

İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

UZAKTAN RANDEVU TAKİP MODÜLÜ İÇEREN BİR DİŞ HEKİMLİĞİ HASTANE YÖNETİM BİLİŞİM SİSTEMİNİN GELİŞTİRİLMESİ

Bilgisayar Bilimleri Müh. Önder GÖÇER
Enformatik Anabilim Dalı
Enformatik Programı

Danışman
Yrd.Doç.Dr. Sevinç GÜLSEÇEN

Ekim, 2006

İSTANBUL



İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

UZAKTAN RANDEVU TAKİP MODÜLÜ İÇEREN BİR DİŞ HEKİMLİĞİ HASTANE YÖNETİM BİLİŞİM SİSTEMİNİN GELİŞTİRİLMESİ

Bilgisayar Bilimleri Müh. Önder GÖÇER
Enformatik Anabilim Dalı
Enformatik Programı

Danışman
Yrd.Doç.Dr. Sevinç GÜLSEÇEN

Ekim, 2006

İSTANBUL

Bu çalışma 04.10.2006 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı Enformatik programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Yrd.Doç. Sevinç GÜLSEÇEN (Danışman)
İstanbul Üniversitesi
Fen Fakültesi

Doç.Dr. Hülya ÇALIŞKAN
İstanbul Üniversitesi
Fen Fakültesi

Doç.Dr. Mehpare TİMOR
İstanbul Üniversitesi
İşletme Fakültesi

Y.Doç.Dr. Zerrin AYVAZ REİS
İstanbul Üniversitesi
Hasan Ali Yücel Eğitim Fakültesi

Y.Doç.Dr. Zuhal TANRIKULU
Boğaziçi Üniversitesi
Uygulamalı Bilimler Yüksek Okulu

ÖNSÖZ

Yüksek lisans tez çalışmam boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Yrd.Doç.Dr.Sevinç GÜLSEÇEN'e en içten dileklerimle teşekkür ederim.

Çalışmamın uygulama geliştirme aşamasındaki değerli katkılarından dolayı arkadaşım Gökhan BALCI'ya teşekkür ederim.

Çalışmamın Uzaktan Randavu Takip Bölümünün geliştirme aşamasındaki değerli katkılarından dolayı Netisis Bilgi Teknolojileri Ltd.Şti. çalışanlarına teşekkür ederim.

Bu çalışma boyunca katkılarını esirgemeyen ve 1993 yılından günümüze Diş Hekimliği Hastane Yönetim Sistemi konusunda bilgi dağarcığıma katkıda bulunan tüm İstanbul Üniversitesi Diş Hekimliği Fakültesi Akademik ve İdari Personeli'ne teşekkürü borç bilirim.

Yine tez çalışmam süresince gösterdikleri anlayış ve sabır için aileme teşekkür ederim.

Ekim, 2006 **Önder GÖÇER**

İÇİNDEKİLER

ÖNSÖZ I

İÇİNDEKİLER II

ŞEKİL LİSTESİ V

TABLO LİSTESİ VI

EK LİSTESİ VII

KISALTMALAR VIII

ÖZET IX

SUMMARY X

1. GİRİŞ 1

2. GENEL KISIMLAR 3

2.1. YÖNETİMDE BİLGİ TANIMI 3

2.1.1. Yönetimde Bilgi ile İlgili Kavramlar 3

2.1.2. Küreselleşme ve Bilgi Toplumu 6

2.1.3. Bilginin Nitelikleri 8

2.1.3.1. İyi Bilginin Nitelikleri 9

2.1.4. Bilişim Sistemleri 10

2.2. TEMEL BİLİŞİM SİSTEMLERİ 12

2.2.1. Elektronik Veri İşlem Sistemleri 13

2.2.2. Ofis Otomasyon Sistemleri 15

2.2.3. Yönetim Bilişim Sistemleri 15

2.2.4. Karar Destek Sistemleri 16

2.2.5. Üst Yönetim Bilgi Sistemleri 17

2.2.6. Uzman Sistemler 18

2.3. YÖNETİM BİLİŞİM SİSTEMİ 20
2.3.1. Yönetim Bilişim Sistemi Geliştirme Çalışmasında İzlenecek Adımlar 25
2.3.2. Yönetim Bilişim Sistemi'nin Uygulama Sorunları 28
2.3.3. Yönetim Bilişim Sistemi Kuruluşunda Dikkate Alınması Gereken Hususlar 31
2.4. HASTANE YÖNETİM BİLİŞİM SİSTEMİ 32
2.4.1. Yönetimi Desteklemeye İlişkin Sistemler 34
2.4.1.1. Kaynak Kullanımı ve Programlama Sistemi 34
2.4.1.2. Mali Yönetim Bilişim Sistemi 35
2.4.1.3. Malzeme ve Tesis Yönetim Bilişim Sistemi 36
2.4.1.4. Personel Yönetim Bilişim Sistemi 38
2.4.2. Tanı Ve Tedaviyi Desteklemeye İlişkin Sistemler 39
2.4.2.1. Bilgisayar Destekli Tanı ve Tedavi Bilişim Sistemi 40
2.4.2.1. Bilgisayar Destekli Hasta Takip Bilişim Sistemi 40
2.4.2.1. Bilgisayar Destekli Hastalık Kayıtlarını Tutma Bilişim Sistemi 40
2.4.2.1. Bilgisayar Destekli Laboratuvar ve Eczane Bilişim Sistemi 41
2.4.3. Entegre Hastane Yönetim Bilişim Sistemi 42
2.4.4. Modüler Hastane Yönetim Bilişim Sistemi 45
2.5. HASTANE YÖNETİM BİLİŞİM SİSTEMLERİNDE KULLANILAN TEKNOLOJİLER 46
2.5.1. Veri Tabanı Teknolojisi 46
2.5.2. Programlama 50
2.5.3. Bilgi İletişim Teknolojileri 52
3. MALZEME VE YÖNTEM 57
3.1. UYGULAMA YÖNTEMİ 57
3.1.1. Problemin Tanımı 57
3.1.1.1. Ülkemizde Sağlık Sektörünün Yapısı 57
3.1.1.2. Ülkemizde Ağız-Diş Sağlığı Hizmetleri 57
3.1.1.3. Diş Hekimliği Alanında Bilişim Hizmetleri Kullanımı 58
3.1.1.4. Örnek Uygulama İçin Kurum Seçimi 59
3.1.1.5. İstanbul Üniversitesi Diş Hekimliği Fakültesine Ait Büyüklükler 61
3.1.1.6. Yurtdışı Örnek Uygulama 62

3.1.2. Analiz 63
3.1.2.1. Kullanıcı Giriş 64
3.1.2.2. Ana Sayfa 64
3.1.2.3. Genel Programlar 65
3.1.2.4. Hasta Kayıt 66
3.1.2.5. Oral Diagnoz 67
3.1.2.6. Klinik Takip 68
3.1.2.7. Fatura İşlemleri 69
3.1.2.8. Makbuz Basımı 70
3.1.2.9. Makbuz İşlemleri 71
3.1.2.10. Tanımlamalar 72
3.1.2.11. Uzaktan Randevu Takip 73
3.1.3. Tasarım 75
3.1.4. Kod Geliştirme 77
3.1.5. Veri Tabanı 77
3.2. MALZEME 77
3.2.1. Kodlama Dili 77
3.2.2. Veri Tabanı 77
3.2.3. İşletim Sistemi 77
4. BULGULAR 78
5. TARTIŞMA VE SONUÇ 80
KAYNAKLAR 82
EKLER 85
ÖZGEÇMİŞ 194

ŞEKİL LİSTESİ

Şekil 2.1 : Bilgi Hiyerarşisi 5
Şekil 2.2 : Bilişim Sistemleri Öğelerine Bütünsel Bakış 11
Şekil 2.3 : Temel İşletme Bilişim Sistemleri ve Alt Sistemleri 24
Şekil 3.1 : İstanbul Üniversitesi Diş Hekimliği Fakültesi İdari Yapılanması 59
Şekil 3.2 : İstanbul Üniversitesi Diş Hekimliği Fakültesi Akademik Yapılanması 60
Şekil 3.3 : Michigan Üniversitesi Dişçilik Okulu "Diş Hekimliği Enformatik Klinik Uygulama Mimarisi"..... 63
Şekil 3.4 : Kullanıcı Giriş Ekranı 64
Şekil 3.5 : Yetkisiz Kullanıcı Uyarı Mesajı..... 65
Şekil 3.6 : Hasta Kayıt Program Görünümü..... 66
Şekil 3.7 : Oral Diagnoz Anamnez ve Oral Görünüm Bilgileri Giriş Ekranı..... 67
Şekil 3.8 : Oral Diagnoz-Ağız Diş Şeması ve Tedavi Planlaması ile Tedavi Kodları Ekranı 68
Şekil 3.9 : Klinik Takip Ekranı ve Diş Şeması..... 69
Şekil 3.10 : Fatura Kayıt, Fatura İşlem ve Fatura Detay Ekranları 70
Şekil 3.11 : Klinik Takip Ekranında Makbuz Basımı 71
Şekil 3.12 : Kullanıcı Tanımlama Ekranı..... 73
Şekil 3.13 : Randevu Talep Giriş Ekranı..... 74
Şekil 3.14 : Randevu Bilgi Giriş Ekranı..... 75

TABLO LİSTESİ

Tablo 2.1 : İş Dünyasının Çağdaş Devrimi 8
Tablo 2.2 : Organizasyonlarda Bilişim Sistemleri Uygulamaları ve Bilgi Kullanımı 14
Tablo 3.1 : İstanbul Üniversitesi Diş Hekimliği Fakültesi Veri Büyüklüğü 61
Tablo 3.2 : İstanbul Üniversitesi Diş Hekimliği Fakültesi Statü Durum Tablosu..62

EK LİSTESİ

Ek-A. : Diş Tedavileri Fiyat Tarifesi-2006 85
Ek-B. : ICD-10 Tanı Kodları 90
Ek-C. : Masaüstü Uygulama Programı Kaynak Kodu 100
Ek-D. : Uzaktan Randevu Takip Programı Kaynak Kodu 131
Ek-E. : Tablo yapıları ve ilişki diagramı 186

KISALTMALAR

Tez içinde kullanılan kısaltmalar:

ABD	Ana Bilim Dalı
ANSI	American National Standards Institute
ASP	Active Server Pages
BS	Bilişim Sistemi
DBMS	Data Base Management System
DHYBS	Dişhekimliği Hastane Yönetim Bilişim Sistemi
EVİS	Elektronik Veri İşlem Sistemi
FIPS	Federal Information Processing Standard
HYBS	Hastane Yönetim Bilişim Sistemi
ISO	International Standards Organization
KDS	Karar Destek Sistemi
MYBS	Mali Yönetim Bilişim Sistemi
OOS	Ofis Otomasyon Sistemi
PERT	Project Evaluation and Review Technique
PYBS	Personel Yönetim Bilişim Sistemi
SQL	Structured Query Language
TKS	Ticari Kayıt Sistemi
URT	Uzaktan Randevu Takip
US	Uzman Sistem
ÜDYBS	Üst Düzey Yönetici Bilgi Sistemi
VTYS	Veri Tabanı Yönetim Sistemi
www	World Wide Web
YBS	Yönetim Bilişim Sistemi
YZ	Yapay Zeka

ÖZET

UZAKTAN RANDEVU TAKİP MODÜLÜ İÇEREN BİR DİŞ HEKİMLİĞİ HASTANE YÖNETİM BİLİŞİM SİSTEMİNİN GELİŞTİRİLMESİ

Yönetim Bilişim Sistemleri her işletmede olduğu gibi, hastanelerin de kuruluş amaçları doğrultusunda etkili ve verimli faaliyet göstermelerinde yöneticilerin verecekleri doğru kararları destekleyecek en önemli bileşen olarak kabul edilmektedir.

Hastane etkinliğinin attırılmasında Hastane Yönetim Bilişim Sistemi stratejik bir rol oynamaktadır. Bu işlevini; iletişimi geliştirerek, yapılan işleri otomatikleştirerek ve işlerin yapılış şeklinde dönüşüm sağlayarak gerçekleştirmektedir.

Diş Hekimliği Hastane Yönetim Bilişim Sistemi kurulmasında ve geliştirilmesinde genel kabul görmüş yöntemlere uyulması, amaca uygun sağlıklı altyapı ve donanımın kurulması, yazılımın örgütle birlikte yaşayan ve gelişen bir bileşen olduğu da göz ardı edilmeksizin doğru şekilde yapılandırılması verim almayı ve etkin olmayı kaçınılmaz kılacaktır.

Bu çalışmada İstanbul Üniversitesi Diş Hekimliği Fakültesinde mevcut bilişim altyapısı üzerinde sorunsuz çalışabilecek ve Uzaktan Randevu Takip Modülüne sahip bir Diş Hekimliği Hastane Yönetim Bilişim Sisteminin geliştirilmesi amaçlanmıştır.

SUMMARY

THE IMPLEMENTATION OF A DENTAL HOSPITAL INFORMATION MANAGEMENT SYSTEM INCLUDING REMOTE PATIENT APPOINTMENT FACILITY

Management Information Systems like in every enterprise, are accepted the most important component to support the right decisions that the administrators make effectively and efficiently in the aims of establishing the dental hospitals.

Hospital Management Information System plays a strategical role in increasing hospital efficiency. This function is realized by developing communication, automatizing the works and supplying transformation how the works done.

It will be unavoidable to have success by obeying accepted techniques establishing and improving the Dental Faculty Hospital Administration System, establishing suitable infrastructure and the equipments, without ignoring the importance of the software that lives and develops with the organization.

In this research, it is aimed to develop a Dental Hospital Management Information System that has Remote Appointment Following Module and a system that can work on current informatics structure without any problems at Istanbul University Dental Faculty.

1. GİRİŞ

İşletmelerin yönetiminde; planlama, strateji oluşturma, karar verme ve denetleme süreçlerinin temel öğelerinden biri “bilgi”dir. Sermaye, işgücü, makine gibi somut kaynaklara son zamanda stratejik önem atfedilen kavramsal bir kaynak olarak ’bilgi’ eklenmiştir. Verimliliğin, kalitenin ve yeni teknolojileri kullanmanın günümüz rekabetçi ortamında başarı önkoşulu olarak algılandığını ve ’bilgi’nin işletme yönetiminde stratejik temel öğe olarak karşımıza çıktığını görmekteyiz.

Sağlık sistemi içerisinde önemli bir yeri olan hastanelerin yönetiminde de; rekabetin artması, harcamaların önü alınmaz bir şekilde yükselmesi, kalite ve verimlilik arayışları kaynakları daha etkin şekilde kullanmayı zorunlu kılmaktadır. Tüm bu amaçlara ulaşmada ‘Hastane Yönetim Bilişim Sistemi’nin ne denli vazgeçilmez bir gereklilik olduğu günümüzde açıkça ortadadır.

Diş Hekimliği Hastaneleri, sağlık sistemimiz içerisinde Tedavi Edici Sağlık Hizmetleri kapsamında değerlendirilerek; Özel teşebbüs, Özel Dal Devlet Hastanesi ya da Üniversite Hastanesi olarak yer almaktadır.

İşlevsel olarak sağlık hizmeti üreten diğer hastanelerle paralellik görünse de; hizmetin niteliği, örgütsel birimlerin farklılığı, teşhis ve tedavi aşamasında takibi yapılan bilgilerin faklılığı ile temel bir Hastane Yönetim Bilgi Sistemi (HYBS) içerisinde ihtiyaçları karşılanamamaktadır.

Ülkemizde HYBS üzerine yapılmış akademik çalışmalar bulunmakla birlikte, Diş Hekimliği Hastane Yönetim Bilgi Sistemi üzerine yapılmış bir çalışmaya rastlanmamıştır. Bu tez çalışmasının bu eksikliği gidererek, hastane yöneticilerinin ve yönetim kademesinde yer alanların dikkatini “Yönetim Bilişim Sistemine” çekerek, bilgiyi elde etme ve kullanma anlayışında farkındalık yaratacağına inanılmaktadır.

Tezin diğer bölümleri şu şekilde ilerlemektedir;
İkinci bölümde yönetimde bilgi tanımı, bilgi ile ilgili kavramlar, Küreselleşme ve bilgi toplumu, bilginin nitelikleri ve bilişim sistemleri kavramları üzerinde durularak, temel bilişim sistemleri, yönetim bilişim sistemleri, hastane yönetim bilişim sistemeleri ve hastane yönetim bilişim sistemlerinde kullanılan teknolojiler incelenmiştir.

Üçüncü bölümde Problemin tanımı yapılarak uygulama yöntemi tanıtılarak kullanılan malzeme hakkında bilgi verilmiştir.

Dördüncü bölümde standartlar üzerinde durulmuştur.

Beşinci bölümde sonuçlar incelenerek geleceğe yönelik araştırmalar için önerilerde bulunulmuştur.

2. GENEL KISIMLAR

2.1. YÖNETİMDE BİLGİ TANIMI

2.1.1. Yönetimde Bilgi ile İlgili Kavramlar

Büyük Larouse Sözlük ve Ansiklopedisi (1992) bilgiyi “düşünme, yargılama, akıl yürütme, okuma, araştırma, gözlem ve deney sonucunda elde edilen düşünsel ürün” olarak tanımlamaktadır.

Türkçe akademik çalışmalarda ulaşılabilen ortak tanım H. Ülgen’in 1977 yılındaki “Yönetim Bilişim Sistemi” adlı çalışmasında ortaya koyduğu şekliyle benimsendiği görülmektedir. Bu tanıma göre; bilgi kavramının, latince "informatio" kökünden olduğu, biçim verme eylemi olarak tanımlandığı ve kişinin veriye yönelttiği anlam şeklinde de tanımlandığını görüyoruz. Yönetim alanında ise bilgi "yöneticinin karar almasına yardımcı olan öğeler" şeklinde tanımlanmaktadır. Daha dar anlamıyla işletme yönetiminde bilgi; "raporlar halinde üst düzeylere çıkan, emirler şeklinde alt kademelere inen günlük faaliyetler aracılığı ile işletmenin işlevsel bölümlerini birbirine bağlayan, müşteri ve ortak ilişkileriyle işletme dışına yönelen ve nihayet hesap durumu çizelgeleri ve basın yayın organlarıyla işletme içine yönelen bir olgu olarak ifade edilmektedir.

Develi (2006), Karahanlılar’dan bize kadar gelmiş, bilinen ilk Türkçe eser Balasagunlu Yusuf Has Hacib’in “Kutatgu Bilig” adlı eserinin ana fikrini de bir anlamda bilginin yüceliği, bilgili olan insanın hayvandan ayrılan ve yücelen insan olduğu fikrinin bir zincir halinde İbni Sina ve Farabi yoluyla antik dönem filozofları Aristo ve Eflatun’a kadar ulaştığını Dilaçar’ın Kutadgu Bilig incelemesinde ileri sürüldüğünü aktarmaktadır.

Öğüt’e (2003) göre bilgi; toplanmış, organize edilmiş, yorumlanmış ve belli bir yöntemle etkin karar vermeyi gerçekleştirmek amacıyla ilgili birime sevk edilmiş,

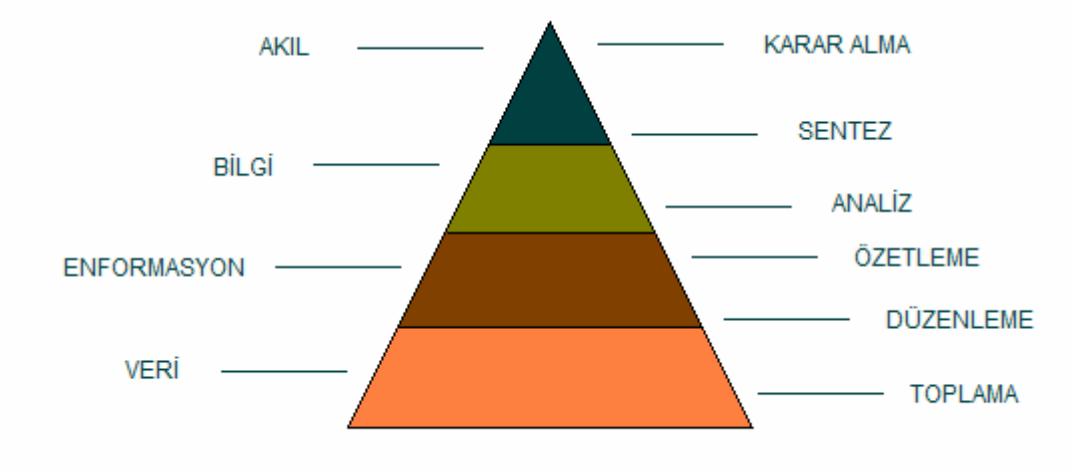
belirli bir amaç doğrultusunda süreçlenen, yararlı biçime dönüştürülmüş ve kullanıcıya değer sağlayan verilerdir. Kısaca, veri, kararları ve davranışları etkilediği zaman bilgi olmaktadır. Bilginin formları şunlardır; veri (data), yazı (text), ses (sound) ve resim (image). Diğer yandan, bilgi ile ilgili işlevler ise; bilginin üretilmesi, süreçlendirilmesi, depolanması ve iletilmesidir.

Yönetimde bilgi, belirli amaçlara ulaşmak veya belirli bir anlayışı geliştirmek için verilerin ya da ham bilgilerin bir işlem sonucunda yöneticilere yararlı biçime sokulmuş şeklidir. Bu tanımdan bilginin bir dönüşüm sürecinin sonucu olduğu görülmektedir. Şu halde, hammaddenin bir üretim süreci sonunda ürün haline dönüştürülmesine benzer biçimde veri ya da işlenmemiş bilgiler de veri işleme süreci sonucu bilgiye dönüştürülebilir. Bu arada üretim süreci sonunda elde edilen ürünler satılıp, değerlendirilmedikçe bir yarar sağlamadığı gibi, veri işleme süreci sonunda elde edilen bilgi de yerinde kullanılmayıp, yönetim faaliyetleri ve karar almayı desteklemedikçe bir değer kazanamayacaktır (Çimen ve Ateş, 2006).

Bir örgütün yaşaması için gerekli unsurlardan en önemlisi bilgidir. Nitekim en küçüğünden en büyüğüne kadar tüm kurum ve kuruluşlarda bilgi toplama, kaydetme, saklama, yeniden bulma ve dağıtıma ilişkin bir sistemin gereği açıktır.

Barutçugil'in (2002) "Bilgi Yönetimi" isimli çalışmasında veri, enformasyon ve bilgi kavramlarının tanımı kısaca aşağıdaki gibi verilerek, bilginin farklı nitelikleri olan unsurların birbirleriyle bağlantıları dikkate alınarak tanımlanması, bu ayrımların yapılmaması durumunda kavramların birbirleri yerine, yanlış kullanılmalarına sebebiyet verdiğine dikkat çekilerek, aralarındaki ilişkiler Şekil 2.1'de belirtilmektedir (Önder, 2005).

Veri (Data): Olaylara ilişkin nesnel gerçekler olup birbiriyle ilişkilendirilmemiştir. Veri kurumsal amaçlara bağlı olarak işlemlerin yapılandırılmamış bir biçimde kaydedilmesidir. Modern kurumlarda veri, teknolojik sistemlerde saklanır. Veri özümlenmemiş ve yorumlanmamış gözlemler, işlenmemiş gerçekler olarak tanımlanabilir. Çoğu kez bir anlamı, içeriği yoktur.



Şekil 2.1. Bilgi Hiyerarşisi (Barutçugil, 2002)

Enformasyon: Düzenlenmiş veri olarak tanımlanabilir. Daha geniş anlamda, son kullanıcılar için anlamlı ve yararlı veri bütünü olarak ifade edilebilir. Yalnızca ilgili kişi için bir anlam taşımaktadır. Veriden daha zengin bir ifadeye sahip olan enformasyon, yazılı, sözlü veya görsel bir mesajdır. Veriyi katma değer ekleyerek enformasyona dönüştürürken; bağlam, sınıflandırma, hesaplama, düzeltme ve yoğunlaştırma gibi zihinsel süreçlerden geçirilir.

Bilgi: Kişisel anlamda düzenlenmiş ve özümlenmiş enformasyondur. Öğrenme ve deneyim yoluyla kazanılmış olan önceki bilgilerle bütünleşmiştir. Kararlara ve davranışlara yol gösterir. İnsanlar arasında iletişim yoluyla enformasyon akışı bilginin yaratılmasını sağlar. Bilgi deneyim ve değerlere ilişkin enformasyonun akışkan bir karmasıdır. Karşılaştırmalar, varılan sonuçlar, ilişkilendirmeler ve iletişim enformasyonun bilgiye dönüşümü sırasındaki süreçlerdir (Önder, 2005).

Durna'nın (2002) "Yenilik Yönetimi" adlı eserinde; bilgi ve teknolojideki değişimler incelenirken "Bilgi Patlaması" kavramından söz edilerek, 17. yy.'ın ortalarında iki adet bilimsel derginin olduğu, 18. yy'ın ortalarına kadar 10 adet, 1800 yılına kadar 100 adet, 1850 yılına kadar 1.000 adet bilimsel derginin varlığı ve 1980'li yıllarda ise 30.000 ile 100.000 arasında olduğunun tahmin edildiği ve bilginin kendisini beslediği düşünülürse, gelecekte katlanarak artacağının beklendiği ifade edilir. İletişim ve bilgisayar teknolojisindeki ilerlemelerin üretilen bilgi miktarını olağanüstü düzeyde arttıracağından bahsedilerek, bilgi toplumunun itici gücünün bilgi ve bilgiyi işleyen

bilgisayarlar olduğu, bilişim teknolojilerinin bu desteğini alan insanoğlunun, yenilikçiliği ve yaratıcılığı ile bir yandan sorunlara yeni çözümler üretebildiği bir yandan da toplumu yeniden şekillendirdiğinden bahsedilerek, bilgi dağıtım teknolojilerinin toplumun geniş katmanlarında yaygınlaşmasıyla, bilgi çevresinin gelecekte çok daha genişleyeceği ve katılımlı hale geleceği vurgulanmaktadır.

2.1.2. Küreselleşme ve Bilgi Toplumu

Öğüt (2003), bilgiyi bütün ekonomik sistemlerin temelinde yer alan ve bilgiye dayanmayan bir toplumsal ve ekonomik sistem düşünmenin mümkün olmadığını, günümüzde kapsamı ve boyutları genişleyen uluslararası ekonomik savaşın en görkemli silahının bilgi olduğu savunulduğunu, sanayi ekonomisinin kaynaklarının ya da üretim faktörlerinin, toprak, işgücü, sermaye ve girişim gücü iken, bilgi ekonomisinin en merkezi kaynağının bilgi olduğunu ve dolayısıyla, bilgi ekonomisini diğer ekonomilerden ve sanayi ekonomisinden ayıran farkın, bilginin ekonomik üretim faktörleri içinde birincil önceliğe sahip olması ve bilgi teknolojileri yardımıyla bilginin üretimindeki ve kullanımındaki artış olduğunu ifade etmektedir.

Pare ve Sicotte (2001) yönetim bilgi sitemi alanında yapılan birçok araştırmanın, bilgisayar destekli uygulamaların organizasyonel performans üzerinde olumlu etkilerde bulunduğunu ortaya koyduğunu ifade ederler. Bu çalışmalar aynı zamanda, sağlık hizmeti üreten kurumların bu konuda bir istisna olmadıklarını, ancak, sağlık hizmeti üreten kurumların bir çok kendine özgü nedenden dolayı bilgi teknolojileri konusunda "amatör" olduklarını gösterdiğini ve hala birçok tıp merkezinin işlevlerini manuel olarak sürdürdüğünü, günümüz modern devletlerinin sağlık sektöründe çalışan doktorlar ve sağlık yöneticileri, hastalara kaliteli sağlık hizmetini verirken maliyetleri düşük tutmak için bilgi teknolojilerinden başka alternatiflerinin olmadığını kabul ettiklerini belirtmişlerdir (Köse, 2004).

Günümüz ekonomisini ve ekonomik büyümeyi ancak, bilgiyi, değer ve servet üretme sürecinin merkezine yerleştiren bir kuram açıklayabilir. Bilginin katma değer yaratmada ve rekabette en önemli faktör durumuna geldiği günümüzde, bilgi ekonomisinde güç, en kapsamlı bilgi kaynaklarına sahip olan ülkelere ve organizasyonlara doğru yönelmektedir. Bilgi ekonomisinde organizasyonlar, çalışabilmek için daha çok veriye

ve teknik bilgiye gereksinim duymaktadır. Bilgi teknolojilerinden etkin şekilde yararlanamayan devletler, organizasyonlar ve bireyler bilgi ekonomisinde rekabet şanslarını yitireceklerdir.

Önder (2005) “Operasyonel verimliği”; 1950'lerde rekabet avantajı elde etmenin yolu, 1960'larda 'stratejik planlama', 1970'lerde 'pazar pozisyonu', 1980'lerde 'mükemmel uygulamalar' rekabet avantajı sağlayan olgular şeklinde tanımlar. 1980'lere kadar bilgi işlem sürecinin hızı ve önemi çok fazla algılanmamıştı. Bu yıllardan önce yöneticilerin bilginin toplanması, işlenmesi ve organizasyonlara dağıtı1ması hakkında fazla şey bilmesine gerek yoktu, teknolojik yapı minimum düzeydeydi. Bilgi kendi başına, firma için önemli bir varlık olarak düşünülmemekteydi. Yönetim süreci yüz yüze ilişkiler olarak kabul edildi, kişisel yetenek yüksek değildi, global koordinasyon süreci de düşüktü. 1990'lardan bu yana ise işletmelerin sahip oldukları ve geliştirdikleri 'temel yetkinlikler' uzun vadede önem arz eden ve rekabet gücünü arttıran unsurlardı. O halde, uzun vadede sürdürülebilir rekabet avantajı sağlamak, ihtiyaç duyulan sürekli yeni ürünler yaratabilen 'temel yetkinlikler' olarak görülmektedir. 'Bilgi ve bilginin yönetimi' bu temel yetkinliklerin özünü oluşturmaktadır

Laudon (2005), iş dünyasını etkilediğini düşündüğü dünya çapındaki dört büyük değişimi şöyle sıralamaktadır; Global ekonominin ortaya çıkması ve güçlenmesi bu değişimlerden ilkidir. İkincisi, endüstriyel ekonomilerin ve toplumların hızlı bir şekilde bilgi ve enformasyon tabanlı ekonomilere dönüşümüdür. Girişimciliğin farklı bir yapı kazanması üçüncü değişim olarak gösterilmektedir. Dijital şirketlerin ortaya çıkması ile dördüncü değişim de gerçekleşmiştir. Firmalar ve yönetimlerine bir dizi yeni sorgulamalar getiren bu dört değişim Tablo 2.1'de daha açık bir şekilde belirtilmiştir (Önder, 2005).

2.1.3. Bilginin Nitelikleri

Güleş ve Özata (2005), W. Adakeye ve B. Adeoti’den tercümeyle bilgiyi organizasyonu bir arada tutan kaynaştırıcı bir öğe, fiziksel eşya ve dayanıklı tüketim mallarından tamamen farklı olarak, içeriği kaynağına (iç veya dış) ve şekline göre (sayısal olan veya sayısal olmayan) farklı biçimlerde sınıflandırılan, sayısal olmayan bilgilerin yapılandırılmış veya yapılandırılmamış şekillerde olduğunu ifade ederler. İç bilgi bir

organizasyonun içerisinde oluşturulan ve sadece organizasyonun içindeki karar verenler için değer taşıyan bilgidir. Dış bilgi ise organizasyonun dört duvarının dışındaki kişi ve kurumlar tarafından üretilen bilgidir.

Tablo 2.1. İş Dünyasının Çağdaş Devrimi (Laudon, 2002)

Küreselleşme	• Küresel Pazar piyasasında yönetim ve kontrol • Dünya piyasalarında rekabet • Küresel iş grupları • Küresel dağıtım sistemleri
Enformasyon Ekonomisinin Yükselişi	• Bilgi ve enformasyon tabanlı ekonomiler • Yeni ürünler ve hizmetler • Bilgi; önemli üretken ve stratejik varlık • Zamana dayalı rekabet • Kısa ürün hayatı • Değişken iş dünyası • Bilgi bazında sınırlı personel
İşletmelerin Değişimi	• Hiyerarşi kademesinin azaltılması • Yerinde yönetim ve karar verme (Yetki merkeziyetçi değil) • Esneklik • Bağımsız birimler • Düşük işlem ve koordinasyon bağlantıları • Personelin güçlendirilmesi • Ortak işbirliği ve takım çalışması
Dijital İşletmelerin Ortaya çıkması	• Müşteriler, tedarikçiler ve çalışanlarla dijital ortamda bağlantıların kurulması • Önemli iş süreçlerinin bilgisayar ağları tarafından sağlanması • Yönetimde enformasyon teknolojilerinin kullanılması • Çevresel değişimleri hızlı algılama ve reaksiyon göstermek.

Sonsuz olması ve paylaşıldıkça çoğalmasının yanı sıra bilgiyi diger kaynaklardan ayıran iki önemli özellik daha bulunmaktadır (Güleş ve Özata, 2005). Bunlar;

a. Bilgi dinamiktir. Kullanıldığı zaman vardır. Yalnızca kullanılarak statik olarak depolanabilir. Bu açıdan endüstri çağını başlatan elektrik enerjisine benzerlik gösterir.

b. Bilginin değeri görecelidir. Ona ihtiyaç duyan ve onun değerini bilenler için bilgi çok değerlidir. Her yerde herkes için aynı değeri taşımaz.

2.1.3.1. İyi Bilginin Nitelikleri:

İyi bilginin sahip olması gereken özellikleri ise şu şekilde sıralamak mümkündür;

- *Doğruluk:* Bilgi hatalardan arındırılmış olmalıdır. Bilginin gerektirdiği doğruluk derecesi, kararın amacına ve niteliğine göre farklılık gösterecektir.

- *Amaca Uygunluk:* Bilgi karar verilen konu ve konularla ilgili olmalıdır. Her yönetim kademesi ve fonksiyonel birim için gerekli bilginin kapsamı ve detayı farklılık gösterecektir.

- *Zamanlılık:* Bilgi gerekli yer ve zamanda hazır olmalıdır. Bilgi doğru ve uygun olmasına rağmen zamanında gelmemişse anlamı yoktur. Zamanlılık özellikle koşulların sürekli değiştiği ortamda, kararların verilmesinde önemli olmaktadır.

- *Noksansızlık:* Karar vericiye sunulan bilgi tam ve eksiksiz olmalıdır.

- *Denetlenebilirlik:* Bilgi, kendisinin doğruluğunu ve noksansızlığını belirlemeye uygun olmalıdır. Bilginin doğruluğu ve noksansızlığı doğru olarak kabul edilen bilgi ile karşılaştırılarak belirlenebilir. Ancak bilginin doğruluğunun tespit edilebilmesi için çoğunlukla bilginin orijinal kaynağına inmek gerekmektedir.

- *Kısalık:* Bilginin içeriği olabildiğince kısa ve öz olmalıdır. Bilginin kapsamı genişledikçe gereksiz ayrıntılardan dolayı karar vermek güçleşecektir.

- *Güncellik:* Sunulan bilgi karar verilecek en son durumu yansıtmalıdır.

- *Ekonomiklik:* Bilginin bir maliyeti vardır. Bilgi, üretilmesi beklenen değerden daha pahalı olmamalıdır.

Hiçbir şey bilgisiz hareket edemez ve genellikle bilginin büyük bir güç olduğuna ve ona sahip olanın ise güçlü olduğuna inanılır. Modern çağda başarılı olmak için her yönetim faaliyeti mutlaka bilgiye dayanmalıdır.

Gates ve Hamingway (1999) “Dijital Sinir Sistemiyle Çalışmak” adlı eserlerinde bilgi

için; organizasyonları rakiplerinden farklılaştıracak, diğerleriyle araya mesafe koyacak en iyi yöntemin bilgiden kusursuz bir biçimde yararlanabilmek olduğunu, bilgiyi toplama, yönetme ve kullanma biçiminin kazanmayı ve kaybetmeyi belirlediğini ifade etmişlerdir (Güleş ve Özata, 2005).

Şunu da ifade etmek gerekir ki, bilgi miktarının sürekli katlanarak arttğı günümüzde, organizasyonların başarısı bilgiye sahip olma yanında, elde edilen bu bilgilerin en iyi biçimde yönetilmesini gerekli kılmaktadır. Günümüzde bilginin yönetilmesi için kullanılan temel araçlar ise Bilişim Teknolojileri'dir. Sağladığı yüksek bilgi işleme, saklama ve taşıma yeteneği sayesinde bu teknolojiler, organizasyonların etkin, etkili ve verimli çalışmasının anahtarı olmaya başlamıştır.

2.1.4. Bilişim Sistemleri

Bilişim Sistemleri (BS) kavramı ile; bilginin toplanması, saklanması, işlenmesi, erişilmesi ve dağıtılmasına hizmet eden teknolojiler (bilgisayar, veri depolama araçları, ağ ve iletişim araçları, yazılım geliştirme araçları), uygulama ve hizmetlerin (bilgi-işlem, uygulama yazılımı geliştirme, bilgi bankaları ve bilgi erişim hizmetleri) bütünü, sistem üzerindeki bilgiler ve insan kaynakları (davranışsal temeller) kastedilmektedir (Öğüt, 2003).

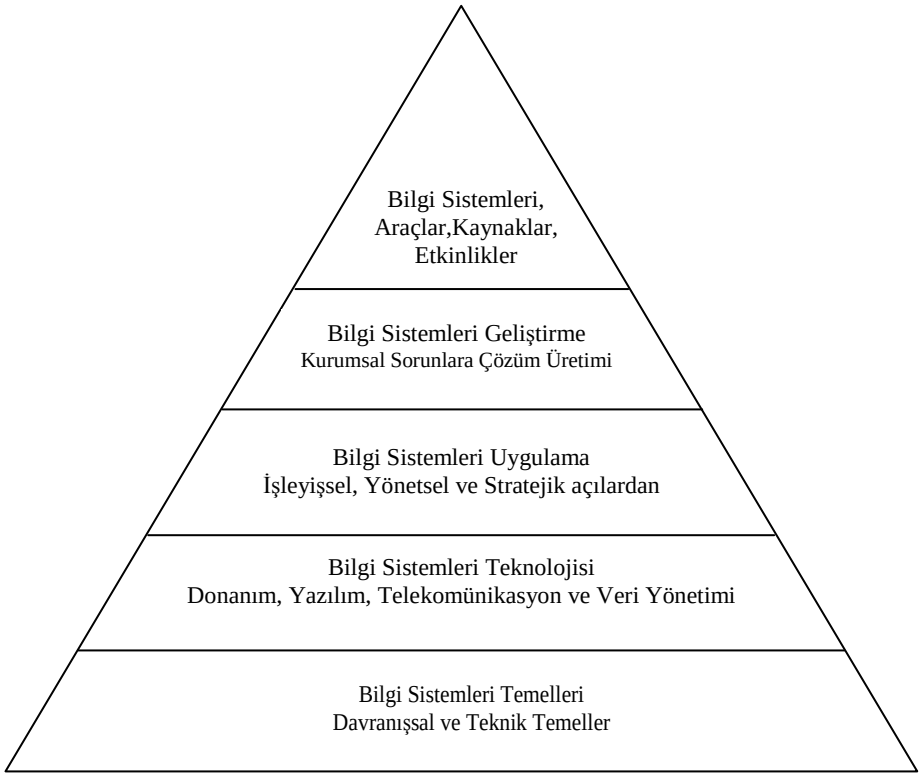
Günümüzde pekçok insan yapısı sistem otomatik olarak işlemekte, büyük bir kısmıda küçük yada büyük bir bilgisayar içermektedir. Bilgi işlemeye yönelik olarak gerçekleştirilen her türlü sisteme BS adı verilmektedir. Bu alandaki teknolojiye de Bilişim Teknolojisi denmektedir (Sarıdoğan, 2004).

Öğüt (2003), J.A. O'Brien'den (1993) yararlanarak Bilgi Sistemleri Öğelerini bütünsel olarak Şekil 2.2. de olduğu gibi ifade etmektedir.

Sarıdoğan (2004), bilgisayar tabanlı olarak gerçekleştirilen otomatik sistemlerin ortak özellikleri arasında bilgisayar donanımı, yazılım, veri kullanım yöntemi ve insan olduğunu vurgulayarak, çok amaçlı olarak geliştirilmiş tek bir bilgisayarın kendi başına bir sistem değil, yalnızca bir araç olduğunu, ancak bu bilgisayarın belirli işler yapmak üzere özel bir yazılım çalıştırıyor ve çeşitli giriş/çıkış cihazlarına bağlantı sağlıyorsa

küçük de olsa gerçek anlamda bir bilgisayar sistemi olarak adlandırılabileceğini ifade eder.

Bensghir (1996), bilişim sistemlerinin öğelerini altı başlık altında şöyle sınıflamıştır; girdi birimi (input unit), çıktı birimi (output unit), merkezi işlem birimi (central processing unit), yazılım (software), prosedürler (procedures) ve insan kaynakları (systems personnel) (Öğüt, 2003).



Şekil 2.2 - Bilgi Sistemleri Öğelerine Bütünsel Bakış (Öğüt, 2003)

- Girdi Birimleri: Bilgi sistemlerinin temel girdisi olan, veriler, girdi birimleri aracılığı ile sisteme aktarılmaktadır. Girdi birimleri olarak günümüzde; manyetik teypler, optik okuyucular, diskler, tarayıcılar, chipler vb. gibi donanımlar kullanılmaktadır.

- Merkezi İşlem Birimi: Merkezi işlem birimi, komutların işletilmesini sağlamak üzere kıyaslama, saklama ve karşılaştırma işlemlerini icra eden bir donanımdır. Merkezi işlem birimi; bu işlemleri yürütmek üzere aritmetik ünite (arithmetic-logic unit), iç hafıza ve kontrol ünitelerine (control unit) sahiptir.

- Çıktı Birimleri: Girdi ve Çıktı birimleri olmaksızın, organizasyonlarda, bilgi sistemlerinin kullanıcı komutlarını alması, verileri bilgiye dönüştürmesi ve ilgili birimlere iletmesi olası değildir. Merkezi işlem biriminde çeşitli işleme tabi tutularak çıktılara dönüşen bilgiler, çıktı birimleri ile kullanıcıların hizmetine sunulmaktadır. Çıktı birimlerine örnek olarak mikro filmler, manyetik ve optik diskler, yazıcı ve çiziciler verilebilir.

- İnsan Kaynakları: Organizasyonlarda genellikle, bilişim sistemleri personeli olarak, sistemin tasarımından uygulama alanına gelinceye kadar yapılması gereken işleri yürütmekle görevli sistem analisti, sistem mühendisi, sistem tasarımcısı, veri tabanı yöneticisi, programcı ve operatör kadroları bulunmaktadır.

- Prosedürler: Bilişim sistemi içinde yer alan, girdi ve çıktı birimleri ile merkezi işlem birimi arasında işbirliği sağlayan çeşitli işletim ve kullanıcı yönergeleri, sistemin prosedürleri olarak hazırlanmaktadır. Prosedüre ilişkin değişkenler ve prosedürlerin içerdiği parametreler, bilişim sistemleri kullanımı ve yönetimi açısından önemli sayılmaktadır.

- Yazılım: Bilişim Sisteminin çalışma yöntemini ve görevlerini belirleyen, donanımı destekleyen programlar setidir. Bilişim Sisteminden beklenen hedefleri gerçekleştirmek üzere girdi ve çıktı birimleri ile merkezi işlem biriminin faaliyetlerini kontrol etmek üzere hazırlanan programlar ve komutlar dizisi "yazılım" olarak adlandırılmaktadır.

2.2. TEMEL BİLİŞİM SİSTEMLERİ

Bilişim sistemleri (BS), işletme organizasyonlarınca üretilen ve yine işletme faaliyet süreçlerinde kullanılan verilerin işlenmesi amacıyla kullanılmakta, içsel (internal) ve dışsal (external) kullanım için çok çeşitli bilgi ürünlerine kaynaklık etmektedir. Bilgi sistemlerinin genellikle, görevler etrafında değil, sonuçlar etrafında örgütlenmesi önerilmektedir (Öğüt, 2003).

Organizasyonların hizmet ve üretim alanlarındaki farklılıklardan dolayı yapılan işin türüne göre farklı nitelikte bilgi gereksinimleri vardır. Organizasyonlar bu bilgi gereksinimlerini kurdukları BS ile karşılamaktadırlar. Genel olarak BS'nin; yönetim seviyelerine, fonksiyonel alanlarına ve destek sağlama biçimlerine göre sınıflandırıldıkları görülmektedir.

Sarıdoğan (2004), Bilişim Sistemlerini kullanım yerleri ve taşıdıkları özelliklere göre şu başlıklar altında sınıflandırmıştır;

- Çevrim içi sistemler,
- Gerçek zamanlı sistemler,
- Karar destek sistemleri,
- Bilgi tabanlı sistemeler,
- Veri tabanı yönetim sistemleri,
- Kişisel Bilgisayar sistemleri,
- Ofis otomasyon sistemleri,
- Atölye otomasyon sistemleri,
- İletişim sistemleri,
- Endüstriyel sistemler,
- Kontrol sistemleri.

Günümüzde, çok sayıda organizasyonun, kaliteyi yükseltmek, maliyeti düşürmek ve gerek üretim gerekse hizmetteki çevrim sürelerini kısaltmak için daha fazla BS'den yararlandığı gözlemlenmektedir. Çağdaş organizasyonlarda yönetime karar desteği sağlama şekillerine göre yararlanılan temel BS Tablo 2.2 gösterildiği gibi altı grupta toplamak mümkündür;

2.2.1. Elektronik Veri İşlem Sistemleri (Electronic Data Processing Systems):

Elektronik Veri İşlem Sistemleri (EVİS), kurumsal faaliyetlere ilişkin rutin verilerin süreçlenmesini sağlayan bilgi sistemleridir. Satışlar, ödemeler, satın alımlar biçiminde sıralanabilen rutin kurumsal faaliyet işlemlerini elektronik veri işlem sistemleri, müşteri, ürün, stok, sipariş gibi kategoriler biçiminde sınıflandırmakta, ilgili verileri toplulaştırmakta, işlemekte ve kullanıma hazırlamaktadır. EVİS, kurumsal faaliyetlerin desteklenmesinde yaşamsal rol oynamaktadır(Öğüt, 2003).

Tablo 2.2: Organizasyonlarda Bilgi Sistemleri Uygulamaları ve Bilgi Kullanımı (Öğüt,2003)

Bilgi Sistemleri	**Bilgi Kullanımı**
Elektronik Veri Sistemleri (Ticari Kayıt Sistemleri)	Organizasyon faaliyetlerine ilişkin rutin veriler
Ofis Otomasyon Sistemleri	Operasyonel düzey için dijital veriler
Yönetim Bilgi Sistemleri	Genel yönetim düzeyi için, planlama, kontrol, karar verme amaçlı bilgiler
Karar Destek Sistemleri	Tepe yöneticileri için kurum-içi ve kurum-dışı grafiksel bilgiler
Üstdüzey Yönetici Bilgi Sistemleri	Etkileşimli destek içeren, analitik bilgiler
Uzman Sistemler	Spesifik konularda uzmanlık önerileri içeren, yapay zeka destekli üst bilgiler

EVİS, Ticari Kayıt Sistemleri (TKS) olarak da adlandırılmaktadır. TKS işletmenin günlük rutin kayıtlarının tutulmasına olanak sağlayan sistemler olup, operasyonel seviyede hizmet verirler. Bu sistemler işletmede yürütülen her türlü ticari faaliyetle ilgili verilerin toplanması, saklanması ve diğer BS'lere aktarılması fonksiyonlarını yerine getirirler. İşletmenin alt kademe çalışanları TKS yardımıyla her türlü ticari faaliyeti veri tabanına kaydederler.

TKS'ler bir örgütün diğer BS'lerinin temelini oluşturmaktadır. Çünkü bu sistemler bir örgütün temel işletme faaliyetleri ile ilgili verilerin elde edilmesini ve işlenmesini desteklemektedir. Bu veriler olmadan siparişlerin verilmesi, müşteri faturalarının düzenlenmesi, çalışanların ücretlerinin ödenmesi çok güç olurdu (Güleş ve Özata, 2005).

EVİS aşağıdaki temel özelliklere sahiptir (Önder, 2005);

- Kayıt ile ilgili işlemlerin elde edilip, kayıtların muhafaza edilmesine yöneliktir.
- Dosya kökenlidir.
- Çıktısı genellikle periyodiktir.
- Öncelikle operasyonel seviye yönetim için bilgi üretir.

- Yöneticinin özel bilgi istekleri için, sınırlı esnekliğe sahiptir.
- Tipik olarak fonksiyona dayalıdır. Uygulamalar birbirinden bağımsız olarak geliştirilir.

2.2.2. Ofis Otomasyon Sistemleri (Office Automation Systems):

Ofis Otomasyon Sistemleri (OOS), bir ofiste çalışanların verinin verimliliğini arttırmak amacıyla tasarlanmış olan kelime işlemci, tablolama ve hesaplama sistemleri, masaüstü yayıncılık, elektronik posta, elektronik takvim, randevu, program ve planlama sistemi gibi unsurlardan oluşan sistemlerdir (Önder, 2005). İşletmede telefon, faks, elektronik mektup, intranet uygulamaları ve video konferans uygulamaları, OOS'nin sayabileceğimiz birkaç unsurunu oluşturmaktadır.

2.2.3. Yönetim Bilişim Sistemleri (Management Information System):

Yönetim Bilgi Sistemleri (YBS), karar alma süreçlerine katkı yapan bilgileri, optimum zaman dilimi içinde, ekonomik ve doğru biçimde yönetime sağlayan, bütünleşik insan-makina sistemleridir. YBS, yapısal ve fonksiyonel (işlevsel) açıdan karmaşıklaşan bilgi çağı organizasyonlarını etkinleştirme ve işlemleri basitleştirme açısından önemli sayılmaktadır. YBS, organizasyonlarda bilgi teknolojileri kullanımının yönetsel boyutuna önem vermesi ve bilgi sistemleri uygulamalarında sistem yaklaşımını benimsemesinden dolayı, günümüz organizasyonlarının kurumsal etkinlik ve hizmet kalitesi düzeylerinin yükseltilmesinde yaşamsal bir faktör olarak değerlendirilmektedir (Öğüt, 2003).

Yönetim Bilişim Sistemi, VİS'den sonraki evredir. İki sistem arasındaki en temel fark, veri işleme sistemindeki kullanıcının yerini; YBS'lerinde yöneticinin almış olmasıdır.

Organizasyonun tüm kademelerindeki yöneticilerin, işlerini etkili olarak yapabilmeleri için ihtiyaç duydukları bilgileri toplayan, analiz eden ve özetleyen sistem YBS'dir. Bu yönüyle üst kademe yöneticilerin strateji ve politikalarını uygulamalarına destek sağlayan araçlardır. Orta kademe yöneticilere; çalışanların performansı, ürünlerin kalitesi ve müşteri memnuniyetinin değerlendirilmesi için bilgi desteği sağlar. Alt

kademe yöneticiler için ise, günlük işlemlerin etkili biçimde yapılmasını sağlayan bilgiler sunar.

Öğüt (2003) günümüzde yöneticilerin, genellikle bilgi yokluğundan (noninformation) değil, bilgi çokluğundan (information overload) kaynaklanan sorunlar yaşamakta olduklarına dikkat çekerek, bu bağlamda yöneticilere gereksiz ve ilintisiz bilgi setleri sunan bilgi sistemlerinin, "management misinformation systems" biçiminde nitelendirildikerini ifade eder. Bir organizasyonun amaçlarına ulaşması, iç ve dış çevredeki değişimlere karşı sağladığı uyum ve ortaya çıkan değişimler karşısında yönetimin gereksinim duyduğu doğru, zamanlı ve anlamlı bilgileri sağlayarak, etkin karar alma faaliyetini gerçekleştirmesiyle yakından ve doğrudan ilgili olduğu düşünülürse, YBS'nin bilgi çağı organizasyonları açısından taşıdığı önem daha iyi anlaşılacaktır.

Geleneksel olarak satış, envanter ve üretim bilgilerinin karar alıcı bireylere hızlı bir şekilde aktarma yolu olan YBS, bütün organizasyon birimleri ile entegre oluncaya kadar genişletilebilmektedir. YBS'nin üst düzey yöneticilerin karar almadaki rollerini yansıtmak için yönetim destek sistemleri kavramı kullanılmaktadır. Organizasyonlar için tasarımlanan; YBS, değişen çevresel koşullara uyum sağlayıcı olmalıdır. Değişime uyum sürecinde, bilgi sistemleri, geri-bildirim, değerlendirme ve veri düzenlemesi yoluyla başarı sağlamaya odaklanmalıdır (Öğüt, 2003).

Yönetim Bilişim Sistemi kavramı daha ayrıntılı olarak Bölüm 2.3.de incelenmiştir.

2.2.4. Karar Destek Sistemleri (Decision Support Systems):

Karar Destek Sistemleri (KDS), karar alma süreçlerine etkileşimli ve fiili destek sağlayan bilgi sistemleridir. KDS, son kullanıcı yöneticiler (end-user managers) için istatistiksel analizler, grafik ve model destekler, veri tabanı ve rapor üretim teknikleri gibi yazılım paketleri sağlamaktadır. Öte yandan, KDS'nin elektronik veri işlem ve ofis otomasyon sistemlerinin geliştirilmiş ileri düzeyi olduğu ileri sürülmektedir. Ancak, karar destek sistemleri, elektronik veri işlem ve ofis otomasyon sistemlerine oranla etkileşimli bir süreci ifade etmektedir (Öğüt, 2003).

KDS, yöneticinin yerine geçerek bizzat kararlar alan sofistike elektronik araçlar olmaktan çok, onun belli bir yargıya kolaylıkla ve etkili olarak varmasını desteklemek ve yöneticinin isabetli karar almadaki etkinliğini artırmak ve alınan kararı iyileştirmek amacıyla geliştirilmektedir.

KDS'yi YBS'den ayıran yapısındaki modelleme, bilgi sistemi ve sorun çözme elemanlarıyla yapısal olmayan sorunların çözümüne destek sağlamasıdır. Bu yönüyle KDS, spesifik karar süreçlerine destek sağlamak amacıyla bilgi üretimine odaklanan bilgi sistemleridir. KDS; birey/makine etkileşimine imkan sağlayarak, nitelikli kararlara kaynaklık eden sinerjik karar sürecine katkı yapmaktadır.

Karar destek sistemlerinin temel özellikleri şunlardır:

- Kullanıcılara esneklik, uyumluluk ve hızlı cevaplar sunar.
- Kullanıcılara girdi ve çıktıları basma ve kontrol izni verir.
- Profesyonel programcılardan çok az veya hiç yardım almadan çalışır.
- Kararlar ve problemler için destek sağlar.
- Karmaşık analiz ve modelleme araçları kullanır.
- Girdi veri miktarı oldukça fazladır.
- Yöneylem araştırması, maksimum kar, minimum maliyet, genel anlamda kantitatif teknikler ile yapılan işlemlerdir.
- Çıktı olarak analizler ve duyarlılıklar verir.
- Orta ve üst yönetime karar aşamasında destek sağlar.
- Karar alma sürecinin tüm aşamalarında kullanılabilir.
- Bireysel ve grup tabanlı karar verme desteği sağlar.
- Yapısal olmayan veya yarı yapısal kararlar için kullanılır.
- Değişen koşullara ve karar durumlarına uyabilecek esnekliktedir.
- Karar vericinin yerine geçmekten ziyade, ona karar vermesinde yardımcı olur.

2.2.5. Üst Yönetim Bilgi Sistemleri (Executive Information Systems):

En temel görev ve sorumlulukları, uzun dönemli kurumsal planları belirlemek ve stratejik kararlar almak olan üst düzey yöneticilerin gereksinimlerine uygun BS tasarımı ve geliştirilmesi YBS literatüründe önemsenen bir tartışmadır. Üst Düzey Yönetici Bilgi Sistemleri (ÜYBS), organizasyon tepe yönetiminin gereksinim duyduğu kritik bilgileri,

uygun zaman dilimi içinde, istenilen biçimde, kurum içinden ya da dışından sağlayan bilgi sistemleridir. ÜYBS, üst düzey yöneticilerin yapısal nitelikte olmayan kararlarına, ileri grafik ve iletişim teknolojileri aracılığıyla stratejik destek sağlamak amacıyla geliştirilmiştir (Öğüt, 2003).

ÜYBS'nin özellikleri özet olarak şöyle ifade edilebilir (Önder, 2005):

- Grafikseldir.
- Kullanımı kolaylaştıran arayüzlere sahiptir.
- Geniş fakat özet bilgi sağlar.
- Tercihe bağlı olarak detaylı seviyeye doğru genişler.
- Birçok veri kaynaklarını bütünleştirir.

2.2.6. Uzman Sistemler (Expert Systems):

Günümüzde bilgisayarlar hesaplama, konuşma, hatırlama, sayıları mukayese etme ve çizim yapma gibi birçok insan yeteneklerini taklit edebilmektedirler. Araştırmacılar, bu yetenekleri ve akıllı insan davranışlarını taklit edebilen yazılım ve donanım geliştirerek bilgisayarların gücünü arttırmak üzere çalışmaktadır. Yani esas ilgi, anlama, sonuç çıkarma, öğrenme, bilgi toplama, kendi kendini iyileştirmek için gayret sarf etme, insanın duyumsal ve mekanik yeteneklerini taklit etme özelliğine sahip sistemler üzerindedir. Bu genel araştırma alanı, "Yapay Zeka" (YZ) olarak bilinmektedir (Önder, 2005).

Yapay Zeka terimi ilk kez 1956 yılında matematikçi iki genç profesör, Marvin Minsky ve John McCarty tarafından Dortmund konferansından sonra ortaya atılmıştır. Sayısız uygulama sahası olan YZ kavramı ile ilgili olarak yapılan birçok tanımlama mevcuttur. YZ, bilgisayar biliminin akıllı, yani dili kullanabilme, öğrenme, akıl yürütme, problem çözme gibi niteliklere sahip bilgisayar sistemleri tasarımlamakla uğraşan koludur. YZ, insanın düşünce yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmaktadır. Uzman Sistem (US), bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılmış sistemlerdir (Önder, 2005).

Öğüt (2003), üstbilgi tabanlı sistemlerden olan ve YZ teknikleri kullanımı içeren US'i kullanıcılara belirli uzmanlık konularında danışmanlık desteği sağlayan, pratik sorunlara çözüm önermek amacıyla kullanılan sistemler olarak tanımlar.

Ş. Kavuncubaşı (2000) Uzman Sistemlerin, genellikle özel bilgi ve deneyim gerektiren iyi tanımlanamayan görevlerin yerine getirilmesini sağlayan bilgisayar uygulamalarını içerdiğini ve belirsizlik ortamında, iyi tanımlanmamış sorunlara tanı koyma ve sorunu çözmek için model tasarlama konularında uzmanların entelektüel çalışmalarına katkıda bulunduğunu belirtir. (Köse, 2004)

Long (1989)'a göre bir US; sorulara cevap veren, açıklama için sorular soran, öneriler yapan ve genellikle karar verme sürecine yardım eden bilgisayara dayalı etkileşimli bir sistemdir. Aslında US'ler insanın düşünme sürecini simüle ettiğini, bir ya da birden fazla uzmanın bilgilerinin, bir US'in bilgi tabanına girildiğinde, kullanıcılar, insan uzmanın etkileştiği gibi aynı şekilde sistemle etkileşerek bu bilgiyi çıkarabilirler. Bir US'in amacı uzmanlığı önce uzmandan bilgisayar sistemine, daha sonra da uzman olmayan diğer insanlara aktarmaktır. Bu işlem bilgi kazanma (uzmanlar ve diğer kaynaklardan), bilginin temsili (bilgisayarlarda), bilginin çıkarımı ve bilginin diğer kullanıcılara aktarımı gibi dört faaliyeti içerir. US'in en önemli özelliği, muhakeme kabiliyetidir. Program bilgi tabanında depolanmış olan uzmanlığa ulaşabilir ve çıkarımlar yapabilir (Köse, 2004).

Uzman Sistemler, olağan olarak özel uzmanlık bilgisine sahip bireylerce alınan, yapısal olmayan ve yarı yapısal kararların bilgi teknolojileri aracılığıyla gerçekleştirilmesini sağlayan bilgisayar yazılımlarıdır. US, organizasyonlarda 1980'Ierden itibaren kullanılmaya başlanmakla birlikte, uzman sistemlerin işletmeler dünyasında çok geniş potansiyel kullanım imkanı olduğu ileri sürülmektedir (Öğüt, 2003).

En genel anlamıyla US, YZ'yı kullanarak insanı taklit eden sofistike sistemlerin geliştirilmesidir. Bilgisayarlar, insan hareketlerini taklit eden makineleri kullanabilir ve belirli bir uzmanlık sahasındaki insanın düşünme sürecini taklit ederek kendi kendine öğrenerek, kararlar verebilir.

2.3. YÖNETİM BİLİŞİM SİSTEMİ

Yönetim Bilişim Sistemi (YBS) kavramının yapılmış ve benimsenmiş evrensel tek bir tanımı yoktur. Bu alanda çalışan her kişi ayrı bir tanım getirebilmektedir.

Dilimize "Yönetim Bilgi Sistemi" ya da "Yönetim Bilişim Sistemi" olarak yerleşen, İngilizce «Management Information System» kavramının karşılığıdır.

Kavramı oluşturan bu sözcüğü ayrı ayrı ele aldığımızda, (yönetim, bilgi, sistem) bunların yeni gelişen ve nesnelleşmekte olan üç soyut sözcük olduğu görülür. Böylesi üç sözcükten oluşan bir kavramın da net bir tanımını yapmak güçtür. Ancak yine de bu üç sözcüğün bilinen anlamlarını esas alarak ve uygulama gerçeklerine de dayanarak YBS için şu tanımlar verilmiştir;

Ülgen (1977) YBS'yi özellikle elektronik bilgisayarların yönetim sürecinde kullanılmasıyla ortaya çıkan bir kavram olarak, "bir örgütün yönetiminde kullanılan bilgilerin işlenmesi ve iletilmesini sağlayan sistem" şeklinde tanımlamıştır.

Diğer bir tarife göre YBS, bir örgütteki tüm gerekli bilgi akışını sağlayan, iç ve dış çevreden verileri bütünleştirip, bilgi ve işleme desteği veren, gerekli biçimde sunan, bilgisayar destekli bir sistemdir (Gümüştekin, 2004).

Reynolds (1988) örgütlerdeki işleyiş, yönetim ve karar verme süreçlerini desteklemek için gerekli bilgiyi sunmak üzere değişik kaynaklardan alınan verileri bütünleyebilen bir bilgisayar sistemidir şeklinde tanımlamıştır (Çimen, 1994).

Yönetici bir karar alırken, kararını destekleyecek bilgilere gereksinim duyar. Gereksiz bilgi ile karar alma zorlaşır ve giderek olanaksızlaşır. Bunu önlemek için, bilgi ve veri akışının iyi düzenlenmesi gerekir. Bu aşamada geliştirilen sistem, YBS olarak adlandırılır. Bu sistem yardımıyla, her düzeydeki yönetici, gerektiği kadar ve yalnızca kendini ilgilendiren bilgiyle donatılır. Yazışmalarla kaybolacak zamanı kazanmak için de, bilgisayarlardan yararlanılır (Gümüştekin, 2004).

Öte yandan Ülgen (1977), çeşitli yazarların YBS sistemi kavramını farklı açılardan, çeşitli şekillerde tanımladıklarını, bazılarının bu kavramla eş anlamda kullanılan bir çok yeni deyim ve kavramlar ortaya atıklarını aktarmaktadır. "Örneğin, T.B. Glans ve arkadaşları; yönetim sistemlerinin (management systems), işletmenin faaliyetlerinin önemli bir yönünü oluşturduğu için genel olarak iş sistemleri adıyla anıldığından ve toplam sistem (total system), birleştirilmiş sistem (unified system), hedefe yönelik sistem (goal directed system), bütünleşik sistem (integrated system) ve bilişim sistemi (information system) kavramlarının da eş anlamda kullanıldığından söz etmektedir. R. L. Ackoff ise bilgisayar uygulamalı yönetim sistemlerini basitten karmaşığa doğru şu şekilde sıralamaktadır: Bilgi işlem sitemi (Data processing system), bilişim sistemi (information system), yönetim bilişim sistemi (management information system), yönetim karar sistemi (management decision system), yönetim karar ve destek sistemi (management decision and support system), sorun tanıma sistemi (problem identification system) ve yönetim sistemi (management system). Ancak bu kavramlar arasında en yaygın olarak kullanılanlar; bilgi işlem sistemi, yönetim bilişim sistemi ve yönetim sistemleri kavramlarıdır."

Gümüştekin'e (2004) göre YBS;

- Veri toplama, kaydetme ve işleme işlevlerini yerine getirir.
- Yöneticilere karar almak için ham veri değil, seçilmiş, işlenmiş ve düzenlenmiş veri başka deyişle bilgi sağlar.
- Değişik kaynaklardan verileri, bütünleşik veri tabanında toplar.
- Bilgiyi toplayıp aktaran bilgisayar destekli bir sistemdir.
- Yöneticilerinin zamanında ve kolayca alabilecekleri yapısal bilgi sağlar.
- Örgütün değişen bilgi gereksinimlerini karşılar.
- YBS, yalnızca yetkili işgörenlerin bilgi alımıyla sınırlandırılmış, güvenlik sistemiyle sarılıdır.
- Örgüt içinden ve dışından geçmiş, bugün ve geleceğe yönelik bilgi sağlar.
- Bilgisayar, donanım (hardware) ve yazılım (software), incelemeler, planlama, karar alma, denetleme, veri tabanı vb. öğeleri kullanır.

Plunkett ve Attner (1992) "Introduction to Management" isimli eserlerinde YBS'ni, yalnızca karar almak için bilgi sağlayan değil, aynı zamanda örgütün değişik

parçalarının birbirine bağlanmasına yardım eden, örgütte bilgi akışını gerçekleştiren sistemler dizisi olarak tanımlar. Muhasebeden, ücretlerden, yükleme ve teslimden, satışlardan, karar almak için tüm düzeylerden genel müdüre akan özet raporlar, stok sistemleri, bütçeler ve finansal raporlar gibi temel olan tüm denetlemeler, örgütte YBS'nin öğeleridir (Gümüştekin, 2004).

Evren ve Karpuz'a (1988) göre; yöneticinin dünyası, şimdiki ile gelecek zaman arasında bir değişim ve belirsizlik içindedir. Karar sürecindeki bu belirsizliği azaltmak için, örgütün iç ve dış çevredeki etkinliklerine ilişkin (uygun, anlamlı, doğru, zamanlı, uygun biçimde, incelenebilir, belirli bir içeriği olan ve eksiksiz) bilgilere gereksinimi vardır. Elde edilen bilgiler, işletmenin yönetim düzeylerini birbiri ile ilişkilendiren bir bilgi ağında, düzeylerin uyumlaştırılması ile bütünleşik bir yapıda biçimlenmelidir (Gümüştekin, 2004).

Tarondeau (1975) YBS'ni, yöneticilere karar vermeleri için yararlı bilgileri sağlamak amacıyla işletme-içi ve dışı verileri toplayıp organize edebilen iletişim (haberleşme) araçlarının aralarındaki bütünleşik ilişki olarak tanımlarken YBS'nin temel amacının, iyi kararlar alabilmek için örgütün olanaklarını arttırmak olduğunu öne sürmektedir. Bu özelliğiyle YBS, örgütün içinden ve çevreden gelen verilerin karar süreci için yararlı bir biçime dönüştürüldüğü, dinamik bir değişim sürecidir (Ülgen, 1977).

Gümüştekin'e (2004) göre YBS, yönetimin gereksinimlerini karşılamaya yönelik olmalıdır. Bu YBS'nin en önemli özelliğidir. Sistem, üst düzeyden alt düzeye doğru ve yönetim gereksinimleri doğrultusunda, işletmenin amaçları göz önüne alınarak geliştirilir. Orta düzey yönetimin ya da alt düzey yönetimin gereksinimleri, sistemin temelini oluşturur. Dolayısıyla, YBS'nin yönetim tarafından yönlendirilmesi zorunludur.

Bir sistemin başarıya ulaşmasındaki temel etken insandır. İyi tasarlanmış bir YBS ile etkinlik sağlayabilmek için, gerek üst düzey yönetimin bu konuya yeterli desteği sağlaması, gerekse alt düzey yönetimin bu konuya inanmış olması gerekir. Aksi taktirde sistem sürdürülebilir başarılı bir sistem olmaktan uzaklaşacaktır.

Head (1972) YBS kavramı ile ilgili olarak yapılan tanımlamaları üç ana grupta toplamaktadır (Ülgen, 1977);

a. YBS'ni bir donanım düzeni olarak gören tanımlar: Bu görüşe göre YBS, yöneticilere terminaller ve türlü çevre birimleriyle bağlanmış bir bilgisayar sistemi ya da bir merkezi işlem birimi ve çevre birimleri ile bunlar arasındaki bağlantı şebekesi olarak tanımlanmaktadır. Örneğin Kelly (1970), YBS'nin gelişmesine değinirken konuyu bilgisayar donamının gelişmesiyle sınırlı olarak işlemektedir.

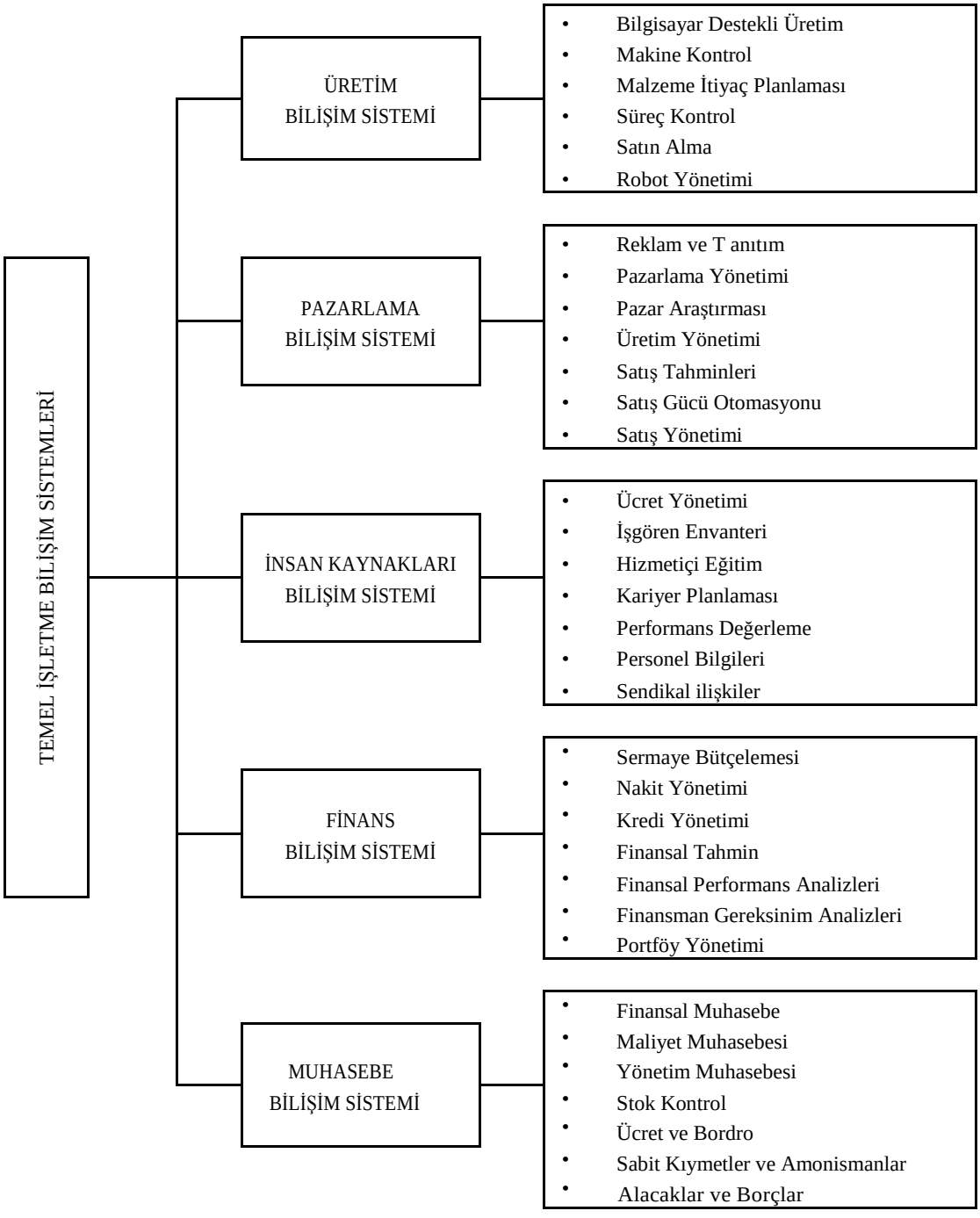
b. YBS'ni bir yazılım sistemi olarak gören tanımlar: Bu görüşe göre ise YBS bir bilgisayarı bilgileri toplayacak, işleyecek, saklayacak ve iletecek biçimde yönetecek bilgisayar programları ve kullanım yöntemleri topluluğudur.

c. YBS'ni, işletme yöneticilerine gerekli bilgileri vererek, yönetsel kararları destekleyen bir sistem olarak gören tanımlar: Bu görüş açısından YBS, gerektiği anda gerekli bilgiyi vererek yönetim kararlarını destekleyen sistemdir. Nitekim bir çok yazar YBS kavramını bu açıdan ele alarak tanımlamaktadır.

Sanders (1975), YBS'ni, bir işletmede yöneticileri zamanlı ve etkin bilgi ile donatabilmek amacıyla geliştirilen, bilgisayar temeline dayanan bütünleşik bilgi işlem yöntemlerinin tümü olarak tanımlamaktadır (Ülgen, 1977).

Örgütler büyüdükçe bilgi iletişim sorunları da artmaktadır. Bilgiler, işletme içinde bilgi akışını sağlayan bir iletişim şebekesi aracılığıyla bir yerden diğerine aktarılır. Bu bilgi akışında, gerekli bilgilerin hangi yöntemlerle seçileceği, kaydedileceği ve etkin kararlar alması için yöneticiye nasıl besleneceği, tepe yönetiminin sorumluluğunda olup, etkin bir YBS'nin varlığını gerektirir. Etkin bir YBS, verileri ya da bilgileri mümkün olduğu kadar çıkış noktalarından toplar, daha sonra bunları düzenleneceği, hesaplanacağı, özetleneceği ve karar organları için yararlı bir biçime sokulacağı bilgi işlem merkezlerine gönderir. Özet olarak YBS örgütün yaşama ve gelişmesinin sağlanması ile örgütsel faaliyetlerin planlanması, yürütülmesi, organizasyonu, koordinasyonu ve kontrolü için, yönetimin gereksinme duyduğu doğru, zamanlı ve anlamlı bilgiyi sağlayan ve geliştiren sistemdir.

Hoşcan (2003) YBS adlı eserinde temel işletme bilişim sistemlerini ve alt sistemlerini Şekil 2.3. olduğu gibi 5 ana grupta toplamıştır (Önder, 2005).



Şekil 2.3. Temel İşletme Bilişim Sistemleri ve Alt Sistemleri

2.3.1. YBS Geliştirme Çalışmasında İzlenecek Adımlar:

Uçkan ve Fişek (1974) bir YBS tasarımında ve uygulamaya geçişte genellikle şu faaliyetlerin yürütüldüğünden söz etmektedirler (Ülgen, 1977);

- Mevcut durum tespiti,
- Sistem çözümleme ve ilk (geçici) tasarım,
- İlk tasarım kontrolü ve nihai tasarım,
- Tasarıma göre sistem ihtiyaçlarının saptanması,
- Uygulamaya geçiş hazırlıkları,
- Pilot uygulama, sistem değerlendirme ve düzeltme,
- Uygulama ve uygulama sırasında müdahale ve düzeltmeler.

Bocchino bir YBS çalışmasında izlenecek adımları on başlık altında toplamaktadır (Ülgen, 1977).

a. *Sistem çalışma grubunun örgütlenmesi*: Sistemi düzenleyecek olan çalışma grubu çok geniş olmamalı, ancak örgütün temel fonksiyonel bölümlerinin temsilcilerini kapsamalıdır. Çalışmanın önemli zaman gerektirmesi nedeniyle, kilit noktalarındaki yöneticilerin yerlerine astlarını görevlendirmeleri mümkündür. Tepe yönetimi ise ayrıca yönlendirici bir komite olarak etkin olacaktır. Bu arada sık-sık ya da sürekli olarak, işletme dışından yabancı bir uzmanın danışmanlığından yararlanılabilir. Çalışma grubu önce çalışmanın programı ve genel kaidelerini belirleyecek, görev dağıtımı yapacak daha sonra da proje ve alt projeleri tanımlayıp, boyutlandıracak ve ilgili organların incelemesine sunacaktır. Çalışma grubunun faaliyetleri, grup lideri tarafından belirli zamanlarda düzenlenen raporlarla tepe yönetimine bildirilecek, arzulanan değişme ve yeniliklerin yürütmeye konması istenecektir. Öte yandan gerektiğinde kimi yürütme organlarının görüş ve önerileri sorulacak, oturumlara katılmaları beklenecektir.

b. *Çalışma grubunun ve örgütün amaçlarının belirlenmesi:* Kurumun amaçlarının ayrıntılarıyla belirtilmesi ve bu amaçla yapılan görüşmeler koordinasyona yardımcı olacaktır. Nitekim, örgütün uzun dönem amaçları belirlendiğinde, buna bağlı olarak kısa dönem hedefleri de bu amaçlara ulaşmak üzere geliştirilebilecek ve YBS gerekleri

içinde çalışma grubu tarafından yorumlanabilecektir. Bu arada örgütün amaçları, YBS çalışma grubu amaçlarına çevrilecek, bu ise örgütün bilgi ihtiyaçlarının, yürütmenin örgütsel amaçlara ulaşmak için gereksinme duyduğu biçimde değerlendirilmesiyle gerçekleşecektir.

c. *Kurumun bilgi gereklerinin saptanması* : Bu evrede kurumun şimdi, yakın gelecekte ve uzun dönemde gereksinme duyacağı bilgiler incelenecek, tüm bilgi kayıt, işlem, iletme ve saklama faaliyetleri gözden geçirilecektir. Bu amaçla yönetimin karar sürecinde ihtiyacı olan bilgiler ve kullanılan tüm basılı belgeler derlenecek, kurum içinden çevreye ve çevreden kuruma bilgi giriş/çıkışı düzenlenecek, mevcut BS'nin maliyeti ve etkinliği açısından verilerin değerlenmesi ile geçerli ya da kullanılabilir sistemlerin muhtemel maliyetlerinin tahmini yapılacak ve nihayet alt sistemler düzeyinde kurumun bilgi ihtiyaçlarının genel özellikleri belirlenecektir. Kurumun bilgi gereklerinin saptanması, kurumsal faaliyetlerin ve bu faaliyetlerdeki kritik karar noktalarının anlaşılması temeline dayanmalıdır.

d. *Çalışma hedeflerinin ve programının düzenlenmesi:* Bu evre YBS çalışma gurubunun ilk uygulaması olup, tüm YBS'ni etkileyen bir ön çalışmadır. Burada çalışmanın yapılacağı uygulama alanlarının sırası yanında, sistemin tasarımı ve uygulamasında izlenecek adımlar da belirlenecektir. Çalışma grubu bu evrede, karmaşık projelerin yürütülmesinde kullanılan PERT (Project Evaluation and Review Technique) tekniğinden yararlanabilir. Çalışma hedefleri, bilgi kaynakları, raporlar ve yönetsel gereksinmeler dikkate alınarak; çalışma programı ise mevcut sistemin çözümlenmesine ve her zaman oluşan karışıklıklar ve beklenmeyen durumlara yeteri kadar zaman ayırarak düzenlenmelidir. Genel hedef ve programlar tasarım ve çalışma ilerledikçe ayrıntılara indirgenecektir.

e. *Mevcut bilişim sürelerinin çözümlenmesi* : Bu evrede kurum içinde mevcut bilişim sistemi ve alt sistemlerinin bilgi akış şemaları, grafikleri çizilir. Diğer bir deyimle bilgi akışının başlangıcından işlenmesi, kullanımı ve saklanması ya da yok edilmesine kadar ayrıntıları belirlenir ve çözümlenir. Bu amaçla bilgi akışında kullanılan form ve raporlar listelenir, bunların kapsamı ve biçimi değerlenir, düzenlenme frekansı, sayısı, karakter türü (alfabetik-sayısal), muamele sayısı v.b. unsurları belirlenir, ham bilginin ya da verilerin bilgiye dönüştürülmesinde yapılan işlemler tanımlanır.

f. *Yeni düzenlenen sistemin işletim niteliklerini geliştirmek:* Yeni YBS bir takım ihtiyaçları karşılamak ve belirli zorlamaları tatmin etmek amacıyla geliştirilmiştir. İşletim nitelikleri bu ihtiyaçları ve zorlamaları kapsayacak ve aynı zamanda ulaşılması zorunlu olan ayrıntılı bilişim sisteminin amaçlarını tanımlayacaktır. Çalışma grubu bu amaçla; kurum içindeki faaliyetlere ilişkin, yönetim-denetim bilgi akışını etkileyen unsurları geliştirecek, yönetime doğru ve zamanlı bilgi sağlayan bir sistemi kurarken kararların uygulama noktalarına hızla iletilmesini sağlayacaktır.

g. *Geliştirilen yeni bilişim sisteminin tasarımı:* Çalışma grubu bu evrede ise; bilgi işlem süreçlerini tanımlamak için sistem akış diyagramları, işlem akış diyagramları, v.b. kullanacak; girdi, çıktı, dosya gereksinmeleri, işlem süreçleri ve sırası ile gerekli bilgisayar programları ve insan-makina ilişkilerini kapsayan belgeleri tanımlayacak; YBS'ni oluşturan yöntem ve işlemlerin sürdürülebilirliğini, uygunluğunu ve genişletilmesini sağlayan prosedürlerin taslağını çıkartacak; yeni YBS için gerekli belgelerin tasarımını planlayıp denetleyecek; tüm performans standartlarını, kantitatif ölçülerle ifade edecek; YBS kurulduğunda geçerli olan örgütsel yapıyı denetleyecek ve nihayet sistemdeki tüm verilerin elde edilme, birbirlerini etkileme ve işlenme zamanlarını tanımlayacaktır.

h. *Yeni YBS açısından bilgi işleme donatımını seçmek ve değerlemek:* Bazı hallerde bilgi işleme aracı gerekmeyip elle işlem yeterli olabilir. Ancak bu kararın türlü bilgi işlem yöntem, prosedür, form ve programlarının ayrıntılı bir biçimde hazırlanmasından önce verilmesi gerekecektir. Zira kullanılan bilgi işlem araçlarının özellikleri, söz konusu yöntem, prosedür form ve programları etkileyebilir. Bu evrede, artık çalışma grubu sistemin ihtiyaçlarını, sıkıntılarını ve yapımcılarla görüşeceği amaçlarını yeteri derecede bilmektedir. Bu nedenle yapımcıların bilgi işlem araçlarını satın alma ya da kiralama koşulları, bakım-onarım hizmetleri, teknik destekleri, yerleşme durumu v.b. hususlarda ayrıntılı bilgi toplayarak, sistem için gerekli olan donanımı seçerler.

i. *Bilişim sisteminin işletim prosedürlerini ayrıntılı hale getirmek* : Çalışma grubu bu evrede, geliştirilen yeni bilişim sisteminin tasarımını (7. evre) kullanılacak olan bilgi işlem araçlarının işletim karakteristiklerine göre inceleyecek, her bilgisayar uygulaması için program akış diyagramları hazırlanacak; ana program akış diyagramları, alt program akış diyagramlarına indirgenecek; programlanan akış diyagramları, kullanılan

programlama dilinin kaideleriyle uyuşacak biçimde bilgisayar komutlarına dönüştürülecek, programlar test edilecek, program tanımlarını, operatör talimatlarım v.b. destekleyici bilgileri kapsayan belgeler hazırlanacak, ayrıntılı form ve prosedürler düzenlenecektir.

j. *Yeni YBS'ne geçişin uygulanması;* Bu sonuncu evrede ise önce yeni YBS kurumun tüm kademelerine açıklayıcı toplantılarla tanıtılacak, bu kademedeki personel eğitilecektir. Kullanılacak olan donanım yerleştirilecek ve gerekli dosya düzenlemeleri yapılacaktır. Bu evre bazı hallerde öncekilerden daha fazla çözümleme ve yeniden tasarım çalışmasına neden olabilir. Ancak bu evrede, çözümleme ve yeniden tasarım dinamik bir çevrenin etkisindedir. Çözümleme ve yeniden tasarım bundan böyle YBS'nin temel bölümlerinden biri olmak zorundadır. Zira denetim bilişim sistemi hizmet ettiği örgütle birlikte büyür ve gelişir.

2.3.2. YBS'nin Uygulama Sorunları

Ackoff (1967), bir çok YBS'nin tasarımında temel alınan hatalı varsayımları beş ortak başlıkta toplamaktadır (Ülgen, 1977);

a. Bir çok yöneticinin sıkıntı çektiği en önemli eksiklik işle ilgili bilginin yetersizliğidir. Ancak çoğu kez yönetici, bilgi eksikliğinden çok, kendisiyle ilgili olmayan bilgilerin çokluğundan şikayetçidir. Şu halde YBS bilgileri sunarken, ilgili olmayanları elemeli yani filtre etmelidir.

b. Yönetici arzuladığı bilgiye gereksinme duyar: Çoğu YBS tasarımcısı hangi bilgilere gereksinme duyulduğunu, yöneticinin kendisine sorarak belirler. Bu yöntem, yöneticilerin gereksinme duydukları bilgileri bildikleri varsayımına dayanır. Gerçekte yöneticinin hangi bilgilere ihtiyacı olduğu hangi kararları verme durumunda olduğuna bağlıdır. Her kararın dayanacağı bilgilerin saptanması ise, o karar sürecinin çözümlenmesini gerektirir.

c. Yöneticiye gereksinme duyduğu bilgi verildiğinde karar vermedeki başarısı artacaktır: Bu varsayım ise, yöneticinin gereksinme duyduğu bilgi ile donatılması halinde, bu bilgilerin etkin biçimde kullanımı bir sorun yaratmaz görüşüne dayanır. Ancak yöneticinin bu bilgiyi en iyi şekilde nasıl kullanabileceği belirtilmelidir. Zira

yöneticiye gereksinme duyduğu bilgi sağlansa bile, karmaşık karar süreçlerinde bu bilgiyi etkin biçimde kullanamayabilir. Bu nedenle yöneticiye bu bilgileri nasıl kullanacağına ilişkin kurallar ya da alınan kararların performansını değerlendiren, hatalarını gösteren bir geri besleme (feed back) düzeni kurulmalıdır.

d. Daha iyi haberleşme başarıyı arttırır: Bu varsayım bir örgütteki yöneticiler arasında iyi bir haberleşme ilişkisinin kurulması örgütün başarısını arttırır ilkesine dayanmaktadır. Bir bakıma, yöneticiler verdikleri kararlardan birbirlerini ve bölümlerini haberdar ederlerse, örgütün daha iyi işlemesi mümkün olabilir. Ancak, Ackoff, yöneticileri verdikleri kararlardan birbirlerini haberdar ettikleri için örgüt amaçlarına ters düşen sonuçlara ulaşıldığını kanıtlayan örnekler vermektedir.

Şu halde, bir örgütün bölümlerini çekişme haline sokacak bir bilgi alış-verişi, örgütsel performansı geliştirecek yerde, zarara sokacaktır.

e. Bir yöneticinin bilişim sisteminin işleyişini bilmesi gerekmez sadece sistemi nasıl kullanacağını bilmesi yeterlidir. Çoğu YBS tasarımcısı, bu yaklaşımın ışığında, kurduğu sistemin yönetici tarafından mümkün olduğu kadar anlaşılmaz kalmasına gayret eder. Tasarımcı sistemin nasıl kullanılacağını yöneticiye basit olarak anlatırken, daha fazla bilgiye ihtiyaçları olmadığını söylerler. Böylece tasarımcı yöneticinin bu konuda bilgisiz kalmasını sağlayacak, yöneticinin sistemi bütünüyle değerleyebilmesini engeller, yöneticiyi bilgisizliğini açıklamak korkusuyla, sistemi anlamaya çalışmaktan alıkoyarlar. Sonuç olarak yöneticinin denetiminde olması gereken sistem, tasarımcı ve programcıların yönetiminde, yöneticiyi denetimine almış olacaktır.

YBS'lerinin başarısını etkileyen bu varsayımlar yanında; tasarım ve uygulamanın başarısızlığını etkileyen nedenleri, Doğrusöz ve diğerleri Teichroew'den (1974) tercüme ederek aynen şu şekilde sıralamaktadırlar (Ülgen, 1977):

a. YBS'ni kullanacak olan yöneticilerin tasarım ve geliştirme işine katılmaması,

b. Sistemin evrimsel bir gelişmeye tabi tutulması gerektiği ilkesinden habersiz olunması veya ihmal edilmesi,

c. YBS konusundaki bilgilerin yetersiz olduğu, henüz sağlam bir teorisinin bulunmaması,
d. Sistem donanımının iyi seçilmemiş olması,

e. Sistemi kurma çabalarının, yönetici tarafından, yeterli dikkatle yönetilmemesi.

Birrien ise, YBS uygulamasında en önemli sorunun bütünleşik bir sistemin gerçekleştirilememesi olduğu görüşündedir. Bütünleşik bir sistem ise, bir çok alt sistemin birleşmesinden oluşur. Böylece yönetim için gerekli tüm bilgilerin merkezi bir dosyada ya da bilgi bankasında toplanması sağlanacak ve gerektiğinde bu bilgilere kolayca ulaşılacaktır. Ancak bu durum çoğu zaman arzulanan biçimde gerçekleşmemekte, uygulamada teknik ve yönetsel diye gruplayabileceğimiz bir takım sorunlarla karşılaşılmaktadır (Ülgen, 1977);

1. Teknik sorunlar, donanım ve yazılım sorunlarını kapsarlar. Örneğin, erişim zamanı, bellek kapasitesi, programlama zorlukları v.b. hususlar sistemin uygulamasını yavaşlatır. Ancak bu sorunlar görelidir. Nitekim bir yöneticinin emrindeki donanım gerçekte rasyonel bir kullanımını sağlayabilir, ama atıl kalmaktadır. Bazı hallerde ise bir kaç saniyelik erişim zamanını kazanmak için, karmaşık simülasyon modellerini düzenleyip zaman alıcı ve maliyetleri arttıran, astarı yüzünden pahalı çalışmalara girişilebilir.

2. Yönetsel sorunlar ise yönetici ile ilgili sorunlardan oluşur. Gerçekte, arzu edilen yönetim bilişim sisteminin uygulanmasında, aşılması gereken en güç engel yöneticinin kendisidir. Yöneticinin kendisine bağlı olan bu sorunları dört ana gruba ayırabiliriz (Ülgen, 1977);

- Yöneticinin bilgi alma – işleme – iletme açısından yetersiz olması,
- Değişmeye karşı durgunluk,
- Geçerli eğitim eksikliği,
- Gerçekleştirilecek görevin boyutlarının bilinmemesi.

2.3.3. YBS Kuruluşunda Dikkate Alınması Gereken Hususlar

Ülgen (1977), K. Kitagawa'dan tercümeyle bir YBS kurarken dikkate alınması gereken hususları şu dört maddede sıralamaktadır:

a. Yöneticiler yönetim ve özellikle politik kararlar için gerekli olan bilgi türlerini açıklıkla belirlemelidir.

b. YBS'ni tüm işletmeye uygulamak bugün için ekonomik olmayabilir. Bu nedenle yönetici alt sistemler arasında en etkin ve ekonomik olanları seçerek uygulamaya koymalı, daha sonra tüm sistemi işletmeye uygulamanın ekonomik yollarını araştırmalıdır.

c. Tasarım ve uygulamada çalışacak uzmanların eğitim ve çalışmaları planlanmalı, bilgisayarın ve sistemin yetenekleri herkesin anlayabileceği biçimde tanıtılmalıdır.

d. Örgütsel yapı, esnek bir şekilde YBS'ne göre düzenlenmelidir. YBS'nin güncelliğini koruyabilmesi, şirket politikaları paralelinde gelişebilmesi için bilişim sorumlusunun üst yönetim içerisinde bizzat yer alması ya da danışman olarak kendisinde faydalanılması yerinde olacaktır.

Bir önemli nokta da hazır yazılım mı alınacağı yoksa işletmeye özel bir sistemin mi geliştirileceğine karar verilmesidir. O'Connell (1991), son yıllarda yazılım endüstrisinin hızlı bir gelişim gösterdiğini, çoğu kurumun, ihtiyaçlarını büyük ölçüde karşılayacak bir yazılım bulmasının çok da zor olmadığını, özellikle, ön muhasebe, genel muhasebe, bordro, personel, imalat, finansal planlama ve bunun gibi diğer bir çok alanda hazırlanan yazılımlar son derece popüler hale geldiğini, üstelik bu yazılımların çoğunun, iyi hazırlanmış dokümantasyona ve teknik desteğe sahip olduklarını, ayrıca yeni sürümler ve güncellemelerle kullanıcıların karşısına çıktıklarını savunmaktadır (Özen, 2002).

"Outsourcing" olarak dışarıdan hizmet alınmasının verimliliği ve sürdürülebilirliği günümüzde tartışılmaya devam eden bir kavramdır. Bu kavram ve yöntem üzerinde durulmasının temel nedeni ise bilginin stratejik önemi ve maliyetlerdir.

2.4. HASTANE YÖNETİM BİLİŞİM SİSTEMİ (HYBS)

Hastaneler toplumun tüm kesimlerine hizmet sunan, önemli sayılabilecek nitelikte işgücünü istihdam eden, sosyal güvenlik, eğitim ve sağlık sigortası gibi başka sosyal sektörlerle çok önemli ve yakın ilişkileri bulunan ve oldukça karmaşık bir örgüt yapısına sahip sosyo-ekonomik kurumlar niteliğindedirler. Aynı şekilde sağlık sektöründeki organizasyonlardan en fazla kaynak tüketeni ve dolayısıyla da en fazla önem taşıyanı hastanelerdir (Güleş ve Özata, 2005).

Sağlık hizmeti sunan bir kuruluş veya herhangi bir işletme, eğer varlıklarını sürdürmek, pazarda pay sahibi olmak ve müşterilerini kaybetmemek istiyorsa, yeni ürünler veya kaliteli hizmet üretecek yöntemler bulmak zorundadır. Unutmamak gerekir ki, yenilik gözden geçirme faaliyeti değildir. Yenilik bilinmeyen köklü değişimlerle ilgilidir ve yenilik çalışmaları bir plan dahilinde yapılmalıdır. Yeniliğin diğer bir amacı da verimliliği arttırmaktadır. Ancak çalışanların endişe1erini ortadan kaldıracak açıklamaların yapılmadığı hallerde, yenilikten kaçınmak doğru bir davranış biçimi olacaktır (Malkoçoğlu, 1995).

Hastanelerde sunulan sağlık hizmetinin en iyi şekilde verilebilmesi, gelir ve giderlerin izlenebilmesi, gelir kaçaklarının önlenmesi, kaynakların doğru yönlendirilmesi, hastane yönetimine stratejik kararlar vermek üzere bilgi sağlanması amaçlarıyla hastanelerde bilişim sistemlerinin kurulmasına ihtiyaç vardır (Köksal, 2002). Bu arada unutmamak gerekir ki HYBS ile gelecek teknolojik yeniliğin verimli bir şekilde uygulanması, iyi eğitilmiş ve sisteme inanan bir ekiple uygulanması başarıyı getirecektir, aksi ise zaman kaybı, mali yük ve verimsizlikle sonuçlanabilecektir.

Hastane etkinliğinin artırılmasında da HYBS stratejik bir rol oynamaktadır. Bu işlevini; iletişimi geliştirerek, yapılan işleri otomatikleştirerek ve işlerin yapılış şeklinde dönüşüm sağlayarak gerçekleştirmektedir. HYBS manuel yöntemle yapılan işlerin ve prosedürlerin otomatikleştirilmesi için kullanıldığında (örneğin finansal yönetim, klinik ve diğer yönetim işlemlerinde) işlerin yapılma süresi kısalmakta ve maliyetler azalmaktadır. Ayrıca toplanan veriler, ihtiyaç duyulan noktalara eksiksiz ve hızlı

biçimde iletilmekte, böylece bilgiye ulaşmak için harcanan zaman kısalarak performans artmaktadır (Güleş ve Özata, 2005).

Yeni yöntemler ve ekipmanların çıktılar üzerinde etkisinin verimliliği arttırması, diğer işletmeleri olduğu kadar, hastaneler ve sağlık kuruluşlarını yenilik yapmaya itmektedir. Teknolojik değişim ve yenilik bir yandan değişimin orijinal ve yararlı olması gibi iki kuram üzerinde odaklaşırken, diğer yandan bir değişim süreci olarak kabul edildiğinden uygulamaları belirli plan ve programla olması gerekmektedir (Malkoçoğlu, 1995).

HYBS bir yönüyle de hastane işletmelerinin çeşitli düzeylerindeki karar alıcılarına yardımcı olmak amacıyla bilgi toplama ve bilgi yayma fonksiyonlarını üstlenen değişik kaynaklardan elde edilen verileri bütünleyebilen sistemdir. HBYS; güvenlik ve bilgi erişimi, hasta kayıt-kabul ve kartoteks sistemi, hasta evrak/statü takibi, poliklinik, klinik, tıbbi kayıt, radyoloji, eczane, laboratuar, ameliyathane, doğum odası, acil servis, diyet planlaması, randevu takip, döner sermaye ve muhasebe, personel, bordro, ayniyat, demirbaş takibi, stok kontrol, satın alma, ihale dosyaları, v.b. alanlarında kullanılabilmektedir.

HYBS'nin işlevi; ait olduğu kuruluşun bilgi taleplerini doğru, zamanında, tam ve uygun biçimde karşılamaktır. Bir hastanenin günlük işlemleri olarak;

- Hastaya ilişkin başvuru, kabul, sağlık durumu, sevk ve taburcu kayıtları.
- Tanı ve tedaviye ilişkin hastaya uygulanan işlemler ve klinik, laboratuvar, radyoloji, ameliyathane, eczane, terapi, diyet etkinlikleri.
- Genel yönetime ilişkin personel, sabit tesis, cihaz ve malzeme durumları ve yönetimleri
- Mali işlere ilişkin muhasebe, müşteri hesapları, vergilendirme vb. konular sayılabilir.

Hastane bilgi sisteminden beklenen sadece günlük işlemleri kaydetmek ve izlemekle sınırlı kalmayıp, aynı zamanda yönetime, denetim ve planlama işlevlerinde de yardımcı olmasıdır (Çimen, 1994).

Çimen (1994), J. Charles Austin'den (1988) tercümeyle HYBS'ni dört ayrı grupta incelemiştir. Bunlar;

a. Yönetimi desteklemeye ilişkin sistemler,
b. Tanı ve Tedaviyi Desteklemeye ilişkin sistemler,
c. Entegre (Tam Bütünleşik) Sistemler,
d. Modüler HYBS'dir.

2.4.1. Yönetimi Desteklemeye İlişkin Sistemler

Çimen (1994) YBS'leri açısından HYBS'lerini incelediği çalışmasında, ülkemizde 1990'lı yıllarda sadece bazı hastanelerde bilgisayarlardan yararlanıldığını, bunlarında rutin faaliyetlerden öte kullanılamadığını yazmaktadır. Aynı çalışmasında "bu bilgisayar sistemleri ile hastadan alınan ücretlerin faturalandırılmasından, malzeme takibine kadar pek çok işi takip etmede ve hatta günlük işlemlerin takibi yanında bilgisayarların yönetime bilgi sağlamak amacıyla da kullanılmaları gerektiğini" belirtmektedir. Gerçekten de ülkemizde başarılı HYBS'lerinin kurulması 1990'lı yıllarda bilişim sanayisinde yaşanan gelişmeler sonrasında hız kazanmıştır. İlk uygulamalarda genel amaçlı orta ve büyük boy bilgisayarlardan yararlanılmakta iken, kişisel bilgisayarlarda ve ağ teknolojilerinde akıl almaz bir hızla yaşanan gelişmelere paralel olarak uygulamalar da hızla gelişmiştir.

HYBS genellikle; kaynak kullanımı ve programlama sistemlerinden, mali yönetim sistemlerinden malzeme ve tesis yönetim sistemlerinden, personel yönetim sistemlerinden oluşmuş bir bütünleşik (entegre) YBS'dir. Bu alt sistemler sırasıyla kısaca incelenecektir.

2.4.1.1. Kaynak Kullanımı ve Programlama Sistemi

Yüksek bir enflasyon ortamında ihtiyaçların çok büyük parasal yatırım gerektirmesi ve kaynakların gittikçe azalması, etkin kaynak kullanımını gündeme getirmektedir. Bu gün bir çok hastanede dolu yatak oranını, klinik ve acil servis faaliyetlerini, malzeme ve cihaz kullanımını programlamak ve izlemek üzere bilgi sistemleri geliştirilmiştir. Bu suretle etkin bir kaynak dağılımı gerçekleştirilmekte ve hastalara da kolaylık sağlanmış olmaktadır. Özellikle hastanın yatış günü ve ameliyat gününün önceden belirlenmesinde bu sistem çok yararlı olmaktadır. Bilişim sistem yazılımı; daha önce hastaneye kabul

edilmiş hastaların tanı, operasyon ve tedavilerinin sonuçları, hastanın yaşı ve cinsiyeti gibi verilere dayanarak hesaplanan ortalama yatış ve işlem sürelerinin belirlenmesinde ve yeni başvuruların değerlendirilmesinde kullanılabilir. Bu değerlendirme sonucuna göre gelecek günlerin hasta kabul sırası ve yoğunluğu belirlenebilir.

Sistem kuruluş aşamasında, kaynakların kullanımı konusunda ayrıntılı analiz çalışmalarının yapılarak kaynakların tespit edilmesi ve tasarlanan sisteme girilmesi gerekecektir.

Bu surette sistemden elde edilen raporlar hasta kabul personeline hastanede yatan hastaların durumları hakkında ve mevcut boş yataklar ile ilgili bilgi sağlar. Bu sistem sayesinde personel ve servislerin daha ekonomik kullanımı, servisler arası hasta aktarımı ve personel desteği sağlanmış olur.

Polikliniklerde ön kayıt ve randevu sistemi daha da önemli ve gereklidir. Bilgi sistemi, her başvuruda kabul gün ve saatini bildirecek biçimde düzenlenir. Ayrıca analiz yoluyla da kliniklerde hasta tiplerine, haftanın günlerine ve günün saatlerine göre "boş geçen zaman" oranları hesaplanabilecektir. Bu bilgi, hastane yöneticilerine görev dağıtımının programlanmasında yardımcı olacaktır. Kabul edilecek hastaların bir gün önceden otomatik olarak verilmesi, kliniklerin hasta kayıtlarının tutulmasında da kullanılabilecektir.

Kısaca kaynak kullanımı ve programlama sistemi, hastane kaynaklarından en iyi şekilde kullanımını mümkün kılarak, talebi karşılaması, bütçe düzenleme, orta ve uzun dönem planlama konularında hastane yöneticisine yol gösterici olabilecektir.

2.4.1.2. Mali Yönetim Bilişim Sistemi (MYBS)

Maliyetlerin hızla yükseldiği ve mali yönden hesap verme sorumluluğunun ve hesap verilecek makamların hızla arttığı talep çağında hastane yöneticileri şiddetli bir biçimde güvenilir mali yönetim sisteminin gerekliliğini hissetmektedirler.

Genellikle hastanelerde bilgisayar kullanımı ilk olarak MYBS alanında olmuştur. Sağlık hizmetleri alanında en önemli sorununun, bu gün olduğu gibi geçmişte de gelir-gider

dengesinin sağlanarak sağlam bir yapı içerisinde büyümek olduğu bilinmektedir. MYBS, hastanelerde verimlilik performansının ne olduğunu gösterme açısından yöneticilerin vazgeçemeyecekleri bir konudur.

MYBS, bir örgütün kaynaklarının oluşumunu, bu kaynakların kullanılma biçimini,örgütün işlemleri sonucunda bu kaynaklarda meydana gelen artış ve azalışları ve örgütün finansal açıdan durumunu açıklayan bilgileri üreten ve bunları ilgili kişi ve kuruluşlara ileten sistemlerdir.

Hastane muhasebe sistemi ise; hastane işletmesi hakkında parasal yönden bilgi sağlamak üzere veriler toplar, bunları birleştirir, analiz eder ve raporlar hazırlayarak yöneticilerin (Hastane müdürü, Başhekim, Yönetim kurulu) mali konular da bilgilendirilmesini sağlar. Bir hastane bilgi sistemi içinde, sistem analisti mevcut muhasebe işlemlerini bilgisayara uyarlarken muhasebeci ve ilgili memurlar ile çalışmalıdır. Çünkü ortak bir sistemin iyi bir şekilde kurulması ancak düzenli bir işbirliği ile mümkündür. Tahsil olunacak alacakların belirlenmesi için hastanenin mali işlemleri bir bütün olarak ele alınmalıdır. Mali yönetim bilgi işlem süreci hasta kabulünden başlayıp yapılan her türlü harcamanın listelenmesi ve toplam miktarın tahsil edilmesine kadar devam etmelidir.

Bağımsız mali alt sistem aşağıdaki konuları içerir;

- Bordro hazırlama ve hesap çıkarma (Personel veri sisteminden gerekli bilgiler alınır),
- Yapılacak ödemelerin hesabı (Satın alma birimi ile koordine edilir),
- Faturalandırma ve Tahsilat takibi,
- Ana defter (Defter-i kebir) hesabı,
- Genel giderlerin paylaştırılması için gider dağıtım sistemi,
- İlgili kişi ve kuruluşlara sunulacak mali raporları hazırlama,
- Bütçe düzenleme ve bütçe denetimi.

2.4.1.3. Malzeme ve Tesis Yönetim Bilişim Sistemi

Tesisler ve malzemelerin daha verimli yönetiminde yöneticilere bilgi sunmayı hedefler.

Bakım onarım ve takibi, peryodik bakımların planlanması, enerji yönetimi, tesis bakımlarını izleme, projeleri programlama ve denetim konuları bu kapsam da yer alan konuların bazılarıdır.

Hastane yönetiminde bilgisayar destekli malzeme YBS'ne geçilmesinin en önemli sebeplerinden biri, hastaneye girişi yapılan ve kullanılan her tür malzemenin yönetim kademesi tarafından takip ve kontrolünün yapılabilmesini sağlamaktır. Her tür demirbaş ve sarf malzemesinin tüketimi yönetim tarafından sıkı bir şekilde denetlenebildiği takdirde hastanenin gelir gider dengesini optimum düzeyde tutmak daha kolay olacaktır.

Sistem, istenilen tarih aralıkları için raporlar düzenleyebilecektir. Bu rapor, her madde için maliyeti, alımların niteliğini, eldeki mevcut stok durumunu, tekrar düzenlenen noktaları, stok dışı şartları, otomatik olarak vermesi gerekli bilgileri içerecek şekilde olmalıdır.

Clemans (1979), hastane de mevcut araç-gereç ve cihazların bakım işlerinin programlanması ve bu konuda rapor hazırlamak amacı ile de sistemler geliştirilebileceğini, bu sistemlerin, denetçi raporları, yedek parça envanter denetimi, bakım gideri gibi işlevleri üstlendiklerini, ayrıca hastane yöneticisi malzeme tesis yönetim sisteminden işgücü dağıtımı, görevlilerin ve kaynak kullanımının denetimi gibi konularda da yararlanabildiklerini belirtmiştir (Çimen, 1994).

Endüstri de olduğu gibi hastane işletmelerinde de enerji tasarrufu önemli bir yer işgal etmektedir. Bu konuda hastanede enerji kullanımını izlemek üzere, paket programlar geliştirilmiştir. Bilgisayar destekli bu model de gerçek kullanım miktarı ile hesapla bulunan ihtiyaç miktarı karşılaştırılmakta ve tasarruf olanakları araştırılmaktadır.

Malzemenin kabulü, hastanenin ihtiyaç olarak tespit ettiği kalemlerin uygun bir prosedür içerisinde satın alınarak hastaneye girişinin yapılmasını, malzemenin kullanım safhası ise, hastaneye alınmış olan malzemelerin gerek hastane içinde stoklanarak, gerekse stok yapılmadan kullanımını yöneten fonksiyonları içermektedir. Bu iki temel faaliyetin yerine getirilmesinde YBS prensipleri kullanılır.

2.4.1.4. Personel Yönetim Bilişim Sistemi (PYBS)

Personel bir hastanenin en önemli kaynağıdır. Bu önemli kaynağın elde tutulabilmesi için bir hastane bütçesinin önemli bir kısmının personel giderlerine ve sosyal yardımlara ayrılmakta olduğu bilinen bir gerçektir. işgücü planlaması ve verimlilik analizi konularında da yararlanmak üzere hastanede iyi bir PYBS'ne ihtiyaç vardır. Hastane PYBS'de yer alacak bazı önemli fonksiyonları şu şekilde sıralamak mümkündür;

a. Hastane personelinin özlük dosyalarının tutulması ve saklanması.

b. Gerekli durumlarda bu dosyalardan bilgi alma ve yeni kayıtların ilave edilmesi.

c. Otomatik olarak görev denetiminin sağlanması.

d. Hastanenin her maliyet merkezine ait çalışma analiz raporlarının çıkartılması.

e. Personel problemlerini açığa çıkaracak raporların alınması, örneğin nöbet problemi ve işe devamsızlık.

f. Çalışanların özel yetenek ve belge durumlarının bilinmesi.

g. Hastane ücret bordrosu sistemine bağlı olarak personel giderlerinin maliyet dağılımının hesaplanması.

h. Personelin verimliliği ve kalite kontrolü ile ilgili bilgi sağlanması.

Personel Bilgi Sisteminden şu faydalar amaçlanmaktadır;

- İş görev tanımları hakkında bilgi üretmek. Bu bilgi işe müracaat eden kişiye yapacağı işin içeriğini ve ödenecek ücreti bildirir.
- Personel giderlerinin sabit tutulması, dengelenmesi, ve azaltılmasını sağlamak.
- Kırtasiye giderlerinin azaltılmasını sağlamak.
- Bütçe tahmini, işgücü analizi, kalite kontrolü, ve verimlilik standartlarının geliştirilmesi, devam durumunun analizi için personel bölümü olarak gerekli sayısal verilerin alınmasını sağlamak.

Bilgisayar sistemi tarafından üretilen raporlar, Aylık personel analizi, Üç aylık devamsızlık raporları, bölümler arasındaki üretkenlik; verimlilik raporları, yetkili

şahıslar tarafından talep edildiğinde alınabilir. Bu sistemin insan gücü kullanılarak yapılandan daha güvenilir ve ucuz olduğu bildirilmiştir.

Personel yönetim bilgi sistemi, iş yükü analizleri yaparak hastanenin eleman ihtiyacını gösterecek şekilde de dizayn edilebilmektedir.

2.4.2. Tanı ve Tedaviyi Desteklemeye ilişkin Sistemler

Tanı ve tedavi konuları ile ilgili bilişim sistemlerine hastane yönetiminde olduğu kadar ilgi gösterilmemiştir. Klinik bilişim sistemlerinin geliştirilmesinde YBS'ne nazaran bazı olumsuzluklar ve güçlükler yaşanmıştır. Bu tür çalışmalar büyük bir finans kaynağı ve emek gücü gerektirmektedir. Bu istemlerin çok güvenilir ve dikkatli bir tasarımın sonucu olması şarttır. Bu bilişim sistemlerinin güvenilirliğini sağlayabilmek için bazı özel şartların yerine getirilmesi gerekir. Bu şartları ise şu şekilde sıralayabiliriz.

- Sistemleri kullanacak personel dikkatli seçilmiş ve iyi yetiştirilmiş olmalıdır.
- Bilgisayar sistemlerinin ve yazılımının kullanılmasında, güvenliğin, gizliliğin mutlak sağlanması gerekir.

Doktorun karar vermesine yardımcı olmak Üzere, hastaya ait veriler ile tıbbi bilgileri karşılaştıran sistemlerde iki ayrı yöntem söz konusudur. Birinci sistemlerde doğrudan sinyal işlemleri uygulanmaktadır (Elektrokardiogramın yorumlanması ve elektrosefalogramların analizi konularında uygulanmaktadır), ikinci sistemlerde ise tıbbi düşünce yürütme ve karar verme beklenmektedir. Burada alınan tahlil sonuçlarına göre değerlendirme yapılır.

Tanı ve Tedaviyi Desteklemeye ilişkin Sistemler, hekimlere veya diğer sağlık personeline alacağı klinik kararlarda destek sağlayan bilgisayar programlarıdır. Bu sistemler bir bakıma karar desteği sağlamak için klinik veri ya da medikal bilgiyle ilgilenen bilgisayar sistemleridir. Günümüzde hekimler tıbbi bilginin miktarında meydana gelen artış nedeniyle bu bilgileri etkin bir şekilde yönetebilmek/değerlendirmek ve uzman yokluğunda uygun seçenekler arasında karar verebilmek için karar desteğine ihtiyaç duymakta ve bu amaçla klinik konularda akıl

yürütme özelliğine sahip karmaşık bilgisayar programlarından yararlanmaktadırlar (Güleş ve Özata, 2005).

2.4.2.1. Bilgisayar Destekli Tanı ve Tedavi Bilişim Sistemi

Bilgisayardan elde edilen ve tanıya yardımcı olan bilgiler yanında hastayı izleme ve tedavi konularında yararlanmak üzere bir takım sistemler de geliştirilmiştir. Bu konuda; bilgisayarda oluşturulan tedavi protokolü ve uyarı sistemleri, Radyasyon tedavisini planlama sistemleri, bilgisayar destekli tanı ve tedavi bilişim sistemleri için ilginç örneklerdir.

2.4.2.2. Bilgisayar Destekli Hasta Takip Bilişim Sistemi

Hasta takip sistemlerinde bilgisayar, hastanın yaşam sinyallerini sürekli olarak izlemek ve periyodik olarak fizyolojik verilerini göstermek işlevini üstlenmektedir. ilk aşamada hastaya bağlanan izleme donanımından elde edilen analog veri, bilgisayarda işlenecek ve görüntülenecek biçimde sayısala (dijitale) dönüştürülür. Bu veriler bellekte saklanarak, periyodik olarak ya da istenildiğinde görüntülenir. Burada amaç, öğretim ve eğitimdir. Bilgisayar programları, programlanmış düşünme ve karar verme kurallarına göre klinik verilerine trend analizleri uygulanarak, sınırlı verileri genişletmektedir. Trend verileri hasta sinyallerinin zaman içerisindeki değişimini görüntüden izlemeye de yaramaktadır. Hasta izleme sistemi; yoğun bakım servisinde, hasta yatağının yanında bulundurulacağı gibi bilgi aktarımı amacı ile uzaktaki bir yoğun bakım merkezine de bağlanabilmektedir (Çimen, 1994)

2.4.2.3. Bilgisayar Destekli Hasta Kayıtlarını Tutma Bilişim Sistemi

İnsanlar ihtiyaç duydukları sağlık hizmetini, kimi zaman bir zorunluluktan (örneğin acil bir durum) kimi zaman da kapasite, yetkinlik veya bakım kalitesinin daha iyi olması gibi gerekçelerle, farklı sağlık kurumlarından alırlar. Hizmetin farklı kurumlardan alınmasına bağlı olarak, teşhis ve tedavi ile ilgili tüm bilgiler sağlık kurumlarının arşivlerinde dağınık bir biçimde saklanır ve gereksinin duyulduğunda bu bilgilere ulaşılması neredeyse imkansız hale gelir. Ayrıca ulaşılsa bile, kağıda dayalı olmaları nedeniyle, bu kayıtlardan zamanında ve tam olarak faydalanılması mümkün olmaz. Bilgilerin kağıda dayalı olarak arşivlenmesi ve arşivden tekrar çıkartılıp değerlendirilmesindeki hata oranının yüksekliği, işin ikinci kez tekrarlanmasına, tedavinin gecikmesine, yatılan gün süresinin uzamasına, ve medikal hata riskinin

artmasına yol açmaktadır. Hasta dosyasının kullanıma hazır olmadığı durumlarda sağlık personeli, hastanın tıbbi hikayesi olmaksızın tedavi yapmak zorunda kalmaktadır. Oysa ki bir hekimin, hastanın geçmiş hastalık hikayesine veya daha önce başvurduğu kurumlarda· yapılan işlemlerine anında ulaşması sağlık hizmetinin etkinliğini arttırmak açısından bir zorunluluktur (Güleş ve Özata, 2005).

Hastalık kayıtlarının tedavi ve bakım süresince tutulması bir anlamda tüm hastanelerde bir zorunluluktur. Buradaki sistemin kolaylığı hastanın yattığı andan itibaren hastaneden çıkıncaya kadar geçen zamanda tıbbi kayıtlarının bilgisayar ortamında bir arşiv oluşturularak tutulmasıdır. Taburcu olunca da bir arşiv kaydı verilir. Hastanın tekrar başvurusunda ya da ayakta tedavisinde bu arşiv kaydı esas alınır. Hastalık kayıtları aynı zamanda doküman olarak bir denetim aracı ve doktor muayenelerinde de bir başvuru kaynağıdır. Özellikle eğitim hastanelerinde, araştırmalara veri tabanı olarak hizmet gören bir kaynak sayılmaktadır. Bu nedenlerle hastalık kayıtlarının dikkatli bir şekilde tutulması gerekmektedir (Çimen, 1994).

Hastalık kayıtları yönetiminde bilgisayar desteği büyük yarar sağlamaktadır. Bu sistem; genelde hastaların kimlik bilgilerini, hastalığın sınıflamasını kayıtların indekslenmesini içerir. Böylece herhangi bir araştırma ve denetimde kayıtları geri çağırmak mümkündür.

Bazı sistemler, bağlı sistem olarak geri çağırma olanağına sahiptirler. Bu şekilde hastalık kayıt özetleri acil serviste, ayakta tedavi gören kliniklerde ve başvuru bürolarındaki terminal ekranlarında görülebilir. Tabi bu amaç için Doktorların kolaylıkla ulaşıp kullanabileceği bilgisayarların kurulu bulunması gerekmektedir.

2.4.2.4. Bilgisayar Destekli Laboratuar ve Eczane Bilişim Sistemi

Laboratuar bilişim sistemleri; test işlemlerini otomatik olarak yapmak ve laboratuar verilerini iş/emek biçiminde, iki aşamalı işlevi üstlenmektedir. Test yapan laboratuar aletlerini bilgisayara bağlayarak bulguların sonuçlarının alınması ve hasta dosyasına işlenmesi sağlanır. Bir hastanede laboratuar bilgi sistemi her birimde bağımsız olarak çalışabileceği gibi laboratuar otomasyon sistemine de bağlanarak da çalışabilir. Bir laboratuar bilgi işlem sistemi; test isteklerinin kaydı, örnek toplanmasını programlama ve test işlemleri, tamamlanan test sonuçlarını kaydetme, ilgili yere hemen iletilmesi gereken test raporlarını düzenleme, hastaya ait test özetlerini hazırlama, laboratuara ait

istatistik bilgileri çıkarma, laboratuarın yönetim kademesince denetimi ve kalite kontrolü için gerekli kayıtları tutma işlevini üstlenmektedir.

Komple bir laboratuar bilgi sisteminden beklenen alet otomasyonu ile bilgi işlem fonksiyonlarının birlikte yürütüldüğü bir yönetim bilgi sistemidir. Böylece hastadan test örneği alındıktan sonra, otomatik olarak bütün işlemler tamamlanır ve sonuç bildirilerek kayıt edilir.

Hastane eczanelerinde de tam bir otomasyon ihtiyacı mevcuttur. İlaçların siparişi, stoklanması, ve dağıtımının dikkatli biçimde denetimi, az da olsa tıbbi hatalardan sakınmak üzere güvenli kayıtlar tutmak ve saklamak bir zorunluluk olarak karşımıza çıkmaktadır. Aynı zamanda muhasebe ve karlılık hesapları içinde bilgi sistemine ihtiyaç duyulur. Eczane bilgi sistemleri de; bağımsız ya da hastane bilgi sistemine entegre olarak çalışabilmektedir. Genelde bağımsız eczane bilgi sistemleri, zehirli ilaçların denetimi, ilaç siparişi ve envanter kontrolü, hastalara ilaç dağıtımının denetimi, ilaçlarla ilgili bilgi birikiminin oluşturulması ve hasta borçlarının hesaplanması konularına yönelik geliştirilmiştir.

2.4.3. Entegre (Tam Bütünleşik) Hastane Yönetim Bilişim Sistemi

Tam bütünleşik bilişim sistemlerinde amaç, hastanedeki tüm ana bilişim sistemi fonksiyonlarının bir arada bağlantılı (entegre) olarak yürütülmesidir. Çimen (1994), hastane entegre bilişim sistemi kavramı kapsamında yer alan, konuları şu şekilde ifade etmiştir;

- Hastane Genel Bilgi Sistemi: Hastanede yapılan günlük işlemleri kolaylaştırmak amacı ile, tüm hastane için dizayn edilmiş bir bağlı (on-line) sistemin kurulmasıdır.
- Tıbbi Bilgi Sistemi: Tanı ve tedavi gibi konularda, tıbbi yorumlarda bulunmaya yardımcı olan, veri toplayan ve işleyen bir sistemdir.
- Yönetim Bilgi Sistemi: Hastane yönetiminde düşünce yürütme ve karar verme, planlama ve denetim görevlerinde yararlanmak amacıyla verileri toplayan ve işleyen bir sistemdir.
- Veri Tabanı Yönetim Sistemi: Kullanılacağı programlardan bağımsız olarak, bir biri ile ilişkili görülen olabildiğince çok veriyi toplamak, bu verileri gerektiğinde geri

çağırılacak ve özel raporlar haline getirilebilecek nitelikteki genel bir format içerisinde saklama işlemlerini gerçekleştirir.

Çimen (1994), entegre hastane bilgi sistemlerinde aşağıdaki koşulların yerine getirilmesinin beklendiğini belirtir;

- Hastanelerde anahtar bölgelere yerleştirilmiş olan özel terminaller vasıtasıyla online sistemi ile doğrudan veri girişinin sağlanması, geri çağrılması ve görüntülenmesi,
- Hastanenin istenen her birimine bilgi iletilebilmesi için otomatik bilgi gönderecek sistemin kurulması,
- Hastanedeki günlük işlerin birleştirilmesi ile, sistemin bir bütünü oluşturması,
- Tam bütünleşik sistemin kurulmasında;veri dosyalarının düzenli olması, veri tanımlarının kullanılması, bilgi işleme ve alt sistemler arasında bilgi işlem yöntemlerinin otomatik yürütülmesi koşullarının yerine getirilmesi beklenmektedir.

Hastane faaliyetlerine yönelik geliştirilen entegre HYBS, bir hastanenin tüm uygulamalarına cevap verebilecek özellikte hazırlanmış tıbbi, idari ve mali bilgi bütünlüğünü entegre bir ortamda sağlayan organizasyon etkinliğini artıran bir uygulama yazılımıdır. Bu yazılım paketi ile hastanelerin personel kullanılarak izlenen özel veya genel çalışmaları, bilgisayar ortamına taşınır. Hastanenin bilgi akışının doğruluğu, hızı ve güvenliği artarken hastane genelinde ve bölüm seviyesinde mevcut fonksiyona yararlı bir çözüm sağlanır.

Hastaneler, insan sağlığı ile ilgili yoğun kapasitede veriyi işler, kontrol eder ve saklar. Bütün faaliyetler boyunca bu veriler, muhtelif yerlerden hastanın dosya veya dosyalarına değişik formatlarda ve ortamlarda doldurulur, her birimde de değişik amaçlı yorumlanır. Her yorum ve esas olan bilgi, ihtiyaç duyulduğunda erişilmek için arşivlenir. işte kurulacak entegre YBS gerektiğinde erişilmek üzere bu bilgi birikiminin oluşturulmasını sağlar.

Kullanıcı işlevselliği bakımından modüler olan bir yapının, tüm sistem boyunca entegre bir veri tabanı içerisinde tüm bu modüler işlevleri birleştiren özelliğe sahip olması gereklidir. Hastanın laboratuar tahlilleri ile ilgili istekler; isteği yapan bölümün işlevi

olduğu kadar bu istekle ilgili tahsilat ve sonuç girme işlemini yapan tahakkuk bürosu, vezne, laboratuar gibi bölümleri de ilgilendirmektedir.

Entegre yapı birimler arasında oluşmaya başladıkça kuruluşun elle doldurulmuş belge tutma ve takip etme ihtiyacı ortadan kalkmaya, gereksiz olmaya başlayacaktır. Ancak, terminallerin ulaşmadığı birimler var ise bu tür birimlerin sisteme katılmamalarından dolayı genel sistem akışının bozulmasını engelleyecek şekilde bir entegre yapı oluşturulmalıdır.

Modüler yapı içerisinde donanımlar dağıtılmış olabilir (Laboratuarda, Eczanede, Acil serviste, Radyolojide ayrı sistemlerin olması gibi). Burada entegre yapı yazılımı, donanımın dağılmış olmasını gözetmeksizin bütünde tek bir sistem olarak çalışabilme özelliğini gösterecek yapıda olmalıdır.

Hastaneler için kurulacak Entegre YBS belli başlı standardı olmayan bu sektörün dinamizmine uygun bir esnekliğe sahip olmalıdır. Hastanelerdeki kullanıcıların farklı kesimlerden oluşması (Doktorlar, Hemşireler, Memurlar, Yöneticiler, Diğer görevliler v.b.) kullanılacak yazılımın belli bir esnekliğe (kolay kullanım, parametrik yapı) sahip olması zorunluluğunu getirmektedir. Bu denli farklı uçlardan oluşan kullanıcı kitlesinde kullanım bir kesime kolay gelirken diğerine de zor gelebilir. Bu nedenle kullanılacak entegre sistem her türlü son kullanıcının yaptığı işe yatkın ve kullanıcısına hitap edebilecek esnekliğe sahip olmalıdır. Yazılım esnekliği yanında yazılımın üzerinde çalıştığı veri tabanı, işletim sistemi ve donanımında aynı esnekliğe sahip olması gereklidir.

Kurulacak Entegre YBS, esnek bir menüye sahip olması, kullanıcıların rollerinin tanımlanmasını ve iş organizasyonuna göre baştan tanımlanmış görevlerine göre hareket etmesini sağlayacaktır. Bu esneklik içerisinde kullanıcının görev tanımında değişiklikler oluştukça sisteme yeni fonksiyonları tanıtabilecektir. Örneğin, Laboratuarda sadece kan alma işlemi yapan kişinin karşısında olması gereken ekranın hiç bir fazla bilgiye gerek olmadan sadece kendi işi ile ilgili olarak "Kan alma ve test onaylama” menü seçeneğinin olması hem eğitimi hem de uygulamaya geçişi kolaylaştıracaktır (Çimen, 1994).

Entegre HYBS kuruluş aşamasında sistemin; donanım, işletim sistemi, veri tabanı belirlenirken kuruluş ihtiyacının ne olduğu ön planda tutulmalı kullanışsız bir donanım veya yazılım sahibi olma riski konu ile ilgili olanlarca göz önünde bulundurulmalıdır.

2.4.4. Modüler Hastane Yönetim Bilişim Sistemi

Hastaneler entegre uygulamalar için önemli bir platform olduğu gibi farklı işletim ve uygulama ortamlarının kendi içlerinde ve diğer birimlerle olan ilişkilerini de kapsar (Laboratuar, Radyoloji, Nükleer Tıp, Poliklinik, Hasta Kabul, Eczane gibi). Bu bölümlerdeki faaliyetler göz önüne alındığında modüler bir tasarım yapılması gereği ortaya çıkmaktadır. Ancak modüler yapı entegre sistem ile uyumlu hale getirilmediği takdirde birimler birbirinin işleyişinden habersiz olarak sonuç bekleyici bir biçimde faaliyetlerini sürdüreceklerdir (Çimen, 1994).

Modüler yapı kuruluşun ilk yatırım maliyetini düşürür. Özellikle safhalara bölünmüş kuruluş işlemlerinde ilk elde demode olabilecek büyük donanım yatırımının yapılmasına gerek kalmayacaktır. Örneğin ilk safhada döner sermaye uygulamasının faturalama kısmından başlanarak, alacakların takibi, muhasebe, stok, ambar, klinik, poliklinik, Laboratuar, Eczane, Radyoloji zincirlemesi içerisinde sonuca gidilir. Böyle bir yaklaşımla, modül bazında eğitim programı kolaylıkla uygulanabilir ve son aşamaya kadar kuruluş içerisindeki faaliyetlerin oturtulması sağlanır.

Modüler yapının sağladığı diğer bir avantaj, bütünlüğün korunması için merkezi bir yapı gerektirmemesidir. Bu yapı ile hizmet veren birimler bizzat hizmet sunan birim olarak kendi modülleri ile, ilişkin hizmetler için diğer birimlerle veya diğer birimlerden aldıkları hizmetle faaliyetlerini sürdürebileceklerdir. Böyle bir yapı ilk yatırım maliyetini büyük ölçüde düşürecektir.

2.5. HASTANE YÖNETİM BİLİŞİM SİSTEMLERİNDE KULLANILAN TEKNOLOJİLER:

2.5.1. Veri Tabanı Teknolojileri:

Veri Tabanı, düzenli bilgiler topluluğudur. Bir diğer değişle verilerin belli özelliklerine göre gruplanıp yönetilmesi işine veritabanı yönetimi denir. Bu tanım çerçevesinde bir fihristli telefon defteri de veritabanıdır (Gözüdeli, 2004). Kelimenin anlamı bilgisayar ortamında saklanan düzenli verilerle sınırlı olmamakla birlikte, daha çok bu anlamda kullanılmaktadır. Bilgisayar terminolojisinde, sistematik erişim imkanı olan, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir. Bir başka tanımı da, bir bilgisayarda sistematik şekilde saklanmış, programlarca işlenebilecek veri yığınıdır. Çölkesen (2006), veritabanını bilgilerin belirli bir disiplin altında saklanması ve gerektiğinde hızlı bir şekilde aranıp bulunması/sorgulanması için geliştirilmiş bir saklama/sorgulama sistemi olarak tanımlar. Bir veri tabanını oluşturmak, saklamak, çoğaltmak, güncellemek ve yönetmek için kullanılan programlara Veri Tabanı Yönetim Sistemi (VTYS) (Data Base Management System-DBMS) adı verilir.

Uygulamalar, çeşitli kaynaklardan üretilen verileri yönetebilmek üzere veritabanı ortamında saklarlar. Veritabanı ortamı, uygulama programlarının verilere daha hızlı erişmesi, raporlaması yoluyla performans artışı ve kolay veri yönetim ortamı sağlar. Veritabanı programlamada kullanılan veritabanı seviyeli diller, SQL (Structred Query Language) standartları çerçevesinde şekillenmiştir (Gözüdeli, 2004).

Veritabanını bilgisayar ortamında gerçeklerken çeşitli yaklaşımlar geliştirilmiştir. Bunlardan en popüler ve en çok tercih edileni İlişkisel Veritabanı yaklaşımıdır (Gözüdeli, 2004). SQL dili ve üzerine kurulu olan ilişkisel veritabanı sistemleri, bilgisayar endüstrisindeki en önemli temel teknolojilerden biridir. SQL bugün standart veritabanı dili olarak karşımıza çıkıyor. Kişisel bilgisayarlardan tutun, ana bilgisayarlar ve el araçları üzerinde çalışanlara kadar yüzlerce veritabanı ürünü SQL'i destekliyor. Resmi uluslararası bir SQL standardı oluşturularak, o günden beri iki kez genişletilmiştir. Neredeyse tüm önde gelen ticari yazılım ürünleri, veri

yönetimi konusunda SQL'e güveniyor ve SQL, dünyanın en büyük üç yazılım şirketi olan Microsoft, Oracle ve IBM'in veritabanı ürünlerinin çekirdeğini oluşturuyor. SQL, Linux'un ve açık kaynak (open source) hareketinin yaygınlaşmasını hızlandıran açık kaynaklı veritabanı ürünlerinin de kalbi durumundadır. Bir IBM araştırma projesi olarak yaşamına belirsizlik içinde başlayan SQL, hem önemli bir bilgisayar teknolojisi, hem de büyük bir market gücü olarak baskınlığını sürdürmektedir (Groff ve Weinberg, 2003).

SQL'in işlevleri aşağıdaki gibi sıralanabilir:

- Veri tanımlama; SQL, bir kullanıcının saklanan verilerin yapısını, düzenlenişini ve birbirleriyle olan ilişkilerini tanımlamasına izin verir.
- Veri elde etme; SQL, bir kullanıcının veya uygulama programının veritabanından veri almasına ve kullanmasına izin verir.
- Veri üzerinde değişiklik yapma; SQL, bir kullanıcının veya uygulama programının, yeni veriler eklemek, eski verileri silmek ve daha önceden saklanmış olan verileri değiştirmek yoluyla veritabanını güncellemesine izin verir.
- Erişim kontrolü; SQL, bir kullanıcının verileri elde etme, değiştirme, silme ve verilere ekleme yapma yeteneğini kısıtlayarak, saklanan verileri yetkisiz erişimden korumak amacıyla kullanılabilir.
- Veri paylaşımı; SQL, eş zamanlı çalışan kullanıcıların veri paylaşımını ayarlayarak, birbirleriyle çakışmamalarını sağlamak için kullanılabilir.
- Veri bütünlüğü; SQL, veritabanındaki bütünlük kısıtlamalarını tanımlayarak, tutarsız güncellemelerden veya sistem hatalarından kaynaklanan bozulmalardan korur.

SQL, ne tek başına bir veritabanı yönetim sistemi, ne de başlı başına bir üründür. Bunun yerine SQL, bir veritabanı yönetim sisteminin bütünleşik bir parçası, bir dil ve VTYS ile iletişim kurmak için bir araçtır. Veritabanı motoru, VTYS'nin kalbidir ve veritabanındaki verilerin yapılandırılmasından, saklanmasından ve elde edilmesinden sorumludur. Diğer VTYS parçalarından (bir form etkinliği, rapor yazarı veya etkileşimli

sorgu etkinliği gibi), kullanıcı tarafından yazılmış uygulama programlarından ve hatta diğer bilgisayar sistemlerinden gelen SQL isteklerini kabul eder.

- SQL, etkileşimli bir sorgu dilidir. Kullanıcılar veri almak için etkileşimli bir SQL programına komutlar yazarlar ve bunu ekran üzerinde görüntülerler. Bu da, yoğun veritabanı sorguları için uygun ve kullanımı kolay bir araçtır.
- SQL bir veritabanı programlama dilidir. Programcılar, veritabanındaki verilere erişebilmek için uygulama programlarına SQL komutları gömerler. Hem kullanıcı tarafından yazılmış programlar, hem de küçük veritabanı uygulama programları (rapor yazıcılar ve veri giriş araçları gibi) veritabanına erişim için bu tekniği kullanır.
- SQL bir veritabanı yönetim dilidir. Bir küçük bilgisayar veya ana bilgisayar veritabanını yönetmekle sorumlu olan veritabanı yöneticisi, veritabanı yapısını belirlemek ve depolanmış verilere erişimi kontrol etmek için SQL kullanır.
- SQL bir istemci/sunucu dilidir. Kişisel bilgisayar programları, paylaşılan veri depolayan veritabanı sunucuları olan bir ağ üzerinden iletişim kurmak için SQL kullanır. Bu istemci/sunucu mimarisi, ticari tipteki uygulamalarda son derece popüler hale gelmiştir.
- SQL, bir Internet veri erişim dilidir. Kurumsal verilerle etkileşime giren Internet Web sunucuları ve Internet uygulama sunucularının tümü SQL'i, kurumsal veritabanIarına erişimde standart dilolarak kullanır.
- SQL dağıtık bir veritabanı dilidir. Dağıtık veritabanı yönetim sistemleri, birbirine bağlı birçok bilgisayar sistemi arasında verilerin dağıtılmasına yardımcı olarak SQL kullanır. Her bir sistem üzerindeki VTYS yazılımı SQL'i, veri erişimi için istekler göndererek diğer sistemlerle iletişim kurmak için kullanır.
- SQL bir veritabanı ağ geçidi (gateway) dilidir. Farklı VTYS ürünlerinin karışımından oluşan bir bilgisayar ağında SQL çoğunlukla bir VTYS markasının başka bir markayla iletişim kurmasını sağlayan bir ağ geçidi olarak kullanılır.

Tüm bu nedenlerden dolayı SQL, ilişkisel bir veritabanında depolanan verilerle kişileri, bilgisayar programlarını ve bilgisayar sistemlerini bağlamak için kullanışlı ve güçlü bir araçtır.

SQL Standartları: SQL'in resmi standardı ilk olarak Amerikan Ulusal Standartlar Enstitüsü (ANSI - American National Standards Institute) ve Uluslararası Standartlar

Örgütü (ISO - International Standards Organization) tarafından 1986'da yayımlandı ve sırayla 1989, 1992 ve 1999 yıllarında genişletildi. SQL ayrıca, bir A.B.D. Federal Bilgi İşleme Standardı'dır (FIPS - Federal Information Processing Standard). Bu yüzden de büyük devlet bilgisayar kontratlarında başlıca gerekliliklerden biridir. Yıllar geçtikçe diğer uluslararası, devlet ve satıcı grupları, çağrı düzeyli arabirimler ve nesne tabanlı uzantılar gibi yeni SQL yeteneklerinin standartlaştırılmasına öncülük etti. Bu yeniliklerin birçoğu ANSI/ISO standardına eklendi. Gelişen bu standartlar, SQL'in onaylanmasına resmi bir damga vurdu ve pazarda kabul görmesini hızlandırdı (Groff ve Weinberg, 2003).

Internet Veritabanı Erişimi: 1990'ların sonlarındaki Internet ve World Wide Web patlaması ve bunların standart tabanlı temeliyle birlikte SQL, kendine yeni bir rol edindi: Internet veri erişimi standardı. Web'in gelişiminin ilk zamanlarında geliştiriciler, veritabanlarındaki bilgileri alıp Web sayfaları üzerinde göstermenin bir yoluna ihtiyaç duyuyorlardı ve SQL'i veritabanı çıkışları için ortak dil olarak kullandılar. Yakın zamanda farklı istemci, uygulama sunucusu ve veritabanı sunucusu katmanlarına sahip üç katlı (three-tiered) Internet mimarilerin doğuşuyla birlikte SQL uygulama ve veritabanı sıraları arasındaki standart köprü haline geldi. Gelecekte Internet'te SQL'in rolü, Web sitesi mimarilerinin ötesine geçip, yardımcı uygulamaları ve bir Web Servisleri mimarisinde dağıtık nesneleri de kapsayacaktır (Groff ve Weinberg, 2003).

Endüstri Altyapısı: SQL'in artan önemine katkıda bulunan belki de en önemli faktör, tamamen SQL üzerine kurulu bir bilgisayar endüstrisinin doğmuş olmasıdır. SQL tabanlı ilişkisel veritabanı sistemleri, bu altyapının önemli bir parçasıdır. SQL kullanan ve SQL tabanlı bir veritabanı gerektiren ticari uygulamalar da rapor araçları, veri girme araçları, tasarım araçları, programlama araçları ve SQL'in kullanımını kolaylaştıran başka araçlar kadar önemli olan diğer bir parçadır. Deneyimli SQL programcılarından oluşan bir ordu da altyapının önemli bir parçasıdır. Diğer önemli parça ise, SQL'i çevreleyen ve SQL uzmanlarını artıran eğitim ve destek hizmetleridir. SQL danışmanlığı, optimizasyonu ve performans ayarlama gibi konularda tamamen ayrı bir SQL alt endüstrisi oluşmuştur. Bu altyapının tüm parçaları birbirini güçlendirir ve diğer parçaların başarısına katkıda

bulunur. Basitçe söylemek gerekirse; veri yönetimi sorunlarını çözme de en kolay, en az riskli ve en az masraflı çözüm her zaman, SQL'e dayalı olan çözümdür.

İlişkisel veritabanı öncesinde kullanılan sistemlerin başarılı olduğu uygulamalar olsa da, bazı eksiklikleri vardı, başlıcaları şöyledir (Önder 2005):

- Fiziksel veri depolama ve erişim gereksinimleri, verinin mantıksal tasarımını sınırlayıcı şekilde etkiliyordu. Veritabanları temsil edilen sistemleri değil, depolama aygıtlarının yapısını yansıtacak biçimde tasarlanıyordu.

- Veriyi elde etmek ve güncellemek, verileri kayıt-kayıt işleyen uygulamalar tarafından yapılmak zorundaydı. Bu veri kümesini güncellemeyi bir tek işlem olarak uygulamak olanaksızdı.

- Veri bütünlüğü, veri üzerinde değişiklik yapan uygulamaların sorumluluğundaydı. Dolayısıyla veritabanı bütünlüğü, veriye erişen her uygulamanın bütünlük seviyesinin ötesinde garanti altına alınamıyordu.

Sonuç olarak, veri eklemek, güncellemek, silmek ve elde etmek karmaşık bir programlama göreviydi. Veriye erişen her program bütünlüğe zarar verebilirdi.

2.5.2. Programlama Teknolojileri

Programlama dilleri, bilgisayar donanımını fazla bilmeden program geliştirmeye imkan veren bir kodlama şeklidir. her biri farklı bir alana yönelmiş veya genel amaçlı olarak tasarlanmış Basic, Pascal, C, Fortran gibi adlandırılan birçok programlama dili vardır. Bunlar genel olarak üst veya orta düzey programlama dili olarak adlandırılır. En temelde, çalışabilen bir program makine kodunda olmalıdır. Mikroişlemci ancak makine kodundaki komutları çözümler ve yürütebilir. Ancak, makine koduyla program geliştirmek oldukça zor ve zahmetli olduğundan assembly dili ve üst düzey programlama dilleri geliştirilmiştir. Amaç programcıların kolayca anlayabilecekleri ve kodlama yapabilecekleri, konuşma dillerine daha yakın dillerin kullanılmasını sağlamaktır (Çölkesen, 2006).

Çölkesen (2006), Görsel programlama dillerini, kullanıcıya görsel arayüz sunan Windows, Görsel arayüzlü Unix/Linux işletim sistemleri üzerinde asıl program kısmına ek olarak görsel arayüz oluşturma olanağı sunan diller şeklinde tanımlar. Bu tür diller hem amaçlanan algoritmanın tasarlanması hem de kullanıcıyla etkileşim için pencere tabanlı görsel arayüz tasarlanması imkanı verirler. Çok değişik görsel programlama dilleri geliştirilmiştir. Örneğin ilk akla gelenleri Windows için Visual Basic, Visual C, Delphi; Linux için GTK+/GNOME kütüphanesidir.

Borland Delphi, Microsoft Windows Platformu için uygulamalar geliştirmek amacıyla tasarlanmış ve yaygın olarak kullanılan bir uygulama geliştirme aracıdır. Delphi özellikle Pascal temeli olan programcılar için nesne yönelimli ve görsel bir programlama ortamıdır. Zengin bileşen kütüphanesine, veritabanı özelliklerine ve uygulamaları daha iyi test etmek için geliştirilmiş yardımcı programlara sahiptir. Özellikle yeni sürümleriyle (Delphi 5 ve 6), veritabanı uygulamaları, ODBC bağlantısı, XML ve Web tabanlı uygulamalar geliştirmeyi kolay bir şekilde sağlar. Delphi 7 ve sonraki uyarlamalarında ise Win32 ve .NET platformu için de program geliştirmek mümkündür. GNU/Linux platformu üzerinde geliştirme imkanı sağlayan Kylix isimli bir sürümü de bulunmaktadır. Ne yazık ki Kylix yeni uyarlamalarda devam ettirilmemiştir. Nesne, sınıf, kalıtım, fonksiyon, aşırı yükleme (overloading) gibi temel teknikleri ve daha fazlasını içeren bu yönüyle de C++'dan aşağı kalmayan güçlü ve esnek bir programlama dilidir.

Delphi, görsel ve grafik kullanıcı arabirimine sahip uygulama geliştirme aracıdır. Bir anlamda Pascal ya da Turbo Pascal dilinin, grafiksel uygulama geliştirme ortamına dönüştürülmüş şeklidir. Programlama gücü ve deneyiminin yanı sıra, değişik veritabanlarına erişim sağlar. BDE olarak adlandırılan kendi veritabanı sisteminin yanı sıra ODBC ile yaygın olarak birçok veritabanına erişim sağlar. Yanık (1997), Delphi'yi bir çok yönüyle Pascal'a benzeyen, Pascal tabanlı nesne yönelimli göresel program geliştirme aracı olarak tanımlar.

Delphi programlama dili nesne yönelimli bir dil olduğu için eklenen bütün nesnelerin (Formlar da dahil) kodlarını oluşturur. Bu kodları "Unit" ler içerisinde barındırır.

İnternet platformu için daha çok Web tabanlı programlama dilleri olarak Java, ASP, PHP gibi dillerin yaygın olarak kullanıldığı görülmektedir.

2.5.3. Bilgi İletişim Teknolojileri

Network sistemleri, 1994 yılında internetin genel kullanıma açılması ile birlikte daha farklı şekillerde kullanılmaya başlandı. "Networking" kavramından ayrı olarak "internetworking" kavramı gelişti (Dolanbay, 2000). Bu internetworking işlemi, içine intemeti de alan networkleri, ya da networklerin internet aracılığı ile iletişimini tanımlamakta kullanılmaktadır. İntemetin bağımsız kişiler tarafından kullanımına ek olarak, şirket içi ya da şirketler arası iletişimde kullanılan iki farklı yöntem de söz konusudur: Intranet ve Extranet.

İnternet: 1994'te genel kullanıma açılıp, beş yıl içinde on milyonlarca kullanıcıya sahip olmuştur. 1969 yılında, Los Angeles'ta bulunan California Üniversitesi'nden Boucler Hall'un "bu mesajı alıyor musunuz?" diye UCLA Stanford Araştırma Enstitüsü, UC Santa Barbara ve Salt Lake City'deki Utah Üniversitesine gönderdiği mesajla kullanımına başlanan internette, şu anda yalnızca instant mesajlardan günde ortalama 100 milyon adet gönderilmektedir. (Dolanbay, 2000)

Bu networklerin networkü, kendi kendisini kopyalayabilen, geometrik olarak çoğalan bir mikro organizma gibidir.

İnternet, günümüz çalışma koşullarını tamamen yeniden tanımlayıp, gelecek yıllara bambaşka bir altyapı ile ulaşmamıza ön ayak olmaktadır. İnternet üzerinde, ne istediğinizi bilyorsanız ulaşmak zor olmayacaktır.

Dolanbay (2000), internetin hızla yaygınlaşmasının nedenlerini şu şekilde sıralar;

- Bilgisayarların günden güne daha rahat kullanımı,
- Geliştirilen donanım, işletim sistemi ve uygulamaların, kişilerin kolaylıkla bilgisayar sistemlerine adapte olmasını ve kullanılabilmesini sağlaması,
- Teknolojinin hızlı gelişimi sonucu, bilgisayarların yaşamın her alanında daha yoğun kullanılması dolayısıyla bilgisayar bilgisinin artması,

- Global bilginin önemi arttıkça, ona ulaşmayı sağlayacak araçların sayısı ve türü artarak, maliyetlerinin düşmesi,
- World Wide Web intemetin temelini oluşturmasıdır.

Dolanbay (2000) internetten yararlanmanın başlıca nedenleri şu şekilde sıralamaktadır;

- İşletmenin imajını güçlendirmek,
- Müşteri hizmetlerini yaygınlaştırmak ve kolay ulaşılır kılmak,
- Yeni hizmet alanları yaratmak,
- Görselliği ve ulaşılabilirliği sağlamak,
- Çalışmaların takibini ve standardizasyonunu gerçekleştirmek,
- Pazar payını artırmak,
- Müşteri memnuniyetini artırmak,
- Maliyetleri düşürmek,
- Geç olmadan hız kazanmak.

İnternet'in tek amaca yönelen bir kullanımı ya da en uygunu şeklinde bir belirleme yapmak olanaklı değildir. E-ticaret amacına yönelik olarak çalışmalar paralelinde, ayrıca birlikte çalışılan şirketlere ortak bir extranet sistemi, ya da farklı yerlerde konumlanan ofisler arasındaki iletişimin hızlı ve güncel olarak oluşturulabilmesi için de bir intranet sistemi oluşturulabilir.

İnternet ortamının beraberinde getirdiği çeşitli dezavantajlar şöyle sıralanabilir Dolanbay (2000);

- İşletmede mevcut personelin internet ortamında performansı istenen düzeye ulaştırılamayabilir.
- Piyasa rekabeti, internet kullanıcıları tarafından çok daha rahat değerlendirilmektedir.
- Yeni ve farklı kuralları olan bir pazara yönelik olarak çalışmalar yeniden şekillendirilmesi gerekebilir.
- İstenmese bile, internet ortamına herkesin erişme olanağı vardır.

World Wide Web (www): İnternete grafik arabirimi ekleyen uygulamanın adıdır ve sonuçta bir yazılım olarak değerlendirilebilir. İnternet kullanımının bu kadar çabuk yaygınlaşmasındaki en önemli nedenlerden bir tanesi kullanım kolaylığıdır.

İnternet browserları aracılığı ile, isteyen herkes, sunmak istediği bilgiyi görsel şekilde kullanıcılarına açtı. Kullanıcılar da bu bilgilere birkaç adres yazma ve tıklama ile ulaşabilir hale geldi. 1994 yılında, internet adresi olan işletme sayısı 17,000 iken, bu rakam 1995'te 70,000 oluverdi. Şu anda bu rakamın kesin tahmini neredeyse olanaksızlaşmıştır. (Dolanbay, 2000)

İnternet üzerinde bu tür hizmetlerin sağlanması için, işletmeler kendilerine ait yöntemleri belirlemek durumundadır. Bir şirket, web'de hizmet sunmaya başladığında, 24 saat kesintisiz kullanıcıların istediği yerde ve her zaman hazır olduğunu açıklamış olmak tadır.

Web Ortamı: Mevcut network sistemlerine ve bilgi akışına web avantajları eklendiğinde, geleneksel yöntemlerle ulaşılamayacak yararlar elde edildi, şöyle ki;

Platform bağımsızlığı: HTML ve Java'nın kullanımı, bilginin bilgisayar sistemlerinden bağımsızlaşmasını sağladı. Böylece, farklı sistemlere yönelik özel çözümler geliştirmek işlemi yerini internette kullanılabilecek şekilde tasarımlara bıraktı. Bu da yıllardır alışageldiğimiz ve yaygın olarak kullanılan yöntemlerin tekrar gözden geçirilmesine ve yeni yatırımlara neden olmaktadır. (Dolanbay,2000)

Global erişilebilirlik: Web'de yer alan tüm bilgiler, tanımlandıkları formatta dünyanın herhangi bir yerinde bulunan başka bir bilgisayardan erişilebilecek şekilde tanımlanabilmektedir.

Uluslararası kullanıcı arabirimi: Kullanıcılar, web kullanımını biliyorlarsa, internette sunulan hizmetlerden yararlanmak için ayrıca eğitim almak durumunda değildirler.

Bilim adamları, hukukçular, finans çalışanları gibi belli bir gruba hizmet veren sitelere bağlanan kullanıcılar, kendileriyle aynı meslekte olan diğer kullanıcılarla internet

üzerinde görüşebilmektedirler. Bu iletişim özel olarak hazırlanan bir konuşma penceresinden olabileceği gibi, video konferans şeklinde de gerçekleştirilebilir.

İntranet: İnternet teknolojisinde kullanılan araçlardan, işletme içi iletişim amacı ile kullanılacak şekilde yararlanılması intranet olarak isimlendirilmektedir. İşletmelerde zaten kullanılmakta olan networklerin internet teknolojileri aracılığı ile iletişime geçirilmesi, internetin yararlarının işletme çalışmalarına eklenmesi olarak değerlendirilebilir.Ancak bir intranet sitesinin mutlaka internete bağlı olması gerekli değildir. İntranet sistemleri, internette kullanılan aygıt ve araçların farklı yöntemlerle tanımlanması ile de oluşturulabilmektedir (Dolanbay, 2000).

İntranet sitelerinde dikkat edilmesi gereken noktalar şunlardır:

- Periyodik güncelleme,
- İşletme içinde tüm yöneticilerin efektif şekilde kullanmasının sağlanması,
- Kullanımını yaygınlaştırmak için farklı yöntemlerin geliştirilmesi,
- Tüm personele intranet kullanım eğitimi sağlanması,
- Gelişmelerin yakından takibi ve sisteme entegrasyonu,
- Kullanıcılar tam olarak hazır olmadan kesinlikle geçiş sürecinin tamamlanmaması.

Extranet: Farklı türde bilgi işlem platformları içeren işletmeler arasında network sisteminin kurulması olarak özetlenebilir. Bu farklılığın getirdiği en önemli kısıtların başında, donanım farklılığı değil, çalışma yöntemlerinin farklılığı gelmektedir. Extranet sistemleri ile birlikte, benzer çalışmaların işletmeler arasında standartlaştırılması gerçekleştirilebilir. Bu standartlaşma, çalışmaların ortak gerçekleştirilmesi anlamına gelmeyip, yalnızca bilgi değişimini de içerebilir. Ancak bu tür yapılar, extranetlerin kullanım amaçlarına uygun değildir (Dolanbay, 2000).

Extranet sistemleri, internet ile işletme dahilinde kullanılmakta olan intranet sistemleri arasında bir köprü olarak tanımlanabilir. Extranet sistemleri sanal duvarlarla internet kullanıcılarından ayrılarak, işletme içi ya da birlikte çalışılan işletmeler arası iletişimde internet altyapısının kullanılması amacını taşımaktadır.

Extranet çözümleri, büyük organizasyonların daha önce de kullanmakta olduğu EDI sistemlerinin, internet temeline aktarılmış ve kapsamının genişletilmiş/geliştirilmiş versiyonları olarak ele alınabilir.

İnternetin getirdiği potansiyel ve intranetin sunduğu iletişim kolaylığını tek bir ortamda kullanılması temeline dayanan extranet sistemlerinin güvenlik bileşenleri, tasarımlarının en can alıcı noktasıdır.

Özellikle merkezi olmayan işletmelerin bünyesinde çeşitli amaçlara yönelik olarak hazırlanan intranet sitelerine tüketici ilişkilerinin de eklenmesi, extranet sistemlerine geçiş anlamına gelmektedir. Extranet sistemleri aracılığı ile, hem işletme içi BS'leri kontrol altında tutulabilir (intranet), hem de elektronik ortamın potansiyelini yakalamak için fırsatlar değerlendirilebilir (internet). Bu iki temel yapının tek bir ortamda bütünleştirilmesi, işletme çalışmalarının bir bütün olarak değerlendirilebilmesine olanak tanımaktadır. Bu bütünlük, oluşturulması göreceli olarak zor, ancak performansı yüksek bir sistem entegrasyonu olarak değerlendirilmelidir. İnternetin sunduğu tüm avantajlar extranet sistemleri için de geçerlidir; platform bağımsız, görsel iletişim, maliyetlerde ortaya çıkacak düşüşler, anında erişim ve bilgi paylaşımı.

Verilerin elde edilmesi ve bunların bilgiye dönüştürülmesinde işletme personelinin yetkinliği, pazardaki değişimlere uygun hareket edilmesi ve rekabette avantajlı duruma gelmek için önemli bir yetenektir. Extranet sistemlerinin bu kapsamdaki yeri, veri elde etme ve bunları bilgiye dönüştürüp değerlendirmede, tedarik, üretim ve sunum hizmetlerinin her noktasının bir arada değerlendirilebilmesidir.

3. MALZEME VE YÖNTEM:

3.1. UYGULAMA YÖNTEMİ

3.1.1. Problemin Tanımı

3.1.1.1. Ülkemizde Sağlık Sektörünün Yapısı

Ülkemizde sağlık sektörü 1961 tarih ve 224 sayılı "Sağlık Hizmetlerinin Sosyalleştirilmesi" hakkındaki kanunun öngördüğü biçimde, kademeli hizmet anlayışı çerçevesinde örgütlenmiştir. Örgütlenmenin temel basamaklarını şu şekilde incelemek mümkündür:

- Koruyucu sağlık hizmetleri,
- Tedavi Edici Sağlık Hizmetleri,
    - Birinci Basamak Tedavi Hizmetleri: Hastalıkların ayakta teşhis ve tedavi edilmesi amacıyla hizmet veren kademe olup şu birimlerden oluşmaktadır: Sağlık evleri, sağlık ocakları, kurum tabiplikleri, sağlık istasyonları, ana-çocuk sağlığı merkezleri, poliklinikler, hekim muayenehaneleri, sağlık merkezleri ve dispanserler.
    - İkinci Basamak Tedavi Hizmetleri: Hastaların yatırılarak teşhis ve tedavi edildikleri kurumlardır. İkinci basamak sağlık kuruluşlarından kimileri şunlardır: Devlet hastaneleri, askeri hastaneler, özel hastaneler.
    - Üçüncü Basamak Tedavi Hizmetleri: İleri düzeyde teşhis, bakım ve tedavi gerektiren hastalara hizmet veren kurumlardır. Örnek olarak şu kurumlar sıralanabilir: Özel dal hastaneleri (Onkoloji hastaneleri, sanatoryumlar, kemik hastalıkları hastaneleri vb.), eğitim hastaneleri, numune hastaneleri, üniversite hastaneleri.
- Rehabilite Edici Sağlık Hizmetleri,
- Diğer Sağlık Hizmetleri.

3.1.1.2. Ülkemizde Ağız-Diş Sağlığı Hizmetleri

Ülkemizde Ağız ve Diş Sağlığı Hizmetleri başlıca 4 şekilde verilmektedir;

- Diş Hekimi Muayenehaneleri,
- Müstakil Ağız-Diş Sağlığı Merkezleri,
- Hastaneler bünyesindeki Diş Tedavi ve Protez Merkezleri,
- Diş Hastaneleri.

Sağlık Bakanlığına bağlı olarak verilmekte olan Ağız ve Diş Sağlığı Hizmetleri Mayıs 2005 itibariyle; 2 Diş Hastanesi, 35 Müstakil Ağız ve Diş Sağlığı Merkezi, 101 Hastaneler bünyesinde Diş Tedavi ve Protez Merkezi şeklindedir. Yine Sağlık Bakanlığı verilerine göre Türkiye genelinde yaklaşık 20.000 diş hekimi çalışmaktadır. Bunların %74'ü özel sektörde, %26'sı ise kamuda çalışmaktadır (Kaynak: www.saglik.gov.tr).

Ülkemizde Diş Hekimliği Hastaneleri başlıca üç tipte bulunmaktadır, bunlar:

1. Üniversitelere bağlı olarak kurulu bulunan diş hekimliği fakültesi hastaneleri,
2. Özel dal devlet hastaneleri,
3. Özel sektör diş hastaneleridir.

3.1.1.3. Diş Hekimliği Alanında Bilişim Hizmetleri Kullanımı

YÖK Tez Merkezinde yapılan incelemelerde Diş Hekimliği Hastane Yönetim Bilişim Sistemi konusunda, daha önce yapılmış bir tez çalışmasına rastlanamamıştır. Bununla birlikte Güleş ve Özata'nın (2005), "Sağlık Bilişim Sistemleri" adlı eserlerinde, 'Diş Hekimliği Alanında Bilişim Sistemleri Kullanımı' başlığı altında, BS'nin son dönemde Diş Hekimliği alanında da yaygın olarak kullanılmakta olduğunu ve "Dental Enformatik"in yeni bir alan olarak karşımıza çıktığını ifade ederler. Yine aynı eserde, Bu alanın araştırmalar için temel veritabanı oluşturulması ve dişle ilgili sağlık problemlerinin çözümü için, diş hekimliği ile bilgisayar teknolojisini birleştirdiği ifade edilerek, diş hekimliği alanında kullanılan başlıca BS'leri şu şekilde sıralanmıştır;

- Karar Verme Modelleri: Diş hekimliği alanında daha nitelikli karar alabilmek için KDS'den faydalanılmaktadır. Bu sistemlere örnek olarak da hastaların radyografik ve klinik özelliklerinin değerlendirilmesi ve kemik içi lezyonları ile ilgili tanımlamaları yapmak amacıyla geliştirilmiş ORAD (Oral Radiographic Differential Diagnosis) yazılımı verilebilir.
- Diş Tablolama Sistemleri: Diş hekimliğinde dokümantasyon, tıp alanında olduğu gibi önemli olmakla birlikte, klinik kayıtların tutulmasında kullanılan pictografik

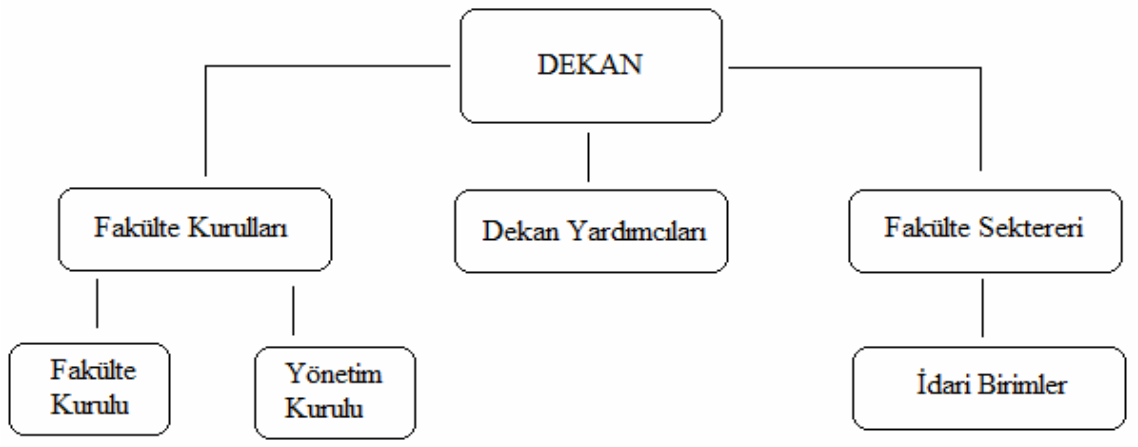
semboller, kodlar ve renkler açısından farklılık göstermektedir. Geleneksel Dental kayıtlar ağız içindeki 32 kalıcı dişin ve süt dişlerinin bilgilerini içerir. Hekim renkli kalemler veya kodlar kullanarak ağız içi (Oral) veya radyografik bulguları bu tabloya işler. Günümüzde ise teknolojideki gelişmeye bağlı olarak birçok diş hekimliği fakültesinde elektronik tablolama sistemlerinden faydalanılır. Bununla birlikte elektronik tablolama ile ilgili uluslararası standartlar olmadığından çok farklı niteliklerde tablolama programlarına rastlamak mümkündür.

- Ağız Kanser Tarama Sistemi: OralCDx olarak adlandırılan bilgisayar temelli bir sistemden ağız lezyonlarının analiz edilmesinde yararlanılmaktadır. Sistem iyi ve kötü huylu lezyonların kesin olarak belirlenmesinde son derece etkili sonuçlar vermektedir.
- Diş Çürüğü Tarama Sistemi: Diş çürüklerinin tespiti, diş hekimlerinin verdiği önleyici sağlık hizmetlerinden birini oluşturmaktadır. Logion Caries Detector olarak adlandırılan bir sistem, diş çürüklerinin tespiti ve tanımlanmasında hekimlere yardım etmek için tasarlanmıştır.

3.1.1.4. Örnek Uygulama İçin Kurum Seçimi

Bu tez çalışmasında üçüncü basamak tedavi kurumları arasında yer alan İstanbul Üniversitesi Diş Hekimliği Fakültesi Hastanesi için örnek bir HYBS geliştirilmesi amaçlanmıştır. İstanbul Üniversitesi Diş Hekimliği Fakültesi 1908 yılında kurulmuş ülkemizin Diş Hekimliği Eğitimi veren ilk ve en büyük fakültesidir.

İstanbul Üniversitesi Diş Hekimliği Fakültesinin idari yapılanması Şekil 3.1'de görülmektedir.

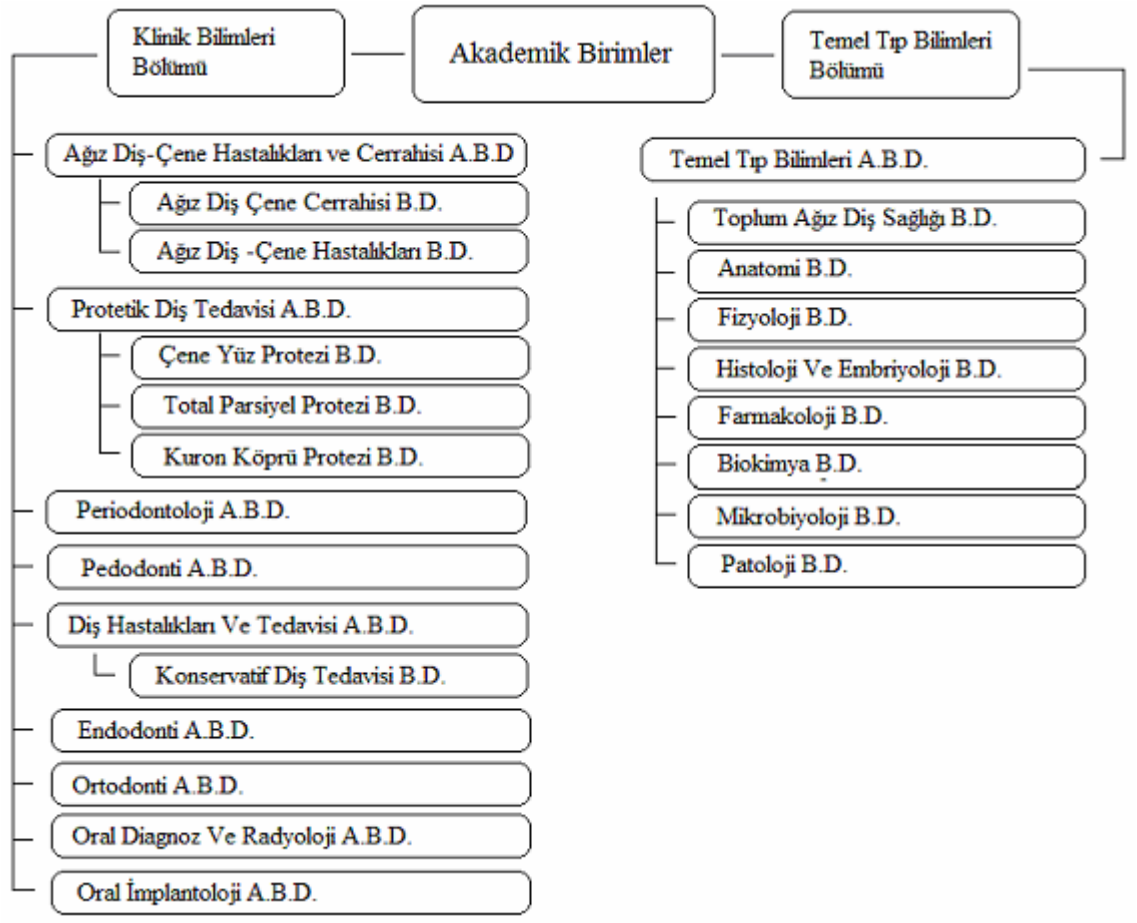


Şekil 3-1: İstanbul Üniversitesi Diş Hekimliği Fakültesi İdari Yapılanması

Fakülte İdari Birimleri şunlardır;

- Öğrenci Bürosu,
- Özlük İşleri Bürosu,
- Özlük Hakları Bürosu
- Satınalma Bürosu,
- Ayniyat ve Levazım Saymanlığı,
- Bakım Onarım Bürosu,
- Evrak Kayıt Bürosu,
- Döner Sermaye Saymanlığı.

İstanbul Üniversitesi Diş Hekimliği Fakültesinin Akademik yapılanması Sekil 3.2'de görülmektedir.



Şekil 3.2: İstanbul Üniversitesi Diş Hekimliği Fakültesi Akademik Yapılanması.

3.1.1.5. İstanbul Üniversitesi Diş Hekimliği Fakültesine Ait Büyüklükler

İstanbul Üniversitesi Diş Hekimliği Hastanesi yataklı hastane statüsünde olmayıp ayakta tedavi kurumudur ve yıllık ortalama 30.000 yeni hasta başvurmaktadır. Bu yönüyle sirkülasyonun çok yüksek olduğu bir hastanedir. Kanımızca bu yoğunlukta bir hastane DHYBS olmaksızın yönetilemez. Hastane verilerinin büyüklüğünü göstermesi adına bazı veriler Tablo-3.1'de sunulmuştur (Kaynak: Fakülte kayıtlarından derlenmiştir).

Tablo 3.1: İstanbul Üniversitesi Diş Hekimliği Fakültesi Veri Büyüklüğü

	Yıllar	
Açıklamalar	**2004**	**2005**
Fakülteye başvuran yeni hasta sayısı	30.274	33.337
Tedavi için bölümlere gönderilen hasta sayısı	126.083	123.802
Basılmış olan nakit ve kredi kartı makbuz sayısı	38.366	50.165
Basılmış olan fatura sayısı	61.921	61.630
Basılmış olan makbuzlara ait makbuz kalemi sayısı	65.485	97.782
Basılmış olan faturalara ait fatura kalemi sayısı	359.108	390.811

Fakülteye başvuran yeni hasta sayısının yıllık yaklaşık olarak 30.000 olduğu ve tedavi için bölümlere gönderilen hasta sayısının yaklaşık 120.000 olduğu göz önüne alındığında her hastanın ortalama olarak tedavisinin 4 bölümde tamamlandığı kabul edilebilir. Genellikle hasta tedavisinin tamamı tek bir bölümde tek bir seansta tamamlanmaz bu sebeple başvuran evraklı hastalar, fakülte tarafından tesbit edilmiş sevk geçerlilik süresi içerisinde istediği bölümde tedavisini olabilmektedir. Sevk geçerlilik süresi SSK ve Bağ-Kur sevkli hastalar için 6 ay, diğer kurum sevklerinde bir aydır.

İ.Ü. Diş Hekimliği Fakültesi Hastanesinde Teşhis ve Tedavi Planlaması uygulaması Tıp Fakültesi Hastanesi yada diğer Hastanelerden farklılık göstermektedir. Diş Hekimliği Fakültesine Kayıt/Kabulü yapılan hastalar, Oral Diagnoz ve Radyoloji ABD tedavi planlaması yapılarak tedavi için ilgili bölümlere yönlendirilmektedir. Bu yönüyle Oral Diagnoz ve Radyoloji ABD genel bir dağıtıcı pozisyonundadır. Yönlendirilen hasta, tedavi planındaki sırada diğer bölümlerdeki tedavilerini yaptırabilmektedir.

Başvuran hastalar Statü Grubuna göre işlem görmektedirler. Tablo 3.2 de görüldüğü gibi Sevkli, Medikolu ve Emekli hastalar nakit ödeme yapmazlar ve yapılan tedavi bedelleri sevk süre sonunda tahsil edilmek üzere fatura düzenlenerek ilgili kuruma gönderilir. SSK ve Bağ-Kur ile Fakültenin anlaşması olmadığı için bu statü türündeki hastalar sevkli geldikleri durumda Resmi Gazete Fiyat tarifesi üzerinden fiyatlandırılarak Nakit yada Kredi Kartı ile Makbuz karşılığında peşin ödeme yaparak Tedavisi tamamlanınca yada 6 aylık sevk süresi bitiminde Makbuzlardaki tedavi kalemlerini içeren fatura döner sermaye birimi tarafından kendileri için düzenlenir ve bu fatura ile birlikte ödeme makbuzlarını sosyal güvenlik kurumlarına beyan ederek tedavi masraflarını geri alabilmektedirler.

Tablo 3.2: İstanbul Üniversitesi Diş Hekimliği Fakültesi Statü Durum Tablosu

Hasta Statü Durumu		Ödeme Şekli	Ücret Türü
Sevkli	(Emekli Sandığı, Çalışan)	Fatura	A Ücret
Medikolu	(Üniversite Mensubu ve Öğrencisi)	Fatura	A Ücret
Emekli	(Emekli Sandığı, Emekli)	Fatura	A Ücret
SSK	(SSK, Çalışan/Emekli)	Makbuz	A Ücret
Bağ-Kur	(Bağ-Kur, Çalışan/Emekli)	Makbuz	A Ücret
Ücretli	(Sosyal Güvencesi Olmayan)	Makbuz	B Ücret
Ücretsiz		-	Gretis

Oral Diagnoz ve Radyoloji Anabilim Dalı tarafından tedavi planlaması yapılarak kliniklere yönlendirilen hasta ilgili kliniklere tedavi planlamasında belirtilen sırada başvurusunu yapar. Kliniklerin çalışma şekli tek tip olmamakla birlikte genellikle başvuran hasta bir ilk muayene sonrası hemen tedavisi yapılabilecek durumda ise yönlendirmesi yapılır yada ileri tarihe randevu verilir. Randevu yapısı ise belirli bir ileri tarihe randevu verilmesi yada hastanın deftere kaydının yapılarak sıra ile defterden çağırma yöntemi uygulanmaktadır.

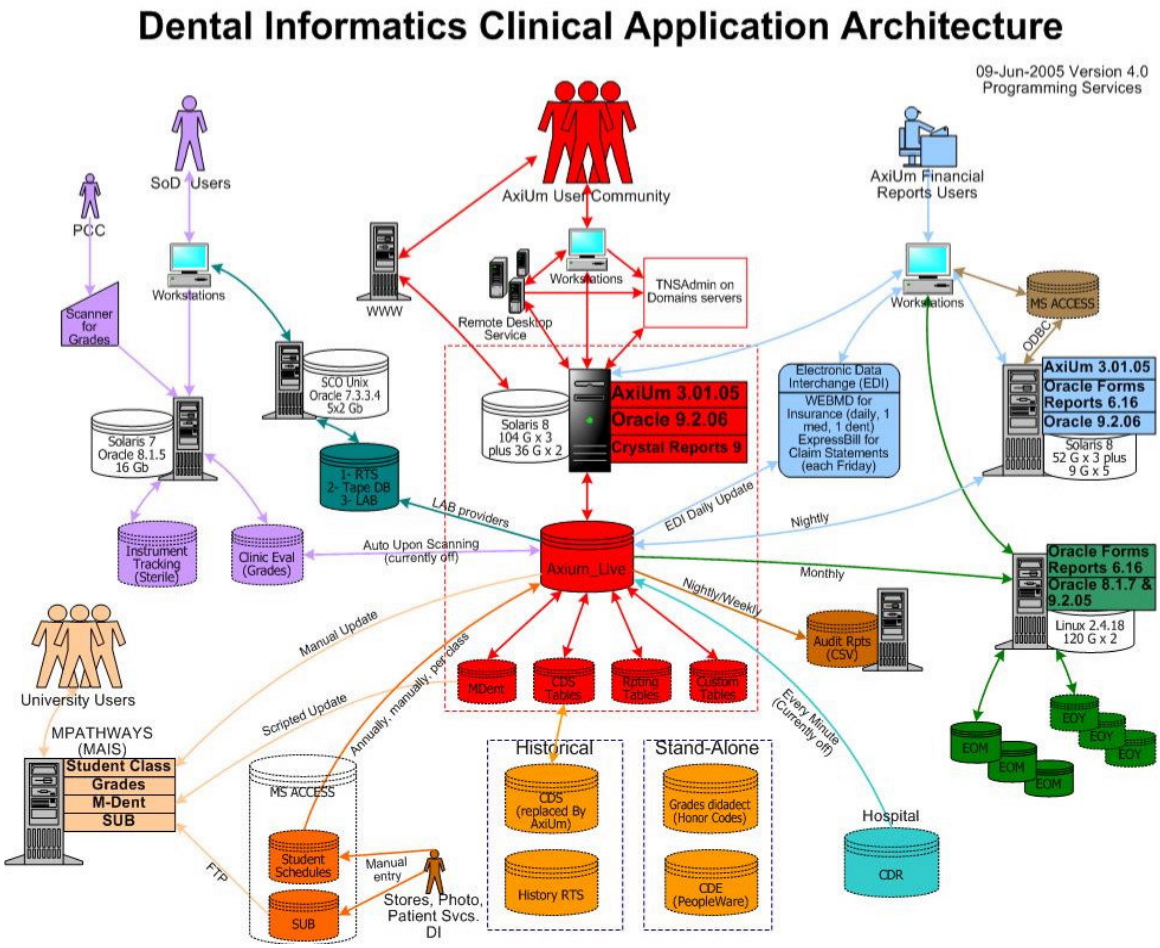
3.1.1.6. Yurtdışı Örnek Uygulama

Yurtdışında Diş Hekimliği Bilişim uygulamalarının incelenmesi internet ortamında yapılmıştır. Bu alanda tesbit edilebilen geniş ve gelişmiş uygulamalardan biri, Amerikan Michigan Üniversitesi Dişçilik Okulunda, “Dental Informatics” adı altında bir bölüm

tarafından Diş Hekimliği Enformatik çalışmaların yürütüldüğünü görmekteyiz. Yapılanmasında üç temel bölüm bulunduğu görülmektedir. Bunlar;

1. Sistem Destek,
2. Elektronik Öğretim Labratuvarı,
3. Programlama Servislerinden oluşmaktadır.

Michigan Üniversitesi Dişçilik Okulu "Diş Hekimliği Enformatik Klinik Uygulama Mimarisi" Şekil 3.3'de görüldüğü gibidir.



Şekil 3.3- Michigan Üniversitesi Dişçilik Okulu "Diş Hekimliği Enformatik Klinik Uygulama Mimarisi"

3.1.2. Analiz

İstanbul Üniversitesi Diş Hekimliği Fakültesi için geliştirilen Programlar iki gurup kullanıcı tarafından kullanılacaktır. Birinci gurupta Fakültede yerleşik kullanıcılar; Oral Diagnoz, Pedodonti ve Protez katlarında olduğu gibi uygulama ve işleyiş açışından

doktorlar tarafından, Hasta Kayıt ve Klinik katlarında olduğu gibi veri kayıt görevlileri tarafından, Vezne Mutemetleri Veznedar ve Döner Sermaye görevlileri tarafından ve nihayet Dekanlıkta Yöneticiler tarafından kullanılmak üzere tasarlanmıştır. İkinci Gurupta ise Uzak kullanıcı olarak Internet ortamında randevu talep eden hastalar tarafından kullanılabilecektir.

Bunlardan birinci gurupta yer alan programlar Borland Delphi programlama diliyle, ikinci gurupta yer alan Randevu Programları ASP ile geliştirilmiştir.

3.1.2.1. Kullanıcı Giriş

Babil DHYBS Yerel Kullanıcıları kimlik doğrulama ve yetki kontrollerinin yapıldığı arayüzdür (Şekil3.4).

İşlev:

- Kullanıcı adı ve şifresi girilir. Çalışma yılı görüntülenen ilk değerden (en son çalışılan yıl görüntülenir) farklı ise girilir.
- Kullanıcı kodu ve şifresi sitemde tanımlı ise ana sayfaya yönlendirilir.
- Kullanıcı adı ve şifresi sistemde tanımlı değil ise uyarı mesajı döndürülür.



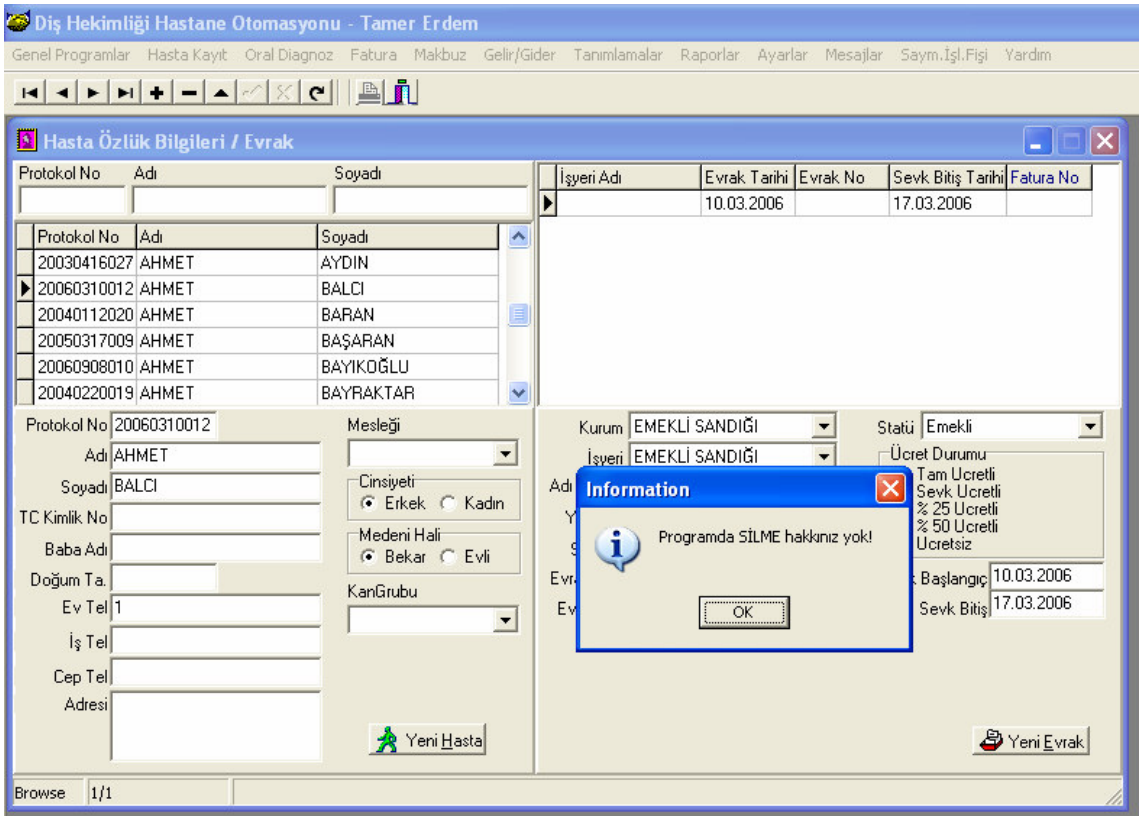
Şekil 3.4 Kullanıcı Giriş Ekranı

3.1.2.2. Ana Sayfa

Kullanıcı doğrulaması yapılarak sisteme giriş yapan kullanıcının önüne gelen ilk ekrandır.

İşlev:

- Ana sayfada kullanıcı yetki durumuna göre menü hakkı ayarlaması yapılarak, yetkisi olmayan menü başlıkları sönük görüntülenecek yada işlem tipine göre işlem yapmaya yetkisi olmadığı şeklinde uyarı ekranı görüntülenecektir (Şekil 3.5).
- Ana sayfa bir tepe menü ve programlar tarafından ortak kullanılan bir scroll bar içermektedir. Tepe menüye tıklanarak alt menülere ulaşmak mümkündür.



Şekil 3.5 Yetkisiz Kullanıcı Uyarı Mesajı

3.1.2.3. Genel Programlar

Diğer Programlar tarafından kullanılacak kodlu bilgilerin yer aldığı tepe menü adıdır. Bu bölümde yer alan programlar şunlardır;

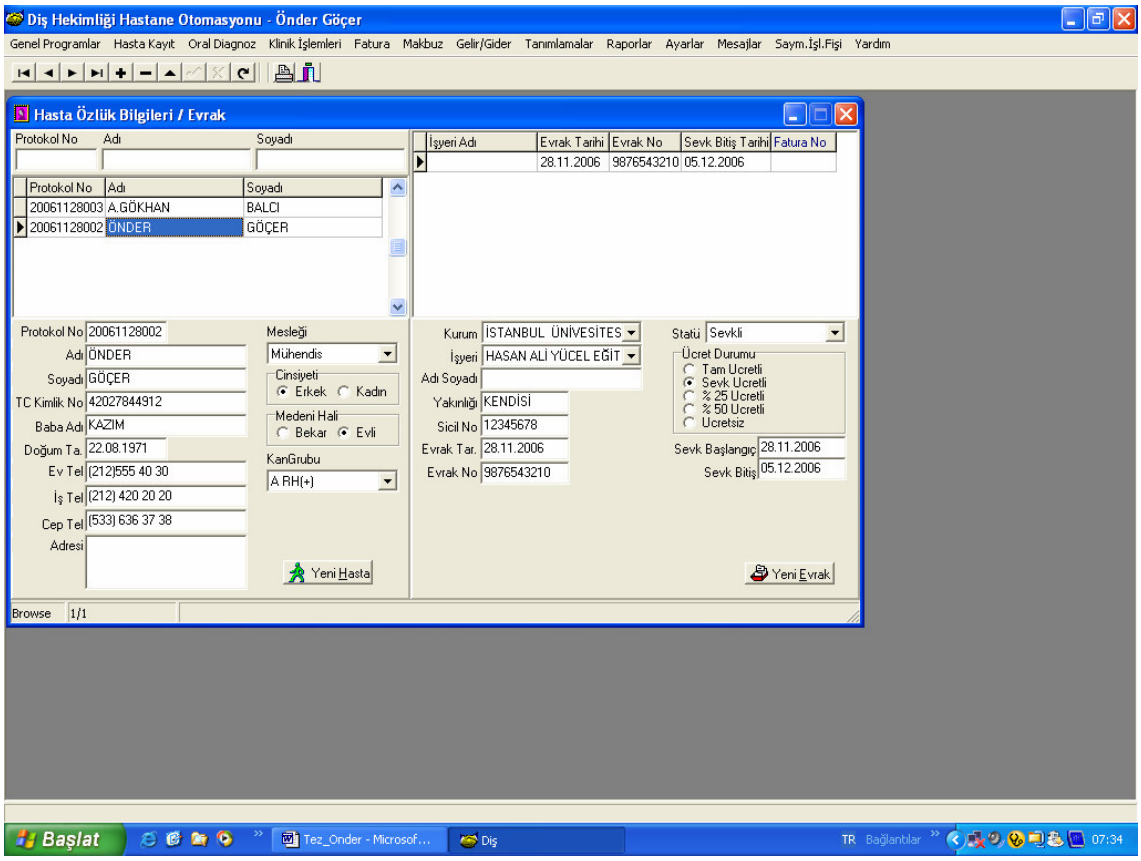
- İl ve İlçe Tanımları
- Meslek Tanımları
- Kan Grupları
- Unvan Tanımları
- Statü Tanımları
- Anamnez Kodları

- Kürsü Tanımları
- Yapılan Tanımları
- Paket Tanımları
- Kurum ve İşyeri Tanımları
- Kurum Grup Tanımları.

Tanımlanmış olan bu bilgiler, diğer programlarda veri girişi sırasında tekrar eden verilerin listeden seçilerek (ListBox ve Combobox şeklinde) kolayca ve hatasız girilmesinin yanı sıra sorgu ve sıralama imkanı da sunacaktır.

3.1.3.4. Hasta Kayıt

Fakülteye başvuran hastanın özlük bilgilerinin ve evrak (statü) bilgilerinin girildiği bölümdür. Hastanın fakültedeki tüm işlemlerinde burada girilen bilgilerin temel oluşturacağı göz önünde bulundurulduğunda verilerin tam ve eksiksiz girilmesinin önemi ortaya çıkmaktadır. Hasta Kayıt Ekranı görünümü Şekil 3-6 da görülmektedir.

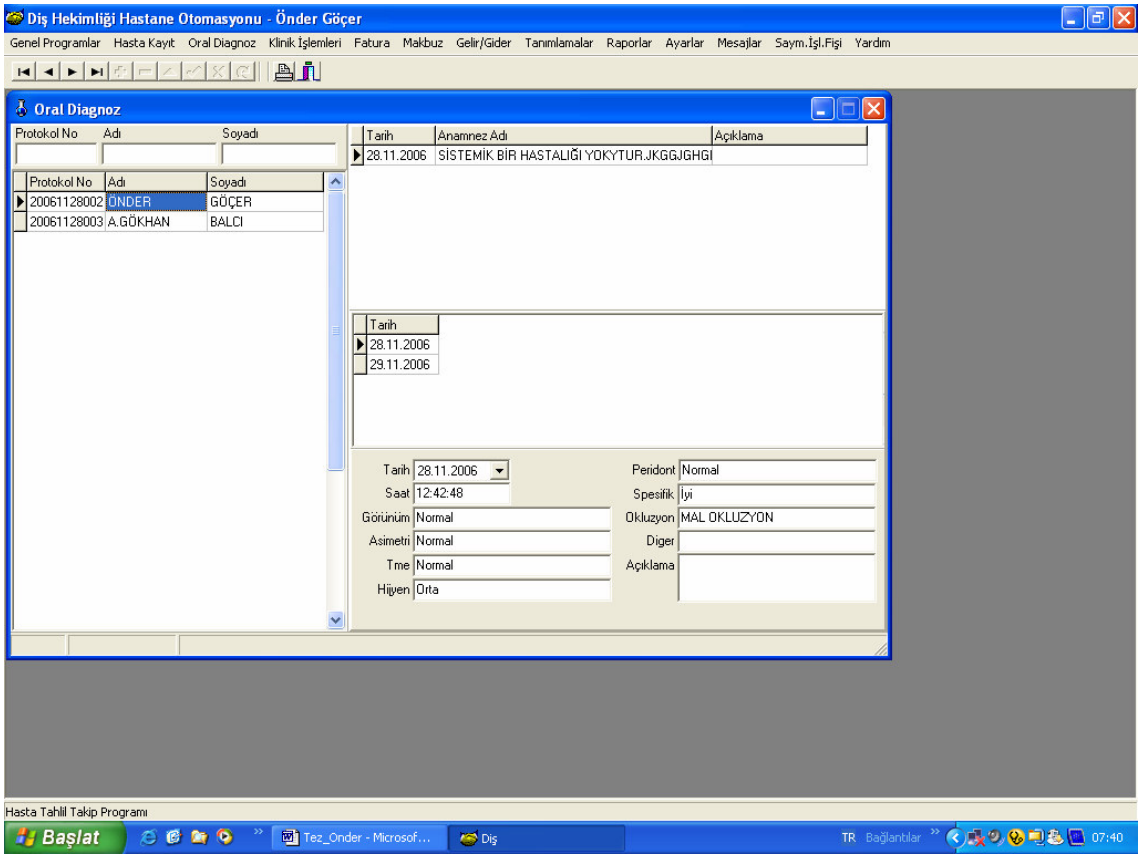


Şekil 3.6 Hasta Kayıt Program Görünümü

Hasta Evrak Bilgileri girişi de bu bölümden yapılmaktadır. Evrak bilgisi girilen hastaya Muayene ücreti otomatik olarak tahakkuk ettirilmektedir. Statü türü ücretli olan hastalar Makbuz karşılığında Nakit yada Kredi kartı ile ödeme yapmakta iken, Evraklı hastalar için ödeme evrakı olarak fatura kesilerek kurumlarından tahsil edilmesi yoluna gidilmektedir.

3.1.2.5. Oral Diagnoz

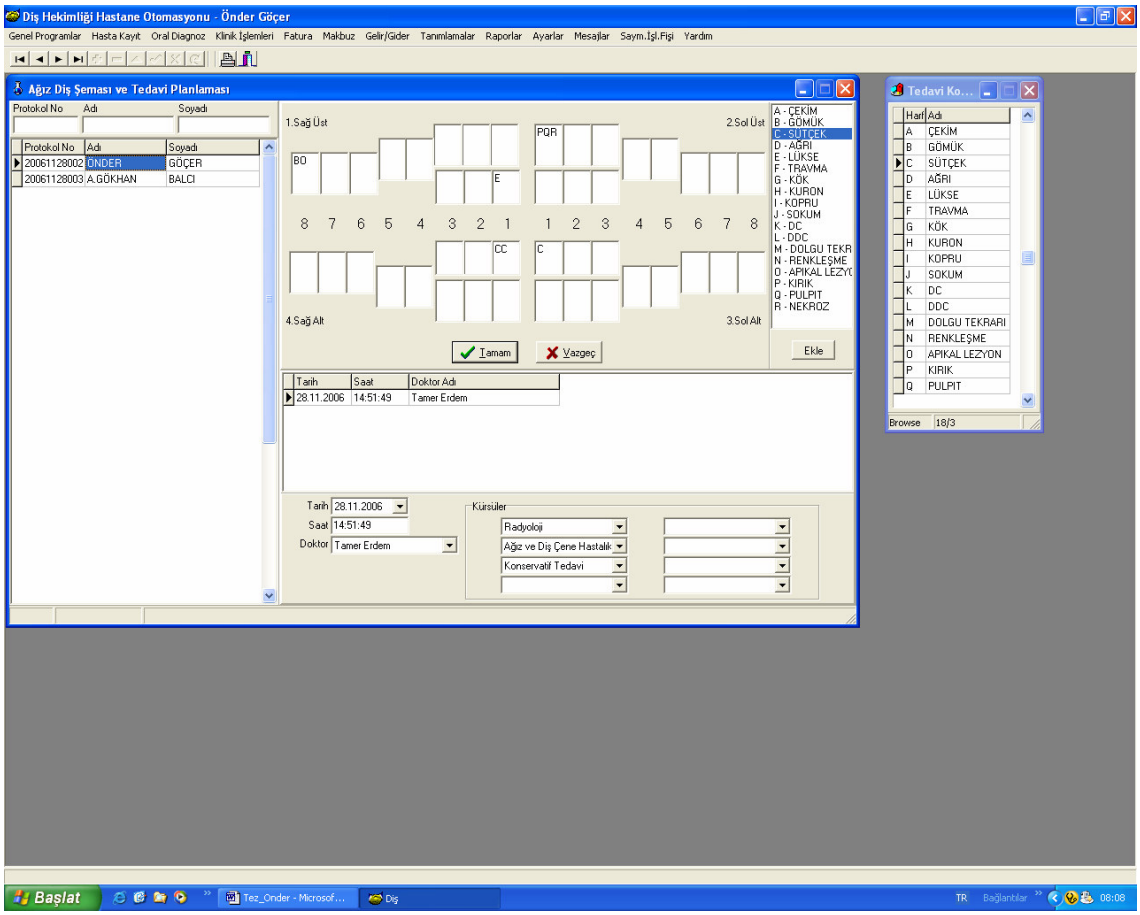
Oral Diagnoz ve Radyoloji A.B.D. tarafından kullanılan Hasta Muayene bilgilerinin girişinin yapılabildiği programlardır. Şekil 3-7 de görüldüğü gibi ekranın sol bölümünden istenilen hasta bilgisine kolayca ulaşılabilmekte ve bu hasta için Anamnez ve Oral Görünüm bilgileri girilebilmektedir.



Şekil 3.7 Oral Diagnoz Anamnez ve Oral Görünüm Bilgileri Giriş Ekranı

Tepe menüde Oral Diagnoz seçeneği altından bulunan Tedavi Planlaması seçilerek Ağız ve Diş Şeması ve Tedavi Planlaması ekranında dişler için uygulanacak işlemler belirtilerek tedavi planlaması da yapılabilmektedir. Yine bu bölümde yer alan Tedavi

Kodları programı vasıtasıyla kodlu bilgi şeklinde oluşturulan kodlar, dinamik olarak Diş Şeması programı tarafından kullanılabilmektedir. Şekil 3.8'de örnek Ağız Diş Şeması ve Tedavi Planlaması Ekranı ile birlikte Tedavi Kodları Giriş Ekranı görülmektedir. Ağız diş şemasına en çok giriş işlemi yapılacağı için ilk değer olarak 'Ekle' şeklinde görülen düğme tıklandığında etiketi değişerek 'Çıkar' ve 'Sil' seçenekleri görülmektedir. Bu yöntemle fonksiyonlar eksiksiz olarak yerine getirilirken giriş ekranında sadelik sağlanması amaçlanmıştır. Bu ekranda Diş Şeması üzerinde tedavi bilgisi girilirken (ekleme modunda), istenirse tedavi kodu olarak harfler tuşlanabileceği gibi ekranın sağında bulunan kodlar da tıklanarak giriş yapılabilir.



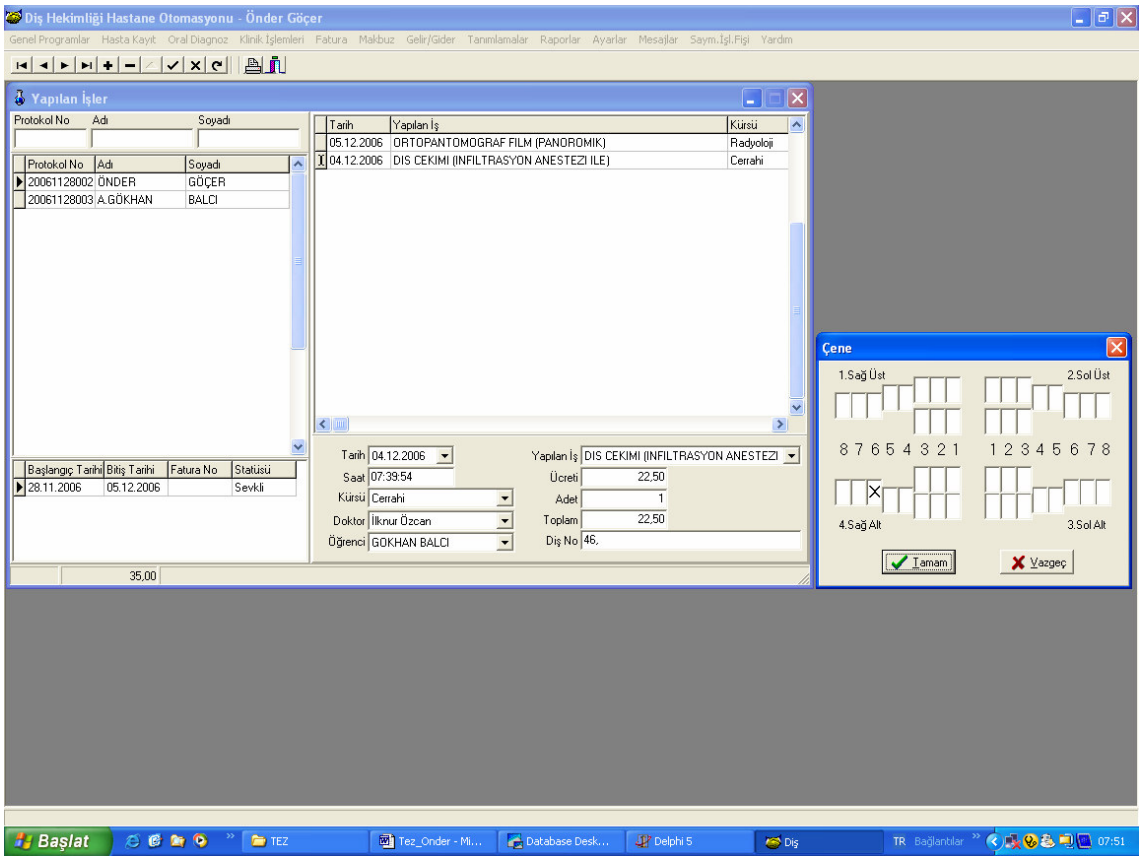
Şekil 3.8 Oral Diagnoz-Ağız Diş Şeması ve Tedavi Planlaması ile Tedavi Kodları Ekranı

3.1.2.6. Klinik Takip

Klinik Bilgilerin giriş ve takibinin yapılabildiği programlardır.

İşlev:

- Şekil 3-9 de görüldüğü gibi, öncelikle ekranın sol bölümünden istenilen hasta bilgisine ulaşılır.
- Daha sonra kullanıcı Tarih, Saat, Kürsü, Doktor/Öğrenci bilgilerini girer.
- Yapılan iş seçilerek girilir.
- Yapılan işin ücreti hasta statü durumuna göre dosyadan görüntülenir.
- Diş Numarası girişi için DişNo alanına çift tıklandığında Şekil 3-9 da görüldüğü gibi Diş Numaraları Ekranı açılır. Bu ekranda Mouse ile tıklayarak tedavinin uygulandığı diş seçilir. Tamam Düğmesine basılarak giriş tamamlanır.



Şekil 3.9 Klinik Takip Ekranı ve Diş Şeması

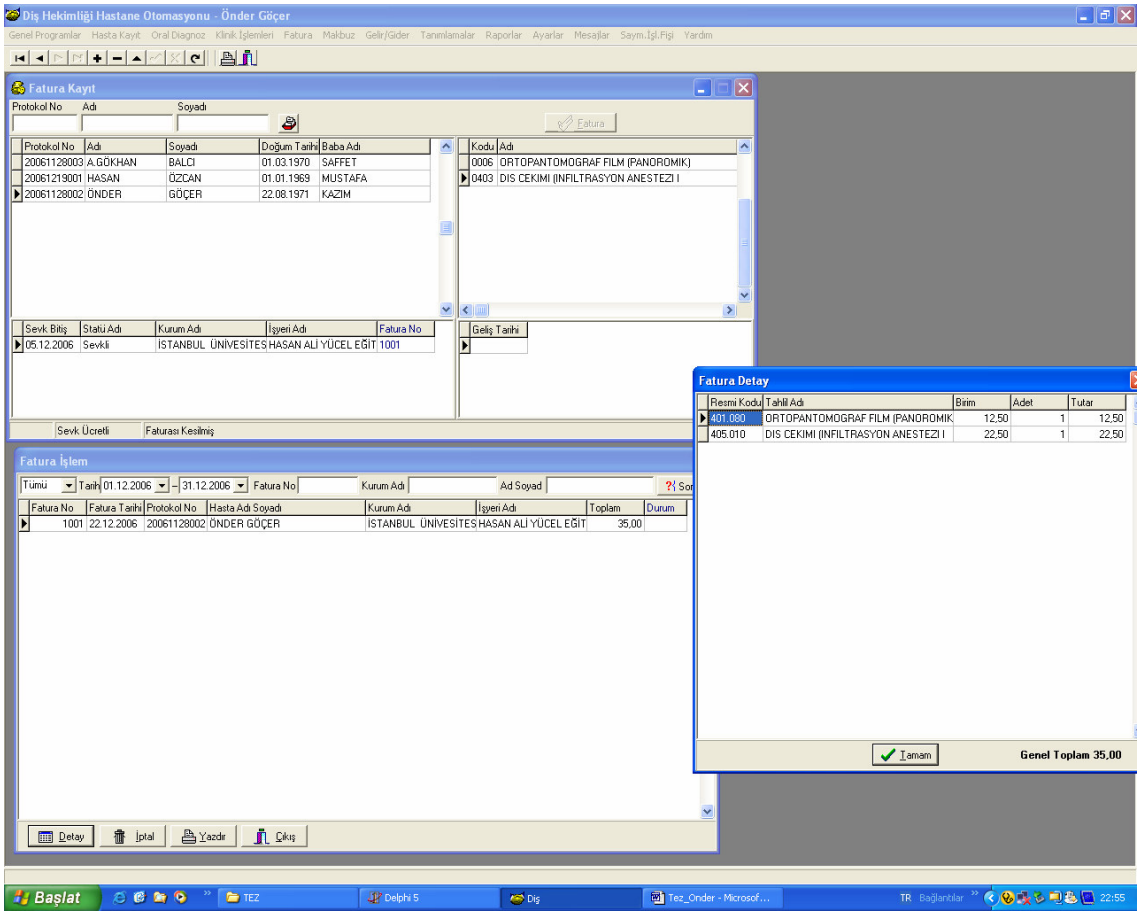
3.1.2.7. Fatura İşlemleri

Fatura ve Ödeme Takip işlemlerinin yapılabildiği programlardır.

İşlev:

- Kayıt Programı yardımıyla, Faturası kesilebilecek durumda olan hastaların bilgilerini görüntülemek ve fatura oluşturmak mümkündür.

- Fatura İşlem Programı yardımıyla daha önce faturası oluşturulmuş hastaların fatura detaylarını incelemek, tekrar basmak ya da basılmış bir faturayı iptal etmek mümkündür.
- Fatura Ödeme Menü seçeneği yardımıyla daha önce faturası oluşturulmuş faturaların ödeme dekont numaralarını ve tarihlerini girerek ödeme takibi yapılması mümkündür. Ödeme kaydı yapılmış bir faturanın ödeme kaydının iptali de yine bu bölümde yapılabilmektedir.



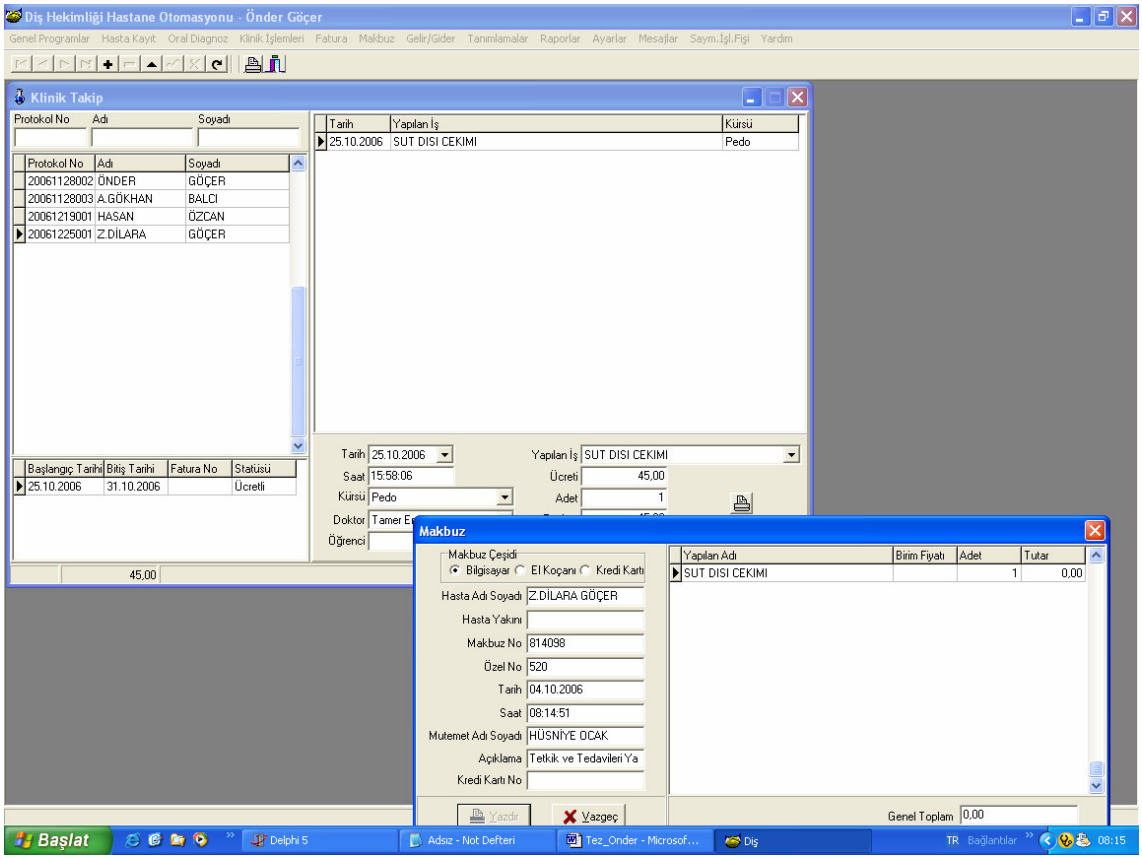
Şekil 3.10 Fatura Kayıt, Fatura İşlem ve Fatura Detay Ekranları

3.1.2.8. Makbuz Basımı

Makbuz basımı klinik veri girişi sonrasında gerçekleştirilen bir işlemdir. Kullanıcı ve hasta açısından daha kolay ve verimli bir sistem olduğu göz önünde tutularak makbuz basma işlemi Klinik Veri Giriş Ekranındaki Makbuz Düğmesi vasıtasıyla açılan Makbuz Ekranından yapılabilmektedir (Şekil3.11).

İşlev:

- Makbuz Türü Radyo Düğmeleri vasıtasıyla seçilir. Makbuz Türleri şu üç şekilde olabilir;
    - Bilgisayardan Yazıcı vasıtasıyla basılan Nakit Ödeme Makbuzu,
    - Bilgisayar sisteminde meydana gelen teknik bir arıza yada diğer gerekçelerle kullanılabilen el koçanı (ende olarak ihtiyaç duyulur),
    - Kredi Kartı Makbuzu.
- Hasta Adı, Makbuz Numarası, Özel Numara, Tarih, Saat, Mutemet, Yapılan Tedavi Adı, Birim Fiyatı, Adedi ve Tutarı dosyadan otomatik olarak görüntülenir.
- Ödemeyi yapan kişinin yakınlığı bilgisi zorunlu olarak mutlaka girilmelidir. Bilgiler tam olduğu durumda Yazdır Düğmesi aktif olmaktadır.



Şekil 3.11 Klinik Takip Ekranında Makbuz Basımı

3.1.2.9. Makbuz İşlemleri

Makbuz Basımı dışındaki her türlü mutemetlik işlemleri Makbuz Tepe mönüsü altında yer almaktadır.

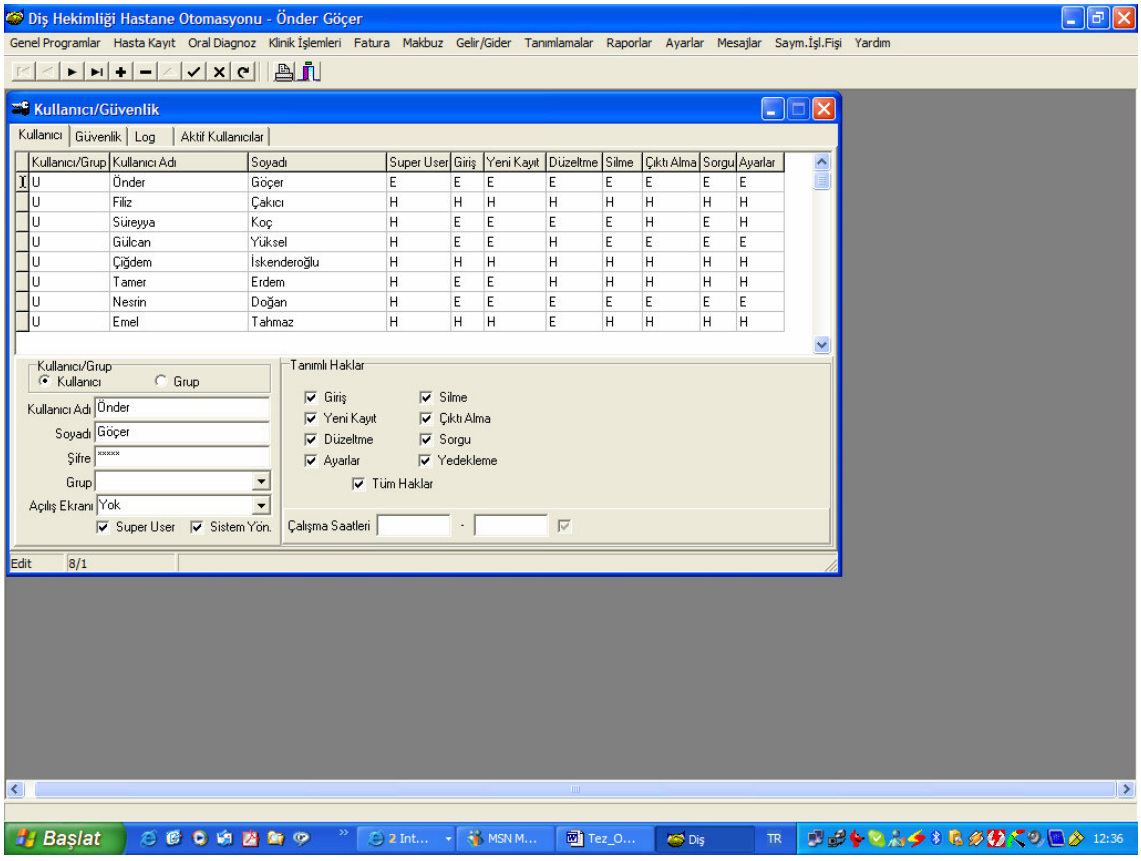
İşlev:

- Sayman Mutemedi İşlem seçeneği ile basılmış makbuzlar sorgulanabilir, tekrar yazdırılabilir, iptal ve iade işlemleri yapılabilir, SSK'lı ve Bağ-Kur'lu hastalar için fatura dökümü yapılabilir.
- Vezne Alındısı Kesme seçeneği ile Şartname Bedeli, Teminatlar ve diğer gelir türlerine göre tahsilat ve makbuz kesme işlemi yapılabilir.
- Vezne Alındısı İşlem Seçeneği ile kesilmiş bir alındının görüntülenmesi, tekrar basılması, iadesi, iptali, ve faturalanması mümkündür.
- Vezne Alındısı Açıklaması bölümünde gelir türleri ve bu gelir türlerine ait muhasebe hesap kodları tanımlanır.

3.1.2.10. Tanımlamalar

Bu tepe mönü başlığı altında yer alan programlar aşağıda listelenmiştir.

- Kullanıcı Tanımı; yeni kullanıcı, grup ve yetki tanımlama ve değişiklik işlemlerinin yapılmasını sağlayan programdır (Şekil 3.12). Her kullanıcı üyesi olduğu grubun haklarına da sahiptir. Süper Kullanıcı hakkı verilen kullanıcılar tüm haklara sahip olurlar. İstenirse her kullanıcıya bir açılış ekranı atanabilir, bu durumda programı açan kullanıcının kendi çalışma ekranına yönlenmesi sağlanır. Kullanıcıların çalışma saatleri için kısıtlama koymak mümkündür.
- Şifre Değişikliği; bu program kullanıcının kendi şifresini değiştirmesine olanak sağlar. Bu işlev için öncelikle eski şifrenin girilmesi gerekmektedir.
- Şirket Tanımı; Şirket bilgilerinin ve yetkililerin tanımlandığı bölümdür.
- Personel Tanımı, İdari ve akademik personelin tanımlandığı bölümdür.
- Öğrenci Tanımı; Öğrenci adı, soyadı ve Numarasının girildiği bölümdür.
- Tatil Günü Tanımlamaları bu tepe menü başlığı altında yer almaktadır.



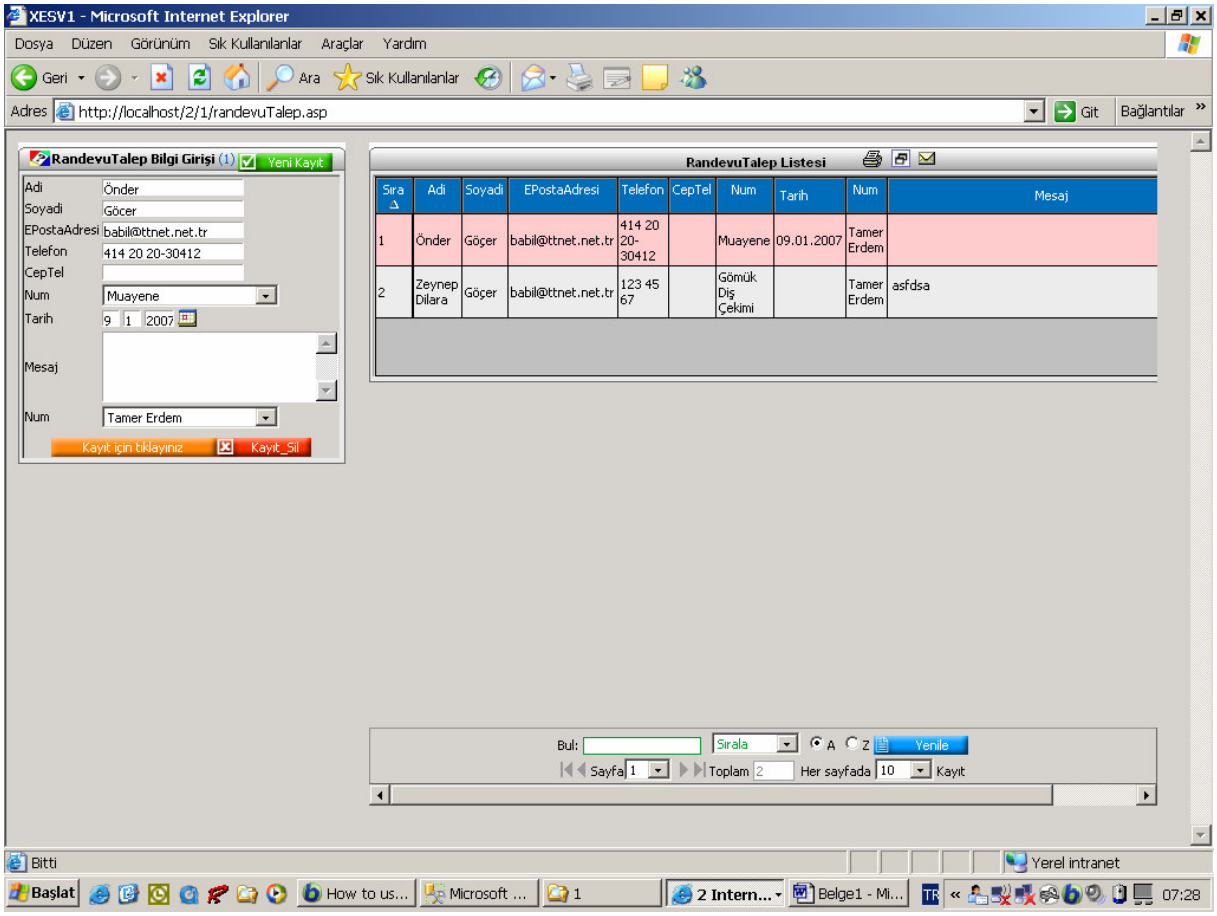
Şekil 3.12 Kullanıcı Tanımlama Ekranı

3.1.2.11. Uzaktan Randevu Takip

Hasta ve Doktorların uzak kullanıcı olarak sistemi kullanmalarına olanak sağlamaya yönelik olarak tasarlanmıştır. URT iki temel programdan oluşmaktadır bunlar Randevu Talep ve Randevu Bilgi Giriş programlarıdır (Şekil 3.13 ve Şekil 3.14).

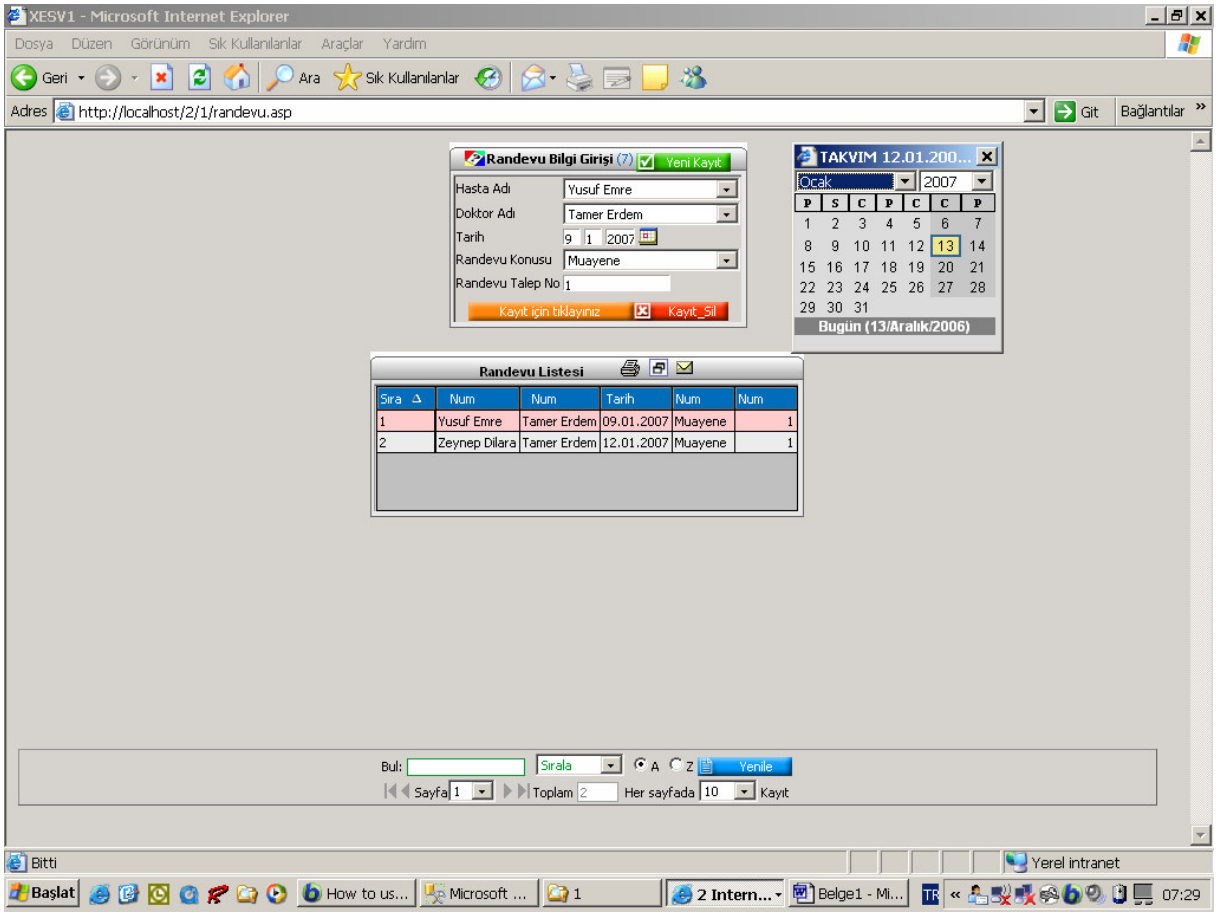
İşlev:

- Randevu Talep Programı fakülteye ilk defa başvuracak hastaların kullanımı için tasarlanmıştır. Hasta adı, soyadı, e-posta adresi, telefon numaraları, başvuru gerekçesi, tarih, açıklayıcı ek bilgiler ve isterse doktor ismi girerek randevu talebinde bulunur. Bu kullanıcı ekranı için Listeleme ve Arama fonksiyonları da tanımlanmıştır. Uygulama anında bu bilgilerin başka kullanıcılar tarafından görülmesinin sakıncalı bulunması durumunda sorgulama işlevinin iptal edilmesi mümkündür. Yeni kayıt, değişiklik ve kayıt silme işlevleri ilgili düğmeler vasıtasıyla kolaylıkla yapılabilmektedir.



Şekil 3.13 Randevu Talep Giriş Ekranı

- Randevu Bilgi Girişi; Fakültede kayıtlı hastaların randevu taleplerini girmelerini sağlamak için tasarlanmıştır. Hasta ismini Listbox'tan seçerek girebilmekte, Doktor, Tarih ve Randevu konusunu belirterek randevu talebi girebilmektedir. Bu kullanıcı ekranı için Listeleme ve Arama fonksiyonları da tanımlanmıştır. Uygulama anında bu bilgilerin başka kullanıcılar tarafından görülmesinin sakıncalı bulunması durumunda sorgulama işlevinin iptal edilmesi mümkündür. Yeni kayıt, değişiklik ve kayıt silme işlevleri ilgili düğmeler vasıtasıyla kolaylıkla yapılabilmektedir. Tarih girişi için Takvim pencere ekranında seçerek girme mümkündür.



Şekil 3.14 Randevu Bilgi Giriş Ekranı

3.1.3. Tasarım

DHYBS iki faklı yapıda tasarlanmış bunlardan birincisi Masaüstü Uygulaması şeklinde yerel kullanıcı programları, diğeri ise uzak kullanıcılar (hastalar) için hazırlanmış bulunan Randevu Takip Sistemidir.

Tasarlanan DHYBS kapsamı şu şekilde belirlenmiştir;

1. Evrak Kayıt Bürosu – Evrak Takip Programı,

2. Maaş Bürosu – Bordro, Personel Devam Kontrol Sistemi,

3. Özlük Hakları Bürosu – Doküman Destek Sistemi,

4. Bakım Programı,

5. Satınalma; Bütçe, Sipariş Takip, İrsaliye-Fatura,

6. Ayniyat; Malzeme Takip, Demirbaş Takip, Makbuz İstem,

7. Öğrenci Bürosu, Öğrenci Not Takip Programı,

8. Hasta Takip Sistemi

a. Hasta Kayıt/Kabul,
b. Evrak Kayıt (Statü takip)
c. Oral Diagnoz ve Radyoloji;
    i. Anamnez Takip,
    ii. Oral Takip,
    iii. Ağız-Diş Şeması ve Tedavi Planlaması,
    iv. Radyoloji,
d. Cerrahi Klinik Takip,
e. Kuron Köprü Protez,
f. Total Parsiyel Protez,
g. Pedodonti,
h. Periodontoloji,
i. Konservatif Tedavi,
j. Endodonti,
k. Ortodonti,
l. Ağız Hastalıkları,
m. Çene-Yüz Protezi,
n. Oral İmplantoloji,
o. Temel Tıp Bilimleri;
    i. Anatomi,
    ii. Mikrobiyoloji,
    iii. Toplum Ağız-Dis Sağ.,
    iv. Biokimya,
    v. Fizyoloji,
    vi. Histoloji,
p. Döner Sermaye;
    i. Fatura Takip,
    ii. Tahsilat Takip,
    iii. Vezne Takip,
    iv. Muhasebe,
r. Kodlu Bilgiler ve Tanımlamalar,
s. Raporlar ve Listeler.

3.1.4. Kod Geliştirme

Kod geliştirme aşamasında kullanılan ürünler Malzeme başlığı altında ayrıntılı olarak ele alınmıştır.
Masaüstü uygulama programı kaynak kodları Ek-C'de sunulmuştur.
URT uygulama programı kaynak kodları Ek-D'de sunulmuştur.

3.1.5. Veri Tabanı

DHYBS veritabanı ilişkisel olarak tasarlanmıştır. Her bir tablonun kimlik alanı (Num) mevcuttur ve diğer tablolar ile arasındaki ilişkiler bu alanlar üzerinden sağlanır. Tablo yapıları ve ilişki diyagramı Ek-E'de sunulmuştur.

3.2. MALZEME

3.2.1. Kodlama Dili

Masaüstü uygulaması için yazılım geliştirme dili olarak Borland Delphi 7 sürümü tercih edilmiştir. Delphi güçlü veri yapısı, görsel ve hızlı program geliştirme ortamı, veritabanları ile yüksek performanslı ve güvenli çalışma ortamı sunması sebebiyle tercih edilmiştir.

Uzaktan Randevu Takip Programları ise ASP ortamında geliştirilmiştir. Güçlü bir dil olması ve kolay kaynak temin edilebilmesi sebebiyle tercih edilmiştir.

3.2.2. Veri Tabanı

Veritabanı olarak, kolay kaynak temini, yaygınlığı ve güncelliği sebebiyle Microsoft SQL Server 2005 sürümü tercih edilmiştir.

3.2.3. İşletim Sistemi

Sistemin geliştirilmesi sırasında platform olarak Windows XP işletim sistemi kullanılmıştır. Web servisi uygulama sunucusu için, Microsoft Internet Information Services (IIS) tercih edilmiştir.

4. BULGULAR

Sağlık Bilişim Sistemleri konusunda çalışırken bilinmesi gerekli bir takım standartlar vardır. Bunlar;

a. Uygulanabilecek Teşhis ve Tedavi Kalemleri Maliye Bakanlığı tarafından Bütçe Uygulama Talimatı şeklinde Tebliğ olarak yayınlanmaktadır. Listede Diş Hekimliği ile ilgili kalemleri içeren bölüm "Ek-7" başlığı altında yer almaktadır (Ek-A). Diş Hekimliği Teşhis Tedavi Listesinde Bulunmayan Tedaviler için Tıp Fakültesi listesi kullanılabilmektedir. www.bumko.gov.tr adresinden Mevzuat / Bütçe Uygulama Talimatları-Tebliğler bölümünden ulaşılabilir.

b. Uluslararası Hastalık Tanı Kodları, ICD-10 Tanı kodları 01.07.2005 tarihi itibarıyle fatura üzerinde yer almak zorundadır (2005 Yılı Bütçe Uygulama Talimatı Sıra No:4 (Tedavi Yardımı) Madde 21.1. g bendi). Tanı kodları Sağlık Bakanlığının http://www.saglik.gov.tr/icd10/default.asp internet adresinden indirilebilir. Diş Hekimliğinde sıklıkla kullanılan ICD-10 tanı kodları listesi Ek-B de sunulmuştur.

c. Emekli Sandığı On-Line bağlantı için bir çalışma içerisindedir. Hastaya uygulanmış tedavilerin takibi ve tekrar uygulama için bekleme süre kontrollerinde bu çalışmanın işlevsel bir görev üsteleneceği açıktır. Bununla birlikte çalışma henüz uygulamaya geçmemiştir.

d. İstanbul Üniversitesi Diş Hekimliği Fakültesi SSK ve Bağ-Kur sevkli hastalarından nakit ücret almak ve fatura kesmek suretiyle hizmet vermektedir. Bu sebepten dolayı şimdilik o kurumların faturalandırma standartları talep edilmemektedir.

Yapılan çalışma da ne kadar iyi bir analiz yapılmış olursa olsun, sistemin güçlü ve zayıf yönleri uygulama devreye alındığı zaman görülebilecektir. Tasarlanan DHYBS çok kapsamlı bir uygulamadır. Bu tez çalışması içerisinde Hasta Takip Sistemi üzerinde

durulmuş ve özellikle Hasta Kayıt, Oral Diagnoz ve Tedavi Planlaması, Randevu Takip Sistemleri üzerinde çalışılmıştır. Bu bölümlerin seçilme sebebi ise hastanın fakülteye ilk gelişte karşı karşıya kaldığı bölümler olmasıdır.

5. TARTIŞMA VE SONUÇ

Yapılabilecek uygulamaların hayal gücü ile sınırlı olduğu günümüzde yapılmış olanlar yapılabileceklerin yanında gerçekten çok az yer tutmaktadır. Dijital görüntüler üzerinden ortodonti ve Endodonti amaçlı ölçümler yapabilen entegre bir sistem kurulması dijital radyografiye geçiş ile birlikte verimlilik ve ekonomi sağlayacaktır. Üçboyutlu simülasyon programları yardımıyla hastaların bilgilendirilmesi veya öğrencinin eğitiminde yardımcı materyal olarak kullanılması da mümkündür.

Ülkemizde 20 tane Diş Hekimliği Fakültesi bulunmasına rağmen, bu güne kadar DHYBS konusunda yapılmış bir tez çalışmasının bulunmadığı tespit edilmiştir.

Bir organizasyonun amaçlarına ulaşması, iç ve dış çevredeki değişimlere karşı sağladığı uyum ve ortaya çıkan değişimler karşısında yönetimin gereksinim duyduğu doğru, zamanlı ve anlamlı bilgileri sağlayarak, etkin karar alma faaliyetini gerçekleştirmesiyle yakından ve doğrudan ilgili olduğu düşünülürse, İstanbul Üniversitesi Diş Hekimliği Fakültesi içinde gelişmiş DHYBS'nin kurulmasının taşıdığı önem daha iyi anlaşılacaktır.

Yazılım geliştirme alanında, dışarıdan hizmet alınmasının verimliliği ve sürdürülebilirliği günümüzde tartışılmaya devam eden bir kavramdır. Bu kavram ve yöntem üzerinde durulmasının temel nedeni ise bilginin stratejik önemi ve maliyetlerdir. Bununla birlikte ülkemizde kamunun iyi yetişmiş elemanların istihdamını devam ettirmede karşılaştığı sorunlar da göz önünde tutularak sürdürülebilir bir çözümün üretilebileceği düşünülmektedir.

Hiç şüphe yok ki diş hekimliği hastanelerinde de sunulan sağlık hizmetinin en iyi şekilde verilebilmesi, gelir ve giderlerin izlenebilmesi, gelir kaçaklarının önlenmesi, kaynakların doğru yönlendirilmesi, hastane yönetimine stratejik kararlar vermek üzere bilgi sağlanması amaçlarıyla DHYBS'nin kurulmasına ihtiyaç vardır. Bu arada

unutmamak gerekir ki teknolojik yeniliklerin verimli bir şekilde uygulanması, iyi eğitilmiş ve sisteme inanan bir ekiple uygulanması başarıyı getirecektir aksi ise zaman kaybı, mali yük ve verimsizlikle sonuçlanabilecektir.

İstanbul Üniversitesi Diş Hekimliği Fakültesi oluşturulan DHYBS sayesinde daha verimli yönetilebilecek, kalite ve verimlilik artışıyla hizmet alan ve veren açısından memnuniyet oluşması sağlanacaktır.

Bu çalışma sırasında görülmüştür ki “Dental Enformatik” enformatik bilimi açısından doğru seçilmiş bir çalışma dalıdır. Gerek yurtdışındaki çalışmaların araştırılmasında gerekse bu tez çalışması yürütülürken, yapılabilecekleri bir yüksek lisans tez çalışmasına sığdırmanın mümkün olmadığı, uzun süreler alacağı ve bir ekip çalışmasıyla başarıya ulaşılabileceği tespit edilmiştir.

Hiç şüphe yok ki ülkemizde de “Dental Enformatik” alanında yapılabilecekler için fazla gecikilmeyecektir. Bu aşamada konunun genişliği de göz önünde tutularak çalışma alanları şu başlıklar altında incelenebilir:

- Hasta Kayıt Sistemleri,
- Dijital Radyografi Ölçümleme ve Tanı Sistemleri,
- Teşhis ve Tedaviye Yardımcı Karar Destek Sistemleri,
- Tedavi Süreç Takip ve Simülasyon Sistemleri,
- E-Learning yöntemlerinin diş hekimliğinde öğrenci ve hasta eğitimi için kullanılması.

KAYNAKLAR

1. Ağız ve Diş Sağlığı Hizmetleri [online], T.C. Sağlık Bakanlığı, http://www.saglik.gov.tr/default.asp?sayfa=birimler&sinifi=hizmet&sid=168&cid=2, [Ziyaret Tarihi: 03.10.2006]

2. Bütçe Uygulama Talimatı [online], T.C. Maliye Bakanlığı Bütçe ve Mali Kontrol Genel Müdürlüğü http://www.bumko.gov.tr/mevzuat/Butcetalimatlari/2006buttalimatlari/2006buttalimatlari.htm

   http://www.bumko.gov.tr/mevzuat/Butcetalimatlari/2006buttalimatlari/SiraNo6/EK7/ek7.htm
   [Ziyaret Tarihi: 03.10.2006]

3. CANTU, M., 2003, Masterin Delphi 7, Alfa Basım Yayım Dağıtım Ltd.Şti. İstanbul, ISBN:975-297-339-6.

4. CANTU, M., 2000, Delphi 5 Uygulama Geliştirme Klavuzu, Alfa Basım Yayım Dağıtım Ltd.Şti. İstanbul, ISBN:975-316-327-4.

5. ÇİMEN, M., 1994, Yönetim Bilgi Sistemlerinin İncelenmesi ve Zübeyde Hanım Doğumevi Eczanesinde Modüler Hastane Yönetim Bilgi Tasarımına Ait Bir Çalışma, Bilim Uzmanlığı Tezi, Hacettepe Üniversitesi Sağlık Bilimleri Enstitüsü.

6. ÇİMEN, M. ve ATEŞ, M., 2006, Yönetim Bilgi Sistemleri[online], http://www.merih.net/m1/wmetate23.htm [Ziyaret Tarihi: 26.04.2006].

7. ÇÖLKESEN, R., 2006, Veri Yapıları ve Algoritmalar, Papatya Yayıncılık Eğitim Bilgisayar Sistemleri San.ve Tic. A.Ş., İstanbul, ISBN 975-6797-23-1.

8. DEVELİ, H., 2006, Yusuf Has Hacip – Kutatgu Bilig, Alkım Yayınevi, İstanbul, ISBN: 975-992065-4.

9. DOLANBAY, C., 2000, e-ticaret Strateji ve Yöntemler / Yeni Ekonomide Başarının Anahtarı, Meteksan Sistem ve Bilgisayar Teknolojileri A.Ş.,Ankara, ISBN 975-7746-28-2.

10. DURNA, U., 2002, Yenilik Yönetimi, Nobel Yayın Dağıtım Ltd.Şti., Ankara, ISBN: 975-591-360-2.

11. GÖZÜDELİ, Y., 2004, MS SQL Server ile Temel Veritabanı Programlama Tasarım ve Gerçekleme, Seçkin Yayıncılık San.Tic.A.Ş., Ankara, ISBN:975-347-838-0.

12. GROFF, J. R., ve WEINBERG, P. N., 2003, SQL:The Complite Refference, Alfa Basım Yayım Dağıtım Ltd. Şti. ISBN:975-297-307-8.

13. GÜLEŞ, H. K. ve ÖZATA, M., 2005, Sağlık Bilişim Sistemleri, Nobel Yayın Dağıtım Ltd.Şti., Ankara, ISBN 975-591-813-2.

14. GÜMÜŞTEKİN, G. E., 2004, İşletmelerde Yönetim Bilişim Sistemleri, Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt:11, Sayı:1, ISSN-1302-0064, 125-141

15. HENDERSON, K., 2003, SQL Server Depolanmış Yordamları, XML ve HTML,Alfa Basım Yayım Dağıtım Ltd.Şti. İstanbul, ISBN:975-297-337-X.

16. KARAGÜLLE, İ., 2004, Borland Delphi 8 VCL.NET Başlangıç Rehberi, Türkmen Kitapevi, İstanbul, ISBN: 975-6392-30-4.

17. KÖKSAL, A., 2002, Hastane İşletmeciliğinde Yönetim Bilgi Sisteminin Önemi ve Kurulurken Dikkat Edilmesi Gereken Hususlar,TSE, Ekonomik ve Teknik Dergi-Standard, Yıl: 41, Sayı:483, Mart 2002, ISSN 1300-8366.

18. KÖSE, T., 2004, Hastane Yöneticilerinin Hastanelerde kullanılan Yönetim Bilgi Sistemlerinin Yönetim Fonksiyonlarına Katkısı Konusundaki Görüşleri, Yüksek Lisans Tezi, Hacettepe Üniversitesi Sağlık Bilimleri Enstitüsü.

19. MALKOÇOĞLU, K., 1995, Teknolojik Değişim ve Yeniliklerin Hastane Yönetimine Etkisi: Üniversite Hastanelerinde Klinik Bazında Teknolojik Değişim ve Yeniliklere Karşı Çalışanların Tepkileri Açısından Bir İnceleme, Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.

20. Milliyet, Büyük Larouse Sözlük ve Ansiklopedisi, Interpress Yayınları, Cilt IV, İstanbul, 1992, s.1637.

21. ÖĞÜT, A., 2003, Bilgi Çağında Yönetim, Nobel Yayın Dağıtım Ltd.Şti., Ankara, ISBN 975-591-300-9.

22. ÖNDER, E., 2005, Yönetim Bilişim Sistemleri Kapsamında Web Tabanlı İlişkisel Veritabanı Yönetim Sistemleri ve Bir Uygulama, Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.

23. ÖZEN, Ü., 2002, Uygulama Yazılımlarının Kurum İçerisinde Hazırlanması ya da satın alınması için karar destek çatısı oluşturulması: hastane otomasyonu alımının değerlendirilmesi, Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi, Cilt 3, Sayı 1. Aynı zamanda; http://www.cumhuriyet.edu.tr/edergi/makale/131.pdf [Ziyaret Tarihi: 28.07.2006]

24. SARIDOĞAN, M. E., 2004, Yazılım Mühendisliği, Papatya Yayıncılık Eğitim Bilgisayar Sistemleri San.ve Tic. A.Ş., İstanbul, ISBN 975-6797-57-6.

25. ÜLGEN, H., 1977, Yönetim Bilişim Sistemi, İstanbul Üniversitesi İşletme Fakültesi, İşletme Fakültesi Dergisi, Cilt:6, Sayı:2, 135-157.

26. YANIK, M., 1997, Borland Delphi ile Görsel Prgoramlama, Beta Basım Yayım Dağıtım A.Ş., İstanbul, ISBN: 975-486-627-9.

EKLER

EK-A. Diş Tedavileri Fiyat Tarifesi-2006

DİŞ TEDAVİLERİ FİYAT TARİFESİ **EK-7**

SIRA NO	KODU	İŞLEM ADI	2006 BİRİM FİYAT (YTL)
1		**1. TEŞHİS VE TEDAVİ PLANLAMASI**	
2	401.010	Diş hekimi muayenesi (diş tabibi)	7,00
3	401.020	Uzman diş hekimi muayenesi (doktor)	9,00
4	401.030	Konsültan dişhekimi muayenesi	4,00
5	401.040	Konsültan uzman diş hekimi muayenesi	5,00
6	401.050	Diş röntgen filmi (periapikal) adedi	4,00
7	401.060	Oklüzal film (adedi)	3,00
8	401.070	Ekstra-oral röntgen filmi (adedi)	6,50
9	401.080	Ortopantomograf film (panoromik)	12,50
10	401.090	Sefalometrik film (adedi)	9,00
11	401.100	Antero-posterior sefalometrik film (adedi)	8,00
12	401.110	El-Bilek filmi (adedi)	7,00
13	401.120	TME röntgen filmi (adedi ve tetkiki)	9,00
14	401.130	Siyalografi (adedi)	5,00
15	401.140	Vitalite kontrolü	2,00
16	401.150	Bite-wing film (adedi)	3,00
17	401.160	Dijital Radyografi	8,50
18		**2. TEDAVİ VE ENDODONTİ**	
19	402.010	Amalgam dolgu (bir yüzlü)	20,00
20	402.020	Amalgam dolgu (iki yüzlü)	24,00
21	402.030	Amalgam dolgu (üç yüzlü)	26,00
22	402.040	İnley dolgu-oklüzal	8,00
23	402.050	İnley dolgu (iki yüzlü)	8,00
24	402.060	İnley dolgu (üç yüzlü)	10,00
25	402.070	Öndiş kompozit dolgu	21,00
26	402.080	Arkadiş kompozit dolgu	21,00
27	402.090	Öndiş Işınlı kompozit dolgu	26,00
28	402.100	Arkadiş Işınlı kompozit dolgu (bir yüzlü)	26,00
29	402.110	Arkadiş Işınlı kompozit dolgu (iki yüzlü)	28,00
30	402.120	Arkadiş Işınlı kompozit dolgu (üç yüzlü)	30,00
31	402.130	Kuafaj (dolgu hariç)	12,00
32	402.140	Amputasyon (dolgu hariç)	8,00
33	402.150	Kanal tedavisi- tek kanal (üst dolgu hariç)	46,00
34	402.152	Kanal tedavisi- iki kanal (üst dolgu hariç)	63,00
35	402.153	Kanal tedavisi- üç kanal (üst dolgu hariç)	89,00
36	402.154	Kanal tedavisi- ilave her kanal için	27,00

37	402.160	Black 5 (kole) dolgusu (amalgam)	6,00
38	402.170	Black 5 (kole) dolgusu (cam iyonomer))	6,00
39	402.180	Black 5 (kole) dolgusu (kompozit))	6,00
40	402.190	Cam ionomer dolgu	21,00
41	402.200	Dentin pinli restorasyonu (pin başına) (dolgu hariç)	7,00
42	402.210	İntrakoronel bleaching (diş başına) (en fazla altı diş)	2,50
43	402.220	Extrakoronel bleaching (tek çene)(Teknisyen ücreti hariç)	21,00
44	402.230	Dentin bağlayıcısı (diş başına)	3,50
45	402.240	Kanal içi post uygulaması (diş başına)(dolgu hariç)	7,00
46	402.250	Aşırı kole hassasiyeti tedavisi (tam çene)	4,00
47	402.260	Aşırı kole hassasiyeti tedavisi (tek diş)	2,00
48	402.270	Gangren veya periapikal lezyon tedavisi (her kanal için)	12,50
49	402.271	Gangren veya periapikal lezyonlu dişte kanal tedavisi- tek kanal (üst dolgu hariç)	49,00
50	402.272	Gangren veya periapikal lezyonlu dişte kanal tedavisi- iki kanal (üst dolgu hariç)	69,50
51	402.273	Gangren veya periapikal lezyonlu dişte kanal tedavisi- üç kanal (üst dolgu hariç)	93,00
52	402.280	Dolgu sökülmesi (her diş için)	6,00
53	402.290	Ekstirpasyon (her diş için)	11,00
54	402.300	Pansuman(seans başına)	1,50
55	402.310	Onley (laboratuar ücreti hariç)	13,00
56	402.320	Kanal dolgusu sökümü (her kanal için)	6,50
57	402.330	Kompozit venner	31,00
58		**3. PEDODONTİ**	
59	403.010	Fissur örtülmesi (sealant) (tek diş)	7,00
60	403.020	Yerel flor uygulaması (çürük proflaksisi) yarım çene	7,00
61	403.030	Prefabrike kron (Malzeme ücreti hariç)	23,00
62	403.040	Yer tutucu (sabit)(Teknisyen ücreti hariç)	20,00
63	403.050	Yer tutucu (hareketli)(Teknisyen ücreti hariç)	20,00
64	403.060	Çocuk protezi (akrilik, bölümlü) (tek çene)(Tek. Ücreti hariç)	35,00
65	403.070	Çocuk protezi (tam) (tek çene)(Tek. Ücreti hariç)	35,00
66	403.080	Strip kron	12,00
67	403.090	Kompomer dolgu (diş başına)	21,00
68	403.100	Açık apeksli dişte kanal tedavisi (her kanal için)	25,00
69	403.110	Bilinçli sedasyon	18,00
70		(Bu bölümde yer verilmeyen tedaviler için bu listenin diğer	
71		bölümlerindeki fiyatlar uygulanır.)	
72		**4. PROTEZ**	
73	404.010	Tam protez (akrilik) (tek çene)	150,00
74	404.020	Bölümlü protez (akrilik, tek çene)	150,00
75	404.030	Tam protez (metal kaideli-tek çene)	150,00
76	404.040	Bölümlü protez (metal kaideli, tek çene)	150,00
77	404.050	Immediat protez (akrilik) (tek çene)	45,00
78	404.060	Rebazaj (kaide yenileme-tek çene)	26,00
79	404.070	Proteze yumuşak akrilik uygulaması	21,00
80	404.080	Besleme (tek çene)	18,00
81	404.090	Tamir (akrilik protezde kırık veya çatlak)	9,00
82	404.100	Kroşe ilavesi	9,00
83	404.110	Metal iskelet tamiri	12,00
84	404.120	Diş ilavesi (tek diş)	9,00

85	404.130	Gnatoloji, TME-kas muayenesi (okluzyon kontrolü)	12,00
86	404.140	Oklüzal aşındırmalar (tek çene)	12,00
87	404.150	Gece plağı (bruksizm için)	21,00
88	404.160	Pinley ve çeşitleri	13,50
89	404.170	Tek parça döküm kron	40,00
90	404.180	Venner kron (akrilik)	40,00
91	404.181	Venner kron (seramik)	60,00
92	404.190	Pivo (çivili kron) veya post-core	40,00
93	404.200	Jaket kron (akrilik)	21,00
94	404.201	Jaket kron (tam seramik kron-metal desteksiz)	65,00
95	404.210	Teleskop primer kron-kopingli	32,00
96	404.220	Maryland (andheziv) köprü	23,00
97	404.230	Roach köprü	20,00
98	404.240	Geçici kron (her diş için)	8,00
99	404.250	Kron sökümü (her ayak üye için)	4,00
100	404.260	Düşmüş kron-köprü simantasyonu (her sabit tutucu için)	4,00
101	404.270	Kron, köprü tamiri (her üye için)	9,00
102	404.280	Damak yarığı protezi (over-denture)	50,00
103	404.290	Yeni doğanda preoperatif aparey (veren ücreti hariç)	32,00
104	404.300	Velum uzantılı konuşma apereyi	42,00
105	404.310	Geçici opturatör (cerrahi plaklar)	20,00
106	404.320	Basit çene defektlerine protetik tedavi	41,00
107	404.330	Komplike çene defektlerine protetik tdv	65,00
108	404.340	Yüz protezleri	50,00
109	404.350	Göz protezleri	50,00
110	404.360	Hassas tutuculu kronlar	40,00
111	404.370	Periodontal protez (hareketli,tek çene)	53,00
112	404.380	Periodontal protez (sabit-her üye için)	18,00
113	404.390	Laminate veneer	18,00
114	404.400	İmplant üstü kron köprü (üye başına)	41,00
115	404.410	İmplant üstü tam protez	150,00
116	404.420	İmplant üstü bölümlü protez	150,00
117		**5. AĞIZ DİŞ VE ÇENE CERRAHİSİ**	
118	405.010	Diş çekimi (infiltrasyon anestezi ile)	20,00
119	405.011	Diş çekimi (rejyonal anestezi ile)	21,00
120	405.020	Komplikasyonlu diş çekimi (infiltrasyon anestezi ile)	20,00
121	405.021	Komplikasyonlu diş çekimi (rejyonal anestezi ile)	25,00
122	405.030	Gömülü diş çekimi (mukoza retansiyonlu)	65,00
123	405.040	Gömülü diş çekimi (kemik retansiyonlu)	70,00
124	405.050	Gömülü kanin-premolar diş çekimi	70,00
125	405.060	Kök ucu rezeksiyonu (tek diş için) (diş dolgusu hariç)	70,00
126	405.070	Alveolit cerrahi tedavisi	10,00
127	405.080	Kanama müdahalesi	9,00
128	405.090	Alveol plastiği (yarım çene)	35,00
129	405.100	Alveol düzeltilmesi (tek çene)	40,00
130	405.110	Kist operasyonu (küçük)	65,00
131	405.120	Kist operasyonu (büyük)	75,00
132	405.130	Osteomiyelit operasyonu (tek çene)	47,00
133	405.140	Çene lüksasyonu (basit)	17,00
134	405.150	Vestibuloplasti operasyonu	70,00
135	405.160	Sinüs plastiği	50,00

136	405.170	Sert doku greftleme (greft ücreti hariç)	40,00
137	405.180	Biopsi	7,50
138	405.190	Apse drenajı (ekstraoral)	12,00
139	405.200	Fizik tedavi (infraruj) (seansı)	3,00
140	405.210	Reimplantasyon (tek diş)	18,00
141	405.220	Ototransplantasyon	18,00
142	405.230	Subperiostal implant (tek ünite)	100,00
143	405.240	Stomatit tedavisi (kısa süreli)	20,00
144	405.250	Stomatit tedavisi (uzun süreli)	28,00
145	405.260	Kemik içi implant (tek ünite)	120,00
146	405.270	Torus operasyonu (yarım çen)	35,00
147	405.280	Odontojenik tümör operasyonu (küçük)	70,00
148	405.290	Odontojenik tümör operasyonu (büyük)	150,00
149	405.300	Cerrahi nevralgi tedavisi	28,00
150	405.310	Küçük ameliyat	115,00
151	405.320	Orta ameliyat	155,00
152	405.330	Büyük ameliyat	210,00
153	405.340	Narkozlu diş çekimi (genel anestezi ile)	33,00
154	405.350	Oro-antral fistül tedavisi	80,00
155	405.360	Enjeksiyon	3,00
156	405.370	Premedikasyon ve sedasyon	8,00
157	405.380	Ortodontik tedavi amaçlı gömük dişlerin üzerinin açılması	19,00
158	405.390	T.M.E. Mekanoterapi	19,00
159	405.400	T.M.E. İçi enjeksiyon (tek taraflı)	3,50
160	405.410	Artrosentez (tek taraflı)	27,00
161	405.420	Lokal anestezi (İnfiltrasyon)	3,00
162	405.430	Lokal anestezi (rejyonal)	4,00
163		**6. PERİODONTOLOJİ**	
164	406.010	Periodontal apse tedavisi	10,00
165	406.020	Detartraj (diş taşı temizliği) (yarım çene)	13,00
166	406.030	Subgingival küretaj (yarım çene)	16,00
167	406.040	Gingivektomi(yarım çene)	17,00
168	406.050	Flep operasyonu (yarım çene)	60,00
169	406.060	Hemiseksion (kök amputasyonu) kanal tedavisi hariç	17,00
170	406.070	Serbest diş eti grefti (yarım çene)	55,00
171	406.080	Koronale kaydırma veya sliding flep (yarım çene)	55,00
172	406.090	Periodontal şine (splint) (geçici tek çene)	17,00
173	406.100	Periodontal şine (splint) (devamlı tek çene)	78,00
174	406.110	Aşırı kole hassasiyet tedavisi(yarım çene)	4,25
175	406.120	Frenektomi	14,00
176	406.130	Gingivoplasti (yarım çene)	7,00
177	406.140	Bio materyal uygulaması (diş başına)	25,00
178	406.150	Tunnel operasyonu (diş başına)	17,00
179	406.160	Membran uygulaması	27,00
180	406.170	Bağ dokusu grefti (diş başına)	27,00
181	406.180	Subgingival ilaç uygulaması	8,00
182		**7. ORTODONTİ**	
183	407.010	Sefalometrik film analizi (en fazla üç defa)	5,00
184	407.020	Antero-posterior sefalometrik film analizi	5,00
185	407.030	Bilgisayarlı sefalometrik film analizi (en fazla üç defa)	6,00
186	407.040	Bilgisayarlı büyük TME fonksiyon testi	33,00

187	407.050	Bilgisayarlı kas tonus analizi (EMG)	27,00
188	407.060	Kemik yaşı tayini	5,00
189	407.070	Ortodontik fotograf	5,00
190	407.080	Ortodontik fotograf analizi	3,00
191	407.090	Ortodontik model yapımı (en fazla üç defa)	5,00
192	407.100	(*) Ortodontik model analizi	5,00
193	407.110	(*) Angle sınıf 1.(class 1) anomalilerin ortodontik tedavisi	180,00
194	407.120	(*)Angle sınıf 2.(class 2) anomalilerin ortodontik tedavisi	225,00
195	407.130	(*) Angle snıf 3.(class 3) anomalilerin ortodontik tedavisi	270,00
196	407.140	(*) Kısa süreli ortodontik tedavi	65,00
197	407.150	Önleyici ortodontik tedavisi (dil paravanası)	17,00
198	407.160	(*) Pekiştirme tedavisi	45,00
199	407.170	Pekiştirme aygıtı (havley pi)	45,00
200	407.180	Sabit pekiştirme aygıtı (Lingual ratainer)	22,00
201	407.190	(*) Tek çeneyi ilgilendiren aparey yapımı	45,00
202	407.200	(*) İki çeneyi ilgilendiren aparey yapımı(aktivatör)	103,00
203	407.210	(*) Ağız dışı aparey tatbiki (headgear-yüzarkı)	45,00
204	407.220	(*) Ağız dışı aparey tatbiki (chinecap-çenelik)	45,00
205	407.230	Kayıp apareyin yeniden yapımı (tek çene)	85,00
206	407.240	Aparey tamiri	5,00
207	407.250	(*)Hızlı maksiller expansiyon apereyi	65,00
208	407.260	(*)Arkteli tatbiki (tek çene-her uygulama için)	12,00
209	407.270	(*)Band (braket) tatbiki (her diş için)	2,50
210	407.280	(*)Sefelometrik cerrahi planı	6,50
211	407.290	(*)Okluzal cerrahi splint	85,00
212	407.300	(*)Ortodondik ameliyat arkı (tek çene)	85,00
213	407.310	(*)Çıkan bantın tatbiki	2,50
214	407.320	(*)Açık kapanış ortodontik tedavisi	210,00

.....

EK-B. ICD-10 Tanı Kodları:

K00	**Diş çıkması ve gelişimi bozuklukları**
K00.0	**Anodonti**
K00.1	**Fazla sayıda dişler**
K00.2	**Dişlerin büyüklük ve şekil anomalileri**
K00.3	**Diş renklenmeleri**
K00.4	**Diş oluşum bozuklukları**
K00.5	**Dişlerin yapısında herediter bozukluklar, başka yerde sınıflanmamış**
K00.6	**Diş çıkmasındaki bozukluklar**
K00.7	**Teething sendromu**
K00.8	**Diş gelişiminin diğer bozuklukları**
K00.9	**Diş gelişim bozukluğu, tanımlanmamış**
K01	**Gömülü ve sıkışık dişler**
K01.0	**Gömülü diş**
K01.1	**Sıkışık diş**
K02	**Diş çürükleri**
K02.0	**Mine çürüğü**
K02.1	**Dentin çürüğü**
K02.2	**Sement çürüğü**
K02.3	**Durmuş dental çürükler**
K02.4	**Odontoklazi**
K02.8	**Diş çürükleri, diğer**
K02.9	**Diş çürüğü, tanımlanmamış**
K03	**Diş sert dokusunun diğer hastalıkları**
K03.0	**Dişin atrizyonu**
K03.1	**Diş abrazyonu**
K03.2	**Diş erozyonu**
K03.3	**Dişin patolojik rezorpsiyonu**
K03.4	**Hipersementozis**
K03.5	**Diş ankilozu**
K03.6	**Dişte birikimler [accretions]**
K03.7	**Dişin sert dokusunun diş çıktıktan sonrası renk değişikliği**
K03.8	**Dişin sert dokusunun diğer tanımlanmış hastalıkları**
K03.9	**Diş sert dokularının hastalığı, tanımlanmamış**
K04	**Pulpa ve periapikal dokuların hastalığı**
K04.0	**Pulpit**
K04.1	**Pulpa nekrozu**
K04.2	**Pulpa dejenerasyonu**
K04.3	**Pulpada anormal sert doku oluşumu**
K04.4	**Pulpal kaynaklı akut apikal periodontit**
K04.5	**Kronik apikal periodontit**
K04.6	**Periapikal apse, sinüs oluşumuyla birlikte**
K04.7	**Periapikal apse, sinüs oluşumu olmadan**
K04.8	**Radiküler kist**
K04.9	**Pulpa ve periapikal dokuların diğer ve tanımlanmamış hastalıkları**
K05	**Gingivit ve periodontal hastalıklar**
K05.0	**Akut gingivit**

K05.1	Kronik gingivit
K05.2	Akut periodontit
K05.3	Kronik periodontit
K05.4	Periodontoz
K05.5	Periodontal hastalıklar, diğer
K05.6	Periodontal hastalık, tanımlanmamış
K06	Gingiva ve edentüloz alveoler kenar diğer bozuklukları
K06.0	Gingival gerileme
K06.1	Gingival büyüme
K06.2	Gingival ve edentülöz alveolar ridge lezyonları, travma ile beraber
K06.8	Gingiva ve edentülöz alveoler ridge diğer tanımlanmış bozuklukları
K06.9	Gingiva ve edentülöz alveoler ridge bozukluğu, tanımlanmamış
K07	Dentofasyal anomaliler [malokluzyon dahil]
K07.0	Çene büyüklüğünün ana anomalileri
K07.1	Çene-kafa kaidesi anomalileri
K07.2	Diş arkı anomalileri
K07.3	Diş pozisyon anomalileri
K07.4	Maloklüzyon, tanımlanmamış
K07.5	Dentofasiyal fonksiyonal anomaliler
K07.6	Temporomandibüler eklem bozuklukları
K07.8	Dentofasiyal anomaliler, diğer
K07.9	Dentofasiyal anomali, tanımlanmamış
K08	Dişler ve destekleyici yapıların diğer hastalıkları
K08.0	Dişlerin eksfoliasyonu; sistemik sebeplerden dolayı
K08.1	Diş kaybı; kaza, çekim veya lokal periodontal hastalıktan dolayı
K08.2	Edentülöz alveoler kenar atrofisi
K08.3	Gömülü diş kökü
K08.8	Dişler ve destekleyici yapıların diğer tanımlanmış bozuklukları
K08.9	Dişler ve destekleyici yapıların bozukluğu, tanımlanmamış
K09	Oral bölge kistleri, başka yerde sınıflanmamış
K09.0	Gelişimsel odontojenik kistler
K09.1	Oral bölge gelişimsel (nonodontojenik) kistleri
K09.2	Çene kistleri, diğer
K09.8	Oral bölge diğer kistleri, başka yerde sınıflanmamış
K09.9	Oral bölge kisti, tanımlanmamış
K10	Çenenin diğer hastalıkları
K10.0	Çenenin gelişimsel bozuklukları
K10.1	Dev hücreli granülom, merkezi
K10.2	Çenenin enflamatuvar durumları
K10.3	Çene alveoliti
K10.8	Çenenin diğer tanımlanmış hastalıkları
K10.9	Çene hastalığı, tanımlanmamış
K11	Tükrük bezleri hastalıkları
K11.0	Tükrük bezi atrofisi
K11.1	Tükrük bezi hipertrofisi
K11.2	Sialoadenit
K11.3	Tükrük bezi apsesi
K11.4	Tükrük bezi fistülü
K11.5	Sialolitiazis
K11.6	Tükrük bezi mukoseli
K11.7	Tükrük salgısı bozuklukları
K11.8	Tükrük bezleri diğer hastalıkları
K11.9	Tükrük bezi hastalığı, tanımlanmamış

K12	**Stomatit ve ilgili lezyonlar**
K12.0	**Tekrarlayan oral aftlar**
K12.1	**Stomatit, diğer formları**
K12.2	**Ağız selülit ve apsesi**
K13	**Dudak ve oral mukozanın diğer hastalıkları**
K13.0	**Dudak hastalıkları**
K13.1	**Yanak ve dudak ısırma**
K13.2	**Dili dahil ağız epitelinin lökoplaki ve diğer bozuklukları**
K13.3	**Tüylü lökoplaki**
K13.4	**Oral mukozanın granüloma ve granüloma-benzeri lezyonları**
K13.5	**Oral submuköz fibrozis**
K13.6	**Oral mukozanın irritatif hiperplazisi**
K13.7	**Oral mukozanın diğer ve tanımlanmış lezyonları**
K14	**Dil hastalıkları**
K14.0	**Glossit**
K14.1	**Jeografik dil**
K14.2	**Median romboid glossit**
K14.3	**Dil papillalarının hipertrofisi**
K14.4	**Dil papillalarının atrofisi**
K14.5	**Plikalı dil**
K14.6	**Glossodini**
K14.8	**Dilin diğer hastalıkları**
K14.9	**Dil hastalığı, tanımlanmamış**
K20	**Özofajit**
K21	**Gastro-özofajial reflü hastalığı**
K21.0	**Gastro-özofajial reflü hastalığı, özofajit ile**
K21.9	**Gastro-özofajial reflü hastalığı, özofajitsiz**
K22	**Özofagusun diğer hastalıkları**
K22.0	**Kardia akalazisi**
K22.1	**Özofagus ülseri**
K22.2	**Özofagus tıkanıklığı**
K22.3	**Özofagus perforasyonu**
K22.4	**Özofagus diskinezisi**
K22.5	**Özofagus divertikülü, kazanılmış**
K22.6	**Gastro-özofajial laserasyon-hemoraji sendromu**
K22.8	**Özofagusun hastalıkları diğer, tanımlanmış**
K22.9	**Özofagusun hastalığı, tanımlanmamış**
K23*	**Özofagus bozuklukları, başka yerde sınıflanmış hastalıklarda**
K23.0*	**Tüberküloz Özofajiti (A18.8†)**
K23.1*	**Megaözofagus, Chagas hastalığında (B57.3†)**
K23.8*	**Özofagus bozuklukları, başka yerde sınıflanmış diğer hastalıklarda**
K25	**Mide ülseri**
K25.0	**Akut Mide ülseri, hemoraji mevcut**
K25.1	**Akut Mide ülseri, perforasyon mevcut**
K25.2	**Akut Mide ülseri, hemoraji ve perferasyon mevcut**
K25.3	**Akut Mide ülseri, hemoraji veya perforasyon yok**
K25.4	**Kronik veya tanımlanmamış Mide ülseri, hemoraji mevcut**
K25.5	**Kronik veya tanımlanmamış Mide ülseri, perforasyon mevcut**
K25.6	**Kronik veya tanımlanmamış Mide ülseri, hemoraji ve perforasyon mevcut**
K25.7	**Kronik Mide ülseri, hemoraji veya perforasyon yok**
K25.9	**Akut veya kronik olarak tanımlanmamış Mide ülseri, hemoraji veya perforasyon yok**
K26	**Duodenum ülseri**
K26.0	**Akut Doudenum ülseri, hemoraji mevcut**

K26.1	Akut Doudenum ülseri, perforasyon mevcut
K26.2	Akut Doudenum ülseri, hemoraji ve perferasyon mevcut
K26.3	Akut Doudenum ülseri, hemoraji veya perforasyon yok
K26.4	Kronik veya tanımlanmamış Doudenum ülseri, hemoraji mevcut
K26.5	Kronik veya tanımlanmamış Doudenum ülseri, perforasyon mevcut
K26.6	Kronik veya tanımlanmamış Doudenum ülseri, hemoraji ve perforasyon mevcut
K26.7	Kronik Doudenum ülseri, hemoraji veya perforasyon yok
K26.9	Akut veya kronik olarak tanımlanmamış Doudenum ülseri, hemoraji veya perforasyon yok
K27	Peptik ülser, yeri tanımlanmamış
K27.0	Akut peptik ülser, yeri tanımlanmamış, hemoraji mevcut
K27.1	Akut peptik ülser, yeri tanımlanmamış, perforasyon mevcut
K27.2	Akut peptik ülser, yeri tanımlanmamış, hemoraji ve perferasyon mevcut
K27.3	Akut peptik ülser, yeri tanımlanmamış, hemoraji veya perforasyon yok
K27.4	Kronik veya tanımlanmamış peptik ülser, yeri tanımlanmamış, hemoraji mevcut
K27.5	Kronik veya tanımlanmamış peptik ülser, yeri tanımlanmamış, perforasyon mevcut
K27.6	Kronik veya tanımlanmamış peptik ülser, yeri tanımlanmamış, hemoraji ve
K27.7	Kronik peptik ülser, yeri tanımlanmamış, hemoraji veya perforasyon yok
K27.9	Akut veya kronik olarak tanımlanmamış peptik ülser, yeri tanımlanmamış, hemoraji veya
K28	Gastrojejunal ülser
K28.0	Akut Gastrojejunal ülser, hemoraji mevcut
K28.1	Akut Gastrojejunal ülser, perforasyon mevcut
K28.2	Akut Gastrojejunal ülser, hemoraji ve perferasyon mevcut
K28.3	Akut Gastrojejunal ülser, hemoraji veya perforasyon yok
K28.4	Kronik veya tanımlanmamış Gastrojejunal ülser, hemoraji mevcut
K28.5	Kronik veya tanımlanmamış Gastrojejunal ülser, perforasyon mevcut
K28.6	Kronik veya tanımlanmamış Gastrojejunal ülser, hemoraji ve perforasyon mevcut
K28.7	Kronik Gastrojejunal ülser, hemoraji veya perforasyon yok
K28.9	Akut veya kronik olarak tanımlanmamış Gastrojejunal ülser, hemoraji veya perforasyon yok
K29	Gastrit ve duodenit
K29.0	Akut hemorajik gastrit
K29.1	Akut gastrit, diğer
K29.2	Alkolik gastrit
K29.3	Kronik yüzeyel gastrit
K29.4	Kronik atrofik gastrit
K29.5	Kronik gastrit, tanımlanmamış
K29.6	Gastritler, diğer
K29.7	Gastrit, tanımlanmamış
K29.8	Duodenit
K29.9	Gastroduodenit, tanımlanmamış
K30	Dispepsi
K31	Mide ve duodenum hastalıkları, diğer
K31.0	Akut mide dilatasyonu
K31.1	Yetişkin hipertrofik pilorik stenozu
K31.2	Mide saat camı striktürü ve stenozu
K31.3	Pilorospazm, başka yerde sınıflanmamış
K31.4	Gastrik divertikül
K31.5	Duodenum obstrüksiyonu
K31.6	Mide ve duodenum fistülü
K31.8	Mide ve duodenum diğer tanımlanmış hastalıkları
K31.9	Mide ve duodenum hastalığı, tanımlanmamış
K35	Akut apandisit
K35.0	Akut apandisit, yaygın peritonit ile
K35.1	Akut apandisit, peritoneal apse ile

K35.9	Akut apandisit, tanımlanmamış
K36	Apandisit, diğer
K37	Apandisit, tanımlanmamış
K38	Appendiksin diğer hastalıkları
K38.0	Appendiks hiperplazisi
K38.1	Apandiküler konkresyonlar
K38.2	Appendiks divertikülü
K38.3	Appendiks fistülü
K38.8	Appendiks diğer tanımlanmış hastalıkları
K38.9	Appendiksin hastalığı, tanımlanmamış
K40	İnguinal herni
K40.0	Bilateral inguinal herni; obstrüksiyonlu, gangrensiz
K40.1	Bilateral inguinal herni, gangrenli
K40.2	Bilateral inguinal herni, obstrüksiyonsuz veya gangrensiz
K40.3	Tek taraflı veya tanımlanmamış inguinal herni, obstrüksiyonlu,gangrensiz
K40.4	Tek taraflı veya tanımlanmamış inguinal herni, gangrenli
K40.9	Tek taraflı veya tanımlanmamış inguinal herni, gangrensiz veya obstrüksiyonsuz
K41	Femoral herni
K41.0	Bilateral femoral herni, obstrüksiyonlu, gangrensiz
K41.1	Bilateral femoral herni, gangrenli
K41.2	Bilateral femoral herni, obstrüksiyonsuz veya gangrensiz
K41.3	Tek taraflı veya tanımlanmamış femoral herni, obstrüksiyonlu,gangrensiz
K41.4	Tek taraflı veya tanımlanmamış femoral herni, gangrenli
K41.9	Tek taraflı veya tanımlanmamış femoral herni, obstrüksiyonsuz veya gangrensiz
K42	Umbilikal herni
K42.0	Umbilikal herni; obstrüksiyonlu, gangrensiz
K42.1	Umbilikal herni, gangrenli
K42.9	Umbilikal herni, obstrüksiyonsuz, gangrenli
K43	Ventral herni
K43.0	Ventral herni, obstrüksiyonlu, gangrensiz
K43.1	Ventral herni, gangrenli
K43.9	Ventral herni, obstrüksiyon ve gangrensiz
K44	Diyafragma hernisi
K44.0	Diyafragma hernisi, obstrüksiyonlu, gangrensiz
K44.1	Diyafragma hernisi, gangrenli
K44.9	Diyafragma hernisi, gangren ve obstrüksiyonsuz
K45	Abdominal diğer herniler
K45.0	Abdominal herniler diğer, tanımlanmış obstrüksiyonlu, gangrensiz
K45.1	Abdominal herni diğer, tanımlanmış gangrenli
K45.8	Abdominal herni diğer, tanımlanmış obstrüksiyon ve gangrensiz
K46	Abdominal herniler, tanımlanmamış
K46.0	Abdominal herni, tanımlanmamış obstrüksiyonlu, gangrensiz
K46.1	Abdominal herni, tanımlanmamış gangrenli
K46.9	Abdominal herni, obstrüksiyon ve gangrensiz tanımlanmamış
K50	Crohn hastalığı [regional enterit]
K50.0	Crohn hastalığı, ince barsak
K50.1	Crohn hastalığı, kalın barsak
K50.8	Crohn hastalığı, diğer
K50.9	Crohn hastalığı, tanımlanmamış
K51	Ülseratif kolit
K51.0	Ülseratif (kronik) enterokolit
K51.1	Ülseratif (kronik) ileokolit
K51.2	Ülseratif (kronik) proktit

K51.3	**Ülseratif (kronik) rektosigmoidit**
K51.4	**Kolon psödopolipozisi**
K51.5	**Mukozal proktokolit**
K51.8	**Ülseratif kolit, diğer**
K51.9	**Ülseratif kolit, tanımlanmamış**
K52	**Enfektif olmayan diğer gastroenterit ve kolit**
K52.0	**Gastroenterit ve kolit, radyasyona bağlı**
K52.1	**Toksik gastroenterit ve kolit**
K52.2	**Allerjik ve dietetik gastroenterit ve kolit**
K52.8	**Gastroenterit ve kolit diğer, tanımlanmış enfektif olmayan**
K52.9	**Gastroenterit ve kolit, enfektif olmayan, tanımlanmamış**
K55	**Barsağın vasküler bozuklukları**
K55.0	**Barsağın akut vasküler bozuklukları**
K55.1	**Barsağın kronik vasküler bozuklukları**
K55.2	**Kolon anjiodisplazisi**
K55.8	**Barsağın vasküler bozuklukları, diğer**
K55.9	**Barsağın vasküler bozukluğu, tanımlanmamış**
K56	**Paralitik ileus ve barsak obstrüksiyon, hernisiz**
K56.0	**Paralitik ileus**
K56.1	**İntususepsiyon**
K56.2	**Volvulus**
K56.3	**Safra taşı ileusu**
K56.4	**Barsak tıkanıklıkları diğer**
K56.5	**Barsak adezyonları [bantlar], obstrüksiyonlu**
K56.6	**Barsak obstrüksiyonu, diğer ve tanımlanmamış**
K56.7	**İleus, tanımlanmamış**
K57	**Barsağın divertiküler hastalığı**
K57.0	**İnce barsak divertiküler hastalığı, perforasyon ve apse mevcut**
K57.1	**İnce barsak divertiküler hastalığı, perforasyon veya apse olmadan**
K57.2	**Kalın barsak divertiküler hastalığı, perforasyon ve apse mevcut**
K57.3	**Kalın barsak divertiküler hastalığı, perforasyon veya apse olmadan**
K57.4	**İnce barsak ve kalın barsak divertiküler hastalığı, perforasyon ve apse mevcut**
K57.5	**İnce barsak ve kalın barsak divertiküler hastalığı, perforasyon veya apse**
K57.8	**Barsak divertiküler hastalığı, tanımlanmamış kısım, perforasyon ve apse mevcut**
K57.9	**Barsak divertiküler hastalığı, tanımlanmamış kısım, perforasyon veya apse olmadan**
K58	**İrritabl barsak sendromu**
K58.0	**İrritabl barsak sendromu, diareli**
K58.9	**İrritabl barsak sendromu, diaresiz**
K59	**Fonksiyonel barsak bozuklukları, diğer**
K59.0	**Kabızlık**
K59.1	**Fonksiyonel diare**
K59.2	**Nörojenik barsak, başka yerde sınıflanmamış**
K59.3	**Megakolon, başka yerde sınıflanmamış**
K59.4	**Anal spazm**
K59.8	**Fonksiyonel barsak bozuklukları diğer, tanımlanmış**
K59.9	**Fonksiyonel barsak bozuklukları, tanımlanmamış**
K60	**Anüs ve rektal bölge fissür ve fistülü**
K60.0	**Anal fissür, akut**
K60.1	**Anal fissür, kronik**
K60.2	**Anal fissür tanımlanmamış**
K60.3	**Anal fistül**
K60.4	**Rektal fistül**
K60.5	**Anorektal fistül**

K61	**Anal ve rektal bölge apseleri**
K61.0	**Anal apse**
K61.1	**Rektal apse**
K61.2	**Anorektal apse**
K61.3	**İskiorektal apse**
K61.4	**İntrasfinkterik apse**
K62	**Anüs ve rektumun diğer hastalıkları**
K62.0	**Anal polip**
K62.1	**Rektal polip**
K62.2	**Anal prolapsus**
K62.3	**Rektal prolapsus**
K62.4	**Anüs ve rektum stenozu**
K62.5	**Anüs ve rektum hemorajisi**
K62.6	**Anüs ve rektum ülseri**
K62.7	**Radyasyon proktiti**
K62.8	**Anüs ve rektumun diğer tanımlanmış hastalıkları**
K62.9	**Anüs ve rektum hastalığı, tanımlanmamış**
K63	**Barsağın diğer hastalıkları**
K63.0	**Barsak apsesi**
K63.1	**Barsak perforasyonu (travmatik olmayan)**
K63.2	**Barsak fistülü**
K63.3	**Barsak ülseri**
K63.4	**Enteroptoz**
K63.5	**Kolon polibi**
K63.8	**Barsağın diğer tanımlanmış hastalıkları**
K63.9	**Barsak hastalığı, tanımlanmamış**
K65	**Peritonit**
K65.0	**Akut peritonit**
K65.8	**Peritonitler, diğer**
K65.9	**Peritonit, tanımlanmamış**
K66	**Periton diğer bozuklukları**
K66.0	**Peritoneal adezyonlar**
K66.1	**Hemoperiton**
K66.8	**Peritonun diğer tanımlanmış bozuklukları**
K66.9	**Periton bozuklukları, tanımlanmamış**
K67*	**Peritonun bozuklukları, başka yerde sınıflanmış enfeksiyöz hastalıkları**
K67.0*	**Klamidya peritoniti (A74.8†)**
K67.1*	**Gonokokkal peritonit (A54.8†)**
K67.2*	**Sifilitik peritonit (A52.7†)**
K67.3*	**Tüberküloz peritoniti (A18.3†)**
K67.8*	**Peritonun diğer bozuklukları, başka yerde sınıflanmış hastalıklarda**
K70	**Alkolik karaciğer hastalığı**
K70.0	**Alkolik yağlı karaciğer**
K70.1	**Alkolik hepatit**
K70.2	**Karaciğerin alkolik fibrozis ve sklerozisi**
K70.3	**Karaciğer alkolik sirozu**
K70.4	**Alkolik karaciğer yetmezliği**
K70.9	**Alkolik karaciğer hastalığı, tanımlanmamış**
K71	**Toksik karaciğer hastalığı**
K71.0	**Toksik karaciğer hastalığı, kolestaz ile**
K71.1	**Toksik karaciğer hastalığı, hepatik nekroz ile**
K71.2	**Toksik karaciğer hastalığı, akut hepatit ile**
K71.3	**Toksik karaciğer hastalığı, kronik persistan hepatit ile**

K71.4	Toksik karaciğer hastalığı, kronik lobuler hepatit ile
K71.5	Toksik karaciğer hastalığı, kronik aktif hepatit ile
K71.6	Toksik karaciğer hastalığı, hepatit ile, başka yerde sınıflanmamış
K71.7	Toksik karaciğer hastalığı, karaciğer fibroz ve sirozu ile
K71.8	Toksik karaciğer hastalığı, diğer karaciğer bozuklukları ile
K71.9	Toksik karaciğer hastalığı, tanımlanmamış
K72	Karaciğer yetmezliği, başka yerde sınıflanmamış
K72.0	Akut ve subakut karaciğer yetmezliği
K72.1	Karaciğer yetmezliği, kronik
K72.9	Karaciğer yetmezliği, tanımlanmamış
K73	Kronik hepatit, başka yerde sınıflanmamış
K73.0	Kronik persistan hepatit, başka yerde sınıflanmamış
K73.1	Kronik lobüler hepatit, başka yerde sınıflanmamış
K73.2	Kronik aktif hepatit, başka yerde sınıflanmamış
K73.8	Kronik hepatit, başka yerde sınıflanmamış diğer
K73.9	Kronik hepatit, tanımlanmamış
K74	Karaciğer fibroz ve sirozu
K74.0	Hepatik fibroz
K74.1	Hepatik skleroz
K74.2	Hepatik fibroz, hepatik skleroz ile
K74.3	Primer biliyer siroz
K74.4	Sekonder biliyer siroz
K74.5	Biliyer siroz, tanımlanmamış
K74.6	Karaciğerin diğer ve tanımlanmamış sirozu
K75	Enflamatuvar karaciğer hastalıkları, diğer
K75.0	Karaciğer apsesi
K75.1	Portal ven flebiti
K75.2	Reaktif hepatit, spesifik olmayan
K75.3	Granülomatöz hepatit, başka yerde sınıflanmamış
K75.8	Enflamatuvar karaciğer hastalıkları diğer, tanımlanmış
K75.9	Enflamatuvar karaciğer hastalığı, tanımlanmamış
K76	Karaciğerin diğer hastalıkları
K76.0	Yağlı karaciğer, başka yerde sınıflanmamış
K76.1	Karaciğerin kronik pasif konjesyonu
K76.2	Karaciğerin merkezi hemorajik nekrozu
K76.3	Karaciğer enfarktüsü
K76.4	Peliozis hepatis
K76.5	Hepatik veno-okluzif hastalık
K76.6	Portal hipertansiyon
K76.7	Hepatorenal sendrom
K76.8	Karaciğerin diğer tanımlanmış hastalıkları
K76.9	Karaciğer hastalığı, tanımlanmamış
K77*	Karaciğer bozuklukları, başka yerde sınıflanmış hastalıklarda
K77.0*	Karaciğer bozuklukları, başka yerde sınıflanmış enfeksiyöz ve parazitik hastalıklarda
K77.8*	Karaciğer bozuklukları, başka yerde sınıflanmış hastalıklarda
K80	Safra taşı
K80.0	Safra kesesi taşı, akut kolesistit ile
K80.1	Safra kesesi taşı, diğer kolesistit ile
K80.2	Safra kesesi taşı, kolesistit olmadan
K80.3	Safra yolu taşı, kolanjit ile
K80.4	Safra yolu taşı, kolesistit ile
K80.5	Safra yolu taşı, kolanjit veya kolesistit olmadan
K80.8	Kolelitiazis, diğer

K81	**Kolesistit**
K81.0	**Akut kolesistit**
K81.1	**Kronik kolesistit**
K81.8	**Kolesistitler, diğer**
K81.9	**Kolesistit, tanımlanmamış**
K82	**Safra kesesinin diğer hastalıkları**
K82.0	**Safra kesesi obstrüksiyonu**
K82.1	**Safra kesesi hidropsu**
K82.2	**Safra kesesi perforasyonu**
K82.3	**Safra kesesi fistülü**
K82.4	**Safra kesesi kolesterolozisi**
K82.8	**Safra kesesinin diğer tanımlanmış hastalıkları**
K82.9	**Safra kesesi hastalığı, tanımlanmamış**
K83	**Safra yolunun diğer hastalıkları**
K83.0	**Kolanjit**
K83.1	**Safra yolu obstrüksiyonu**
K83.2	**Safra yolu perforasyonu**
K83.3	**Safra yolu fistülü**
K83.4	**Oddi sfinkter spazmı**
K83.5	**Bilier kist**
K83.8	**Safra yollarının diğer tanımlanmış hastalıkları**
K83.9	**Safra yolu hastalığı, tanımlanmamış**
K85	**Akut pankreatit**
K86	**Pankreasın diğer hastalıkları**
K86.0	**Kronik pankreatit, alkole bağlı**
K86.1	**Kronik pankreatitler, diğer**
K86.2	**Pankreas kisti**
K86.3	**Pankreas psödokisti**
K86.8	**Pankreasın diğer tanımlanmamış hastalıkları**
K86.9	**Pankreas hastalığı, tanımlanmamış**
K87*	**Safra kesesi, safra yolu ve pankreas bozuklukları, başka**
K87.0*	**Safra kesesi ve safra yolu bozuklukları, başka yerde sınıflanmış hastalıklarda**
K87.1*	**Pankreas bozuklukları, başka yerde sınıflanmış hastalıklarda**
K90	**Barsak malabsorpsiyonu**
K90.0	**Çöliak hastalığı**
K90.1	**Tropikal sprue**
K90.2	**Kör halka sendromu, başka yerde sınıflanmamış**
K90.3	**Pankreatik steatore**
K90.4	**Malabsorpsiyon, intoleransa bağlı, başka yerde sınıflanmamış**
K90.8	**Barsak malabsorbsiyonları, diğer**
K90.9	**Barsak malabsorpsiyon, tanımlanmamış**
K91	**Sindirim sisteminin girişim sonrası bozuklukları, başka yerde sınıflanmamış**
K91.0	**Kusma, gastrointestinal cerrahi sonrası**
K91.1	**Gastrik cerrahi sonrası sendromları**
K91.2	**Malabsorbsiyon cerrahi sonrası, başka yerde sınıflanmamış**
K91.3	**Girişim sonrası barsak obstrüksiyonu**
K91.4	**Kolostomi ve enterostomi malfonksiyonu**
K91.5	**Postkolesistektemi sendromu**
K91.8	**Sindirim sisteminin diğer girişim sonrası bozuklukları, başka yerde sınıflanmamış**
K91.9	**Sindirim sisteminin girişim sonrası bozuklukları, tanımlanmamış**
K92	**Sindirim sisteminin diğer hastalıkları**
K92.0	**Hematemez**
K92.1	**Melana**

K92.2	**Gastrointestinal hemoraji, tanımlanmamış**
K92.8	**Sindirim sisteminin diğer tanımlanmış hastalıkları**
K92.9	**Sindirim sistemi hastalığı, tanımlanmamış**
K93*	**Sindirim organlarının diğer bozuklukları, başka yerde sınıflanmış hastalıklarda**
K93.0*	**Barsaklar, periton ve mezenterik bezlerin tüberküloz bozuklukları (A18.3†)**
K93.1*	**Megakolon, Chagas hastalığında (B57.3†)**
K93.8*	**Sindirim organlarının diğer tanımlanmış bozuklukları, başka yerde sınıflanmış**

EK-C. Masa Üstü Uygulama Programı Kaynak Kodu:

Tamamı çok büyük yer tutacağından 3 örnek programın kaynak kodu aşağıda verilmiştir. Program kaynak kodu CD ortamında tam sürüm olarak mevcuttur.

Dis.dpr

```
program Dis;

uses
  Forms,
  windows,
  dbtables,
  Main in 'MAIN.PAS' {FormMain},
  Childwin in 'CHILDWIN.PAS' {MDIChild},
  About in 'about.pas' {AboutBox},
  Setting in 'Setting.pas' {FormSet},
  Company in 'Company.pas' {FormCompany},
  User in 'User.pas' {FormUser},
  Password in 'Password.pas' {FormPassword},
  First in 'First.pas' {FormFirst},
  Data in 'Data.pas' {FormData},
  Genel in 'Genel.pas',
  Query in 'Query.pas' {FormQuery},
  TahlilG in 'TahlilG.pas' {FormTahlil},
  Lab in 'Lab.pas' {FormLab},
  excel in 'excel.pas' {FormExcel},
  TahlilRpr in 'TahlilRpr.pas' {QRTahlil},
  Meslek in 'Meslek.pas' {FormMeslek},
  Kurum in 'Kurum.pas' {FormKurum},
  Il in 'Il.pas' {FormIl},
  Statu in 'Statu.pas' {FormStatu},
  Hasta in 'Hasta.pas' {FormHasta},
  Sonuc in 'Sonuc.pas' {FormSonuc},
  TahGrup in 'TahGrup.pas' {FormTahgrup},
  GrBas in 'GrBas.pas' {FormGrBas},
  TahRpr in 'TahRpr.pas' {QRTah},
  HastaSec in 'HastaSec.pas' {FormHastasec},
  TahKay in 'TahKay.pas' {FormTahkay},
  Fatkay in 'Fatkay.pas' {FormFatkay},
  Rapor in 'Rapor.pas' {FormRapor},
  RaporRpr in 'RaporRpr.pas' {QRRapor},
  Fatura in 'Fatura.pas' {FormFatura},
  FatRpr in 'FatRpr.pas' {QRFat},
  GunlukRpr in 'GunlukRpr.pas' {QRGunluk},
  FatIptal in 'FatIptal.pas' {FormFatIptal},
  FatDetay in 'FatDetay.pas' {FormFatDetay},
  EmekliRfr in 'EmekliRfr.pas' {FormEmekliRfr},
  EmekliRpr in 'EmekliRpr.pas' {QREmekli},
  FaturaRfr in 'FaturaRfr.pas' {FormFaturaRfr},
  FaturaRpr in 'FaturaRpr.pas' {QRFatura},
  LabRfr in 'LabRfr.pas' {FormLabRfr},
  LabRpr in 'LabRpr.pas' {QRLab},
  EvrakRfr in 'EvrakRfr.pas' {FormEvrakRfr},
  EvrakRpr in 'EvrakRpr.pas' {QREvrak},
```

```
FatOdeme in 'FatOdeme.pas' {FormFatOdeme},
Odeme in 'Odeme.pas' {FormOdeme},
OdemeRfr in 'OdemeRfr.pas' {FormOdemeRfr},
OdemeRpr in 'OdemeRpr.pas' {QROdeme},
GunlukRfr in 'GunlukRfr.pas' {FormGunlukRfr},
TekidRfr in 'TekidRfr.pas' {FormTekidRfr},
TekidRpr in 'TekidRpr.pas' {QRTekid},
GunEvrRpr in 'GunEvrRpr.pas' {QRGunEvr},
SifreDeg in 'SifreDeg.pas' {FormSifreDeg},
MakNo in 'MakNo.pas' {FormMakNo},
Makbuz in 'Makbuz.pas' {FormMakbuz},
MakIslem in 'MakIslem.pas' {FormMakIslem},
MakDetay in 'MakDetay.pas' {FormMakDetay},
MakbuzRpr in 'MakbuzRpr.pas' {QRMakbuz},
MakbuzRfr in 'MakbuzRfr.pas' {FormMakbuzRfr},
MakRpr in 'MakRpr.pas' {QRMak},
MakFatDt in 'MakFatDt.pas' {FormMakFatDt},
MakFat in 'MakFat.pas' {QRMakFat},
MakOdeme in 'MakOdeme.pas' {FormMakOdeme},
KasaRfr in 'KasaRfr.pas' {FormKasaRfr},
KasaRpr in 'KasaRpr.pas' {QRKasa},
GelGdRfr in 'GelGdRfr.pas' {FormGelGdRfr},
Gider in 'Gider.pas' {FormGider},
GiderRpr in 'GiderRpr.pas' {QRGider},
GelGdRpr in 'GelGdRpr.pas' {QRGelGd},
Cikis in 'Cikis.pas' {FormCikis},
GelirRfr in 'GelirRfr.pas' {FormGelirRfr},
GelirRpr in 'GelirRpr.pas' {QRGelir},
MesajYaz in 'MesajYaz.pas' {FormMesajYaz},
MesajOku in 'MesajOku.pas' {FormMesajOku},
MesajBak in 'MesajBak.pas' {FormMesajBak},
MesajTek in 'MesajTek.pas' {FormMesajTek},
Yedekleme in 'Yedekleme.pas' {FormYedekleme},
Mailset in 'Mailset.pas' {FormMailSet},
MakVezne in 'MakVezne.pas' {FormMakVezne},
VezneRpr in 'VezneRpr.pas' {QRVezne},
MakVezneIslem in 'MakVezneIslem.pas' {FormMakVezneIslem},
VezOdeme in 'VezOdeme.pas' {FormVezOdeme},
MakToVez in 'MakToVez.pas' {FormMakToVez},
VezRpr in 'VezRpr.pas' {QRVez},
VezRfr in 'VezRfr.pas' {FormVezRfr},
KasaVezRpr in 'KasaVezRpr.pas' {QRKasaVez},
VezAcik in 'VezAcik.pas' {FormVezAcik},
VezFatDt in 'VezFatDt.pas' {FormVezFatDt},
VezFat in 'VezFat.pas' {QRVezFat},
YilSonu in 'YilSonu.pas' {FormYilSonu},
Unvan in 'Unvan.pas' {FormUnvan},
Personel in 'Personel.pas' {FormPersonel},
SonucBsl in 'SonucBsl.pas' {FormSonucBsl},
SonucDok in 'SonucDok.pas' {FormSonucDok},
SonucRpr in 'SonucRpr.pas' {QRSonucRpr},
Verem in 'Verem.pas' {FormVerem},
VeremRp in 'VeremRp.pas' {QRVeremRp},
Grafik in 'Grafik.pas' {FormGrafik},
SayIslRpr in 'SayIslRpr.pas' {QRSayIsl},
VezOdmRpr in 'VezOdmRpr.pas' {QRVezOdm},
SayMhKod in 'SayMhKod.pas' {FormSayMhKod},
SayFis in 'SayFis.pas' {FormSayFis},
KurumGrp in 'KurumGrp.pas' {FormKurumGrp},
```

```
  KurumGrpRfr in 'KurumGrpRfr.pas' {FormKurumGrpRfr},
  KurumGrpRpr in 'KurumGrpRpr.pas' {QRKurumGrp},
  ResmiFat in 'ResmiFat.pas' {FormResmiFat},
  ResmiRfr in 'ResmiRfr.pas' {FormResmiRfr},
  ResmiRpr in 'ResmiRpr.pas' {QRResmiRpr},
  Tatil in 'Tatil.pas' {FormTatil},
  SskIcmRfr in 'SskIcmRfr.pas' {FormSskIcmRfr},
  KanGrubu in 'KanGrubu.pas' {FormKanGrubu},
  Ogrenci in 'Ogrenci.pas' {FormOgrenci},
  AnmKod in 'AnmKod.pas' {FormAnmKod},
  Ordi in 'Ordi.pas' {FormOrdi},
  Kursu in 'Kursu.pas' {FormKursu},
  Isler in 'Isler.pas' {FormIsler},
  DisSec in 'DisSec.pas' {FormDissec},
  Tedavi in 'Tedavi.pas' {FormTedavi},
  TedPlan in 'TedPlan.pas' {FormTedplan};

{$R *.RES}

  const
  s1:string='e';
  s2:string='b';
begin
  Session.AddPassword(s2+s1+s1);
  Session.AddPassword(S1+s2+s1+s1);
{  If GlobalfindAtom('calis')=0 then
     Atom:=GlobalAddAtom('calis')
  else
     Application.Terminate;}
  Application.Initialize;
  Application.CreateForm(TFormData, FormData);
  Application.CreateForm(TFormMain, FormMain);
  Application.CreateForm(TFormPassword, FormPassword);
  Application.CreateForm(TFormAbout, FormAbout);
  Application.CreateForm(TFormFirst, FormFirst);
  FormMain.Show;
  FormPassword.ShowModal;
  FormFirst.Show;
  Application.Title := 'Diş';
  Application.Run;
end.
```

Main.pas

```
unit Main;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, Menus,
  StdCtrls, Dialogs, Buttons, Messages, ExtCtrls, ComCtrls, StdActns,
  ActnList, ToolWin, ImgList, Db, DBTables, DBCtrls, Shellapi, ComObj;

type
  TFormMain = class(TForm)
    MainMenu1: TMainMenu;
    Help1: TMenuItem;
    HelpAboutItem: TMenuItem;
    StatusBar: TStatusBar;
    Settings1: TMenuItem;
    Settings2: TMenuItem;
    ToolBar1: TToolBar;
    Navigator: TDBNavigator;
    ToolButton7: TToolButton;
    ToolButton1: TToolButton;
    PrintBtn: TBitBtn;
    ExitBtn: TBitBtn;
    GenelProgramlar1: TMenuItem;
    MnItemKullanici1: TMenuItem;
    irketTanm1: TMenuItem;
    GenelProgramlar2: TMenuItem;
    LabaratuarTanm1: TMenuItem;
    TahlilTanm1: TMenuItem;
    lveleTanmlar1: TMenuItem;
    MnItemKurumGrp: TMenuItem;
    MeslekTanmlar1: TMenuItem;
    StatTanmlar1: TMenuItem;
    HastaKayt1: TMenuItem;
    HastazlkBilgileri1: TMenuItem;
    TahlilSonular1: TMenuItem;
    TahlilGrupTanmlar1: TMenuItem;
    Fatura1: TMenuItem;
    FaturaKayt1: TMenuItem;
    Raporlar1: TMenuItem;
    GnlkHastaRaporlar1: TMenuItem;
    Faturaptal1: TMenuItem;
    EmekliListesi1: TMenuItem;
    N1: TMenuItem;
    N2: TMenuItem;
    FaturaRaporlar1: TMenuItem;
    TahlilRaporlar1: TMenuItem;
    EvrakRaporlar1: TMenuItem;
    Faturademe1: TMenuItem;
    Dekontsuzdeme1: TMenuItem;
    N3: TMenuItem;
    KurumExtresi1: TMenuItem;
    TekidYazs1: TMenuItem;
    ifreDeitirme1: TMenuItem;
    Makbuz1: TMenuItem;
    MakbuzNo1: TMenuItem;
    Makbuzlem1: TMenuItem;
```

```
MakbuzRaporlar1: TMenuItem;
Makbuzdemeleri1: TMenuItem;
N4: TMenuItem;
KasaDefteri1: TMenuItem;
GelirGider1: TMenuItem;
N5: TMenuItem;
GelirGiderRaporu1: TMenuItem;
GiderGirii1: TMenuItem;
SrmDosyas1: TMenuItem;
Timer1: TTimer;
YTLevirici1: TMenuItem;
GelirRaporu1: TMenuItem;
TimerOut: TTimer;
Mesajlar1: TMenuItem;
MesajYazma1: TMenuItem;
GelenGidenMesajlar1: TMenuItem;
N6: TMenuItem;
Yedekleme1: TMenuItem;
MnItemmail: TMenuItem;
VezneAlnds1: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
VezneAlndslem1: TMenuItem;
SymMutVezneAl1: TMenuItem;
VezneAlndsRaporlar1: TMenuItem;
VezneAlndsdemesi1: TMenuItem;
VezneAlndsAklamalar1: TMenuItem;
N9: TMenuItem;
MnItemYSonu: TMenuItem;
nvanTanmlar1: TMenuItem;
PersonelTanmlar1: TMenuItem;
SonuBalklar1: TMenuItem;
N10: TMenuItem;
N11: TMenuItem;
RaporAklamalarXLS1: TMenuItem;
SaymlFii1: TMenuItem;
SaymanlklemFiiDzenleme1: TMenuItem;
N12: TMenuItem;
MuhasebeKodlar1: TMenuItem;
KrumGrupTanimlari1: TMenuItem;
FaturaGrupRaporlari1: TMenuItem;
ResmiFaturaRaporlar1: TMenuItem;
ResmiFaturaKesmeBankaTahsilat1: TMenuItem;
TatilGnleri1: TMenuItem;
SSKcmalListesi1: TMenuItem;
KanGruplar1: TMenuItem;
renciTanmlar1: TMenuItem;
AnamnezKodlar1: TMenuItem;
OralDiagnoz1: TMenuItem;
KrsTanmlar1: TMenuItem;
Yaplanler1: TMenuItem;
TedaviKodlar1: TMenuItem;
TedaviPlanlama2: TMenuItem;
Kliniklemleri1: TMenuItem;
procedure FileNew1Execute(Sender: TObject);
procedure HelpAbout1Execute(Sender: TObject);
procedure FileExit1Execute(Sender: TObject);
procedure Settings2Click(Sender: TObject);
procedure ExitBtnClick(Sender: TObject);
```

```
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure KullancTanmlar1Click(Sender: TObject);
procedure irketTanm1Click(Sender: TObject);
procedure LabaratuarTanm1Click(Sender: TObject);
procedure TahlilTanm1Click(Sender: TObject);
procedure PrintBtnClick(Sender: TObject);
procedure lveleTanmlar1Click(Sender: TObject);
procedure MnItemKurumGrpClick(Sender: TObject);
procedure MeslekTanmlar1Click(Sender: TObject);
procedure StatTanmlar1Click(Sender: TObject);
procedure HastazlkBilgileri1Click(Sender: TObject);
procedure TahlilSonular2Click(Sender: TObject);
procedure TahlilGrupTanmlar1Click(Sender: TObject);
procedure FaturaKayt1Click(Sender: TObject);
procedure GnlkHastaRaporlar1Click(Sender: TObject);
procedure Faturaptal1Click(Sender: TObject);
procedure EmekliListesi1Click(Sender: TObject);
procedure FaturaRaporlar1Click(Sender: TObject);
procedure TahlilRaporlar1Click(Sender: TObject);
procedure EvrakRaporlar1Click(Sender: TObject);
procedure Faturademe1Click(Sender: TObject);
procedure Dekontsuzdeme1Click(Sender: TObject);
procedure KurumExtresi1Click(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure TekidYazs1Click(Sender: TObject);
procedure ifreDeitirme1Click(Sender: TObject);
procedure MakbuzNo1Click(Sender: TObject);
procedure Makbuzlem1Click(Sender: TObject);
procedure MakbuzRaporlar1Click(Sender: TObject);
procedure Makbuzdemeleri1Click(Sender: TObject);
procedure KasaDefteri1Click(Sender: TObject);
procedure GelirGiderRaporu1Click(Sender: TObject);
procedure GiderGirii1Click(Sender: TObject);
procedure SrmDosyas1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure YTLevirici1Click(Sender: TObject);
procedure GelirRaporu1Click(Sender: TObject);
procedure TimerOutTimer(Sender: TObject);
procedure MesajYazma1Click(Sender: TObject);
procedure GelenGidenMesajlar1Click(Sender: TObject);
procedure Yedekleme1Click(Sender: TObject);
procedure MnItemmailClick(Sender: TObject);
procedure VezneAlnds1Click(Sender: TObject);
procedure VezneAlndslem1Click(Sender: TObject);
procedure VezneAlndsdemesi1Click(Sender: TObject);
procedure SymMutVezneAl1Click(Sender: TObject);
procedure VezneAlndsRaporlar1Click(Sender: TObject);
procedure VezneAlndsAklamalar1Click(Sender: TObject);
procedure MnItemYSonuClick(Sender: TObject);
procedure nvanTanmlar1Click(Sender: TObject);
procedure PersonelTanmlar1Click(Sender: TObject);
procedure SonuBalklar1Click(Sender: TObject);
procedure RaporAklamalarXLS1Click(Sender: TObject);
procedure SaymanlklemFiiDzenleme1Click(Sender: TObject);
procedure MuhasebeKodlar1Click(Sender: TObject);
procedure KrumGrupTanimlari1Click(Sender: TObject);
procedure FaturaGrupRaporlari1Click(Sender: TObject);
```

```
    procedure ResmiFaturaRaporlar1Click(Sender: TObject);
    procedure ResmiFaturaKesmeBankaTahsilat1Click(Sender: TObject);
    procedure TatilGnleri1Click(Sender: TObject);
    procedure SSKcmalListesi1Click(Sender: TObject);
    procedure KanGruplar1Click(Sender: TObject);
    procedure renciTanmlar1Click(Sender: TObject);
    procedure AnamnezKodlar1Click(Sender: TObject);
    procedure OralDiagnoz1Click(Sender: TObject);
    procedure KrsTanmlar1Click(Sender: TObject);
    procedure Yaplanler1Click(Sender: TObject);
    procedure TedaviKodlar1Click(Sender: TObject);
    procedure TedaviPlanlama2Click(Sender: TObject);
  private
    { Private declarations }
    procedure CreateMDIChild(const Name: string);
    procedure DoOnIdle(Sender: TObject; var Done: Boolean);
  public
    { Public declarations }
    Function KontrolTable(DataSet: TDataSet; Apply: Boolean): Boolean;
    Function Hakbul(Tname:String): String;
  end;

var
  FormMain: TFormMain;
  Txt         : string;
  SuperUser   : Boolean;
  SysYon      : Boolean;
  UserNum     : Integer;
  Username    : String;
  Mutemet     : String[20];
  Vezne       : String[20];
  Sayman      : String[20];
  SaymanUnvan : String[20];
  Imza        : String;
  Makam       : String;
  Timeout     : Boolean;
  Acilis      : Boolean;
  UyariMsj    : String[30];
  Kaydet      : Boolean;
  AktifForm   : String;
  LogNum      : Integer;
  ChartType   : String[3];
  Filtre      : String;
  Atom        : Word;
  AliasName   : String[8];
  Donem       : word;
  LocalIp     : String;
  Girsaat     : TTime;
  Ciksaat     : TTime;

const
  s1:string='e';
  s2:string='b';

implementation

{$R *.DFM}

uses ChildWin, About, Genel, Setting, User, Company,
```

```
    Data, Lab, TahlilG, excel, TahlilRpr, LabRpr, Kurum, Meslek, Il, Statu,
    Hasta, Sonuc, TahGrup, TahRpr, Fatkay, GunlukRpr, FatIptal, EmekliRfr,
    FaturaRfr, LabRfr, EvrakRfr, FatOdeme, Odeme, OdemeRfr, GunlukRfr,
    TekidRfr, SifreDeg, MakNo, MakIslem, MakbuzRfr, MakOdeme, KasaRfr,
    GelGdRfr, Gider, GiderRpr, Cikis, GelirRfr, MesajYaz, MesajOku,
    Yedekleme, Mailset, MakVezne, MakVezneIslem, VezOdeme, MakToVez, VezRfr,
    VezAcik, YilSonu, Unvan, Personel, SonucBsl, V, SayFis, SayMhKod,
  KurumGrp, KurumGrpRfr, ResmiRfr, ResmiFat, Tatil, SskIcmRfr, KanGrubu,
  Ogrenci, AnmKod, Ordi, Kursu, Isler, Tedavi, TedPlan;


procedure TFormMain.DoOnIdle(Sender: TObject; var Done: Boolean);
begin
  Timer1.Enabled := False;
  Timer1.Enabled := True;
end;


Function TFormMain.Hakbul(Tname:String): String;
begin
If Superuser then
  begin
  result:='EEEEEEEE';
  Exit;
  end;
result:='';
FormData.TableSecurity.Open;
If FormData.TableSecurity.Findkey([Usernum, Tname]) then
   result:=result+FormData.TableSecurityDView.asstring+FormData.TableSecurityDnew.asstring
           +FormData.TableSecurityDupdate.asstring+FormData.TableSecurityDdel.asstring
           +FormData.TableSecurityDprint.asstring+FormData.TableSecurityDQuery.asstring
           +FormData.TableSecurityDSet.asstring
else
   begin
   FormData.TableUser.Open;
   FormData.TableUser.Findkey([Usernum]);
   If FormData.TableSecurity.Findkey([FormData.TableUserGroupNum.asinteger, Tname]) then
      result:=result+FormData.TableSecurityDView.asstring+FormData.TableSecurityDnew.asstring
              +FormData.TableSecurityDupdate.asstring+FormData.TableSecurityDdel.asstring
              +FormData.TableSecurityDprint.asstring+FormData.TableSecurityDQuery.asstring
              +FormData.TableSecurityDSet.asstring+FormData.TableUserDBackup.asstring
   else
      If FormData.TableUser.Findkey([Usernum]) then
         result:=result+FormData.TableUserDView.asstring+FormData.TableUserDnew.asstring
                 +FormData.TableUserDupdate.asstring+FormData.TableUserDdel.asstring
                 +FormData.TableUserDprint.asstring+FormData.TableUserDQuery.asstring
                 +FormData.TableUserDSet.asstring+FormData.TableUserDBackup.asstring
      else
         result:='HHHHHHHH';
   end;
end;

Function TFormMain.KontrolTable(DataSet: TDataSet; Apply: Boolean): Boolean;
begin // Table Name'i parametre olarak göster //
  result:=True;
  If (Not Acilis) then
     begin
     If Kaydet then
        if (TdbDataSet(DataSet).State in dsEditModes) then
```

```
          FormMain.Navigator.BtnClick(nbpost);
    Exit;
    end;
  if (TdbDataSet(DataSet).State in dsEditModes) then
    If not Timeout then
    begin
      case Application.MessageBox('Kaydetmek istiyor musunuz?','Uyarı !', mb_IconStop+
         mb_YesNoCancel+ mb_defbutton3) of
       idyes    : begin
                  FormMain.Navigator.BtnClick(nbpost);
                  result:=True;
                  end;
       idno     : result:=True;
       idcancel : result:=False;
      end; // case //
    end;
 end;


procedure TFormMain.CreateMDIChild(const Name: string);
var
  Child: TMDIChild;
begin
  { create a new MDI child window }
  Child := TMDIChild.Create(Application);
  Child.Caption := Name;
  if FileExists(Name) then Child.Memo1.Lines.LoadFromFile(Name);
end;

procedure TFormMain.FileNew1Execute(Sender: TObject);
begin
  CreateMDIChild('NONAME' + IntToStr(MDIChildCount + 1));
end;

procedure TFormMain.HelpAbout1Execute(Sender: TObject);
begin
  FormAbout.ShowModal;
end;

procedure TFormMain.FileExit1Execute(Sender: TObject);
begin
  Close;
end;

procedure TFormMain.Settings2Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('SETTING'),1) then
  If FormSet=nil then
  Application.CreateForm(TFormSet, FormSet)
  else
    begin
    FormSet.WindowState:=wsNormal;
    FormSet.Show;
    end;
end;

procedure TFormMain.ExitBtnClick(Sender: TObject);
begin
Close;
```

```
end;

procedure TFormMain.FormCreate(Sender: TObject);
begin
FormData.TableCompany.open;
  If FormData.TableCompanyTimeout.AsInteger>0 then
     begin
     Timer1.Interval:=FormData.TableCompanyTimeout.AsInteger*60000;
     Application.OnIdle := DoOnIdle;
     end;
  If (Not SysYon) or (Aktifform<>'Yedekleme') then
     begin
     TimerOut.Interval:=FormData.TableCompanyUyari.AsInteger*1000;
     UyariMsj:=FormData.TableCompanyMesaj.AsString;
     TimerOut.Enabled:=True;
     end;
Mutemet:=FormData.TableCompanyMutemet.asstring;
Vezne:=FormData.TableCompanyVezne.asstring;
Sayman:=FormData.TableCompanySayman.asstring;
SaymanUnvan:=FormData.TableCompanySaymanUnvan.asstring;
Imza:=FormData.TableCompanySignname.asstring;
Makam:=FormData.TableCompanySignlabel.asstring;
FormData.TableCompany.close;
Timeout:=False;
Acilis:=True;
 If FormData.TableSetConfirm_delete.asstring='-' then
    Navigator.ConfirmDelete:=False
 else
    Navigator.ConfirmDelete:=True;
end;

procedure TFormMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
If (not Timeout) and (Acilis) then
   If FormData.TableSetConfirm_Exit.asstring='+' then
     If Application.MessageBox('Programdan çıkmak istiyor musunuz?','Exit !', mb_IconStop+
           mb_YesNo+ mb_defbutton1)<>Idyes then
       Abort;
If SysYon then
   begin
   If not FormData.TableCompanyAcilis.AsBoolean then
     If Application.MessageBox('Programı diğer kullanıcılara da açmak istiyor musunuz?','Kullanıma
Açma !', mb_IconStop+
           mb_YesNo+ mb_defbutton1)= Idyes then
           begin
           FormData.TableCompany.edit;
           FormData.TableCompanyAcilis.AsBoolean:=True;
           FormData.TableCompany.post;
           end;
   FormData.TableUser.First;
   while not FormData.TableUser.EOF do
      begin
      If (not FormData.TableUserSysYon.AsBoolean) and (not FormData.TableUserGirCik.AsBoolean)
then
       If Application.MessageBox('Programı kullanıcıya da açmak istiyor
musunuz?',PChar(FormData.TableUserNameSur.AsString), mb_IconStop+
          mb_YesNo+ mb_defbutton1)= Idyes then
          begin
          FormData.TableUser.edit;
```

```
          FormData.TableUserGirCik.AsBoolean:=True;
          FormData.Tableuser.post;
          end;
     FormData.TableUser.Next;
     end;
   end;
Session.RemovePassword(s2+s1+s1);
Userlogout(Lognum);
end;

procedure TFormMain.KullancTanmlar1Click(Sender: TObject);
begin
  If FormUser=nil then
     Application.CreateForm(TFormUser, FormUser)
  else
     Begin
     FormUser.WindowState:=wsNormal;
     FormUser.Show;
     End;
end;

procedure TFormMain.irketTanm1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('COMPANY'),1) then
  If FormCompany=nil then
     Application.CreateForm(TFormCompany, FormCompany)
  else
     begin
     FormCompany.WindowState:=wsNormal;
     FormCompany.Show;
     end;
end;

procedure TFormMain.LabaratuarTanm1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('LAB'),1) then
  If FormLab=nil then
     Application.CreateForm(TFormLab, FormLab)
  else
     begin
     FormLab.WindowState:=wsNormal;
     FormLab.Show;
     end;
end;

procedure TFormMain.TahlilTanm1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('TAHLIL'),1) then
  If FormTahlil=nil then
     Application.CreateForm(TFormTahlil, FormTahlil)
  else
     begin
     FormTahlil.WindowState:=wsNormal;
     FormTahlil.Show;
     end;
end;

procedure TFormMain.PrintBtnClick(Sender: TObject);
begin
```

```
If (FormData.TableSetPrintout.asstring='E') and ((AktifForm='Lab') or (AktifForm='Tahlil')
    or (AktifForm='Aircraft') or (AktifForm='Fleet')or (AktifForm='Charge')) then
   begin
   If FormExcel=nil then
      Application.CreateForm(TFormExcel, FormExcel);
   exit;
   end;
If AktifForm='Tahlil' then
  begin
  Application.CreateForm(TQRTahlil, QRTahlil);
  If FormTahlil.TableTahlil.Filter<>'' then
     QrTahlil.QueryTahlil.Filter:=FormTahlil.TableTahlil.Filter;
  If FormData.TableSetPrintout_pre.asstring='-' then
     QRTahlil.QuickReport.Print
  else
     QRTahlil.QuickReport.Preview;
  end
else
  If AktifForm='Sonuc' then
     begin
     Application.CreateForm(TQrTah, QrTah);
     QrTah.QuickReport.Dataset:=FormSonuc.TableSonuc;
     QrTah.Lprot.caption:=FormSonuc.QueryHastaProtno.asstring;
     QrTah.LAdi.caption:=FormSonuc.QueryHastaAdi.asstring;
     QrTah.LSoyAdi.caption:=FormSonuc.QueryHastaSoyAdi.asstring;
     QrTah.LTar.caption:=datetostr(FormSonuc.TableGeltarTarih.asdatetime);
     QrTah.LStatu.caption:=FormSonuc.TableEvrakStatuAdi.asstring;
     QrTah.QRDBText1.Dataset:=FormSonuc.TableSonuc;
     QrTah.QRDBText2.Dataset:=FormSonuc.TableSonuc;
     QrTah.QRDBText3.Dataset:=FormSonuc.TableSonuc;
     QrTah.QRDBText4.Dataset:=FormSonuc.TableSonuc;
     QrTah.QRDBText5.Dataset:=FormSonuc.TableSonuc;
     FormSonuc.TableSonuc.Filtered:=True;
     If FormData.TableSetPrintout_pre.asstring='-' then
        QrTah.QuickReport.Print
     else
        QrTah.QuickReport.Preview;
     FormSonuc.TableSonuc.Filtered:=False;
     end
  else
     If AktifForm='Gider' then
        begin
        Application.CreateForm(TQRGider, QRGider);
        If FormData.TableSetPrintout_pre.asstring='-' then
           QRGider.QuickReport.Print
        else
           QRGider.QuickReport.Preview;
        end;


end;

procedure TFormMain.lveleTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('IL'),1) then
  If FormIl=nil then
     Application.CreateForm(TFormIl, FormIl)
  else
     begin
```

```
    FormIl.WindowState:=wsNormal;
    FormIl.Show;
    end;
end;

procedure TFormMain.MnItemKurumGrpClick(Sender: TObject);
begin
If not HakKontrol(Hakbul('KURUM'),1) then
  If FormKurum=nil then
    Application.CreateForm(TFormKurum, FormKurum)
  else
    begin
    FormKurum.WindowState:=wsNormal;
    FormKurum.Show;
    end;
end;

procedure TFormMain.MeslekTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('MESLEK'),1) then
  If FormMeslek=nil then
    Application.CreateForm(TFormMeslek, FormMeslek)
  else
    begin
    FormMeslek.WindowState:=wsNormal;
    FormMeslek.Show;
    end;
end;

procedure TFormMain.StatTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('STATU'),1) then
  If Formstatu=nil then
    Application.CreateForm(TFormstatu, Formstatu)
  else
    begin
    Formstatu.WindowState:=wsNormal;
    Formstatu.Show;
    end;
end;

procedure TFormMain.HastazlkBilgileri1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('HASTA'),1) then
  If FormHasta=nil then
    Application.CreateForm(TFormHasta, FormHasta)
  else
    begin
    FormHasta.WindowState:=wsNormal;
    FormHasta.Show;
    end;
end;

procedure TFormMain.TahlilSonular2Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('SONUC'),1) then
  If FormSonuc=nil then
    Application.CreateForm(TFormSonuc, FormSonuc)
  else
```

```
    begin
    FormSonuc.WindowState:=wsNormal;
    FormSonuc.Show;
    end;
end;

procedure TFormMain.TahlilGrupTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('TAHGRUP'),1) then
  If FormTahgrup=nil then
    Application.CreateForm(TFormTahgrup, FormTahgrup)
  else
    begin
    FormTahgrup.WindowState:=wsNormal;
    FormTahgrup.Show;
    end;
end;

procedure TFormMain.FaturaKayt1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('FATURA'),1) then
  If FormFatkay=nil then
    Application.CreateForm(TFormFatkay, FormFatkay)
  else
    begin
    FormFatkay.WindowState:=wsNormal;
    FormFatkay.Show;
    end;
end;

procedure TFormMain.GnlkHastaRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPGUNLUK'),1) then
   begin
   Application.CreateForm(TFormGunlukRfr, FormGunlukRfr);
   FormGunlukRfr.ShowModal;
   end;
end;

procedure TFormMain.Faturaptal1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('FATURA'),1) then
   begin
   Application.CreateForm(TFormFatiptal, FormFatiptal);
   FormFatiptal.ShowModal;
   end;
end;

procedure TFormMain.EmekliListesi1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPEMEKLI'),1) then
   begin
   Application.CreateForm(TFormEmekliRfr, FormEmekliRfr);
   FormEmekliRfr.ShowModal;
   end;
end;

procedure TFormMain.FaturaRaporlar1Click(Sender: TObject);
begin
```

```
If not HakKontrol(Hakbul('RAPFATURA'),1) then
  begin
  Application.CreateForm(TFormFaturaRfr, FormFaturaRfr);
  FormFaturaRfr.ShowModal;
  end;
end;

procedure TFormMain.TahlilRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPTAHLIL'),1) then
  begin
  Application.CreateForm(TFormLabRfr, FormLabRfr);
  FormLabRfr.ShowModal;
  end;
end;

procedure TFormMain.EvrakRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPEVRAK'),1) then
  begin
  Application.CreateForm(TFormEvrakRfr, FormEvrakRfr);
  FormEvrakRfr.ShowModal;
  end;
end;

procedure TFormMain.Faturademe1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('ODEME'),1) then
  begin
  Application.CreateForm(TFormFatOdeme, FormFatOdeme);
  FormFatOdeme.ShowModal;
  end;
end;

procedure TFormMain.Dekontsuzdeme1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('ODEME'),1) then
 If FormOdeme=nil then
    Application.CreateForm(TFormOdeme, FormOdeme)
 else
    begin
    FormOdeme.WindowState:=wsNormal;
    FormOdeme.Show;
    end;
end;

procedure TFormMain.KurumExtresi1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPKURUM'),1) then
  begin
  Application.CreateForm(TFormOdemeRfr, FormOdemeRfr);
  FormOdemeRfr.ShowModal;
  end;
end;

procedure TFormMain.FormDestroy(Sender: TObject);
begin
{GlobalDeleteAtom(Atom);}
end;
```

```
procedure TFormMain.FormShow(Sender: TObject);
begin
{  If GlobalFindAtom('TAHLIL')<>0 then
    Application.Terminate;}
end;

procedure TFormMain.TekidYazs1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPTEKID'),1) then
  begin
  Application.CreateForm(TFormTekidRfr, FormTekidRfr);
  FormTekidRfr.ShowModal;
  end;
end;

procedure TFormMain.ifreDeitirme1Click(Sender: TObject);
begin
 Application.CreateForm(TFormSifreDeg, FormSifreDeg);
 FormSifreDeg.ShowModal;
end;

procedure TFormMain.MakbuzNo1Click(Sender: TObject);
begin
Application.CreateForm(TFormMakNo, FormMakNo);
FormMakNo.ShowModal;
end;

procedure TFormMain.Makbuzlem1Click(Sender: TObject);
begin
 Application.CreateForm(TFormMakIslem, FormMakIslem);
 FormMakIslem.ShowModal;
end;

procedure TFormMain.MakbuzRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPSAYMAN'),1) then
  begin
  Application.CreateForm(TFormMakbuzRfr, FormMakbuzRfr);
  FormMakbuzRfr.ShowModal;
  end;
end;

procedure TFormMain.Makbuzdemeleri1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('MAKODEME'),1) then
 If FormMakOdeme=nil then
    Application.CreateForm(TFormMakOdeme, FormMakOdeme)
 else
    begin
    FormMakOdeme.WindowState:=wsNormal;
    FormMakOdeme.Show;
    end;
end;

procedure TFormMain.KasaDefteri1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPKASA'),1) then
  begin
```

```
  Application.CreateForm(TFormKasaRfr, FormKasaRfr);
  FormKasaRfr.ShowModal;
  end;
end;

procedure TFormMain.GelirGiderRaporu1Click(Sender: TObject);
begin
Application.CreateForm(TFormGelGdRfr, FormGelGdRfr);
FormGelGdRfr.ShowModal;
end;

procedure TFormMain.GiderGirii1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('GIDER'),1) then
 If FormGider=nil then
    Application.CreateForm(TFormGider, FormGider)
 else
    begin
    FormGider.WindowState:=wsNormal;
    FormGider.Show;
    end;
end;

procedure TFormMain.SrmDosyas1Click(Sender: TObject);
begin
shellexecute(0,'open', 'SURUM.txt','','',SW_NORMAL);
end;

procedure TFormMain.Timer1Timer(Sender: TObject);
begin
  Application.CreateForm(TFormCikis, FormCikis);
  Tur:=1;
  FormCikis.ShowModal;
  If Timeout then FormMain.Close;
end;

procedure TFormMain.YTLevirici1Click(Sender: TObject);
begin
shellexecute(0,'open', 'Convert.exe','','',SW_NORMAL);
end;

procedure TFormMain.GelirRaporu1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPGELIR'),1) then
  begin
  Application.CreateForm(TFormGelirRfr, FormGelirRfr);
  FormGelirRfr.ShowModal;
  end;
end;

procedure TFormMain.TimerOutTimer(Sender: TObject);
begin
If (AktifForm='Yedekleme') or  (AktifForm='Password') then exit;
FormData.TableCompany.open;
If Not FormData.TableCompanyAcilis.AsBoolean then
  begin
  Acilis:=False;
  Application.CreateForm(TFormCikis, FormCikis);
  FormCikis.Label1.Caption:=UyariMsj;
```

```
   FormCikis.Label4.Caption:='Kayıt edilmemiş bilgilerinizi kaydetmek istiyor musunuz?';
   Tur:=2;
   FormCikis.ShowModal;
   FormMain.close;
   end;
FormData.TableCompany.close;
If (not SysYon) then
   begin
   FormData.TableUser.FindKey([UserNum]);
   If (Time>FormData.TableUserCiksaat.AsDateTime) or
(Time<FormData.TableUserGirSaat.AsDateTime) or (not FormData.TableUserGirCik.AsBoolean)then
      begin
      Acilis:=False;
      Application.CreateForm(TFormCikis, FormCikis);
      FormCikis.Label1.Caption:='Kısıtlı Kullanım';
      FormCikis.Label4.Caption:='Kayıt edilmemiş bilgilerinizi kaydetmek istiyor musunuz?';
      Tur:=2;
      FormCikis.ShowModal;
      FormMain.close;
      end;
   end;

end;

procedure TFormMain.MesajYazma1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('MESAJ'),1) then
  If FormMesajYaz=nil then
     Application.CreateForm(TFormMesajYaz, FormMesajYaz)
  else
     begin
     FormMesajYaz.WindowState:=wsNormal;
     FormMesajYaz.Show;
     end;
end;

procedure TFormMain.GelenGidenMesajlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('MESAJ'),1) then
  If FormMesajOku=nil then
  Application.CreateForm(TFormMesajOku, FormMesajOku)
  else
     begin
     FormMesajOku.WindowState:=wsNormal;
     FormMesajOku.Show;
     end;
end;

procedure TFormMain.Yedekleme1Click(Sender: TObject);
begin
  Application.CreateForm(TFormYedekleme, FormYedekleme);
  FormYedekleme.ShowModal;
end;

procedure TFormMain.MnItemmailClick(Sender: TObject);
begin
  Application.CreateForm(TFormMailSet, FormMailSet);
  FormMailSet.ShowModal;
end;
```

```
procedure TFormMain.VezneAlnds1Click(Sender: TObject);
begin
  Application.CreateForm(TFormMakVezne, FormMakVezne);
  FormMakVezne.ShowModal;
end;

procedure TFormMain.VezneAlndslem1Click(Sender: TObject);
begin
  Application.CreateForm(TFormMakVezneIslem, FormMakVezneIslem);
  FormMakVezneIslem.ShowModal;
end;

procedure TFormMain.VezneAlndsdemesi1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('VEZODEME'),1) then
  If FormVezOdeme=nil then
     Application.CreateForm(TFormVezOdeme, FormVezOdeme)
  else
     begin
     FormVezOdeme.WindowState:=wsNormal;
     FormVezOdeme.Show;
     end;
end;

procedure TFormMain.SymMutVezneAl1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('MAKTOVEZ'),1) then
  If FormMaktovez=nil then
     Application.CreateForm(TFormMakToVez, FormMakToVez)
  else
     begin
     FormMaktovez.WindowState:=wsNormal;
     FormMaktovez.Show;
     end;
end;

procedure TFormMain.VezneAlndsRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPVEZNE'),1) then
   begin
   Application.CreateForm(TFormVezRfr, FormVezRfr);
   FormVezRfr.ShowModal;
   end;
end;

procedure TFormMain.VezneAlndsAklamalar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('VEZCIK'),1) then
  If FormVezAcik=nil then
     Application.CreateForm(TFormVezAcik, FormVezAcik)
  else
     begin
     FormVezAcik.WindowState:=wsNormal;
     FormVezAcik.Show;
     end;
end;

procedure TFormMain.MnItemYSonuClick(Sender: TObject);
```

```
begin
  Application.CreateForm(TFormYilSonu, FormYilSonu);
  FormYilSonu.ShowModal;
end;

procedure TFormMain.nvanTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('UNVAN'),1) then
  If FormUnvan=nil then
     Application.CreateForm(TFormUnvan, FormUnvan)
  else
     begin
     FormUnvan.WindowState:=wsNormal;
     FormUnvan.Show;
     end;
end;

procedure TFormMain.PersonelTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('PERSONEL'),1) then
  If FormPersonel=nil then
    Application.CreateForm(TFormPersonel, FormPersonel)
  else
     begin
     FormPersonel.WindowState:=wsNormal;
     FormPersonel.Show;
     end;
end;

procedure TFormMain.SonuBalklar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('SONUCBSL'),1) then
  If FormSonucBsl=nil then
     Application.CreateForm(TFormSonucBsl, FormSonucBsl)
  else
     begin
     FormSonucBsl.WindowState:=wsNormal;
     FormSonucBsl.Show;
     end;
end;

procedure TFormMain.RaporAklamalarXLS1Click(Sender: TObject);
var
ExcelApp: OleVariant;

begin
  try
    ExcelApp := GetActiveOleObject('Excel.Application');
  except
    try
      ExcelApp := CreateOleObject('Excel.Application');
    except
      ShowMessage('Excel dosyası açılamadı. Excel yüklü olmayabilir?');
      Exit;
    end;
  end;
ExcelApp.Workbooks.Open(FileName :=
ExtractFilePath(Application.ExeName)+'RPR\RaporAciklama.XLS', ReadOnly := True);
ExcelApp.Visible := True;
```

```
ExcelApp := UnAssigned;

end;

procedure TFormMain.SaymanlklemFiiDzenleme1Click(Sender: TObject);
begin
  Application.CreateForm(TFormSayfis, FormSayfis);
  FormSayfis.ShowModal;
end;

procedure TFormMain.MuhasebeKodlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('SAYISFIS'),1) then
  If FormSayMhKod=nil then
     Application.CreateForm(TFormSayMhKod, FormSayMhKod)
  else
     begin
     FormSaymhkod.WindowState:=wsNormal;
     FormSaymhkod.Show;
     end;
end;

procedure TFormMain.KrumGrupTanimlari1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('KURUMGRP'),1) then
  If FormMeslek=nil then
     Application.CreateForm(TFormKurumGrp, FormKurumGrp)
  else
     begin
     FormKurumGrp.WindowState:=wsNormal;
     FormKurumGrp.Show;
     end;
end;

procedure TFormMain.FaturaGrupRaporlari1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPFATURAGRP'),1) then
   begin
   Application.CreateForm(TFormKurumGrpRfr, FormKurumGrpRfr);
   FormKurumGrpRfr.ShowModal;
   end;
end;

procedure TFormMain.ResmiFaturaRaporlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPRESMIFAT'),1) then
   begin
   Application.CreateForm(TFormResmiRfr, FormResmiRfr);
   FormResmiRfr.ShowModal;
   end;
end;

procedure TFormMain.ResmiFaturaKesmeBankaTahsilat1Click(Sender: TObject);
begin
  Application.CreateForm(TFormResmiFat, FormResmiFat);
  FormResmiFat.BtnIptal.Enabled:=False;
  FormResmiFat.TableResmiFat.Insert;
  FormResmiFat.TableResmiFatOdemeTuru.AsString:='B';
  FormResmiFat.DbEdit4.Enabled:=True;
```

```
  FormResmiFat.DbEdit1.Enabled:=True;
  FormResmiFat.DbEdit15.Enabled:=True;
  FormData.TableVezAcik.First;
  while not FormData.TableVezAcik.EOF do
  begin
    If FormData.TableVezAcikKdv.Asinteger>0 then
       begin
       FormResmiFat.CbAciklama.Items.add(FormData.TableVezAcikAciklama.AsString);
       FormResmiFat.CbKdv.Items.add(FormData.TableVezAcikKdv.AsString);
       end;
    FormData.TableVezAcik.Next;
  end;
  FormResmiFat.CbAciklama.ItemIndex := 0;
  FormResmiFat.CBAciklama.Visible:=True;
  FormResmiFat.DbEdit9.Enabled:=False;
  FormResmiFat.ShowModal;
end;

procedure TFormMain.TatilGnleri1Click(Sender: TObject);
begin
Application.CreateForm(TFormTatil, FormTatil);
FormTatil.ShowModal;
end;

procedure TFormMain.SSKcmalListesi1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('RAPSSK'),1) then
   begin
   Application.CreateForm(TFormSskIcmRfr, FormSskIcmRfr);
   FormSskIcmRfr.ShowModal;
   end;
end;

procedure TFormMain.KanGruplar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('KANGRUBU'),1) then
  If FormKanGrubu=nil then
     Application.CreateForm(TFormKanGrubu, FormKanGrubu)
  else
     begin
     FormKanGrubu.WindowState:=wsNormal;
     FormKanGrubu.Show;
     end;
end;

procedure TFormMain.renciTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('OGRENCI'),1) then
  If FormOgrenci=nil then
     Application.CreateForm(TFormOgrenci, FormOgrenci)
  else
     begin
     FormOgrenci.WindowState:=wsNormal;
     FormOgrenci.Show;
     end;
end;

procedure TFormMain.AnamnezKodlar1Click(Sender: TObject);
begin
```

```
If not HakKontrol(Hakbul('ANMKOD'),1) then
  If FormAnmkod=nil then
     Application.CreateForm(TFormAnmKod, FormAnmKod)
  else
     begin
     FormAnmkod.WindowState:=wsNormal;
     FormAnmkod.Show;
     end;
end;

procedure TFormMain.OralDiagnoz1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('ORDI'),1) then
  If FormOrdi=nil then
     Application.CreateForm(TFormOrdi, FormOrdi)
  else
     begin
     FormOrdi.WindowState:=wsNormal;
     FormOrdi.Show;
     end;
end;

procedure TFormMain.KrsTanmlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('KURSU'),1) then
  If FormKursu=nil then
     Application.CreateForm(TFormKursu, FormKursu)
  else
     begin
     FormKursu.WindowState:=wsNormal;
     FormKursu.Show;
     end;
end;

procedure TFormMain.Yaplanler1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('IS'),1) then
  If FormIsler=nil then
     Application.CreateForm(TFormIsler, FormIsler)
  else
     begin
     FormIsler.WindowState:=wsNormal;
     FormIsler.Show;
     end;

end;

procedure TFormMain.TedaviKodlar1Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('TEDAVI'),1) then
  If FormTedavi=nil then
    Application.CreateForm(TFormTedavi, FormTedavi)
  else
     begin
     FormTedavi.WindowState:=wsNormal;
     FormTedavi.Show;
     end;
end;
```

```
procedure TFormMain.TedaviPlanlama2Click(Sender: TObject);
begin
If not HakKontrol(Hakbul('TEDAVI'),1) then
  If FormTedplan=nil then
     Application.CreateForm(TFormTedplan, FormTedplan)
  else
     begin
     FormTedplan.WindowState:=wsNormal;
     FormTedplan.Show;
     end;
end;

end.
```

Ordi.pas

```
unit Ordi;

interface

uses
  BDE, Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, DBCtrls, Mask, Db, DBTables, ComCtrls, ExtCtrls, Menus, Grids,
  DBGrids, Buttons,  ComObj;

type
  TFormOrdi = class(TForm)
    DataSourceHasta: TDataSource;
    PopMenu: TPopupMenu;
    LoadPos1: TMenuItem;
    SavePos1: TMenuItem;
    StatusBarSonuc: TStatusBar;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    DbGrid1: TDBGrid;
    Memo1: TMemo;
    QueryHasta: TQuery;
    QueryHastaadsoy: TStringField;
    QueryHastanum: TIntegerField;
    QueryHastaadi: TStringField;
    QueryHastasoyadi: TStringField;
    QueryHastaprotno: TStringField;
    QueryHastacins: TStringField;
    Panel2: TPanel;
    DBGrid2: TDBGrid;
    TableAnamnez: TTable;
    TableAnamnezNum: TAutoIncField;
    TableAnamnezHasta_num: TIntegerField;
    TableAnamnezTarih: TDateField;
    TableAnamnezAnmkod_num: TIntegerField;
    TableAnamnezAciklama: TStringField;
    TableAnamnezAnmAdi: TStringField;
    DataSourceAnamnez: TDataSource;
    Panel5: TPanel;
    Panel6: TPanel;
    Panel7: TPanel;
    DBGrid3: TDBGrid;
    DataSourceOral: TDataSource;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
```

```
  DBEdit6: TDBEdit;
  Label8: TLabel;
  DBEdit5: TDBEdit;
  Label9: TLabel;
  DBEdit4: TDBEdit;
  Label10: TLabel;
  DBEdit7: TDBEdit;
  Label11: TLabel;
  DBEdit8: TDBEdit;
  Label12: TLabel;
  DBMemo1: TDBMemo;
  Label13: TLabel;
  TableOral: TTable;
  TableOralNum: TAutoIncField;
  TableOralHasta_Num: TIntegerField;
  TableOralTarih: TDateField;
  TableOralGorunum: TStringField;
  TableOralAsimetri: TStringField;
  TableOralTme: TStringField;
  TableOralHijyen: TStringField;
  TableOralPeridont: TStringField;
  TableOralSpesifik: TStringField;
  TableOralOkluzyon: TStringField;
  TableOralDiger: TStringField;
  TableOralAciklama: TStringField;
  DTTarih: TDateTimePicker;
  QueryEvrak: TQuery;
  TableOralSaat: TTimeField;
  Label14: TLabel;
  DBEdit9: TDBEdit;
  procedure FormCreate(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  procedure FormResize(Sender: TObject);
  procedure LoadPos1Click(Sender: TObject);
  procedure SavePos1Click(Sender: TObject);
  procedure Edit1Exit(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure Panel1Enter(Sender: TObject);
  procedure Panel2Enter(Sender: TObject);
  procedure TableAnamnezNewRecord(DataSet: TDataSet);
  procedure TableAnamnezAfterPost(DataSet: TDataSet);
  procedure Panel5Enter(Sender: TObject);
  procedure TableOralNewRecord(DataSet: TDataSet);
  procedure TableOralAfterPost(DataSet: TDataSet);
  procedure TableOralBeforePost(DataSet: TDataSet);
  procedure TableOralAfterScroll(DataSet: TDataSet);
  procedure Edit1DblClick(Sender: TObject);
  procedure Edit2DblClick(Sender: TObject);
  procedure Edit3DblClick(Sender: TObject);
  procedure Panel2Exit(Sender: TObject);
  procedure Panel5Exit(Sender: TObject);
  procedure TableAnamnezBeforePost(DataSet: TDataSet);
private
  { Private declarations }
procedure FilterAyar;
Function EvrakKontrol(tarih:TDatetime) : boolean;
```

```
  public
    { Public declarations }

  end;

var
  FormOrdi: TFormOrdi;
  Hak       : String[7];
  TopTutar   : Real;
  rec       : Tbookmark;



implementation

uses Main, Genel, Query, Data,
  Oral;

{$R *.DFM}

Function TFormOrdi.EvrakKontrol(tarih:TDatetime) : boolean;
begin
QueryEvrak.ParamByName('p_hastanum').Asinteger     := QueryHastanum.asinteger;
QueryEvrak.ParamByName('p_tar1').AsDate     := tarih;
QueryEvrak.ParamByName('p_tar2').AsDate     := tarih;
QueryEvrak.Open;
result:=QueryEvrak.RecordCount>0;
QueryEvrak.Close;

end;

procedure TFormOrdi.FilterAyar;
begin
If (Edit1.Text='') and (Edit2.Text='') and (Edit3.Text='') then
   Exit;
QueryHasta.close;
QueryHasta.Sql.Clear;
QueryHasta.SQL:=Memo1.Lines;
If Edit1.Text<>'' then
   QueryHasta.SQL.Add('WHERE PROTNO='''+Edit1.Text+'''')
else
   If Edit2.Text<>'' then
      begin
      QueryHasta.SQL.Add('WHERE UPPER(ADI) LIKE '''+Edit2.Text+'%''');
      If Edit3.Text<>'' then
         QueryHasta.SQL.Add('AND UPPER(SOYADI) LIKE '''+Edit3.Text+'%''');
      QueryHasta.SQL.Add('ORDER BY ADSOY');
      end
   else
      If Edit3.Text<>'' then
         begin
         QueryHasta.SQL.Add('WHERE UPPER(SOYADI) LIKE '''+Edit3.Text+'%''');
         QueryHasta.SQL.Add('ORDER BY ADSOY');
         end;
QueryHasta.open;
end;

procedure TFormOrdi.FormCreate(Sender: TObject);
begin
```

```
 TableAnamnez.Databasename:=Aliasname;
 TableOral.Databasename:=Aliasname;
 QueryHasta.Databasename:=Aliasname;
 QueryEvrak.Databasename:=Aliasname;


 Hak:=FormMain.Hakbul('ORDI');
 HakKontrol(Hak,5);
 AktifForm:='Ordi';
 If FormData.TableSetScreen.asstring='U' then
   LoadPos(Usernum, 'FormOrdi', FormOrdi)
 else
   LoadPos(-1, 'FormOrdi', FormOrdi);
 DTTarih.Date:=now;

 TableAnamnez.Open;
 TableOral.Open;
 QueryHasta.Open;

FormMain.Navigator.DataSource:=DataSourceHasta;

end;

procedure TFormOrdi.FormActivate(Sender: TObject);
begin
FormMain.Navigator.DataSource:=DataSourceHasta;
AktifForm:='Ordi';
end;

procedure TFormOrdi.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
Action:=caFree;
FormOrdi:=nil;
AktifForm:='';
end;

procedure TFormOrdi.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
begin
If FormMain.Navigator.DataSource=DataSourceAnamnez then
   CanClose := FormMain.KontrolTable(TableAnamnez, ModalResult = mrOK)
else
  If FormMain.Navigator.DataSource=DataSourceOral then
     CanClose := FormMain.KontrolTable(TableOral, ModalResult = mrOK);
end;

procedure TFormOrdi.FormResize(Sender: TObject);
begin
If FormOrdi.Width<>800 then FormOrdi.Width:=800;
If FormOrdi.Height<515 then FormOrdi.Height:=515;
end;

procedure TFormOrdi.LoadPos1Click(Sender: TObject);
begin
If FormData.TableSetScreen.asstring='U' then
   LoadPos(Usernum, Screen.ActiveForm.name, Screen.ActiveForm)
else
   LoadPos(-1, Screen.ActiveForm.name, Screen.ActiveForm);
```

```
end;

procedure TFormOrdi.SavePos1Click(Sender: TObject);
begin
Savepos;
end;

procedure TFormOrdi.Edit1Exit(Sender: TObject);
begin
FilterAyar;
end;

procedure TFormOrdi.FormShow(Sender: TObject);
begin
Edit2.setfocus;
end;

procedure TFormOrdi.Panel1Enter(Sender: TObject);
begin
 FormMain.Navigator.DataSource:=DataSourceHasta;
end;

procedure TFormOrdi.Panel2Enter(Sender: TObject);
begin
 FormMain.Navigator.DataSource:=DataSourceAnamnez;
end;

procedure TFormOrdi.TableAnamnezNewRecord(DataSet: TDataSet);
begin
TableAnamnezTarih.AsDateTime:=now;
If not EvrakKontrol(now) then
   begin
   ShowMessage ('O tarihte kayıtlı açık evrak mevcut değil!');
   FormMain.Navigator.BtnClick(nbcancel);
   end;
end;

procedure TFormOrdi.TableAnamnezAfterPost(DataSet: TDataSet);
begin
If DBISaveChanges((TableAnamnez As TBDEDataSet).Handle)  <>  DBIERR_NONE  Then
   ShowMessage('Anamnez.db : DBISaveChanges işleminde hata ile karşılaşıldı');
end;

procedure TFormOrdi.Panel5Enter(Sender: TObject);
begin
 FormMain.Navigator.DataSource:=DataSourceOral;
end;

procedure TFormOrdi.TableOralNewRecord(DataSet: TDataSet);
begin
TableOralTarih.AsDateTime:=now;
TableOralSaat.AsDateTime:=now;
If not EvrakKontrol(TableOraltarih.AsDateTime) then
   begin
   ShowMessage ('O tarihte kayıtlı açık evrak mevcut değil!');
   FormMain.Navigator.BtnClick(nbcancel);
   end;
DbEdit1.SetFocus;
end;
```

```
procedure TFormOrdi.TableOralAfterPost(DataSet: TDataSet);
begin
If DBISaveChanges((TableOral As TBDEDataSet).Handle) <> DBIERR_NONE Then
  ShowMessage('Oral.db : DBISaveChanges işleminde hata ile karşılaşıldı');
end;

procedure TFormOrdi.TableOralBeforePost(DataSet: TDataSet);
begin
TableOralTarih.AsDateTime:=DTTarih.Date;
If not EvrakKontrol(TableOraltarih.AsDateTime) then
  begin
  ShowMessage ('O tarihte kayıtlı açık evrak mevcut değil!');
  Abort;
  end;
end;

procedure TFormOrdi.TableOralAfterScroll(DataSet: TDataSet);
begin
DTTarih.Date:=TableOralTarih.AsDateTime;
end;

procedure TFormOrdi.Edit1DblClick(Sender: TObject);
begin
QueryHasta.close;
QueryHasta.Sql.Clear;
QueryHasta.SQL:=Memo1.Lines;
QueryHasta.SQL.Add('ORDER BY protno');
QueryHasta.open;
end;

procedure TFormOrdi.Edit2DblClick(Sender: TObject);
begin
QueryHasta.close;
QueryHasta.Sql.Clear;
QueryHasta.SQL:=Memo1.Lines;
QueryHasta.SQL.Add('ORDER BY adi, soyadi');
QueryHasta.open;
end;

procedure TFormOrdi.Edit3DblClick(Sender: TObject);
begin
QueryHasta.close;
QueryHasta.Sql.Clear;
QueryHasta.SQL:=Memo1.Lines;
QueryHasta.SQL.Add('ORDER BY soyadi, adi');
QueryHasta.open;
end;

procedure TFormOrdi.Panel2Exit(Sender: TObject);
begin
If (DatasourceAnamnez.state=dsedit) or (DatasourceAnamnez.state=dsinsert) then
  TableAnamnez.post;
end;

procedure TFormOrdi.Panel5Exit(Sender: TObject);
begin
If (DatasourceOral.state=dsedit) or (DatasourceOral.state=dsinsert) then
  TableOral.post;
```

```
end;

procedure TFormOrdi.TableAnamnezBeforePost(DataSet: TDataSet);
begin
If not EvrakKontrol(Tableanamneztarih.AsDateTime) then
  begin
  ShowMessage ('O tarihte kayıtlı açık evrak mevcut değil!');
  Abort;
  end;
end;

end.
```

...

EK-D. URT Programı Kaynak Kodu:

Default.asp

```
<frameset  rows="50,*">
   <frame name="menu" src="menu.asp" marginwidth="10" marginheight="10" scrolling="auto" frameborder="0">
   <frame name="alan" src="" marginwidth="10" marginheight="10" scrolling="auto" frameborder="0">
</frameset>
```

Menu.asp

```
<table>
<tr>
<td><a href="randevu.asp" target="alan">randevu girişi</a> ..... </td>
<td><a href="randevutalep.asp" target="alan">randevu talep girişi</a></td>
</tr></table>
```

Randevu.asp

```
<!--#include  file="../includes/stitle.asp"-->
     <table cellspacing="0" cellpadding="10" align="center" class= "buyukcerceveci" width="100%" id="sagTABLE" height="96%">
          <tr>
              <td align="center" id="girTD" valign="top">
                    <iframe frameborder="0" src="Randevugir.asp" width="100%" name= "girFRAME" id="girFRAME" scrolling="no"></iframe>
              </td>
            <tr>
          <td id="lisTD" VALIGN="top" ALIGN="center" height="100%" >
                    <iframe frameborder="0" src="Randevulis.asp" name="lisFRAME" id="lisFRAME" scrolling="auto" width="100%"  height="100%"></iframe>
          </td>
</tr>
</table>
<!--#include  file="../includes/sbodyson.asp"-->
```

Randevugir.asp

```
<!--#include  file="sConnection.asp"-->
<!--#include  file="../includes/sfunctions.asp"-->
<!--#include  file="../includes/scombolar.asp"-->
<!--#include  file="../includes/sguven.asp"-->
<!--#include  file="../includes/stitle.asp"-->

<%
s_nonew=0 'standart GİR
'KEYs
'unique key !!!
if replacer(request("xG110_num"))>"" then yG110_num=request("xG110_num") else yG110_num=session(sys_key_1)

if request("kayityapx")>"" then
          sil_hepsi=0
          zorunlularyok=0
          eskiler=0
```

```
		call zorunlular_varmi

		if sil_hepsi=0 and zorunlularyok=1 then
			eskiler=1
			yG110_num=0
		else
			call kayityap
			if sil_hepsi=1 then yG110_num=0
			end if
		end if

'değişkenleri sıfırla
  yG000_Hasta_Num=""
  yG000_Doktor_Num=""
  yO000_Tarih_xgun=""
   yO000_Tarih_xayi=""
   yO000_Tarih_xyil=""
   yG000_RandevuKonusu_Num=""
   yG000_RandevuTalep_Num=""
'
yenici="(yeni kayıt)"
comgoster = 1
kayitid=yG110_num
if 1=1  and yG110_num>0  then
    sql="SELECT  * FROM [Randevu] WHERE 1=1 "
    sql=sql &  " and num = " & yG110_num&""
    set rsgir=db.Execute(sql)
    if not rsgir.eof then
		yenici = ""
		reado = " readonly "
		comgoster = 2
		yG000_Hasta_Num= trim(rsgir("Hasta_Num"))
		yG000_Doktor_Num= trim(rsgir("Doktor_Num"))
		yO000_Tarih_xgun= day(rsgir("Tarih"))
		yO000_Tarih_xayi= month(rsgir("Tarih"))
		yO000_Tarih_xyil= year(rsgir("Tarih"))
		yG000_RandevuKonusu_Num= trim(rsgir("RandevuKonusu_Num"))
		if rsgir("RandevuTalep_Num") >"" then yG000_RandevuTalep_Num=
formatnumber(rsgir("RandevuTalep_Num"),0)
		end if
else
		if eskiler=1 then
		    yG110_num= request.form("xG110_num")
		    yG000_Hasta_Num= request.form("xG000_Hasta_Num")
		    yG000_Doktor_Num= request.form("xG000_Doktor_Num")
		    yO000_Tarih_xgun = request.form("xO000_Tarih_xgun")
		    yO000_Tarih_xayi = request.form("xO000_Tarih_xayi")
		    yO000_Tarih_xyil = request.form("xO000_Tarih_xyil")
		    yG000_RandevuKonusu_Num= request.form("xG000_RandevuKonusu_Num")
		    yG000_RandevuTalep_Num= request.form("xG000_RandevuTalep_Num")
		    kayitid=yG110_num
		    if kayitid>"" then yenici=""
		end if
end if
'
'
'
' programın adı nedir?
programin_basligi =labelci("Randevugir","Randevu Bilgi Girişi")
```

```
'labelleri bulalım
            LG110_num=labelci("num","Num")
            LG000_Hasta_Num=labelci("Hasta_Num","Num")
            LG000_Doktor_Num=labelci("Doktor_Num","Num")
            LO000_Tarih=labelci("Tarih","Tarih")
            LG000_RandevuKonusu_Num=labelci("RandevuKonusu_Num","Num")
            LG000_RandevuTalep_Num=labelci("RandevuTalep_Num","Num")
'
%>

<!--#include  file="../includes/sborgirust.asp"-->

<table <%=sborder%> class=tablogiris align=center cellspacing=0 cellpadding=1 width=100% id=girTABLE>
<form  method="post" name="form_Randevu" id="form_Randevu" onsubmit="return doSubmit(this)">
<input type=hidden value="[Randevu]" name=xtablom id=xtablom size=0 >
<input type=hidden value="<%=yG110_num%>" name=xG110_num id=xG110_num size=0 >
    <tr>
        <td Class=FCTDLabel>
          <label Class=FCLabel id=LG000_Hasta_Num><%=LG000_Hasta_Num%> </label>
        </td>
        <td Class=FCTDForm >
            <SELECT name=xG000_Hasta_Num id=xG000_Hasta_Num class= FCCombos LAB=LG000_Hasta_Num tabindex=1 onchange="comboDiger(this)"  >
              <% response.write combolar("[dishek][Hasta]", "Num", "Adi","", yG000_Hasta_Num, 1,"","",1)  %>
            </SELECT>
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LG000_Doktor_Num><%=LG000_Doktor_Num%></label>
        </td>
        <td Class=FCTDForm >
            <SELECT name=xG000_Doktor_Num id=xG000_Doktor_Num class=FCCombos LAB=LG000_Doktor_Num tabindex=2 onchange="comboDiger(this)"  >
              <% response.write combolar("[dishek][Doktor]","num","Adi","",yG000_Doktor_Num,1,"","",1)  %>
            </SELECT>
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LO000_Tarih><%=LO000_Tarih%></label>
        </td>
        <td Class=FCTDForm >
            <input class=FCDay type=text value="<%=yO000_Tarih_xgun%>" name=xO000_Tarih_xgun id=xO000_Tarih_xgun maxlength=2 LAB=LO000_Tarih tabindex=3 size="20" >
            <input class=FCMonth type=text value="<%=yO000_Tarih_xayi%>" name=xO000_Tarih_xayi id=xO000_Tarih_xayi maxlength=2 LAB=LO000_Tarih tabindex=4 size="20" >
            <input class=FCYear type=text value="<%=yO000_Tarih_xyil%>" name=xO000_Tarih_xyil id=xO000_Tarih_xyil maxlength=4 LAB=LO000_Tarih  tabindex=5 size="20" >
            <img src="../image/takvim.gif" align=absmiddle border=0 class=FCTakvimImg onclick="takvimci('xO000_Tarih','');" >
        </td>
    </tr>
```

```
    <tr>
        <td Class=FCTDLabel>
            <label                                                    Class=FCLabel
id=LG000_RandevuKonusu_Num><%=LG000_RandevuKonusu_Num%></label>
        </td>
        <td Class=FCTDForm >
            <SELECT      name=xG000_RandevuKonusu_Num      id=xG000_RandevuKonusu_Num
class=FCCombos LAB=LG000_RandevuKonusu_Num tabindex=6 onchange="comboDiger(this)"  >
              <%                                                          response.write
combolar("[dishek][RandevuKonusu]","Num","Adi","",yG000_RandevuKonusu_Num,1,"","",1)  %>
            </SELECT>
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label                                                    Class=FCLabel
id=LG000_RandevuTalep_Num><%=LG000_RandevuTalep_Num%></label>
        </td>
        <td Class=FCTDForm >
            <input          class=FCNumber          type=text                    xPrec=10
value="<%=yG000_RandevuTalep_Num%>"                  name=xG000_RandevuTalep_Num
id=xG000_RandevuTalep_Num LAB=LG000_RandevuTalep_Num  tabindex=7 size="20" >
        </td>
    </tr>
<table <%=sborder%> class=tablogiris align=center cellspacing=0 cellpadding=1 width=100%
id=girTABLE>
    <tr>
        <td colspan=2>
    <!--#include  file="../includes/skayitbuton.asp"-->
        </td>
    </tr>
</form>
</TABLE>


<%
sub kayityap_duzeltilecekse_acilacak ()

'
' KAYIT ICIN DEGISKENLERI ALALIM VE VIRGULLE DEGISTIRELIM
yG000_Hasta_Num=replacer(request.form("xG000_Hasta_Num") )
yG000_Doktor_Num=replacer(request.form("xG000_Doktor_Num") )
yO000_Tarih_xgun=request.form("xO000_Tarih_xgun")
yO000_Tarih_xayi=request.form("xO000_Tarih_xayi")
yO000_Tarih_xyil=request.form("xO000_Tarih_xyil")
yO000_Tarih_xst=request.form("xO000_Tarih_xst")
yO000_Tarih_xdk=request.form("xO000_Tarih_xdk")
yG000_RandevuKonusu_Num=replacer(request.form("xG000_RandevuKonusu_Num") )
yG000_RandevuTalep_Num=nokta("0"&request.form("xG000_RandevuTalep_Num"))
yG110_num=request.form("xG110_num")

if request.form("kayityapx")="Kayıt_Sil" then sil=1   ' sil=request("sil")

'
'   KAYIT SILME VE UPDATE ISLEMI
    if yG110_num>0   then
                    if sil=1 then
                            sql = "delete from [Randevu] WHERE num =" & yG110_num&""
                            'response.write (sql)
```

```
                    set ors=db.execute(sql)
                        response.write("<center><span                    class=FCRdelete>Kayıt
Silinmiştir.</span></center>")
                else
                    sql = "update [Randevu] set "
                    sql=sql & "Hasta_Num='"  & yG000_Hasta_Num & "' ,"
                    sql=sql & "Doktor_Num='"  & yG000_Doktor_Num & "' ,"
                    if  yO000_Tarih_xgun>""  then        sql=sql  &  "Tarih='"  &  sqltarih(
right("00"&yO000_Tarih_xgun,2)  &"."&          right("00"&yO000_Tarih_xayi,2)  &"."&
right("0000"&yO000_Tarih_xyil,4) ) & "'  ,"
                        sql=sql & "RandevuKonusu_Num='"  & yG000_RandevuKonusu_Num & "' ,"
                        sql=sql & "RandevuTalep_Num="  & yG000_RandevuTalep_Num & " "
                         sql=sql & " WHERE num=" & yG110_num
                                ' response.write(sql)
                                set ors=db.execute(sql)
                                response.write("<center><span  class=FCRupdate>Kayıt
Düzeltilmiştir.</span></center>")
                        end if
  else
'   YENI KAYIT ISLEMI
                        sql = "insert into [Randevu] ("
                            sql=sql& "Hasta_Num, "
                            sql=sql& "Doktor_Num, "
                             if  yO000_Tarih_xgun>""  and  yO000_Tarih_xayi>""  and
yO000_Tarih_xyil>""  then sql=sql& "Tarih, "
                            sql=sql& "RandevuKonusu_Num, "
                            sql=sql& "RandevuTalep_Num "
          sql=sql&") values ("
                             sql=sql & """ &  yG000_Hasta_Num & "' ,"
                             sql=sql & """ &  yG000_Doktor_Num & "' ,"
                             if  yO000_Tarih_xgun>""  and  yO000_Tarih_xayi>""  and
yO000_Tarih_xyil>""  then  sql=sql  &  """  &  sqltarih(right("00"&yO000_Tarih_xgun,2)  &"."&
right("00"&yO000_Tarih_xayi,2) &"."&  right("0000"&yO000_Tarih_xyil,4) ) & "'  ,"
                             sql=sql & """ &  yG000_RandevuKonusu_Num & "' ,"
                             sql=sql & nokta2(0 + yG000_RandevuTalep_Num) & " "
                    sql=sql & ")"
                    'response.write(sql)
                    set ors=db.execute(sql)
                    response.write("<center><span                    class=FCRinsert>Kayıt
Yapılmıştır.</span></center>")
          end if
end sub

%><!--#include  file="../includes/sborgiralt.asp"-->
   <!--#include  file="../includes/svalidation.asp"-->
   <!--#include  file="../includes/sbodyson.asp"-->
   <!--#include  file="../includes/sgirislem.asp"-->
   <!--#include  file="../includes/skayityap.asp"-->
    <!--#include  file="sConnectionDis.asp"-->
<!-- <<<<<<<<<< END OF SOURCE >>>>>>>>>> -->
```

Randevulis.asp

```
<!--#include  file="sConnection.asp"-->
<!--#include  file="../includes/sfunctions.asp"-->
<!--#include  file="../includes/sguven.asp"-->
<!--#include  file="../includes/stitle.asp"-->
<!--#include  file="../includes/slishtc.asp"-->


<div id=divx style="position:absolute;top:0px;left:0px;"></div>


<%
' BIRDEN FAZLA KEY VAR ISE BU ALT TABLODUR.. O ZAMAN ANA TABLODAN 2. VE 3. KEYLERIN GELMESI GEREKIR
sql="SELECT  [Randevu].num, Y2_Hasta.Adi as Y2_Hasta_Adi, Y3_Doktor.Adi as Y3_Doktor_Adi, [Randevu].Tarih, Y5_RandevuKonusu.Adi as Y5_RandevuKonusu_Adi, [Randevu].RandevuTalep_Num FROM [Randevu]  LEFT OUTER JOIN Hasta as Y2_Hasta  on [Randevu].Hasta_Num=Y2_Hasta.Num LEFT OUTER JOIN Doktor as Y3_Doktor  on [Randevu].Doktor_Num=Y3_Doktor.num LEFT OUTER JOIN RandevuKonusu as Y5_RandevuKonusu on [Randevu].RandevuKonusu_Num=Y5_RandevuKonusu.Num"

 if xlissorgu>"" then
                        sql=sql & " where ("
                        sql=sql & "  convert(nvarchar,Y2_Hasta.Adi )  like '%"&xlissorgu&"%' "
                        sql=sql & "  or  convert(nvarchar,Y3_Doktor.Adi )  like '%"&xlissorgu&"%' "
                        sql=sql & "  or  convert(nvarchar,[Randevu].Tarih)  like '%"&xlissorgu&"%' "
                        sql=sql  &  "  or  convert(nvarchar,Y5_RandevuKonusu.Adi  )  like '%"&xlissorgu&"%' "
                        sql=sql  &  "  or  convert(nvarchar,[Randevu].RandevuTalep_Num)  like '%"&xlissorgu&"%' "
                        sql=sql & ")"
end if

if xlissira>"" then  sql=sql & " ORDER BY [Randevu]." & xlissira & " " & xlisyon

        'response.write sql & "<hr>" :response.flush
        'response.end

'
' programın adı nedir?
listenin_basligi =labelci("Randevulis","Randevu Listesi")
'kolon isimlerini hazırlayalım
        Lsira=labelci("sira","Sıra")
        LG000_Hasta_Num=labelci("Hasta_Num","Num")
        LG000_Doktor_Num=labelci("Doktor_Num","Num")
        LO000_Tarih=labelci("Tarih","Tarih")
        LG000_RandevuKonusu_Num=labelci("RandevuKonusu_Num","Num")
        LG000_RandevuTalep_Num=labelci("RandevuTalep_Num","Num")
'
'
sayfaci=1
perci=10
%>
                <!--#include  file="../includes/slisrecordset.asp"-->

                <div align="center" class="tableContainer" id="data">
                        <div id=sabiteklenti><!-- free code area 001--></div>
```

```
                <!--#include  file="../includes/sborlisust.asp"-->

      <table  <%=sborder%>  align=center id=DataTable cellpadding="0" cellspacing="1">
            <THEAD>
                                        <TR id=theadTR>
                <th t=I id=sabit ><%=Lsira%></th>
                <TH sortCol="Hasta_Num" t=T><%=LG000_Hasta_Num%></TH>
                <TH sortCol="Doktor_Num" t=T><%=LG000_Doktor_Num%></TH>
                <TH sortCol="Tarih" t=D><%=LO000_Tarih%></TH>
                <TH                                                    sortCol="RandevuKonusu_Num"
t=T><%=LG000_RandevuKonusu_Num%></TH>
                <TH sortCol="RandevuTalep_Num" t=I><%=LG000_RandevuTalep_Num%></TH>
                                        </TR>
            </THEAD>
        <%if rs.eof then %>
                                          <!--#include  file="../includes/slisEOF.asp"-->
        <%else ' EOF %>

            <TBODY>

             <%
             k=0
               while not rs.eof and k < rs.pagesize
                 k=k+1
                   if k mod 2=0 then rengim="#eCeCec"
                   if k mod 2=1 then rengim="#cccccc"
             ' ************************************************************************

'************************************************************************
                   response.write "<TR bgcolor=" & rengim & " " & ilkkim & "
onclick=""javascript:win_gonder('" & rs("num") & "', ", "                    )"" title=""Bu kaydı değiştirmek
için tıklayınız.""">" & vbcr
             ' ************************************************************************
                   response.write "<TD class=locked>" & rs.AbsolutePosition & "</TD>" & vbcr '
SIRA
                   'Num
                   response.write "<TD>"& vbcr
                   response.write rs("Y2_Hasta_Adi")& vbcr
                   response.write "</TD>"& vbcr
                   'Num
                   response.write "<TD>"& vbcr
                   response.write rs("Y3_Doktor_Adi")& vbcr
                   response.write "</TD>"& vbcr
                   'Tarih
                   response.write "<TD>"& vbcr
                   response.write rs("Tarih")& vbcr
                   response.write "</TD>"& vbcr
                   'Num
                   response.write "<TD>"& vbcr
                   response.write rs("Y5_RandevuKonusu_Adi")& vbcr
                   response.write "</TD>"& vbcr
                   'Num
                   response.write "<TD align=right>"& vbcr
                   if              rs("RandevuTalep_Num")>""              then              response.write
formatnumber(rs("RandevuTalep_Num"),0)& vbcr
                   response.write "</TD>"& vbcr
                        '
**********************************************************************
```

```
                response.write "</TR>"& vbcr
                rs.movenext
                wend
            %>

        <%end if ' EOF %>

            <TR>
                <TD VALIGN="top" align="center"  id=dummyTD colspan=50></TD>
            </TR>
            </TBODY>
        </TABLE>
                        <!--#include  file="../includes/sborlisalt.asp"-->
            <%' lisin kendisine özel bu alanlar listedeki form'un içinde kullanılacak !
              gir_action="Randevugir.asp"
              form2key1="xG110_num"
            %>
    <!--#include  file="../includes/slisform.asp"-->
  </div>
                        <!--#include  file="../includes/sbodyson.asp"-->
                        <!--#include  file="sConnectionDis.asp"-->
<!-- <<<<<<<<<< END OF SOURCE >>>>>>>>>> -->
```

RandevuTalep.asp

```
<!--#include  file="../includes/stitle.asp"-->
<table cellspacing="0" cellpadding="10" align="center" class="buyukcerceveci" width="100%" id="sagTABLE" height="96%">
        <tr>
                <td align="center" id="girTD" valign="top">
                <iframe frameborder="0" src="RandevuTalepgir.asp" width="100%" name="girFRAME" id="girFRAME" scrolling="no"></iframe>
                </td>
                <td id="lisTD" VALIGN="top" ALIGN="center" height="100%" >
                <iframe frameborder="0" src="RandevuTaleplis.asp" name="lisFRAME" id="lisFRAME" scrolling="auto" width="100%"  height="100%"></iframe>
                </td>
        </tr>
</table>
<!--#include  file="../includes/sbodyson.asp"-->
```

RandevuTalepGir.asp

```
<!--#include  file="sConnection.asp"-->
<!--#include  file="../includes/sfunctions.asp"-->
<!--#include  file="../includes/scombolar.asp"-->
<!--#include  file="../includes/sguven.asp"-->
<!--#include  file="../includes/stitle.asp"-->

<%
s_nonew=0 'standart GİR
'KEYs
'unique key !!!
if replacer(request("xG110_Num"))>"" then yG110_Num=request("xG110_Num") else yG110_Num=session(sys_key_1)

if request("kayityapx")>"" then
```

```
  sil_hepsi=0
  zorunlularyok=0
  eskiler=0
  call zorunlular_varmi

        if sil_hepsi=0 and zorunlularyok=1 then
              eskiler=1
              yG110_Num=0
        else
              call kayityap
              if sil_hepsi=1 then yG110_Num=0
              end if
end if

'değişkenleri sıfırla
  yA000_Adi=""
  yA000_Soyadi=""
  yA000_EPostaAdresi=""
  yA000_Telefon=""
  yA000_CepTel=""
  yG000_RandevuKonusu_Num=""
  yO000_Tarih_xgun=""
   yO000_Tarih_xayi=""
    yO000_Tarih_xyil=""
  yE000_Mesaj=""
  yG000_Doktor_Num=""
'
yenici="(yeni kayıt)"
comgoster = 1
kayitid=yG110_Num
if 1=1  and yG110_Num>0  then
        sql="SELECT  * FROM [RandevuTalep] WHERE 1=1 "
 sql=sql &  " and Num = " & yG110_Num&""

        set rsgir=db.Execute(sql)

        if not rsgir.eof then
              yenici = ""
              reado = " readonly "
              comgoster = 2
           yA000_Adi= trim(rsgir("Adi"))
           yA000_Soyadi= trim(rsgir("Soyadi"))
           yA000_EPostaAdresi= trim(rsgir("EPostaAdresi"))
           yA000_Telefon= trim(rsgir("Telefon"))
           yA000_CepTel= trim(rsgir("CepTel"))
           yG000_RandevuKonusu_Num= trim(rsgir("RandevuKonusu_Num"))
           yO000_Tarih_xgun= day(rsgir("Tarih"))
           yO000_Tarih_xayi= month(rsgir("Tarih"))
           yO000_Tarih_xyil= year(rsgir("Tarih"))
           yE000_Mesaj= trim(rsgir("Mesaj"))
           yG000_Doktor_Num= trim(rsgir("Doktor_Num"))
        end if
else
    if eskiler=1 then
        yG110_Num= request.form("xG110_Num")
        yA000_Adi= request.form("xA000_Adi")
        yA000_Soyadi= request.form("xA000_Soyadi")
        yA000_EPostaAdresi= request.form("xA000_EPostaAdresi")
        yA000_Telefon= request.form("xA000_Telefon")
```

```
            yA000_CepTel= request.form("xA000_CepTel")
            yG000_RandevuKonusu_Num= request.form("xG000_RandevuKonusu_Num")
            yO000_Tarih_xgun = request.form("xO000_Tarih_xgun")
            yO000_Tarih_xayi = request.form("xO000_Tarih_xayi")
            yO000_Tarih_xyil = request.form("xO000_Tarih_xyil")
            yE000_Mesaj= request.form("xE000_Mesaj")
            yG000_Doktor_Num= request.form("xG000_Doktor_Num")
            kayitid=yG110_Num
            if kayitid>"" then yenici=""
            end if
end if
'
'
'
' programın adı nedir?
programin_basligi =labelci("RandevuTalepgir","RandevuTalep Bilgi Girişi")
'labelleri bulalım
            LG110_Num=labelci("Num","Num")
            LA000_Adi=labelci("Adi","Adi")
            LA000_Soyadi=labelci("Soyadi","Soyadi")
            LA000_EPostaAdresi=labelci("EPostaAdresi","EPostaAdresi")
            LA000_Telefon=labelci("Telefon","Telefon")
            LA000_CepTel=labelci("CepTel","CepTel")
            LG000_RandevuKonusu_Num=labelci("RandevuKonusu_Num","Num")
            LO000_Tarih=labelci("Tarih","Tarih")
            LE000_Mesaj=labelci("Mesaj","Mesaj")
            LG000_Doktor_Num=labelci("Doktor_Num","Num")
'
%>
<!--#include  file="../includes/sborgirust.asp"-->
<table <%=sborder%> class=tablogiris align=center cellspacing=0 cellpadding=1 width=100% id=girTABLE>
<form method="post" name="form_RandevuTalep" id="form_RandevuTalep" onsubmit="return doSubmit(this)">
          <input type=hidden value="[RandevuTalep]" name=xtablom id=xtablom size=0 >
         <input type=hidden value="<%=yG110_Num%>" name=xG110_Num id=xG110_Num size=0 >
     <tr>
          <td Class=FCTDLabel>
              <label Class=FCLabel id=LA000_Adi><%=LA000_Adi%></label>
          </td>
          <td Class=FCTDForm >
              <input class=FCText type=text value="<%=yA000_Adi%>" name=xA000_Adi id=xA000_Adi maxlength=20 LAB=LA000_Adi  tabindex=1 size="20" >
          </td>
     </tr>
     <tr>
     <td Class=FCTDLabel>
        <label Class=FCLabel id=LA000_Soyadi><%=LA000_Soyadi%></label>
      </td>
      <td Class=FCTDForm >
          <input class=FCText type=text value="<%=yA000_Soyadi%>" name=xA000_Soyadi id=xA000_Soyadi maxlength=20 LAB=LA000_Soyadi  tabindex=2 size="20" >
          </td>
     </tr>
     <tr>
       <td Class=FCTDLabel>
          <label Class=FCLabel id=LA000_EPostaAdresi><%= LA000_EPostaAdresi%> </label>
          </td>
          <td Class=FCTDForm >
```

```
            <input class=FCText type=text value="<%=yA000_EPostaAdresi%>"
name=xA000_EPostaAdresi id=xA000_EPostaAdresi maxlength=40 LAB=LA000_EPostaAdresi
tabindex=3 size="20" >
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LA000_Telefon><%=LA000_Telefon%></label>
        </td>
        <td Class=FCTDForm >
            <input class=FCText type=text value="<%=yA000_Telefon%>" name=xA000_Telefon
id=xA000_Telefon maxlength=20 LAB=LA000_Telefon tabindex=4 size="20" >
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LA000_CepTel><%=LA000_CepTel%></label>
        </td>
        <td Class=FCTDForm >
            <input class=FCText type=text value="<%=yA000_CepTel%>" name=xA000_CepTel
id=xA000_CepTel maxlength=20 LAB=LA000_CepTel tabindex=5 size="20" >
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel
id=LG000_RandevuKonusu_Num><%=LG000_RandevuKonusu_Num%></label>
        </td>
        <td Class=FCTDForm >
            <SELECT name=xG000_RandevuKonusu_Num id=xG000_RandevuKonusu_Num
class=FCCombos LAB=LG000_RandevuKonusu_Num tabindex=6 onchange="comboDiger(this)" >
              <% response.write
combolar("[dishek][RandevuKonusu]","Num","Adi","",yG000_RandevuKonusu_Num,1,"","",1) %>
            </SELECT>
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LO000_Tarih><%=LO000_Tarih%></label>
        </td>
        <td Class=FCTDForm >
            <input class=FCDay type=text value="<%=yO000_Tarih_xgun%>"
name=xO000_Tarih_xgun id=xO000_Tarih_xgun maxlength=2 LAB=LO000_Tarih tabindex=7
size="20" >
            <input class=FCMonth type=text value="<%=yO000_Tarih_xayi%>"
name=xO000_Tarih_xayi id=xO000_Tarih_xayi maxlength=2 LAB=LO000_Tarih tabindex=8
size="20" >
            <input class=FCYear type=text value="<%=yO000_Tarih_xyil%>"
name=xO000_Tarih_xyil id=xO000_Tarih_xyil maxlength=4 LAB=LO000_Tarih tabindex=9 size="20"
>
            <img src="../image/takvim.gif" align=absmiddle border=0 class=FCTakvimImg
onclick="takvimci('xO000_Tarih','');" >
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LE000_Mesaj><%=LE000_Mesaj%></label>
        </td>
        <td Class=FCTDForm >
```

```
            <textarea class=FCTextarea     name=xE000_Mesaj id=xE000_Mesaj LAB=LE000_Mesaj
tabindex=10 rows="1" cols="20" ><%=yE000_Mesaj%></textarea>
        </td>
    </tr>
    <tr>
        <td Class=FCTDLabel>
            <label Class=FCLabel id=LG000_Doktor_Num><%=LG000_Doktor_Num%></label>
        </td>
        <td Class=FCTDForm >
            <SELECT     name=xG000_Doktor_Num     id=xG000_Doktor_Num     class=FCCombos
LAB=LG000_Doktor_Num tabindex=11 onchange="comboDiger(this)"  >
              <%                                                                    response.write
combolar("[dishek][Doktor]","num","Adi","",yG000_Doktor_Num,1,"","",1)  %>
            </SELECT>
        </td>
    </tr>
<table <%=sborder%> class=tablogiris align=center cellspacing=0 cellpadding=1 width=100%
id=girTABLE>
    <tr>
        <td colspan=2>
    <!--#include  file="../includes/skayitbuton.asp"-->
        </td>
    </tr>
</form>
</TABLE>


<%
sub kayityap_duzeltilecekse_acilacak ()

'
' KAYIT ICIN DEGISKENLERI ALALIM VE VIRGULLE DEGISTIRELIM
yA000_Adi=replacer(request.form("xA000_Adi") )
yA000_Soyadi=replacer(request.form("xA000_Soyadi") )
yA000_EPostaAdresi=replacer(request.form("xA000_EPostaAdresi") )
yA000_Telefon=replacer(request.form("xA000_Telefon") )
yA000_CepTel=replacer(request.form("xA000_CepTel") )
yG000_RandevuKonusu_Num=replacer(request.form("xG000_RandevuKonusu_Num") )
yO000_Tarih_xgun=request.form("xO000_Tarih_xgun")
yO000_Tarih_xayi=request.form("xO000_Tarih_xayi")
yO000_Tarih_xyil=request.form("xO000_Tarih_xyil")
yO000_Tarih_xst=request.form("xO000_Tarih_xst")
yO000_Tarih_xdk=request.form("xO000_Tarih_xdk")
yE000_Mesaj=replacer(request.form("xE000_Mesaj") )
yG000_Doktor_Num=replacer(request.form("xG000_Doktor_Num") )
yG110_Num=request.form("xG110_Num")

if request.form("kayityapx")="Kayıt_Sil" then sil=1   ' sil=request("sil")

'
'   KAYIT SILME VE UPDATE ISLEMI
    if yG110_Num>0   then
                    if sil=1 then
                            sql = "delete from [RandevuTalep] WHERE Num =" & yG110_Num&""
                            'response.write (sql)
                      set ors=db.execute(sql)
                            response.write("<center><span                          class=FCRdelete>Kayıt
Silinmiştir.</span></center>")
                else
```

```
                                sql = "update [RandevuTalep] set "
                                        sql=sql & "Adi='"  & yA000_Adi & "' ,"
                                        sql=sql & "Soyadi='"  & yA000_Soyadi & "' ,"
                                        sql=sql & "EPostaAdresi='"  & yA000_EPostaAdresi & "' ,"
                                        sql=sql & "Telefon='"  & yA000_Telefon & "' ,"
                                        sql=sql & "CepTel='"  & yA000_CepTel & "' ,"
                                        sql=sql & "RandevuKonusu_Num='" & yG000_RandevuKonusu_Num & "' ,"
                                        if yO000_Tarih_xgun>"" then sql=sql & "Tarih='" & sqltarih( right("00"&yO000_Tarih_xgun,2) &"."& right("00"&yO000_Tarih_xayi,2) &"."& right("0000"&yO000_Tarih_xyil,4) ) & "'  ,"
                                        sql=sql & "Mesaj='"  & yE000_Mesaj & "' ,"
                                        sql=sql & "Doktor_Num='"  & yG000_Doktor_Num & "' "
                                                sql=sql & " WHERE Num=" & yG110_Num
                                                ' response.write(sql)
                                                set ors=db.execute(sql)
                                                response.write("<center><span class=FCRupdate>Kayıt Düzeltilmiştir.</span></center>")
                                end if
  else
'   YENI KAYIT ISLEM
                        sql = "insert into [RandevuTalep] ("
                                        sql=sql& "Adi, "
                                        sql=sql& "Soyadi, "
                                        sql=sql& "EPostaAdresi, "
                                        sql=sql& "Telefon, "
                                        sql=sql& "CepTel, "
                                        sql=sql& "RandevuKonusu_Num, "
                                         if yO000_Tarih_xgun>"" and yO000_Tarih_xayi>"" and yO000_Tarih_xyil>"" then sql=sql& "Tarih, "
                                        sql=sql& "Mesaj, "
                                        sql=sql& "Doktor_Num "
            sql=sql&") values ("
                                         sql=sql & "'" &  yA000_Adi & "' ,"
                                         sql=sql & "'" &  yA000_Soyadi & "' ,"
                                         sql=sql & "'" &  yA000_EPostaAdresi & "' ,"
                                         sql=sql & "'" &  yA000_Telefon & "' ,"
                                         sql=sql & "'" &  yA000_CepTel & "' ,"
                                         sql=sql & "'" &  yG000_RandevuKonusu_Num & "' ,"
                                         if yO000_Tarih_xgun>"" and yO000_Tarih_xayi>"" and yO000_Tarih_xyil>"" then sql=sql & "'" & sqltarih(right("00"&yO000_Tarih_xgun,2) &"."& right("00"&yO000_Tarih_xayi,2) &"."&  right("0000"&yO000_Tarih_xyil,4) ) & "'  ,"
                                         sql=sql & "'" &  yE000_Mesaj & "' ,"
                                         sql=sql & "'" &  yG000_Doktor_Num & "' "
                        sql=sql & ")"
                                'response.write(sql)
                                set ors=db.execute(sql)
                                response.write("<center><span class=FCRinsert>Kayıt Yapılmıştır.</span></center>")
            end if
end sub

%><!--#include file="../includes/sborgiralt.asp"--><!--#include file="../includes/svalidation.asp"--><!--#include file="../includes/sbodyson.asp"--><!--#include file="../includes/sgirislem.asp"--><!--#include file="../includes/skayityap.asp"--><!--#include file="sConnectionDis.asp"--><!-- <<<<<<<<<< END OF SOURCE >>>>>>>>>> -->
```

RandevuTalepLis.asp

```
<!--#include  file="sConnection.asp"-->
<!--#include  file="../includes/sfunctions.asp"-->
<!--#include  file="../includes/sguven.asp"-->
<!--#include  file="../includes/stitle.asp"-->
<!--#include  file="../includes/slishtc.asp"-->


<div id=divx style="position:absolute;top:0px;left:0px;"></div>


<%
' BIRDEN FAZLA KEY VAR ISE BU ALT TABLODUR.. O ZAMAN ANA TABLODAN 2. VE 3. KEYLERIN GELMESI GEREKIR
sql="SELECT [RandevuTalep].Num, [RandevuTalep].Adi, [RandevuTalep].Soyadi, [RandevuTalep].EPostaAdresi, [RandevuTalep].Telefon, [RandevuTalep].CepTel, Y7_RandevuKonusu.Adi as Y7_RandevuKonusu_Adi, [RandevuTalep].Tarih, [RandevuTalep].Mesaj, Y10_Doktor.Adi as Y10_Doktor_Adi FROM [RandevuTalep]  LEFT OUTER JOIN RandevuKonusu as Y7_RandevuKonusu on [RandevuTalep].RandevuKonusu_Num=Y7_RandevuKonusu.Num LEFT OUTER JOIN Doktor as Y10_Doktor  on [RandevuTalep].Doktor_Num=Y10_Doktor.num"

  if xlissorgu>"" then
                              sql=sql & " where ("
                              sql=sql & "  convert(nvarchar,[RandevuTalep].Adi)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].Soyadi)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].EPostaAdresi)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].Telefon)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].CepTel)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,Y7_RandevuKonusu.Adi  )  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].Tarih)  like '%"&xlissorgu&"%' "
                              sql=sql  &  "  or  convert(nvarchar,[RandevuTalep].Mesaj)  like '%"&xlissorgu&"%' "
                              sql=sql & "  or  convert(nvarchar,Y10_Doktor.Adi )  like '%"&xlissorgu&"%' "
                              sql=sql & ")"
end if

if xlissira>"" then  sql=sql & " ORDER BY [RandevuTalep]." & xlissira & " " & xlisyon

        'response.write sql & "<hr>" :response.flush
        'response.end

'
' programın adı nedir?
listenin_basligi =labelci("RandevuTaleplis","RandevuTalep Listesi")
'kolon isimlerini hazırlayalım
        Lsira=labelci("sira","Sıra")
        LA000_Adi=labelci("Adi","Adi")
        LA000_Soyadi=labelci("Soyadi","Soyadi")
        LA000_EPostaAdresi=labelci("EPostaAdresi","EPostaAdresi")
        LA000_Telefon=labelci("Telefon","Telefon")
        LA000_CepTel=labelci("CepTel","CepTel")
```

```
        LG000_RandevuKonusu_Num=labelci("RandevuKonusu_Num","Num")
        LO000_Tarih=labelci("Tarih","Tarih")
        LE000_Mesaj=labelci("Mesaj","Mesaj")
        LG000_Doktor_Num=labelci("Doktor_Num","Num")
'
'
sayfaci=1
perci=10
%>
                <!--#include  file="../includes/slisrecordset.asp"-->

                <div align="center" class="tableContainer" id="data">
                        <div id=sabiteklenti><!-- free code area 001--></div>

                <!--#include  file="../includes/sborlisust.asp"-->

      <table  <%=sborder%>  align=center id=DataTable cellpadding="0" cellspacing="1">
            <THEAD>
                                        <TR id=theadTR>
                 <th t=I id=sabit ><%=Lsira%></th>
                 <TH sortCol="Adi" t=T><%=LA000_Adi%></TH>
                 <TH sortCol="Soyadi" t=T><%=LA000_Soyadi%></TH>
                 <TH sortCol="EPostaAdresi" t=T><%=LA000_EPostaAdresi%></TH>
                 <TH sortCol="Telefon" t=T><%=LA000_Telefon%></TH>
                 <TH sortCol="CepTel" t=T><%=LA000_CepTel%></TH>
                 <TH                                    sortCol="RandevuKonusu_Num"
t=T><%=LG000_RandevuKonusu_Num%></TH>
                 <TH sortCol="Tarih" t=D><%=LO000_Tarih%></TH>
                 <TH sortCol="$$noSort$$" t=T><%=LE000_Mesaj%></TH>
                 <TH sortCol="Doktor_Num" t=T><%=LG000_Doktor_Num%></TH>
                                        </TR>
            </THEAD>
        <%if rs.eof then %>
                                          <!--#include  file="../includes/slisEOF.asp"-->
        <%else ' EOF %>

            <TBODY>

             <%
             k=0
                while not rs.eof and k < rs.pagesize
                  k=k+1
                    if k mod 2=0 then rengim="#eCeCec"
                    if k mod 2=1 then rengim="#cccccc"
             ' ************************************************************************


'************************************************************************
                    response.write "<TR bgcolor=" & rengim & " " & ilkkim & "
onclick=""javascript:win_gonder('" & rs("Num") & "', ", "                )"" title=""Bu kaydı değiştirmek
için tıklayınız.""> " & vbcr
             ' ************************************************************************
                    response.write "<TD class=locked>" & rs.AbsolutePosition & "</TD>" & vbcr '
SIRA
                    'Adi
                    response.write "<TD>"& vbcr
                    response.write rs("Adi")& vbcr
                    response.write "</TD>"& vbcr
                    'Soyadi
```

```
                    response.write "<TD>"& vbcr
                    response.write rs("Soyadi")& vbcr
                    response.write "</TD>"& vbcr
                    'EPostaAdresi
                    response.write "<TD>"& vbcr
                    response.write rs("EPostaAdresi")& vbcr
                    response.write "</TD>"& vbcr
                    'Telefon
                    response.write "<TD>"& vbcr
                    response.write rs("Telefon")& vbcr
                    response.write "</TD>"& vbcr
                    'CepTel
                    response.write "<TD>"& vbcr
                    response.write rs("CepTel")& vbcr
                    response.write "</TD>"& vbcr
                    'Num
                    response.write "<TD>"& vbcr
                    response.write rs("Y7_RandevuKonusu_Adi")& vbcr
                    response.write "</TD>"& vbcr
                    'Tarih
                    response.write "<TD>"& vbcr
                    response.write rs("Tarih")& vbcr
                    response.write "</TD>"& vbcr
                    'Mesaj
                    response.write "<TD>"& vbcr
                    response.write " <div class=""FCTextareaDiv""> "& vbcr
                    response.write rs("Mesaj")& vbcr
                    response.write "</div>"& vbcr
                    response.write "</TD>"& vbcr
                    'Num
                    response.write "<TD>"& vbcr
                    response.write rs("Y10_Doktor_Adi")& vbcr
                    response.write "</TD>"& vbcr
                        '
*************************************************************************
                    response.write "</TR>"& vbcr
                    rs.movenext
                    wend
                %>

            <%end if ' EOF %>

                <TR>
                    <TD VALIGN="top" align="center"  id=dummyTD colspan=50></TD>
                </TR>

                </TBODY>
            </TABLE>

                    <!--#include  file="../includes/sborlisalt.asp"-->
                <%' lisin kendisine özel bu alanlar listedeki form'un içinde kullanılacak !
                  gir_action="RandevuTalepgir.asp"
                  form2key1="xG110_Num"
                %>
        <!--#include  file="../includes/slisform.asp"-->
    </div>
            <!--#include  file="../includes/sbodyson.asp"-->
            <!--#include  file="sConnectionDis.asp"-->
<!-- <<<<<<<<<< END OF SOURCE >>>>>>>>>> -->
```

sConnection.asp

```
<%
	Set db = Server.CreateObject("ADODB.Connection")
	db.ConnectionTimeOut = 1200 ' 1200/60sec=20 min
	db.CommandTimeOut = 240 ' 240/60sec=4 min
	db.ConnectionString = "Provider=SQLOLEDB; Data Source=ONDERGOCER; Initial Catalog=dishek; User ID=sa;pwd=onder"
	dbConnectionString = db.ConnectionString
	db.open
%>
```

sConnectionDis.asp

```
<%
	db.close
	set db=nothing
%>
```

sTitle.asp

```
<!--#include file="sConstants.asp"-->

<%
	server.scripttimeout=600
	'response.write "<h1>" & server.scripttimeout & "</h1>"
	Response.Expires = 0
	Response.Expiresabsolute = Now() - 1
	Response.CacheControl = "no-cache"
	Response.AddHeader "pragma","no-cache"
	Response.AddHeader "cache-control","private"
	session.timeout=180

' response.write request.servervariables("URL") & " - "& Session.SessionID &"<hr>"
'bukim=replace(lcase(request.servervariables("URL")),application("urlbasi"),"")
'bukim=request("ll")

if request("ll")>"" then
		helpcix=request("ll")&".asp"
	else
	helpcix=replace(lcase(request.servervariables("URL")),application("urlbasi"),"")
  end if
bukim=helpcix

'response.write "  url:"&request.servervariables("URL")
'response.write "  urlB:"&application("urlbasi")
'response.write "  B:"&bukim

tit01="XES"
	select case bukim
				case "default.asp"	tit01="XES / Netisis Bilgi Teknolojileri"
	end select


title="XESV1"'  /  " & session("user_FullName")
helpicon="<a style=""cursor: help;"" onClick=""javascript:helpgoster()""><img src=""image/helpicon.gif""  alt=""HELP DOSYASI ICIN TIKLAYINIZ"" border=0></a>"


 if request.form("xlissorgu")>"" then xlissorgu=trim(replacer(request.form("xlissorgu")))
```

```
 if request.form("xlissira")>""  then xlissira =trim(replacer(request.form("xlissira")))
 if request.form("xlisyon")>""   then xlisyon  =trim(replacer(request.form("xlisyon")))

%>
<SCRIPT language="JavaScript">
function                                                                                                    helpgoster()
        {penakan("/SystemPrograms/Help/ik_help.asp?xhelp_program=<%=helpcix%>","dbhelpci",825
,520) ;}
</SCRIPT>


<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-9" />
<META http-equiv="Content-Type" content="text/html; charset=windows-1254" />
<META http-equiv="Content-Language" content="TR" />
<TITLE><%=title%></TITLE>

<LINK rel="StyleSheet" type="text/css" href="styles/<%=s_css%>" />

</head>

<body leftmargin=0 topmargin=0 class=x01BODY >
<script>
//so backspace doesn't go back
document.onkeydown = checkForBackspace;
function checkForBackspace()
{
        //we can backspace in a textbox
        if(!(window.event.srcElement.tagName=='INPUT'                                                   ||
window.event.srcElement.tagName=='TEXTAREA' ))
        {
                if(window.event && window.event.keyCode == 8)
                {
                        // try to cancel the backspace
                        window.event.cancelBubble = true;
                        window.event.returnValue = false;
                        return false;
                }
        }
}
</script>
```

sBodyson.asp

```
</BODY>
</HTML>
```

sfunctions.asp

```
<%
'gloabal imagelar için.
printBackround="printzemin.jpg"
printLogo="Ppolis.gif"
'response.write ("tansel")

s_clientTitle= " AVC Danışmanlık / Istanbul "
s_appShortName = " Netisis Genel Amaçlı Raporlama Sistemi </font>"
s_netisisCopyright = "Netisis Bilgi Teknolojileri Ltd. Şti."
```

```
'#########################################################################

'#########################################################################
function yazresponse()
	response.write "<hr>"
	for each a in request.form
	   response.write "form:" & a & "--" & request.form(a) &"<br>"
	next
	for each a in request.querystring
	   response.write "querystring:" & a & "--" & request.querystring(a) &"<br>"
	next
	response.write "<hr>"
			for each a in request.servervariables
				xbody=xbody & a&"="&request.servervariables(a)&"<br>"
			next
			response.write xbody & "<br>"
end function
'#########################################################################

'#########################################################################
' TIRNAKLARI VS DEGISTIRMEK ICNI !!
Function replacer(gelen)
	gelen	=Replace(gelen, Chr(39), "&#39;")
	gelen	=Replace(gelen, Chr(45)&Chr(45), "&#45;&#45;")
	gelen	=Replace(gelen, Chr(60), "&#60;")
	gelen	=Replace(gelen, Chr(62), "&#62;")
	'gelen	=Replace(gelen, Chr(39), "&#39;")
	'gelen	=Replace(gelen, Chr(39), "&#39;")
	replacer=gelen
End Function
'#########################################################################

'#########################################################################
'para cevırıcı !
function nokta(a)
	'response.write "<h1>gelen:"&a&"</h1>"
	if len(trim(a))=0 or isnull(a) then a=0
	yborc=cstr(a)
	yborc=replace(yborc," ","")
	yborc=replace(yborc,".","")
	yborc=replace(yborc,",",".")
			if not isnumeric(yborc) then yborc=0
			'if yborc>10000000000000000 then yborc=0
	if yborc>"" then
	     tutar	=yborc
	 else
	     tutar	=0
	end if
	nokta=tutar
			'response.write "<h1>("&tutar&")</h1>"
end function
'#########################################################################

'#########################################################################
'para cevırıcı !
function nokta2(a)
	nokta2=nokta(a)
end function
'#########################################################################
```

```
'#########################################################################
'tarih cevırıcı !
Function sqltarih(sqltar2)
'response.write "<h1>"&sqltar&"</h1>"
'response.end
        if sqltar2<>"00.00.0000" and sqltar2<>"00.00.0000 00:00" then
                sqltar=cdate(sqltar2)
           sqltarih = trim ( right("0000"&year(sqltar),4) & right("00"&Month(sqltar),2) &
right("00"&day(sqltar),2) & " " & right("00"&hour(sqltar),2)  & ":" & right("00"&minute(sqltar),2)  )
        else
           sqltarih = "NULL"
        end if

End Function 'sqltarih

Function sqltarih2 (yil,ay,gun)
        if yil>"" and ay>"" and gun>"" then
           sqltarih2 = trim ( right("0000"&yil,4) & right("00"&ay,2) & right("00"&gun,2) )
        else
           sqltarih2 = ""
        end if

End Function 'sqltarih

'#########################################################################

'#########################################################################
'labelleri bulalım ve hazırlayalım ' lisan bilgisi zaten sessiondan geliyor ... yoksa default language
kullanılacak ! o da yoksa, yanındaki bilgiler
function labelci(geldegis , geldeger)
        scriptnamex= split(request.servervariables("SCRIPT_NAME"),"/")
        scriptname=scriptnamex(ubound(scriptnamex))
        if session("lisan")>"" then
                lisan=cint(session("lisan"))
                session("lisan")=cstr(lisan) ' 0 var ise başında atalım !
        else
                session("lisan")="1"
                lisan="1"
        end if

        'application("L_" & lisan & scriptname & "_" & geldegis ) = ""
        label_nedir = application("L_" & lisan & scriptname & "_" & geldegis )

        if label_nedir > "" then
                        labelci = label_nedir
                else
                        on error resume next 'DB de hata falan varsa yada table yoksa bana sorun
çıkartmasın !!!
                                sqllng_label=                                "SELECT top 1 *
FROM sLnglabel WHERE lng_program='" & scriptname &"' and  lng_label = '" & geldegis &"' and
lng_lng='" & lisan & "'  "
                                sqllng_label=sqllng_label & "union "
                                sqllng_label=sqllng_label & "SELECT top 1 * FROM sLnglabel
WHERE                                                                lng_label = '"
& geldegis &"' and lng_lng='" & lisan & "'  "
                                set rslng_label=db.Execute(sqllng_label)
                                if not rslng_label.eof then
                                        ' kendisini bulduysak kullanalım !
```

```
                        geldeger = rslng_label("lng_value")
        else
                                ' eger kendisi yoksa default labelcilere bakalım !!!
                                sqllng_label2="SELECT * FROM sLnglabel WHERE
lng_program="' & scriptname &"' and  lng_label = '" & geldegis &"' and lng_lng='1'  "
                                set rsdefault=db.Execute(sqllng_label2)

                                if rsdefault.eof then  ' eger defaultda yoksa
                                        ' eger degişkenler bulunmamışsa, default
lisan için programdan gelenleri yazalım !
                                sqllng_insert="INSERT INTO sLnglabel
(lng_label, lng_value, lng_lng, lng_program) VALUES ('" & geldegis &"','"&geldeger&"','1','" &
scriptname & "') "
                                set rsdefault_yaz=db.Execute(sqllng_insert)

                                        ' bu lisan için de defaultları yazalım ve
adamı bundan sonrası için yormayalım
                                sqllng_insert="INSERT INTO sLnglabel
(lng_label, lng_value, lng_lng, lng_program) VALUES ('" & geldegis
&"','"&geldeger&"','"&lisan&"','" & scriptname & "') "
                                set rsdefault_yaz=db.Execute(sqllng_insert)
                                end if
                        end if
                Err.Clear
                On Error GoTo 0

                labelci=geldeger
                application("L_" & lisan & scriptname & "_" & geldegis )=geldeger
        end if
end function

'#####################################################################

' ************************************************************************************
' ************************************************************************************
'DETAY SATIRLARDAKI ZORUNLU ALANLAR VAR MI?
sub detay_zorunlu_varmi()
                for each coklu in request.form
                if zorunlularda_problemvar=0 then
                        if left(coklu,boykey) = boykeyx then
                tmp_ek=replace(coklu,boykeyx,"") '
                        if tmp_ek<>"_0" then
                                if  mid(coklu,3,1)="1"  or  mid(coklu,3,1)="2"  then  '
zorunluysa !!!
                                if instr("ABCDEF", mid(coklu,2,1)) then
                                        if request.form(coklu)>"" then
                                                tmp=1
                                        else
                                                zorunlularda_problemvar=1

                                        end if
                                end if
                                if instr("GHIJKLMNUOPQ", mid(coklu,2,1)) then
                                        zorunlularda_problemvar=1
                                        if      mid(coklu,3,1)="1"      AND
request.form(coklu) => 0 then zorunlularda_problemvar=0
                                        if      mid(coklu,3,1)="2"      AND
request.form(coklu) > 0  then zorunlularda_problemvar=0
                                end if
```

```
                                                                    if zorunlularda_problemvar=1 then
response.write "<h3>("&tmp_ek&") numaralı satırda kayıt için yeterli bilgi girilmemiş veya sorun var...</h3>"
                                                        end if
                                                end if
                                        end if ' tmp_ek
                                        end if ' zorunlularda_problemvar=0
                                        next
end sub 'detay_zorunlu_varmi()
' ************************************************************************************
' ************************************************************************************


'#####################################################################
function alert(x)
alert="<script language=""JScript"">alert('"&replace(x,"'","$")&"');</script>"
response.write alert
end function
'#####################################################################

'#####################################################################
%>
<SCRIPT LANGUAGE="JavaScript">
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
function penakan(website,webname,winwidth,winheight)   // BU FONKSOIYONU MODULA ILE DEGISTIR !!!
{
        var heightspeed = 10; // vertical scrolling speed (higher = slower)
        var widthspeed = 15;  // horizontal scrolling speed (higher = slower)
        var leftdist = 0;    // distance to left edge of window
        var topdist = 0;     // distance to top edge of window
        if (document.all)
        {
                //var winwidth = window.screen.availWidth - leftdist;
                //var winheight = window.screen.availHeight - topdist;
                var sizer = window.open("",webname,"left=" + leftdist + ",top=" + topdist + ",width=1,height=1,scrollbars=yes");
                for (sizeheight = 10; sizeheight < winheight; sizeheight += heightspeed)
                {
                        sizer.resizeTo("1", sizeheight);
                }
                for (sizewidth = 10; sizewidth < winwidth; sizewidth += widthspeed)
                {
                        sizer.resizeTo(sizewidth, sizeheight);
                }
                sizer.location = website;
        }
        else
        {
                window.location = website;
                window.focus();
        }
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
function takvimci(fld, xEk)
```

```
{
	if (!xEk) {xEk=''} //else {alert(xEk)}
	var gidenGunAyYil = new Array;
	gidenGunAyYil['yuklealani'] = fld;

	var xyegun =document.getElementById(gidenGunAyYil['yuklealani'] + "_xgun" + xEk)
	var xyeay =document.getElementById(gidenGunAyYil['yuklealani'] + "_xayi" + xEk)
	var xyeyil=document.getElementById(gidenGunAyYil['yuklealani'] + "_xyil" + xEk)

	gidenGunAyYil['yuklegun'] = xyegun.value;
	gidenGunAyYil['yukleay'] = xyeay.value;
	gidenGunAyYil['yukleyil'] = xyeyil.value;

	//screenY = document.body.offsetHeight
	//screenX = window.screen.availWidth
	tempX = event.screenX - 100
	tempY = event.screenY + 8
	var dialogTakvim =
showModalDialog('../includes/stakvimci.asp',gidenGunAyYil,'dialogHeight:180px;dialogWidth:180px;di
alogTop:'+tempY+'px;dialogLeft:'+tempX+'px;status:no;resizable:no;help=no');
	if (dialogTakvim)
	{
		xyegun.value=dialogTakvim['vgun'];
		xyeay.value=dialogTakvim['vay'];
		xyeyil.value=dialogTakvim['vyil'];

		// Tansel : Burada eğer saat ve dakka varsa şimdiki zamanı koymak için
		var xyesaat = document.getElementById(gidenGunAyYil['yuklealani'] +
"_xst")
		var xyedakika = document.getElementById(gidenGunAyYil['yuklealani'] +
"_xdk")
		var datetime = new Date();
		if (xyesaat)
		{
			var hour = datetime.getHours() ;
			xyesaat.value = hour
		}
		if (xyedakika)
		{
			var minute = datetime.getMinutes();
			xyedakika.value = minute
		}
	}
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
//##########################################################
//bunu kendim icin kullancagım silmeyin!
function tellMe(yesNo)
{
	var ben=self
	var ustum=window.parent
	var icindeBulundugumEleman=null
	var ustumunicindeBulunduguEleman=null
	var ustumunWidthi=0
	if (yesNo==1) //burda, yukarıdakı degişkenler değişiyor dikkat et!
	{
		do
```

```
                        {
                                if
(ustum.document.getElementById(ben.name).parentElement.tagName!="TD")

        icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement.parentE
lement.id
                                else

        icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement.id

                                alert('benim iframe\'mimin adı= ' + ben.name + '\n' +
'ustumdeki iframe ise adı= ' + ustum.name +
                                        '\n benim, üstümde içinde bulunduğu
eleman ise ' + icindeBulundugumEleman)

                                ben=ustum
                                ustum=ben.parent
                        }
                        while (ben!=ustum)
        }       //if (tellme=1) //burda, yukarıdakı degişkenler değişiyor dikkat et!
}
//bunu kendim icin kullancagım silmeyin!
//################################################################

//################################################################
// burası lisASP'deki lisFORM'u submit yapabilmek için gerekti.
function win_gonder(gelenkey1, gelenkey2, gelenkey3)
{
        var kimeGit=eval('self.parent.' + document.lisFORM.target)
        if (kimeGit)
        {
                kimeGit.opener=self
                if (document.lisFORM.key1) lisFORM.key1.value=gelenkey1
                if (document.lisFORM.key2) lisFORM.key2.value=gelenkey2
                if (document.lisFORM.key3) lisFORM.key3.value=gelenkey3
                //alert(document.lisFORM.action.indexOf('?'))
                if (document.lisFORM.action.indexOf('?')==-1)
                        document.lisFORM.action=document.lisFORM.action                 +
'?lisvar=<%=request.querystring("lisvar")%>'
                setTimeout('lisFORM.submit();',250)
        }
}
// burası lisASP'deki lisFORM'u submit yapabilmek için gerekti.
//################################################################
////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////
function gonder(gelen)
{
        //alert(personelx.action.substr(0,11) + 'teesstt')
        personelx.filenamegir.value=gelen.fnamegir;
        personelx.filenamelis.value=gelen.fnamelis;
        //alert (personelx.hrPersonel_id.value)
        if (personelx.filenamelis.value>"")
                personelx.action=personelx.action.substr(0,11)+ '?lisvar=1'
        else
                personelx.action=personelx.action.substr(0,11)+ '?lisvar=0'
        personelx.submit();
}
```

```
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
function gonderimmi(gelen)
{
	if (event.keyCode==13) hrPersonelFORM.submit()
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
function PrintLis(mailOrPrinter)
{
	if (document.all)
		if (document.all.DataTable)
		{
			//getLisContent();
			var subWindow
=window.open('about:blank','def','width=800,height=600,nomenu,scrollbars=yes,status=yes,toolbar=no,di
rectories=no,menubar=no,resizable=yes')
			subWindow.document.write(getLisContent())
			subWindow.moveTo(0,0)
			subWindow.focus();
			if (subWindow.confirm("Bu sayfayı yazdırmak istiyor
musunuz?")){subWindow.print()}
		}
		else
		{
		return false;
		}
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
function sendMailOnClient()
{
   //SET MESSAGE VALUES
   var to = escape("email adresini yazınız!");
	//var cc = "";
	//var bcc = "";
   var subject = "A Preformatted Email Message";
   var body =
      "Chandler,\n\n\tI'm sorry, I can't make it tonight.  " +
      "I have to rearrange my sock drawer. " +
      "\n\nSincerely,\n\nMonica"

   // BUILD MAIL MESSAGE COMPONENTS
   var doc = "mailto:" + to +
	//"?cc=" + cc +
	//"&bcc=" + bcc +
      "&subject=" + escape(subject) +
      "&body=" + getLisContent();

   // POP UP EMAIL MESSAGE WINDOW
	//var hr=window.open()
	top.location = doc;
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
```

```
function getLisContent()
{

                        var icerik5 = ""
                        icerik5=icerik5 + "                                <html><head><meta HTTP-
EQUIV=\"content-type\" CONTENT=\"text/html; charset=UTF-8\"><title>Liste İçin Print</title>"
                        icerik5=icerik5          +          "          <body          topmargin=50
background=\"./image/<%=printBackround%>\"  style=\"background-repeat:  no-repeat;background-
position: center;\">"
                        icerik5=icerik5 + "<table cellspacing=3 border=0 width=\"100%\"
align=center style=\"font-size:8pt;font-family:tahoma;\">"
                        icerik5=icerik5 + "<tr><td colspan=50 style=\"font-size:15px;font-
weight:bold;color:#398ECE ;\" align=center>"
                        icerik5=icerik5 + document.all.baslikSPAN.innerText
                        icerik5=icerik5 + "</td></tr>"
                        icerik5=icerik5 + "<tr><td colspan=50 style=\"font-size:15px;font-
weight:bold;color:#398ECE ;\" align=center>"
                        icerik5=icerik5 + "<%=session("bakilan_tamAdi")%>"
                        icerik5=icerik5 + "</td></tr>"
                        icerik5= icerik5 + document.all.HeaderTable.innerHTML
                        icerik5= icerik5 + document.all.spen.innerHTML
                        icerik5 = icerik5.replace('<THEAD>','')
                        icerik5 = icerik5.replace('</THEAD>','')
                        icerik5 = icerik5.replace('<TBODY>','')
                        icerik5 = icerik5.replace('</TBODY>','')
                        icerik5 = icerik5.replace('<TABLE class=data','<TA BLE class=data')


                        var regexpForWidth =/width/gi
                        icerik5=icerik5.replace(regexpForWidth,' wi dth ')
                        icerik5=icerik5 + "<center><hr><img src=\"/image/<%=printLogo%>\"
width=140 height=100 ><div style=\"font-size:8pt;font-family:tahoma;\"><%=s_clientTitle%><br><div
style=\"font-size:10px\"><%=s_appShortName%></div></center>"

                        var regexpForOnClick ="/onclick=\"javascript:win_gonder/g"
                        icerik5=icerik5.replace(regexpForOnClick,'')


                        var kapaIcerik="</td></tr></table>"
                        var temizIcerik=icerik5      + kapaIcerik
//                      alert(temizIcerik)
                        return temizIcerik;
}
////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////
function checkSayiControl()
{
        var whichCode = event.keyCode;
        //alert(whichCode)
        strCheck=",8,9,13,35,36,37,38,39,40,46,48,49,50,51,52,53,54,55,56,57,96,97,98,99,100,101,102
,103,104,105,"
        var whichCode2 = ',' +whichCode+ ',';
        if (strCheck.indexOf(whichCode2) == -1) return false;  // Not a valid key
}
////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////
```

```
// Tansel : Burada bir kolonu readonly yaparsak yani mevcutlardan gelince update edilemesin dersek bu fonksiyonu kullanıyoruz.
function FormReadOnly(formEleman)
{
	alert ('Bu alanı değiştirmeye yetkiniz yok!');
	formEleman.blur();
}
////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
//TOBALA İÇİN EKLENENLER
//Removes all content from list box
function RemoveAll(targetCombo)
{
	//alert(targetCombo.name)
	var itargetComboLength = targetCombo.options.length;
	for (var i = 0; i < itargetComboLength; i++)
		targetCombo.options.remove(1);
	targetCombo.selectedIndex = 0;
}

//Get SUBoptions for the selected options
function GetOptions(tbl, valCol, txtCol, whereCol, whereColVal, maxCombo)
{
	var szRequest = "tbl=" + tbl + "&valCol=" + valCol  + "&txtCol=" + txtCol + "&whereCol=" + whereCol + "&whereColVal=" + whereColVal + "&maxCombo=" + maxCombo  ;
	var objHTTP = new ActiveXObject("Microsoft.XMLHTTP");
	var szURL = "./includes/sTobala.asp";
	var szHttpMethod = "POST";
	objHTTP.Open(szHttpMethod, szURL, false);
	objHTTP.SetRequestHeader("Content-Type", "application/x-www-form-urlencoded");
	objHTTP.Send(szRequest);

	var szReply = objHTTP.ResponseText;
	//alert (szReply)
	if (objHTTP.status != 200)
	{
		//failure
		szReply = "";
	}
	//alert('szReply=' + szReply)
	return szReply;
}

//Display options in the targetCombo
function DisplayOptions()
{
	var sourceCombo = event.srcElement
	var targetCombo = document.getElementById(sourceCombo.targetCombo)

	targetCombo.selectedIndex = -1;

	RemoveAll(targetCombo)
	var whereValue = sourceCombo.options[sourceCombo.selectedIndex].value
	var xmlOptions = GetOptions(targetCombo.tbl, targetCombo.valCol, targetCombo.txtCol, targetCombo.whereCol, whereValue, targetCombo.maxCombo);
	var objXmlDom = new ActiveXObject("Microsoft.XMLDOM");
	if (!objXmlDom.loadXML(xmlOptions))
	{
```

```
            var sErr = "Response XML String is messed up\n" + xmlOptions;
                    //alert(sErr);
                    clearOtherOptions(targetCombo)
            }
            else
            {
                    var nodes = objXmlDom.getElementsByTagName("z:row");

                    for (var i = 0; i < nodes.length; i++)
                    {
                            //alert(i)

            targetCombo.add(addOption(nodes[i].getAttribute(targetCombo.valCol),nodes[i].getAttribute(ta
rgetCombo.txtCol)));
                    }
                    if (nodes.length>0)
                    {
                            if (nodes[0].getAttribute('diger')=='b')
                            {
                                    var Opt=addOption('-1000','<%=labelci("Diğer","Diğer")%>')
                                    Opt.setAttribute('tabloadi', targetCombo.tbl)
                                    Opt.setAttribute('tabloID', targetCombo.valCol)
                                    Opt.setAttribute('tabloValue', targetCombo.txtCol)
                                    Opt.setAttribute('tabloWhere',    targetCombo.whereCol    +    '=\"    +
whereValue +'\")
                                    Opt.style.color='blue'
                                    targetCombo.add(Opt);
                            }
                    }
                    clearOtherOptions(targetCombo)
            }
}

function clearOtherOptions(combo)
{
            tutCombo=document.getElementById(combo.targetCombo)
            while(tutCombo)
            {
                    RemoveAll(tutCombo)
                    tutCombo=document.getElementById(tutCombo.targetCombo)
            }
}
function addOption(val, txt)
{
                            var objOption = document.createElement("option");
                            objOption.value = val;
                            objOption.text = txt;
                            //objOption.setAttribute('aa','asd')
                            return objOption;
}
//TOBALA İÇİN EKLENENLER
////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
//Detay Table için Eklenenler
function noHeaderScroll(gelen)
{           gelen.scrollTop = '0px'
}
function scrollHeader(gelen,gelen1)
```

```
{	gelen.scrollLeft = gelen1.scrollLeft }
//////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////
</script>

<!--Sfunctions bitti-->
```

sCombolar.asp

```
<%
	'KOMBOLARI AYARLAYALIM
	function combolar(comtablo, comalanid, comalanadi,comiptal ,comgelen, comgoster, comustbilgi, comwhere, comseciniz)
		'response.write "<h1>giden comtablo:"&comtablo&"</h1>"
		comDBadi=""
		nokta_nerde=instr(comtablo,"][")
		if nokta_nerde>0 then
			comDBadi=mid(comtablo,1,nokta_nerde-1)
			comDBadi=replace(comDBadi,"[","")
			comtablo=replace(comtablo,"["&comDBadi&"]","")
		end if
		'db degiştirme işlemi !!!#####################################
		' Kılıççççççç
		tansel_iptal = 0
		if tansel_iptal=10 then
			set rs_using_db=db.Execute("select DB_NAME(0)")
			using_db_name=rs_using_db(0)
			if comDBadi>"" then
				set rs_using_db=db.Execute("use " & comDBadi)
			end if
		end if
		' ############################################################
		'on error resume next 'DB de hata falan varsa yada table yoksa bana sorun çıkartmasın !!!
	'combomulti=0
		'if comgoster = 3 then comgoster = 1 : combomulti=1
		if comalanadi="" then comalanadi=comalanid
				' bu satır silinecek !!! unutmaaaaaaaaaaaa
				' veysel gecici olarak ekledi !!!
				' çalışmaya başlayınca kalkacak !
			application("combo_"+comtablo+"_"+comalanid)=""
			sqlcombolarA=""
			sqlcombolarB=""
			A=0 : X1=""
			B=0 : X2=""
			if application("combo_" & comtablo & "_" & comalanid) >""then
				'replace yapıp gelen bilgiyi yakalıycas
				'aşagıda aynısından bi tane daha var.. sadece if için eklendi

		temp_combo=application("combo_" & comtablo & "_" & comalanid)
		'alert ("combo_" & comtablo & "_" & comalanid & " varmış !")
			'response.write "<h1>aaaaaaaaaa</h1>"
			else
				' Tansel burada max combo sayısını öğreniyoruz.
				comboBoyMax = 3
```

```
                                                    sqlcombolar="SELECT distinct top
"&comboBoyMax&" "&comalanid &",("& comalanadi & ") as kombo_gostericisi "
                                                    if comiptal >"" then
sqlcombolar=sqlcombolar & "," & comiptal
                                                    if comustbilgi >"" then
sqlcombolar=sqlcombolar & "," & comustbilgi
                                                    sqlcombolar=sqlcombolar & " FROM " &
comtablo
                                                    tmpkeyTip =
tipNedir(comalanid,comtablo)
                                                        if tmpkeyTip=1 then
sqlcombolar=sqlcombolar & " where " & comalanid & ">" "
                                                        if tmpkeyTip=2 then
sqlcombolar=sqlcombolar & " where " & comalanid & ">0 "
                                                        if tmpkeyTip=3 then
sqlcombolar=sqlcombolar & " where " & comalanid & ">'19000101' "
                                                    sqlcombolar=sqlcombolar & " and (" &
comalanadi & ") > " "
                                                    if comwhere>"" then
sqlcombolar=sqlcombolar & " and " & comwhere & " "
                                                    if comgoster=2 and comgelen>"" then
sqlcombolar = sqlcombolar & " and " & comalanid & "=" & comgelen & """
                                                    sqlcombolar=sqlcombolar & " order by
kombo_gostericisi asc "
                                                    'alert (sqlcombolar)
                                                        'response.write
"<hr>"&sqlcombolar&"<hr>" : response.flush':response.end
                                                set rscombolar=db.Execute(sqlcombolar)

                                                if tmpkeyTip = 1 then tmpSeciniz ="""""""""
                                                if tmpkeyTip = 2 then tmpSeciniz =""""0""""
                                                if tmpkeyTip = 3 then tmpSeciniz =""""19000101""""
                                                temp_combo = "<OPTIONo
value="&tmpSeciniz&" "&ustbilgi&" class=select01ilk >netisis_seciniz</OPTION>" & vbcrlf
                                                while not rscombolar.eof
                                                    combolar_id=trim(rscombolar(0))
                                                    combolar_adi=trim(rscombolar(1))
                                                        'if isnull(combolar_adi) then
combolar_adi="Seçiniz"
                                                        ttci=""
                                                        if comiptal>"" then if
rscombolar(comiptal)=true then ttci="t"
                                                        ustbilgi=""
                                                        if comustbilgi>"" then if
rscombolar(comustbilgi)>"" then ustbilgi=" ustbilgi="""&rscombolar(comustbilgi)&""""
                                                        temp_combo = temp_combo &
"<OPTION"&ttci&" value="""&combolar_id&""" "&ustbilgi&">"&combolar_adi&"</OPTION>" &
vbcrlf
                                                    rscombolar.movenext
                                                    wend
                                                    ' Tansel : Burada combo boyunu tuan
değişken ile maksimumu hesaplıyoruz.
                                                        comboDiger = 0
                                                        sql_Count = "select count
("&comalanid&") as MaxComboBoyu from "&comtablo&""
                                                            sql_Count = sql_Count
& " where 1 = 1 "
                                                        if comwhere>"" then
sql_Count = sql_Count & " and " & comwhere & " "
```

```
                                                                    if comgoster=2 and comgelen>"" then sql_Count = sql_Count & " and " & comalanid & "='" & comgelen & "'"
                                                                    'alert (sql_Count)

                                                                    set rs_Count=db.Execute(sql_Count)
                                                                    if rs_Count("MaxComboBoyu") > comboBoyMax then
                                                                        'alert (rs_Count("MaxComboBoyu"))
                                                                        comboDiger = 1
                                                                        temp_combo = temp_combo & "<OPTION value=""-1000"" style=""color:blue;"" "&ustbilgi&" name=""gidecekDegerler"" "
                                                                        temp_combo = temp_combo & " tabloadi=""&comtablo&"" tabloID=""&comalanid&"" tabloValue=""&comalanadi&"" tabloWhere=""&comwhere&"">"&labelci("Combo_select_Diger", "Diğer")&"</OPTION>" & vbcrlf
                                                                    end if
                                                                ' Tansel : Burada combo boyunu tuan değişken ile maksimumu hesaplıyoruz.

                                                                application("combo_" & comtablo & "_" & comalanid)=temp_combo
                                                                'alert ("combo_" & comtablo & "_" & comalanid & " oluştu !")
                                                            end if


                                                            ' buraya kadar comboları ya db den yada app den aldık hazırldık.. şimdi işleyelim ...
                                                            temp_combo=application("combo_" & comtablo & "_" & comalanid)
                                                            temp_combo=replace(temp_combo, "<OPTION value="&comgelen&" " , "<OPTION value="&comgelen&" selected " )
                                                            temp_combo=replace(temp_combo, "<OPTIONt value="&comgelen&" " , "<OPTION value="&comgelen&" selected " )

                                                            if comgelen>"" then

        comgelen_multi=split(comgelen,", ")
                                                                if UBound(comgelen_multi)>0 then
                                                                    for i=0 to UBound(comgelen_multi)

        temp_combo=replace(temp_combo, "<OPTION value=" & comgelen_multi(i) & " " , "<OPTION value=" & comgelen_multi(i) & " selected " )

        temp_combo=replace(temp_combo, "<OPTIONt value=" & comgelen_multi(i) & " " , "<OPTION value=" & comgelen_multi(i) & " selected " )
                                                                    next
                                                                end if
                                                            end if

                                                            'seçiniz lafını değiştirmek istiyorsan 1 yada 2 gonder yada ne eklersen, 0 gelirse dokunma !!!
```

```
                                        if comseciniz>0 and not ( comgoster=2 and comgelen>"" ) then
                        if comseciniz=1 then comsecinizx="Seçiniz..."
                        if comseciniz=2 then comsecinizx="Boş olanlar..."

                                                temp_combo=replace(temp_combo, "netisis_seciniz" , labelci("Combo_select_"&comseciniz , comsecinizx) )
                                                        temp_combo=replace(temp_combo, "<OPTIONo" , "<OPTION" )
                                        end if

                                        ' Burada eğer gelen combo max boyundan combo sayısı fazla ise diğer kolonu koymak için.
                                        if comboDiger = 1 then
                                                'alert ("comboDiger")
                                        end if


                                        combolar=temp_combo

                                Err.Clear
                                'On Error GoTo 0


                                'db degiştirme işlemi !!!#####################################
                                if tansel_iptal=10 then
                                        set rs_using_db=db.Execute("use " & using_db_name)
                                end if
                                ' ################################################################
                        end function



                        function tipNedir(a,b)
                                'Tansel : Burada gelen kolonun tipine göre whreri yağpıyoruz
                                        'Key olamayacak alanlar Asp karşılıkları
                                        'bit=11,image=205,text=201,ntext=203
                                        'Key olacak alanlar Asp karşılıkları
                                        'Sayı ---- smallint=2,int=3,real=4,float=5,money:smallmoney=6,tinyint=17,bigint=20,decimal:numeric=131
                                        'Text ---- sql_variant=12,binary:timestamp=128,char=129,nchar=130,varchar=200,nvarchar=202,varbinary=204
                                        'Tarih ---- datetime:smalldatetime=135
                                        sqlKolonType = " select " & a & " from " & b & " where (1 = 2) "
                                        'response.write sqlKolonType
                                        set rs_type = db.Execute(sqlKolonType)
                                        keyTip = 1
                                        For Each fldF In rs_type.Fields
                                                if fldf.type = 2 or fldf.type = 3 or fldf.type = 4 or fldf.type = 5 or fldf.type = 6 or fldf.type = 17 or fldf.type = 20 or fldf.type = 131 then keyTip = 2
                                                if fldf.type = 135 then keyTip = 3
                                        Next
                                        tipNedir = keyTip
                        end function
%>
<script>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
// Tansel : Burası combo boyları max derğeri geçince çalışacak olan script
function comboDiger(gelenCombo)
{
	var dialogAcildi = 0;

	if (gelenCombo.value == '-1000')
	{
		var sonucGonder = new Array;
		var gidecekDegerler = gelenCombo.options[gelenCombo.options.length-1];
		sonucGonder['gelenCombo']=gelenCombo;
		//Bunlar serverside olduğu için querysistring ile gönderiyorum.
		ComboSqlTabloAdi = gidecekDegerler.tabloadi;		// Sorgu yapılacak SQL tablo adı
		ComboSqlID = gidecekDegerler.tabloID;			// Sorgu yapılacak SQL tablo adı
		ComboSqlValue = gidecekDegerler.tabloValue;		// Sorgu yapılacak SQL tablo adı
		ComboSqlValueLabel = 'Daha Olmadı';			// Sorgu yapılacak SQL tablo adı
		ComboZorunluDeger = gidecekDegerler.tabloWhere;	// Sorgu yapılacak kolonlar
		tansel=0;
		if (tansel==1){alert (ComboSqlTabloAdi);alert (ComboSqlID);alert (ComboSqlValue);alert (ComboZorunluDeger);}

		// Tansel : Bunu eğer diolog açılmış ve direk kapatılmış ise kullanıyoruz
		dialogAcildi = 1;

		var dialog1 = showModalDialog('./includes/ssearchCombo.asp?sComboSqlTabloAdi='+ComboSqlTabloAdi+'&sComboSqlID='+ComboSqlID+'&sComboSqlValue='+ComboSqlValue+'&sComboSqlValueLabel='
			+ComboSqlValueLabel+'&sComboZorunluDeger='+ComboZorunluDeger+"

	,sonucGonder,'dialogHeight:170px;dialogWidth:305px;status:no;resizable:yes;help=no');
	}
	if (dialog1)
	{
		gelenKacEleman = dialog1['gelenKacEleman']
		selectedEleman = dialog1['seciliOlan']
		boy= dialog1['gelenCombo'].length-1;
		//alert (dialog1['gelenCombo'].options[boy].value)
		tutselekt = dialog1['gelenCombo'].options[boy].cloneNode(true);
		for(i0=boy; i0>0; i0--)
		{
			//alert (dialog1['gelenCombo'].options[i0].value)
			dialog1['gelenCombo'].options[i0] = null;
		}
		// Burada dönen değerlere göre yeni optionları oluşturalım.
		//alert (gelenKacEleman)
		x=0
		for(i1=1;i1<gelenKacEleman+1;i1++)
		{
			dialog1['gelenCombo'].options[i1] = new Option(dialog1['gidenDegerler'+x+''],dialog1['gidenIDler'+x+''], false, false);
			if (selectedEleman == dialog1['gidenIDler'+x+''])dialog1['gelenCombo'].options[i1].selected=true;
			x++
		}
		dialog1['gelenCombo'].insertBefore(tutselekt)
```

```
	}
	else
	{
		// Tansel : Bunu eğer diolog açılmış ve direk kapatılmış ise kullanıyoruz
		if (dialogAcildi != 0 )
		{
			// Burada combo direk kapatılınca seçilmeden önceki değerine döndürüyoruz
			if (gelenCombo.seciliOption){gelenCombo.selectedIndex = gelenCombo.seciliOption;}
			else {gelenCombo.selectedIndex = 0 ;}
		}
	}
	gelenCombo.seciliOption = gelenCombo.selectedIndex;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
</script>
```

sborgirust.asp

```
<table align="center" CELLPADDING="0" CELLSPACING="0"  <%=sborder%>  id=girBorTABLE>
 <tr id=girBorTD>
    <td width="5px"><img src="image/lissol.gif"  alt="" border=0></td>
    <td background="image/lisortaust.gif" align=center>
		<table align=center cellspacing=0 cellpadding=1>
			<tr>
				<td align=left><%=helpicon%></td>
				<td align=center class=girBaslik style="color:black;" nowrap>
					<b><%=programin_basligi%></b>
					<font color="#0066cc">
						<%
						if yenici>"" then
							response.write yenici
							else
							if kayitid>"" then
								response.write "(" & kayitid & ")"
							end if
						end if
						%>
					</font>
				</td>
				<td align=right >
				<%
				'Buraya menüden "lisvar=1" geliyor buda kontrol edilecek.
				if s_nonew=0 then%>
					<INPUT class=buttonyeni type="button" value="<%=labelci("button_newrecord","Yeni Kayıt") %>" onClick="yeni_kayit()" >
					<SCRIPT language=javascript>
						function yeni_kayit()
						{
						var hangi_url	=self.location.href
						var hangi_ek	=self.location.search
						var kok_url = hangi_url.replace(hangi_ek,"")
						//alert(kok_url) ;
						self.location.href=kok_url ;
						}
					</SCRIPT>
				<% End If %>
```

```
                                        </td>
                                    </tr>
                                </table>
    </td>
    <td><img src="image/lissag.gif"  alt="" border=0></td>
  </tr>
  <tr>
    <td background="image/lissolo.gif" ></td>
    <td>
</table>
<!-- ********************************  -->
```

skayitbuton.asp

```
<table  align=center cellspacing=0 cellpadding=0>
        <tr >
        <td  align=center>
        <input class=button0 type="submit" name="kayityapx" id="kayityapx" value="Kayıt için tıklayınız">
        </td>
        <td  align=right>
        <% if s_nonew=0 then
        if yenici="" then %>
                <input type="submit" name="kayityapx" id="kayityapx" value="Kayıt_Sil" class="buttonsil" onClick="return kontrolcu_iptal();">
                <%      end if
                        end if %>
                </td>
        </tr>
</table>
```

sborgiralt.asp

```
    <td background="image/lissago.gif" width=5px><img src="image/lissago.gif" border=0></td>
  </tr>
  <tr>
    <td background="image/lissola.gif" ><img src="image/blank.gif" width=1 height=5 alt="" border=0></td>
    <td background="image/lisortaalt.gif"><img src="image/blank.gif" width=1 height=5 alt="" border=0></td>
    <td background="image/lissaga.gif"><img src="image/blank.gif" width=1 height=5 alt="" border=0></td></tr></TABLE>
```

sborlisust.asp

```
<table CELLPADDING="0" CELLSPACING="0" border=0 align=center id=anatablo>
  <tr id=baslikTR>
    <td width=5px id=sabit><img src="image/lissol.gif"  alt="" style="z-index:231;"></td>
    <td background="image/lisortaust.gif" align=center  valign=middle id=baslik>
                <span id=baslikSPAN class="lisBaslik"><b><%=listenin_basligi%></b></span>
                   
                <IMG SRC="image/lisprinter.gif" BORDER=0 ALT="<%=labelci("lisprint","Sayfayı yazdırmak için tıklayınız.")%>" ALIGN="ABSMIDDLE" onclick="PrintLis();" style="CURSOR: hand;">
                <IMG SRC="image/lisopennew.gif" id="openInNewWindow" BORDER=0 ALT="<%=labelci("lisnewwindow","Sayfayı yeni pencerede açmak için tıklayınız.")%>" ALIGN="ABSMIDDLE" onclick="openInNewWindow();" style="CURSOR: hand;">
```

```
                <IMG SRC="image/lismail.gif" id="emailgonder" BORDER=0
ALT="<%=labelci("lisemail","Listeyi e-mail ile göndermek için tıklayınız.")%>"
ALIGN="ABSMIDDLE" onclick="sendMailOnClient();" style="CURSOR: hand;">
                </td>
    <td id=baslik><img src="image/lissag.gif"  alt="" border=0></td>
 </tr>
 <tr>
    <td background="image/lissolo.gif" class=locked></td>
    <td>
```

svalidation.asp

```
<script>

var tutgrup="
var controlled="
var mustci="<font color=red>*</font>"
var msg="
var msgForDate="
function InitializeValidation() //misal (divTasarim,style)
{
// burda attacheventler koyuyoruz....ayrıca
//                      xL109 ilk(0) x traş...
//                       ||||
//                       1234
//          1=TipAdı
//          2=must mı degil mi??
//          3=Key...beni ilgilendirmiyo sanırım
//          4=Küsürat(Scale)....toplamı=precision ....o da xPrec de geliyor bana zaten...
//          Bunu da getcharAt(elm, whichChar) ile alıyoruz...
          for (var i=0;i<document.forms.length;i++)
          {
                    eval(document.forms[i])
                    for (var ii=0;ii<document.forms[i].elements.length;ii++)
                    {
                              elm=document.forms[i].elements[ii]
                              prnt = elm.parentNode
                              //bunu yeni ekledim multi insert için...bi eleman'ın TR'si eger style.display
none ise valide etmicez..
                              //buna ek olarak style.display none olan hiçbiseyi valide etmicez artık (bunu
atlamısız)
                              while (prnt.tagName!='TR')
                              {
                                        prnt = prnt.parentNode
                              }
                              if (prnt.style.display=='none') { elm.gizli = 'evet'}  // bunu elm gizli diye
cevirdim cunku master satırda form elemanını display none yaparsam kopyaladıklarımda da cıkmıyo..
                              if (elm.id)
                              {
                                        if (elm.id!='kayityapx' && elm.id!='xtablom' && elm.type!='hidden'
&& elm.type!='submit'  && elm.style.display!='none'  && elm.gizli!='evet' )
                                        {
//                                                  alert(elm.id + '---' + elm.noChek)
                                                  if (elm.parentNode.className=='FCTDFormD')
                                                  {

          noKhaki='this.parentNode.style.backgroundColor=\'khaki\'"
```

```
                                        clearKhaki                                      =
'this.parentNode.style.backgroundColor=\'\'"
                                }
                                else
                                {
                                        noKhaki="
                                        clearKhaki = "
                                }
                                elm.onfocus     =     addEvent(elm,     'onfocus',
'this.style.backgroundColor=\'khaki\'; '+ noKhaki +';', 'text')
                                elm.onblur      =      addEvent(elm,      'onblur',
'this.style.backgroundColor=\'\'; '+ clearKhaki +';', 'text')

                                tipNo=getTypex(elm.id,1)
                                if (tipNo==49) tipNo=75
                                if (tipNo==50) tipNo=67
                                if (tipNo==51) tipNo=79
                                if (!eval(data_tipleri[tipNo])) tipNo=87
//                              alert(eval(data_tipleri[tipNo]))
                                tipAdi=data_tipleri[tipNo][0]
//                              alert(elm.id + '----' + getcharAt(elm, 2)+ '----tipno=' + tipNo)
//                              alert('form adı=' +document.forms[i].name + '\neleman adi='
+ elm.name + '\n tipNo=' + tipNo + '\n tipAdi=' + tipAdi)
                                if (tipNo==79 || tipNo==80) // datetime or smalldatetime
                                        {
                                        elm.onkeydown = addEvent(elm, 'onkeydown',
checkInt.toString(), 'function')
//                                      elm.attachEvent("onkeydown", checkInt)
                                        elm.onkeyup     =     addEvent(elm,     'onkeyup',
TabNext.toString(), 'function')
//                                      elm.attachEvent("onkeyup", TabNext)
                                        elm.onblur      =      addEvent(elm,      'onblur',
checkDate.toString(), 'function')
//                                      elm.attachEvent("onblur", checkDate)
//                                      return false
                                        }
                                if ((tipNo>70 && tipNo<79) ||(tipNo==90))       ////90
dediğimiz Z..sadece YIL !
                                        {
                                                elm.onkeyup = addEvent(elm, 'onkeyup',
checkDecimals.toString(), 'function')
//                                              elm.attachEvent("onkeyup",
checkDecimals)
                                                elm.onblur = addEvent(elm, 'onblur',
checkRange.toString(), 'function')
//                                              elm.attachEvent("onblur", checkRange)
                                                elm.onkeydown      =      addEvent(elm,
'onkeydown', checkInt.toString(), 'function')
//                                              elm.attachEvent("onkeydown", checkInt)
//                                              alert(elm.maxLength)
                                                if (elm.maxLength!='2147483647')
                                                        {

        elm.detachEvent("onkeyup",checkDecimals);    //Bu detach RA BAKMAM GEREKEBİLİR
TANSEL !!!! 20.07.2005
                                                        }
                                                else
                                                        {
```

```
        kusurNokta=eval(getcharAt(elm, 4))
                                                                topUzunluk=elm.xPrec ?
eval(elm.xPrec) : 0

        kalanNoktali=eval(topUzunluk-kusurNokta)
                                                                if      (tipNo==77      ||
tipNo==78) kalanNoktali=kalanNoktali-2

        kalanNoktali2=Math.round((kalanNoktali/3)-0.49)
//                                                              if
(kalanNoktali/3==Math.round(kalanNoktali/3))    alert(elm.id    +    kalanNoktali/3    +'    eşittir'    +
Math.round(kalanNoktali/3))
                                                                if
(kalanNoktali/3==Math.round(kalanNoktali/3)) kalanNoktali2=kalanNoktali2-1
//                                                              alert('kalanNoktali2='  +
topUzunluk +'-' + kusurNokta + '---->' + kalanNoktali2)
//                                                              kalanNoktali2

        kusurNokta2=(kusurNokta>0) ? 1 : 0
//
        alert(elm.id+'='+kusurNokta2+'\ntopUzunluk=' + topUzunluk)

        elm.maxLength=(eval(kalanNoktali2+kusurNokta2+kalanNoktali+kusurNokta))
                                                                var punto= new Number
(elm.currentStyle.fontSize.replace('pt',''))

                                                                var     punto2     =new
String(elm.maxLength*punto)
                                                                if (elm.type!='radio' &&
elm.type!='checkbox')elm.style.width=punto2 + 'px'

//                                                              alert(kalanNoktali2+  '--'
+ kusurNokta2+ '--' + kalanNoktali+ '--' + kusurNokta)
                                                                var
labci=eval('document.all.' + elm.LAB)

        labciText=labci.innerText
//
        labci.insertAdjacentHTML('beforeEnd',  topUzunluk  +  '--'  +  kusurNokta  +  '---'  +
(elm.maxLength))
                                                                        }
                                                                 //maxlength gelmişse !!!
        //                                                      return false
                                                        }
                                                if (getcharAt(elm, 2)==1)
                                                        {
                                                                var labci=eval('document.all.' + elm.LAB)
                                                                labciText=labci.innerText
//                                                              alert(labciText  +  '----'  +  getcharAt(elm,
2)+ '----tipno=' + tipNo)
                                                                if (labciText.indexOf('*')==-1)
                                                                        {

        labci.insertAdjacentHTML('beforeEnd',mustci)
                                                                        }
                                                        }

                                        }
```

```
                                        else
                                        {
                                        }
                                }
                                //else{alert('bu elemanın adı yok')}
                        }
                }
}

function checkDate()
{
        elm=event.srcElement
        elmName=new String(elm.id)


        if (elm.value!='')
        {
                if (elmName.lastIndexOf('_xgun')>-1)
                {
                        if (elm.value<1 || elm.value>31)
                        {
                                alert('Tarih girerken GÜN değeri 1-31 arasında olmalıdır!')
                                elm.focus();elm.select();return false;
                        }
                }
                if (elmName.lastIndexOf('_xayi')>-1)
                {
                        if (elm.value<1 || elm.value>12)
                        {
                                alert('Tarih girerken AY değeri 1-12 arasında olmalıdır!')
                                elm.focus();elm.select();return false;
                        }
                }
                if (elmName.lastIndexOf('_xyil')>-1)
                {
                        if (elm.value<1900 || elm.value>2079)
                        {
                                alert('Tarih girerken YIL değeri 1900-2079 arasında olmalıdır!')
                                elm.focus();elm.select();return false;
                        }
                }
        }
//      alert(ne)
}
function allDate (dayObj, monthObj, yearObj)
{
        if (dayObj.value.length>0 || monthObj.value.length>0 || yearObj.value.length>0)
        {
                if      (!(dayObj.value.length>0        &&      monthObj.value.length>0      &&
yearObj.value.length>0))
                return('lutfen tarih bilgisini gun/ay/yil olarak giriniz!!')
        }
        return ('')
}
function validateDate (dayObj, monthObj, yearObj)
{
        //alert(34234)
        if (dayObj.value.length>0 && monthObj.value.length>0 && yearObj.value.length>0)
        {
```

```
                    var Gun, Ay, Yil;
                    Gun=(dayObj.name.lastIndexOf('_xgun')>-1) ? 'gun' : ''
                    Ay=(monthObj.name.lastIndexOf('_xayi')>-1) ? 'ayi' : ''
                    Yil=(yearObj.name.lastIndexOf('_xyil')>-1) ? 'yil' : ''

//                  alert('Gun=' + Gun + '\n' + 'Ay=' + Ay + '\n' + 'Yil=' + Yil)
                    if (Gun=='gun' && Ay=='ayi' && Yil=='yil')
                    {
                            neGun=dayObj.value
                            neAy =monthObj.value
                            neYil=yearObj.value
                            var  strDate= neAy+'-'+neGun+'-'+neYil;
//                          alert(strDate)
                            var objDate = new Date (strDate);
                      if ((neAy-1) != objDate.getMonth() || neGun != objDate.getDate() || neYil != objDate.getFullYear())
                       {
//                              alert(dayObj.id)
                              dayObj.style.backgroundColor='red'
                              dayObj.focus();
                              dayObj.select();
                              return( lbladi(dayObj.LAB) + ' Bilgisinde hata varxx!')
                            }
                    }
            }
return ('');
}

function checkRange()
{
//                  alert('checkRange');
            var deger=''
            var elm9=event.srcElement
            tipNo=getTypex(elm9.id,1)
            if (!eval(data_tipleri[tipNo])) tipNo=87

//          alert(tipNo)
            tipAdi=data_tipleri[tipNo][0]
            Lower=data_tipleri[tipNo][2]
            Upper=data_tipleri[tipNo][3]
            var degerTxt=new String(elm9.value)
//          var regexpForI =/,|\.| /g
            var regexpForI =/\.| /g
            var degerTxt2=degerTxt.replace(regexpForI,'')
            regexpForI =/\,| /g
            degerTxt2=degerTxt2.replace(regexpForI,'.')
//          alert(degerTxt2 + '----' + Lower + '--' + Upper)
            deger=new Number(degerTxt2)
//          alert(Lower + '-'+ deger +'-' + Upper)
            if (deger<Lower || deger>Upper || isNaN(deger))
            {
                    alert('Lütfen ' + lbladi(elm9.LAB) +' alanına ' + Lower + ' ile ' + Upper + ' arasında değer giriniz!')
                    elm9.value='';
            }
}
```

```
function getTypex(txt, whichChar)
{
		var txt2=new String(txt)
		kod=txt2.charCodeAt(whichChar) //burda asci koduna gore elemanın array içindeki yerını buluyoruzz.
		return kod;
}
function getcharAt(elm, whichChar)
{
		txt=new String(elm.id)
		kod=txt.charAt(whichChar) //burda asci koduna gore elemanın array içindeki yerını buluyoruzz.
//		alert(kod + '----' + event.keyCode)
		return kod;
}
function getType(txt)
{
	if (txt.length>3)
	{
		if (txt!='kayityapx' && txt!='xtablom')
		{
			var txt2=new String(txt)
	//		alert(txt2.charCodeAt(1) + '---->' + txt2.charAt(1) + '---->' + txt2)
			kod=txt2.charCodeAt(1)
			neym=data_tipleri[kod][0]
			neym1=data_tipleri[kod][1]
			neym2=data_tipleri[kod][2]
			neym3=data_tipleri[kod][3]
	//		alert(kod + '---->' + neym+ '---->' + neym1+ '---->' + neym2+ '---->' + neym3+ '---->' + txt2)
			if (neym=='numeric')
			{
				if (Math.pow(10,37)>neym3) alert('buyuktur')
			}
		}
	}
}
function doSubmit(form) //misal (divTasarim,style)
{
	eval(form)
	var result=''
	msg=''
	msgForDate=''
	var controlled=''
	var fokusElm=''
	var theCollection = eval(form.elements)
	for (var i=0;i<theCollection.length;i++)
	{
		elm=theCollection[i]
//		alert (elm.id + '\n' + getcharAt(elm, 2))
		if ((controlled!='') && (fokusElm==''))
		{
//			alert('ilk oltaya takılan=' + controlled.replace(',','')+'Q');
			fokusElm=controlled.replace(',','')
			fokusElm='x' + fokusElm.substr(1,fokusElm.length)
//			alert(fokusElm)
		}
		result += i + '.eleman = ' + elm.id
				+ '---tipi=' + elm.type
```

```
                                + '---tagName=' + elm.tagName
                                + '---LAB=' + elm.LAB
                                + '---value=' + elm.value+'\n'
                tipNo=getTypex(elm.id,1)
                ////burası eger adam tum tarih bilgilerini (gun, ay , yıl) doldurmamıssa calısacak..alan zorunlu olmak zorunda degil...
                Gun=elm.id.lastIndexOf('_xgun')>-1
                //alert(Gun + '--' + elm.id)
                if ((tipNo==79 || tipNo==80) && Gun && (elm.style.display!='none') && (elm.gizli!='evet')) // datetime or smalldatetime
                {
//                              alert('BURASI TARİHIN GUN TARAFI !!!')
                                msgForDate += allDate(theCollection[i], theCollection[i+1], theCollection[i+2])
                                msgForDate += validateDate(theCollection[i], theCollection[i+1], theCollection[i+2])
                }
                ////burası eger adam tum tarih bilgilerini (gun, ay , yıl) doldurmamıssa calısacak..alan zorunlu olmak zorunda degil...
//              alert(elm.id + '---' + elm.type)
                if ((getcharAt(elm, 2)==1) && (elm.type!='hidden') && (elm.style.display!='none') && (elm.gizli!='evet')) //eger zorunlu bi alansa AMA hidden degilse!!!!
                {
                        ////burası zorunlu alan ise valid date olmadıgını kontrol ediyor
//                      if (tipNo==79 || tipNo==80) // datetime or smalldatetime
//                      {
//                              msgForDate += validateDate(theCollection[i], theCollection[i+1], theCollection[i+2])
//                      }
                        ////burası zorunlu alan ise valid date olmadıgını kontrol ediyor
                        if (controlled.indexOf(elm.id)==-1 && elm.disabled==false )
                        {
                                                switch(elm.type)
                                                {
                        // BU ALANLARIN HEPSİ AYNI ŞEKİLDE VALİDATE EDİLİYOR
                                                case "text":
                                                        if (elm.value=="")
                                                        {
                                                                if (msg.indexOf(lbladi(elm.LAB) + ' alanına bilgi girmelisiniz!\n')==-1) msg+=lbladi(elm.LAB) + ' alanına bilgi girmelisiniz!\n';
                                                                controlled+=elm.id + ','
                                                        }
                                                                break
                                                case "textarea":
                                                        if (elm.value=="")
                                                        {
                                                                msg+=lbladi(elm.LAB) + ' alanına bilgi girmelisiniz!\n';
                                                                controlled+=elm.id + ','
                                                        }
                                                                break
                                                case "password":
                                                        if (elm.value=="")
                                                        {
                                                                msg+='Lütfen ' + lbladi(elm.LAB) + ' giriniz!\n';
                                                                controlled+=elm.id + ','
                                                        }
```

```
										break
								case "file":
									if (elm.value=="")
									{
										msg+='Lütfen	'	+
lbladi(elm.LAB) + ' adı ile belirtilen alanda dosya seçiminizi yapınız!\n';
										controlled+=elm.id + ','
									}
										break
						// BU ALANLARIN HEPSİ AYNI ŞEKİLDE VALİDATE EDİLİYOR
								case "checkbox":

	msg+=bakradyo(form,elm)
										controlled+=elm.id + ','
										break
								case "radio":

	msg+=bakradyo(form,elm)
										controlled+=elm.id + ','
										break
								case "select-one":
						//					alert ('i='+i+'----'+elm.name + '--
'+elm.type+'---' + document.all.slk0.options.length)
									if
(elm.options[elm.options.selectedIndex].value==0 || elm.options[elm.options.selectedIndex].value=='')
									{
										msg+='Lütfen bir ' +
lbladi(elm.LAB) + ' seçiniz.\n';

										controlled+=elm.id + ','
									}
										break
							}
											// end switch
						}
											//end	if
(controlled.indexOf(elm.LAB)==-1)

				}
											//
	if (elm.must)
	}
											//	END FOR
LOOP
//	document.all.dv.innerHTML=result
//			alert(result)
			if (msg + msgForDate>'')
						{
//						alert(fokusElm)
						if (!eval(document.getElementById(fokusElm)))
						{
//						alert(fokusElm + ' not found')
							if (eval(document.getElementById(fokusElm+'_xgun')))
								document.getElementById(fokusElm+'_xgun').focus();
						}
						else
						{
							 document.getElementById(fokusElm).focus();
						}
						alert(msg + msgForDate);
```

```
                        return false;
                        }
}

function lbladi(objText) //misal (divTasarim,style)
{
        if (eval('document.all.'+objText))
        {
                return eval('document.all.'+objText).innerText
        }
        else
        {
                return 'Boş Kalan bi yer var ama neresi??' + objText
        }
}


function bakradyo(objForm,obj)
{
        var msg2=''
        var grupadi=obj.id
        var group=objForm[obj.id];
        var checked = false;
                if (!group.length)
                        checked = obj.checked;
                else
              for (var r = 0; r < group.length; r++)
                        {
                                if ((checked = group[r].checked))
                                        break;
                        }
                if (!checked)
                {
                                msg2='Lütfen bir ' + lbladi(obj.LAB) + ' seçiniz.\n'
                                if (msg.indexOf(msg2)!=-1) msg2=''
                }
//      alert(msg2)
        return msg2
}



function TabNext()
{
obj=event.srcElement
objId=obj.id
objIdGenel=objId.substr(0,objId.length-3)

        if (obj.value.length == obj.maxLength)
        {
                if (objId.lastIndexOf('_xgun')>-1)
                {
                                if (eval('document.forms[0].' + objId.replace('_xgun', '_xayi')))
                                {
                                        eval('document.forms[0].'       +       objId.replace('_xgun',
'_xayi')).focus();
                                        eval('document.forms[0].'       +       objId.replace('_xgun',
'_xayi')).select();
```

```
                                        }
                                }
                                if (objId.lastIndexOf('_xayi')>-1)
                                {
                                        if (eval('document.forms[0].' + objId.replace('_xayi', '_xyil')))
                                        {
                                                eval('document.forms[0].'       +       objId.replace('_xayi',
'_xyil')).focus();
                                                eval('document.forms[0].'       +       objId.replace('_xayi',
'_xyil')).select();
                                        }
                                }
                        }
}
////////////////////////////////////////////////////////////val.asp/////////////////////////////////////////////

function checkInt()
{
        var whichCode = event.keyCode;
//      alert(whichCode)
//      ali.innerHTML=whichCode
        strCheck=",8,9,13,35,36,37,38,39,40,46,48,49,50,51,52,53,54,55,56,57,96,97,98,99,100,101,102
,103,104,105,109,189,"
        //46 yı cıkardım
        // 46 delete tuşudur !!!
        // 109 ve 189 eksi tuşudur !!yeni ekledim
        var whichCode2 = ',' +whichCode+ ',';
        if (strCheck.indexOf(whichCode2) == -1) return false;  // Not a valid key
//      alert (whichCode2)
}



function checkDecimals()
{
if(event.keyCode==9) return false;
var fld=event.srcElement
var e=event
milSep='.'
decSep=','
var sep = 0;
var decLength = getcharAt(fld,4);
if (decLength==0) decSep=''
var key = '';
var tut = '';
var tut2 = '';
var i = j = 0;
var len = len2 = 0;
var whichCode = (window.Event) ? event.which : event.keyCode;

        var whichCode2 = ',' +whichCode+ ',';
        strCheck2=",8,9,13,35,36,37,38,39,40,46,48,49,50,51,52,53,54,55,56,57,96,97,98,99,100,101,10
2,103,104,105,"
        if (strCheck2.indexOf(whichCode2) == -1) return false;  // Not a valid key

strCheck=",8,9,13,35,36,37,38,39,40,48,49,50,51,52,53,54,55,56,57,96,97,98,99,100,101,102,103,104,10
5,"
var aux = aux2 = '';
if (whichCode == 13 || whichCode == 37 || whichCode == 39 || whichCode == 36 ) return true;  // Enter
```

```
//                    enter                                    sol ok                    sag ok
                      home tusu
key = String.fromCharCode(whichCode);  // Get key value from key code


len = fld.value.length;
for (ii=0;ii<decLength;ii++) { tut+='0' }

for(i = 0; i < len; i++)
{
        if (fld.value.charAt(0) == '0' && len==1) // ilk ve tek  char = 0
        {
                fld.value='0'
                fld.value=fld.value + tut
                return;
        }
        if ((fld.value.charAt(i) != '0') && (fld.value.charAt(i) != decSep)) break;
}
aux = '';
for(; i < len; i++)
if (strCheck.indexOf(','+fld.value.charCodeAt(i)+',')!=-1)
{
        aux += fld.value.charAt(i);                                                    //burda     virgul
ve noktalardan arta kalanları alıyor
}
len = aux.length;  //virgul ve noktalardan arta kalanlar

if (len <= decLength)
{
        for (ii=0;ii<decLength-len;ii++) { tut2 +='0' }
        fld.value = '0'+ decSep + tut2 + aux;
}
if (len > decLength)
        {
        aux2 = '';
        for (j = 0, i = len - decLength - 1; i >= 0; i--)
        {
                if (j == 3)
                {
                        aux2 += milSep;
                        j = 0;
                }
        aux2 += aux.charAt(i);
        j++;
        }// for bitti

        fld.value = '';
        len2 = aux2.length;

        for (i = len2 - 1; i >= 0; i--) fld.value += aux2.charAt(i);

        fld.value += decSep + aux.substr(len - decLength, len);
        } //if (len > 2)
return false;
}
////////////////////////////////////////////////////////////////val.asp/////////////////////////////////////////////
function toDate(dateString,dateType)
{
```

```
return dateString.substring(6,10) + dateString.substring(3,5) + dateString.substring(0,2) +
                    dateString.substring(11,13) + dateString.substring(14,16) + dateString.substring(17,19)

}

function kontrolcu_iptal()
   {
                        if (confirm("Kaydı silmek istediğinize emin misiniz ? ") )
                                {
//                      deger4.onmouseup=new Function {"return false;";}
//                                              dd=new Function("alert('Function created at " +(new
Date())+"')")
                                              dd=new Function("return (true)")
                                              document.forms[0].onsubmit= dd
                                              document.forms[0].submit();
                                }
                          else
                                {
                                              return  false;
                                }

          }

var data_tipleri=new Array()
var base=10
data_tipleri[65]=new Array("char"                ,"A"       ,"text"                  ,"text"                  )
data_tipleri[66]=new Array("nchar"               ,"B"       ,"text"                  ,"text"                  )
data_tipleri[67]=new Array("nvarchar"            ,"C"       ,"text"                  ,"text"                  )
data_tipleri[68]=new Array("varchar"             ,"D"       ,"text"                  ,"text"                  )
data_tipleri[69]=new Array("text"                ,"E"       ,"text"                  ,"text"                  )
data_tipleri[70]=new Array("ntext"               ,"F"       ,"text"                  ,"text"                  )
data_tipleri[71]=new Array("int"                 ,"G"       ,-2147483648             ,2147483648
)
data_tipleri[72]=new Array("smallint"            ,"H"       ,-32768                  ,32768
)
data_tipleri[73]=new Array("bigint"              ,"I"       ,-9223372036854775808
                    ,9223372036854775808          )
data_tipleri[74]=new Array("tinyint"             ,"J"       ,0
          ,255                    )
data_tipleri[75]=new Array("numeric"             ,"K"       ,(-1)*(Math.pow(10,38))+1
                                ,Math.pow(10,38)-1          )
data_tipleri[76]=new Array("decimal"             ,"L"       ,(-1)*(Math.pow(10,38))+1
                                ,Math.pow(10,38)-1          )
data_tipleri[77]=new Array("money"               ,"M"       ,new Number(-922337203685477.5808)
,new Number(922337203685477.5808)       )
data_tipleri[78]=new Array("smallmoney"          ,"N"       ,new Number(-214748,3648)
,new Number(214748,3648)                )
data_tipleri[79]=new Array("datetime"            ,"O"       ,new Date("January 1, 1753")             ,new
Date("December 31, 9999")          )
data_tipleri[80]=new Array("smalldatetime"       ,"P"       ,new Date("January 1, 1900")             ,new
Date("June 6, 2079")          )
data_tipleri[81]=new Array("timestamp"           ,"Q"       ,"text"                  ,"text"
)
data_tipleri[82]=new Array("bit"                 ,"R"       ,"text"                  ,"text"                  )
data_tipleri[83]=new Array("binary"              ,"S"       ,"text"                  ,"text"                  )
data_tipleri[84]=new Array("varbinary"           ,"T"       ,"text"                  ,"text"                  )
data_tipleri[85]=new Array("float"               ,"U"       ,"text"                  ,"text"
                    )//şimdilik
```

```
data_tipleri[86]=new Array("image"                ,"V"      ,"text"                ,"text"                )
data_tipleri[87]=new Array("real"                 ,"W"      ,"text"                ,"text"
                                        )//şimdilik
data_tipleri[88]=new Array("sql_variant"          ,"X"      ,"text"                ,"text"                )
data_tipleri[89]=new  Array("uniqueidentifier"            ,"Y"          ,"text"                        ,"text"
)
//data_tipleri[90]=new  Array("bilinmeyen"                ,"Z"          ,"text"                        ,"text"
)
data_tipleri[90]=new Array("YIL"                  ,"Z"      ,1900
                  ,2020                   )


InitializeValidation();
function cancelBack()
{
alert(event.keyCode)
}
function f1()
{
alert('keypress')
}
function f2()
{
alert('keydown')
}

function addEvent(elm, evnt,  fn, textOrFunction)
{
        mEvnt = (eval('elm.' + evnt)) ? eval('elm.' + evnt).toString() : ''
        mBlurFinal = stripFunction(mEvnt)
        if (textOrFunction == 'function') fn=stripFunction(fn)
        return new Function (mBlurFinal +  '\n' + fn)
}

function stripFunction(txt)
{
        vIlk = txt.indexOf('{')+2
        v2 = txt.lastIndexOf('}')
        v3 = txt.indexOf('{')-1
        return  txt.substr(vIlk, txt.length-vIlk-1)
}

<%if request.form="ozcan_bakacak_edgişecek" then       %>
        if (top!=self && document.all.girTABLE)
        {
                orjWidth=document.all.girTABLE.offsetWidth+16
                self.resizeTo(orjWidth, document.all.girTABLE.offsetHeight)
        }
<%end if%>
</script>
```

sgirişişlem.asp

```
<script>
var girBorTABLEWidth=0
var parentBODYWidth=0
var parent_girBorTABLEWidthFARK
if (self.parent && document.all.girBorTABLE)
{
```

```
	girBorTABLEWidth=document.all.girBorTABLE.offsetWidth
	girBorTABLEHeight=document.all.girBorTABLE.offsetHeight

	parentBODYWidth  = self.parent.document.body.offsetWidth
	parentBODYHeight  = self.parent.document.body.offsetHeight


	parent_girBorTABLEWidthFARK = parentBODYWidth - girBorTABLEWidth
	parent_girBorTABLEHeightFARK = parentBODYHeight - girBorTABLEHeight

	if (parent_girBorTABLEWidthFARK < 300)
	{
		document.all.girBorTABLE.width=parentBODYWidth - 32
		girBorTABLEWidth = document.all.girBorTABLE.offsetWidth
	}
//	if (parent_girBorTABLEHeightFARK < 200)
//	{
//		document.all.girBorTABLE.width=parentBODYWidth - 32
//		girBorTABLEWidth = document.all.girBorTABLE.offsetWidth
//	}

//	alert('parentBODYWidth=' + parentBODYWidth + '\n, girBorTABLEWidth=' + girBorTABLEWidth
//		+ '\n, parentBODYWidth - girBorTABLEWidth=' + parent_girBorTABLEWidthFARK )

//	alert('parentBODYHeight=' + parentBODYHeight + '\n, girBorTABLEHeight=' + girBorTABLEHeight
//		+ '\n, parentBODYHeight - girBorTABLEHeight=' + parent_girBorTABLEHeightFARK )

	if (parent.girTD)
	{	parent.girTD.height=girBorTABLEHeight
		parent.girTD.width=Math.min(parentBODYWidth, girBorTABLEWidth)
	}
	self.resizeTo(Math.min(parentBODYWidth, girBorTABLEWidth), girBorTABLEHeight)
//	alert(self.parent.document.parentNode.tagName) burda bulmaya caliştim ben neyin içindeyim ama olmadı...
}

</SCRIPT>

<SCRIPT>
//if (self.parent.sagTABLE) alert(self.parent.girTD.offsetHeight)
if (top!=self && document.all.girBorTABLEx) // BEN IFRAMIN ICINDE İSEM....BENIM IFRAME'MİMİ KENDİ TABLOM KADAR GENİŞLET !!!
{	var ben=self
	var ustum=window.parent
	var icindeBulundugumEleman=null
	var ustumunicindeBulunduguEleman=null
	var ustumunWidthi=0
	var myList=null
	if (self.parent.lisFRAME) myList=self.parent.lisFRAME.document.all.lisBorTD

	var tellMe=0
	if (tellMe==1) //burda, yukarıdakı degişkenler değişiyor dikkat et!
	{	do
		{if (ustum.document.getElementById(ben.name).parentElement.tagName!="TD")
```

```
	icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement.parentE
lement.id
		else

	icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement.id

		alert('benim iframe\'mimin adı= ' + ben.name + '\n' + 'ustumdeki iframe ise adı= ' +
ustum.name +
		'\n benim, üstümde içinde bulunduğu eleman ise ' + icindeBulundugumEleman)

		ben=ustum
		ustum=ben.parent
		}
		while (ben!=ustum)
	}	//if (tellme=1) //burda, yukarıdakı degişkenler değişiyor dikkat et!


	if (document.all.girBorTABLE)
	{		myList = null
			myListWidth = 0
			myListHeight = 0

			//height ile ilgili
			if (myList)
			{
//				alert(self.parent.lisFRAME.document.all.lisBorTD)
				myListWidth = myList.offsetWidth
				myListHeight = myList.offsetHeight
			}
			//height ile ilgili


	//iframe'mımın içinde bulundugu elemanın widthi nedir?
	if (ustum.document.getElementById(self.name).parentElement.tagName!="TD")

	icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement.parentE
lement //bazen fieldSet'in iicnde oluyorum da !!!
					else

	icindeBulundugumEleman=ustum.document.getElementById(ben.name).parentElement

			tutmyWidth=girBorTD.offsetWidth
			myWidthartiYUZ=tutmyWidth+100

//			alert(icindeBulundugumEleman.offsetWidth + '--' + myWidthartiYUZ)

			if (ustum!=top)  //ustum en tepe diilse onun ustunun widthi nedir?? CUNKU
lis VARSA ve GENİŞ ise benim iframe'mımın icinde bulundugu TD'yi de genişletiyor!!!
			{
				if
(ustum.parent.document.getElementById(ustum.name).parentElement.tagName!="TD")
				{

	ustumunicindeBulunduguEleman=ustum.parent.document.getElementById(ustum.name).parent
Element.parentElement //bazen fieldSet'in iicnde oluyorum da !!!

	ustumunWidthi=ustumunicindeBulunduguEleman.offsetWidth
				}
```

```
                                        else
                                        {

        ustumunicindeBulunduguEleman=ustum.parent.document.getElementById(ustum.name).parent
Element

        ustumunWidthi=ustumunicindeBulunduguEleman.offsetWidth
                                        }
                                        myWidth= Math.max(tutmyWidth, ustumunWidthi-20) // bu 20
aslında scroll bar..bunu da kontrol etmek lasım
                                }
                                else
                                {
                                myWidth=                                        Math.min(myWidthartiYUZ,
icindeBulundugumEleman.offsetWidth)
                                }


                                if (self.parent.name!='ikmain') girBorTABLE.style.width='100%'; else
girBorTABLE.align='center'
//                              myWidth= girBorTABLE.offsetWidth + self.document.body.scrollTop
//burda amac scrollbar cıkmıssa onları haric tutmak..şimdilik askıda !!!
                                myHeight= girBorTABLE.offsetHeight + self.document.body.scrollLeft -
myListHeight   //BILMIYORUM NERDEN GELIYOR !!!!
//                              alert('ben girTD! widthim=' + myWidth)
                                //adamın içinde bulunduğu daha geniş ise onun kadar genişlesin!!!
//                              myHeight =          Math.max(myHeight,
self.parent.document.all.girTD.offsetHeight)
//                              self.setTimeout('self.resizeTo(' + myWidth + ','+ girBorTABLE.offsetHeight
+')',250)
                                self.resizeTo(myWidth, girBorTABLE.offsetHeight)
        }
}
else
{
//alert(girBorTABLE.clientWidth)
//if (document.all.girBorTABLE) girBorTABLE.style.width=girBorTABLE.offsetWidth
}

        //burasıını 22 martta ekledim...gir kendi frameinin widthinin ve heightini içindeki tabloya göre
ayarlasın diye
</SCRIPT>
```

slishtc.asp

```
<style>
        td#dummyTD
        {
                height:expression(document.all.lisFormDIV.offsetHeight); /*IE5+ only*/;;
                background-color : silver;
        }
        div#lisFormDIV
        {
                position:absolute;
                left:   expression(((document.all.data.clientWidth      -       this.offsetWidth)/2)    +
document.getElementById("data").scrollLeft); /*IE5+ only*/;
                top:    expression(document.all.data.clientHeight       -       this.offsetHeight       +
document.getElementById("data").scrollTop); /*IE5+ only*/
                width:100%;
```

```
		/*		width:expression(Math.min(document.body.offsetWidth,
document.all.DataTable.offsetWidth)+30);*/
		background-color : #D6D3CE;
		border:1px outset silver;
		z-index: 1322222;
	}
	div.tableContainer
	{
		width:100%;
		height: 100%;
		overflow: auto;
	}
	#DataTable
	{
		empty-cells:show;
		padding:2;
		background-color : black;
		font-size:7pt;
		font-family:tahoma;
	<%if slisdrdpYOK=1 then%>
		behavior:url('includes/slissort.htc');
	<%else%>
		behavior:url('includes/slissort.htc') url('includes/slisdrdp.htc') ;
	<%end if%>
		dragColor: #76C1DB;
		dragObjColor:#089CD6;
	}
	th.baslik, td#baslik
	{
		position:relative;
		top: expression(document.getElementById("data").scrollTop); /*IE5+ only*/
		z-index: 13;
	}
	TH
	{
		border-collapse:collapse;
		border-width:1px;
		border-style:outset;
		border-color:silver;
		font-size:7pt;
		font-family:tahoma;
		font-weight:normal;
		color:white;
		position:relative;
		top: expression(document.getElementById("data").scrollTop); /*IE5+ only*/
		z-index: 11;
		background-color :#006DBA;
		padding:2;
	}
	td.locked
	{
		left: expression(document.getElementById("data").scrollLeft);
		/*IE5+ only*/
		position: relative;
		border-right-color : Black;
		border-right-style : solid;
		border-right-width : 0.1mm;
	}
	#sabit
```

```
		{
			top: expression(document.getElementById("data").scrollTop); /*IE5+ only*/
			left: expression(document.getElementById("data").scrollLeft); /*IE5+ only*/
			position: relative;
			z-index: 12;

		}
		TBODY
		{
			overflow: auto;
			slColor:#ffcccc;
			hlColor:#BEC5DE;
		<%if shighlightYOK = 0 then%>
			behavior:url('includes/slishighlight.htc');
		<%end if%>
			cursor:hand;
		}
		#DataTable td
		{
		}
		#sabiteklenti
		{
			top: expression(document.getElementById("data").scrollTop-2); /*IE5+ only*/
			left: expression(document.getElementById("data").scrollLeft); /*IE5+ only*/
			position: relative;
			z-index: 23423443;
			width:100%;
			background-color : threedface;
		}
		#baslikSPAN
		{
			font-family: Arial, Helvetica, sans-serif;
			font-size: 11px;
			font-weight: bolder;
		}
</style>
```

slisrecordset.asp

```
<%

'---- LockTypeEnum Values ----
Const adLockReadOnly = 1
Const adLockPessimistic = 2
Const adLockOptimistic = 3
Const adLockBatchOptimistic = 4

'---- CursorLocationEnum Values ----
Const adUseServer = 2
Const adUseClient = 3

'---- CursorTypeEnum Values ----
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3

  if request.form("xlissayfa")>"" then sayfaci=cint(request.form("xlissayfa"))
  if request.form("xlisperci")>"" then perci=cint(request.form("xlisperci"))
```

```
set rs=server.createobject("adodb.recordset")

        rs.ActiveConnection = db
        rs.LockType = adLockReadOnly
        rs.CursorType = 3

'alert (sql)
'response.write(sql)
'response.flush

        rs.open sql,db

if not rs.eof then
        if perci = 1 then
                rs.PageSize = rs.RecordCount
        else
                rs.PageSize = perci
        end if
        rs.CacheSize = rs.PageSize
        rs.AbsolutePage=sayfaci
        intPageCount = rs.PageCount
        intRecordCount = rs.RecordCount
end if

%>
```

slisform.asp

```
<%
'lisform=1 -->arama yaptık ama bulamadık demek
'lisform="" -->EOF'a düşmedik demek...
if lisform=1 or lisform="" then
%>
<div id="lisFormDIV">
        <TABLE id=mukTABLE cellpadding="0" cellspacing="0" align=center width=100%>
          <tr id=mukTR>
            <td id=mukTD>
              <form  action="<%=gir_action %>" target="girFRAME" method="post" id="lisFORM"
name="lisFORM" >
                <INPUT type="hidden" value=""  name="<%=form2key1 %>" id="key1">
                <!-- vb if ile aşagıdaki isimlere gore alanları ekle yada cıkartttttttttt -->
                <%if form2key2>"" then  %>
                  <INPUT  type="hidden"  value="<%=session(form2key2)  %>"
name="<%=form2key2 %>" id="key2">
                <%end if 'form2value2>"" then  %>

                <%if form2key3>"" then  %>
                  <INPUT  type="hidden"  value="<%=session(form2key3)  %>"
name="<%=form2key3 %>" id="key3">
                <%end if 'form2value2>"" then  %>
              </form>
            </td>
          </tr>
        </TABLE>
        <TABLE cellpadding="0" cellspacing="0" align=center  >
          <tr>
            <td>
                        <!--#include file="smukayese.asp"-->
```

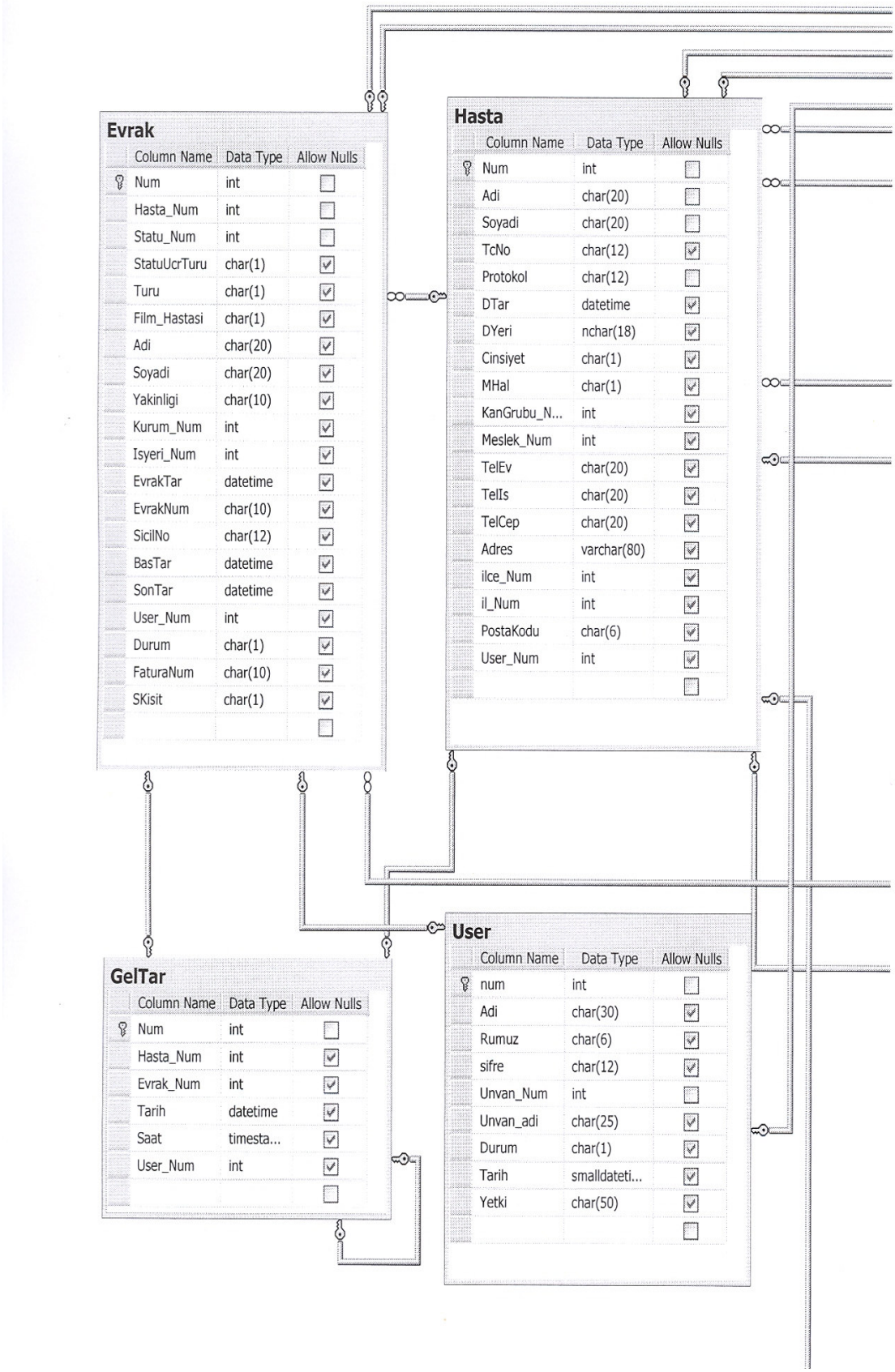
```
                        <!--#include file="ssayfala.asp"-->
                      </td>
                    </tr>
        </TABLE>
</div>
<% End If %>
```

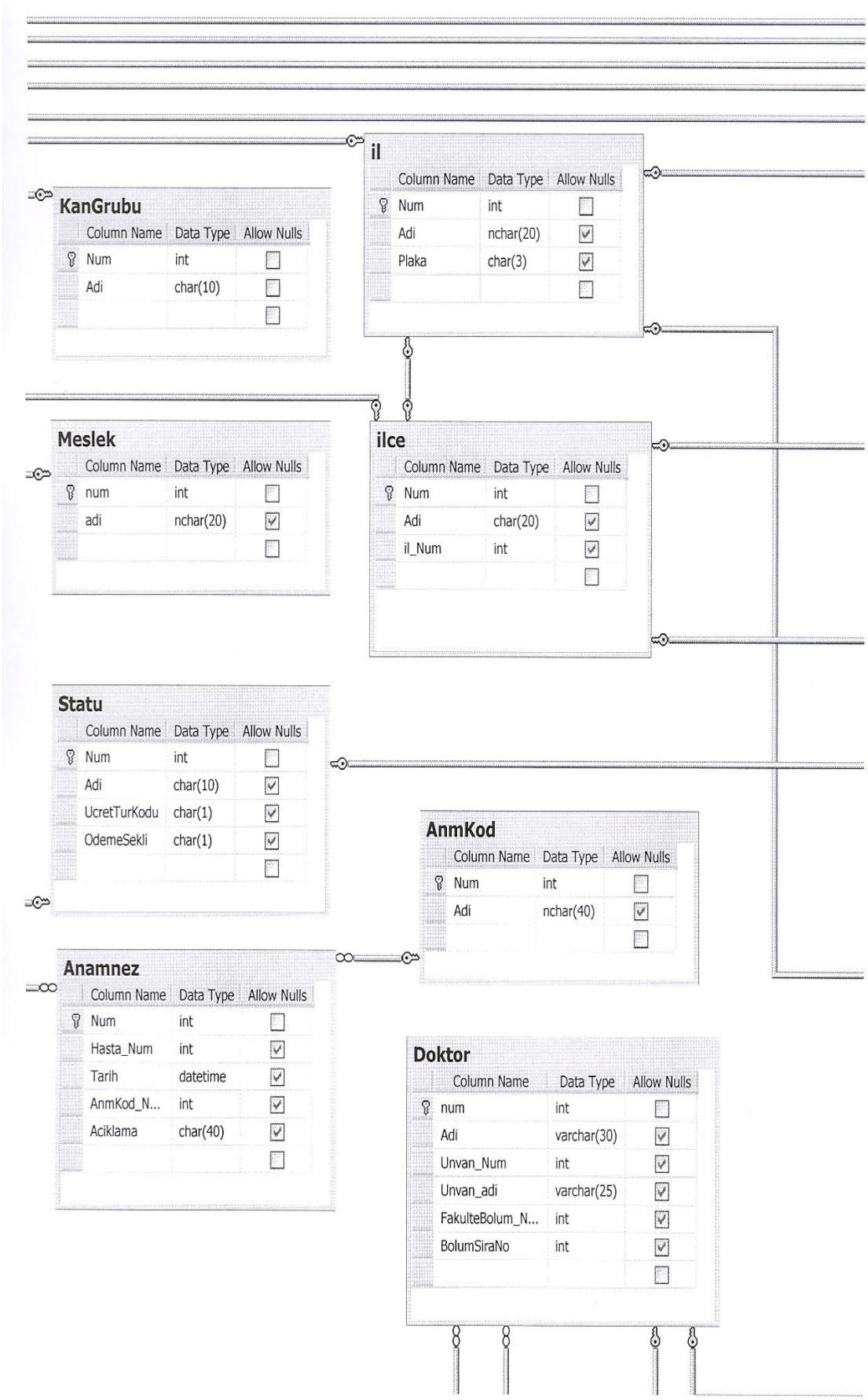
slisEOF.asp

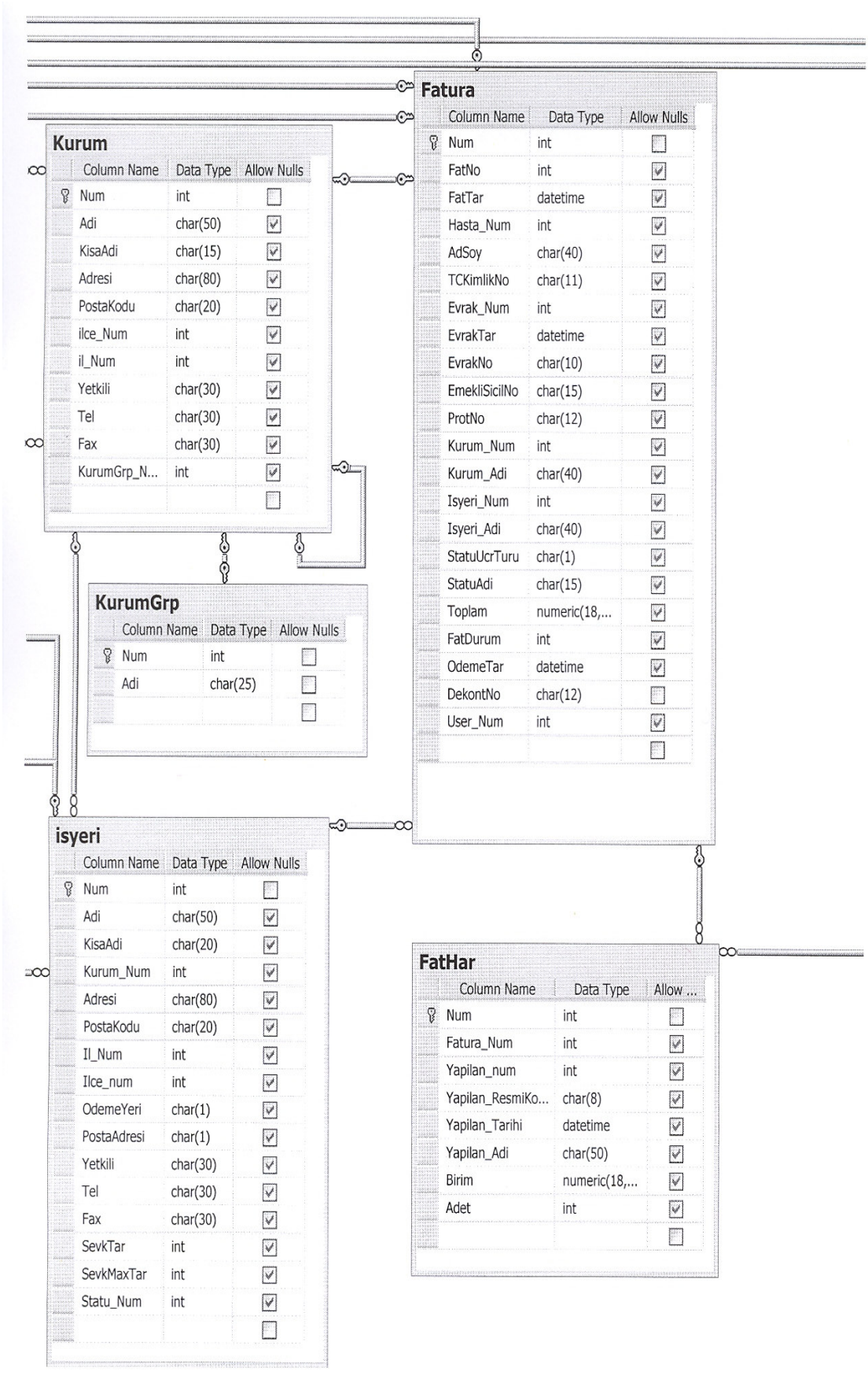
```
 <%
          msg=""
          lisform=0
           if xlissorgu>"" then
                  msg=labelci("lis_query_eof","Aradığınız kritere göre kayıt bulunamadı")
                  lisform=1
           else
                   msg= listenin_basligi & labelci("lis_eof"," içinde görüntülenecek kayıt bulunamamıştır.")
                   lisform=0
           end if
      %>

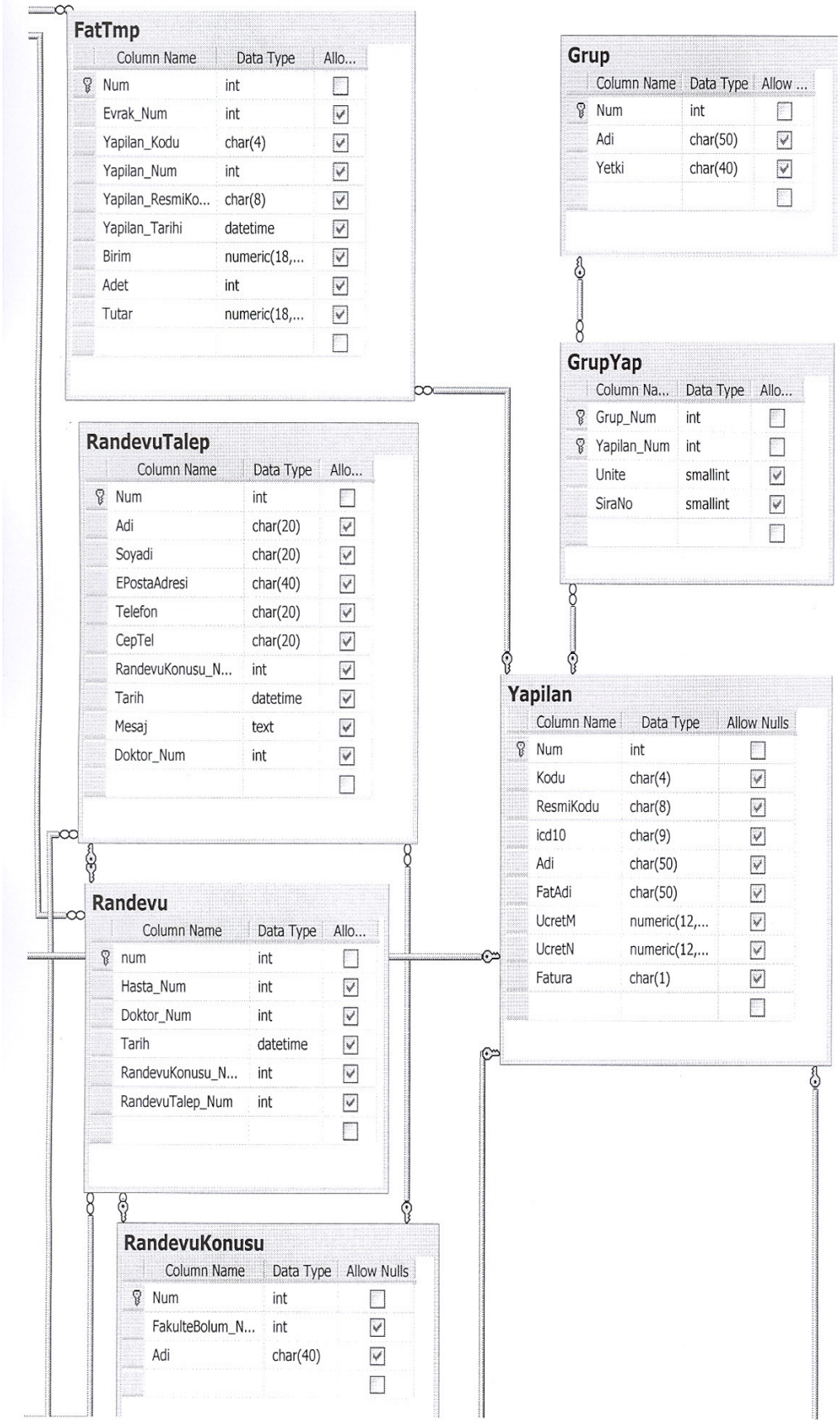
<TR align=left >
<td colspan="100" align=center class=listetd style="color:white;">
<%=msg%></td></tr>
```

EK-E. Tablo Yapıları Ve İlişki Diyagramı:




il
Column Name Data Type Allow Nulls
Num int
Adi nchar(20)
Plaka char(3)
KanGrubu
Column Name Data Type Allow Nulls
Num int
Adi char(10)
Meslek
Column Name Data Type Allow Nulls
num int
adi nchar(20)
ilce
Column Name Data Type Allow Nulls
Num int
Adi char(20)
il_Num int
Statu
Column Name Data Type Allow Nulls
Num int
Adi char(10)
UcretTurKodu char(1)
OdemeSekli char(1)
AnmKod
Column Name Data Type Allow Nulls
Num int
Adi nchar(40)
Anamnez
Column Name Data Type Allow Nulls
Num int
Hasta_Num int
Tarih datetime
AnmKod_N... int
Aciklama char(40)
Doktor
Column Name Data Type Allow Nulls
num int
Adi varchar(30)
Unvan_Num int
Unvan_adi varchar(25)
FakulteBolum_N... int
BolumSiraNo int


Kurum
Column Name Data Type Allow Nulls
Num int
Adi char(50)
KisaAdi char(15)
Adresi char(80)
PostaKodu char(20)
ilce_Num int
il_Num int
Yetkili char(30)
Tel char(30)
Fax char(30)
KurumGrp_N... int
Fatura
Column Name Data Type Allow Nulls
Num int
FatNo int
FatTar datetime
Hasta_Num int
AdSoy char(40)
TCKimlikNo char(11)
Evrak_Num int
EvrakTar datetime
EvrakNo char(10)
EmekliSicilNo char(15)
ProtNo char(12)
Kurum_Num int
Kurum_Adi char(40)
Isyeri_Num int
Isyeri_Adi char(40)
StatuUcrTuru char(1)
StatuAdi char(15)
Toplam numeric(18,...
FatDurum int
OdemeTar datetime
DekontNo char(12)
User_Num int
KurumGrp
Column Name Data Type Allow Nulls
Num int
Adi char(25)
isyeri
Column Name Data Type Allow Nulls
Num int
Adi char(50)
KisaAdi char(20)
Kurum_Num int
Adresi char(80)
PostaKodu char(20)
Il_Num int
Ilce_num int
OdemeYeri char(1)
PostaAdresi char(1)
Yetkili char(30)
Tel char(30)
Fax char(30)
SevkTar int
SevkMaxTar int
Statu_Num int
FatHar
Column Name Data Type Allow ...
Num int
Fatura_Num int
Yapilan_num int
Yapilan_ResmiKo... char(8)
Yapilan_Tarihi datetime
Yapilan_Adi char(50)
Birim numeric(18,...
Adet int

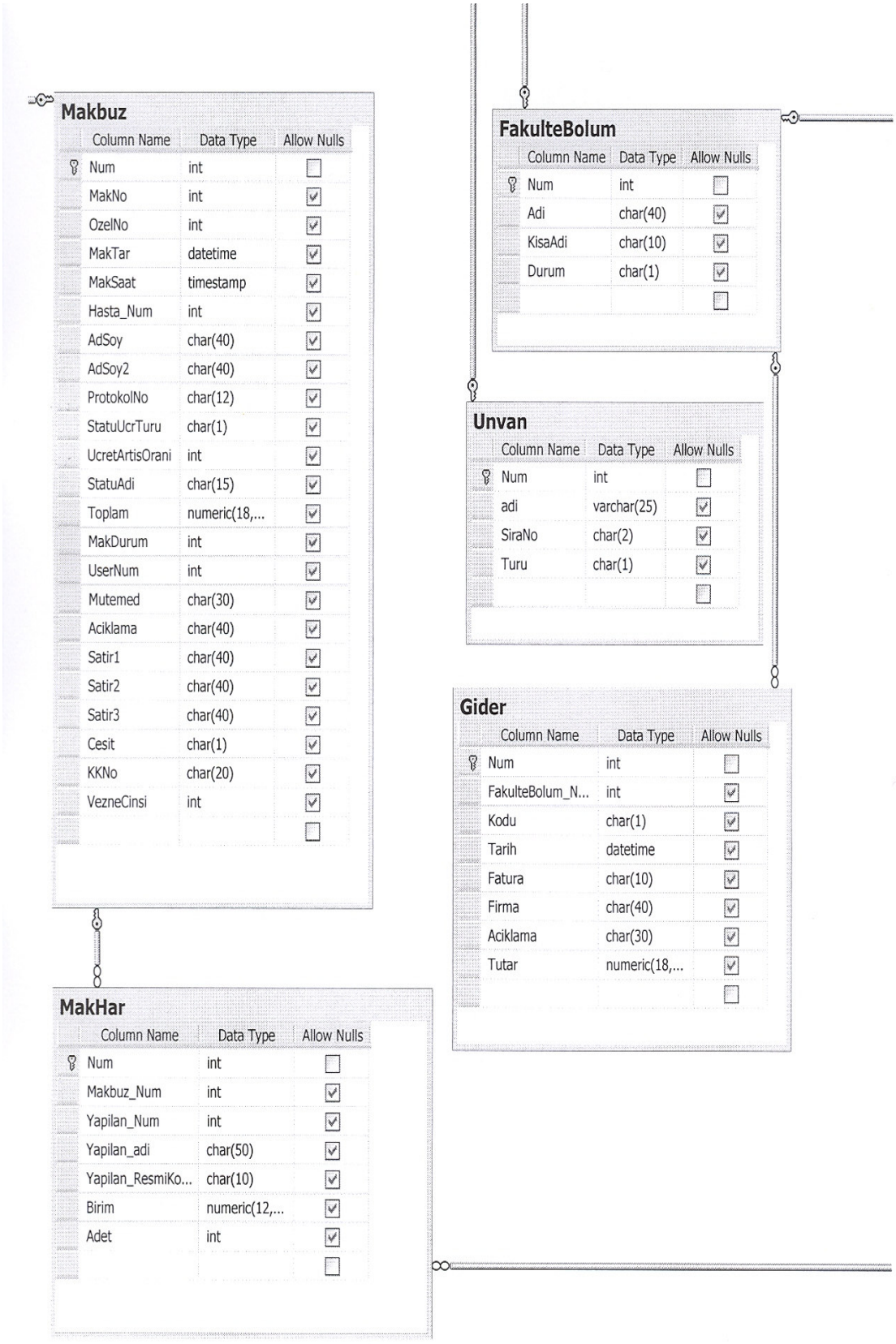

FatTmp
Column Name Data Type Allo...
Num int
Evrak_Num int
Yapilan_Kodu char(4)
Yapilan_Num int
Yapilan_ResmiKo... char(8)
Yapilan_Tarihi datetime
Birim numeric(18,...
Adet int
Tutar numeric(18,...
Grup
Column Name Data Type Allow ...
Num int
Adi char(50)
Yetki char(40)
GrupYap
Column Na... Data Type Allo...
Grup_Num int
Yapilan_Num int
Unite smallint
SiraNo smallint
RandevuTalep
Column Name Data Type Allo...
Num int
Adi char(20)
Soyadi char(20)
EPostaAdresi char(40)
Telefon char(20)
CepTel char(20)
RandevuKonusu_N... int
Tarih datetime
Mesaj text
Doktor_Num int
Yapilan
Column Name Data Type Allow Nulls
Num int
Kodu char(4)
ResmiKodu char(8)
icd10 char(9)
Adi char(50)
FatAdi char(50)
UcretM numeric(12,...
UcretN numeric(12,...
Fatura char(1)
Randevu
Column Name Data Type Allo...
num int
Hasta_Num int
Doktor_Num int
Tarih datetime
RandevuKonusu_N... int
RandevuTalep_Num int
RandevuKonusu
Column Name Data Type Allow Nulls
Num int
FakulteBolum_N... int
Adi char(40)

Firma

Column Name	Data Type	Allow Nulls
Num	int	☐
Adi	char(30)	☑
Yazar1	char(30)	☑
Yazar2	char(10)	☑
PKodu	char(10)	☑
Sehir	char(20)	☑
Ulke	char(20)	☑
Tel1	char(30)	☑
Tel2	char(30)	☑
Fax	char(30)	☑
Telex	char(30)	☑
Sita	char(30)	☑
Web	char(30)	☑
eposta	char(30)	☑
imzaadi	char(20)	☑
imzaetiketi	char(20)	☑
Mutemed	char(20)	☑
Vezne	char(20)	☑
Sayman	char(20)	☑
SaymanUnvan	char(20)	☑
TimeOut	int	☑
Acilis	numeric(18,...	☑
Mesaj	char(20)	☑
Uyari	int	☑
Tahsilatci	int	☑
KKVade	int	☑
KKUyari	numeric(18,...	☑
YedekLocal	numeric(18,...	☑
YedekServer	numeric(18,...	☑
YedekCd	numeric(18,...	☑
YedekHata	numeric(18,...	☑
Logo	image	☑

DisPlan

Column Name	Data Type	Allow Nulls
TedPlan_Num	int	☐
d11	char(15)	☑
d12	char(15)	☑
d13	char(15)	☑
d14	char(15)	☑
d15	char(15)	☑
d16	char(15)	☑
d17	char(15)	☑
d18	char(15)	☑
d21	char(15)	☑
d22	char(15)	☑
d23	char(15)	☑
d24	char(15)	☑
d25	char(15)	☑
d26	char(15)	☑
d27	char(15)	☑
d28	char(15)	☑
d31	char(15)	☑
d32	char(15)	☑
d33	char(15)	☑
d34	char(15)	☑
d35	char(15)	☑
d36	char(15)	☑
d37	char(15)	☑
d38	char(15)	☑
d41	char(15)	☑
d42	char(15)	☑
d43	char(15)	☑
d44	char(15)	☑
d45	char(15)	☑
d46	char(15)	☑
d47	char(15)	☑
d48	char(15)	☑
d51	char(15)	☑
d52	char(15)	☑
d53	char(15)	☑
d54	char(15)	☑
d55	char(15)	☑
d61	char(15)	☑
d62	char(15)	☑
d63	char(15)	☑
d64	char(15)	☑


Makbuz
Column Name Data Type Allow Nulls
Num int
MakNo int
OzelNo int
MakTar datetime
MakSaat timestamp
Hasta_Num int
AdSoy char(40)
AdSoy2 char(40)
ProtokolNo char(12)
StatuUcrTuru char(1)
UcretArtisOrani int
StatuAdi char(15)
Toplam numeric(18,...
MakDurum int
UserNum int
Mutemed char(30)
Aciklama char(40)
Satir1 char(40)
Satir2 char(40)
Satir3 char(40)
Cesit char(1)
KKNo char(20)
VezneCinsi int
FakulteBolum
Column Name Data Type Allow Nulls
Num int
Adi char(40)
KisaAdi char(10)
Durum char(1)
Unvan
Column Name Data Type Allow Nulls
Num int
adi varchar(25)
SiraNo char(2)
Turu char(1)
Gider
Column Name Data Type Allow Nulls
Num int
FakulteBolum_N... int
Kodu char(1)
Tarih datetime
Fatura char(10)
Firma char(40)
Aciklama char(30)
Tutar numeric(18,...
MakHar
Column Name Data Type Allow Nulls
Num int
Makbuz_Num int
Yapilan_Num int
Yapilan_adi char(50)
Yapilan_ResmiKo... char(10)
Birim numeric(12,...
Adet int

SKisit

Column Name	Data Type	Allow Nulls
Evrak_Num	int	☑
Kisit	char(40)	☑
		☐

TedaviBolumu

Column Name	Data Type	Allow Nulls
🔑 Num	int	☐
Adi	char(25)	☑
KatBilgisi	char(10)	☑
Durum	char(1)	☑
		☐

FatInd

Column Na...	Data Type	Allow Nulls
🔑 FatNo	int	☐
		☐

ResmiBolum

Column Name	Data Type	Allow Nulls
🔑 Num	int	☐
Adi	char(40)	☑
		☐

Kimden

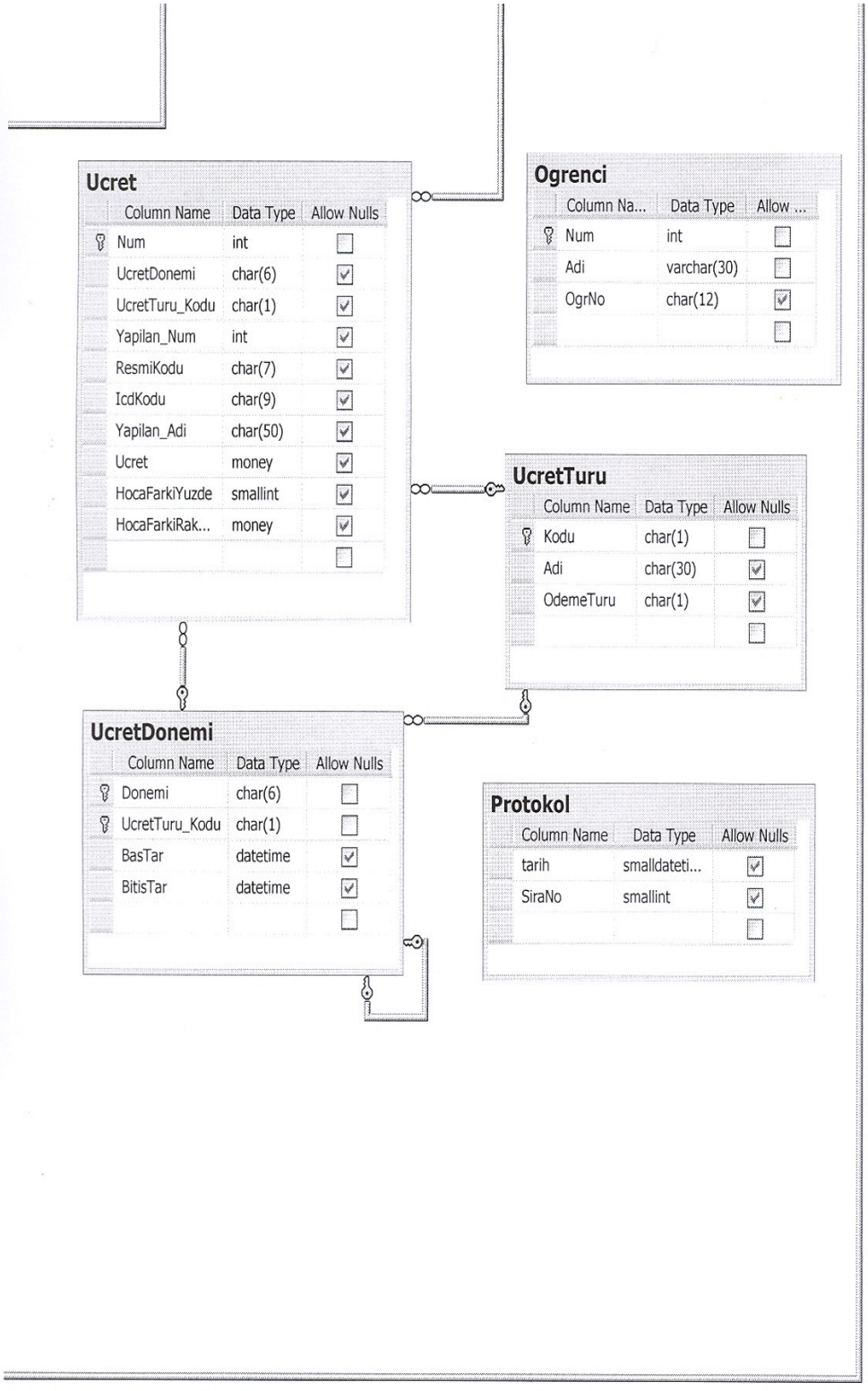
Column Name	Data Type	Allow Nulls
🔑 id	int	☐
adi	nvarchar(50)	☑
		☐

Kursu

Column Name	Data Type	Allow Nulls
🔑 Num	int	☐
Adi	char(40)	☑
KisaAdi	char(8)	☑
		☐

MKodu

Column Name	Data Type	Allow Nulls
🔑 Kodu	char(10)	☐
Adi	varchar(50)	☐
		☐


Ucret
Column Name Data Type Allow Nulls
Num int
UcretDonemi char(6)
UcretTuru_Kodu char(1)
Yapilan_Num int
ResmiKodu char(7)
IcdKodu char(9)
Yapilan_Adi char(50)
Ucret money
HocaFarkiYuzde smallint
HocaFarkiRak... money
Ogrenci
Column Na... Data Type Allow ...
Num int
Adi varchar(30)
OgrNo char(12)
UcretTuru
Column Name Data Type Allow Nulls
Kodu char(1)
Adi char(30)
OdemeTuru char(1)
UcretDonemi
Column Name Data Type Allow Nulls
Donemi char(6)
UcretTuru_Kodu char(1)
BasTar datetime
BitisTar datetime
Protokol
Column Name Data Type Allow Nulls
tarih smalldateti...
SiraNo smallint

ÖZGEÇMİŞ

22 Ağustos 1971 tarihinde Muğla / Fethiye'de doğdu. İlk öğrenimini Fethiye/Kargı Şehit Refik Cesur İlkokulunda, orta öğrenimini Fethiye Ortaokulunda, lise öğrenimini Fethiye Lisesinde 1987 yılında tamamladı. Aynı yıl Fırat Üniversitesi Bilgisayar Programcılığı Programında yüksek öğrenime başladı, 1988-89 Öğretim yılında İstanbul Üniversitesi TBMYO Bilgisayar Programcılığı Programını tamamladı. 1991 yılında İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Bilimleri Mühendisliği Bölümünde lisans öğrenimine başlayıp 1996 yılında mezun oldu. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İstatistik Programında 1996 yılında Yüksek Lisans eğitimine başladı. İş yoğunluğu sebebiyle tez aşamasında ara verdiği eğitimine halen İstanbul Üniversitesi Fen Bilimleri Enstitüsü Enformatik Programında devam etmektedir.

1989-1990 Exim A.Ş.'de Programcı olarak, 1990-1992 yıllarında Danışman Bilgisayar A.Ş.'de Programcı ve Yazılım Destek Sorumlusu olarak, 1992-1994 yıllarında İstanbul Üniversitesi Teknik Bilimler Meslek Yüksek Okulu'nda Programcı olarak, 1994-1996 Yıllarında Biltim Bilgisayar Hizmetleri A.Ş.'de Uzman Programcı olarak görev yaptı. 1996 senesinde Babil Bilgisayar Firmasını kurmuş halen çalışmalarını bu firmada devam ettirmektedir. Evli ve 2 çocuk babasıdır.