
Revizyon No	Revize Edilmedi
Hazırlayan Adı / Unvanı	Kalime Temsal / Kalite Yönetim Temsilcisi
Açıklamalar	Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.
Yayın Durumu	<input checked="" type="checkbox"/>
Dosya Bağlantısı	Düzeltilici Faaliyetler Prosedürü

Dokümanı Onayla Reddet

Internet %100



## EK 21: REVİZYONU ONAYLANMIŞ DOKÜMAN

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

**Doküman Bilgileri**

Doküman Adı: Düzeltilici Faaliyetler Prosedürü  
Kategori: PROSEDÜRLER  
Yayın Tarihi/Revizyon Tarihi: 04 Temmuz 2008 Cuma / 04 Temmuz 2008 Cuma  
Revizyon No: 1  
Sorumlu Unvan: Kalite Yönetim Temsilcisi  
Hazırlayan Adı/Unvanı: Kalime Temsal / Kalite Yönetim Temsilcisi  
Onaylayan Adı/Unvanı: Mustafa Enüst / Genel Müdür  
Açıklamalar: Düzeltilici faaliyetlerle ilgili prosedürün yer aldığı doküman.  
Yayın Durumu:   
Dosya Bağlantısı: Düzeltilici Faaliyetler Prosedürü  
Yönetimsel Araçlar: Revizyon talebi

Sistem Menüsü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler

## EK 22: REVİZYON LİSTESİ

ana sayfa doküman gezgini

e-kys  
web tabanlı dokümantasyon sistemi

Ana Sayfa > Doküman Gezgini

**Klasörler**

Revizyon Listesi - Windows Internet Explorer

Doküman	Format	Yayın Tarihi	Hazırlayan	Onaylayan
Düzeltilici Faaliyetler Prosedürü	[PDF]	04-07-2008, 23:35	Kalime Temsal / Kalite Yönetim Temsilcisi	Mustafa Enüst / Genel Müdür
Düzeltilici Faaliyetler Prosedürü	[PDF]	04-07-2008, 23:51	Kalime Temsal / Kalite Yönetim Temsilcisi	Mustafa Enüst / Genel Müdür

Sistem Menüsü

Ana Sayfa

Doküman Gezgini  
Yeni Doküman Ekle  
Profil Bilgilerini Değiştir

Bekleyen Görevler