

T.C.
ÇALIŞMA VE SOSYAL GÜVENLİK BAKANLIĞI
İŞ SAĞLIĞI VE GÜVENLİĞİ GENEL MÜDÜRLÜĞÜ

**TEHLİKELİ KİMYASAL MADDELERİN İTHALAT
KONTROLÜ İŞLEMLERİNİN İŞ SAĞLIĞI VE
GÜVENLİĞİ AÇISINDAN DEĞERLENDİRİLMESİ İLE
KONTROL BELGESİ YÖNETİM BİLGİ SİSTEMİ
YAPISININ OLUŞTURULMASI**

Mesut AKANER

(İş Sağlığı ve Güvenliği Uzmanlık Tezi)

ANKARA-2014

T.C.
ÇALIŞMA VE SOSYAL GÜVENLİK BAKANLIĞI
İŞ SAĞLIĞI VE GÜVENLİĞİ GENEL MÜDÜRLÜĞÜ

**TEHLİKELİ KİMYASAL MADDELERİN İTHALAT
KONTROLÜ İŞLEMLERİNİN İŞ SAĞLIĞI VE
GÜVENLİĞİ AÇISINDAN DEĞERLENDİRİLMESİ İLE
KONTROL BELGESİ YÖNETİM BİLGİ SİSTEMİ
YAPISININ OLUŞTURULMASI**

Mesut AKANER

(İş Sağlığı ve Güvenliği Uzmanlık Tezi)

**Tez Danışmanı
Sakine OVACILLI**

ANKARA-2014

T.C.
Çalışma ve Sosyal Güvenlik Bakanlığı
İş Sağlığı ve Güvenliği Genel Müdürlüğü

ONAY

Çalışma ve Sosyal Güvenlik Bakanlığı, İş Sağlığı ve Güvenliği Genel Müdürlüğü İş Sağlığı ve Güvenliği Uzman Yardımcısı Mesut AKANER'in, **Sakine OVACILLI** danışmanlığında tez başlığı "**Tehlikeli Kimyasal Maddelerin İthalat Kontrolü İşlemlerinin İş Sağlığı ve Güvenliği Açısından Değerlendirilmesi İle Kontrol Belgesi Yönetim Bilgi Sistemi Yapısının Oluşturulması**" olarak teslim edilen bu tezin tez savunma sınavı/...../2014 tarihinde yapılarak aşağıdaki jüri üyeleri tarafından "**İş Sağlığı ve Güvenliği Uzmanlık Tezi**" olarak kabul edilmiştir.

JÜRİ BAŞKANI

ÜYE

ÜYE

ÜYE

ÜYE

Yukarıdaki imzaların adı geçen kişilere ait olduğunu onaylarım.

Kasım ÖZER
İSGGM Genel Müdürü

TEŐEKKÜR

Tez hazırlık süreci ve alıőma ve Sosyal Güvenlik Bakanlıđı İő Sađlıđı ve Genel M¼d¼rl¼đ¼'ndeki alıőma hayatım boyunca kıymetli bilgi, deneyim ve desteklerini esirgemeyen baőta Genel M¼d¼r¼m Sayın Kasım ÖZER olmak üzere, İő Sađlıđı ve Güvenliđi Genel M¼d¼r Yardımcısı Sayın İsmail GERİM, İő Sađlıđı ve Güvenliđi Genel M¼d¼r Yardımcısı Sayın Dr. H. N. Rana G¼VEN ve İő Sađlıđı ve Güvenliđi Genel M¼d¼r Yardımcısı Sayın Ahmet ETİN'e teőekk¼rlerimi sunarım. Deđerli bilgi ve deneyimleriyle tez alıőmama önemli ölç¼de katkı sađlayan tez danıőmanım Sayın Sakine OVACILLI'ya yardımlarından dolayı teőekk¼rlerimi sunarım.alıőmamın her aőamasında verdiđi katkı ve destek için deđerli eőim Özge AKANER'e teőekk¼r ve ő¼kranlarımı sunarım.

ÖZET

TEHLİKELİ KİMYASAL MADDELERİN İTHALAT KONTROLÜ İŞLEMLERİNİN İŞ SAĞLIĞI VE GÜVENLİĞİ AÇISINDAN DEĞERLENDİRİLMESİ İLE KONTROL BELGESİ YÖNETİM BİLGİ SİSTEMİ YAPISININ OLUŞTURULMASI

Mesut AKANER

İş Sağlığı ve Güvenliği Genel Müdürlüğü Uzmanlık Tezi, Mayıs 2014

Danışman: Sakine OVACILLI

Ülkemize her yıl binlerce çeşit kimyasal madde ve müstahzar ithal edilmektedir. Bu kimyasal maddeler, kimya sektörünün yanısıra; tarımdan tekstile, otomotivden inşaata kadar pek çok sektörde gerek hammadde gerekse nihai ürün olarak kullanılmaktadır. Ancak ekonomi ve üretime sağladıkları katkının yanı sıra insan ve çevre sağlığını olumsuz yönde etkilemektedirler. İş sağlığı ve güvenliği perspektifinden bakıldığında kimyasallar en önemli risk faktörlerinden biridir. İş kazalarının ve meslek hastalıklarının önlenmesinde kimyasal maddelerin tetkiki, ülkeye kontrol dahilinde sokulması, kullanıldığı alanların belirlenmesi ve bu bilgilerin harmanlanarak zararları etkilerinin ve etkilerin boyutunun tespiti önem taşımaktadır. Bu amaçla İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğünce Tehlikeli Kimyasal Maddelerin Kontrolüne ilişkin olarak gerçekleştirdiği kontrol belgesi işlemleri yürütülmektedir. Her yıl yaklaşık 22.000 başvurunun incelendiği İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğünde işlemlerin etkin şekilde, en kısa sürede, mümkün olan en az işgücü kullanımı ile standart prosedürler çerçevesinde verilmesi elzemdir.

Bu bağlamda hazırlanan tez çalışmasında Kontrol Belgesi işlemlerinin nitelikleri ve sağlamakta olduğu faydalar incelenmiş, bilgi teknolojileri kullanarak kontrol belgesi hizmetinin daha verimli sunulabilmesi amacıyla gerekli altyapı oluşturulmuş ve dokümanite edilmiştir. Bilgi sistemi ile elde edilecek verilerin anlamlı bilgiler halinde kullanılabilmesi ve kurumsal stratejilerin belirlenmesinde yön gösterici olması amacıyla karar destek ve raporlama sistemi oluşturulmuştur.

Anahtar Kelimeler: Tehlikeli Kimyasal Maddeler, Kontrol Belgesi, Yönetim Bilgi Sistemleri, İthalat Kontrolü

ABSTRACT

EVALUATION OF THE PROCEDURES IN THE CONTROL OF HAZARDOUS CHEMICALS IMPORT IN TERMS OF OCCUPATIONAL HEALTH AND SAFETY AND PREPARATION OF MANAGEMENT INFORMATION SYSTEM FRAMEWORK

Mesut AKANER
Directorate General of Occupational Health and Safety
Thesis of OHS Expertise, May 2014

Supervisor: Sakine OVACILLI

Each year, thousands kinds of chemical substances and preparations are imported to our country. These chemicals are used in chemical industry as well as in many other sectors such as agriculture, textiles, automotive, construction not only as raw materials but also as end products. However, beside their contribution to the economy and production they bring out some adverse effects on human and environmental health. Regarding the occupational health and safety perspective, chemicals are one of the most important risk factors. Examination of the chemicals, import surveillance, determination of usage area and evaluation of all information to identify the hazards and risks have a great significance in the prevention of the occupational diseases and accidents.

In this sense, approval of control certificates for hazardous chemicals are carried out by The Institute for Occupational Safety and Health. As nearly 22.000 appliances are evaluated each year, It is imperative to serve the import dealers in a fastest, most standardized and effective way with minimum workforce requirement. Improving the service quality , and providing traceable and standardized procedures are only possible by supplying with proper automation system and software.

In this regard, the thesis mainly focused on the examination of the benefits of Control Document certification, it also provided and documented the framework of the software and database required for more effective servicing. A reporting module

with graphical interface, database management and programming has been completed as a pre-decision support system to provide assistance in the determination of the strategies.

Keywords: HazardousChemicals, Control Certificate, Management Information Systems, Import Control

İÇİNDEKİLER

KABUL VE ONAY SAYFASI	
TEŞEKKÜR	
TÜRKÇE ÖZET	
İNGİLİZCE ÖZET	
İÇİNDEKİLER	
ŞEKİLLER LİSTESİ	
TABLolar LİSTESİ	
SİMGE VE KISALTMALAR	
GİRİŞ VE AMAÇ	1
GENEL BİLGİLER.....	4
KİMYASAL MADDELERİN İTHALATI VE KONTROL BELGELERİ.....	4
GEREÇ VE YÖNTEMLER	19
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİNİN OLUŞTURULMASI	19
SİSTEMİN TASARIMINDA KULLANILAN YAKLAŞIMLAR.....	23
BULGULAR	43
KONTROL BELGESİ İŞLEMLERİ SİSTEM ANALİZİ	43
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ TASARIMI.....	51
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ DEĞERLENDİRME SİSTEMİ	66
ÜLKEMİZDE VE DÜNYADA BENZER ÇALIŞMALAR	68
TARTIŞMA	71
SONUÇLAR.....	75
KAYNAKLAR	77
ÖZGEÇMİŞ.....	81
EKLER.....	82

ŞEKİLLER LİSTESİ

Şekil1: Kimyasalların Diğer Sektörlerde Kullanımı.....	7
Şekil 2:Kontrol Belgesi İşlemleri İş Akış Şeması	47
Şekil 3:İthalat Sınırı Belirlenen Kimyasallar İçin Yıllık Mevcut Durum Raporu.....	67
Şekil 4:Kontrol Belgesi Teknik Ekip Performans Değerlendirmesi	68

TABLolar LİSTESİ

Tablo 1:Ülkemizde 2008-2012 yılları arasında gerçekleşen Kimya Sektörü İthalatı... 5	5
Tablo 2: Sektörel Ürünler Bazında Kimya Sektörü İthalatı 6	6
Tablo 3: İSGÜM Tarafından Kontrolü Gerçekleştirilen Tehlikeli Kimyasal Maddeler ve Müstahzarlar..... 11	11
Tablo 4: Kimyasal Maddelerin Kullanıldığı Bazı Faaliyet Kolları 12	12
Tablo 5: SWOT Analizi 48	48

SİMGELER VE KISALTMALAR

G.T.İ.P.	Gümrük Tarife İstatistik Pozisyonu
İSG	İş Sağlığı ve Güvenliği
İSGÜM	İş Sağlığı ve Güvenliği Enstitü Müdürlüğü
KDS	Karar Destek Sistemi
MGBF	Malzeme Güvenlik Bilgi Formu
MSDS	Material Safety Data Sheet (MGBF)
NACE	Tehlike Sınıfı Kod Sistemi
SGK	Sosyal Güvenlik Kurumu

GİRİŞ VE AMAÇ

Kimya sanayi temel kimyasalları ve onun ürünleri olan gübreleri, boyaları, gazları ve tıbbi ürünleri içerdiğinden bir ülkenin toplam ekonomik gelişmesinde önemli bir rol oynamaktadır. Kimya sanayi birçok sektöre girdi sağlamaktadır. Geniş bir ürün yelpazesine sahip olan kimya sektörü ithalata bağımlı bir sektördür. Kimya sektöründe kullanılan hammaddelerin %70'i ithal edilmektedir. %30'u ise yerli üretimle karşılanmaktadır. Kimya sanayi, ağır sanayi olarak kabul edilen endüstri dallarından biridir. Erimiş metaller, toksik ve aşındırıcı maddeler, hava yoluyla solunum sisteminin maruz kaldığı kokular-dumanlar-maddeler ve gürültü, iş sağlığı ve güvenliği açısından en önemli riskleri oluşturmaktadır. Kimya sanayi sadece üretim odaklı çalışmamakta, aynı zamanda depolama ve lojistik hizmetleri de vermektedir [1].

Ancak kimyasalların kullanıldığı ve iş sağlığı ve güvenliği açısından önemli bir risk etmeni olarak karşımıza çıktığı tek sektör kimya sektörü değildir. Sanayinin hemen hemen her alanında hatta bazı hizmet sektörlerinde tehlikeli kimyasallar çalışanların sağlığını etkileyen, nihai ürün, hammadde, yan ürün veya ürünün bir parçası olarak yerini almaktadır. Bu kimyasallar parlama ve patlamaya neden olmalarının yanı sıra oksitleyici veya reaktif maddeler olabilmektedirler. Kanserojen ve mutajen kimyasallar meslek hastalıklarının sebeplerindedir.

Kullanılan kimyasal hammaddelerin ağırlıklı olarak ithal ürünler olması nedeniyle kimyasalların ithalat izni aşamasında kontrolü,iş sağlığı ve güvenliği yönünden önleyici bir yaklaşım olarak ön plana çıkmaktadır. Kullanımı uygun olmayan kimyasal maddeler ve müstahzarların piyasaya sürülmeden gümrük aşamasında tespiti, piyasaya sürülmüş ve çeşitli alanlarda kullanılan kimyasalların ise etkilerinin izlenmesi iş sağlığı ve güvenliği kapsamında alınabilecek en etkili önlemlerdendir. Bu amaçla İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğünce Kontrol Belgesi işlemleri yürütülmektedir. Mevcut durumda işlemler yazılı dilekçeler ve onaylar üzerinden yürütüldüğünden işlemlerin izlenebilirliği kısıtlıdır. İşlemlerin yürütümü ve genel değerlendirme amaçlı raporların hazırlanması önemli bir işyüktür.Örneğin kontrol belgelerine ilişkin toplam 5.000.000 sayfa evrak bulunmakla birlikte ülkeye giren kimyasalların miktarı gibi temel bilgilere kısa sürede ulaşılması güçtür. Gerçekleştirilen işlemler bir süreç gerektirdiğinden gümrük işlemlerinde yavaşlamaya,şikâyetlere neden olmakta, önemli bir işgücü ihtiyacını beraberinde getirmektedir.

Bu bağlamda, hazırlanan tez çalışmasında, kontrol belgelerine ilişkin sistemin etkinliğinin artırılmasına katkı sağlanması ve genel değerlendirme aracı geliştirilerek strateji belirlemede kullanılabilmesi amaçlanmıştır. Hazırlanan tez kapsamında kontrol belgesi başvuru ve takip süreçlerinin gerçek zamanlı olarak yürütülmesi için sistem analizi yapılarakbir model önerilmiş; süreçlerin takibi ve performans ölçümleri, kimyasalların iş sağlığı ve güvenliği açısından değerlendirilmesi ve kayıtların tutularak anlamlı veriler olarak strateji belirlemede kullanılması amacıyla bir yönetim bilgi sistemi altyapısı ve genel değerlendirme aracı oluşturulmuştur.

Kontrol belgesi başvuru ve değerlendirme işlemleri de dahil olmak üzere tüm sistemin veritabanı yapısı oluşturularak veritabanı fiziksel olarak hayata geçirilmiştir. Sistem tasarımı ve analiz çalışmaları tüm işlemler için tamamlanmıştır. Ancak elektronik imza veya mobil imza ile çevrimiçi başvuru, başvuruların kabulü ve değerlendirilmesi, sonuçlandırılarak başvuru sahibine yasal geçerliliği olan çevrimiçikontrol belgesinin verilmesi ve ilgili kurum ve kuruluşlara web servisler aracılığı ile bilgi akışı işlemlerini içeren modüllerin yapılan inceleme sonuçları ve alınan görüşler doğrultusunda 24 adam/aylık bir uzman çalışması neticesinde tamamlanabileceği tespit edilmiştir. Bu nedenle ilgili modüller için kodlama işlemleri

tez kapsamı dışında tutularak paralel bir çalışma ile yürütölmektedir. Tez kapsamında tüm sistemin analizi ve altyapısının oluşturulması ile verilerin değeriendirilerek raporlanması için yazılım modöllerinin hazırlanması üzerinde durulmuştur.

GENEL BİLGİLER

KİMYASAL MADDELERİN İTHALATI VE KONTROL BELGELERİ

Ülkemizde Kimya Sektörünün Durumu

Türk kimya sanayi, ağırlıklı olarak petrokimya, sabun, deterjan, gübre, ilaç, boya-vernik, sentetik elyaf, soda gibi çeşitli kimyasal hammadde ve tüketim ürünlerinin üretiminin gerçekleştirildiği tesislerden oluşmaktadır. Sektörde faaliyet gösteren firmalar, ölçek ve sermaye kaynakları açısından farklılık göstermektedir. Sektörde yer alan firmaların önemli bir kısmı, küçük ve orta ölçekli işletmelerden oluşmakla birlikte; büyük ölçekli firmalar ile çok uluslu şirketler de sektörde yer almaktadır.

Türk kimya sanayinin 2011 yılı TÜİK verilerine göre ithalatı 37,8 milyar ABD doları iken ihracatı 13 milyar ABD doları olarak gerçekleşmiştir. Bu da Türk Kimya Sanayinin hammadde ve teknoloji bakımından dışa bağımlı bir sektör olduğunu ve bu itibarla kimya sanayinde ithal ikamesi mahiyetinde yatırımlar yapılmasının önümüzdeki yıllarda büyük önem taşıdığını göstermektedir[2].

Kimya sektöründe dış ticaretin yarıya yakını AB ülkeleri ile yapılmaktadır. 2008 yılında ülkemizde dış ticaretin %52'si AB ülkeleri ile yapılırken, bu rakam 2009 yılında %51, 2010 yılında %49,2'ye düşmüştür. Bunun temel nedeni ekonomik kriz nedeniyle sektörün yeni pazar arayışına girmesidir. 2009 yılında dış ticaretimizin %27,2'si Yakın ve Orta Doğu ülkeleri ve Diğer Asya ülkeleri ile yapılırken bu oran 2010 yılında %29,4'e yükselmiştir. 2011 yılında AB ile dış ticaretimiz %47,3 oranına gerilerken, Yakın ve Orta Doğu ülkeleri ve Diğer Asya ülkeleri ile bu oran %31,7'ye yükselmiştir. 2012 yılında bu eğilim devam etmiştir. AB ile dış ticaretimiz %46,3 oranına gerilerken, Yakın ve Orta Doğu ülkeleri ve Diğer Asya ülkeleri ile bu oran %33,6'ya yükselmiştir[3].

Kimya sektörü ithalatı 2006 yılından itibaren her yıl artarak 2008 yılında 30 milyar ABD dolarına ulaşmıştır. Ekonomik krizin etkileri sonucu 2009 yılında ithalat, 2008 yılına göre %21 oranında azalarak 24 milyar ABD doları olarak gerçekleşmiştir. 2010 yılında ekonomik krizin etkilerini atlatan sektörün ithalatı 2009 yılına göre %26,75 oranında artarak 30,5 milyar ABD dolarına yükselmiştir. 2011 yılında ise bir önceki yıla göre %23,9 artarak 37,8 milyar ABD doları olarak gerçekleşmiş, 2012 yılında bir önceki yıla göre %4 oranında azalmıştır [2].

	2008	2009	2010	2011	2012
Kimyasalların ve kimyasal ürünlerin imalatı	22.273.924	16.992.636	22.309.364	28.204.899	27.405.977
Temel eczacılık ürünlerinin ve eczacılığa ait malzemelerin imalatı	4.735.852	4.419.350	4.770.114	5.065.194	4.326.148
Kauçuk ve plastik ürünlerin imalatı	3.453.795	2.709.151	3.493.202	4.483.930	4.476.603
KİMYA SANAYİ TOPLAMI	30.463.571	24.121.137	30.572.680	37.754.023	36.208.728
İmalat sanayi içindeki payı (%)	20,21	21,66	20,98	20,49	20,54
İMALAT SANAYİ TOPLAMI	150.711.582	111.342.174	145.714.811	184.272.373	176.234.422
Genel ithalat içindeki payı (%)	15,08	17,12	16,48	15,68	15,30
GENEL ÜLKE İTHALATI	201.963.574	140.928.421	185.544.332	240.841.676	236.544.494

Tablo 1: Ülkemizde 2008-2012 yılları arasında gerçekleşen Kimya Sektörü İthalatı (Bin ABD Doları)[3]

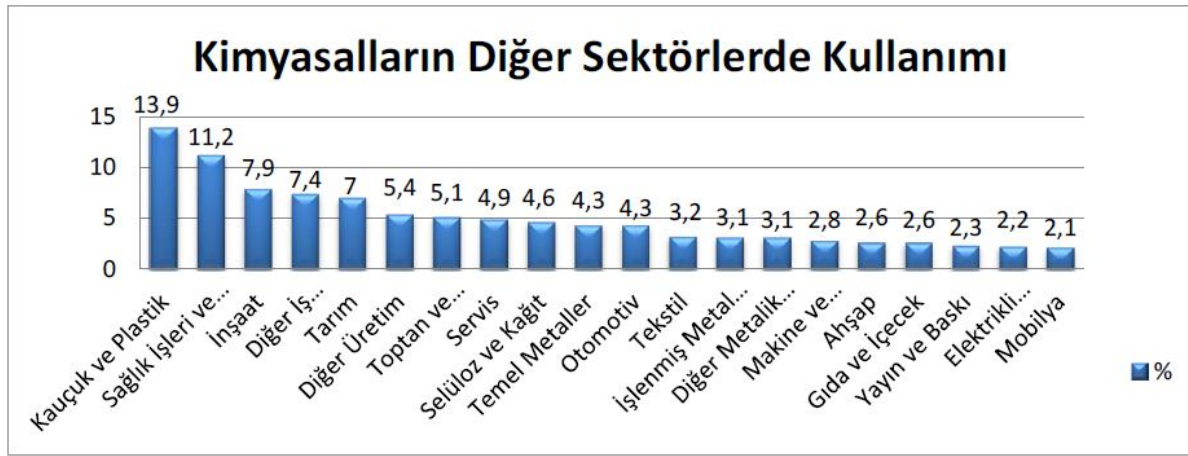
Ülkeler bazında kimya sektörü ithalatımız incelendiğinde, 2012 yılında kimya ürünleri ithalatı yaptığımız başlıca ülkelerin; Rusya, İran, İtalya, Almanya, Suudi Arabistan ve Hindistan olduğu görülmektedir. 2012 yılında tüm ülkelere yönelik kimya ürünleri ithalatımızdaki %4 oranındaki artış, daha önceki senelere göre çok düşük bir oranda gerçekleşmiştir[4].

Fasıl	Ürün	2010 (\$)	2011 (\$)	Değişim (%) 2010/11	2012 (\$)	Değişim (%) 2011/12
27	Mineral yakıtlar / yağlar	35.196.457.490	49.927.256.540	41,85	55.283.947.213	10,73
28	İnorganik kimyasallar	1.423.768.297	1.721.687.297	20,92	1.592.248.148	-7,52
29	Organik kimyasallar	4.400.234.322	5.504.434.997	25,09	5.064.683.894	-7,99
30	Eczacılık ürünleri	4.410.050.801	4.697.444.788	6,52	3.995.651.408	-14,94
31	Gübreler	1.016.776.690	1.374.433.876	35,18	1.382.707.545	0,60
32	Boya, macun, vernik	1.545.047.169	1.884.496.534	21,97	1.828.420.770	-2,98
33	Parfümeri, kozmetik	9984.293.208	1.071.449.335	8,85	1.027.868.899	-4,07
34	Sabunlar	625.557.650	813.407.302	30,03	806.543.975	-0,84
35	Albüminoid madde	429.274.234	492.772.908	14,79	494.949.712	0,44
36	Barut, patlayıcı madde	35.950.863	52.853.819	47,02	48.002.966	-9,18
37	Fotoğrafçılık, sinemacılık eşyası	223.865.593	239.921.688	7,17	214.223.350	-10,71
38	Muhtelif kimyasallar	1.795.444.922	2.204.854.048	22,80	2.053.213.144	-6,88
39	Plastik ve plastikten mamül eşya	9.730.432.011	12.578.501.480	29,27	12.505.191.520	-0,58
40	Kauçuk ve kauçuktan eşya	1.667.483.166	2.480.578.720	48,76	2.045.821.838	-17,53
Genel Toplam		63.484.636.416	85.044.093.332	33,96	88.343.474.382	3,88

Tablo 2: Sektörel Ürünler Bazında Kimya Sektörü İthalatı

Alt sektörler itibariyle 2012 yılı kimya ithalatımıza bakıldığında, ilk sırada yer alan ürünlerin ihracatımızdaki sıralamaya benzer şekilde, mineral yakıtlar ve yağlar ile plastik ve plastikten mamul eşyalar olduğu ve bunları organik kimyasallar ile eczacılık ürünleri ürün gruplarının takip ettiği görülmektedir[5].

Avrupa Kimya Sanayi Konseyi (The European Chemical Industry Council - CEFIC) tarafından yayınlanan tabloda 2012 yılı verilerine göre kimyasalların diğer sektörlerde dağılımı verilmektedir.



Şekil1: Kimyasalların Diğer Sektörlerde Kullanımı

Tehlikeli Kimyasal Maddeler

Kimyasal maddeler sanayimizin ve günlük yaşantımızın içinde bir çok alanda yer alıp katı, sıvı ve gaz hallerinde kullanılmaktadırlar. Dünyada her yıl en az 400 milyon ton kimyasal madde üretilmekte ve her tür endüstride kullanılmakta olduğundan, günümüzde çalışanların birçoğu kimyasal tehlikeye şu ya da bu şekilde maruz kalmaktadır. Bu kimyasal maddelerin birçoğu insanlara ve çevreye büyük zararlar verebilmektedir. Sanayide kullanılan kimyasal maddeler, dikkatsizlik ve yanlış kullanım sonucu oluşan is kazalarının ve sağlık problemlerinin başlıca nedenleri arasındadırlar[6].

Kimyasal Maddelerle Çalışmalarda Sağlık ve Güvenlik Önlemleri Hakkında Yönetmelik tehlikeli kimyasal maddeleri; patlayıcı, oksitleyici, çok kolay alevlenir, kolay alevlenir, alevlenir, toksik, çok toksik, zararlı, aşındırıcı, tahriş edici, alerjik,

kanserojen, mutajen, üreme için toksik ve çevre için tehlikeli özelliklerden bir veya birkaçına sahip maddeler ve müstahzarlar veya yukarıda sözü edilen sınıflamalara girmemekle beraber kimyasal,fiziko-kimyasal veya toksikolojik özellikleri ve kullanılma veya işyerinde bulundurulma şekli nedeni ile çalışanların sağlık ve güvenliği yönünden risk oluşturabilecek maddeler veya mesleki maruziyet sınır değeri belirlenmiş maddeler olarak tanımlamaktadır[7].

İlgili yönetmelik hükümlerine göre ülkemizde tehlikeli kimyasallar aşağıdaki sınıflamalara ayrılır. Bu sınıflandırma AB ülkelerindeki mevzuatlar ile uyumludur[8].

Tehlikeli kimyasal maddeler,uluslararası sınıflandırmada 9 sınıfa ayrılmıştır:

1.Sınıf Kimyasal Maddeler: “Patlayıcı Maddeler”

Patlayıcılık ve piroteknik etki elde etmek için imal edilen, depolanan, satılan, kullanılan vb. maddeleri (dinamit, barut, ateşleme fitilleri, dinamit kapsülleri, nitrogliserin vb. ile çeşitli maytap, işaret ve şenlik fişekleri gibi maddeler) kapsar. Bunlarla ilgili her türlü kontrol ve denetim Mülki amirliklerin yetkisinde olup Emniyet Müdürlüğünce, 14/08/1987 tarih ve 87/12028 sayılı Bakanlar Kurulu Kararıyla yürürlüğe konulan “Tekel Dışı Bırakılan Patlayıcı Maddelerle Av Malzemesi ve Benzerlerinin Üretimi, İthali, Taşınması, Saklanması, Depolanması, Satışı, Kullanılması, Yok Edilmesi, Denetlenmesi Usul ve Esaslarına İlişkin Tüzük” hükümlerine göre yürütülür.

2.Sınıf Kimyasal Maddeler:“Gazlar”

Özelliklerine göre, sıkıştırılmış, sıvılaştırılmış, basınç altında bir sıvıda çözülmüş veya soğukta bastırılarak sıvılaştırılmış gazlar olarak ayrılırlar.Bu gazlar, sadece kabın parçalanmasına neden olabilecek özellikte olabileceği gibi yanıcı ve zehirli özelliklere de sahip olabilmektedir.

3.Sınıf Kimyasal Maddeler: “Alev Alabilen Sıvılar”

Bu sınıfta yer alan sıvılar kapsamına 65 °C veya daha düşük sıcaklıkta tutuşabilir bir buhar oluşturabilen alev alabilen, saf sıvılar, sıvı karışımları, katıların sıvılardaki çözeltileri veya süspansiyonları (Boya, vernik, lak ve benzeri) girer.

4.Sınıf Kimyasal Maddeler: “Alev Alabilen Katılar”

Özelliklerine göre; alev alabilen katılar; kendiliğinden ısınıp, tutuşarak yanabilen maddeler; su ile temasında, yanıcı gaz oluşumuna yol açan maddeler (taşıma şartlarında kolayca tutuşabilen veya sürtünme nedeniyle ortaya çıkabilecek hafif sıcaklık artışları sonucu tutuşup yanabilen ve patlayıcı sınıfına dahil edilmeyen) ile su ile temasında yanıcı özellikte gaz oluşumuna yol açan katı maddeler olarak alt sınıflara ayrılırlar.

5.Sınıf Kimyasal Maddeler: “Oksitleyici Maddeler ve Organik Peroksitleri”

Bu sınıfta yer alan maddeler kapsamına; kendileri yanıcı olmamakla beraber, genellikle oksijen oluşturarak veya başka bir mekanizma ile diğer maddelerin yanmasına yol açan veya katkıda bulunan maddeler ile kendiliğinden patlayarak parçalanma, çok hızlı yanma, şok veya sürtünme etkisine duyarlık, başka maddelerle hızlı bir şekilde birleşme ve göze zarar verme özelliklerinden bir veya birkaçına sahip organik peroksitleri ve benzerleri, girer.

6.Sınıf Kimyasal Maddeler: “Zehirli ve Mikrop Bulaştırıcı Maddeler”

Bu sınıftaki maddeler özelliklerine göre; zehirli maddeler ve mikrop bulaştırıcı maddeler olarak iki alt guruba ayrılır. Bu kapsamdaki maddelerin yutulması, solunması veya deri ile teması sonucunda insan sağlığının bozulmasına, yaralanmaya ve ölüme yol açabilmekte, gaz halinde olanlar zehirli veya boğucu olabilmekte; insanlarda veya hayvanlarda hastalığa yol açtığı deneyimlerle bilinen veya bundan şüphelenilen mikroorganizmaları veya bunların toksinlerini içeren maddeler bu sınıfta yer almaktadır.

7.Sınıf Kimyasal Maddeler:“Radyoaktif Maddeler”

Birim kütle başına aktiviteleri 70 kBq/kg dan daha büyük olan herhangi bir radyoizotop veya bu radyoizotopun bileşikleri veya bunların başka maddelerle karışımları bu sınıfa girmektedir. Bu sınıftaki maddeler Başbakanlığa bağlı Türkiye Atom Enerjisi Kurumu yetki ve sorumluluğundadır.

8.Sınıf Kimyasal Maddeler: “Aşındırıcı (Korozyif) Maddeler”

Bu sınıfta; canlı dokular ile temasında ağır hasar oluşturan maddeler ile taşıma sırasında başka materyaller üzerine dökülünce fiziksel hasar oluşturan veya korozyona (paslanma veya aşınmaya) yol açan kimyasal bakımdan reaktif maddeler girer.

9.Sınıf Kimyasal Maddeler:“Diğer Tehlikeli Maddeler”

Yukarıda sayılan madde ve ürünlerin dışında kalan diğer tehlikeli madde ve ürünleri bu sınıf içinde değerlendirilir.

İSGÜM Tarafından Kontrolü Gerçekleştirilen Tehlikeli Kimyasal Maddeler ve Müstahzarlar

İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğünce Ekonomi Bakanlığı tarafından çıkarılan 2014/13 no’lu “İş Sağlığı ve Güvenliğini Etkileyen Bazı Maddelerin İthaline İlişkin Tebliğ” kapsamında aşağıda G.T.İ.P. numaraları verilen kimyasallara kontrol belgesi yoluyla ithalat izni verilmektedir.

G.T.İ.P.	Eşyanın Tanımı
2707.10	Benzol (benzen)
2707.20	Toluol (toluen)
2707.50.00.00.11	Solvent nafta (çözücü nafta)
2707.50.00.00.19	Diğerleri
2707.99.11.00.00	200°C'ye kadar sıcaklıkta, hacim itibariyle %90 veya daha fazla damıtılmış ham hafif yağlar
2707.99.19.00.00	Diğerleri
2710.12.21.00.00	White Spirit
2901.10.00.90.11	Hekzan
2901.10.00.90.12	Heptan
2902.20.00.00.00	Benzen (benzol)
2902.30.00.00.00	Toluen (toluol)
2902.41.00.00.00	o-Ksilen
2902.42.00.00.00	m-Ksilen
2902.43.00.00.00	p-Ksilen
2902.44.00.00.00	Ksilen izomerleri karışımları
32.08	Yalnız organik çözücüler içerisinde çözelti halinde bulunanlar
3506.10.00.90.11	Solvent içerenler
3506.91.00.90.13	Solvent içerenler
3506.99.00.90.11	Solvent içerenler
3814.00	Tarifenin başka yerinde belirtilmeyen veya yer almayan organik karma çözücüler ve incelticiler; boya ve vernik çıkarmada kullanılan müstahzarlar
3824.90.40.00.00	Vernikler ve benzeri ürünler için anorganik karma çözücüler ve incelticiler
39.01; 39.02; 39.03; 39.04; 39.05; 39.06; 39.07; 39.08; 39.09; 39.10; 39.11; 39.12 ve 39.13	Yalnız organik çözücüler içerisinde çözelti halinde bulunanlar (katı halde olanlar hariç)
40.05	Yalnız organik çözücüler içerisinde çözelti halinde bulunanlar (katı halde olanlar hariç)

Tablo 3: İSGÜM Tarafından Kontrolü Gerçekleştirilen Tehlikeli Kimyasal Maddeler ve Müstahzarlar

Kimyasalların Kullanım Alanlarının Tespitinde NACE Tehlike Sınıflandırması

Tehlikeli Kimyasallar Yönetmeliği Kapsamında ithalatı gerçekleştirilen kimyasal maddelerin kullanıldığı işyerlerinin bildirim gerekmektedir. Tez projesi ile bildirimde yer alan SGK sicil numarası üzerinden SGK veritabanlarına ulaşılarak ilgili kimyasalların kullanıldığı işyerlerinin 6 haneli faaliyet koduna ulaşılacak faaliyet alanları tespit edilecektir. Kimyasalların kullanıldığı bazı temel faaliyet alanlarına ilişkin NACE kodları aşağıdaki tabloda yer almaktadır[9].

NACE Rev.2 Altılı Kod	NACE Rev.2 Altılı Tanım	Tehlike Sınıfı
20.11.01	Sanayi gazları imalatı (hidrojen, asal gazlar, azot, oksijen, karbondioksit ve ametallerin diğer inorganik oksijen bileşikleri, soğutucu-dondurucu gazlar ile hava gibi sıvı veya sıkıştırılmış inorganik sanayi gazları ve tıbbi gazlar)	Çok Tehlikeli
20.12.01	Boya maddeleri ve pigment imalatı (birincil formda veya konsantre olarak herhangi bir kaynaktan) (hazır boyalar hariç)	Çok Tehlikeli
20.13.02	Metalik halojenler, hipokloritler, kloratlar ve perkloratların imalatı (çamaşır suyu dahil)	Çok Tehlikeli
20.13.03	Sülfidler (sülfürler), sülfatlar, fosfinatlar, fosfonatlar, fosfatlar ve nitratların imalatı (şap dahil)	Çok Tehlikeli
20.13.04	Karbonatların imalatı (sodyum, kalsiyum ve diğerleri) (çamaşır sodası dahil)	Çok Tehlikeli
20.13.06	Uranyum, plütonyum ve toryum cevherlerinin zenginleştirilmesi (nükleer reaktörler için yakıt kartuşları dahil)	Çok Tehlikeli
20.13.07	Diğer metal tuzları ve temel inorganik kimyasalların imalatı (izotoplar ve bunların bileşikleri, oksometalik/peroksometalik asitlerin tuzları, siyanürler, boratlar, hidrojen peroksit, kükürt, kavrulmuş demir piritler, piezo-elektrik kuvarı vb.)	Çok Tehlikeli
20.13.90	Başka yerde sınıflandırılmamış kimyasal elementler, inorganik asitler ve bileşiklerin imalatı (klor, iyot, flor, bor, silisyum, fosfor, arsenik gibi metaloidler, skandium, cıva, oksitler, hidroksitler, hidrojen klorür vb.)	Çok Tehlikeli
20.14.01	Temel organik kimyasalların imalatı (hidrokarbonlar, alkoller, asitler, aldehitler, ketonlar, sentetik gliserin, azot fonksiyonlu bileşikler vb.) (etil alkol, sitrik asit dahil)	Çok Tehlikeli
20.14.04	Odunun ve kömür katranının damıtılması (odun kreozotu, odun naftası, bitkisel zift, benzol, toluol, fenol, naftalin vb.)	Çok Tehlikeli
20.14.05	Çam terebentin esansı ve zamk imalatı	Tehlikeli
20.15.01	Fosfatlı veya potasyumlu gübreler, iki (azot ve fosfor veya fosfor ve potasyum) veya üç besin maddesi (azot, fosfor ve potasyum) içeren gübreler, sodyum nitrat ile diğer kimyasal ve mineral gübrelerin imalatı	Çok Tehlikeli
20.15.02	Bileşik azotlu ürünlerin imalatı (nitrik asit, sülfonitrik asit, saf amonyak, amonyum klorür (nişadır), amonyum karbonat, nitritler, potasyum nitratlar vb.) (gübreler hariç)	Çok Tehlikeli
20.16.01	Birincil formda poliamitler, üre reçineleri, melamin reçineleri, vb. plastik hammaddelerin imalatı	Çok Tehlikeli
20.16.02	Birincil formda alkid reçine, polyester reçine, epoksi reçine, poliasetal, polikarbonat ile diğer polieter ve polyester imalatı	Çok Tehlikeli
20.16.03	Birincil formda polimerlerin imalatı (etilen, propilen, stiren, vinil klorür, vinil asetat, vinil esterleri, akrilik vb. polimerleri ile sertleştirilmiş proteinler, doğal kauçuğun kimyasal türevleri dahil)	Çok Tehlikeli
20.16.04	Birincil formda silikon ve polimer esaslı iyon değiştiricileri imalatı	Çok Tehlikeli
20.16.05	Birincil formda diğer amino reçineler, fenolik reçineler, poliüretanlar, politerpenler, polisülfürler, selüloz ve kimyasal türevleri ile diğer petrol reçineleri imalatı	Çok Tehlikeli
20.17.01	Birincil formda sentetik kauçuk imalatı	Çok Tehlikeli
20.20.11	Böcek ilacı, kemirgen ilacı, küf ve mantar ilacı, yabancı otları mücadele ilacı imalatı	Çok Tehlikeli
20.20.12	Dezenfektan imalatı (tarımsal ve diğer kullanımlar için) (hijyenik maddeler, bakteriyostatlar ve sterilize ediciler dahil)	Çok Tehlikeli

Tablo 4: Kimyasal Maddelerin Kullanıldığı Bazı Faaliyet Kolları

NACE Rev.2 Altılı Kod	NACE Rev.2 Altılı Tanım	Tehlike Sınıfı
20.20.13	Çimlenmeyi önleyici ve bitki gelişimini düzenleyici ürün imalatı	Çok Tehlikeli
20.20.14	Diğer zirai kimyasal ürünlerin imalatı (gübre ve azotlu bileşik imalatı hariç)	Çok Tehlikeli
20.30.11	Polyester, akrilik ve vinil polimer esaslı boya ve vernik imalatı	Çok Tehlikeli
20.30.12	Macun imalatı (dolgu, cam, sıvama için olanlar ile üstübeç, vb. dahil)	Çok Tehlikeli
20.30.13	Diğer boya, vernik ve ilgili ürünlerin imalatı (resim ve tabelacı boyaları, matbaa mürekkepleri, hazır boya pigmentleri, matlaştırıcılar, renklendiriciler, astarlar, cam firit, camlaştırılabilir emay ve sırlar, solvent, inceltici (tiner))	Çok Tehlikeli
20.41.01	Kapalı alanlar için kokulu müstahzarlar ve koku gidericiler ile suni mumların imalatı (kişisel kullanım için olanlar hariç)	Tehlikeli
20.41.03	Ham gliserin (gliserol) imalatı	Tehlikeli
20.41.04	Sabun, yıkama ve temizleme müstahzarları (deterjanlar) ile sabun olarak kullanılan müstahzarlar imalatı (kişisel bakım için olanlar ile ovalama toz ve kremleri hariç)	Tehlikeli
20.41.06	Cila, krem ve ovalama krem ve tozlarının imalatı (ayakkabı, mobilya, yer döşemesi, kaporta, cam, metal vb. için)	Tehlikeli
20.42.01	Ağız veya diş bakım ürünleri imalatı (diş macunu, vb. ile takma dişleri ağızda sabit tutmaya yarayan macun ve tozlar ile diş temizleme iplikleri dahil)	Tehlikeli
20.42.02	Kolonya imalatı	Tehlikeli
20.42.03	Parfüm ve koku verici diğer sıvı ürün, manikür/pedikür müstahzarı, güneş koruyucu ürünler, dudak ve göz makyajı ürünü, banyo tuzu, kozmetik veya kişisel bakım amaçlı pudra, sabun ve organik yüzey aktif müstahzarı, deodorant, vb. imalatı (kolonya hariç)	Tehlikeli
20.42.04	Şampuan, saç kremi, saç spreyi, jöle, saç düzleştirme ve perma ürünleri, saç losyonları, saç boyaları, vb. imalatı	Tehlikeli
20.51.21	Barut, vb. itici tozların imalatı	Çok Tehlikeli
20.51.22	Hazır patlayıcılar, emniyet fitilleri, çarpma kapsülleri, infilak fitilleri, ateşleyiciler, dinamit, elektrikli kapsüller, havai fişekler, sis işaretleri, işaret fişekleri, vb. patlayıcı veya piroteknik malzeme imalatı (barut hariç)	Çok Tehlikeli
20.51.23	Kibrit imalatı	Çok Tehlikeli
20.52.05	Tutkal imalatı (kazein esaslı, hayvansal esaslı, nişasta esaslı, kauçuk esaslı, plastik esaslı, polimer esaslı vb. olanlar)	Çok Tehlikeli
20.59.01	Fotografik levha ve filmlerin (hassaslaştırılmış, ışığa maruz kalmamış olanlar), anında baskılanan filmlerin, fotoğrafçılıkta kullanılan kimyasal müstahzarların ve karışimsız (saf) ürünlerin imalatı	Çok Tehlikeli
20.59.03	Aktif karbon imalatı	Çok Tehlikeli
20.59.04	Yağlama müstahzarları (hidrolik fren sıvıları dahil), vuruntu önleyici müstahzarlar ile katkı maddeleri ve antifrizlerin imalatı	Çok Tehlikeli
20.59.05	Yazım ve çizim mürekkepleri ve diğer mürekkeplerin imalatı (matbaa mürekkebi imalatı hariç)	Tehlikeli
20.59.06	Peptonlar, diğer protein maddeleri ve bunların türevlerinin ve deri tozlarının imalatı	Tehlikeli
20.59.07	Laboratuvar için hazır kültür ortamları, model hamurları, kompozit diyagnostik reaktifler veya laboratuvar reaktifleri imalatı	Çok Tehlikeli

Tablo 4: Kimyasal Maddelerin Kullanıldığı Bazı Faaliyet Kolları (Devam)

NACE Rev.2 Altılı Kod	NACE Rev.2 Altılı Tanım	Tehlike Sınıfı
20.59.08	Elektronikte kullanılan macun kıvamında (dope edilmiş) olan kimyasal elementler ile bileşiklerin imalatı	Çok Tehlikeli
20.59.09	Bitirme (apreleme dahil) maddeleri, boya hammaddesi ve benzeri ürünlerin sabitlenmesini veya boyayıcılığını hızlandıran boya taşıyıcı maddelerin imalatı	Çok Tehlikeli
20.59.10	Dekapaj (temizleme) müstahzarları, eritkenler, hazır vulkanizasyon hızlandırıcı maddeler, kauçuk veya plastikler için plastikleştirici bileşikler ve stabilizatörler, başka yerde sınıflandırılmamış katalitik müstahzarlar imalatı	Çok Tehlikeli
20.59.11	Jelatin ve jelatin türevleri ile süt albüminlerinin imalatı	Çok Tehlikeli
20.59.12	Kimyasal olarak değiştirilmiş veya yenilemeyen hayvansal veya bitkisel katı ve sıvı yağlar ve yağ karışımlarının imalatı (linoksin, teknik ve sanayi amaçlı bitkisel sabit sıvı yağlar, sanayide kullanılan sıvı yağlar, vb.)	Tehlikeli
20.59.13	Biyodizel, vb. biyoyakıt imalatı (bitkisel veya hayvansal yağlardan elde edilen uzun zincirli yağ asitlerinin mono alkil esterleri) (%70 veya daha fazla petrol yağı ile karıştırılmış biyodizelden ürünler hariç)	Çok Tehlikeli
20.59.14	Başka yerde sınıflandırılmamış diğer kimyasal ürünlerin imalatı (vakum tüpleri için emiciler, pirolinyitler, kazan taşı önleyici bileşikler, yağ emülsiyonlaştırıcıları, dökümhanelerde kullanılan yardımcı kimyasal ürünler ve hazır bağlayıcılar, vb.)	Çok Tehlikeli
20.59.15	Yangın söndürücü müstahzarları ve dolun malzemeleri imalatı	Çok Tehlikeli
20.60.01	Kardelenmemiş ve taranmamış suni ve sentetik elyaf imalatı	Çok Tehlikeli
20.60.02	Sentetik filament ipliği ve sentetik monofilamentlerin, şeritlerin ve benzerlerinin imalatı (poliamidden ve polyesterden yüksek mukavemetli filament iplikler dahil) (bükülü, katlı ve tekstürize olanlar hariç)	Tehlikeli
22.11.17	Kauçuktan iç lastiklerin imalatı (dış lastikler için değişebilir sırtlar, kolonlar ve şeritlerin imalatı dahil)	Çok Tehlikeli
22.11.18	Kauçuktan dış lastik imalatı (motosikletler, bisikletler, otomobiller, otobüsler, kamyonlar, hava taşıtları, traktörler ve diğer araç ve donanımlar için) (dolgu veya alçak basınçlı lastikler dahil)	Çok Tehlikeli
22.11.19	Lastik tekerleklerinin yeniden işlenmesi ve sırt geçirilmesi (lastiğin kaplanması)	Çok Tehlikeli
22.19.01	Kauçuktan hijyenik ve eczacılık ürünlerinin imalatı (prezervatifler, emzikler, hijyenik eldivenler vb. dahil)	Tehlikeli
22.19.02	Kauçuktan tüp, boru ve hortumların imalatı (vulkanize kauçuktan)	Tehlikeli
22.19.03	Kauçuktan giyim eşyası ve giysi aksesuarlarının imalatı (giysiler, eldivenler vb.)	Tehlikeli
22.19.04	Kauçuktan süpürgelerin ve fırçaların imalatı	Tehlikeli
22.19.05	Kauçuk ayakkabı/bot tabanları ve ayakkabı/botların diğer kauçuk parçalarının imalatı	Tehlikeli
22.19.06	Kauçuktan yer döşemeleri ve paspasların imalatı	Tehlikeli
22.19.07	Kauçuk kaplanmış, emdirilmiş, sıvanmış ve lamine edilmiş tekstil kumaşlarının imalatı, ana bileşeni kauçuk olanlar (kord bezi hariç)	Tehlikeli
22.19.08	Kauçuktan paket lastiği, tütün kesesi, cam silecekleri, tarih ıstapmaları için karakterler, tapalar, lavabo pompaları, şişeler için tıpa ve halkalar ile sert kauçuktan diğer çeşitli eşyaların imalatı	Tehlikeli
22.19.09	Kauçuktan konveyör bantları ve taşıma kayışlarının imalatı	Tehlikeli

Tablo 4: Kimyasal Maddelerin Kullanıldığı Bazı Faaliyet Kolları (Devam)

NACE Rev.2 Altılı Kod	NACE Rev.2 Altılı Tanım	Tehlike Sınıfı
22.19.10	Rejenere kauçuk imalatı, birincil formda veya levha, tabaka veya şerit halinde	Tehlikeli
22.19.12	Kauçuktan silgi, rondela, conta, tekne veya iskele usturmaçaları, gözenekli vulkanize kauçuktan teknik işlerde kullanılan diğer eşyalar ile demiryolu, kara yolu taşıtları ve diğer araçlar için kalıplanmış parçaların imalatı	Tehlikeli
22.19.13	Vulkanize edilmiş (kükürtle sertleştirilmiş) kauçuk imalatı (ip, kordon, levha, tabaka, şerit, çubuk ve profil halinde)	Tehlikeli
22.21.03	Plastikten mamul halde tüp, boru, hortum ve bunların bağlantı elemanlarının imalatı (suni bağırsaklar dahil)	Tehlikeli
22.21.04	Plastikten yarı mamul halde profil, çubuk, tabaka, levha, blok, film, folyo, şerit, vb. ile monofilament imalatı (nylon brandalar dahil)	Tehlikeli
22.22.43	Plastik poşet, çöp torbası, çanta, torba, çuval, file, sandık, kutu, kasa, damacana, şişe, bidon, makara, masura, bobin, tıpa, kapak, kapsül vb. paketleme malzemelerinin imalatı (idrar torbası dahil)	Tehlikeli
22.23.03	Plastikten depo, tank, fıçı ve benzeri kapların imalatı	Tehlikeli
22.23.04	Plastikten prefabrik yapıların imalatı	Tehlikeli
22.23.05	Vinil, linolyum (muşamba) gibi esnek yer kaplamaları ile plastik zemin, duvar ve tavan kaplamalarının imalatı (duvar kağıdı hariç)	Tehlikeli
22.23.06	Plastikten merdiven, merdiven korkuluğu, panjur, güneşlik, jaluzi, stor, vb. eşya ile bunların parçalarının imalatı	Tehlikeli
22.23.07	Plastikten banyo küvetleri, lavabolar, klozet kapakları, oturakları ve rezervuarları ile benzeri sıhhi ürünlerin imalatı (kalıcı tesisat için kullanılan montaj ve bağlantı parçaları dahil)	Tehlikeli
22.23.08	Plastikten/PVC'den kapı, pencere, bunların kasaları, pervazları, kapı eşikleri, vb. imalatı	Tehlikeli
22.23.90	Başka yerde sınıflandırılmamış plastik inşaat malzemelerinin imalatı (plastik suni taş-mermerit imalatı)	Tehlikeli
22.29.01	Plastikten sofrta, mutfak, banyoda kullanılan eşya (silikon kek kalıbı, leğen, tas, kova vb.) ve diğer ev eşyası imalatı	Tehlikeli
22.29.02	Plastikten dikişsiz giyim eşyası ve giysi aksesuarlarının imalatı (eldiven dahil)	Tehlikeli
22.29.03	Plastikten büro ve okul malzemelerinin imalatı	Tehlikeli
22.29.04	Ayakkabı ve terliklerin plastik parçalarının imalatı (plastik ayakkabı kalıbı imalatı dahil)	Tehlikeli
22.29.05	Makine, mobilya, kaporta, el aletleri ve benzerlerinin plastikten bağlantı parçaları, plastikten taşıyıcı bantların ve konveyör bantlarının imalatı	Tehlikeli
22.29.06	Plastik başlık (koruma amaçlı olanlar hariç), izolasyon bağlantı parçaları ile lambaların, aydınlatma ekipmanlarının, ışıklı tabelaların, vb.nin başka yerde sınıflandırılmamış plastik kısımlarının imalatı	Tehlikeli
22.29.07	Plastikten mandal, askı, sünger, sabunluk, tarak, bigudi, toka, saç firketesi, boncuk, biblo, heykelcik ve diğer eşyalar ile mamul haldeki kendinden yapışkanlı levha, şerit vb. ürünlerin imalatı	Tehlikeli
22.29.90	Başka yerde sınıflandırılmamış diğer plastik ürünlerin imalatı	Tehlikeli

Tablo 4: Kimyasal Maddelerin Kullanıldığı Bazı Faaliyet Kolları (Devam)

6331 sayılı İş Sağlığı ve Güvenliği Kanununun “Tehlike sınıfının belirlenmesi” başlıklı 9 uncu maddesinde; işyeri tehlike sınıflarının, 31/5/2006 tarihli ve 5510 sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanununun 83 üncü maddesine göre belirlenen kısa vadeli sigorta kolları prim tarifesi de dikkate alınarak, İş Sağlığı ve Güvenliği Genel Müdürünün Başkanlığında ilgili taraflarca oluşturulan komisyonun görüşleri doğrultusunda, Bakanlıkça çıkarılacak tebliğ ile tespit edileceği, hükmü yer almaktadır.

Belirtilen hükme istinaden belirlenen işyeri tehlike sınıfları, 26/12/2012 tarihli ve 28509 sayılı Resmi Gazete’de İş Sağlığı ve Güvenliğine İlişkin İşyeri Tehlike Sınıfları Tebliği ekinde yayımlanmıştır. Ayrıca, 25/11/2009 tarihli ve 27417 sayılı Resmî Gazete’de yayımlanan İş Sağlığı ve Güvenliğine İlişkin Tehlike Sınıfları Listesi Tebliği de yürürlükten kaldırılmıştır[10].

Söz konusu tebliğde yer alan hükümler uyarınca;

Tehlike sınıfının tespitinde bir işyerinde yürütülen asıl işin tehlike sınıfı dikkate alınır. İşveren asıl iş faaliyet değişikliğini en geç bir ay içerisinde Çalışma ve Sosyal Güvenlik Bakanlığına bildirir.

Asıl işin tayininde tereddüde düşülmesi halinde işyerinin kuruluş amacına bakılır. İşyerinde birden fazla asıl iş tanımına uygun faaliyetin yürütülmesi halinde, bu işlerden tehlike sınıfı yüksek olan iş esas alınır.

İşyeri tehlike sınıfına yapılan itirazlar Çalışma ve Sosyal Güvenlik Bakanlığınca değerlendirilir. Değerlendirme, Bakanlık tescil kayıtları üzerinden ve işyerinin tesciline esas alınan asıl iş dikkate alınarak sonuçlandırılır. Gerekli görülmedikçe işyerlerinde yapılan asıl işe ilişkin inceleme yapılmaz.

Bakanlıkça yapılan denetim ve incelemelerde işyerinde yapılan asıl işin tescil kayıtlarından farklı olması halinde, denetim ve incelemeye ilişkin kayıtlar dikkate alınarak işyeri tehlike sınıfı yeniden belirlenebilir.

Alınan karar işyerine tebliğ edilir. Kararın işyerinin tehlike sınıfı değişikliğine neden olması halinde tehlike sınıfının gerektirdiği iş ve işlemleri, işverenler 90 gün içinde yerine getirmek ve İş Sağlığı ve Güvenliği Genel Müdürlüğüne bildirmek zorundadır.

İşyeri Tehlike Sınıfları Listesindeki faaliyet kodunun tehlike sınıfına ait itirazlardan, İş Sağlığı ve Güvenliği Genel Müdürlüğü tarafından oluşturulacak Komisyonda alınacak karar sonucunda, itiraza ilişkin faaliyet kodunun tehlike sınıfının değişmesi halinde ilgili tebliğde gerekli değişiklik yapılır[9].

Kimyasalların Zararlı Etkilerinin Belirlenmesinde MSDS dokümanlarının Kullanımı

Güvenlik Bilgi Formu, tehlikeli maddelerin ve müstahzarların; özelliklerine ilişkin ayrıntılı bilgileri, bulunduğu işyerlerinde madde ve müstahzarın tehlikeli özelliklerine göre alınacak güvenlik önlemlerini insan sağlığı ve çevrenin, tehlikeli maddelerin ve müstahzarların olumsuz etkilerinden korunmasına yönelik gerekli bilgileri içeren belgeyi ifade eder[11].

Güvenlik Bilgi Formları, tedarik zincirinde yer alanlara madde ve karışımların kullanımından ortaya çıkan risklerin yönetimiyle ilgili sorumluluklarını karşılamada yardımcı olacak bilgileri sunan ve Avrupa Birliği tarafından da kabul gören, kimyasallar ürünlerin küresel anlamda aynı prosedürlerle dolaşımının sağlanması amacıyla Birleşmiş Milletler nezdinde 1970'li yıllarda yapılan çalışmalar günümüzde Global Harmonized System (GHS) başlığı altında küresel uyum sistemi haline getirilmiş önemli iletişim araçlarıdır. Güvenlik Bilgi Formları (EC) No 1907/2006 (REACH) Yönetmeliği sisteminin ayrılmaz parçalarıdır.

Güvenlik Bilgi Formu, kimyasal ürünlerin kimlik kartı gibidir; bu formlarda üreticiler hakkında bilgi, ürün hakkındaki tehlike bilgileri, ürün içeriği, ürüne maruz kalınması halinde alınacak sağlık ve çevre ile ilgili tedbirler, taşınması, depolanması ve kullanımı esnasında alınacak tedbirler ve sınıflandırılması, yangın ve kaza halinde neler yapılacağı, atık olması durumundaki bertaraf bilgileri, toksikoloji ve ekotoksikoloji bilgileri gibi bilgiler verilmektedir.

Türkiye’de 26 Aralık 2008 tarihinde yürürlüğe giren “Tehlikeli Maddeler ve Müstahzarlara İlişkin Güvenlik Bilgi Formlarının Hazırlanması ve Dağıtılması Hakkında Yönetmelik” ile üreticiler, ithalatçılar ve dağıtıcı firmalar müşterilerine Türkiye’de yayınlanmış sınıflandırma ve etiketleme mevzuatına göre Türkçe olarak Güvenlik Bilgi Formları vermekle yükümlüdürler. Bu Güvenlik Bilgi Formlarını, akredite edilmiş bir kurum tarafından sertifikalanmış personeller hazırlar. İthalatlarda da bu yükümlülük geçerlidir, ithal edilen ürünlerin Türkçe dilinde ve mevzuatında yeniden düzenlenmesi zorunlu hale getirilmiştir[12]. Kontrol Belgesi Müracaatlarında verilmesi zorunlu olan MSDS dokümanlarının ilgili maddeleri incelenerek kimyasalın hangi zararlı etkilerinin olduğu insan sağlığını ve güvenliğini ne şekilde etkileyeceği ve korunma için yapılması gerekenler sistem içerisinde istatistiksel olarak incelenerek raporlar içerisinde kullanılmaktadır.

GEREÇ VE YÖNTEMLER

İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİNİN OLUŞTURULMASI

Çalışmanın Amacı

İthal edilen tehlikeli kimyasalların kontrolü için 2014/13 sayılı “İş Sağlığı ve Güvenliğini Etkileyen Bazı Kimyasalların İthalatına İlişkin Tebliğ” gereğince İSGÜM tarafından düzenlenen kontrol belgesi başvuruları, yıllık 20.000-22.000 başvuruya ulaşmakta ve her başvuruda yaklaşık 20-25 sayfa doküman bulunmaktadır. Dokümanların mevcut yapıda evrak sistemine aktararak incelenmesi ve sonuçlandırılması uzun zaman almakta ve büyük iş gücü gerektirmektedir. İşlemlerin mümkün olduğunca seri şekilde gerçekleştirilmesi önem arz etmektedir. Aksi halde gecikme nedeniyle hem kimyasalların ülkemizde kullanımını geciktirmekte hem de gümrükte bekleme sürelerinin uzaması şikâyetlere sebebiyet vermektedir.

Diğer bir önemli konuda; Malzeme Güvenlik Bilgi Formlarının standardizasyonunun sağlanmasıdır. Malzeme Güvenlik Bilgi Formları kimyasal maddelerin kullanımı ve depolanması sırasında oluşabilecek İş Sağlığı ve İş Güvenliği risklerini ortadan kaldırmaya yönelik çalışmaların önemli bir parçasını oluşturan ve kullanıcıyı doğru ve yeterli düzeyde bilgilendirmek amacıyla hazırlanan, ilgili kimyasal maddelerin tehlike ve riskleri ile diğer bilgileri içeren dokümanlardır.

Piyasaya arz edilen tehlikeli maddelerin ve müstahzarların insan sağlığı ve çevre üzerinde yaratabilecekleri olumsuz etkilere karşı etkin kontrolünü ve verimli gözetimini sağlamak temel amaçtır. Ancak mevcut durum içerisinde formların standart başlıklar altında farklı formatlarda hazırlanıyor olması kontrol imkanını azaltmaktadır. Bu nedenle hazırlanacak sistem içerisinde Malzeme Güvenlik Bilgi Formları belli bir formatta ve veri yapısı içerisinde alınacaktır. Bu sayede formlar üzerinde arama ve incelemeler mümkün hale gelecektir.

Kontrol belgesi sisteminde üretilecek veriler sayesinde bir üretim tesisinde İş sağlığı güvenliği açısından hangi tedbirlerin alınması gerektiği listelenebilecektir. Bu bilgiler işyerlerinde gerçekleştirilecek denetimlerde yardımcı olacaktır.

Kontrol belgesi işlemleri esnasında toplanan verilerin İş Sağlığı ve Güvenliği istatistikleri açısından değerlendirilmesiyle kimyasalların İSG yönünden etkilerinin sektörel bazda incelenebilmesi ve meslek hastalıkları ve iş kazaları ile ilişkilendirilebilmesi mümkün olacaktır. Örneğin; SGK ile gerçekleştirilen iş birliği çerçevesinde çalışanların maruz kaldığı hastalıkların faaliyet alanı bazında sınıflandırılması ile ithal edilen tehlikeli kimyasalların kullanıldığı alanlarda görülen hastalıklar ilişkilendirilebilecektir. Yada gerçekleşen iş kazaları ile malzeme güvenlik bilgi formlarında yer alan korunma önlemleri arasında korelasyonlar çıkarılabilecektir.

Çalışmanın Altyapısı ve Yasal Dayanağı

- 2014/13 sayılı İş Sağlığı ve Güvenliğini Etkileyen Bazı Kimyasalların İthalatına İlişkin Tebliğ
- Tehlikeli Maddelerin ve Müstahzarların Sınıflandırılması, Ambalajlanması ve Etiketlenmesi Hakkında Yönetmelik
- Tehlikeli Maddeler ve Müstahzarlara İlişkin Güvenlik Bilgi Formlarının Hazırlanması ve Dağıtılması Hakkında Yönetmelik

Çalışmanın Hedefleri

- Kontrol Belgesi işlemlerinin hızlandırılarak daha etkin hizmet sağlanması
- Kontrol Belgesi otomasyonunun sağlanması ile personel iş yükünün azaltılması
- Kontrol Belgesi başvurularının online gerçekleştirilmesi ile evrak işlemlerinin ortadan kaldırılması
- Malzeme Güvenlik Formlarının standart hale getirilmesi ve formlar üzerinde arama ve inceleme işlemlerinin gerçekleştirilmesi
- Hâlihazırda yazılı olarak yürütülen işlemlerin sayısal ortama taşınması ile istatistiksel verilerin üretilmesi ve iş sağlığı ve güvenliğinin artırılmasında kullanılması
- Kontrol Belgesi işlemlerinde barkodlama sistemi ve web servisler kullanılarak muhtemel sahteciliklerin önüne geçilmesi
- Kamu kurum ve kuruluşları ile bilgi paylaşımına olanak sağlanması
- Loglama işlemleri ile bilgi güvenliğinin sağlanması

Çalışmadan Faydalanacak Taraflar

Çalışma ile ülkemizde tehlikeli kimyasal maddeleri hammadde olarak kullanan her türlü üretici, imalatçı, ithalatı gerçekleştiren dış ticaret firmaları, kimyasalları iç piyasada satın alan ve kullanan tüm sektörlerdeki firmalar denetim altına alınacak, hangi kimyasalın hangi sektörde ne kadar kullanıldığı ülkeye dağılımı tespit edilecektir. Ayrıca kurulacak sistem ile Gümrük ve Ticaret Bakanlığına veri akışı sağlanabilecek muhtemel sahteciliklerin önüne geçilecektir. Oluşturulacak veritabanı ile üretilen istatistikler, birçok kuruma çalışmalarında yardımcı olacaktır.

Çalışmanın tehlikeli kimyasal maddelerin ithalatı ile uğraşan firmaların gümrük işlemlerini kolaylaştıracağı ve gümrükte bekleme maliyetlerini azaltacağı göz önünde bulundurulduğunda ilgili ithalat sektöründe bir miktar iyileşme sağlayacağı ve bu durumun artan rekabet dolayısı ile düşecek ithal ürün fiyatları nedeniyle iç piyasayı da olumlu etkileyerek üretim sektörünün büyümesine destek olabileceği düşünülmektedir.

Ayrıca istihdam edilenlerin daha sağlıklı ve güvenli bir ortamda çalışmaları sağlanarak işyerlerinde verimliliğin ve üretimin artacağı, bu artışın istihdama olumlu katkısı olacağı ve ayrıca İSG alanında verilecek eğitimlerle ülke genelinde istihdamın niteliğinin geliştirileceği düşünülmektedir.

Kontrol Belgesi İşlemleri Sistem Analizine İlişkin Veri Kaynakları

Çalışma kapsamında, kontrol belgelerinin kayıt ve takip sisteminin iş akışının analizi ve etkin bir sistem modeli tasarlanması amacıyla ilgili taraflarla birebir görüşmeler ve kurumsal toplantılar gerçekleştirilmiştir. Bu kapsamda, özellikle sistemden faydalanan taraflar, benzer uygulamaları olan taraflar ve tasarlanan sistemin çalışması için işbirliği yapılması gereken taraflar görüşmeye dahil edilmiştir.

Çalışmanın ilk aşamasında mevcut durumun tespiti, sorunların belirlenmesi, yetkililerin ve konu ile ilgili çalışan personelin görüş ve çözüm önerilerinin alınması amacıyla görüşmeler gerçekleştirilmiştir. Farklı yapılarda başvuru örnekleri alınarak incelenmiştir. Gerçekleştirilen belge onayının yasal altyapısı konusunda bilgi alınmıştır. İş akışı ve süreçlerin tespiti yapılmış, kullanılan standartlar, malzeme güvenlik dokümanı yapısı, risk kodları, kimyasal kayıt numaraları gibi listeler alınmıştır.

Başvuru yapan firmaların yetkilileri ile görüşülerek kendilerinin talep ve beklentileri alınmıştır.

İthalatçı firmaların sisteme girişinin en sorunsuz ve etkin şekilde mümkün kılınması amacıyla benzer bir yapıyı e-bildirge sistemi üzerinden uygulayan Sosyal Güvenlik Kurumu Hizmet Sunumu Genel Müdürlüğünden ilgili Daire Başkanı ve Bilişim uzmanları ile görüşmeler yapılmış teknik altyapıya ilişkin bilgiler alınmıştır.

Yetki kontrollerinin alt yapısı ile sistemin oluşturulması, sunucular üzerinde barındırılması ve destek hizmetlerinin sağlanması amacıyla Bakanlık Bilgi İşlem Dairesi Başkanlığı ile görüşmeler gerçekleştirilmiştir.

İSGÜM'den alınan Kontrol Belgesini talep eden ve kendi mevzuatı içerisinde ithalat izinlerini veren kurum olarak Gümrük ve Ticaret Bakanlığı ile görüşmeler yapılmıştır.

SİSTEMİN TASARIMINDA KULLANILAN YAKLAŞIMLAR

Bilgi Sistemi Yapısı

Çalışma kapsamında oluşturulan sistem verilerin tutulması ve sorgulanmasını ötesinde verileri anlamlı bilgiler halinde kullanılması amacıyla farklı seviyelerdeki bilgi sistemlerinin (Yönetim bilgi sistemleri ve karar destek sistemleri) birleşimi kullanılmıştır.

Yönetim Bilgi Sistemleri

Yönetim bilgi sistemleri (YBS), bir yönetim destek sistemi olup, işletmenin şimdiki ve gelecekteki tüm faaliyetleriyle ilgili raporların hazırlanmasında ve kararların verilmesinde önemli rol oynamaktadır. YBS, bilgiyi doğru zamanda, doğru kişiye uygun, hızlı ve güvenilir biçimde iletecek sistem ve sistem araçlarını kullanarak hizmet vermektedir. Bu tür bilgi sisteminin temel noktası, gerek alt gerek orta gerekse üst düzeydeki yönetici kademesine kararlarını kolaylaştıracak ve onları bilgilendirecek yönde raporlar sunmaktır.

Sprague ve McNurlin YBS'ni beş temel amaca bağlamıştır[13]. Bunlar 1) Çalışma zamanını azaltmak 2) Çalışma maliyetini azaltmak 3) Değerli uzmanlığı taklit etmek 4) Yanıt süresini azaltmak 5) Değerli olan bilgileri saklamaktır[14].

Yönetim bilgi sistemlerine çok karmaşık alanlardan bilgi akışı olmaktadır. Bu karmaşık bilgi akışında gerekli bilgilerin, hangi yöntemlerle seçileceği, kaydedileceği ve etkin kararlar alınması için yöneticinin nasıl besleneceği, etkin bir yönetim bilgi sisteminin varlığı ile mümkündür. Yönetim bilgi sistemi kendi içerisinde ihtiyaç duyduğu alt bilgi sistemlerini de oluşturur. Bu bilgi sistemleri birbirleriyle sürekli bilgi alışverişi içerisinde.

Alt bilgi sistemleri aralarındaki bilgi akışı suretiyle, işletmenin bütün fonksiyonlarını, dolayısıyla bütün yönetim işlevlerini ve yönetim basamaklarını birbirine bağlayarak işletmeyi bir sistem şeklinde bütünleştirir[15].

Yönetim Bilgi sistemi, yüzeysel bir donanım ve yazılım değişikliğinden çok bir teknoloji-insan uyumu projesi olarak algılanması gereken bir olgudur. Bilgi sistemini bilişim sistemlerinden ayıran en önemli fark veri tabanı yönetimine verilen ağırlıktır[16].

Karar Destek Sistemleri

Genel olarak karar destek sistemleri (KDS) yönetici durumundaki yetkililerin karar vermelerinde yardımcı olan sistemlerdir. KDS'leri karar vermeyi kolaylaştırmak ve daha etkili hale getirmek için tasarlanan değişik model, simülasyon ve uygulamaları kapsamı içinde bulunduran sistemlerdir [17]. Bu sistemler yöneticilerin karar vermesine yardımcı olan interaktif ve bilgisayar ortamında olan sistemlerdir.

KDS'ler yöneticilerin karar vermede yardımcı olacak veriye ulaşmasına, özetlemesine ve analiz etmesine yardımcı olur[18]. Esas olan kararın alınması değil, kararın alınmasına destek olmaktır[19]. KDS'nin en az kapsaması gereken ya da erişim sağlaması zorunlu olan hususlar aşağıda sıralanmaktadır[20]. Sistem konusundaki bilgiyi yönetecek bilgielemanları: 1) veri-tabanı; sistem konusundaki tarihsel, güncel ve eksiksiz bilgiler mevcut olmalıdır 2) sayısal modeller; sistem konusunda yeni bilgiler üretir 3) uygulama araçları; politik gereksinimlere dayanarak, bilgiyi sunmak ve farklı biçimlerde karşılaştırmak için kullanılır[14].

Kontrol Belgesi Yönetim Bilgi Sistemi

İthal Edilen Tehlikeli Kimyasal Maddelerin Gözetimi, Kontrol Belgesi Başvuru ve Takip Sistemi verileri yönetim bilgi sistemi anlayışıyla anlamlı bilgiler halinde; erişilebilir ve düzenlenebilir şekilde kullanıcı dostu arayüzler aracılığı ile tutmakta ve yetki gruplarına özelleştirilmiş olarak kullanıcılar ile paylaşmaktadır.

Sistem farklı istatistikleri listeler ve farklı tipte grafikler ile destekleyerek ham bilgiler yerine işlenmiş ve değerlendirmeye hazır hale getirir. Ayrıca tehlikeli kimyasallarla ilgili politika ve stratejilerin belirlenmesi amacıyla temel düzeyde karar destek raporlamaları sunar.

Programlama Teknolojisi Altyapısı

İnternet, dünyamızda çok geniş bir kullanım alanı bulmuş ve çok hızlı bir şekilde yayılma göstermiştir. Bu yayılma sonucunda, insanların istedikleri her konudaki bilgiye internet üzerinden çok kolay bir şekilde ulaşması sağlanmıştır. Günümüzde birçok özel teşebbüs ve kamu kurumları, faaliyet gösterdikleri alanların birçoğunu internet üzerinden gerçekleştirmektedir. İnternet, kullanıcılarına birçok kolaylıklar sağlaması yanında, zamandan ve işgücünden çok önemli tasarruflar sağlamaktadır.

Kullanıcıların zaman ve işgücünden tasarruf sağlayabilmeleri için bazı araçlara gereksinim duyulmaktadır. İnternet üzerindeki bu araçlar ise web tabanlı yazılımlardır. İnternetin hızlı bir şekilde yayılması, web tabanlı yazılımlara olan talebi arttırmıştır. Artan bu taleplerin karşılanması için bu konu ile uğraşan çeşitli kurum ve kuruluşlar yazılımlar üreterek bu pazarda yer almaya çalışmışlardır. İnternet veya web tabanlı yazılımlar diğer kişisel yazılımlara göre çok fazla avantajlarının olması nedeniyle tercih edilmektedir. Bu avantajlardan bazıları,

- Kurulumunun ve kullanımının kolay olması,
- Ekstra masrafların yapılmaması,
- Güncelleme ve program değişikliklerinin tek noktada yapılması nedeniyle daha az işgücü kullanılması,
- Çok geniş kitlelere ulaşma imkânı,
- Bakım masraflarının oldukça az olması ve bakım ve kurulum kolaylığı
- Yönetiminin ve kontrolünün çok kolay olması,
- İnternet bağlantısı olan her yere ulaşma imkânı sağlaması,
- Düşük ilk maliyet ve bakım maliyeti,
- Erişilebilirlik, olarak sıralanabilir[21].

İnternet kullanımının yaygınlaştığı ilk yıllarda, web sayfaları durağan olarak hazırlanmaktaydı. Web sunucusundan, istemci (client) tarayıcılarına (browser) gönderilen içerik, tüm istemciler için aynıydı. Durağan web sayfalarında, web sitesi ziyaretçileri hakkında bilgi toplamak, ziyaretçi tercihlerini öğrenmek gibi sunucu ile ziyaretçi arasındaki etkileşim mümkün değildi. Bu ihtiyacı çözmek için 1990'lı yıllardan itibaren sunucu taraflı programlama(server side programming) dilleri geliştirilmeye başlandı. Bu amaçla ilk olarak CGI (Common Gateway Interface) programlama dili geliştirildi. CGI dilinin kullanımının zorluğu nedeniyle daha kolay ve anlaşılır diller olan Perl, ASP, PHP, Cold Fusion gibi farklı firmaların geliştirdiği fonksiyonel diller kullanılmaya başlandı. Sonraki yıllarda nesne tabanlı programlama (Object Oriented Programming) dilinin gücü ve kolaylıkları ile birlikte Sun firması tarafından Java Servlet ve JSP teknolojisi ile Microsoft firması tarafından ASP.NET teknolojisi geliştirildi. Günümüzde fonksiyonel dil olarak PHP ve ASP, nesne tabanlı dil olarak da Java Servlet ve JSP ile ASP.NET teknolojisi yaygın bir şekilde kullanılmaktadır[22].

Java Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış açık kodlu, nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilen (yorumlanan-interpreted) bir dildir[23].

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

Java bir nesne tabanlı programlama dilidir. Nesneye dayalı programlama bütün hesaplamaların nesnelere bağlamında yapıldığı bir programlama yaklaşımıdır.

Nesneye dayalı programlama diliyle yazılmış bir program verilen görevi gerçekleştirmek için biraraya gelmiş nesnelere topluluğu olarak görülebilir[24].

Nesneye yönelik programlama, modern bir programlama paradigmasıdır. Klasik, bir başka deyişle prosedürel programlamadan farklı olarak bir takım yeni programlama yapıları / olanakları içerir. Bu farklılık ve yenilikler, prosedürel programlamanın yazılım sorunlarına artık çözüm üretememesinden veya çözüm üretse bile maliyetli, güvensiz, vs. olmasından kaynaklanmaktadır.

Hemen hemen tüm diller nesneye yönelik programlama yapmaya imkan verir. Bununla beraber salt nesneye yönelik programlama yapmak için kullanılan, Java gibi programlama dilleri mevcuttur.

Her türlü yazılımda, ister web tabanlı olsun, ister mobil olsun nesneye yönelik programlama kullanılmaktadır. Bu sebeple web tabanlı yazılımlar geliştirirken de nesneye yönelik programlamayı kullanmak avantaj sağlamaktadır. Hatta bu bir avantajdan ziyade, artık zorunluluk olma yolunda ilerlemektedir.

Nesneye yönelik programlama paradigmasının getirdiği yeniliklerden birkaçı ve en önemlileri şu şekilde sıralanabilir;

- Sınıf ve nesne kavramları
- Encapsulation (Kapsülleme)
- Kalıtım – Sınıf türetme
- Çok biçimlilik – Soyutlama
- Nesne arayüzleri
- Overriding (Yeniden Tanımlama)

Yazılan kodun daha derli toplu olması, daha standart olması, daha güvenli olması veya daha esnek olması gibi sebeplerden dolayı MVC denen programlama mimarisi kullanılır.

MVC (model-view-controller), 3 katmandan oluşur ve katmanlar arasında veri akışını sağlar. Model kısmı işlemlerin gerçekleştiği kısım, View kısmı kullanıcı tarafından görüntülenen kısım (kullanıcı ile iletişime geçen kısım) ve Controller kısmı da bu işleyişi yürüten kısımdır.

Sonuç olarak nesneye yönelik programlamadaki asıl amaç profesyonel bir kod düzeni ile güvenliği sağlam uygulamalar geliştirmektir. Bu doğrultuda MVC mimarisini kullanmak avantaj sağlar[25].

Kullanılan Programlama Teknolojisi

Çalışma kapsamında gerçekleştirilen projede istemci makinalarda herhangi bir kurulum gerektirmeyen, heryerden ve her zaman ulaşım imkanı olan web tabanlı yazılım dilleri kullanılmıştır.

Senkronizasyon gerektirmeyen online veritabanı ile gerçek zamanlı raporlar alınabilmektedir. JSP Servlet java teknolojisi ile hazırlanan yazılım nesne tabanlı programlamanın bakım ve performansı avantajlarından yararlanılmasını sağlamakta java classları ile iş mantığı sunucu tarafında işletilmektedir. Güvenlik açısından veri kullanımı gerektirmeyen kısımlarda statik html sayfalarına yer verilmiş olup javascript teknolojisi validasyon işlemlerinin istemci makinalarda gerçekleştirilmesine olanak vermektedir. Formlara dinamik olarak verilerin çekilmesinde ajax kodları ile ilgili veril derleyen sayfalara bağlantılar yapılmış böylece sayfaların yenilenmesi ihtiyacının önüne geçilmiştir. Yazılım GNU lisanslı NetBeans 7.4 idesi ile gerçekleştirilmiş olup JRE 1.7, JDK 1.7 üzerinde çalışan Apache Tomcat 7.0.52 versiyonu kullanılmıştır.

Veritabanı işlemleri ilgili bölümde anlatıldığı üzere normalize edilmiş tablolar üzerinden gerçekleştirilmekte olup Oracle Database 12c versiyonu kullanılmaktadır.

Veritabanı Yapısı

Veritabanlarının ilk kullanımlarından itibaren çok çeşitli mimarilerde teknolojiler gelişmiştir. Dosya veritabanları ilk kullanılan mimaridir. Günümüzde ise en sık kullanılan yapılar ilişkisel mimari ve nesne ilişkisel mimaridir. Tez kapsamında

nesnelere fiziksel ortamda saklanmakta olup, veritabanında nesnelere erişim için adreslere yer verilmektedir.

İlişkisel veritabanlarının teorisi, 1970 yılında IBM araştırma laboratuvarında çalışan bir matematikçinin, Edward Frank Codd'un hazırladığı bir araştırma yazısı ile doğdu. Hiyerarşik ve ağ yapılı veritabanı modellerinde kullanıcının veritabanı yapısını öğrenmeleri gerekiyordu. Bu gerekliliği ortadan kaldırma isteği, bu bilimsel araştırmanın yapılmasını teşvik etti. Diğer sebepler ise, kayıt ya da veri tekrarlılığına temel bir çözüm getirmek ve fiziksel uygulama üzerindeki bağımlılığı sona erdirmektir. Codd'un araştırması, matematiğin iki dalını temel alır: Küme teorisi ve önerme mantığı. İlişkisel veritabanı adını, küme teorisinin bir bileşeni olan "küme ilişkileri" ifadesinden alır. Dr. Codd yazısında, verinin kayıtlar yerine, ilişkiler ya da tablolar içinde depolanmasının veri bağımsızlığını kolaylaştırdığını ileri sürmüştür. Uygulama programları etkilenmeden verilerin değiştirilmesi mümkündür. Bu da, ağ veritabanlarının sorunlarından birini çözmek demektir.

Özellikle iki nedenden dolayı ilişkisel veritabanı mimarisi önem arz etmektedir. Birincisi çok geniş ve yaygın kullanım alanı vardır ve bağımsız veritabanı tasarımında kullanılır. İkincisi ise, veritabanı yönetim sistemlerinin önemli bir bölümünün temelini oluşturan mimaridir. Bu mimarinin anlaşılması, veritabanı yönetim sistemlerine hakim olmayı kolaylaştırmaktadır.

İlişkisel veritabanı içindeki tablolar satır ve sütunlardan oluşur. Sütun, alan olarak da bilinir ve bir alana ait verileri içerir. Satır kayıt olarak da bilinir ve tabloda bulunan her bağımsız girişi içerir. Veri bağımsızlığı sayesinde, kayıtların fiziksel yerleşimi ya da tablodaki satırlar önemini kaybeder. Kullanıcıların veritabanı yapısını bilmeleri gerekmez[26].

Tez Çalışması Veritabanı Teknolojisi

Tez çalışmasında avantajlarını kısaca veri tekrarının önlenmesi, yüksek veritabanı hâkimiyeti ve veri sorgulama ve esneklik olarak sıralayabileceğimiz ilişkisel veritabanı yapısı kullanılmıştır. Tabloların oluşturulmasında veri nesnelere arası bağlantıların yapısı göz önünde bulundurulmuş olup 7 adım metodu kullanılarak

veritabanının etkin bir ilişkisel bir yapıya kavuşması sağlanmıştır. Tabloların oluşturulmasına ilişkin çalışma ilerleyen bölümde verilmektedir.

Normalizasyon ve 7 Adım Metodu

Normalizasyon bir ilişkisel veritabanında verilere erişim ve depolamanın verimli olarak gerçekleştirilebileceği dizaynı gerçekleştirilebileceği dizaynı sağlamaya yönelik izlenen adımlardır. Bu adımlar veri tekrarını azaltır ve tutarsızlıkları önler. İlişkisel veritabanında bir tablo bazı kısıtları karşılıyorsa normal formdadır. Codd'un orijinal çalışması bu tür üç normal formu tanımlamıştır fakat günümüzde kabul edilen 5 veya 6 adım bulunmaktadır. İlk adım birinci norm kısmında normalizasyon için al form (1NF), ikinci adım ikinci normal norm (2NF) v.b. şeklinde isimlendirilir.

Bu kısımda normalizasyon için form kademeleri kullanılmayacak olup birim ilişkileri göz önünde bulundurularak 7 adımdan oluşan ve hassas seviye normal forma denk gelebilecek (4NF) bir metot uygulanacaktır. Metot hem birim ilişki diyagramlarını oluşturmakta, hemde normalize edilmiş kayıt tiplerinin kümesini oluşturmaktadır. Gerçekleştirilen adımlar aşağıda verilmektedir.

Adım 1. Veri elemanlarını 3 sütuna ayır.

Sütun 1 de sadece belirleyiciler

Sütun 2 de sütun 1 deki belirleyiciler tarafından belirlenen veri elemanları

Sütun 3 de ise 1 yada 2 ye ait olamayan veri elemanları bulunacaktır.

Adım 2. Sütun 3 ü düzenle.

- a) İki yada daha fazla belirleyici tarafından belirlenen veri elemanlarını tanımla
- b) Eğer bir veri elemanı, herhangi bir veri eleman kombinasyonu tarafından belirlenemiyorsa, onu belirleyici olarak dikkate al ve birinci sütuna yerleştir.

Adım 3. Veri birimlerini isimlendir.

Adım 4. Veri birimleri arasındaki ilişkileri diyagrama aktar.

Adım 5. Diyagramı birim ilişki diyagramına dönüştür.

Adım 6. Normalize olmamış kayıt tiplerini oluştur.

Adım 7. 4 kural uygulayarak kayıt tiplerini normalize et.

7. Adımda kullanılan 4 kural şu şekildedir:

Kural 1: İki birim arasındaki ilişki (1:1) bire bir ise, bir veri biriminin kayıt tipinin birinci anahtarı, diğer birimin kayıt tipi içerisine yerleştirilmelidir.

Kural 2: İki birim arasındaki ilişki (1:M) bire çok ya da çoka bir (M:1) ise, 1 tarafında olan veri biriminin kayıt tipinin birincil anahtarı M tarafında olan veri biriminin kayıt tipi içerisine yerleştirilmelidir.

Kural 3: İki birim arasındaki ilişki çoka çok (M:M) ise, kayıt tipleri zaten normalize haldedir. İlave olarak iki veri biriminin birincil anahtarlarından oluşan yeni bir kayıt tipi oluşturulmalıdır. Bu iki belirleyici tarafından belirlenen herhangi bir veri elemanları da, bu yeni kayıt tipi içerisine yerleştirilecektir.

Kural 4: Sadece bir veri elemanı içeren herhangi bir kayıt tipi atılmalıdır. Bu veri elemanı önceki kurallardan birisiyle bir kayıt tipi içerisinde zaten yerleştirilmiş durumdadır[27].

Normalize edilmiş veritabanı yapısının bir bölümü örnek olarak aşağıda verilmektedir.

basvuru

Field Type Extra

basvuruid NUMBER

firmaid NUMBER

firmayetkili VARCHAR2(50 BYTE) Allow Null

onaydurum NUMBER

onaypersonel NUMBER
onayamir NUMBER
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null
silmedurumu NUMBER Allow Null
basvurutarih DATE
onaytarih TIMESTAMP(6) Allow Null

basvurubelge

Field Type Extra
basvurubelgeid NUMBER
basvuruid NUMBER
belgeid NUMBER
firmayetkili VARCHAR2(50 BYTE) Allow Null
kabuldurum NUMBER
onaypersonel NUMBER
belgelink VARCHAR2(100 BYTE) Allow Null
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null
silmedurumu NUMBER Allow Null

belgeturler

Field Type Extra
belgeturid NUMBER
belgeturadi VARCHAR2(250 BYTE) Allow Null
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null
silmedurumu NUMBER Allow Null

firmabelgeler

Field Type Extra

firmabelgeid NUMBER

firmaid NUMBER

belgeturid NUMBER

firmayetkili VARCHAR2(50 BYTE) Allow Null

belgelink VARCHAR2(100 BYTE) Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP(6) Allow Null

silmedurumu NUMBER Allow Null

uploadkisitc CHAR(11 BYTE) Allow Null

belgegecerlilikdurumu NUMBER(1)

belgeonayfirmayetkili NUMBER(11) Allow Null

belgekabuldurum NUMBER(11)

belgekabulyetkili NUMBER(11)

firmaceza

Field Type Extra

firmacezaid NUMBER

firmaid NUMBER

cezanneden VARCHAR2(200 BYTE) Allow Null

sonuc VARCHAR2(200 BYTE) Allow Null

aciklama VARCHAR2(200 BYTE) Allow Null

cezapersonel NUMBER

baslamazaman TIMESTAMP(6) Allow Null

bitiszaman TIMESTAMP(6) Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP(6) Allow Null

silmedurumu NUMBER Allow Null

firmalar

Field Type Extra

firmaid NUMBER Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP Allow Null

silmedurumu NUMBER Allow Null

sgkno VARCHAR2(26 BYTE) Allow Null

verginno VARCHAR2(15 BYTE) Allow Null

firmaadres VARCHAR2(300 BYTE) Allow Null

firimatelefon VARCHAR2(15 BYTE) Allow Null

firmafaks VARCHAR2(15 BYTE) Allow Null

firmaeposta VARCHAR2(40 BYTE) Allow Null

firmailece NUMBER(5) Allow Null

firmaile NUMBER(2) Allow Null

firmayetkilic CHAR(11 BYTE) Allow Null

firmayetkiliadi VARCHAR2(100 BYTE) Allow Null

firmayetkilisoyadi VARCHAR2(100 BYTE) Allow Null

firmaadi VARCHAR2(400 BYTE) Allow Null

grupyetki

Field Type Extra

yetkigrupid NUMBER

grupid NUMBER

yetkiid NUMBER

logislem VARCHAR2(10 BYTE)

loguser VARCHAR2(20 BYTE)

logzaman TIMESTAMP

silmedurumu NUMBER

gumrukislem

Field Type Extra

gumrukislemid NUMBER

kontrolbelgeid NUMBER

tarih DATE Allow Null
cekilenmiktar FLOAT(126) Allow Null
kalanmiktar FLOAT(126) Allow Null
kurumyetkili NUMBER(11)
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null

islemasama

Field Type Extra
islemasamaid NUMBER
basvuruid NUMBER
belgeid NUMBER
islemasamabilgiid NUMBER
tamamlanmadurum NUMBER
islempersonel NUMBER
baslamazaman TIMESTAMP(6) Allow Null
bitiszaman TIMESTAMP(6) Allow Null
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null
silmedurumu NUMBER Allow Null

islemasamabilgi

Field Type Extra
islemasamabilgiid NUMBER
islemasamabilgiadi VARCHAR2(300 BYTE) Allow Null
logislem VARCHAR2(10 BYTE) Allow Null
loguser VARCHAR2(20 BYTE) Allow Null
logzaman TIMESTAMP(6) Allow Null
silmedurumu NUMBER Allow Null

kontroldokuman

Field Type Extra

P dokumanid NUMBER

belgelink VARCHAR2(6 BYTE)

belgeaciklama VARCHAR2(200 BYTE) Allow Null

silmedurumu NUMBER

logislem VARCHAR2(10 BYTE)

loguser VARCHAR2(20 BYTE)

logzaman TIMESTAMP Allow Null

kontrolbelgeid NUMBER

kullaniciilar

Field Type Extra

P kullaniciid NUMBER

kullaniciturid NUMBER Allow Null

kullaniciadi VARCHAR2(20 BYTE) Allow Null

sifre VARCHAR2(15 BYTE) Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP(6) Allow Null

silmedurumu NUMBER Allow Null

kullanicigrup NUMBER Allow Null

kullaniciyetki

Field Type Extra

kullaniciyetkiid NUMBER

kullaniciid NUMBER

yetkiid NUMBER

logislem VARCHAR2(10 BYTE)

loguser VARCHAR2(20 BYTE)

logzaman TIMESTAMP

silmedurumu NUMBER

Lkullanicilar

Field Type Extra

P Lkullaniciid NUMBER

Lkullaniciturid NUMBER

Lkullaniciadi VARCHAR2(20 BYTE)

Lsifre VARCHAR2(15 BYTE)

Llogislem VARCHAR2(10 BYTE) Allow Null

Lloguser VARCHAR2(20 BYTE) Allow Null

Llogzaman TIMESTAMP

Lsilmedurumu NUMBER

Lkullanicigrup NUMBER

Lkullaniciyetki

Field Type Extra

Lkullaniciyetkiid NUMBER

Lkullaniciid NUMBER

Lyetkiid NUMBER

Llogislem VARCHAR2(10 BYTE)

Lloguser VARCHAR2(20 BYTE)

Llogzaman TIMESTAMP

Lsilmedurumu NUMBER

Lyetkiislemler

Field Type Extra

Lyetkiislemid NUMBER Allow Null

Llogislem VARCHAR2(10 BYTE) Allow Null

Lloguser VARCHAR2(20 BYTE) Allow Null

Llogzaman TIMESTAMP Allow Null

Lsilmedurumu NUMBER Allow Null

Lislemadi VARCHAR2(300 BYTE) Allow Null

musteriler

Field Type Extra

musteriid NUMBER Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP Allow Null

silmedurumu NUMBER Allow Null

sgkno VARCHAR2(26 BYTE) Allow Null

verginno VARCHAR2(15 BYTE) Allow Null

musteriadres VARCHAR2(300 BYTE) Allow Null

musteritelefon VARCHAR2(15 BYTE) Allow Null

musterifaks VARCHAR2(15 BYTE) Allow Null

musterieposta VARCHAR2(40 BYTE) Allow Null

musteriilce NUMBER(5) Allow Null

musteriil NUMBER(2) Allow Null

naceler

Field Type Extra

P naceid NUMBER

nacekodu VARCHAR2(6 BYTE)

naceaciklama VARCHAR2(200 BYTE) Allow Null

silmedurumu NUMBER

logislem VARCHAR2(10 BYTE)

loguser VARCHAR2(20 BYTE)

logzaman TIMESTAMP Allow Null

gecerlilikbaslangic DATE Allow Null

gecerlilikbitis DATE Allow Null

sessions

Field Type Extra

kullaniciid NUMBER

sessionid VARCHAR2(32 BYTE) Allow Null

songiriszamani VARCHAR2(32 BYTE) Allow Null

ip VARCHAR2(15 BYTE) Allow Null

yetkiislemeler

Field Type Extra

yetkiislemid NUMBER Allow Null

logislem VARCHAR2(10 BYTE) Allow Null

loguser VARCHAR2(20 BYTE) Allow Null

logzaman TIMESTAMP Allow Null

silmedurumu NUMBER Allow Null

islemadi VARCHAR2(300 BYTE) Allow Null

Sistem Güvenliđi ve Kullanıcı Doğrulama

E-Devlet Kapısı

E-devlet, kamu hizmetlerinin, tarafların doğrudan bağlantı kurmaya ya da fiziksel deđiş tokuş yapmaya gerek kalmaksızın açık veya sınırlı kullanıcıların ulaşabildiđi ağlar üzerinden elektronik olarak yapılabilmesini ifade etmektedir. Özetle e-devlet, bilgi ve iletişim teknolojilerinin kamu aktivitelerini desteklemek amacıyla kullanılmasıdır[28].

Kamu hizmetlerinin elektronik formatta sunulmasındaki amaç, zaman ve maliyetin azalması, hizmet kalitesinin geliştirilmesi amaçlanmaktadır. Bu kapsamda Türkiye’de 2003 yılından itibaren önemli yatırımlar ve çalışmalar gerçekleştirilmiş olup birçok devlet kurumu hizmetlerini e-devlet kapısı üzerinden sunmaya başlamıştır.

e-Devletin genel olarak üç önemli amacı vardır. Bunlar; kamusal hizmetleri yaygın ve erişilebilir hale getirmek, hizmet üretim ve yönetim sürecinde vatandaşların istek ve eğilimlerini daha etkin değerlendirmek ve katılımcı yurttaşlığın önünü açmak, devlet kurumlarının daha rasyonel ve verimli islemesini sağlamaktır. Bununla birlikte kamu yönetiminde şeffaflık, bürokrasinin azalması, hızın artırılması, üretkenlik ve verimlilik, kesintisiz hizmet, vatandaşın yönetime etkin katılımı, herkese eşit hizmet, etkin ve hızlı denetim e-Devletin temel amaçları arasında sayılabilir[29].

Elektronik İmza Kütüphaneleri ve Mobil İmza

İmza, bir kimsenin, bir yazının altına bu yazıyı yazdığını veya onayladığını belirtmek için her zaman aynı biçimde yazdığı ad veya işarettir[30]. Elektronik imza ise, 5070 sayılı Elektronik İmza Kanunun'da tanımlandığı şekliyle; başka bir elektronik veriye eklenen veya elektronik veriyle mantıksal bağlantısı bulunan ve kimlik doğrulama amacıyla kullanılan elektronik veriyi ifade eder[31]. Elektronik İmza Kanunu'nda öngörüldüğü üzere, elektronik imza elle atılan imza ile aynı hukuki sonuçları doğuran, imzanın dijital ortamda kullanılan halidir. Elle atılan imzanın değişmeyen biçiminin aksine, elektronik imza verisi değişken bir değerdir. Elektronik imza; imzalanacak olan verinin, imzayı atacak şahsa ait özel anahtar ile birlikte birtakım matematiksel işlemlerden geçirilmesi sonucunda oluşturulan ve imzalanacak veriye eklenen sayısal bir veridir.

Elektronik imza oluşturma işlemi için tanımlanmış söz dizimi, Kriptografik Mesaj Söz Dizimi (Cryptographic Message Syntax - CMS)'dir. CMS; imza oluşturma yanısıra, özetleme, doğrulama ve isteğe bağlı mesaj içeriğinin şifrelenmesi için de kullanılır.

CMS, veri korunması için bir sarmalama (encapsulation) söz dizimi tanımlar. Bu söz dizimi; elektronik imzaları, şifrelemeyi ve çoklu sarmalama işlemi de destekler. Önceden sarmalanmış bir veriye, başka bir şahıs tarafından da elektronik imza eklenebilir. Bunun yanısıra, elektronik imzaya eklenecek ve içerik ile birlikte imzalanacak olan imza tarihi gibi bazı eklentiler de CMS tarafından tanımlanmıştır. Seri imza gibi, imza değeri oluşturulduktan sonra imza ile ilişkilendirilecek olan diğer eklentiler de bu söz diziminde bulunmaktadır. CMS'e göre herhangi tipteki içeriğe sahip olan imzalı veri, içinde sıfır veya daha fazla imza değeri barındırabilir. İmzacılar, farklı tipteki içerikleri paralel olarak birçok kere imzalayabilirler[32].

İmzanın geçerlilik kontrol girdisi, imzadaki içerik özeti ve imzacının açık anahtarıdır. İmzanın doğrulanması için imzacının açık anahtarı, imzalı veride bulunan "sertifikalar verisi" içerisinde bulunmalıdır. Ayrıca, imza doğrulayıcı tarafından hesaplanan içerik özeti ile imzada bulunan içerik özetinin uyumlu olması gerekmektedir. İçerik özeti hesaplanırken kriptografik algoritmalar kullanılır [33].

Elektronik imza Hukuki Altyapısı

Hukuki Altyapı, zamanla gelişen yeni teknolojilerin, tüm e-Devlet uygulamalarında kullanılabilmesidir. Bu nedenle uygulama altyapılarının ve uygulama sonucunda ortaya çıkacak ürünlerin yasal olarak geçerliliğini sağlayacak hukuksal altyapının hazır olması gerekir. e-Devlet uygulamalarında e-İmza5070 sayılı yasayla güvence altına alınmış olmasına rağmen uygulamada henüz yaygınlaşmamıştır. 5070 sayılı Kanuna göre elektronik imza “başka bir elektronik veriye eklenen veya elektronik veriyle mantıksal bağlantısı bulunan ve kimlik doğrulama amacıyla kullanılan elektronik veridir.” Elektronik imza, gerçek hayatta ıslak imza ile yapılan her isin (kanunla hariç tutulanlar dışında: evlenme, tapu işlemleri gibi) elektronik ortamda yapılabilmesi için bir araçtır.

Elektronik imza da tıpkı ıslak imzada olduğu gibi evrakla kişinin kimliği arasında bir ilişki oluşturur. Ancak elektronik imzanın ıslak imzaya oranla üstünlükleri de vardır. Islak imzaya oranla taklit edilmeleri imkânsız denecek oranda zordur, ayrıca sadece kişinin kimlik bilgisiyle değil ayrıca mesajın içeriğiyle de ilişkilidir. Mesajdaki en küçük değişikliklerde bile elektronik imza değişir. Bu nedenle mesajın bütünlüğünü (bir yerden bir yere giderken bozulmadan, değiştirilmeden, orijinal haliyle gitmesi) sağlamada oldukça önemli bir rol oynar.

Elektronik imza ıslak imza gibi görsel bir ifade değildir. Elektronik ortamdaki hemen hemen her uygulamada mesajın sonuna eklenen bir veridir. Elektronik imza sayesinde kimlik doğrulama, veri bütünlüğü ve inkâr edememe özellikleri sağlanır. Ancak elektronik imza verinin gizliliğine ilişkin bir çözüm yaratmaz. Mesajların başkaları tarafından görülmemesi, başkalarından gizlenmesi isteniyorsa şifreleme tekniğinin kullanılması gereklidir. Elektronik uygulamalarda hem gizlilik hem de yukarıda sayılan üç özelliğin sağlanması için imzalama ve şifreleme teknikleri birlikte kullanılırlar.

e-Devlet modelinde kişisel verilerin/bilgilerin korunması hakkında hukuki altyapı vardır. Vatandaşların kamuya ait bilgilere erişim hakkının ve ulusal güvenlikle ilgili hassas bilgilerin uluslararası standartlarda ve çağdaş dünyanın gereklerine uygun olarak sınıflandırılması ve korunması için gerekli hukuki altyapı olmalıdır[29].

E-devlet ve Elektronik İmzanın Tasarlanan Sistemde Kullanımı

Yapılan çalışmada kurgulanan e-devlet kapısı üzerinden erişim yöntemi daha büyük kitlelere erişebilmek, mevcut güvenlik altyapısını kullanabilmek, kullanıcı tanımlamaları ile erişim yetkilendirmelerini e-devlet kapısı üzerinden yönetebilmek ve Bakanlık uygulamalarının tek çatı altında ulaşılabilir olmasını sağlamak amaçlı olarak kullanılmıştır.

Sistem içerisine elektronik imza ve mobil imzanın entegrasyonu ile kullanıcıların gerçekleştireceği işlemler yasal geçerlilik kazanacaktır. Tüm işlemlerin e-imza ile online olarak gerçekleştirilmesi bürokrasiyi azaltacak, zaman kayıplarının ve olası sahteciliğin önüne geçecektir.

BULGULAR

KONTROL BELGESİ İŞLEMLERİ SİSTEM ANALİZİ

İş Akışları Analizi

İş Sağlığı ve Güvenliği Enstitü Müdürlüğünde gerçekleştirilen görüşmeler sonucunda Kontrol Belgesi işlemlerinin

- İthalatçı firma tarafından başvuru evraklarının oluşturulması
- Evrakların sisteme aktarılması ve havale işlemleri
- Evrakların ön incelemesi
- Dokümanların teknik incelemesi
- Cevabi yazı veya kontrol belgesinin hazırlanması ve kargolanması işlemleri aşamalarından oluştuğu tespit edilmiştir.

Sosyal Güvenlik Kurumu ile yapılan görüşmeye göre şirket yetkililerinin başvuruları gerçekleştirmelerinde sisteme girişleri Sosyal Güvenlik Kurumu veritabanlarından alınan yetki onayı ile gerçekleştirilebilecektir.

Gümrük ve Ticaret Bakanlığı ile yapılan görüşmelerde ilgili Bakanlıkça Diğer kurum ve kuruluşlarla entegre olarak çalışacak kurumlar arası bilgi ve belge alışverişini sağlayacak bir yazılım sisteminin oluşturulmakta olduğu bilgisi alınmıştır. Entegrasyonun sağlanabilmesi amacıyla SGK sicil numaralarının yanısıra firmalara

ait vergi numaralarının da birincil tekil anahtar olarak sisteme eklenmesi kararlaştırılmıştır.

İthalatçı firma tarafından gerçekleştirilecek tehlikeli kimyasal ithalatının gümrük işlemlerinin bir parçası olarak kontrol belgesi alınması zorunluluğu bulunmaktadır. Bu amaçla firma tarafından firma yetkilisinin imzası ve firma kaşesi ile onaylanmış fatura sureti, faturadaki her ayrı kalem ürün için ayrı bir kontrol belgesi, aslı ile birlikte yeminli çevirmene yaptırılmış Faturanın Türkçe çevirisi (İngilizce olanlar hariç) , ürüne ait her sayfası ithalatçı firma tarafından kaşelenip imzalanmış Malzeme Güvenlik Bilgi Formunun en son versiyonu, yeminli çevirmene yaptırılmış Malzeme Güvenlik Bilgi Formunun Türkçe çevirisi (İngilizce olanlar hariç), Firmalar ithal edecekleri ürünü kendileri kullanacaksa Son Kullanıcı Beyannamesi, satış amaçlı ithal edeceklerse Taahhütname ve her yıl ilk müracaatta ve yetkili değişiminde şirketi temsile yetkili kişilerin imza sirküleri ile başvuru yapılır. Ayrıca Ürünün gümrükten çekilmesinden sonrasındaki bir ay içinde, hangi firmaya ne miktar ürün verildiği liste halinde Müdürlüğe bildirilir ve belgenin aslı bu liste ile birlikte Müdürlüğe iade edilir. Bildirimin ve belge iadesinin yapılmaması halinde firmanın sonraki başvuruları kabul edilmez.

Alınan başvuru evrakları taranarak tek bir tiff dosyası haline getirilir. e-calisma arayüzü üzerinden evrağın konusu, başvuru sahibi firma bilgileri gibi dosyalar eklenerek sisteme yüklenir. Evrak oluşturulan rota çerçevesinde ilgili amirlerin havalesi ile evrağı inceleyecek memurun ekranına düşer. Eğer evraklarda bir eksik veya hata bulunursa cevabi yazı hazırlanır ve aynı rota üzerinden giden evrak servisine kadar iletilir ve kargo yoluyla firmaya iletilerek eksik evrak bildiri tamamlanır. İSGÜM yetkilileri ile görüşmeler neticesinde belirlenen belge iade nedenleri aşağıda verilmektedir.

- Güvenlik Bilgi Formunun (MSDS) 1907/2006/EC sayılı REACH Tüzüğü'nün Güvenlik Bilgi Formlarına uygun olmaması
- Ürünün son kullanıcısı için Son Kullanıcı Beyannamesinin eksik olması
- Satış amaçlı getirilen ürünler için Taahhütnamenin eksik olması
- Güvenlik Bilgi Formlarının revizyon tarihinin 2008 yılı öncesi olması
- Güvenlik Bilgi Formlarında revizyon tarihi bulunmaması

- Ürünün kullanım yerinin eksik ifade edilmesi
- Ürünle ilgili alınan kontrol belgesi örneği veya analiz rapor örneğinin eksik olması
- İçeriği nedeniyle protokol gerektiren ürünler için Protokol Belgesinin bulunmaması
- Belirttiğiniz G.T.İ.P numarasının İthalat Tebliğinde bulunmaması
- Son kullanıcı beyannamesindeki G.T.İ.P. ve ürün adının, kontrol belgesinde belirtilenden farklı olması
- Kontrol belgesinde ürün miktarının kilogram veya litre birimi olarak belirtilmemesi
- Kontrol belgesinde firma kaşesi-yetkili isim-imza kısmında tarih belirtilmemesi
- Kontrol belgesinde firma kaşesi-yetkili isim-imza kısmındaki tarih ile fatura tarihi arasında 6 aylık sürenin aşılmış olması
- Kontrol belgesinde firma kaşesi-yetkili isim-imza kısmında isim ve soyadı bulunmaması
- Belgedeki mühür ve/veya imzaların belirgin olmaması
- Son kullanıcı beyannamesinde ürünü alan ve satan firmanın ismi, adresi; ürünün ismi, G.T.İ.P.'i, miktarı, kullanım amacı ile maddenin yeniden satılması durumunda veya bir başka müşteriye verilmesi durumunda malı alacak kişinin de aynı belgeyi dolduracağını ve söz konusu kontrole tabii kimyasal maddenin yasa dışı yollarda ve/veya uyuşturucu madde imalinde kullanılmayacağını dair taahhütün bulunmaması
- Taahhütnamede ürünü alan firmanın ismi ve adresi, ürünün ismi ile kontrole tabii kimyasalın yasal yollarla satışının yapılacağını ve satışı yapıldıktan sonra son kullanıcı/müşteri beyannamelerinin (listelerinin) İSGÜM Merkez Müdürlüğüne teslim edileceğine dair beyan olmaması
- Kullanılan G.T.İ.P numarasına göre gerekli teyit yazısının bulunmaması
- Kontrol belgesindeki ürün satış amaçlı getirilmişse kontrol belgesindeki kullanıcı firmadaki ibarenin Muhtelif Firmalar olarak belirtilmemesi

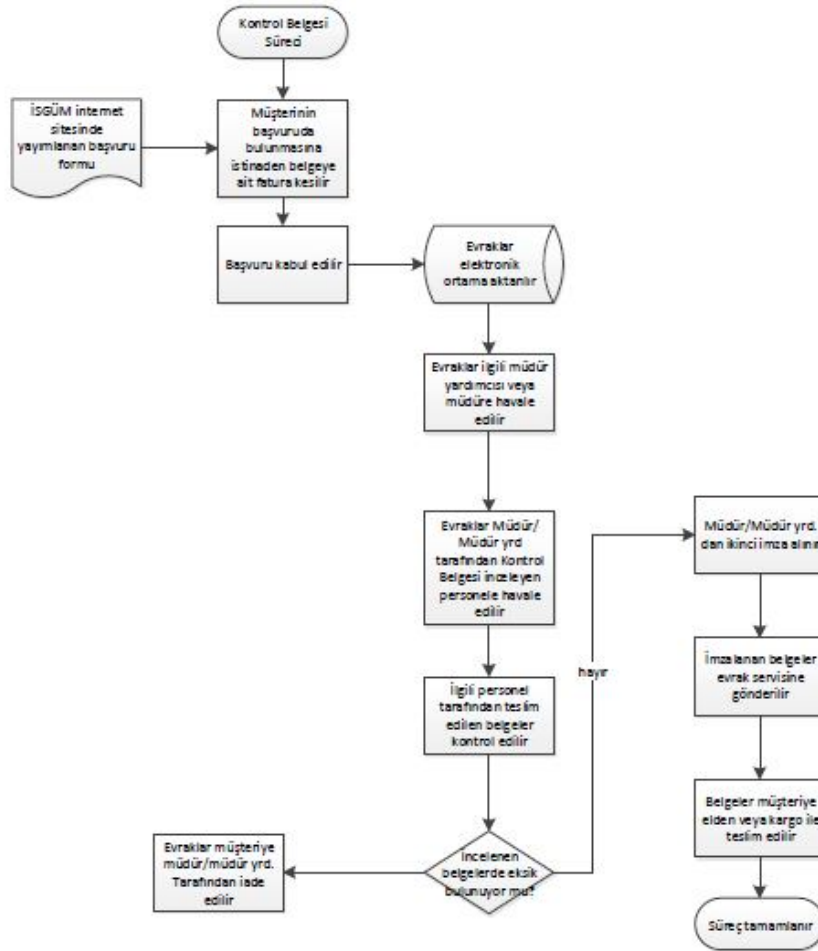
Evrakların şeklen ve mevzuat açısından incelenmesinden sonra ilgili firmaya ait dosya açılarak kontrol belgesi verilmesine mani bir durum olup olmadığı kontrol edilir. Eğer başvuru sahibi firmanın durumu uygunsa malzeme güvenlik formu üzerinden teknik incelemeye başlanır.

Teknik incelemede dikkat edilecek hususlar İSGÜM personeli ile yapılan görüşmelerde aşağıdaki şekilde belirtilmiştir:

- Yetkili kişinin isim, kaşe ve imzasının sadece kontrol belgesinde olması yeterlidir. Diğer sayfalarda ise imza ve kaşenin bulunması gerekmektedir.
- Ürünün nerede kullanılacağı kontrol belgesinde açıkça belirtilmelidir. Ya da ilave hazırlanan bir belgede ürünün kullanılacağı yer tanımlanmış olmalıdır.
- Faturada uluslararası geçerliliği olan birimler kullanılabilir ancak kontrol belgesinde miktar kg/Lt/Mt olarak yazılmalıdır.
- GTİP numarası yanlış ya da eksik yazılmış belgeler “2011/13 nolu tebliğde yer almamaktadır.” olarak ifade edilmeli, GTİP nosu 39 ve 40 ile başlayan ürünler ise katı halde bulunuyorsa “Ürün katı halde olduğu için 2011/13 nolu tebliğ kapsamı dışındadır” olarak ifade edilmelidir.
- Ürün sadece bir firmaya satılacaksa bile taahhütname veya alıcı firmadan son kullanıcı beyannamesi istenmelidir.
- Kontrol belgesinin alt kısmında tarih olmalıdır ve eski tarihler kabul edilmemelidir. (En fazla 6 ay) Ancak belgenin doldurulma tarihi fatura tarihinden önce ise kesinlikle kabul edilmemelidir.
- Malzeme Güvenlik Bilgi Formları (MSDS) Tehlikeli Maddeler Ve Müstahzarlara İlişkin Güvenlik Bilgi Formlarının Hazırlanması Ve Dağıtılması Hakkında Yönetmelik hükümlerine uygun olarak hazırlanmış olmalıdır. MSDS’ler 1907/2006 direktifinde belirtilen hususlara uygun ve üzerinde bu belirtilmiş ise TR ibaresine bakılmaz.
- Direktif dışında Türkçe olan MSDS’lerde TR ibaresine mutlaka bakılmalıdır.
- Her kimyasala ait R ve H fazları bulunmamaktadır. Bu MSDSlerde tehlike sınıflandırması yoktur veya tehlikesiz üründür ifadesinin olup olmadığına bakılmalıdır.
- MSDS’lerde bakılacak olan R kodları; 40, 45, 49, 68; H kodları; 304, 361, 350, 351, 360-D, 360-F’dir.
- Firmaların kendi yazdıkları son kullanıcı beyannameler ve taahhütleri, eğer bizim belirttiğimiz örneklerdeki tüm bilgileri içeriyorsa (bilgi sıralamasına bakılmaksızın) kabul edilmelidir.

- Protokol sadece kanserojen kimyasallar için istenmelidir. (R ve H kodu içeren kimyasallar)
- Acil onaylanması istenen belgeler konusunda aracı firma ile değil yetkili kişiler ile görüşme yapılabilir.
- Bilgisayarda İmza veya kaşesi görünmeyen belgeler iade edilecek, bu belgeler teslim edilmeden önce yetkili kişiye elden bakılacak kaşe ve imza var ise ilgili kişiye gönderilerek elden bakılıp onaylanmalıdır.

İnceleme sonucunun olumlu çıkması halinde onay yazısı e-çalışma evrak sistemi üzerinden hazırlanarak kargolanır veya firmayı temsile yetkili kişiye elde teslim edilir. Gerçekleştirilen standart işleme ilişkin akış şeması aşağıda verilmiştir. Gerekli görülmesi halinde inceleme aşamasında bölge müdürlüklerinden numune talep edilmektedir.



Şekil 2: Kontrol Belgesi İşlemleri İş Akış Şeması

SWOT Analizi

Proje hazırlığında, kontrol belgesi başvurusu yapan firmalar, işlemleri gerçekleştiren kamu personeli, bilgi teknolojileri alanında uzman kişiler, kontrol ve onay işlemlerini gerçekleştiren teknik personel ve kurum yöneticileri ile toplantılar ve görüşmeler gerçekleştirilmiş, bilgi ve görüşler birleştirilerek GZFT analizi yapılmıştır.

<u>Güçlü Yanlar:</u>	<u>Zayıf Yanlar:</u>
<ul style="list-style-type: none">○ Bakanlığın SGK ve diğer devlet kurumları ile yakın iş birliği içinde bulunması○ 2013/13 sayılı “İş Sağlığı ve Güvenliğini Etkileyen Bazı Kimyasalların İthalatına İlişkin Tebliğ” gereğince Kontrol Belgeleri işlemlerinin yürütülmesinde İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğü’nün yetkili kılınması○ Malzeme güvenlik bilgi formlarının standardizasyonunda temel alınacak○ SGK ile çalışan hastalık tanılarının sektörel olarak dağılımı konusunda çalışmaların halihazırda sürdürülüyor olması○ İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğü’nün (İSGÜM), İSG alanında ülkedeki en gelişmiş laboratuvar alt yapısına ve belgelerin teknik olarak incelenmesi için yeterli bilgi birikimine sahip olması○ Projenin tasarımında deneyimli bilişim personelinin çalışmış olması	<ul style="list-style-type: none">○ Kullanıcıların tespitine yönelik olarak kullanılacak SGK e-bildirge veritabanında sadece 4/a kapsamında yer alan işyerlerinin bulunması

Tablo 5: SWOT Analizi

<p><u>Tehditler:</u></p> <ul style="list-style-type: none"> ○ Malzeme güvenlik bilgi formlarının 2 ayrı dilde hazırlanması nedeniyle istatistiklerin üretilmesindeki güçlükler ○ İşyerlerinin elektronik imza sistemine geçişte yaşayacağı zorluklar 	<p><u>Fırsatlar:</u></p> <ul style="list-style-type: none"> ○ Barkodlama sistemi ve web servisler kullanılarak muhtemel sahteciliklerin önüne geçilmesi ○ Konu ile ilgili kişi ve kurumları içeren bir portföy oluşacağından bilgilendirme çalışmalarına olanak sağlanması
--	--

Tablo 5: SWOT Analizi (Devam)

İhtiyaç ve Süreç Analizi

İhtiyaç ve süreç analizinde; görüşmeler başlığı altında gerçekleştirilen toplantılar ve İSGÜM’de gerçekleştirilen iş akışlarının takibine yönelik çalışmalar göz önünde bulundurulmuştur. Yapılan analizde mevcut sistemde yer alan süreçlerin büyük iş gücü gerektiren ve uzun süren süreçler olduğu tespit edilmiştir.

Yıllık ortalama 22000 başvurunun alındığı kontrol belgesi işlemlerinin gerektirdiği iş yükünün ana nedeni olarak aynı ithalatçı tarafından verilen ve/veya aynı ürün için verilen dokümanların tekrar tekrar tüm süreçlerden geçirilmesi göze çarpmaktadır. İSGÜM yetkilileri ile gerçekleştirilen görüşmede her başvuruda ortalama 20 sayfa evrakın sisteme kaydedildiği ifade edilmiştir. Birim personelinden alınan bilgilere göre günlük kontrol belgesi başvuru sayısı 100 ila 200 arasında değişmekte kontrol işlemleri 20-25 kişi tarafından gerçekleştirilmektedir. Evrak kayıt işleminin 3 kişi tarafından gerçekleştirildiği, yaklaşık 3-5 dakika sürdüğü ve kontrol işleminin ise ortalama 5-7 dakika sürdüğü ifade edilmektedir.

Her başvurunun evrak kayıt işleminin tarama ve sisteme giriş işlemleri göz önüne alındığında 10 dakika sürdüğü farz edilirse işlemlerin getirdiği iş yükünün;

Yıllık 22000 başvuru*10 dakika=220.000 adam/dakika=3666 adam/saat = 458 iş günü olduğu hesaplanabilir.

Sürecin ikinci aşamasında evrak en üst amirden gerekli kontrol ve incelemelerin yapılacağı rota içerisindeki tüm personelden geçerek ulaşmaktadır. Bu aşama evrakların ilgili amir ve personellerde takılıp kalmasına ve işlem için beklemesine neden olmaktadır.

Evrakların mevzuatsal olarak uygunluğunun incelenmesi bir zaruret olmakla birlikte tekrar tekrar verilen evrakların incelenmesi veya daha önce verilip verilmediğinin kontrolü zaman kayıplarına neden olmaktadır.

Onaylanan ve reddedilen başvurular için ayrı yazılar hazırlanmaktadır. Hazırlanan yazılar çıktı alınarak ıslak imzaya gitmektedir. İmzalanan evraklar taranarak sisteme girişi yapılmaktadır. Bu süreçte büyük iş gücü ve zaman kaybına neden olmaktadır.

İşlemi tamamlanan evrakların arşivlenmesi hem fiziksel alan, hem iş gücü gereksinimi açısından önem taşınmaktadır. Şu ana kadar arşivlenmiş 5.000.000 sayfa evrakın yanısıra her yıl yaklaşık 400.000 sayfa evrak kontrol belgesi işlemleri dahilinde arşivlenmektedir.

Başvuru sahipleri açısından durum değerlendirildiğinde cevabi yazının gecikmesi ithali için izin alınan ürünün antrepolarda beklemesi nedeniyle maddi zararı, bekleme koşullarında muhafazası ile ilgili sorunları ve getirilen ürünlerin yurt içinde kullanımında gecikmeyi beraberinde getirmektedir.

Günümüze kadar yaklaşık 5.000.000 sayfa evrak işlem görmesine rağmen İSGÜM tarafından hangi yıllarda hangi kimyasalların ülkeye sokulduğu, ne kadar kimyasalın hangi sektörlerde ne amaçla kullanıldığına dair bir rapor bulunmamaktadır. Mevcut sistem içerisinde bu tür istatistiklerin üretilmesi pratik olarak mümkün değildir.

İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ TASARIMI

Yönetim Bilgi Sistemi iki bölümden oluşmakta olup veri girişleri ile başvuruların gerçekleştirildiği sistemin analizleri, ilişkisel veritabanı yapısı, iş akışları ve benzeri tez kapsamında oluşturulmuş olup (bakınız analiz çalışmaları) e-devlet kapısı üzerinde çalışacak başvuru programının kodlanması, test edilmesi ve kullanıma açılması ilgili birimlerden destek olacak bakanlık personelinin yanı sıra 4 kişilik bir teknik ekibin 8 aylık bir çalışmasını öngördüğünden ayrı bir çalışma olarak tutulmuştur. Ancak oluşturulan sistemin etkinliğinin izlenebilmesi amacıyla 100 başvuru kaydı alınarak yaklaşık 2500 sayfalık belgeler OCR sisteminden geçirilerek metin dosyası haline getirilmiş olup, gerekli bölümlerden excel dosyası oluşturularak veritabanına yüklenmiştir.

Online Başvuru ve Takip Sistemi

Firma Yetkilileri Grubu

Firma yetkilisi kendi elektronik imzası üzerinden e-devlet kapısında yer alan sisteme giriş yaptığında TC kimlik numarası halihazırda e-Bildirge kullanıcıları ve yetkili olduğu firmalara ilişkin web servis üzerinden kontrol edilecek, işlem yapılacak firma liste içerisinden seçilerek oturum bilgilerine atılır. Kişi tarafından ilk girişte oluşturulan şifre ile sisteme girişi sağlanır ve aşağıdaki modüller sistemde yer alır.

- Kontrol Belgesi Başvuru Modülü
- Başvuru Takip Modülü
- MGBF Dokümanları Modülü
- Protokol İşlemleri Modülü
- Belge Yönetimi Modülü
- Müşteri Portföyü Modülü
- Bilgilendirme ve Duyurular Modülü
- Kullanıcı İşlemleri Modülü

Sayısal Uygunluk ve Kalite Kontrol Modülü

Sistem tarafından ilgili firmanın önceki yerinde inceleme raporlarına istinaden başvuru hakkı olup olmadığı ve başvuru hakkı yoksa başvuru yapabileceği tarih kontrol edilir. Ticaret sicil gazetesi, imza sirküleri dokümanların sistemde olup olmadığı kontrol edilir.

Yeni başvuru için kontrol belgesi başvuru formu doldurulur. Sistem tarafından üretilen başvuru numarası firma tarafından ilgili bankaya ödeme yapılması için kullanıcıya bildirilir. Başvuru formuna da barkod olarak eklenir.

Protokol işlemleri kapsamında belge yönetim modülünden ticaret sicil gazetesi, imza sirküleri gibi dokümanlar seçilerek başvuruya eklenir. İlgili dosyaların yüklenmesinde direk olarak o protokole ait evraklar seçili gelmelidir.

Taahhütname ve müşteri/son kullanıcı beyannamesi dokümanısıstem tarafından çıktı alınabilecek şekilde oluşturulacak elektronik olarak imzalanabilecek ve/veya ıslak imzalı versiyonu sisteme yüklenebilecektir.

Firma yetkilisi mevcut MGBF dokümanları içerisinde veya yeni bir doküman oluşturarak ithale edilecek ürüne ait MGBF bilgisini sisteme ekler.

Başvuru sahibi Tüm evraklar tamamlandıktan ve ödeme bankaya yapıldıktan sonra başvuruya ait işlemin tamamlanması komutunu sisteme verir. Banka web servisi üzerinden ödeme teyiti alınır. Başvuru işlemi sırasında tüm dokümanların ıslak imzalı kopyalarının alınabilmesi veya firma tarafından çıktı olarak saklanabilmesi için çıktıya uygun korumalı pdf dosyaları oluşturulur. Başvuruda bir eksiklik varsa sistem tarafından kontrol edilerek başvuru tamamlama işlemi esnasında kullanıcıya bildirilir işlem tamamlanmaz.

MGBF Dokümanları Modülü

MGBF dokümanlarının sisteme formlar üzerinden girilebildiği ve form ile ilişkili olarak orijinal MGBF dokümanının sisteme upload edildiği (yüklediği) sistemdir.

MGBF dokümanındaki deęişiklikler revizyon mantığı çerçevesinde tutularak loglanmalı her güncelleme yeni bir revizyon kabul edilmelidir. Bu sayede her başvuruda kullanılan dokümanın başvuru esnasındaki haline ulaşılabilecektir. MGBF dokümanları ile ilgili olarak ekleme ve silme işlemleri tanımlanacak ancak silme işlemleri delete etme yerine flag atılarak gerçekleştirilmelidir.

MGBF dokümanlarında 16 alan bulunmaktadır. Bu alanlara ilgili yönetmelikten ulaşılabilir.

Yapı içerisinde yer alan alanlar:

- Başlık No 1
- Başlık No 2
- Başlık No 3
- Başlık No 9

Kontrol Belgesi Başvuru Takip Modülü

Modül içerisinde yer alan Başvuru arama ekranından, kimyasalın ticari adı, başvuru tarihi, sistem tarafından verilen ve ıslak imzalı dokümanlar ile bilgilendirme kopyalarına otomatik olarak yerleştirilen başvuru barkod numarası üzerinden başvurular aranır. Arama sonucunda listelenen başvurulardan seçilen kayıt için durum detay bilgileri gösterilir. Detay bilgisinde işlemin hangi aşamada olduğu belirtilir. Eğer belgeler incelenerek kontrol belgesi onaylanmış ise elektronik imzalı onay belgesi ve gerekli olduğu durumlarda hizmet teslim fişi çıktıya uygun olarak korumalı PDF olarak hazırlanır. Belgede sistem tarafından üretilen barkod bilgisi yer alır. Başvuruya ait eksik evrak veya düzeltme olması halinde ilgili evraklar veya güncellemeler firma yetkilisi tarafından gerçekleştirilebilir. Başvuru işlemleri yeniden başlatılır. Önceki başvurudaki incelemeler İSGÜM kullanıcılarına bilgi olarak ilgili ekranlarda gösterilir.

Müşteri Portföyü Modülü

Her kontrol belgesine ait olarak müşteri listesi hazırlanması gerekmektedir. Bu nedenle ithalatçı firma tarafından firma SGK sicil numarası sisteme girilir. Firmaya ait

kapasite kullanım raporu sisteme upload edilir ve tespit edilen alanlar bir web formu üzerinden sisteme aktarılır. SGK üzerinden sağlanan web servis kullanılarak ilgili firmanın detay bilgileri alınır.

Protokol İşlemleri Modülü

Protokol gerektiren ürünler için protokol belgesinin ticari ürün bazında oluşturulduğu ve karşılıklı olarak elektronik imza ile imzalandığı dokümandır. Ek dokümanlar olarak işçi listesi işçi sağlık raporları gibi dokümanlar belge yönetim sisteminden yüklenerek protokol ile eşleştirildiği modüldür.

Belge Yönetimi Modülü

İmza Sirküleri, Ticaret sicil gazetesi gibi evrakların sisteme yüklenerek saklandığı ve yüklemelerin loglandığı modüldür. Aynı belgenin farklı versiyonları geçerlilik tarih bilgisi ile sisteme aktarılabilir.

Bilgilendirme ve Duyurular Modülü

MGBF rehberi, İlgili mevzuat vb. ile duyurular ve zorunlu olarak belirlenen sayıda tekrarlanarak ana ekranda görüntülenecek sayfaları içerir. Sistemde ayrıca doldurulmadan diğer işlemlerin gerçekleştirilemeyeceği anket ve/veya formlar yer alır.

Kullanıcı İşlemleri Modülü

Şifre değiştirme, şifre hatırlatma, son işlemler logları ve iletişim bilgileri girişi güncellemesi bu modül üzerinden gerçekleştirilir.

İdare Yöneticileri Grubu

Elektronik imza üzerinden e-devlet kapısında yer alan sisteme giriş yapıldığında TC kimlik numarası sistemden alınarak ikinci bir giriş ekranı getirilir. Sistem aşağıdaki modülleri içermelidir. Her modül için sistem yöneticisince kullanılacak arayüz ile yetki verilmelidir.

- Başvuru Takip Modülü
- Firma Değerlendirme Modülü
- Kontrol Belgesi İstatistikleri Modülü
- İSG İstatistikleri Modülü
- Çalışma raporları Modülü
- Yerinde İnceleme Raporları Modülü
- Başvuru Dağıtım Modülü
- Bilgilendirme ve Duyurular Modülü (Giriş veya onay)
- Kullanıcı işlemleri Modülü
- Yetkilendirme Modülü
- Protokol İşlemleri Modülü

Başvuru Takip Modülü

İSGÜM yetkili kullanıcıları başvuru takip sistemi ile aynı işlevsel yapıya sahip olmakla birlikte detay bilgilerin başvuruya eklenebildiği bir sistemdir. Sistemde başvuru tarih ve saati, başvuru yapan firma ve yetkili kişi bilgileri, başvurunun hangi safhada olduğu başvuru ile ilgili herhangi bir inceleme varsa evrakların uygunluk durumu, işlemin hangi personel tarafından takip edildiği, inceleme kontrol listeleri, firmaya ait işlemleri devam eden başvuru sayısı, firmaya ait tamamlanan ve reddedilen başvuru sayısı başvuru arama işlemi sonucunda seçilen başvuru için görüntülenir.

Firma Değerlendirme Modülü

Başvuru sahibi firmanın daha önceki kontrol belgesi başvurularına ilişkin bilgiler istatistik olarak verilir. İstatistikte önceki başvurulara konu olan ithalat ürününün özelliklerine göre sınıflandırmalar, yıllara göre başvuru sayıları, başvurularındaki eksikliklerin türlerine göre oranları, teste giden kimyasalların uygunluk oranları gibi bilgiler yer alır. Bu sayede firmanın başvurularının ne kadar sorunlu olduğu, çalışma sıklıkları gibi veriler elde edilir.

İSG İstatistikleri Modülü

Tehlikeli kimyasalın MGBF dokümanında verilen özelliklerine göre İSG açısından yarattığı tehlikeleri ve sektörel kullanımlarının tespitini amaçlayan istatistikleri içerir. İhtiyaçlar dahilinde ilgili personel tarafından hazırlanan İstatistik raporları sistemce üretilebilir olmalıdır.

Başvuru Dağıtım Modülü

İSGÜM yöneticisine gelen başvuruların inceleme personeline yönlendirilmesinde kullanılır. İş yükü dağılımının etkin yapılabilmesi amacıyla her personel üzerinde bulunan başvuru sayısı incelemenin bulunduğu aşamaya göre tek bir liste halinde görüntülenir. Seçim listesinden inceleme personeli seçilir alt kısımda yer alan başvuru onay kutucukları seçilerek personele başvuru kontrol işlemi atanmış olur.

Yerinde İnceleme Raporları Modülü

Yerinde incelemede yer alan ve yetkili personelce doldurulan raporlar ve kontrol listeleri Yönetici tarafından sadece görüntülenebilir. Uygun görülmesi halinde elektronik imza ile imzalanır veya raporu hazırlayan kullanıcıya bilgi notu ile geri iletilir.

Kontrol Belgesi İstatistikleri Modülü

Kontrol Belgesi başvuruları ile ilgili raporlamaları içerir. Tüm raporlar PDF ve Office dokümanları şeklinde dışarı aktarılabilir.

- Başvuru yapılan iller bazında raporlar
- İthal ürünün giriş yaptığı gümrükler bazında raporlar
- Başvuruların aylara göre dağılımı raporu
- Başvuruların reddedilme nedenlerine göre raporlar

- Bařvuruların sektörel dağılımı
- Dinamik raporlama ve istatistik

Çalıřma raporları Modülü

Bu modülde ařağıdaki fonksiyonlar bulunacaktır:

- Personelin aylara göre iřlem sayıları,
- Personel bazında kabul ve reddedilen belge sayıları,
- Personellerin ortalama belge inceleme süreleri dağılımı,
- Personelin sistemde kalma süreleri,
- Personeller üzerinde bekleyen iřlemler.

Bilgilendirme ve Duyurular Modülü

Duyuru, mevzuat ve formların sistemde oluşturulduğı formların zorunluluk ve periyodik tekrar kriterlerinin düzenlendiğı modüldür.

Kullanıcı iřlemleri Modülü

řifre deęiřtirme, řifre hatırlatma, son iřlemler logları süreli kullanıcı yetki devri bu modül üzerinden gerekleřtirilir.

Protokol iřlemleri Modülü

Protokol gerektiren ürünler için protokol belgesinin ticari ürün bazında oluşturulduğı ve karřılıklı olarak elektronik imza ile imzalandığı dokümandır. Kullanıcılar firma deęerlendirme modülünü inceleyerek protokolü elektronik imza ile imzalayabilir.

Yetkilendirme Modülü

Sistem yöneticisi tarafından girilen İSGÜM kullanıcıları ve yöneticileri için her bir modül ve işlem bazında ekleme güncelleme silme ve görüntüleme yetkileri kişinin adı seçildiğinde çıkan kontrol listeleri üzerinden sistemde güncellenir.

İSGÜM Yetkili Kullanıcıları

Kullanıcılar elektronik imza ile e-devlet kapısında yer alan sisteme giriş yaptığında TC kimlik numarası sistemden alınarak ikinci bir login ekranı getirilir girilen kullanıcı adı ve şifre kullanıcılar tablosundan kontrol edilerek oturum açılır. Sistem aşağıdaki modülleri içermelidir. Her modül için sistem yöneticisince kullanılacak arayüz ile yetki verilmelidir.

- Başvuru İnceleme Modülü
- Başvuru İnceleme Devri Modülü
- Başvuru Takip Modülü
- Firma Değerlendirme Modülü
- Yerinde İnceleme Raporları Modülü
- Bilgilendirme ve Duyurular Modülü (Giriş)
- İSG İstatistikleri Modülü
- Kullanıcı işlemleri Modülü

Başvuru İnceleme Modülü

Fatura İşlemleri

Bankadan teyit alınan başvuruların faturalarının oluşturulması için yetkili İSGÜM kullanıcısı tarafından kullanılmak üzere fatura oluşturma ekranı hazırlanır. Bu ekranda firma ve başvuru bilgileri başvuru barkod numarası üzerinden alınarak ekrana getirilir. Kabul edilen ödemeler iyi onay kutucukları işaretlenerek elektronik olarak imzalanır. Firma kullanıcısının çıktı alabileceği şekilde sistemde depolanır.

Belge Kabul ve Kontrol İşlemleri

Sistem tarafından ön kontrolleri (eksik evraklar, ödemeler, GTİP numaraları kontrolü vb.) başvuru esnasında başvuru başvurularının ilgili inceleme personeline dağıtım yetkisi olan personellerin yeni başvuru kabul ekranına düşer. Yönetici inceleme personelini seçim kutusundan seçer ve yönlendireceği başvuruları işaretler. Onay verilmesi ile iş dağıtımı tamamlanmış olur.

Kullanıcı Başvuru İnceleme Kontrol Listelerini kullanarak incelemeyi yapar, eğer ret veya düzeltme gerektiren bir durum olursa Uygunsuzluk bildir işlemi ile belgeyi reddeder. Parametrik açıklamaların yanısıra gerekli görülürse başvuruya özel açıklama freetext olarak sisteme girilir.

Eğer başvuru numune alımını gerektiriyorsa (kanserojen madde içeren kimyasallar, tolien, ksilen vb. içeren kimyasallar) Numune alımı için uygun bölge müdürlüğü seçilerek bekleyen numune alımları tablosuna kayıt atılır. Ayrıca Bölge müdürlüğündeki yetkili personele işlem kaydı e-posta yoluyla haber verilir. Bölge Müdürlüğünce gerçekleştirilecek işlem detayları numune işlemleri modülünde anlatılmaktadır. Test sonuçları başvuru inceleyen personel tarafından sisteme aktarılır.

Tüm kontroller gerçekleştirildikten sonra incelemeyi gerçekleştiren personel ön onayı elektronik imzasını kullanarak gerçekleştirir. Ön onaydan geçen belge onay için amirin onay bekleyen başvurular ekranına düşer elektronik imza ile imzalanan dokümana ait işlem tamamlanmış olur. Onay ekranlarında başvuru evrakları, inceleme kontrol listeleri ve bilgi notları görüntülenebilmelidir.

Firma için çıktı alabileceği elektronik imzalı doküman pdf formatında yanısıra işlem başvuru sonuç paylaşım web servisi ile de yetki verilen kurum ve kuruluşlarla da paylaşılır.

Başvuru Takip Modülü

İSGÜM yönetici başvuru modülü ile aynı olup sadece önceliklendirme ve notlar bölümü bu türdeki kullanıcılardan sadece belgeyi inceleyen kişi tarafından görülür.

Başvuru İnceleme Devri Modülü

İnceleme yapan personel herhangi bir nedenle incelenen başvuruyu bir başka personele devrederse bu işlem devir nedeni ile sistemde loglanır. Daha sonra gerçekleştirilen tüm işlem logları yeni kullanıcıya atanır. Başvuru işlemleri rutin şekilde devam ettirilir.

Firma Değerlendirme Modülü

İSGÜM yöneticileri modülünden farklı olarak sadece başvuru eksiklikleri istatistikleri görüntülenir.

Yerinde İnceleme Modülü

İSGÜM tarafından gerçekleştirilen yerinde inceleme çalışmalarında doldurulan formlar ile yerinde inceleme kontrol listelerini içerir. İncelemeye konu olan ürün ve kontrol belgesi başvuruları sisteme aktarılır. Görülen aksaklıklar ve varsa verilen uyarı/ceza sistemde parametrik olarak tutulur ve yeni başvuru kabullerinde değerlendirmeye alınır. İnceleme yapan personel, personelin çalışma süreleri sistemde kayıt altına alınır. Nihai rapor; ilgili personel ve amirler tarafından elektronik imza ile imza altına alınır. Firmanın Bilgilendirme ve duyurular ekranında yayınlanır.

Bilgilendirme ve Duyurular Modülü

Duyuru, mevzuat ve formların sistemde oluşturulduğu formların zorunluluk ve periyodik tekrar kriterlerinin düzenlendiği modüldür.

İSG İstatistikleri Modülü

Tehlikeli kimyasalın MGBF dokümanında verilen özelliklerine göre İSG açısından yarattığı tehlikeleri ve sektörel kullanımlarının tespitini amaçlayan istatistikleri içerir. Kurum tarafından hazırlanacak İstatistik raporları sistemce üretilir hale gelecektir.

Kullanıcı İşlemleri Modülü

Şifre değiştirme, şifre hatırlatma, son işlemler logları süreli kullanıcı yetki devri bu modül üzerinden gerçekleştirilir.

İSGÜM Bölge Müdürlüğü Kullanıcıları

Kişi kendi elektronik imzası üzerinden e-devlet kapısında yer alan sisteme giriş yaptığında TC kimlik numarası sistemden alınarak ikinci bir giriş ekranı getirilir girilen kullanıcı adı ve şifre kullanıcılar tablosundan kontrol edilerek oturum açılır. Sistem aşağıdaki modülleri içermelidir. Her modül için sistem yöneticisince kullanılacak arayüz ile yetki verilmelidir.

- Numune İşlemleri Modülü
- Kullanıcı İşlemleri Modülü

Numune İşlemleri Modülü

İSGÜM yöneticileri veya yetki verilmesi halinde personel numune alımı gerektiği durumlarda ilgili bölgeyi seçerek numune alımı için gerekli bilgileri sisteme girer ve alım talebini tamamlar. Talep bölge müdürlüğünde görevlendirilmiş personelce alınır ve işlemin gerçekleştirilmesi için gereken tüm adımlar (görevlendirme, gümrükten numune alımı, kargolama vb.) işlem tarihi ile sisteme aktarılır. Gerekli evraklar ilgili personelce taranarak sisteme yüklenir.

Kullanıcı işlemleri Modülü

Şifre değiştirme, şifre hatırlatma, son işlemler logları süreli kullanıcı yetki devri bu modül üzerinden gerçekleştirilir.

İSGÜM Sistem Yöneticileri

Sistem yöneticisi yeni kullanıcıların tanımlanması ve parametre ekranlarının kontrolünden sorumludur. Kendi elektronik imzası üzerinden e-devlet kapısında yer alan sisteme giriş yaptığında TC kimlik numarası sistemden alınarak ikinci bir giriş ekranı getirilir girilen kullanıcı adı ve şifre sistem yöneticileri tablosundan kontrol edilerek oturum açılır. Sistem aşağıdaki modülleri içermelidir.

- Kontrol Listeleri Modülü
- Parametreler Modülü
- Personel Modülü
- Kullanıcı işlemleri Modülü

Kontrol Listeleri Modülü

Başvuru inceleme, yerinde inceleme gibi işlemlerde izlenecek adımları ve kontrolleri içeren formların hazırlandığı modüldür. Modül içerisinde eklenecek alanlar parametrik olarak eklenip çıkartılabilir. Ancak eski değerlendirmelerin bozulmasını önlemek amacıyla silinen kayıtlara flag atılır silinmez. Arayüz üzerinden formdaki yerleşim ve statik text alanları tanımlanabilmelidir.

Parametreler Modülü

İSGÜM kurumsal bilgileri ile inceleme uygunsuzluk parametreleri gibi seçmeli alanlar parametrik yapıda olmalıdır. Sistem yöneticisi tarafından ekleme, güncelleme ve silme işlemleri gerçekleştirilebilmelidir. Geriye dönük evraklarda o tarihte geçerli olan değerler kullanılmalıdır.

Personel Modülü

Sisteme kullanıcı tiplerine uygun olarak yeni personellerin eklenmesi veya ayrılan personellerin sistemden çıkarılmasını içeren modüldür. Yeni personel tipleri tanımlanabilmeli ve personel tipleri için default yetki yapıları oluşturulabilmelidir.

Kullanıcı İşlemleri Modülü

Şifre değiştirme, şifre hatırlatma, son işlemler logları süreli kullanıcı yetki devri bu modül üzerinden gerçekleştirilir.

Veri Paydaşları

Veri paylaşımları platform bağımsız olması amacıyla web servisler aracılığıyla sağlanır.

Sistem yöneticisi yeni kullanıcıların tanımlanması ve parametre ekranlarının kontrolünden sorumludur. Kendi elektronik imzası üzerinden e-devlet kapısında yer alan sisteme giriş yaptığında TC kimlik numarası sistemden alınarak ikinci bir giriş ekranı getirilir girilen kullanıcı adı ve şifre sistem yöneticileri tablosundan kontrol edilerek oturum açılır. Sistem aşağıdaki modülleri içermelidir.

- Başvuru Sonucu paylaşım Web Servisi (Sunucu)
- SGK İşyeri Sorgulama Web Servisi (İstemci)
- SGK e-Bildirge Kullanıcısı Doğrulama Web Servisi (İstemci)
- Ödemeler Web Servisi (İstemci)
- Fatura İşlemleri Web Servisi

Başvuru Sonucu paylaşım Web Servisi (Sunucu)

Kontrol belgesi başvuru sonuçları paylaşım web servisi server tarafında hazırlanacak kullanıcı adı, şifre, başvuru no, kurum id bilgilerini alarak sonuç olarak başvuru sonuç id, ve başvuru cevap dosyası iletilir. Tüm veri alışverişleri loglanır.

SGK İşyeri Sorgulama Web Servisi (İstemci)

SGK web servisi üzerinden 26 haneli SGK sicil numarası, kullanıcı adı, şifre gönderilerek işyeri ünvanı, ve diğer bilgiler string olarak getirilir.

SGK e-Bildirge Kullanıcısı Doğrulama Web Servisi (İstemci)

İlgili web servise TC kimlik numarası ve güvenlik parametreleri gönderilerek, yetkili olduğu işyerlerinin listesi alınır.

Ödemeler web servisi (İstemci)

Banka tarafından talep edilmesi halinde kamu kurumları için kendi kurum hesaplarına ilişkin hareketleri görebilecekleri web servisler oluşturulmaktadır. Bu kapsamda oluşturulacak web servisten, ilgili hesap için yatırılan miktar, tarih ve açıklama alanında kontrol belgesi başvuru numarası alınarak ödemeler online sisteme gerçek zamanlı olarak aktarılır.

Fatura İşlemleri Web Servisi

Faturalandırma aşamasına gelen başvuruların firma bilgileri ödeme miktarı açıklamalar gibi parametreler korumalı olarak web servis üzerinden fatura basım işlemleri için ilgili programa aktartılır.

e-İmza Kütüphanesi

Sistemde elektronik/mobil imza desteği olmalıdır. İlgili kanun, yönetmelik ve tebliğlerine uygun olarak elektronik imza oluşturulabilmeli, doğrulanabilmeli ve dokümanlar imzalanabilmelidir.

e – İmza Kütüphanesi J2EE alt yapısı ile geliştirilmiş olmalıdır.Nitelikli elektronik imza sertifikaları ve mobil imzalar temin edilmelidir.

Elektronik Sertifika Hizmet Sağlayıcıları (ESHS) ile çalışabilecek yapıda elektronik/mobil imza için zaman damgası alma fonksiyonu sağlanmalıdır. Zaman Damgası Hizmeti temin edilmelidir.

Teklif edilen sistem kullanıcıların sisteme ekledikleri belgeleri elektronik imza ile imzalamalarına ve imzalı belgeleri doğrulamalarına olanak sağlamalıdır. Elektronik İmzalama ve Doğrulama Modülü sistemin bir parçası olmalıdır

Cryptographic Message Syntax (CMS) ile ETSI TS 101-733/903 ve ETSI 102 Standartlarına uygun olmalıdır. CWA 14170/CWA 14171 standartlarını desteklemelidir. X.509 v3 Standardına uygun tüm sertifikalar ile çalışabilmelidir

Sistem kullanıcıların sisteme ekledikleri belgeleri elektronik imza ile imzalamalarına olanak sağlanacaktır. 5070 sayılı Elektronik İmza Kanunu'nda belirtilen şartları sağlamalı ve "Elektronik İmza ile ilgili Süreçlere ve Teknik Kriterlere İlişkin Tebliğ" ile uyumlu olmalıdır. Sağlanacak Elektronik imzalama/doğrulama yazılımı tarafından oluşturulan elektronik imzaların ETSI TS 101 733'de tarifli yapılan BES ve CAdES-T imza formatlarına uygun olduğu ve imza oluşturulurken ve doğrulanırken yapılan kontrollerin E-İmza Kontrol Listesi'nde belirtilen şartları sağladığı TÜBİTAK tarafından onaylanmış olmalıdır.

Elektronik imza ile imzalama yapılırken kullanıcının elektronik imzasının geçerliği kontrol edilecektir. Geçersiz imza ile imzalama işlemine izin verilmeyecektir.

İstenilen kök sertifika hiyerarşisi ile sertifika geçerlilik ve imza doğrulama ve bütünlük kontrolü yapılabilecektir. Yan yana (paralel) ve iç içe (seri) imza desteği olacaktır.

Elektronik imza ile imzalama yapılırken kullanıcının elektronik imzasının geçerliği kontrol edilecektir. Geçersiz imza ile imzalama işlemine izin verilmeyecektir.

Kurum dışında belgeye muhatap kurum, kuruluş ve vatandaşlar tarafından belge üzerindeki imzaların doğrulanması, imza hiyerarşisinin ve orijinal belgelerin

görülebilmesi amacıyla, açık lisanslı bir imza doğrulama ve belgeler görüntüleme uygulaması sağlanmalıdır.

XML, PDF, JPEG, TIFF, DOC, DOCX formatındaki belgelerde bulunan veriler imzalanabilecektir. Buna ek olarak her türlü binary formattaki belge veya veri imzalanabilecek/doğrulanabilecek ve orijinal belge veya veri elde edilebilecektir.

İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ, KONTROL BELGESİ DEĞERLENDİRME SİSTEMİ

Kontrol belgesi işlemleri sırasında edinilen verilerin değerlendirilmesine ilişkin bir sistem tasarlanmış ve kodu yazılarak test ortamında uygulamaya konulmuş ve sistem kullanılarak örnek veriler üzerinde raporlar hazırlanmıştır.

Sistem tarafından üretilen raporlar; ithal edilen ürünün özellikleri ve İSG riskleri, ithalatçı firma, ithal edilen kimyasalların bölgesel dağılımları, ithal edilen ürünü kullanan firmaların faaliyet alanı ve büyüklüklerine göre dağılımı, ilgili firmalardaki iş kazası ve meslek hastalıkları bilgileri gibi raporları içermektedir. Sistemin ayrıca akademik araştırmalara bilgi sağlamak amaçlı dış kullanıcılara açık raporlar bölümü yer alır. Bu bölümde yetkilendirme kullanıcı tiplerine göre gerçekleştirilir.

Karar destek sistemi raporlamaları ile ilgili olarak ürün takip, ithalat takip ve personel performanslarına yönelik olarak birer örnek rapor aşağıda verilmektedir. Daha detaylı rapor örnekleri ekran çıktılarının yer aldığı ekte verilmektedir. 100 adet kontrol belgesi başvurusunun girişi sisteme yapılmış olup bazı raporlar bu veriler üzerinden oluşturulmuştur. Ancak bazı raporlar ancak çok sayıda verinin sistemde yer almasını gerektirdiğinden veya online başvuru sistemi tarafından üretilen verileri kullandığından bu raporlar için test verisi kullanılmıştır.

Tehlikeli Kimyasal Karnesi

Bu rapor içerisinde öncelikle tehlikeli kimyasalın yapısı, neden olduğu meslek hastalıkları ve parlama, patlama gibi iş kazası riskleri gibi genel bilgilere yer verilir.

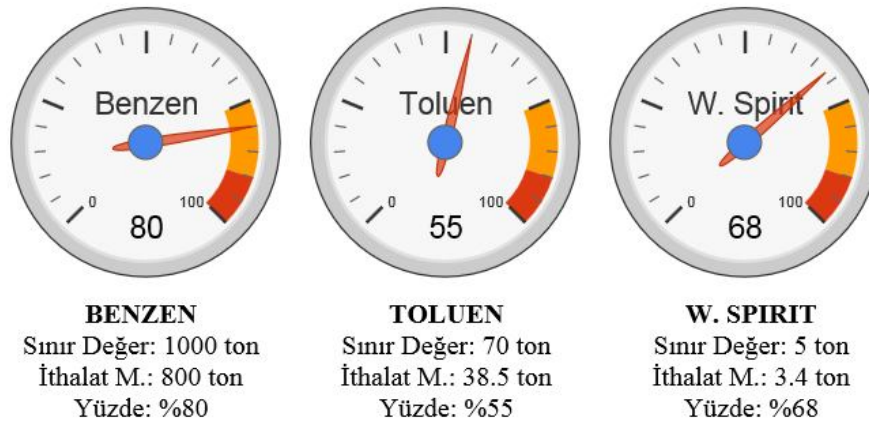
Kimyasal ile ilgili piktogramlar rapora eklenir. İçinde bulunulan yılın ve önceki yılın toplam ithalat miktarları eğer kota belirlenmişse hangi oranda ithalat gerçekleştiği raporlanır.

İthalat detayları bölümünde ithalatı gerçekleştirilen firmalara ulaşılabilir. Beyanlar doğrultusunda ithal edilen ürünlerin hangi faaliyet alanlarında kullanıldığı ve SGK sicil numarası üzerinden hangi işyerlerinde kullanıldığı bu işyerlerindeki iş kazaları ve meslek hastalıkları ile çalışan sayıları, etkilerin görülebilmesi amacıyla raporlanır. Bu işyerlerinde İSGÜM tarafından gerçekleştirilen yerinde inceleme sonuçlarına göre güvensizlik oranları ve görülen eksiklikler listelenerek kimyasalların kullanımı iş sağlığı ve güvenliği açısından incelenmektedir.

İthalat sınırı belirlenen ürün kontrol raporları

Kullanımında kısıt bulunan ve/veya kanserojen, mutajen veya radyoaktif kimyasallar için belirlenen ulusal stratejiler doğrultusunda hedefe ve kısıtlamalar getirilebilir. Bu kısıtlar sisteme girildiğinde ilgili kimyasal ve müstahzarları için gerçek zamanlı olarak o tarihe kadar gerçekleştirilen ithalatlar görüntülenebilir.

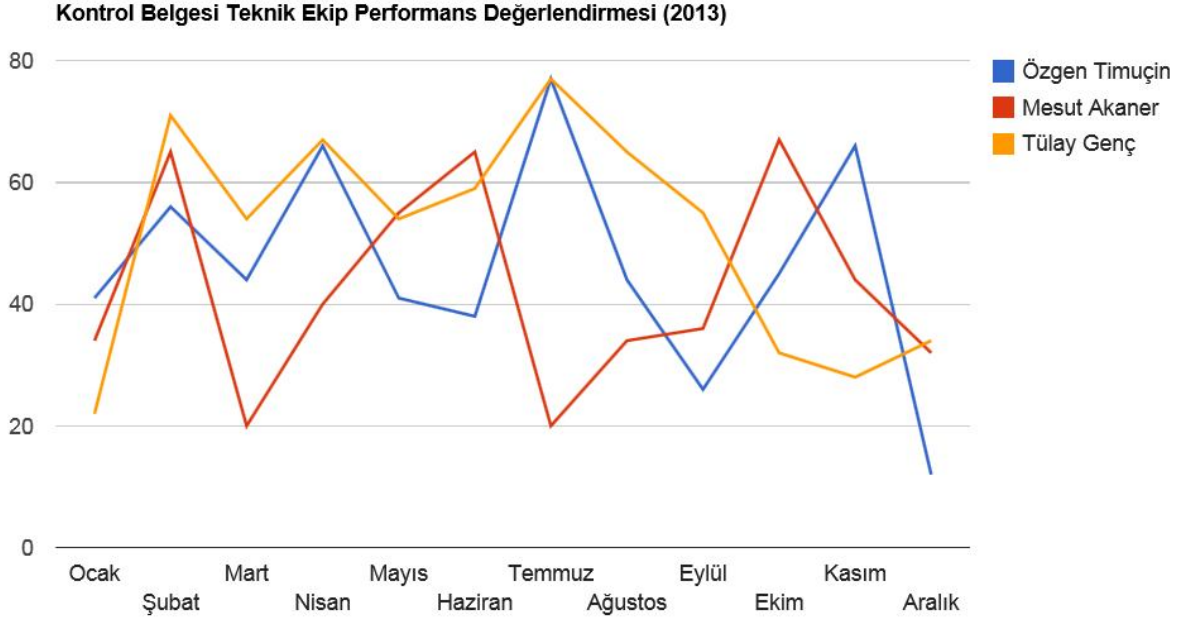
İTHALAT SINIRI BELİRLENEN KİMYASALLAR İÇİN YILLIK MEVCUT DURUM RAPORU (yüzde olarak)



Şekil 3:İthalat Sınırı Belirlenen Kimyasallar İçin Yıllık Mevcut Durum Raporu

Personel Performans İzleme Grafikleri

Bu grafik kullanılarak kontrol belgesi onay işlemlerinde çalışan personelin aylık olarak inceleme sayıları diğer görevli personellerle karşılaştırmalı olarak görülebilmektedir.



Şekil 4: Kontrol Belgesi Teknik Ekip Performans Değerlendirmesi

ÜLKEMİZDE VE DÜNYADA BENZER ÇALIŞMALAR

Gerçekleştirilen çalışmanın ülkemiz ve dünya literatüründeki karşılığına baktığımızda farklı amaçlarla kullanılan pek çok sistemle karşılaşmaktayız. Bu sistemlerin bir kısmı ithal edilen kimyasalların iş sağlığı ve güvenliği açısından değerlendirilmesi üzerinde yoğunlaşırken bir kısmı ithalat işlemlerinin yürütülmesine veya yasaklı ürünlerin kontrolüne ağırlık vermektedir. Avrupa birliği tarafından 2007 yılında uygulanmaya başlanan REACH tüzüğü ve yönetmeliği ile kimyasallar bir envanter altında toplanarak bir kayıt sistemi oluşturulmuştur. Kimyasal kullanımında yüksek düzeyde korunma, risklerin tespiti ve yönetimini sağlama, kimyasalların Avrupa Birliği ülkeleri içerisinde serbest dolaşımını temin gibi temel amaçlar hedeflenmiştir. REACH tüzüğü'nün önemli bir adımı kimyasalların Avrupa Kimyasallar Ajansı sistemine kaydedilmesidir. Bu sayede kimyasallar ve riskleri

gözlenebilmektedir. Benzer şekilde ülkemizde de bazı tehlikeli kimyasalların ithali için başta malzeme güvenlik formları olmak üzere kimyasalın iş sağlığı ve güvenliği açısından önem arz eden bilgileri talep edilmekte uzmanlar tarafından izin aşamasında değerlendirilmektedir.

Dış ticarete bilgi ve belgelerin elektronik ortama taşınmasına yönelik Birleşmiş Milletler bünyesinde bilgisayarlaştırılmış gümrük işletim sistemi olan ASYCUDA geliştirilmiş ve başlatılan UNEDocs projesi kapsamında da halen EDI ve ticaret standartlarına dayanan ve kâğıt temelli olan ticari belgelerin elektronik ortamdaki karşılıklarının oluşturulması ve genel yapı içerisinde elektronik belgelerin ticari firmalar arasındaki alışverişinin sağlanmasına yönelik çalışmalar da sürdürülmektedir[34].

Kimyasalların ithalat kontrolüne ilişkin yapılan literatür taramasında teknik olarak sistem tasarımına yönelik önemli bir bilimsel çalışmaya rastlanmamakla birlikte Başbakanlık tarafından Tek Pencere Sistemi hayata geçirilmiştir. Tek Pencere Sistemi (Single Window), gümrük işlemleri sırasında istenen tüm belgelerin tek noktadan temin edilmesini ve gümrük işlemlerinin tek noktaya yapılacak başvuru ile yürütülerek tamamlanmasını sağlayan sistemdir. Dünyada hâlihazırda Singapur, Güney Kore Almanya, İsveç, Finlandiya, ABD, Guatemala ve Makedonya başta olmak üzere birçok ülkede uygulanmaktadır.

Tek Pencere Sisteminden aşağıda belirtilen faydaların sağlanması beklenmektedir:

- Hâlihazırda muayene ve kontrol ile görevli memurlarca manuel olarak yapılan belge kontrolleri, elektronik ortamda Tek Pencere Sistemi ve BİLGE Sistemi (gümrük işlemlerinin elektronik ortamda gerçekleştirilmesini sağlayan elektronik sistem – “Bilgisayarlı Gümrük Etkinlikleri Sistemi”) entegrasyonu ile gerçekleştirilerek, insan hataları ortadan kaldırılacak, kontrollerin etkinliği artırılacak ve işlem süreleri kısalmaktadır.
- İlgili kurum tarafından belge bilgileri doğrudan Gümrük ve Ticaret Bakanlığına gönderileceğinden belgede sahteciliğin önüne geçilecektir.
- Başvuru, belge, kontrol ve izinler standart hale getirilerek izlenebilirlik artırılacaktır.

- Gmrk ve Ticaret Bakanlıđının kendi mevzuatını uygulama kabiliyeti gçlendirilecektir.
- Firmaların iřlem maliyetleri dřrlerek, uluslararası rekabet gçlerinin artırılması sađlanacaktır.
- Tek Pencere Sisteminin devreye alınmasından sonra kâđıtsız beyanname uygulamalarının da hayata geçirilmesi mmkn olacaktır.
- zellikle e- başvuru çalıřmaları ile uluslararası uygulamalara paralel olarak gmrkl sahada yapılacak kontrollerin piyasa gzetimi ve denetimi ile iç piyasada yapılması sađlanarak eřyaya iliřkin diđer kurumlarca yapılan kontroller nedeniyle gmrk iřlemlerinde yařanan gecikmelerin nne geçilmesi ve bylece gmrk iřlem srelerinin kısaltılması sađlanacaktır.

Bakanlıđımızda dahil olmak zere 9 ayrı bakanlık, Emniyet Genel Mdrlđ, Sermaye Piyasası Kurulu gibi kurumlarda çalıřmalar yrtlmektedir. T.C. řeker Kurumu ile 03.12.2013 tarihinde bařlatılan çalıřmalar sonucunda e-belge uygulaması pilot uygulamaya hazır hale gelmiřtir. Diđer yandan, Trkiye Atom Enerjisi Kurumu ve Enerji Piyasası Dzenleme Kurumu'nda çalıřmalar byk lçde tamamlanmıřtır[35].

TARTIŞMA

Bu bölümde gerçekleştirilen çalışma hem iş sağlığı ve güvenliğine katkıları açısından hem de teknik yapısı itibariyle incelenecektir.

Tehlikeli kimyasal madde ve müstahzarların İSG açısından yarattığı tehlikeler ile ithalat kontrolü ve ithalat verilerinin raporlanmasının önemi ve sağlayacağı yararlar çalışma içerisinde ilgili bölümlerde verilmiştir. Kısaca özetlenecek olursa öncelikle kimyasalların iş sağlığı ve güvenliği açısından içerdiği tehlikeler, ticari isimlendirmelerin dışında kimyasal içeriği, miktarları ve sektörel dağılımının raporlanması İSG konusunda ulusal stratejilerin geliştirilmesine olanak sağlayacaktır.

Kontrol belgesi işlemlerinin en temel öğelerinden biri olan malzeme güvenlik bilgi formu ibrazı ele alındığında ilgili formun teknik personel tarafından incelendiği görülmektedir. Bu dokümanın kimyasalın kullanıldığı işyerindeki işçiye kadar ulaştığı düşünüldüğünde dokümanın doğru hazırlanması büyük önem taşımaktadır. Malzeme güvenlik bilgi formları aynı zamanda bir kullanım kılavuzu olarak düşünülebilir. Kimyasalın taşınması ve depolanmasından bertarafına kadar tüm aşamalarda iş sağlığı ve güvenliği açısından gerekli tedbirler doküman içerisinde yer almaktadır. Tasarlanan sistem sayesinde üretici firmanın hazırlamış olduğu malzeme güvenlik bilgi formunun yanısıra ilgili yönetmeliğe uygun hazır formatta malzeme güvenlik bilgi formunun sisteme ayrı alanlar halinde kaydedilmesi mümkün olacaktır. MSDS dokümanı içerisinde yer alan sağlık ve güvenlik kodları, işaretler, kanserojen, mutajen etkiler gibi pek çok kriter kullanılarak ithal edilen kimyasallar

gruplandırılabilir, incelenebilir ve raporlanabilir. Diğer yandan kimyasalların gösterdiği bu etkiler kullanıldığı işyerlerindeki iş kazaları ve meslek hastalıkları ile eşleştirilerek malzeme güvenlik bilgi formlarının etkinliği ve doğruluğu izlenebilecektir.

Diğer bir konu ise kimyasalların kullanım yerlerinin bildirimidir. Prosedürün bir parçası olarak kimyasalların kullanılacağı işyerlerinin bildirimini sayesinde kimyasalların nelerde kullanıldığının bir kota veya yasaklama olması halinde hangi sektörlerin etkileneceğinin tespiti sağlanmaktadır. İlgili işyerlerinde İSGÜM tarafından gerçekleştirilen yerinde incelemeler ile kimyasalların iş sağlığı ve güvenliği kuralları çerçevesinde malzeme güvenlik bilgi formundaki talimatlara uygun şekilde kullanılıp kullanılmadığının tespitini sağlamakta işyerlerinde farkındalık oluşturmaktadır.

Tehlikeli kimyasal maddelerin kullanıldığı işyerlerinin bildirimini ile SGK iş kazaları ve meslek hastalıkları bildirimini entegre edilerek hangi kimyasalın kullanıldığı yerlerde hangi iş kazası ve meslek hastalığının görüldüğü izlenebilecek, kimyasala özel tehlikeler bazında veriler incelenebilecektir.

Bilgi sistemi incelendiğinde aşağıda belirtilen ana veri kaynakları üzerine kurulduğu görülebilir:

- Sosyal Güvenlik Kurumu İş Kazası ve Meslek Hastalığı Bildirimleri
- Sosyal Güvenlik Kurumu 6 Haneli Faaliyet Alanı Bildirimleri
- SGK İşyeri Tescil ve Bildirge Kayıtları
- İthalatçı firma tarafından yapılan kullanıma ilişkin bildirimler
- Malzeme Güvenlik Bilgi Formlarında yer alan bilgiler
- İSGÜM Yerinde İnceleme Sonuçları

Bu bağlamda bakıldığında veri kaynakları başta Gümrük ve Ticaret Bakanlığı olmak üzere çeşitli kamu kurum ve kuruluşları ile zenginleştirilebilir. Örneğin ithalat kontrollerine ilişkin hâlihazırda İSGÜM tarafından kontrol belgesi işlemleri yürütülmeyen tehlikeli kimyasal ve müstahzarların raporlanması ve bu kimyasalların iş sağlığı ve güvenliği açısından incelenmesi sürdürülen çalışmanın güncel gerekliliklere uygun olarak kendini yenilemesini sağlayacaktır.

Sistem altyapısı veri kaynakları açısından incelendiğinde ayrı veritabanları içerisinde olsa da aynı teknoloji üzerine kurulduklarından gerektiğinde tam uyumlu olarak veri aktarımları ve/veya veritabanlarının birleştirilmesi sağlanabilmektedir. Ayrıca nesne yönelimli programlama yapısı kullanıldığından gerekli düzenlemeler parametrik olarak kolayca gerçekleştirilebilir.

Veritabanı tasarımında kullanılan ilişkisel veritabanı modeli veritabanının kolaylıkla geliştirilebilmesine ve veriler arası ilişkiler doğrultusunda yeni tabloların eklenebilmesine olanak sağlamaktadır. Ayrıca stored prosedürleri desteklemektedir. Ancak sistemde stored prosedürlere yer verilmemektedir. Stored prosedürlerin yerine veritabanı işlemlerinin sınıfların üzerinden yürütülmesi ile aynı sınıfın farklı parametreler için inherit edilmesini sağlamaktadır. Ayrıca veritabanının başka bir veritabanına aktarılması halinde prosedürlerde sorunlar yaşanabilmektedir.

Özellikle raporlama işlemlerinde bir seçenek olarak viewler kullanılabilir. Viewler oluşturularak tablolar arasında kurulmak istenilen ilişki, kullanılacak alanlar tanımlanabilir. Viewlerin oluşturulması erişim kolaylığı, erişim yetkilendirmesi ve aynı sorgunun tekrar çağırılması aşamalarında fayda sağlayabilir. Ancak sistem tasarımında tüm işlemler bilgi sistemi arayüzünde gerçekleştirildiğinde yetkilendirme arayüzleri tetikleyen servlet içerisinde gerçekleştirilmektedir. Yetkilendirme için kullanımı ve takibi kolay koşul yapıları kullanılmaktadır.

Kullanılan Java programlama dili nesne yönelimli programlama dilleri arasında önde gelen dillerden biri olup jsp servlet kod yapısı üzerine oturtulmuştur. Farklı frameworkler bulunmakla birlikte okunabilirlik ve desteklenebilirlik açısından jsp-servlet kod yapısı tercih edilmiştir. Kod yapısı halihazırda Bakanlık Bilgi İşlem Dairesi Başkanlığında kullanılmakta olan yapı ile örtüşmekte olup bu bakımdan her zaman için teknik destek sağlanabilmektedir.

Hazırlanan bilgi sistemi bir karar destek sistemi veya uzman sistem olmaktan ziyade bir yönetim bilgi sistemi olarak karşımıza çıkmaktadır. Karar destek sistemleri ve uzman sistemler kullanıcıya karar verme aşamasında çok daha büyük oranda yardımcı olmakla beraber arkasında dinamik olarak değişen ve güncellenen bir iş mantığı yapısı gerektirdiğinden sürekli olarak karar mekanizmalarının kontrolünü

gerektiren bu sistemlerden kaçınılmıştır. Ancak raporlamalar karar destek sistemlerine yakın hazırlanmış olup, sadece verilerin toplanıp görüntülenmesi şeklinde değil, anlamlı ve çıkarımların yapılabileceği raporlamalar oluşturulmaya çalışılmıştır. Bu aşamada raporların spesifik bir kimyasal veya ithalatçı için gruplandırılarak çıkarılmasında bazı eksiklikler olduğu düşünülebilir.

Çalışmada bilgi sistemi altyapısının oluşturulmasının yanı sıra Kontrol Belgesi işlemlerinin ithalat kontrolü ve iş sağlığı ve güvenliği açısından önemi ve gerekliliğine değinilmiştir. Bu incelemede kimyasallar birer risk etmeni olarak ele alınmıştır. Ancak kimyasalların kullanım alanları sadece kimya sektörü ile sınırlı olmadığından ve iş kazası meslek hastalıkları bildirimlerinde ilgili kimyasalın etkilerini inceleyecek derecede detay bulunmadığından uygun istatistiksel verinin bulunmadığı düşünülebilir. Bu bağlamda iş kazası ve meslek hastalıkları kayıtları ile ithalatçı tarafından bildiri zorunlu olan kullanıcı kayıtlarının örtüştürülerek kullanılması bize eksik olan bu istatistikler zaman içerisinde sağlayacaktır. Kontrolü gerçekleştirilen kimyasalların iş sağlığı ve güvenliği açısından çok farklı etkileri olduğundan bu konuda örneklemelere fazla girilmemiş olup genel olarak büyük ölçüde kanserojen ve mutajen, zehirleyici, alerjik v.b. etki gösterdikleri, parlayıcı ve patlayıcı özelliklere sahip oldukları bilgisi öne çıkmaktadır. Yeterli verinin sisteme entegrasyonu sonrasında spesifik bir kimyasal üzerinde gerçekleştirilecek çalışma fayda sağlayacaktır.

SONUÇLAR

Gerçekleştirilen çalışma ile tehlikeli kimyasal maddelerin ithalat kontrolüne ilişkin olarak bir yönetim bilgi sisteminin tasarımı yapılmıştır. Tasarlanan sistem ile ithalatçı firmanın başvuru işleminden, başvurunun değerlendirmesi, sonuçlandırılması ve ilgili kurum ve kuruluşlara bildirimine kadar tüm süreçlerin çevrimiçi olarak gerçekleştirilmesi için veritabanı, analiz ve tasarım altyapısı oluşturulmuştur. Sistem tasarımında Bakanlık Bilgi İşlem Dairesi Başkanlığınca kullanılmakta olan nesne yönelimli programlama ve ilişkisel veritabanı yapıları desteklenen teknolojik altyapı içerisine oturtularak çalışma gerçekleştirilmiştir. Tasarım çalışmalarında öncelikle veri alışverişine uygun olarak bilgi akışının olduğu veritabanlarındaki veri yapısı incelenerek farklı veritabanlarından alınan verilerin sorunsuz aktarımı için uygun yapı sağlanmıştır. Sistem tasarımında modüler bir yapı kullanılarak güncellemelere imkan sağlanmıştır. Web servislere uyum sayesinde dış sistemlerle entegrasyon sağlanmaktadır.

Tasarımı yapılan sistem ile süreçlerin standardize edilmesi ve kontrolü, çalışan performanslarının incelenmesi, verilerin geriye dönük olarak erişilebilir olması ve istatistiksel olarak değerlendirilebilmesi sağlanacaktır. Sistemde yer alan elektronik imza modülleri ile online başvurulara yasal dayanak oluşturulmuştur. İş takibinin önüne geçilerek, olası sahtecilikler önlenerek ve başvuruların evrak sistemine ve personeline getirdiği yük ortadan kaldırılacaktır. Ülkeye sokulan tehlikeli kimyasallar ve müstahzarlar kontrol altında tutulacak istatistiksel olarak değerlendirilebilecektir.

İş analizi sonucunda yapılan tespitler doğrultusunda, Kontrol Belgesi Sisteminin özellikle veri girişleri, e-devlet kapısı entegrasyonu, e-imza ve mobil imza altyapısının oluşturulması için Döner sermaye işletmesi aracılığıyla alımı planlanan yazılım ve destek hizmeti projelendirilmiştir ve yatırım projesi olarak Kalkınma Bakanlığına sunulmuştur.

Yapılan çalışmanın bir diğer çıktısı tehlikeli kimyasalların ithalat kontrollerinin iş sağlığı ve güvenliği açısından değerlendirilmesi, raporlanması ve stratejik yaklaşımların geliştirilebilmesine ilişkindir. Sistem tarafından tutulacak olan kimyasallara ilişkin malzeme güvenlik formlarında yer alan bilgilere istinaden kimyasallar taşıdığı risklere göre gruplandırılabilmekte, kullanım alanlarına göre değerlendirilebilmektedir.

Hâlihazırda uygulaması devam eden ithal edilen tehlikeli kimyasalların kullanılacağı yerlerin bildirim zorunluluğu sayesinde kimyasalların hangi sektörlerde kullanıldığı tespit edilebilecektir. SGK sicil numarası üzerinden, kaza ve meslek hastalıkları bilgilerine erişilerek kimyasallar ile iş kazası ve meslek hastalıkları arasında korelasyonlar oluşturulabilecektir. Ayrıca İSGÜM tarafından yapılan yerinde inceleme sonuçları kimyasallar bazında değerlendirilerek uygun görülen koşulların iyileştirilmesine yönelik olarak bilinçlendirme, yasal düzenlemeler gibi çalışmalar yürütülebilecektir. Veri girişine yönelik çalışmaların tamamlanması ile sistem entegrasyonu tamamlanarak hem idari faaliyetlerde hem de iş sağlığı ve güvenliği açısından yürütülecek çalışmalarda yarar sağlayacaktır.

İSGÜM tarafından kontrolü gerçekleştirilen tehlikeli kimyasal madde ve müstahzarların İSG açısından getirdiği riskler ilgili teknik ekip tarafından tespit edilerek raporlanmalı bu raporlar sistem üzerinden ithalatçı firmaya ve kullanıcılara ulaşması sağlanmalıdır. Hazırlanacak raporda kimyasalın insan ve çevre sağlığına etkilerinden korunma yollarına kadar malzeme güvenlik bilgi formlarında yer alan bilgilerin yanı sıra özellikle diğer kimyasallarla etkileşimine yer verilmelidir. Böylece kontrol ve denetim çalışmaları bilinçlendirme çalışmaları ile desteklenerek iş kazaları ve meslek hastalıklarının minimuma indirilmesinde riski azaltan bir yaklaşım elde edilmelidir.

KAYNAKLAR


1. Korkmaz O. Türkiye Kimya Sanayinde İş Sağlığı ve İş Güvenliği. ZKÜ Sosyal Bilimler Dergisi. 2011; 7(14): 129-144.
2. Türkiye Kimya Sektörü Strateji Belgesi ve Eylem Planı 2012-2016. T.C. Bilim Sanayi ve Teknoloji Bakanlığı; 2012.
3. Kimya Sektörü Raporu (2013/1). T.C. Bilim, Sanayi ve Teknoloji Bakanlığı; 2013.
4. Bilgi İşlem Merkezi Verileri. T.C. Ekonomi Bakanlığı;
<http://www.ekonomi.gov.tr/index.cfm?sayfa=79192159-19DB-2C7D-3D5AE56731D11E50>.
5. Kimya Sektörü. T.C. Ekonomi Bakanlığı; 2013. HS No : 2705-2715, 28-39, 4001-4010, 4014-4017.
6. Işık Coşkunes F, Tehlikeli Kimyasal Maddelerin Oluşturduğu Riskler İçin Genel ve Özel Önleme Yöntemleri, Ankara : İş Sağlığı ve Güvenliği Enstitüsü Müdürlüğü.
<http://www.isgum.gov.tr/rsm/file/isgdoc/IG2-TKM-onlemler.pdf>.
7. Kimyasal Maddelerle Çalışmalarda Sağlık ve Güvenlik Önlemleri Hakkında Yönetmelik. (2013). T.C. Resmi Gazete, 28733, 12 Ağustos 2013.

8. Küçük S. Endüstriyel Tesislerde Tehlikeli Kimyasal Yönetiminin İlkeleri. IV. İş Sağlığı ve Güvenliği Kongresi : 2007; Adana, Türkiye. 20-21.
9. İş Sağlığı ve Güvenliğine İlişkin İşyeri Tehlike Sınıfları Tebliği. (2012). T.C. Resmi Gazete, 28509, 26 Aralık 2012 (değişiklikler işlenmiştir).
10. <http://www.isvesosyalguvenlik.com/isyeri-tehlike-siniflari-listesi-nace-kodlari-yayimlandi-26-12-2012/>.
11. Tehlikeli Maddeler ve Müstahzarlara İlişkin Güvenlik Bilgi Formlarının Hazırlanması ve Dağıtılması Hakkında Yönetmelik. (2008). T.C. Resmi Gazete, 27092 Mükerrer, 26 Aralık 2008.
12. http://msdshouse.net/bilgi-formu/0_bilgi
13. <http://www.baskent.edu.tr/~msagsan/downloads/YBS.pdf>.
14. Tahirov A. Bilgisayar Destekli Bilgi Sistemleri. Journal of Qafqaz University. 2009; 27: 123.
15. Mizrahi, R. Kobi'lerde Muhasebe Bilgi Sisteminin Etkin Kullanımı Üzerine Bir Araştırma. Organizasyon ve Yönetim Bilimleri Dergisi. 2011; 3(2).
16. Demircan L. M, Moltay A. C. Bilgiyi Yönetmek. İstanbul : Beta Basım Yayım, 1997 : 59.
17. http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=461 09.11.2013.
18. <http://www.odevturk.com/indirrr2.asp?id=52366> 06.11.2007.
19. <http://www.maximumbilgi.com/default.asp?sx=mkl&ID=107> 06.11.2013.
20. <http://www.akdeniz.edu.tr/muhfak/cevre/coastlearn-r/planning/tools3.htm> 06.11.2013.

21. Baęçacı F. Web Tabanlı Programlama Dillerinin Deęerlendirilmesi ve Performans Analizi. Elazığ : Fırat Üniversitesi Fen Bilimleri Enstitüsü; 2005.
22. Örnek Ö. Sunucu Tarafı Programlama Dillerinde Fonksiyonel Dil İle Nesne Tabanlı Programlama Dillerinin Karşılaştırılması. Sakarya : Sakarya Üniversitesi Fen Bilimleri Enstitüsü; 2006.
23. <http://www.elektrikport.com/makale-detay/hangi-programlama-dilini-secmeliyim-programlama-dilleri-ve-uygulama-alanlari/8080#ad-image-0>
24. ODTÜ (Türkiye). ION510 Dersi Online Ders Notları Chapter1. Ankara : Enformatik Enstitüsü; 2014.
25. <http://www.cemdemir.net/php/php-nesneye-yonelik-programlama-1958.html>.
26. Önder E. Yönetim Bilişim Sistemleri Kapsamında Web Tabanlı İlişisel Veritabanı Yönetim Sistemleri ve Bir Uygulama (Yüksek Lisans Tezi). İstanbul : İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Sayısal Yöntemler Bilim Dalı; 2005.
27. Akaner M. Gazi Üniversitesi Bilimsel Araştırma Projeleri Müdürlüğünde Karar Destek Sisteminin Oluşturulması (Yüksek Lisans Tezi). Ankara : Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı; 2004.
28. Saga K. Vision and Strategy for E-Government. Tokyo : Digital Oppurtunity Forum, 2001 : 135.
29. Çelikkol Ö. Kamu Yönetiminde e-Devlet Yapılanması ve Türkiye İçin e-Devlet Model Önerisi. Isparta : Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Kamu Yönetimi Anabilim Dalı; 2008.
30. www.tdk.gov.tr

31. Elektronik İmza Kanunu. (2004). T. C. Resmi Gazete, 25355, 23 Ocak 2004.
32. Cryptographic Message Syntax. IETF RFC 3369. 2002.
33. Selçuk G. H, e-Devlet Uygulamaları İçin Elektronik İmza Formatları. <http://ueimzas.gazi.edu.tr/pdf/poster/56.pdf>.
34. Ünsal S. S., Ticaretin Kolaylaştırılması Bağlamında Tek Pencere Uygulaması (Uzmanlık Tezi). Ankara: T.C. Başbakanlık Gümrük Müsteşarlığı AB ve Dış İlişkiler Genel Müdürlüğü; 2007.
35. Tek Pencere Projesi (Bilgi Notu). Ankara: T.C. Gümrük ve Ticaret Bakanlığı;2014.

ÖZGEÇMİŞ

Kimlik Bilgileri				
İsim Soyadı	MESUT AKANER			
Doğum Tarihi	03.01.1978			
Telefon	0312 296 68 95			
e-posta	makaner@csgb.gov.tr			
Eğitim Dereceleri				
Okul	Bölüm	Alan	Derece	Mezuniyet Tarihi
GAZİ Üniversitesi	Kazaların Çev. ve Tek. Araştırması	İSG	Doktora	Devam ediyor
GAZİ Üniversitesi	Endüstri Mühendisliği	BAP Karar Destek Sistemi	Yüksek Lisans	13.09.2004
ODTÜ	Maden Mühendisliği	ISO 9000 ve OHSAS 18000 Uygulaması	Yüksek Lisans	05.09.2003
ODTÜ	Maden Mühendisliği	Maden Mühendisliği	Lisans	25.06.2000
ÇSGB	İş Sağlığı ve Güvenliği ile ilgili Eğitimler Eğitimcilerin Eğitimi			
SGK Bilgişlem	Çeşitli IBM Eğitimleri Temel Java Eğitimi (70 saat), İleri Java Eğitimi (50 saat)			
Intergraph Co.	Geomedia Professional 6.0 Delta			
SYBASE Inc.	Sertifikalı "Fast Track to Power Builder 7.0" Eğitimi			
SYBASE Inc.	Sertifikalı "Adaptive Server Anywhere Fundamentals" Eğitimi			
Sağlık Bakanlığı	ORACLE-SQL Eğitimi (10 ders)			
ABC Yayıncılık	İletişim Becerileri Seminerleri (20 hafta)			
Askerlik Durumu				
Kısa Dönem Jandarma Çavuş – Kars, Bursa				
İlgi Alanları				
◇ Toplam Kalite Yönetimi (TKY) ◇ Bilgisayar Programcılığı ◇ ISO 9000 Serisi ◇ OHSAS 18000 Serisi ◇ Proje Yönetimi ◇ İş Sağlığı ve Güvenliği				
İş Tecrübesi				
Kurum	Pozisyon			
ÇSGB İSGGM	İSG Uzman Yardımcısı			
SGK Bilgi İşlem Dairesi Başkanlığı	Sözleşmeli Programcı			
PenaTrade	Projeler Koordinatörü			
Mescioğlu Mühendislik	GIS Uzmanı / Kalite Yönetim Temsilcisi / İhale Birim Sorumlusu			
ABC Dergisi Dershaneleri	Kalite Yönetim Temsilcisi / Bilgisayar Programcısı			
ODTÜ Maden Mühendisliği Bölümü	Öğrenci asistanı			
Yabancı Dil Bilgisi				
İyi derecede İngilizce (KPDS Notu : 85)				
Bilgisayar Bilgisi				
Çeşitli programlama dilleri, veritabanı sorgulama dilleri ve yönetim sistemleri, grafik programları				

EKLER

EK 1 : EKLAN GÖRÜNTÜLERİ, VERİTABANI YAPISI VE KOD ÖRNEKLERİ

T.C.
ÇALIŞMA VE SOSYAL GÜVENLİK BAKANLIĞI
İŞ SAĞLIĞI VE GÜVENLİĞİ GENEL MÜDÜRLÜĞÜ

**TEHLİKELİ KİMYASAL MADDELERİN İTHALAT KONTROLÜ
İŞLEMLERİNİN İŞ SAĞLIĞI VE GÜVENLİĞİ AÇISINDAN
DEĞERLENDİRİLMESİ İLE KONTROL BELGESİ YÖNETİM
BİLGİ SİSTEMİ YAPISININ OLUŞTURULMASI**

**ÖRNEK EKРАН GÖRÜNTÜLERİ, VERİTABANI YAPISI
KOD KİTAPÇIĞI**

Mesut AKANER

(İş Sağlığı ve Güvenliği Uzmanlık Tezi)

ANKARA-2014

İçindekiler

PROGRAM KODU ÖRNEKLERİ	4
SERVLETLER	4
ActionMultiplexer	4
<i>Action</i> PAKETİ CLASSLARI.....	9
Action.....	9
BeanleriSessionaDoldur	10
HaritaILDegerTopla.....	11
IIRenkBelirle.....	12
IthalatGaugeHazirla	12
IthalatciArrayList.....	14
KimyasallarArrayList	16
PersonelCalisma	17
PostLoginKontrol	18
SessionAc	20
<i>Database</i> PAKETİ CLASSLARI.....	22
ConnManager.....	22
<i>Exceptions</i> PAKETİ CLASSLARI.....	26
ApplicationException.....	26
ApplicationExceptionMessages	27
<i>Ortakislemeler</i> PAKETİ CLASSLARI.....	29
IstemciIPAI	29
Ortakislemeler.....	29
ParametrelerArraye.....	46
RenkBelirle.....	46
SessionIDGenerator.....	48
TimeStampOlustur	48
<i>Model</i> PAKETİ CLASSLARI.....	49
HaritaIIBean	49
<i>Utilities</i> PAKETİ CLASSLARI.....	50
UriParametreler	50
Util.....	50
FACE SAYFALARI.....	55

Header.....	55
Footer.....	59
ARAYÜZ SAYFA ÖRNEKLERİ.....	60
Anasayfa.jsp.....	60
Aylıksurunithalatgos.jsp.....	60
İthalatraporlari.jsp.....	61
Kimyasalgauges.jsp.....	62
Msdslr.jsp.....	63
Perraporlari.jsp.....	64
Urunraporlari.jsp.....	75
belgemevsim.jsp.....	75
belgerapor.jsp.....	76
ithalatraporlari.jsp.....	76
perraporlari.jsp.....	77
personelbelgedagilim.jsp.....	77
thk.jsp.....	78
urunraporlari.jsp.....	86
VERİTABANI KURULUM SQL ÖRNEKLERİ.....	86
ÖRNEK EKLAN GÖRÜNTÜLERİ.....	93

PROGRAM KODU ÖRNEKLERİ

SERVLETLER

ActionMultiplexer

```
package servlet;
import action.Action;
import action.BeanleriSessionaDoldur;
import action.lthalatGaugeHazirla;
import action.lthalatciArrayList;
import action.KimyasallarArrayList;
import action.SessionAc;
import action.SesssionKontrol;
import action.UrunlthalatListeGetir;
import database.ConnManager;
import java.io.IOException;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import model.lthalatciBean;
import model.KimyasalBean;
import model.LoginBilgiBean;
import utilities.UriParametreler;
public class ActionMultiplexer extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String SS_FROM = "EBL";
    private static final String SS_TO = "JNA";
    public ActionMultiplexer() {
        super();
```

```

    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doPost(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        request.setCharacterEncoding("UTF-8");
        boolean sessionacildimi = false;
        Action anAction;
        HttpSession session = request.getSession(true);
        String actionID = request.getPathInfo();
        Connection conn;
        conn = new ConnManager().connectionOlustur();
        if ((actionID.equals("")) || (actionID.equals(null))) {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/login.jsp").forward(r
equest, response);
            System.out.println("ilk giriş logine gider");
        }
        if (!(actionID.equals("")) && !(actionID.equals(null))) {
            System.out.println("SesssionKontrol için action");
            try {
                anAction = new SesssionKontrol();
                anAction.triggerAction(request, response, conn);
            } catch (Exception ex) {
                ex.printStackTrace();
                System.out.println("catchhhhhh");
                Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        if ("/POSTLOGIN".equals(actionID)) {
            System.out.println("getttttttttttt" + actionID);
            LoginBilgiBean loginbilgi = new LoginBilgiBean();
            loginbilgi.setKullaniciAdi(request.getParameter("KullaniciAdi"));
            loginbilgi.setSifre(request.getParameter("Sifre"));
            request.getSession().setAttribute("loginbilgi", loginbilgi);
            try {
                anAction = new SessionAc();
                anAction.triggerAction(request, response, conn);
            }

```

```

        sessionacildimi = true;
    } catch (Exception ex) {
        ex.printStackTrace();
        System.out.println("catchhhhhh");
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (sessionacildimi) {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/anasayfa.jsp").forward(request, response);
    }
    }
    if ("/PERRAPOR".equals(actionID)) {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/perraporlari.jsp").forward(request, response);
    }
    if ("/ITHALATRAPOR".equals(actionID)) {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/ithalatraporlari.jsp").forward(request, response);
    }
    if ("/ITHALATGAUGE".equals(actionID)) {
        String buyil = String.valueOf(Calendar.getInstance().get(Calendar.YEAR));
        if (request.getAttribute("gaugeyil") == null) {
            request.setAttribute("gaugeyil", buyil);
        }
        //request.getSession(true).setAttribute("gaugeyil", 'cvcvxv');
        try {
            new lthalatGaugeHazirla().gaugeveristring(request, response, conn,
String.valueOf(request.getAttribute("gaugeyil")));
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/kimyasalgauges.jsp").forward(request, response);
        } catch (Exception ex) {
            Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    if ("/AYLIKPERSAYI".equals(actionID)) {
        try {
            new BeanleriSessionaDoldur().personelBeanDoldur(request, response);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/aypersayi.jsp").forward(request, response);
        } catch (Exception ex) {

```

```

        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
if ("/PERPERF".equals(actionID)) {
    try {
        // new ParametrelerArraye().parametrelerArraye(request, response);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/personelperforman
s.jsp").forward(request, response);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
if ("/URUNRAPOR".equals(actionID)) {
    try {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/urunraporlari.jsp").f
orward(request, response);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
System.out.println("aylik ithalatagirmedneennenenen");
if ("/AYLIKURUNITH".equals(actionID)) {
    try {
        System.out.println("aylik ithalatagirdidididi");
        ArrayList<KimyasalBean> kimyasallist = new ArrayList<KimyasalBean>();
        kimyasallist = new KimyasallarArrayList().kimyasallisteOlustur(request, response, conn);
        request.getSession().setAttribute("kimyasallist", kimyasallist);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/aylikurunithalatsec.j
sp").forward(request, response);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
if ("/AYLIKURUNITHG".equals(actionID)) {
    ArrayList<KimyasalBean> kimyasallist = new ArrayList<KimyasalBean>();
    try {
        kimyasallist = new KimyasallarArrayList().kimyasallisteOlustur(request, response, conn);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

request.getSession().setAttribute("kimyasallist", kimyasallist);
String urunlithalatay = "";
try {
    urunlithalatay = new UrunlithalatListeGetir().ithalatlisteOlusturStr(request, response, conn);
} catch (Exception ex) {
    Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
}
request.getSession().setAttribute("urunlithalatay", urunlithalatay);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/aylikurunithalatgos.jsp").forward(request, response);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/sablonlar/aylikuruni
thalat.jsp").forward(request, response);
}
if ("/ITHALATURUNULKE".equals(actionID)) {
    try {
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/sablonlar/haritasabl
on.jsp").forward(request, response);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
if ("/ITHALATFIRMALAR".equals(actionID)) {
    System.out.println("ITHALATFIRMALAR iff");
    ArrayList<IthalatciBean> ithalatcilist = new ArrayList<IthalatciBean>();
    try {
        ithalatcilist = new IthalatciArrayList().ithalatcilisteOlustur(request, response, conn);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
    request.getSession().setAttribute("ithalatcilist", ithalatcilist);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/ithalatcilistegosp.jsp"
).forward(request, response);
}
if ("/ITHALATCIDETAY".equals(actionID)) {
    try {
        String url = request.getQueryString();
        //Map<String, String> urlgelenmap = new HashMap<String, String>();
        Map <String, String> urlgelenmap = new UriParametreler().querymapYap(url);
        ArrayList<KimyasalBean> kimyasallist = new ArrayList<KimyasalBean>();
        kimyasallist = new KimyasallarArrayList().kimyasallisteOlustur(request, response, conn);

```

```

        request.getSession().setAttribute("kimyasallist", kimyasallist);
request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/aylikurunithalatsec.j
sp").forward(request, response);
    } catch (Exception ex) {
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
}
try {
    conn.close();
} catch (SQLException ex) {
    Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
}
}
@Override
public String getServletInfo() {
    return "Short description";
}
}
}

```

Action PAKETİ CLASSLARI

Action

```

package action;
import java.sql.Connection;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public abstract class Action {
public Action() {
    super();
}
public void triggerAction (HttpServletRequest req, HttpServletResponse resp, Connection conn )
throws Exception {
}
public void triggerAction (HttpServletRequest req, HttpServletResponse resp, int sayi, Connection
conn ) throws Exception {
}
public int triggerActionint (HttpServletRequest req, HttpServletResponse resp, Connection conn )
throws Exception {
    return 0;
}
}

```

```

public boolean triggerActionbool (HttpServletRequest req, HttpServletResponse resp, String str,
Connection conn ) throws Exception {
    return false;
}
}

```

BeanleriSessionaDoldur

```

package action;
// buraya eger sessionda yoksa doldur kontrol konulacak
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import database.ConnManager;
import model.PersonelBean;
import ortakislemler.OrtakIslemler;
public class BeanleriSessionaDoldur extends Action {
    private static ServletContext servletContext = null;
public synchronized void personelBeanDoldur(HttpServletRequest request, HttpServletResponse
response) throws Exception {
    JspPageBean jb = new JspPageBean();
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    ArrayList<PersonelBean> PersonelList = new ArrayList();
    try {
        -----
        String selectsqlstr = "SELECT\n" +
"ANONYMOUS.\\"personel\\".\\"personelid\\",\n" +
"ANONYMOUS.\\"personel\\".\\"adi\\",\n" +
"ANONYMOUS.\\"personel\\".\\"soyadi\\",\n" +
"ANONYMOUS.\\"personel\\".\\"kurumu\\",\n" +
"ANONYMOUS.\\"personel\\".\\"tckimlikno\\\"\n" +
"FROM\n" +
"ANONYMOUS.\\"personel\\\"\n" +
"WHERE\n" +
"\"silmedurumu\"=0 \n" +

```

```
"ORDER BY \"adi\", \"soyadi\"";
```

```
-----  
    //st = con.createStatement();  
    //rs = st.executeQuery(sonkarnenosec);  
    while (rs.next()) {  
        PersonelBean bean = new PersonelBean();  
        bean.setPersonelid(rs.getInt(1));  
        bean.setAdi(rs.getString(2));  
        bean.setSoyadi(rs.getString(3));  
        bean.setKurumu(rs.getString(4));  
        bean.setTckimlikno(rs.getString(5));  
        PersonelList.add(bean);  
    }  
    request.getSession().setAttribute("PersonelList", PersonelList);  
    con.close();  
    /*    st.close();*/  
    rs.close();  
} catch (SQLException e) {  
    e.printStackTrace();  
} catch (Exception e) {  
    e.printStackTrace();  
} finally {  
    ConnManager.close(rs);  
    ConnManager.close(st);  
}  
}  
}
```

HaritalLDegerTopla

```
package action;  
import java.util.ArrayList;  
import model.HaritalBean;  
public class HaritalLDegerTopla {  
    public synchronized float ildegerleriTopla(ArrayList<HaritalBean> haritaillist) throws Exception {  
        float illertoplami=0;  
        int i = 0;  
        while (haritaillist.size(>i){  
            HaritalBean haritabean = new HaritalBean();  
            haritabean= haritaillist.get(i);  
            illertoplami=illertoplami+haritabean.getIldeger();  
        }  
    }  
}
```



```

        i++;
    }
    return illertoplami;
}
}

```

IIRenkBelirle

```

package action;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.HaritaIIBean;
import ortakislem.RenkBelirle;
public class IIRenkBelirle {
    public synchronized ArrayList<HaritaIIBean> ilrenkDoldur(HttpServletRequest request,
    HttpServletResponse response, ArrayList<HaritaIIBean> haritaillist) throws Exception {
    float illertoplami=0;
    illertoplami=new HaritaIIDegerTopla().ildegerleriTopla(haritaillist);
    ArrayList<HaritaIIBean> haritaillistdolu= new ArrayList<HaritaIIBean>();
    // burada iller toplami hesaplat
    int i = 0;
    while (haritaillist.size(>)>i){
    float ilyuzdesi=0;
    String ilrenji="";
        HaritaIIBean haritabean = new HaritaIIBean();
        haritabean= haritaillist.get(i);
        ilyuzdesi=haritabean.getIldeger()*100/illertoplami;
        ilrenji= new RenkBelirle().yuzderenkBelirle(request, response, ilyuzdesi);
        haritabean.setIrenk(ilrenji);
        haritaillistdolu.add(haritabean);
        i++;
    }
    return haritaillistdolu;
    }
}

```

IthalatGaugeHazirla

```

package action;
import com.sun.org.apache.xalan.internal.xsltc.runtime.BasisLibrary;
import java.sql.Connection;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import database.ConnManager;
import java.sql.PreparedStatement;
import model.PersonelBean;
import ortakislemler.OrtakIslemler;
public class IthalatGaugeHazirla extends Action {
    private static ServletContext servletContext = null;
    public synchronized String gaugeveristring (HttpServletRequest request, HttpServletResponse
response, Connection con, String yil) throws Exception {
        String gaugestring = "";
        String gaugetablo ="";
        try {
            String selectsqlstr = "SELECT ANONYMOUS.\"olcubirimler\".\"olcubirimadi\",
ANONYMOUS.\"kimyasalkota\".\"yil\", ANONYMOUS.\"kimyasalkota\".\"kimyasaladi\",
ANONYMOUS.\"kimyasalkota\".\"miktar\", sum(ANONYMOUS.\"basvuru\".\"miktar\") as toplammiktar,
(sum(ANONYMOUS.\"basvuru\".\"miktar\") *100/ANONYMOUS.\"kimyasalkota\".\"miktar\") as yuzde
FROM ANONYMOUS.\"kimyasalkota\" INNER JOIN ANONYMOUS.\"basvuru\" ON
ANONYMOUS.\"basvuru\".\"urunid\" = ANONYMOUS.\"kimyasalkota\".\"ithalaturunid\" INNER JOIN
ANONYMOUS.\"olcubirimler\" ON ANONYMOUS.\"kimyasalkota\".\"birimi\" =
ANONYMOUS.\"olcubirimler\".\"olcubirimid\" WHERE
ANONYMOUS.\"kimyasalkota\".\"silmedurumu\"=0 AND ANONYMOUS.\"basvuru\".\"silmedurumu\"=0
and extract(year from ANONYMOUS.\"basvuru\".\"gbeyannametarih\")=? group by
ANONYMOUS.\"olcubirimler\".\"olcubirimadi\", ANONYMOUS.\"kimyasalkota\".\"yil\",
ANONYMOUS.\"kimyasalkota\".\"kimyasaladi\", ANONYMOUS.\"kimyasalkota\".\"miktar\"";
            System.out.print(selectsqlstr+"----"+yil);
            PreparedStatement pst = con.prepareStatement(selectsqlstr);
            pst.setString(1, yil);
            ResultSet rs = pst.executeQuery();
            /// [Label, Value],[Behhhhhhhhhhnzen', 80], [Toluen', 55], [W. Spirit', 68]
            gaugestring = gaugestring+"[Label, 'Value']";
            while (rs.next()) {
                gaugestring = gaugestring+"['"+rs.getString("kimyasaladi") +"'"+ rs.getString("yuzde")+"]";
                gaugetablo=gaugetablo+"<td align=center
><B>"+rs.getString("kimyasaladi")+</b><BR>Sınır Değer: "+rs.getString("miktar")+
"+rs.getString("olcubirimadi")+

```

```

                "<BR>İthalat M.: "+rs.getString("toplammiktar")+
"+rs.getString("olcubirimadi")+"<BR>Yüzde: %"+rs.getString("yuzde")+"</TD>";
        /* <td align=center width=180>
<B>BENZEN</b><BR>
Sınır Değer: 1000 ton<BR>
İthalat M.: 800 ton<BR>
Yüzde: %80
</TD> */
        }
        request.setAttribute("gaugestring", gaugestring);
        request.setAttribute("gaugetablo", gaugetablo);
        rs.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return gaugestring;
}
}

```

IthalatciArrayList

```

package action;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.IthalatciBean;
public class IthalatciArrayList {
    public synchronized ArrayList<IthalatciBean> ithalatcilisteOlustur(HttpServletRequest request,
HttpServletResponse response, Connection con) throws Exception {
        ArrayList<IthalatciBean> ithalatcilist = new ArrayList<IthalatciBean>();
        try {
            String selectsqlstr = "select anonymous.\"firmalar\".\"firmaid\" as ithalatciid,
anonymous.\"firmalar\".\"sgkno\" as ithalatsgkno, anonymous.\"firmalar\".\"verginol\" as
ithalacivergino, anonymous.\"firmalar\".\"firmaadres\" as ithalaciadres,
anonymous.\"firmalar\".\"firmatelefon\" as ithalaticitel, anonymous.\"firmalar\".\"firmafaks\" as

```

ithalatcifaks, anonymous.\`firmalar\`.\`firmaeposta\` as ithalatcieposta,
 anonymous.\`firmalar\`.\`firmail\` as ithalatciilid, anonymous.\`firmalar\`.\`firmaadi\` as ithalatciunvan,
 anonymous.\`iller\`.\`iladi\` as ithalatciiladi from anonymous.\`firmalar\` inner join anonymous.\`iller\` on
 anonymous.\`firmalar\`.\`firmail\` = anonymous.\`iller\`.\`ilid\` where
 anonymous.\`firmalar\`.\`silmedurumu\` = 0 order by \`firmaadi\`";

```

    System.out.print(selectsqlstr);
    PreparedStatement pst = con.prepareStatement(selectsqlstr);
    ResultSet rs = pst.executeQuery();
    while (rs.next()) {
        IthalatciBean ithbean = new IthalatciBean();
        String selectbasvurusayi = "SELECT count(ANONYMOUS.\`basvuru\`.\`firmadi\`) as
basvurusayi FROM ANONYMOUS.\`basvuru\` where ANONYMOUS.\`basvuru\`.\`silmedurumu\`=0
and ANONYMOUS.\`basvuru\`.\`firmadi\`=?"";
        System.out.print(selectbasvurusayi);
        PreparedStatement pstbasvuru = con.prepareStatement(selectbasvurusayi);
        pstbasvuru.setInt(1, rs.getInt("ithalatciid"));
        ResultSet rsbasvuru = pstbasvuru.executeQuery();
        int basvurusayisi = 0;
        while (rsbasvuru.next()) {
            basvurusayisi = rsbasvuru.getInt("basvurusayi");
        }
        ithbean.setIthalatciid(rs.getInt("ithalatciid"));
        ithbean.setIthalatcisgkno(rs.getString("ithalatcisgkno"));
        ithbean.setIthalatcivergino(rs.getString("ithalatcivergino"));
        ithbean.setIthalatciadres(rs.getString("ithalatciadres"));
        ithbean.setIthalatcitel(rs.getString("ithalatcitel"));
        ithbean.setIthalatcifaks(rs.getString("ithalatcifaks"));
        ithbean.setIthalatcieposta(rs.getString("ithalatcieposta"));
        ithbean.setIthalatciilid(rs.getInt("ithalatciilid"));
        ithbean.setIthalatciiladi(rs.getString("ithalatciiladi"));
        ithbean.setIthalatciunvan(rs.getString("ithalatciunvan"));
        ithbean.setBasvurusayi(basvurusayisi);
        ithalatcilist.add(ithbean);
    }
    rs.close();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}

```

```

        return ithalaticilist;
    }
}

```

KimyasallarArrayList

```

package action;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.KimyasalBean;
public class KimyasallarArrayList {
    public synchronized ArrayList <KimyasalBean> kimyasallisteOlustur(HttpServletRequest request,
    HttpServletResponse response, Connection con) throws Exception {
        ArrayList <KimyasalBean> kimyasallist=new ArrayList<KimyasalBean>();
        try {
            String selectsqlstr = "SELECT ANONYMOUS.\"ithalaturun\".\"urunid\",
ANONYMOUS.\"ithalaturun\".\"urunadi\", ANONYMOUS.\"ithalaturun\".\"urunkodu\" FROM
ANONYMOUS.\"ithalaturun\" order by \"urunadi\"";
            System.out.print(selectsqlstr);
            PreparedStatement pst = con.prepareStatement(selectsqlstr);
            ResultSet rs = pst.executeQuery();
            while (rs.next()) {
                KimyasalBean kimbean =new KimyasalBean() ;
                kimbean.setKimyasalid(rs.getInt("urunid"));
                kimbean.setKimyasalkodu(rs.getString("urunkodu"));
                kimbean.setKimyasaladi(rs.getString("urunadi"));
                kimyasallist.add(kimbean);
            }
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return kimyasallist;
    }
}

```

PersonelCalisma

```
package action;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import database.ConnManager;
import java.sql.PreparedStatement;
import model.PersonelBean;
import ortakislemeler.Ortakislemeler;
public class PersonelCalisma extends Action {
    private static ServletContext servletContext = null;
    public synchronized String yilaypersonelcalismaArray(HttpServletRequest request,
HttpServletResponse response, Connection con, String yil) throws Exception {
        String personelcalismaarray = "";
        try {
            String selectsqlstr = "SELECT\n"
                + "ANONYMOUS.\"personel\".\"adi\", \n"
                + "ANONYMOUS.\"personel\".\"personelid\", \n"
                + "ANONYMOUS.\"personel\".\"soyadi\" \n"
                + "FROM\n"
                + "ANONYMOUS.\"personel\" \n"
                + "INNER JOIN ANONYMOUS.\"basvuru\" ON ANONYMOUS.\"personel\".\"personelid\"
= ANONYMOUS.\"basvuru\".\"onaypersonel\" \n"
                + "WHERE\n"
                + "ANONYMOUS.\"basvuru\".\"silmedurumu\"=0 \n"
                + "and \n"
                + "EXTRACT (YEAR FROM ANONYMOUS.\"basvuru\".\"onaytarih\")=?";
            PreparedStatement pst = con.prepareStatement(selectsqlstr);
            pst.setString(1, yil);
            ResultSet rs = pst.executeQuery();
            ArrayList aylar = new ArrayList();
            aylar.add("Aylar");
            aylar.add("Ocak");
            aylar.add("Şubat");
            aylar.add("Mart");
```

```

        aylar.add("Nisan");
        aylar.add("Mayıs");
        aylar.add("Haziran");
        aylar.add("Temmuz");
        aylar.add("Ağustos");
        aylar.add("Eylül");
        aylar.add("Ekim");
        aylar.add("Kasım");
        aylar.add("Aralık");
        while (rs.next()) {
personelcalismaarray = personelcalismaarray+"";
            // + kullanicid = rs.getInt("kullanicid");
        }
        rs.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    return personelcalismaarray;
}
}

```

PostLoginKontrol

```

package action;
import database.ConnManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.LoginBilgiBean;
import servlet.ActionMultiplexer;
public class PostLoginKontrol extends Action {
    public int triggerActionint(HttpServletRequest request, HttpServletResponse response, Connection
conn) throws Exception {
        System.out.println("PostLoginKontrol");
        int kullanicid = 0;

```

```

        boolean kullanicikarsilikvar = false;
        try {
            // Connection conn;
            // conn = new ConnManager().connectionOlustur();
            String sqlstatement = "SELECT ANONYMOUS.\"kullanicilar \".\"kullaniciid\" FROM
ANONYMOUS.\"kullanicilar \" WHERE ANONYMOUS.\"kullanicilar \".\"kullaniciadi\" = ? AND
ANONYMOUS.\"kullanicilar \".\"sifre\" = ? ";
            PreparedStatement pst = conn.prepareStatement(sqlstatement);
            LoginBilgiBean loginbilgi = (LoginBilgiBean) request.getSession().getAttribute("loginbilgi");
            pst.setString(1, loginbilgi.getKullaniciAdi());
            pst.setString(2, loginbilgi.getSifre());
            ResultSet rs = pst.executeQuery();
            while (rs.next()) {
                kullaniciid = rs.getInt("kullaniciid");
            }
            if (kullaniciid==1) {
                String alertmesaj = "Kullanıcı adı/ şifresi hatalı. /n Lütfen sisteme tekrar giriş yapınız";
                request.getSession().setAttribute("jsalertmetin", alertmesaj);
                request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/login.jsp");
            }
            // conn.close();
        } catch (Exception ex) {
            ex.printStackTrace();
            Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
        }
        return kullaniciid;
    }
}

package action;
import database.ConnManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.LoginBilgiBean;
import servlet.ActionMultiplexer;
public class PostLoginKontrol extends Action {

```



```

public int triggerActionint(HttpServletRequest request, HttpServletResponse response, Connection
conn) throws Exception {
    System.out.println("PostLoginKontrol");
    int kullanicIID = 0;
    boolean kullanicikarsilikvar = false;
    try {
        // Connection conn;2q345466
        // conn = new ConnManager().connectionOlustur();
        String sqlstatement = "SELECT ANONYMOUS.\"kullanicilar \".\"kullanicIID\" FROM
ANONYMOUS.\"kullanicilar \" WHERE ANONYMOUS.\"kullanicilar \".\"kullanicIadi\" = ? AND
ANONYMOUS.\"kullanicilar \".\"sifre\" = ? ";
        PreparedStatement pst = conn.prepareStatement(sqlstatement);
        LoginBilgiBean loginbilgi = (LoginBilgiBean) request.getSession().getAttribute("loginbilgi");
        pst.setString(1, loginbilgi.getKullaniciAdi());
        pst.setString(2, loginbilgi.getSifre());
        ResultSet rs = pst.executeQuery();
        while (rs.next()) {
            kullanicIID = rs.getInt("kullanicIID");
        }
        if (kullanicIID==1) {
            String alertmesaj = "Kullanıcı adı/ şifresi hatalı. /n Lütfen sisteme tekrar giriş yapınız";
            request.getSession().setAttribute("jsalertmetin", alertmesaj);
            request.getSession(true).getServletContext().getRequestDispatcher("/websayfalar/login.jsp");
        }
        // conn.close();
    } catch (Exception ex) {
        ex.printStackTrace();
        Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
    }
    return kullanicIID;
}
}

```

SessionAc

```

package action;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import servlet.ActionMultiplexer;
import ortakislemeler.SessionIDGenerator;
import ortakislemeler.IstemciIPAI;
import ortakislemeler.TimeStampOlustur;
public class SessionAc extends Action {
    public void triggerAction(HttpServletRequest request, HttpServletResponse response, Connection
conn) throws Exception {
        System.out.println("sesssion ac");
        int kullanicIID = 0;
        try {
            kullanicIID = new PostLoginKontrol().triggerActionint(request, response, conn);
            System.out.println("kullanicIID====="+kullanicIID);
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (kullanicIID > 0) {
            String sessionId = new SessionIDGenerator().yeniSessionId(15);
            String kullanicIP = new IstemciIPAI().ClientIPsi(request, response);
            String timestamp = new TimeStampOlustur().timeStampgetir();
            try {
                // Connection conn;
                // conn = new ConnManager().connectionOlustur();
                String sqlstatement = "INSERT INTO \"ANONYMOUS\".\"sessions \" VALUES (?, ?, ?, ?)";
                // INSERT INTO "ANONYMOUS"." sessions " VALUES ('1', 'rtyrtyrtyrty', '57576', '10.4.4.4');
                PreparedStatement pst = conn.prepareStatement(sqlstatement);
                pst.setInt(1, kullanicIID);
                pst.setString(2, sessionId);
                pst.setString(3, timestamp);
                pst.setString(4, kullanicIP);
            } catch (SQLException ex) {
                ex.printStackTrace();
                Logger.getLogger(ActionMultiplexer.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}

```

```

}
package action;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import servlet.ActionMultiplexer;
import ortakislemeler.SessionIDGenerator;
import ortakislemeler.IstemciIPAI;
import ortakislemeler.TimeStampOlustur;
    int kullanicIID = 0;
    try {
        kullanicIID = new PostLoginKontrol().triggerActionint(request, response, conn);
        System.out.println("kullanicIID====="+kullanicIID);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Database PAKETİ CLASSLARI

ConnManager

```

package database;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.InitialContext;
import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.sql.DataSource;
public class ConnManager {
....
....
....
}

```

....
....
....

```
private DataSource datasource = null;
public static ConnManager getConnMan(HttpServletRequest req) {
    ConnManager connmanager = null;
    ServletContext servletContext=req.getSession(true).getServletContext();
    if (servletContext.getAttribute(DSNAME) == null) {
        ConnManager newConMan = new ConnManager();
        newConMan.setDataSource(servletContext);
        connmanager = newConMan;
    } else {
        connmanager = (ConnManager) servletContext.getAttribute(DSNAME);
    }
    return connmanager;
}
private void setDataSource(ServletContext servletContext) {
    try {
        if (INITIALCONTEXT == null) {
            INITIALCONTEXT = new javax.naming.InitialContext();
        }
        datasource = (javax.sql.DataSource) INITIALCONTEXT.lookup(DSNAME);
        servletContext.setAttribute(DSNAME, this);
        System.out.println("pool instance created in servlet config for dsname= : " + DSNAME);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
public ResultSet resultSetDondur(Connection con,String sqlstring) {
    ResultSet rs = null;
    try {
        Statement st2 = con.createStatement();
        rs = st2.executeQuery(sqlstring);
        //con.close();
    } catch (Exception e) {
        System.out.println("Veritabanýna bađlantý sađlanamadý" + e);
    }
    return rs;
}
```

////////////////////////////////////

```

public static java.sql.Connection connectionOlustur(){
Connection conn = null;
    try {
        DriverManager.registerDriver(new oracle.jdbc.OracleDriver());
        conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","ANONYMOUS","m1a2k3");

    } catch (SQLException ex) {
        ex.printStackTrace();
        Logger.getLogger(ConnManager.class.getName()).log(Level.SEVERE, null, ex);
    }
    return conn;
}

public java.sql.Connection getConn() {
    System.out.println(" gettt connection =");
    Connection connection = null;
    try {
        connection = datasource.getConnection();
        if (!connection.getAutoCommit())
            connection.setAutoCommit(true);
        System.out.println(" connection = datasource.getConnection");
    } catch (Exception e) {
        System.err.println("[HATA] --> con create fail for : " + DSNAME);
        connection = null;
        e.printStackTrace();
    }
    return connection;
}

public static void close(Connection con) {
    try {
        if (con != null)
            con.close();
    } catch (SQLException e) {
        System.err.println(e.getMessage());
    }
}

public static void close(Statement stmt) {
    try {
        if (stmt != null)
            stmt.close();
    }
}

```

```

    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

public static void close(PreparedStatement pstmt) {
    try {
        if (pstmt != null)
            pstmt.close();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

public static void close(ResultSet rs) {
    try {
        if (rs != null)
            rs.close();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

public static void close(ResultSet rs, Statement stmt) {
    close(rs);
    close(stmt);
}

public static void close(ResultSet rs, PreparedStatement pstmt) {
    close(rs);
    close(pstmt);
}

/**
 * @param con
 * @param b
 */
public static void setAutoCommit(Connection con, boolean b) {
    try {
        con.setAutoCommit(b);
    } catch (SQLException e) {
        System.err.println(e.getMessage());
    }
}

public static void commit(Connection con) {

```

```

try {
    con.commit();
} catch (SQLException e) {
    System.err.println(e.getMessage());
}
}
public static void rollback(Connection con) {
    try {
        con.rollback();
    } catch (SQLException e) {
        System.err.println(e.getMessage());
    }
}
public static void beginTransaction(Connection con) {
    setAutoCommit(con, true);
}
public static void endTransaction(Connection con) {
    commit(con);
    setAutoCommit(con, false);
}
}

```

Exceptions PAKETİ CLASSLARI

ApplicationException

```

package exceptions;
import java.text.MessageFormat;
public class ApplicationException extends Exception {
    private String hataKodu;
    public ApplicationException() {
    }
    public ApplicationException(String kodu,String mesaj) {
        super(mesaj);
        this.hataKodu = kodu;
    }
    public ApplicationException(Exception ex) {
        super(formatExceptionMsg(ApplicationExceptionMessages.genel_1001.getMessage(),
ex.getMessage()));
        hataKodu = ApplicationExceptionMessages.genel_1001.getKodu();
    }
}

```

```

public ApplicationException(ApplicationExceptionMessages exceptionMessages, Object[]
args) {
    super(formatExceptionMsg(exceptionMessages.getMessage(), args));
    hataKodu = exceptionMessages.getKodu();
}
public ApplicationException(ApplicationExceptionMessages exceptionMessages, Object arg) {
    super(formatExceptionMsg(exceptionMessages.getMessage(), arg));
    hataKodu = exceptionMessages.getKodu();
}
public ApplicationException(ApplicationExceptionMessages exceptionMessages) {
    super(exceptionMessages.getMessage());
    hataKodu = exceptionMessages.getKodu();
}
private static String formatExceptionMsg(String messageString, Object[] args) {
    MessageFormat mf = new MessageFormat(messageString);
    for (int i = 0; i < args.length; i++) {
        args[i] = String.valueOf(args[i]);
    }
    return mf.format(args);
}
private static String formatExceptionMsg(String messageString, Object arg) {
    MessageFormat mf = new MessageFormat(messageString);
    Object[] args = new Object[1];
    args[0] = arg;
    return mf.format(args);
}
public String getHataKodu() {
    return hataKodu;
}
public void setHataKodu(String i) {
    hataKodu = i;
}
}

```

ApplicationExceptionMessages

```

package exceptions;
import java.io.Serializable;
public class ApplicationExceptionMessages implements Serializable {
    private String kodu;
    private String message;
}

```



```

public ApplicationExceptionMessages(String yeniKodu, String yeniMsg) {
    setKodu(yeniKodu);
    setMessage(yeniMsg);
}

public static final ApplicationExceptionMessages genel_1001 = new
ApplicationExceptionMessages("G0001", "Sistem hatası! -> {0}");
    public static final ApplicationExceptionMessages kullanıcı_1001 = new
ApplicationExceptionMessages("K0001", "Girilen kullanıcı bilgisi sistemde bulunamadı!");
    public static final ApplicationExceptionMessages parametre_0001 = new
ApplicationExceptionMessages("P0001", "{0} plakalı ilin {1} kodlu ilçesi sistemde bulunamadı!");
    public static final ApplicationExceptionMessages parametre_0002 = new
ApplicationExceptionMessages("P0002", "{0} başlangıç tarihi için geçerli olan ipc asgari azami
parametresi sistemde bulunamadı!");
    public static final ApplicationExceptionMessages parametre_0003 = new
ApplicationExceptionMessages("P0003", "{0} başlangıç tarihi için geçerli olan ipc asgari azami oran
parametresi sistemde bulunamadı!");
    public static final ApplicationExceptionMessages sql_1001 = new
ApplicationExceptionMessages("S0001", "Bu kayıt sistemde zaten mevcut!");
    public static final ApplicationExceptionMessages sql_1002 = new
ApplicationExceptionMessages("S0002", "Veritabanı hatası! -> {0}");
    public static final ApplicationExceptionMessages sql_1003 = new
ApplicationExceptionMessages("S0003", "Identity kolonu geri okumak için insert işleminden sonra
connection'ı kapatma!"); //sqlexception
    public static final ApplicationExceptionMessages sql_1004 = new
ApplicationExceptionMessages("S0004", "Veri Kaynağı hatası!");
    public static final ApplicationExceptionMessages sql_1005 = new
ApplicationExceptionMessages("S0005", "Veri Kaynağı bulunamadı!");
    public static final ApplicationExceptionMessages sql_1006 = new
ApplicationExceptionMessages("S0006", "Silme işlemi istediğiniz kayıtlarla ilişkili kayıtlar olduğundan,
silinemez!");
    public static final ApplicationExceptionMessages sql_1007 = new
ApplicationExceptionMessages("S0007", "Veri Dönüşüm Hatası:{{0}}");

public String getKodu() {
    return kodu;
}

/**
 * @return
 */

public String getMessage() {
    return message;
}

```

```

    }
    /**
     * @param i
     */
    public void setKodu(String i) {
        kodu = i;
    }
    /**
     * @param string
     */
    public void setMessage(String string) {
        message = string;
    }
}

```

Ortakislemeler PAKETİ CLASSLARI

IstemcilPAI

```

package ortakislemeler;
import action.Action;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class IstemcilPAI extends Action {
    public String ClientIPsi(HttpServletRequest request, HttpServletResponse response) throws
Exception {
        String ipAddress = request.getHeader("X-FORWARDED-FOR");
        if (ipAddress == null) {
            ipAddress = request.getRemoteAddr();
        }
        return ipAddress;
    }
}

```

Ortakislemeler

```

package ortakislemeler;
import java.util.Calendar;
import java.util.StringTokenizer;
public class Ortakislemeler {
    /**
     * Ortakislemeler constructor comment.

```

```

*/
public OrtakIslemler() {
    super();
}
public void sifirTemizle() {}
public String sifirTemizle(String gelenDeger) {
    String donenDeger = "";
    int uzunluk;
    uzunluk = gelenDeger.length();
    for (int i = 0; i < uzunluk; i++) {
        if (!(gelenDeger.substring(i, i+1).equals("0"))) {
            donenDeger = gelenDeger.substring(i, uzunluk);
            i = uzunluk;
        }
    }
    return donenDeger.trim();
}
public String virgulEkle(String gelenDeger) {
    String donenDeger = "";
    int uzunluk;
    uzunluk = gelenDeger.length();
    for (int i = uzunluk - 1; i >= 0; i--) {
        if (((((uzunluk - 1 - i) % 3) == 0))
            && !(i == uzunluk - 1))) {donenDeger = "," + donenDeger;
        }
        donenDeger = gelenDeger.substring(i, i + 1) + donenDeger;
    }
    return donenDeger;
}
public String noktaEkle(String gelenDeger) {
    String donenDeger = "";
    int uzunluk;
    uzunluk = gelenDeger.length();
    for (int i = uzunluk - 1; i >= 0; i--) {
        if (((((uzunluk - 1 - i) % 3) == 0))
            && !(i == uzunluk - 1))) {donenDeger = "." + donenDeger;
        }
        donenDeger = gelenDeger.substring(i, i + 1) + donenDeger;
    }
}

```

```

        return donenDeger;
    }
    public String virgulTemizle(String gelenDeger) {
        String aTmpdcstr="";
        int aLen=gelenDeger.length();
        for (int i = 0 ; i < aLen ; i++)
        {
            if (!(gelenDeger.substring(i,i+1).equalsIgnoreCase(",")))
                aTmpdcstr+=gelenDeger.substring(i,i+1);
        }
        return (aTmpdcstr);
    }
    public String bastakiSifirlariKaldir(String gelenDeger) {
        String aSonuc="";
        String aTutGelen="";
        int aLen=gelenDeger.length();
        for (int i = 0 ; i < aLen ; i++)
        {
            if (!(gelenDeger.substring(i,i+1).equalsIgnoreCase("0")))
            {
                aSonuc = gelenDeger.substring(i,aLen);
                return (aSonuc);
            }
        }
        return (aSonuc);
    }
    public static String strChk(String aStr){
        // jsplerde form nesnelere String ile gelen "null" gozukmemesi için
        return (aStr==null?"":aStr);
    }
    public static String strTrim(String aStr){
        // kontrol
        return (strChk(aStr).trim());
    }
    public static String degistir(String inpstr, String ayiklaStr){
        //ayiklaStr = 1 uzunlukta olmalı
        StringBuffer asb = new StringBuffer("");
        for (int i=0; i<inpstr.length(); i++)
        {
            if(!(ayiklaStr.equals(inpstr.substring(i,i+1))))

```

```

        asb.append(inpstr.substring(i,i+1));
    }
    return asb.toString();
}

public static String replace2(String inpstr, String ayiklaStr, String koyStr){
    //ayiklaStr = 1 uzunlukta olmalı
    StringBuffer asb = new StringBuffer();
    int instrLength=0;
    if (inpstr!=null)
    {
        instrLength= inpstr.length();
    }
    for ( int i=0; i<instrLength; i++)
    {
        if(!(ayiklaStr.equals(inpstr.substring(i,i+1))))
            {asb.append(inpstr.substring(i,i+1));
            }
        else
            {asb.append(koyStr);
            }
    }
    return asb.toString();
}

public static boolean sayisalKontrol(String aStr)
// FALSE DONERSE SAYISAL OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = true;
    for (int i=0;i < aStr.length();i++)

        { if (!(
            ((aStr.charAt(i)) >= '0' ) &&
            ((aStr.charAt(i)) <= '9')
            ))
            flag = false;
        }
    return flag;
}

public static boolean sayisalYTLKontrol(String aStr)
// FALSE DONERSE SAYISAL OLMAYAN KARAKTERLER VAR DEMEKTIR.
{
    //System.out.println("SAYISAL KONROL YAPILACAK");
    boolean flag = true;

```

```

int noktayer = 0;
int lenStr = 0;
int sagsayi = 0;
for (int i = 0; i < aStr.length(); i++) {
    lenStr = lenStr + 1;
    if (!(((aStr.charAt(i)) >= '0') && ((aStr.charAt(i)) <= '9')) {

        if (aStr.charAt(i) == '.') {
            if (noktayer > 0) {
                flag = false;
            }
            noktayer = lenStr;
        } else {
            flag = false;
        }
    }
}
if (flag) {
    if (noktayer == 0 ) {
        flag = false;
    } else {
        sagsayi = lenStr - noktayer;
    }
}
if (!(sagsayi == 2)) {
    flag = false;
}
}
return flag;
}

```

```

public static boolean isAlphabetical(String aStr)
// FALSE DONERSE ALFABETIK OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = true;
    String kStr = "1234567890<>+~*!{}&%^$%#|@?!";
    for (int i=0;i < aStr.length();i++)
    {
        for (int j=0;j < kStr.length();j++)
        {
            if (aStr.charAt(i) == kStr.charAt(j) ) {flag = false;}
        }
    }
}

```

```

    }
    return flag;
}
public static boolean isEnglishAlphabetical(String aStr)
// FALSE DONERSE ALFABETIK OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = true;
    String kStr = "ÖöÇçPpýÝöĐÜü1234567890<>+~*/[{}]&%^$½#|@?!";
    for (int i=0;i < aStr.length();i++)
    {
        for (int j=0;j < kStr.length();j++)
        {
            if (aStr.charAt(i) == kStr.charAt(j) ) {flag = false;}
        }
    }
    return flag;
}
public static boolean seriKontrol(String aStr)
// FALSE DONERSE ALFABETIK OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = false;
    String kStr = "abcdefghijklmnoprstuvwxyzABCDEFGHIJKLMNoprstuvwxyz";
    for (int i=0;i < aStr.length();i++)
    {
        flag = false;
        for (int j=0;j < kStr.length();j++)
        {
            if (aStr.charAt(i) == kStr.charAt(j) ) {flag = true;}
        }
        if (!flag){return flag;}
    }
    return flag;
}
public static boolean seriKontrolTurkce(String aStr)
// FALSE DONERSE ALFABETIK OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = false;
    String kStr = " abcçdefgđhýijklmnoöprsṗtuüvyzABCÇDEFGĐHIÝJKLMNOÖPRSṖTUÜVYZ";
    for (int i=0;i < aStr.length();i++)
    {
        flag = false;
        for (int j=0;j < kStr.length();j++)
        {

```

```

        if (aStr.charAt(i) == kStr.charAt(j) ) {flag = true;}
    }
    if (!flag){return flag;}
}
return flag;
}

public static boolean isEnglishAlphabeticalChar(String aStr)
// FALSE DONERSE ALFABETIK OLMAYAN KARAKTERLER VAR DEMEKTIR.
{boolean flag = true;
    String kStr = "ÖöÇçPpýÝöĐ<>+~*!{}&%^$'½#@?!";
    for (int i=0;i < aStr.length();i++)
    {
        for (int j=0;j < kStr.length();j++)
        {
            if (aStr.charAt(i) == kStr.charAt(j) ) {flag = false;}
        }
    }
    return flag;
}

public static String isValidDate(String aStr)
{
    //astr gg.aa.yyyy
    String aReturnStr = null;
    if (!(aStr.trim().length()==10)){return aReturnStr;}
    String aDay,aMonth,aYear;
    Calendar aCal= Calendar.getInstance();
    aCal.clear();
    Calendar aCalToday= Calendar.getInstance();
    StringTokenizer aStringTokenizer= new StringTokenizer(aStr);
    String aDelimiter=null;
    try
    {
        if ( aStr.indexOf(".") > 0 )
            aDelimiter=".";
        else if ( aStr.indexOf("/") > 0 )
            aDelimiter="/";
        if (null!=aDelimiter)
            {
                aDay=
aStringTokenizer.nextToken(aDelimiter);

```



```

aMonth=
aStringTokenizer.nextToken(aDelimiter);

aYear=
aStringTokenizer.nextToken(aDelimiter);

aCal.set(Integer.parseInt(aYear),Integer.parseInt(aMonth)-1,Integer.parseInt(aDay));
        if (
(aCal.get(Calendar.DAY_OF_MONTH)==Integer.parseInt(aDay)) &&
(aCal.get(Calendar.MONTH)==(Integer.parseInt(aMonth)-1)) &&
(aCal.get(Calendar.YEAR)==Integer.parseInt(aYear)) &&
        (!( aCal.after(aCalToday) ))
        )
        { aReturnStr = aDay + "."+ aMonth + "."+
aYear;
        }
    }
} catch (Exception e){
    } finally {
    }
return aReturnStr;
}

public static String isDateDiffOk(String kckTarih,String bykTarih,int dateFark)
{
int gnk = Integer.parseInt(kckTarih.substring(0,2));
int ayk = Integer.parseInt(kckTarih.substring(3,5))-1;
int ylk = Integer.parseInt(kckTarih.substring(6,10));

int gnb = Integer.parseInt(bykTarih .substring(0,2));
int ayb = Integer.parseInt(bykTarih .substring(3,5))-1;
int ylb = Integer.parseInt(bykTarih .substring(6,10));
java.util.Calendar c1 = java.util.Calendar.getInstance();
java.util.Calendar c2 = java.util.Calendar.getInstance();
c1.set(ylk,ayk,gnk);
c1.add(java.util.Calendar.DAY_OF_MONTH,dateFark);
c2.set(ylb,ayb,gnb);
if (c1.before(c2)) {

```

```

        return "nok";
    } else {
        return "ok";
    }
}

public static String isDateOrderOk(String kckTarih,String bykTarih)
{
    int gnk = Integer.parseInt(kckTarih.substring(0,2));
    int ayk = Integer.parseInt(kckTarih.substring(3,5))-1;
    int ylk = Integer.parseInt(kckTarih.substring(6,10));

    int gnb = Integer.parseInt(bykTarih .substring(0,2));
    int ayb = Integer.parseInt(bykTarih .substring(3,5))-1;
    int ylb = Integer.parseInt(bykTarih .substring(6,10));

    java.util.Calendar c1 = java.util.Calendar.getInstance();
    java.util.Calendar c2 = java.util.Calendar.getInstance();
    c1.set(ylk,ayk,gnk);
    c2.set(ylb,ayb,gnb);
    if (c1.after(c2)) {
        return "nok";
    } else {
        return "ok";
    }
}

public static String isDateGreaterToday(String aStr)
{
    //astr gg.aa.yyyy
    String aReturnStr = null;
    String aDay,aMonth,aYear;
    Calendar aCal= Calendar.getInstance();
    aCal.clear();
    Calendar aCalToday= Calendar.getInstance();
    StringTokenizer aStringTokenizer= new StringTokenizer(aStr);
    String aDelimiter=null;
    try
    {
        if ( aStr.indexOf(".") > 0 )
            aDelimiter=".";
        else if ( aStr.indexOf("/") > 0 )
            aDelimiter="/";
    }
}

```

```

        if (null!=aDeliminator)
        {
            aDay= aStringTokenizer.nextToken(aDeliminator);
            aMonth= aStringTokenizer.nextToken(aDeliminator);
            aYear= aStringTokenizer.nextToken(aDeliminator);
            aCal.set(Integer.parseInt(aYear),Integer.parseInt(aMonth)-1,Integer.parseInt(aDay));
            if (!( (aCal.get(Calendar.DAY_OF_MONTH)==Integer.parseInt(aDay)) &&
                (aCal.get(Calendar.MONTH)==(Integer.parseInt(aMonth)-1)) &&
                (aCal.get(Calendar.YEAR)==Integer.parseInt(aYear)) &&
                (!( aCal.after(aCalToday) ))
                ))
            {
                aReturnStr = "Tarih Bugünden büyük olamaz";
            }
        }
    } catch (Exception e){
    } finally {
    }
    return aReturnStr;
}

// verilen Label ve Value stringi consol a yazar.
public static void sout(String aLabel, String aValue){
//System.out.println("-----*" + aLabel +"*-->" +aValue);
}

public static void sout(String aStr){
//System.out.println(aStr);
}

public static String BasaSifirEkle(String gelenDeger,int kontrolDeger) {
    gelenDeger = gelenDeger.trim();
    int len = gelenDeger.length();
    if (len < kontrolDeger)
        for ( int i =len;i<kontrolDeger;i++)
            gelenDeger = "0"+gelenDeger;
    return gelenDeger;
}

public static String BasaBoslukEkle(String gelenDeger,int kontrolDeger) {
    int len = gelenDeger.length();
    if (len < kontrolDeger)
        for ( int i =len;i<kontrolDeger;i++)
            gelenDeger = " "+gelenDeger;
    return gelenDeger;
}
}

```

```

public static String SonaBoslukEkle(String gelenDeger,int kontrolDeger) {
    int len = gelenDeger.length();
    if (len < kontrolDeger)
        for ( int i =len;i<kontrolDeger;i++)
            gelenDeger = gelenDeger+" ";
    return gelenDeger;
}

    public static String getTodayTime(String delimiter)
    {
        Calendar aCal = Calendar.getInstance();
        String aH=    String.valueOf( aCal.get(Calendar.HOUR_OF_DAY));
        String aM=    String.valueOf( aCal.get(Calendar.MINUTE));
        String aS=    String.valueOf( aCal.get(Calendar.SECOND));
        if (aH.length()<2)aH="0"+aH;
        if (aM.length()<2)aM="0"+aM;
        if (aS.length()<2)aS="0"+aS;
        if (delimiter.equals("bos")){
            return ( aH + aM + aS );
        }else{
            return ( aH + ":"+ aM + ":"+ aS );
        }
    }
}

public static boolean tarihKarsilastir(String küçükTarih,String büyükTarih) {
    boolean dogru = true;
    int küçükgun = Integer.parseInt(küçükTarih.substring(0,2));
    int küçükay = Integer.parseInt(küçükTarih.substring(3,5));
    int küçükyil = Integer.parseInt(küçükTarih.substring(6,10));
    int büyükgun = Integer.parseInt(büyükTarih.substring(0,2));
    int büyükay = Integer.parseInt(büyükTarih.substring(3,5));
    int büyükyil = Integer.parseInt(büyükTarih.substring(6,10));
    if (küçükay > büyükay ) {
        dogru = false;
    } else {
        if (küçükay == büyükay && küçükgun > büyükgun) {
            dogru = false;
        } else {
            if (küçükay == büyükay && küçükgun == büyükgun && küçükgun > büyükgun) {
                dogru = false;
            }
        }
    }
}

```

```

    }
    return dogru;
}
// 101 bosmu
/*
String degiskenAdi,
String checkType,
int numParam1,
int numParam2,
*/
public static boolean genelCheck(String checkType,String degiskenAdi,int numParam1,long
numParam2) {
    boolean myGenelCheck = true;
    degiskenAdi=degiskenAdi.trim();
    //bosmu
    if (checkType.equals("101")){
        if (degiskenAdi.equals("")) {
            myGenelCheck = false;
        }
    }
    //sayısal mi
    if (checkType.equals("201")){
        myGenelCheck = sayisalKontrol(degiskenAdi);
    }
    //0 dan büyük mu
    if (checkType.equals("202")){
        if (!(degiskenAdi.equals(""))){
            if (Long.parseLong(degiskenAdi)==0){
                myGenelCheck = false;
            }
        }
    }
    //Alfabetik mi
    if (checkType.equals("301")){
        myGenelCheck = isAlphabetical(degiskenAdi);
    }
    //yanlis tarih formati
    if (checkType.equals("401")){
        String sTarih = "";
        sTarih =isDateValid(degiskenAdi);
    }
}

```

```

if (sTarih==null) {
    sTarih =isDateGreaterToday(degiskenAdi);
    if (sTarih==null) {
        myGenelCheck = false;
    }
}
}
//tarih bugunden buyukmu
if (checkType.equals("402")){
    String sTarih = "";
    sTarih =isDateValid(degiskenAdi);
    if (sTarih==null) {
        sTarih =isDateGreaterToday(degiskenAdi);
        if (!(sTarih==null)) {
            myGenelCheck = false;
        }
    }
}
myGenelCheck=!myGenelCheck;
return myGenelCheck;
}

public static String sDuzelt(String instr) {
    String cevap = null;
    try {
        cevap = instr.replace("\", ' ');
    }
    catch (Exception e){
        cevap = instr;
    }

    return cevap;
}

public static String mDateFunctionIsInvalid(String kDate,String ayraca,String ayracb) {
    String retValue="";
    String mDate = kDate.trim();
    if (retValue.equals("")) {
        if (!(mDate.length()==10)) {
            retValue="Geçersiz Tarih 1";
            return retValue;
        }
    }
}

```

```

}
System.out.println("basla");
if (retValue.equals("")) {
    String ayrac1 = mDate.substring(2,3);
    String ayrac2 = mDate.substring(5,6);
    if (!(ayrac1.equals(ayraca) || ayrac1.equals(ayracb))) {
        retValue="Geçersiz Tarih 2";
        return retValue;
    }
    if (!(ayrac2.equals(ayraca) || ayrac2.equals(ayracb))) {
        retValue="Geçersiz Tarih 3";
        return retValue;
    }
}
boolean syslFlg = false; String sayisalmikntrl =null;long sysdegkntnr=0;
if (retValue.equals("")) {
    sayisalmikntrl =mDate.substring(0,2);syslFlg = false;try {sysdegkntnr =
Long.parseLong(sayisalmikntrl);syslFlg = true;} catch (Exception ex) {}
    if (syslFlg) {
        if (sysdegkntnr==0){
            retValue="Geçersiz Tarih";
            return retValue;
        }
    } else {
        retValue="Geçersiz Tarih 4";
        return retValue;
    }
}
if (retValue.equals("")) {
    sayisalmikntrl =mDate.substring(3,5);syslFlg = false;try {sysdegkntnr =
Long.parseLong(sayisalmikntrl);syslFlg = true;} catch (Exception ex) {}
    if (syslFlg) {
        if (sysdegkntnr==0){
            retValue="Geçersiz Tarih";
            return retValue;
        }
    } else {
        retValue="Geçersiz Tarih 4";
        return retValue;
    }
}

```

```

    }
    if (retValue.equals("")) {
        sayisalmikntrl =mDate.substring(6,10);syslFlg = false;try {sysdegkntnr =
Long.parseLong(sayisalmikntrl);syslFlg = true;} catch (Exception ex) {}
        if (syslFlg) {
            if (sysdegkntnr==0){
                retValue="Geçersiz Tarih";
                return retValue;
            }
        } else {
            retValue="Geçersiz Tarih 4";
            return retValue;
        }
    }
    int dateGun=Integer.parseInt(mDate.substring(0,2));
    int dateAy=Integer.parseInt(mDate.substring(3,5));
    int dateYil=Integer.parseInt(mDate.substring(6,10));
    if (dateAy>12) {
        retValue="Geçersiz Tarih 10";
        return retValue;
    }
    int aylarMax[] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
    if ((dateYil % 4)==0){ aylarMax[2]=29;}
    if (dateGun>aylarMax[dateAy]) {
        retValue="Geçersiz Tarih 10";
        return retValue;
    }
    return retValue;
}
public static int mDateFunctionBuyukluk(String tr1,String tr2) {
    int retValue = 0;
    if (OrtakIslemler.mDateFunctionIsValid(tr1,".","/").equals("")) {
        if (!OrtakIslemler.mDateFunctionIsValid(tr2,".","/").equals("")) {
            retValue = 0;
            return retValue;
        }
    } else {
        retValue = 0;
        return retValue;
    }
}

```



```

String gun1 = tr1.substring(0,2);
String ay1 = tr1.substring(3,5);
String yil1 = tr1.substring(6,10);
long tarihSayisal1 = Long.parseLong(yil1) *
10000+Long.parseLong(ay1)*100+Long.parseLong(gun1);
String gun2 = tr2.substring(0,2);
String ay2 = tr2.substring(3,5);
String yil2 = tr2.substring(6,10);
long tarihSayisal2 = Long.parseLong(yil2) *
10000+Long.parseLong(ay2)*100+Long.parseLong(gun2);
if (tarihSayisal1>tarihSayisal2) {
    retValue = 1;
    return retValue;
} else {
    if (tarihSayisal1<tarihSayisal2) {
        retValue = 2;
        return retValue;
    } else {
        if (tarihSayisal1==tarihSayisal2) {
            retValue = 3;
            return retValue;
        }
    }
}
return retValue;
}

public static String mDateFunctionGetToday() {
    String retValue = "";
    java.util.Calendar cb = java.util.Calendar.getInstance();
    String bugun = Integer.toString(cb.get(java.util.Calendar.DATE));
    String buay = Integer.toString(cb.get(java.util.Calendar.MONTH) + 1);
    String buyil = Integer.toString(cb.get(java.util.Calendar.YEAR));
    if (bugun.length(<2){bugun="0"+bugun;}
    if (buay.length(<2){buay="0"+buay;}
    retValue = bugun+"/"+buay+"/"+buyil;
    return retValue;
}

public static String mDateFunctionAddYear(String tr1,int kacyil) {
    return tr1.substring(6,10)+Integer.toString(Integer.parseInt(tr1.substring(6,10))+kacyil);
}

```

```

public static String mDateFunctionSubYear(String tr1,int kacYil) {
    if (Integer.parseInt(tr1.substring(6,10))>kacYil){
        return tr1.substring(0,6)+Integer.toString(Integer.parseInt(tr1.substring(6,10))-kacYil);
    } else {
        return "";
    }
}

public static String mDateFunctionAddMonth(String tr1,int kacAy) {
    String retVal = "";
    int toplamAy = kacAy+Integer.parseInt(tr1.substring(3,5));
    int sonucGun = Integer.parseInt(tr1.substring(0,2));
    int sonucAy = toplamAy % 12;
    int sonucYil = Integer.parseInt(tr1.substring(6,10))+(toplamAy - sonucAy) / 12;
    if (sonucAy==0){sonucAy=12;sonucYil=sonucYil-1;}
    String bugun = Integer.toString(sonucGun);
    String buay = Integer.toString(sonucAy);
    String buyil = Integer.toString(sonucYil);
    if (bugun.length(<2){bugun="0"+bugun;}
    if (buay.length(<2){buay="0"+buay;}
    retVal = bugun+"/"+buay+"/"+buyil;
    return retVal;
}

public static String mDateFunctionAddOneMonth(String tr1) {
    String retVal = "";
    int toplamAy = 1+Integer.parseInt(tr1.substring(3,5));
    int sonucGun = Integer.parseInt(tr1.substring(0,2));
    int sonucAy = toplamAy % 12;
    int sonucYil = Integer.parseInt(tr1.substring(6,10))+(toplamAy - sonucAy) / 12;
    if (sonucAy==0){sonucAy=12;sonucYil=sonucYil-1;}
    String bugun = Integer.toString(sonucGun);
    String buay = Integer.toString(sonucAy);
    String buyil = Integer.toString(sonucYil);
    if (bugun.length(<2){bugun="0"+bugun;}
    if (buay.length(<2){buay="0"+buay;}
    retVal = bugun+"/"+buay+"/"+buyil;
    return retVal;
}

public static String mDateFunctionSubOneMonth(String tr1) {
    String retVal = "";
    int sonucAy = Integer.parseInt(tr1.substring(3,5))-1;

```

```

        int sonucYil = Integer.parseInt(tr1.substring(6, 10))-1;
        if (sonucAy==0){sonucAy=12;sonucYil=sonucYil-1;}
        String buay = Integer.toString(sonucAy);
        String buyil = Integer.toString(sonucYil);
        if (buay.length(<2){buay="0"+buay;}
        retValue = tr1.substring(0,2)+"/"+buay+"/"+buyil;
        return retValue;
    }
}

```

ParametrelerArraye

```

package ortakislemler;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author MAK
 */
public class ParametrelerArraye {
    public Map parametrelerArraye(HttpServletRequest request, HttpServletResponse response)
throws Exception {
        ArrayList arraylist = new ArrayList();
        Map<String, String[]> m = request.getParameterMap();
        return m;
    }
}

```

RenkBelirle

```

package ortakislemler;
import java.sql.Connection;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author MAK

```

```

*/
public class RenkBelirle {
    String renk0 = "#FFFFFF";
    String renk1 = "#EFF8FB";
    String renk2 = "#CEE5F5";
    String renk3 = "#A9E2F3";
    String renk4 = "#81DAF5";
    String renk5 = "#58D3F7";
    String renk6 = "#2ECCFA";
    String renk7 = "#2ECCFA";
    String renk8 = "#00BFFF";
    String renk9 = "#01A9DB";
    String renk10 = "#0080FF";

    public synchronized String yuzderenkBelirle(HttpServletRequest request, HttpServletResponse
response, float yuzde) throws Exception {
        String renkkodu = "";
        if (yuzde == 0) {
            renkkodu = renk0;
        }
        if (yuzde > 0 && yuzde < 11) {
            renkkodu = renk1;
        }
        if (yuzde > 10 && yuzde < 21) {
            renkkodu = renk2;
        }
        if (yuzde > 20 && yuzde < 31) {
            renkkodu = renk3;
        }
        if (yuzde > 30 && yuzde < 41) {
            renkkodu = renk4;
        }
        if (yuzde > 40 && yuzde < 51) {
            renkkodu = renk5;
        }
        if (yuzde > 50 && yuzde < 61) {
            renkkodu = renk6;
        }
        if (yuzde > 60 && yuzde < 71) {
            renkkodu = renk7;
        }
    }
}

```

```

    if (yuzde > 70 && yuzde < 81) {
        renkkodu = renk8;
    }
    if (yuzde > 80 && yuzde < 91) {
        renkkodu = renk9;
    }
    if (yuzde > 90) {
        renkkodu = renk10;
    }
    return renkkodu;
}
}

```

SessionIDGenerator

```

package ortakislemler;
import java.security.SecureRandom;
import java.util.Random;
public final class SessionIDGenerator {
    char[] characterSet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789".toCharArray();
    public String yeniSessionId(int uzunluk) {
        Random random = new SecureRandom();
        char[] result = new char[uzunluk];
        for (int i = 0; i < result.length; i++) {
            // picks a random index out of character set > random character
            int randomCharIndex = random.nextInt(characterSet.length);
            result[i] = characterSet[randomCharIndex];
        }
        return new String(result);
    }
}

```

TimeStampOlustur

```

package ortakislemler;
import java.sql.Timestamp;
import java.util.Date;
public class TimeStampOlustur {
    public String timeStampgetir() throws Exception {
        String timestampstr = "";
        java.util.Date date = new java.util.Date();
        Timestamp timestamp = new Timestamp(date.getTime());
    }
}

```

```

        timestampstr = String.valueOf(timestamp);
        return timestampstr;
    }
}

```

Model PAKETİ CLASSLARI

HaritalBean

```

package model;

public class HaritalBean {
    private int ilkodu;
    private String degisken;
    private float ildeger;
    private float ilyuzde;
    private String ilrenk;
    /**
     * @return the ilkodu
     */
    public int getIlkodu() {
        return ilkodu;
    }
    /**
     * @param ilkodu the ilkodu to set
     */
    public void setIlkodu(int ilkodu) {
        this.ilkodu = ilkodu;
    }
    /**
     * @return the degisken
     */
    public String getDegisken() {
        return degisken;
    }
    /**
     * @param degisken the degisken to set
     */
    public void setDegisken(String degisken) {
        this.degisken = degisken;
    }
    /**
     * @return the ildeger

```

```

*/
public float getIldeger() {
    return ildeger;
}
/**
 * @param ildeger the ildeger to set
 */
public void setIldeger(float ildeger) {
    this.ildeger = ildeger;
}
/**
 * @return the ilyuzde
 */
public float getIlyuzde() {
    return ilyuzde;
}

```

Utilities PAKETİ CLASSLARI

UrlParametreler

```

package utilities;
import java.util.HashMap;
import java.util.Map;
public class UrlParametreler {
    public Map<String, String> querymapYap(String query) {
        String[] params = query.split("&");
        Map<String, String> map = new HashMap<String, String>();
        for (String param : params) {
            String name = param.split("=")[0];
            String value = param.split("=")[1];
            map.put(name, value);
        }
        return map;
    }
}

```

Util

```

package utilities;
import java.text.DecimalFormatSymbols;
import java.text.NumberFormat;
import java.util.Locale;
public class Util {

```

```

private final static String ytlpattern = "###.###,##";
private final static Locale YTLLocale = Locale.ITALY;
public static double ytl_parsedoublefromYTLLString(String str) throws Exception {
    double ad = 0.0d;
    if (str == null) { //ad =0.0d;
    } else if (str.length() == 0) { //ad =0.0d;
    } else {
        NumberFormat nf = NumberFormat.getInstance(YTLLocale);
        str = str.trim();
        ad = nf.parse(str).doubleValue();
        nf = null;
    }
    return ad;
}

public static String ytl_formatdbltoYTLLString(double ad, boolean isGroupingUsed) {
    ad = Math.round(ad * 100) / 100.0d;
    NumberFormat nf = NumberFormat.getInstance(YTLLocale);
    nf.setGroupingUsed(isGroupingUsed);
    nf.setMinimumFractionDigits(2);
    nf.setMaximumFractionDigits(2);
    String str = nf.format(ad);
    nf = null;
    return str;
}

public static String ytl_formatdblto_TL_String(double ad, boolean isGroupingUsed) {
    //ad = Math.round(ad * 100) / 100.0d;
    NumberFormat nf = NumberFormat.getInstance(YTLLocale);
    nf.setGroupingUsed(isGroupingUsed);
    nf.setMinimumFractionDigits(0);
    nf.setMaximumFractionDigits(0);
    String str = nf.format(ad);
    nf = null;
    return str;
}

public static String ytl_formatdbltoYTLLString(double ad, boolean isGroupingUsed, boolean
hideZero) {
    ad = Math.round(ad * 100) / 100.0d;
    String str = null;
    if ((ad == 0.0d) && (hideZero)) {
        str = "";
    }
}

```



```

    } else {
        NumberFormat nf = NumberFormat.getInstance(YTLLocale);
        nf.setGroupingUsed(isGroupingUsed);
        nf.setMinimumFractionDigits(2);
        nf.setMaximumFractionDigits(2);
        str = nf.format(ad);
        nf = null;
    }
    return str;
}

public static String ytl_formatdbltoYTLString(double ad, boolean isGroupingUsed, boolean
hideZero,boolean withFreeFractionLength) {
    //ad = Math.round(ad * 100) / 100.0d;
    String str = null;
    if ((ad == 0.0d) && (hideZero)) {
        str = "";
    } else {
        NumberFormat nf = NumberFormat.getInstance(YTLLocale);
        nf.setGroupingUsed(isGroupingUsed);
        if (!withFreeFractionLength)
        {nf.setMinimumFractionDigits(2);
        nf.setMaximumFractionDigits(2);
        }
        str = nf.format(ad);
        nf = null;
    }
    return str;
}

/**
 * stringi alir, noktali uzunluk kismini noktali olacak sekilde double yapar ve dondurur
 *
 * @param instr
 * @param noktaliuzunluk
 * @return
 */
public static double getDoubleFromStr(String instr, int noktaliuzunluk) {
    int tamuzunluk = instr.length() - noktaliuzunluk;
    double dbl = Double.parseDouble(instr.substring(0, tamuzunluk));
    dbl += Double.parseDouble("0." + instr.substring(tamuzunluk));
    return dbl;
}

```

```

}
/**
 * double degeri alir integer kismini soluna 0 ekleyerek istenen uzunlukta dondurur
 * @param dbl
 * @param uzunluk
 * @return
 */
public static String getTamsayiStr(double dbl, int uzunluk) {
    StringBuffer sb = new StringBuffer();
    long longdbl = (long) Math.floor(dbl);
    sb.append(longdbl);
    while (sb.length() < uzunluk) {
        sb.insert(0, "0");
    }
    return sb.toString();
}
/**
 * double degeri alir ondalık kisminin sađyna gerekirse 0 ekleyerek 6 uzunlukta
 * @param dbl
 * @param uzunluk
 * @return
 */
// public static String getOndsayiStr(double dbl) {
//     StringBuffer sb = new StringBuffer();
//     double tamsayidbl = Math.floor(dbl);
//     long noktaliKisim = (long) Math.floor((dbl - tamsayidbl) * 1000000);
//     sb.append(noktaliKisim);
//     return sb.toString();
// }
// public static double getOndsayiDbI(double dbl) {
//     double tamsayidbl = Math.floor(dbl);
//     double noktaliKisim = dbl - tamsayidbl;
//     return noktaliKisim;
// }
public static boolean is2digit(double dbl) {
    String s1= String.valueOf(dbl);
    return (s1.length()-s1.indexOf('.')-1)<=2;
}
/**
 * degeri noktadan sonra 2 hane olacak sekilde formatlar <p>

```

```

*
* validasyon true ise ve uygun rakam degil ise null doner<p>
* validasyon false ise ve uygun rakam degil ise sonuc yuvarlanir<p>
* @param value
* @return
*/
public static String getYTL(double value, boolean doValidation) {
    String output = null;
    if ((!doValidation) || isValidYTL(value)) {
        java.text.DecimalFormat myFormatter = new
java.text.DecimalFormat(ytIpattern);
        DecimalFormatSymbols dfs = new DecimalFormatSymbols();
        dfs.setDecimalSeparator('.');
        dfs.setGroupingSeparator(',');
        myFormatter.setDecimalFormatSymbols(dfs);
        output = myFormatter.format(value);
        myFormatter = null;
    }
    return output;
}
public static boolean isValidYTL(double valueA) {
    double valueB = 100 * valueA;
    return (0d == valueB - Math rint(valueB));
}
/**
* noktadan sonra en fazla 2 sayi olabilir.<p>
* true veya false doner.
* @param valueA
* @return
*/
public static boolean isValidYTL(String strA) {
    double valueA = 0;
    double valueB = 0;
    boolean aReturn = false;
    try { //Number number =
        valueA = Double.parseDouble(strA);
        valueB = 100 * valueA;
        aReturn = (0d == valueB - Math rint(valueB));
    } catch (Exception e) {
        aReturn = false;
    }
}

```

```

        e.printStackTrace();
    }
    return aReturn;
}
public static double getRoundedDouble(double dbl) {
    dbl = Math.round(dbl * 100) / 100.0d;
    return dbl;
}
}

```

FACE SAYFALARI

Header

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title></title>
<% String baslik="Sisteme Giriş"; %>
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<% if(!jsalertmetin.equals(null)){
// out.print(jsalertmetin+"dddddd");
}%>
<link rel="stylesheet" href="../websayfalar/sitedosyalar/style.css" type="text/css" media="screen" />
<!--[if IE 6]><link rel="stylesheet" href="style.ie6.css" type="text/css" media="screen" /><![endif]-->
<!--[if IE 7]><link rel="stylesheet" href="style.ie7.css" type="text/css" media="screen" /><![endif]-->
<script type="text/javascript" src="../websayfalar/sitedosyalar/jquery.js"></script>
<script type="text/javascript" src="../websayfalar/sitedosyalar/script.js"></script>
<style type="text/css">
.art-post .layout-item-0 { padding-right: 10px;padding-left: 10px; }
.ie7 .art-post .art-layout-cell {border:none !important; padding:0 !important; }
.ie6 .art-post .art-layout-cell {border:none !important; padding:0 !important; }
</style>
<style type="text/css">

```

```

.glossymenu {
    position: relative;
    padding: 0 0 0 34px;
    margin: 0 auto 0 auto;
    background: url(../websayfalar/sitedosyalar/menub_bg.gif) repeat-x; /*tab background image
path*/
    height: 46px;
    width:960px;
    list-style: none;
    border-bottom-left-radius:5px;
    border-bottom-right-radius:5px;
    border-top-left-radius:20px;
    border-top-right-radius:20px;
}
.glossymenu li {
    float:center;
}
.glossymenu li a {
    float: left;
    display: block;
    color:#000;
    text-decoration: none;
    font-family: "Trebuchet MS", Verdana, Arial;
    font-size: 12px;
    font-weight: bold;
    padding:0 0 0 16px; /*Padding to accomodate left tab image. Do not change*/
    height: 46px;
    line-height: 46px;
    text-align: center;
    cursor: pointer;
}
.glossymenu li a b {
    float: left;
    display: block;
    padding: 0 24px 0 8px; /*Padding of menu items*/
}
.glossymenu li.current a, .glossymenu li a:hover {
    color: #fff;
    background: url(../websayfalar/sitedosyalar/menub_hover_left.gif) no-repeat; /*left tab image
path*/

```

```

        background-position: left;
    }
.glossymenu li.current a b, .glossymenu li a: hover b {
    color: #fff;
    background: url(../websayfalar/sitedosyalar/menub_hover_right.gif) no-repeat right top; /*right
tab image path*/
}
.vaco_content {
    float: left;
    margin: 10px 4px 4px 10px;
    padding: 4px;
    width: 142px;
    height: 107px;
    background: #F8F8F8;
    border: 1px solid #CCC;
    overflow: hidden;
}
.vaco_content img { width: 140px; height: 105px; border: 1px solid #ccc }
-->
</style>
</head>
<body>
<div id="art-page-background-glare-wrapper">
<div id="art-page-background-glare"></div>
</div>
<div id="art-main">
<div class="cleared reset-box"></div>
<div class="art-header">
<div class="art-header-position">
<div class="art-header-wrapper">
<div class="cleared reset-box"></div>
<div class="art-header-inner">
<script type="text/javascript" src="../../websayfalar/sitedosyalar/swfobjectart.js"></script>
<script type="text/javascript">
    jQuery((function (swf) {
        return function () {
            swf.switchOffAutoHideShow();
            swf.registerObject("art-flash-object", "9.0.0",
"/websayfalar/sitedosyalar/expressInstall.swf");
        }
    }

```

```

    })(swfobject);
</script>
<div id="art-flash-area">
<div id="art-flash-container">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="992" height="125" id="art-
flash-object">
    <param name="movie" value="./websayfalar/sitedosyalar/container.swf" />
    <param name="quality" value="high" />
    <param name="scale" value="default" />
    <param name="wmode" value="transparent" />
    <param name="flashvars"
value="color1=0xFFFFFFFF&amp;alpha1=.75&amp;framerate1=12&amp;loop=true&amp;wmode=transp
arent&amp;clip=images/flash.swf&amp;radius=15&amp;clipx=0&amp;clipy=-
78&amp;initalclipw=900&amp;initalcliph=255&amp;clipw=992&amp;cliph=281&amp;width=992&amp;h
eight=125&amp;textblock_width=0&amp;textblock_align=no&amp;hasTopCorners=true&amp;hasBott
omCorners=true" />
<param name="swfliveconnect" value="true" />
    <!--[if !IE]-->
    <object type="application/x-shockwave-flash" data="./websayfalar/sitedosyalar/container.swf"
width="992" height="125">
    <param name="quality" value="high" />
    <param name="scale" value="default" />
    <param name="wmode" value="transparent" />
    <param name="flashvars"
value="color1=0xFFFFFFFF&amp;alpha1=.75&amp;framerate1=12&amp;loop=true&amp;wmode=transp
arent&amp;clip=images/flash.swf&amp;radius=15&amp;clipx=0&amp;clipy=-
78&amp;initalclipw=900&amp;initalcliph=255&amp;clipw=992&amp;cliph=281&amp;width=992&amp;h
eight=125&amp;textblock_width=0&amp;textblock_align=no&amp;hasTopCorners=true&amp;hasBott
omCorners=true" />
<param name="swfliveconnect" value="true" />
    <!--<![endif]-->
    <div class="art-flash-alt"><a href="http://www.adobe.com/go/getflashplayer"></a></div>
    <!--[if !IE]-->
    </object>
    <!--<![endif]-->
</object>
</div>
</div>

```

```
<div class="art-logo">
</div>
</div>
</div>
</div>
</div>
</div>
<div class="cleared reset-box"></div>
<div id="mainmenu">
<ul class="glossymenu">
<li><a href="http://www.templatemo.com/"><b>Belgelendirme R.</b></a></li>
<li><a href="%=aConPath%/Act/ITHALATRAPOR" title="free css templates"><b>İthalat
Raporları</b></a></li>
<li><a href="%=aConPath%/Act/URUNRAPOR" rel="nofollow"
onclick="_gaq.push(['_trackPageview', '/outgoing/menu/flashmo.com']);"><b>Ürün
Raporları</b></a></li>
<li><a href="%=aConPath%/Act/PERRAPOR" rel="nofollow" ><b>Personel
İlişkileri</b></a></li>
<li><a href="http://www.templatemo.com/contact"><b>Çıkış</b></a></li>
</ul>
</div>
<div class="art-box art-sheet">
<div class="art-box-body art-sheet-body">
<div class="cleared reset-box"></div>
<div class="art-layout-wrapper">
<div class="art-content-layout">
<div class="art-content-layout-row">
<div class="art-layout-cell art-content">
<div class="art-box art-post">
<div class="art-box-body art-post-body">
<div class="art-post-inner art-article">
<h2 class="art-postheader"><span class="art-postheadericon">İSG Raporları
</span></h2>
<div class="art-postcontent">

Footer
</div><div class="cleared"></div>
</div> <div class="cleared"></div>
</div></div>
<div class="cleared"></div>
</div></div>
```



```

</div></div>
<div class="cleared"></div>
</div>
        <div class="cleared"></div>
</div></div>
<div class="cleared"></div>
<div class="cleared"></div>
</div>
</body>
</html>

```

ARAYÜZ SAYFA ÖRNEKLERİ

Anasayfa.jsp

```

<%@page contentType="text/html"
pageEncoding="UTF-8"%>
<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<a href="<%=aConPath%>/Act/POSTLOGIN"></a>
<jsp:include page="faces/footer.jsp" />

```

Ayliksurunithalatgos.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<% String baslik="Sisteme Giri?"; %>
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<%
String aylikurungraf=" ['Aylar', 'Miktar'], ['2003', 1336060 ], ['2004', 1538156 ], ['2005', 1576579 ],
['2006', 1600652 ],['2007', 1968113 ], ['2008', 1901067 ]";
%>
<SCRIPT LANGUAGE="javascript">
function Postet()
{
        document.form1.action="<%=aConPath%>/Act/AYLIKURUNITHG";
        document.form1.method="POST";
        document.form1.submit();
}

```


<a href="<%=aConPath%>/Act/ITHALATGAUGE">İthalat Sınırı Belirlenmiş Kimyasallar İçin İthalat Miktarları

<a href="<%=aConPath%>/Act/ITHALATURUNULKE" target="new">İthal Edilen Ürünün İllere Göre Dağılımı

<a href="<%=aConPath%>/Act/ITHALATFIRMALAR" target="new">Tehlikeli Kimyasal İthal Eden Firmalar

<a href="<%=aConPath%>/Act/ITHALATFIRMALAR" target="new">Tehlikeli Kimyasal İthal Eden Firmalar

<jp:include page="faces/footer.jsp" />

Kimyasalgauges.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<% String aConPath = request.getContextPath();
```

```
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
```

```
String tablo=String.valueOf(request.getAttribute("gaugetablo"));
```

```
String veriler=String.valueOf(request.getAttribute("gaugestring"));
```

```
%>
```

```
<head>
```

```
<script type='text/javascript' src='<%=aConPath%>/websayfalar/jsapi.js'></script>
```

```
<script type='text/javascript'>
```

```
    google.load('visualization', '1', {packages:['gauge']});
```

```
    google.setOnLoadCallback(drawChart);
```

```
    function drawChart() {
```

```
        var data = google.visualization.arrayToDataTable([
```

```
<%=veriler%>
```

```
        ]);
```

```
        var options = {
```

```
            width: 600, height: 180,
```

```
            redFrom: 90, redTo: 100,
```

```
            yellowFrom:75, yellowTo: 90,
```

```
            minorTicks: 5
```

```
        };
```

```
        var chart = new google.visualization.Gauge(document.getElementById('chart_div'));
```

```
        chart.draw(data, options);
```

```
    }
```

```
</script>
```

```
</head>
```

```

<body>
<div align=center><center><B>İTHALAT SINIRI BELİRLENEN KİMYASALLAR İÇİN YILLIK MEVCUT
DURUM RAPORU</b><br>
(<%=request.getAttribute("gaugeyil")%> yılı)<BR><BR>
</CENTER></div>
<div id='chart_div' align=center></div>
<div align=center><center>
<table border=0 width=540>
<TR>
<%=tablo%>
</TR>
</TABLE><br></body></html>

```

Msdsler.jsp

```

<script type="text/javascript">
<!--
function confirmation(msdsid) {
var answer = confirm ("MSDS Bilgisini Silmek ?stedi?inize Emin misiniz")
  if (answer) {
var link="pgdnavigator.php?req=msdssilsql&msdsid="+msdsid+"&myID=<?=$_GET[myID]?>";
window.location=link}
}
//-->
</script>
<?
$firmaid=firmaidgetir($myID);
/// echo "rrrrrr=".$firmaid;
$sQuery = "SELECT
msds.msdsbaslik,
msds.msdsid
FROM
msds
WHERE
msds.silmedurum=0
and firmaid='$firmaid'
order by msdsbaslik ";
mb_internal_encoding('UTF-8');
mysql_query('SET NAMES UTF8');
mysql_query("SET CHARACTER SET utf8");
mysql_query("SET COLLATION_CONNECTION = 'utf8_turkish_ci' ");

```

```

$arResult=mydb_query($sQuery);
    $sInfo="";
    $sInfodata="";
?>
<BR>
<table width="800">
<tr>
<td align="center"><strong>MSDS Ad?</strong></td>
<td align="left">&nbsp;</td>
<td align="left">&nbsp;</td>
<td align="left">&nbsp;</td>
</tr>
<?
while ($row = mydb_fetch_array($arResult)){
?>
<tr>
<td align="left"><? echo $row[msdsbaslik]; ?></td>
<td align="center"><a
href="pgdnavigator.php?req=msdsdetay&myID=<?=$myID?>">Detaylar</a></td>
<td align="left"><a
href="pgdnavigator.php?req=msdsguncelle&myID=<?=$myID?>">Güncelle</a></td>
<td align="left"><a href=javascript:confirmation("<?=$row[msdsid]?>")>Sil</a></td>
</tr>
<? } ?>
</table>
<div align="center"><a href="pgdnavigator.php?req=msdsgiris&myID=<?=$myID?>">Yeni MSDS
Formu Ekle</a></div>

```

Perraporlari.jsp

```

<% String baslik="Sisteme Giri?"; %>
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<% if(!jsalertmetin.equals(null)){
// out.print(jsalertmetin+"dddddd");
}%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```

<title></title>
<style type="text/css">
<!--
body {
    margin: 0;
    padding: 0;
    line-height: 1.7em;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 11px;
    background: #404040 url(../websayfalar/sitedosyalar/arkaplan.jpg) no-repeat center top;
}
a:link { color: #0066AA; text-decoration: none}
a:visited { color: #0066AA; text-decoration: none}
a:active { color: #CC0066; text-decoration: underline}
a:hover { color: #CC0066; text-decoration: underline}
b {    font-size: 110%; }
em { color: red; }
#maincontainer {
    width: 980px;
    margin: 0 auto;
    background: #FFFFFF;
}
#mainmenu {
    background: #FFFFFF;
}
.header_left {
    float: left;
    width: 300px;
    padding-top: 68px;
    padding-left: 40px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: bold;
    color: #FFF;
}

.header_left a {
    color: #FFF;
}
.header_left a:hover {

```

```

        color: #9CF;
        text-decoration: none;
    }
    .header_left span {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 11px;
        font-weight: normal;
        color: #99CCFF;
    }
    .header_right {
        float: right;
        width: 320px;
        margin: 20px 20px 0px 0px;
    }
    .note { color:#666; }
    #top_section {
        background: url(../websayfalar/sitedosyalar/header.jpg) no-repeat;
        height: 110px;
    }
    #content_area {
        clear: both;
        min-height: 800px;
        background: #FFF;
    }
    #content_column {
        float: left;
        width: 670px;
    }
    #right_column {
        float: right;
        width: 165px;
        line-height: 1.8em;
    }
    #left_column {
        float: left;
        padding-left: 10px;
        width: 300px;
    }
    #footer {
        clear: both;

```

```

        width: 100%;
        background: black;
        color: #FFF;
        text-align: center;
        padding: 4px 0;
        font-family: Arial, Helvetica, sans-serif;
        font-size: 10px;
    }
    #footer a {
        color: #FFFFFF;
    }
    #template_preview {
        margin: auto;
        margin-bottom: 15px;
        padding: 0px;
        width: 816px;
        height: 816px;
    }
    #preview_footer {
        float: left;
        margin-left: 82px;
        margin-bottom: 20px;
        padding: 20px;
        width: 442px; /* 774px */
        background: #F2F2F2;
        border: 1px solid #DDDDDD;
    }
    #preview_footer_right {
        float: left;
        margin-left: 14px;
        margin-bottom: 20px;
        padding: 0px;
        width: 300px;
    }
    .line {
        height: 2px;
        background: #0066AA;
    }
    .inner_tube {
        margin: 10px;

```



```

}
.right_box {
    margin-top: 10px;
    margin-right: 10px;
}
.templatemo_clear { clear: both; width: 100%; height: 0px; font-size: 0px; }
.templatemo_link_box {
    float: left;
    width: 190px;
    line-height: 1.8em;
    padding: 0px;
}
.title {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 22px;
    font-weight: bold;
    color: #009900;
}
.subtitle {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
    font-weight: bold;
    color: #CC0066;
}
/* 200x200 starts */
#one_column {
    width: 930px;
    margin-left: 36px;
}
#wix_top_banner {
    width: 100%;
    margin: auto;
}
.bottom_right {
    float: left;
    width: 660px;
    margin-top: 10px;
}
holder_bottom {
    float: left;

```

```

margin: 0px;
padding: 0px;
width: 196px;
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
font-size: 12px;
color: #666666;
text-align: left;
}
(holder_200 {
float: left;
margin: 0px 20px 0px 0px;
padding: 5px;
width: 202px;
font-family: Tahoma, Geneva, sans-serif;
font-size: 13px;
font-weight: bold;
text-align: center;
}
.thumbnail_200 {
float: left;
margin: 10px 20px 10px 0px;
padding: 5px 5px 8px 5px;
width: 200px;
background: #F2F2F2 url(../websayfalar/sitedosyalar/tm_2014_tn_bg6.jpg);
border: 1px solid #DDDDDD;
font-family: Tahoma, Geneva, sans-serif;
font-size: 13px;
font-weight: bold;
text-align: left;
}
.wix_200 {
float: left;
margin: 10px 20px 10px 0px;
padding: 5px;
width: 200px;
background: #FFFFFF;
border: 1px solid #DDDDDD;
}
.home_right {
clear: both;

```

```

        float: right;
        margin-right: 22px;
    }
    .flashmo_download a {
        clear: both;
        float: left;
        display: block;
        padding: 10px 5px;
        width: 200px;
        border: 1px solid #DDDDDD;
        background: #F2F2F2 url("../websayfalar/sitedosyalar/download_bg.jpg");
        font-size: 13px;
        font-weight: bold;
        text-align: center;
    }
    .flashmo_download a:hover, .flashmo_download a:active {
        background: #DDDDDD;
        color: #555555;
        text-decoration: none;
    }
    .h10 { height: 10px; }
    .h20 { height: 20px; }
    .h30 { height: 30px; }
    .templatemo_download a {
        float: left;
        display: block;
        margin-right: 20px;
        padding: 16px;
        width: 144px;
        background: #0066AA;
        color: #FFFFFF;
        font-size: 18px;
        font-weight: bold;
        font-family: Arial, Helvetica, sans-serif;
        text-align: center;
    }
    .templatemo_download a:hover, .templatemo_download a:active {
        background: #0099CC;
        color: #FFFFFF;
    }

```

```

        text-decoration: none;
    }
    .thumbnail {
        margin-bottom: 8px;
    }
    .google_sb {
        width: 220px;
    }
    .wix_label {
        margin-top: 8px;
        margin-bottom: 2px;
        color: #666666;
    }
    /* 200x200 ending */
    .top_links {
        margin: 0px 0px 20px 0px;
        padding: 2px 0px 5px 30px;
        background: #045C97;
        font-size: 11px;
    }
    .top_links a:link, .top_links a:visited {
        color: #FFFFFF;
    }
    .top_links a:active, .top_links a:hover {
        color: #CCFF00;
    }
    .thumbnail_frame {
        float: left;
        margin: 10px 4px 4px 10px;
        padding: 5px;
        width: 140px;
        background: #F8F8F8;
        border: 1px solid #DDDDDD;
        font-family: Arial, Helvetica, sans-serif;
        text-align: left;
    }
    .thumbnail_title {
        font-weight: bold;
    }
    .blog_post a:link, .blog_post a:visited {

```

```

        font-family: Arial, Helvetica, sans-serif;
        font-size: 13px;
        color: #333333;
        line-height: 1.7em;
    }
    .blog_post a:active, .blog_post a:hover {
        color: #AA0000;
    }
    .page_navigation { margin-bottom: 10px; }
    div.paging { margin: 0px; padding: 0px; }
    div.paging a, .paging_info { display: inline-block; margin: 0 8px 8px 0; padding: 4px 8px; background-
    color: #FFFFFF; color: #333333; border: 1px solid #999999; }
    div.paging a:hover, div.paging a:active { background-color: #CCCCCC; color: #000000; text-
    decoration: none; }
    div.paging span.current { display: inline-block; margin: 0 8px 8px 0; padding: 4px 8px; background-
    color: #CCCCCC; color: #000000; font-weight: bold; border: 1px solid #999999; }
    div.paging span.disabled { display: inline-block; margin: 0 8px 8px 0; padding: 4px 8px; border: 1px
    solid #DDDDDD; color: #AAAAAA; }
    a.boldbuttons {
        background: transparent url('../websayfalar/roundedgray-left.gif') no-repeat top left;
        display: block;
        float: left;
        font: bold 13px Arial;
        line-height: 22px;
        height: 30px;
        padding-left: 8px;
        text-decoration: none;
    }
    a:link.boldbuttons, a:visited.boldbuttons, a:hover.boldbuttons, a:active.boldbuttons {
        color: white;
    }
    a.boldbuttons span {
        background: transparent url('../websayfalar/roundedgray-right.gif') no-repeat top right;
        display: block;
        padding: 4px 10px 4px 2px; /* Set 10px to be equal or greater than 'padding-left' value above */
    }
    a:hover.boldbuttons { /* Hover state CSS */
        text-decoration: underline;
    }
    .buttonwrapper {

```

```

        overflow: hidden;
        width: 100%;
    }
    .glossymenu {
        position: relative;
        padding: 0 0 0 34px;
        margin: 0 auto 0 auto;
        background: url(../websayfalar/sitedosyalar/menub_bg.gif) repeat-x; /*tab background image
path*/
        height: 46px;
        list-style: none;
    }
    .glossymenu li {
        float:left;
    }
    .glossymenu li a {
        float: left;
        display: block;
        color:#000;
        text-decoration: none;
        font-family: "Trebuchet MS", Verdana, Arial;
        font-size: 12px;
        font-weight: bold;
        padding:0 0 0 16px; /*Padding to accomodate left tab image. Do not change*/
        height: 46px;
        line-height: 46px;
        text-align: center;
        cursor: pointer;
    }
    .glossymenu li a b {
        float: left;
        display: block;
        padding: 0 24px 0 8px; /*Padding of menu items*/
    }
    .glossymenu li.current a, .glossymenu li a:hover {
        color: #fff;
        background: url(../websayfalar/sitedosyalar/menub_hover_left.gif) no-repeat; /*left tab image
path*/
        background-position: left;
    }
}

```

```

.glossymenu li.current a b, .glossymenu li a: hover b {
    color: #fff;
    background: url(../websayfalar/sitedosyalar/menub_hover_right.gif) no-repeat right top; /*right
tab image path*/
}
.vaco_content {
    float: left;
    margin: 10px 4px 4px 10px;
    padding: 4px;
    width: 142px;
    height: 107px;
    background: #F8F8F8;
    border: 1px solid #CCC;
    overflow: hidden;
}
.vaco_content img { width: 140px; height: 105px; border: 1px solid #ccc }
-->
</style>
<script type="text/javascript" src="websayfalar/arayuzdosyalar/swfobject.js"></script>
</head>
<body>
<script>(function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src = "//connect.facebook.net/en_US/all.js#xfbml=1";
    fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');</script>
<div id="maincontainer" >
<div id="top_section">
</div>
<div id="mainmenu">
<ul class="glossymenu">
<li><a href="http://www.templatemo.com/"><b>Home</b></a></li>
<li><a href="http://www.templatemo.com/page/1" title="free css templates"><b>CSS
Templates</b></a></li>
<li><a href="http://www.flashmo.com/page/1" rel="nofollow" onclick="_gaq.push(['_trackPageview',
'/outgoing/menu/flashmo.com']);"><b>Flash Templates</b></a></li>
<li class="current"><a href="<%=aConPath%>/Act/PERRAPOR" rel="nofollow" ><b>Personel
?.</b></a></li>

```

```

</li><a href="http://www.templatemo.com/contact"><b>Ç?k??</b></a></li>
</ul>
</div>
<div style="clear: both; height: 10px;"></div>
<div id="content_area" style="margin-left: 35px; margin-right: 30px; margin-top: 40px; margin-bottom:
30px;">
<a href="<%=aConPath%>/Act/AYLIKPERSONAYI">aylar baz?nda inceleme say?lar?</a>
<div class="templatemo_clear"></div>
</div>
<div id="footer">
<b>Copyright © 2014 </div>
</div>
</body></html>

```

Urunraporlari.jsp

```

<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<BR>
<a href="<%=aConPath%>/Act/AYLIKURUNITH">Seçilen ürünün ayl?k ithalat miktarlar?</a>
<jsp:include page="faces/footer.jsp" />

```

belgemevsim.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<html>
<head>
<script type="text/javascript" src="<%=aConPath%>/websayfalar/jsapi.js"></script>
<script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
        var data = google.visualization.arrayToDataTable([
            ['Dönem', 'Belge Sayısı'],
            ['Aralık-Şubat', 4123],
            ['Mart-Mayıs', 7322],
            ['Haziran-Agustos', 6545],
            ['Eylül-Kasım', 3611]

```



```

    });
    var options = {
        title: 'Dönemsel Belge Sayıları',
        pieHole: 0.4,
    };
    var chart = new google.visualization.PieChart(document.getElementById('donutchart'));
    chart.draw(data, options);
}
</script>
</head>
<body>
<div id="donutchart" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

belgerapor.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<a href="<%=aConPath%>/Act/ONAYRED" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/BELGEMEVSIM" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/AYLARBELGE" target="new">
</a>
<BR>
<jsp:include page="faces/footer.jsp" />

```

ithalatrporlari.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));

```

```

%>
<BR>
<a href="<%=aConPath%>/Act/ITHALATFIRMALAR">
</a>
<BR>
<a href="<%=aConPath%>/Act/ITHALATURUNULKE" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/ITHALATGAUGE" target="new">
</a>
<BR>
<jsp:include page="faces/footer.jsp" />

```

perraporlari.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<a href="<%=aConPath%>/Act/PERORT" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/AYLIKPERSAYI">
</a>
<BR>
<a href="<%=aConPath%>/Act/INCEVRAK" target="new">
</a>
<BR>
<jsp:include page="faces/footer.jsp" />

```

personelbelgedagilim.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));

```

```

%>
<html>
<head>
<script type="text/javascript" src="<%=aConPath%/websayfalar/jsapi.js"></script>
<script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
        var data = google.visualization.arrayToDataTable([
            ['Personel', 'Belge Sayısı'],
            ['Mesut Akaner', 4],
            ['Ayşe Akaner', 6],
            ['Erdal Akaner', 5],
            ['Özge Akaner', 2],
            ['İSGÜM Per1', 7]
        ]);
        var options = {
            title: 'İncelenen Belgelerin Personele Göre Dağılımı',
            is3D: true,
        };
        var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));
        chart.draw(data, options);
    }
</script>
</head>
<body>
<div id="piechart_3d" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

thk.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:x="urn:schemas-microsoft-com:office:excel"
xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882"
xmlns="http://www.w3.org/TR/REC-html40">
<head>
<meta name="Excel Workbook Frameset">
<meta http-equiv=Content-Type content="text/html; charset=Windows-1254">

```

```

<meta name=ProgId content=Excel.Sheet>
<meta name=Generator content="Microsoft Excel 14">
<link rel=File-List href="thk_dosyalar/filelist.xml">
<![if !supportTabStrip]>
<link id="shLink" href="thk_dosyalar/sheet001.htm">
<link id="shLink">
<script language="JavaScript">
<!--
var c_ITabs=1;
var c_rgszSh=new Array(c_ITabs);
c_rgszSh[0] = "Tehlikeli□Kimyasal□Karnesi";
var c_rgszClr=new Array(8);
c_rgszClr[0]="window";
c_rgszClr[1]="buttonface";
c_rgszClr[2]="windowframe";
c_rgszClr[3]="windowtext";
c_rgszClr[4]="threedlightshadow";
c_rgszClr[5]="threedhighlight";
c_rgszClr[6]="threeddarkshadow";
c_rgszClr[7]="threedshadow";
var g_iShCur;
var g_rglTabX=new Array(c_ITabs);
function fnGetIEVer()
{
var ua=window.navigator.userAgent
var msie=ua.indexOf("MSIE")
if (msie>0 && window.navigator.platform=="Win32")
return parseInt(ua.substring(msie+5,ua.indexOf(".", msie)));
else
return 0;
}
function fnBuildFrameset()
{
var szHTML="<frameset rows=\",18\" border=0 width=0 frameborder=no framespacing=0>"+
"<frame src=\""+document.all.item("shLink")[0].href+"\" name=\"frSheet\" noresize>"+
"<frameset cols=\"54,*\" border=0 width=0 frameborder=no framespacing=0>"+
"<frame src=\"\" name=\"frScroll\" marginwidth=0 marginheight=0 scrolling=no>"+
"<frame src=\"\" name=\"frTabs\" marginwidth=0 marginheight=0 scrolling=no>"+
"</frameset></frameset><plaintext>";
with (document) {

```

```

open("text/html", "replace");
write(szHTML);
close();
}
fnBuildTabStrip();
}
function fnBuildTabStrip()
{
var szHTML=
"<html><head><style>.clScroll {font:8pt Courier New;color:"+c_rgszClr[6]+";cursor:default;line-
height:10pt;}"+
".clScroll2 {font:10pt Arial;color:"+c_rgszClr[6]+";cursor:default;line-height:11pt;}</style></head>"+
"<body onclick=\"event.returnValue=false;\" ondragstart=\"event.returnValue=false;\"
onselectstart=\"event.returnValue=false;\" bgcolor="+c_rgszClr[4]+" topmargin=0 leftmargin=0<table
cellpadding=0 cellspacing=0 width=100%>"+
"<tr><td colspan=6 height=1 bgcolor="+c_rgszClr[2]+"></td></tr>"+
"<tr><td style=\"font:1pt\">&nbsp;<td>"+
"<td valign=top id=tdScroll class=\"clScroll\" onclick=\"parent.fnFastScrollTabs(0);\"
onmouseover=\"parent.fnMouseOverScroll(0);\"
onmouseout=\"parent.fnMouseOutScroll(0);\"><a>#171;</a></td>"+
"<td valign=top id=tdScroll class=\"clScroll2\" onclick=\"parent.fnScrollTabs(0);\"
ondblclick=\"parent.fnScrollTabs(0);\" onmouseover=\"parent.fnMouseOverScroll(1);\"
onmouseout=\"parent.fnMouseOutScroll(1);\"><a>&lt;</a></td>"+
"<td valign=top id=tdScroll class=\"clScroll2\" onclick=\"parent.fnScrollTabs(1);\"
ondblclick=\"parent.fnScrollTabs(1);\" onmouseover=\"parent.fnMouseOverScroll(2);\"
onmouseout=\"parent.fnMouseOutScroll(2);\"><a>&gt;</a></td>"+
"<td valign=top id=tdScroll class=\"clScroll\" onclick=\"parent.fnFastScrollTabs(1);\"
onmouseover=\"parent.fnMouseOverScroll(3);\"
onmouseout=\"parent.fnMouseOutScroll(3);\"><a>#187;</a></td>"+
"<td style=\"font:1pt\">&nbsp;<td></tr></table></body></html>";
with (frames["frScroll"].document) {
open("text/html", "replace");
write(szHTML);
close();
}
szHTML =
"<html><head>"+
"<style>A.link,A.visited,A.active {text-decoration:none;"+color:"+c_rgszClr[3]+";}"+
".clTab {cursor:hand;background:"+c_rgszClr[1]+";font:9pt Arial;padding-left:3px;padding-
right:3px;text-align:center;}"+

```

```

".clBorder {background:"+c_rgszClr[2]+";font:1pt;}"+
"</style></head><body onload=\\"parent.fnInit();\" onselectstart=\\"event.returnValue=false;\"
ondragstart=\\"event.returnValue=false;\" bgcolor="+c_rgszClr[4]+
" topmargin=0 leftmargin=0><table id=tbTabs cellpadding=0 cellspacing=0>";
var iCellCount=(c_ITabs+1)*2;
var i;
for (i=0;i<iCellCount;i+=2)
  szHTML+="<col width=1><col>";
var iRow;
for (iRow=0;iRow<6;iRow++) {
  szHTML+="<tr>";
  if (iRow==5)
    szHTML+="<td colspan="+iCellCount+"></td>";
  else {
    if (iRow==0) {
      for(i=0;i<iCellCount;i++)
        szHTML+="<td height=1 class=\\"clBorder\"></td>";
    } else if (iRow==1) {
      for(i=0;i<c_ITabs;i++) {
        szHTML+="<td height=1 nowrap class=\\"clBorder\">&nbsp;</td>";
        szHTML+=
          "<td id=tdTab height=1 nowrap class=\\"clTab\" onmouseover=\\"parent.fnMouseOverTab(\"+i+\");\"
onmouseout=\\"parent.fnMouseOutTab(\"+i+\");\">"+
          "<a href=\\""+document.all.item(\"shLink\")[i].href+\"\" target=\\"frSheet\"
id=aTab>&nbsp;&"+c_rgszSh[i]+&nbsp;</a></td>";
      }
      szHTML+="<td id=tdTab height=1 nowrap class=\\"clBorder\"><a id=aTab>&nbsp;</a></td><td
width=100%></td>";
    } else if (iRow==2) {
      for (i=0;i<c_ITabs;i++)
        szHTML+="<td height=1></td><td height=1 class=\\"clBorder\"></td>";
      szHTML+="<td height=1></td><td height=1></td>";
    } else if (iRow==3) {
      for (i=0;i<iCellCount;i++)
        szHTML+="<td height=1></td>";
    } else if (iRow==4) {
      for (i=0;i<c_ITabs;i++)
        szHTML+="<td height=1 width=1></td><td height=1></td>";
      szHTML+="<td height=1 width=1></td><td></td>";
    } }
} }

```

```

    szHTML+="</tr>";
}
szHTML+="</table></body></html>";
with (frames['frTabs'].document) {
    open("text/html","replace");
    charset=document.charset;
    write(szHTML);
    close();
}
function fnInit()
{
    g_rglTabX[0]=0;
    var i;
    for (i=1;i<=c_ITabs;i++)
        with (frames['frTabs'].document.all.tbTabs.rows[1].cells[fnTabToCol(i-1)])
            g_rglTabX[i]=offsetLeft+offsetWidth-6;
}
function fnTabToCol(iTab)
{
    return 2*iTab+1;
}
function fnNextTab(fDir)
{
    var iNextTab=-1;
    var i;
    with (frames['frTabs'].document.body) {
        if (fDir==0) {
            if (scrollLeft>0) {
                for (i=0;i<c_ITabs&&g_rglTabX[i]<scrollLeft;i++);
                if (i<c_ITabs)
                    iNextTab=i-1;
            }
        } else {
            if (g_rglTabX[c_ITabs]+6>offsetWidth+scrollLeft) {
                for (i=0;i<c_ITabs&&g_rglTabX[i]<=scrollLeft;i++);
                if (i<c_ITabs)
                    iNextTab=i;
            }
        }
    }
}
}

```

```

return iNextTab;
}
function fnScrollTabs(fDir)
{
var iNextTab=fnNextTab(fDir);
if (iNextTab>=0) {
frames['frTabs'].scroll(g_rglTabX[iNextTab],0);
return true;
} else
return false;
}
function fnFastScrollTabs(fDir)
{
if (c_ITabs>16)
frames['frTabs'].scroll(g_rglTabX[fDir?c_ITabs-1:0],0);
else
if (fnScrollTabs(fDir)>0) window.setTimeout("fnFastScrollTabs("+fDir+");",5);
}
function fnSetTabProps(iTab,fActive)
{
var iCol=fnTabToCol(iTab);
var i;
if (iTab>=0) {
with (frames['frTabs'].document.all) {
with (tbTabs) {
for (i=0;i<=4;i++) {
with (rows[i]) {
if (i==0)
cells[iCol].style.background=c_rgszClr[fActive?0:2];
else if (i>0 && i<4) {
if (fActive) {
cells[iCol-1].style.background=c_rgszClr[2];
cells[iCol].style.background=c_rgszClr[0];
cells[iCol+1].style.background=c_rgszClr[2];
} else {
if (i==1) {
cells[iCol-1].style.background=c_rgszClr[2];
cells[iCol].style.background=c_rgszClr[1];
cells[iCol+1].style.background=c_rgszClr[2];
} else {

```



```

        cells[iCol-1].style.background=c_rgszClr[4];
        cells[iCol].style.background=c_rgszClr[(i==2)?2:4];
        cells[iCol+1].style.background=c_rgszClr[4];
    } }
} else
    cells[iCol].style.background=c_rgszClr[fActive?2:4];
} } }
with (aTab[iTab].style) {
    cursor=(fActive?"default":"hand");
    color=c_rgszClr[3];
} } }
function fnMouseOverScroll(iCtl)
{
    frames["frScroll"].document.all.tdScroll[iCtl].style.color=c_rgszClr[7];
}
function fnMouseOutScroll(iCtl)
{
    frames["frScroll"].document.all.tdScroll[iCtl].style.color=c_rgszClr[6];
}
function fnMouseOverTab(iTab)
{
    if (iTab!=g_iShCur) {
        var iCol=fnTabToCol(iTab);
        with (frames["frTabs"].document.all) {
            tdTab[iTab].style.background=c_rgszClr[5];
        } }
}
function fnMouseOutTab(iTab)
{
    if (iTab>=0) {
        var elFrom=frames["frTabs"].event.srcElement;
        var elTo=frames["frTabs"].event.toElement;
        if ((!elTo) ||
            (elFrom.tagName==elTo.tagName) ||
            (elTo.tagName=="A" && elTo.parentElement!=elFrom) ||
            (elFrom.tagName=="A" && elFrom.parentElement!=elTo)) {
            if (iTab!=g_iShCur) {
                with (frames["frTabs"].document.all) {
                    tdTab[iTab].style.background=c_rgszClr[1];
                } } }
} } } }
function fnSetActiveSheet(iSh)

```

```

{
  if (iSh!=g_iShCur) {
    fnSetTabProps(g_iShCur,false);
    fnSetTabProps(iSh,true);
    g_iShCur=iSh;
  }
}
window.g_iLEVer=fnGetLEVer();
if (window.g_iLEVer>=4)
  fnBuildFrameset();
!-->
</script>
<![endif]><!--[if gte mso 9]><xml>
<x:ExcelWorkbook>
<x:ExcelWorksheets>
<x:ExcelWorksheet>
<x:Name>Tehlikeli Kimyasal Karnesi</x:Name>
<x:WorksheetSource HRef="thk_dosyalar/sheet001.htm"/>
</x:ExcelWorksheet>
</x:ExcelWorksheets>
<x:Stylesheet HRef="thk_dosyalar/stylesheet.css"/>
<x:WindowHeight>10920</x:WindowHeight>
<x:WindowWidth>19440</x:WindowWidth>
<x:WindowTopX>0</x:WindowTopX>
<x:WindowTopY>120</x:WindowTopY>
<x:ProtectStructure>False</x:ProtectStructure>
<x:ProtectWindows>False</x:ProtectWindows>
</x:ExcelWorkbook>
</xml><![endif]-->
</head>
<frameset rows="*,39" border=0 width=0 frameborder=no framespacing=0>
<frame src="thk_dosyalar/sheet001.htm" name="frSheet">
<frame src="thk_dosyalar/tabstrip.htm" name="frTabs" marginwidth=0 marginheight=0>
<noframes>
<body>
</body>
</noframes>
</frameset>
</html>

```

urunraporlari.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:include page="faces/header.jsp" />
<% String aConPath = request.getContextPath();
String jsalertmetin = String.valueOf(request.getSession().getAttribute("jsalertmetin"));
%>
<a href="../websayfalar/thk.htm" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/AYLIKURUNITH" target="new">
</a>
<BR>
<a href="<%=aConPath%>/Act/INCEVRAK">
</a>
<BR>
<jsp:include page="faces/footer.jsp" />
```

VERİTABANI KURULUM SQL ÖRNEKLERİ

Navicat Oracle Data Transfer

Oracle Client Version : 10.2.0.5.0

Source Server : oracleconn

Source Server Version : 120100

Source Host : localhost:1521

Target Server Type : ORACLE

Target Server Version : 120100

File Encoding : 65001

```
-- -----
-- Table structure for "ANONYMOUS"."basvuru"
-- -----
```

```
DROP TABLE "ANONYMOUS"."basvuru";
CREATE TABLE "ANONYMOUS"."basvuru" (
"basvuruid" NUMBER NOT NULL ,
"firmaid" NUMBER NOT NULL ,
"firmayetkili" VARCHAR2(50 BYTE) NULL ,
"onaydurum" NUMBER DEFAULT 0 NOT NULL ,
"onaypersonel" NUMBER DEFAULT 0 NOT NULL ,
```

```

"onayamir" NUMBER DEFAULT 0 NOT NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL ,
"basvurutarih" DATE NOT NULL ,
"onaytarih" TIMESTAMP(6) NULL ,
"gtip" VARCHAR2(20 BYTE) NULL ,
"urunid" NUMBER NULL ,
"miktar" FLOAT(126) NULL ,
"birimkodu" NUMBER(3) NULL ,
"gbeyannametarih" DATE NULL ,
"gbeyannameno" VARCHAR2(10 BYTE) NULL ,
"kullanimyer" VARCHAR2(400 BYTE) NULL ,
"faturano" VARCHAR2(15 BYTE) NULL ,
"faturatarih" DATE NULL ,
"menseiid" NUMBER(3) NULL ,
"ithalulkeid" NUMBER(3) NULL ,
"ihracatunvan" VARCHAR2(400 BYTE) NULL ,
"ihracatciadres" VARCHAR2(400 BYTE) NULL ,
"ihracatcitelefon" VARCHAR2(30 BYTE) NULL ,
"ihracatciemail" VARCHAR2(100 BYTE) NULL
)
LOGGING
NOCOMPRESS
NOCACHE
;

```

```

-- -----
-- Table structure for "ANONYMOUS"."basvurubelge"
-- -----
DROP TABLE "ANONYMOUS"."basvurubelge";
CREATE TABLE "ANONYMOUS"."basvurubelge" (
"basvurubelgeid" NUMBER NOT NULL ,
"basvuruid" NUMBER NOT NULL ,
"belgeid" NUMBER NOT NULL ,
"firmayetkili" VARCHAR2(50 BYTE) NULL ,
"kabuldurum" NUMBER DEFAULT 0 NOT NULL ,
"onaypersonel" NUMBER DEFAULT 0 NOT NULL ,
"belgelink" VARCHAR2(100 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,

```

```

"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL
)
LOGGING
NOCOMPRESS
NOCACHE
;
-----
-- Table structure for "ANONYMOUS"."basvurumusteri"
-----
DROP TABLE "ANONYMOUS"."basvurumusteri";
CREATE TABLE "ANONYMOUS"."basvurumusteri" (
"basvurumusteriid" NUMBER NOT NULL ,
"basvuruid" NUMBER NULL ,
"musteriid" NUMBER NULL ,
"miktar" FLOAT(126) NULL ,
"silmedurumu" NUMBER DEFAULT 0 NOT NULL
)
LOGGING

NOCACHE
;
-----
-- Table structure for "ANONYMOUS"."belgeturler"
-----
DROP TABLE "ANONYMOUS"."belgeturler";
CREATE TABLE "ANONYMOUS"."belgeturler" (
"belgeturid" NUMBER NOT NULL ,
"belgeturadi" VARCHAR2(250 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL
)
LOGGING

NOCACHE
;
-----

```

-- Table structure for "ANONYMOUS"."firmabelgeler"

```
DROP TABLE "ANONYMOUS"."firmabelgeler";
CREATE TABLE "ANONYMOUS"."firmabelgeler" (
"firmabelgeid" NUMBER NOT NULL ,
"firmaid" NUMBER NOT NULL ,
"belgeturid" NUMBER NOT NULL ,
"firmayetkili" VARCHAR2(50 BYTE) NULL ,
"belgelink" VARCHAR2(100 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL ,
"uploadkisitc" CHAR(11 BYTE) NULL ,
"belgegecerlilikdurumu" NUMBER(1) DEFAULT 0 NOT NULL ,
"belgeonayfirmayetkili" NUMBER(11) NULL ,
"belgekabuldurum" NUMBER(11) DEFAULT 0 NOT NULL ,
"belgekabulyetkili" NUMBER(11) DEFAULT 0 NOT NULL
)
;
```

-- Table structure for "ANONYMOUS"."firmalar"

```
DROP TABLE "ANONYMOUS"."firmalar";
CREATE TABLE "ANONYMOUS"."firmalar" (
"firmaid" NUMBER NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NOT NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL ,
"sgkno" VARCHAR2(26 BYTE) NULL ,
"verginno" VARCHAR2(15 BYTE) NULL ,
"firmaadres" VARCHAR2(300 BYTE) NULL ,
"firmatelefon" VARCHAR2(15 BYTE) NULL ,
"firmafaks" VARCHAR2(15 BYTE) NULL ,
"firmaeposta" VARCHAR2(40 BYTE) NULL ,
"firmailce" NUMBER(5) NULL ,
"firmail" NUMBER(2) NULL ,
"firmayetkilitc" CHAR(11 BYTE) NULL ,
"firmayetkiliadi" VARCHAR2(100 BYTE) NULL ,
```

```

"firmayetkilisoyadi" VARCHAR2(100 BYTE) NULL ,
"firmaadi" VARCHAR2(400 CHAR) NULL
)
-----
-- Table structure for "ANONYMOUS"."grupyetki"
-----
DROP TABLE "ANONYMOUS"."grupyetki";
CREATE TABLE "ANONYMOUS"."grupyetki" (
"yetkigrupid" NUMBER NOT NULL ,
"grupid" NUMBER NOT NULL ,
"yetkiid" NUMBER NOT NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NOT NULL ,
"loguser" VARCHAR2(20 BYTE) NOT NULL ,
"logzaman" TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL ,
"silmedurumu" NUMBER DEFAULT 0 NOT NULL
)
-----
-- Table structure for "ANONYMOUS"."ikkazasebepalt"
-----
DROP TABLE "ANONYMOUS"."ikkazasebepalt";
CREATE TABLE "ANONYMOUS"."ikkazasebepalt" (
"kazasebepkod" VARCHAR2(3 BYTE) NULL ,
"kazasebepaltkod" VARCHAR2(3 BYTE) NULL ,
"kazasebepaltaciklama" VARCHAR2(200 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL
)
-----
-- Table structure for "ANONYMOUS"."ikyarasebep"
-----
DROP TABLE "ANONYMOUS"."ikyarasebep";
CREATE TABLE "ANONYMOUS"."ikyarasebep" (
"yarasebepkod" VARCHAR2(3 BYTE) NULL ,
"yarasebepaciklama" VARCHAR2(200 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL

```

```

)
-----
-- Table structure for "ANONYMOUS"."ikyarasebepalt"
-----

DROP TABLE "ANONYMOUS"."ikyarasebepalt";
CREATE TABLE "ANONYMOUS"."ikyarasebepalt" (
"yarasebepkod" VARCHAR2(3 BYTE) NULL ,
"yarasebepalkod" VARCHAR2(3 BYTE) NULL ,
"yarasebepaltaciklama" VARCHAR2(200 BYTE) NULL ,
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,
"loguser" VARCHAR2(20 BYTE) NULL ,
"logzaman" TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NULL ,
"silmedurumu" NUMBER DEFAULT 0 NULL
)
-----
-- Table structure for "ANONYMOUS"."iller"
-----

DROP TABLE "ANONYMOUS"."iller";
CREATE TABLE "ANONYMOUS"."iller" (
"ilid" NUMBER(3) NOT NULL ,
"iladi" VARCHAR2(20 BYTE) NOT NULL
)
DROP TABLE "ANONYMOUS"."kimyasalkota";
CREATE TABLE "ANONYMOUS"."kimyasalkota" (
"kimkotaid" NUMBER NOT NULL ,
"kimyasaladi" VARCHAR2(200 BYTE) NOT NULL ,
"yil" VARCHAR2(4 BYTE) NOT NULL ,
"miktar" FLOAT(126) NOT NULL ,
"birimi" VARCHAR2(20 BYTE) NOT NULL ,
"silmedurumu" NUMBER DEFAULT 0 NOT NULL ,
"ithalaturunid" NUMBER NOT NULL
)
;
DROP TABLE "ANONYMOUS"."kontroldokuman";
CREATE TABLE "ANONYMOUS"."kontroldokuman" (
"dokumanid" NUMBER DEFAULT 0 NOT NULL ,
"belgelink" VARCHAR2(6 BYTE) NOT NULL ,
"belgeaciklama" VARCHAR2(200 BYTE) NULL ,
"silmedurumu" NUMBER DEFAULT 0 NOT NULL ,
"logislem" VARCHAR2(10 BYTE) NOT NULL ,

```



```
"loguser" VARCHAR2(20 BYTE) NOT NULL ,  
"logzaman" TIMESTAMP DEFAULT CURRENT_TIMESTAMP NULL ,  
"kontrolbelgeid" NUMBER NOT NULL ,  
"belgetur" NUMBER NULL ,  
"belgeuzanti" VARCHAR2(5 BYTE) NULL  
)
```

```
DROP TABLE "ANONYMOUS"."musteriler";  
CREATE TABLE "ANONYMOUS"."musteriler" (  
"musteriid" NUMBER NULL ,  
"logislem" VARCHAR2(10 BYTE) DEFAULT 0 NULL ,  
"loguser" VARCHAR2(20 BYTE) NULL ,  
"logzaman" TIMESTAMP DEFAULT CURRENT_TIMESTAMP NULL ,  
"silmedurumu" NUMBER DEFAULT 0 NULL ,  
"sgkno" VARCHAR2(26 BYTE) NULL ,  
"verginno" VARCHAR2(15 BYTE) NULL ,  
"musteriadres" VARCHAR2(300 BYTE) NULL ,  
"musteritelefon" VARCHAR2(15 BYTE) NULL ,  
"musterifaks" VARCHAR2(15 BYTE) NULL ,  
"musterieposta" VARCHAR2(40 BYTE) NULL ,  
"musteriilce" NUMBER(5) NULL ,  
"musteriil" NUMBER(2) NULL ,  
"musteriadi" VARCHAR2(400 BYTE) NULL  
)
```

ÖRNEK EKKRAN GÖRÜNTÜLERİ



İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

Belgelendirme Raporları İthalat Raporları Ürün Raporları Personel Raporları Çıkış

İSG Raporları



ÇSGB
T.C. ÇALIŞMA VE
SOSYAL GÜVENLİK
BAKANLIĞI



İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

Belgelendirme Raporları İthalat Raporları Ürün Raporları Personel Raporları Çıkış

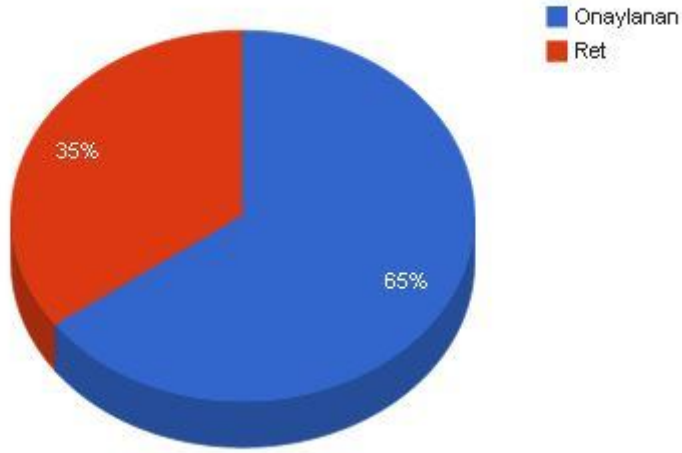
İSG Raporları

Belge Başvurularının Kabul/Red Oranları

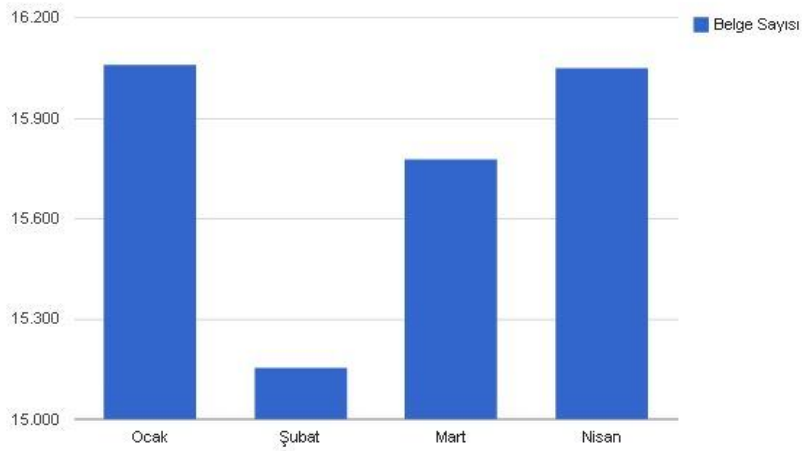
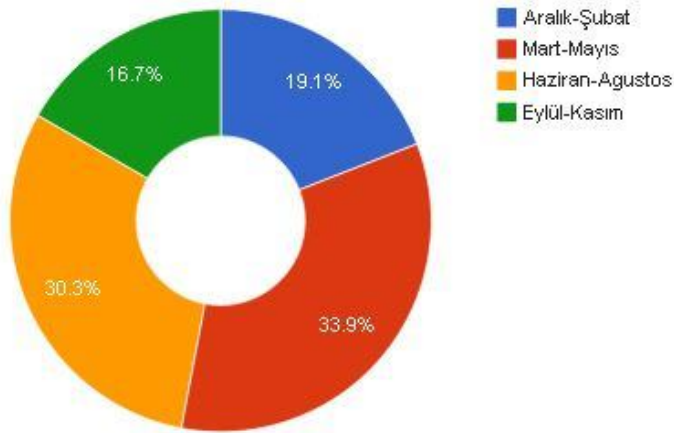
Belge Sayılarının Mevsimsel Dağılımı

Belge Sayılarının Aylara Göre Dağılımı

İncelenen Belgelerin Onaylanma Oranı



Dönemsel Belge Sayıları



Aylara Göre Kontrol Edilen Belge Sayıları (2014)

İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
CSGB
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

Belgelendirme Raporları **İthalat Raporları** Ürün Raporları Personel Raporları Çıkış

İSG Raporları

Tehlikeli Kimyasal İthal Eden Firmalar

İthal Edilen Ürünün İllere Göre Dağılımı

İthalat Sınırı Belirlenmiş Kimyasallar

İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
CSGB
İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

Belgelendirme Raporları **İthalat Raporları** Ürün Raporları Personel Raporları Çıkış

İSG Raporları

Tehlikeli Kimyasal İthalatı Yapan Firmalar

Unvan	İli	SGK No	Vergi No	Başvuru Sayısı	Detaylar
firmaadi	ADANA	3646457456	1231231	3	Detaylar



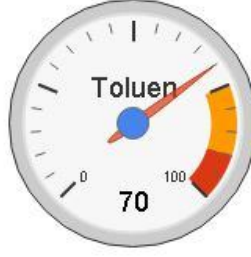
2013 YILI "BENZEN" İTHALATININ İLLER BAZINDA DAĞILIMI

% 0 %0-10 %11-20 %21-30 %31-40 %41-50 %51-60 %61-70 %71-80 %81-90 %91-100

İTHALAT SINIRI BELİRLENEN KİMYASALLAR İÇİN YILLIK MEVCUT DURUM RAPORU
(2014 yılı)



Benzen
Sınır Değer: 90 ton
İthalat M.: 54 ton
Yüzde: %60



Toluen
Sınır Değer: 120 kg
İthalat M.: 84 kg
Yüzde: %70



W. Spirit
Sınır Değer: 160 ton
İthalat M.: 45 ton
Yüzde: %28.125

İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

CSGB

İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

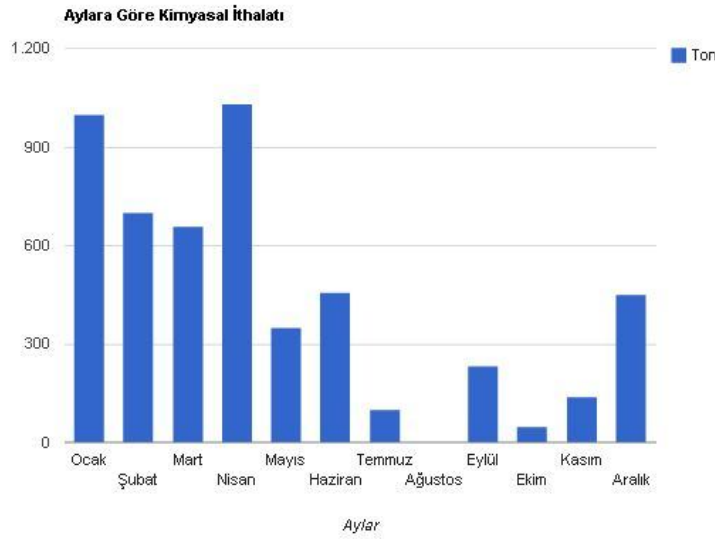
Belgelendirme Raporları İthalat Raporları **Ürün Raporları** Personel Raporları Çıkış

İSG Raporları

Tehlikeli Kimyasal Karnesi

Seçilen Ürünün Aylık İthalat Miktarları

Kimyasal: Benzen Yıl: 2013 Göster

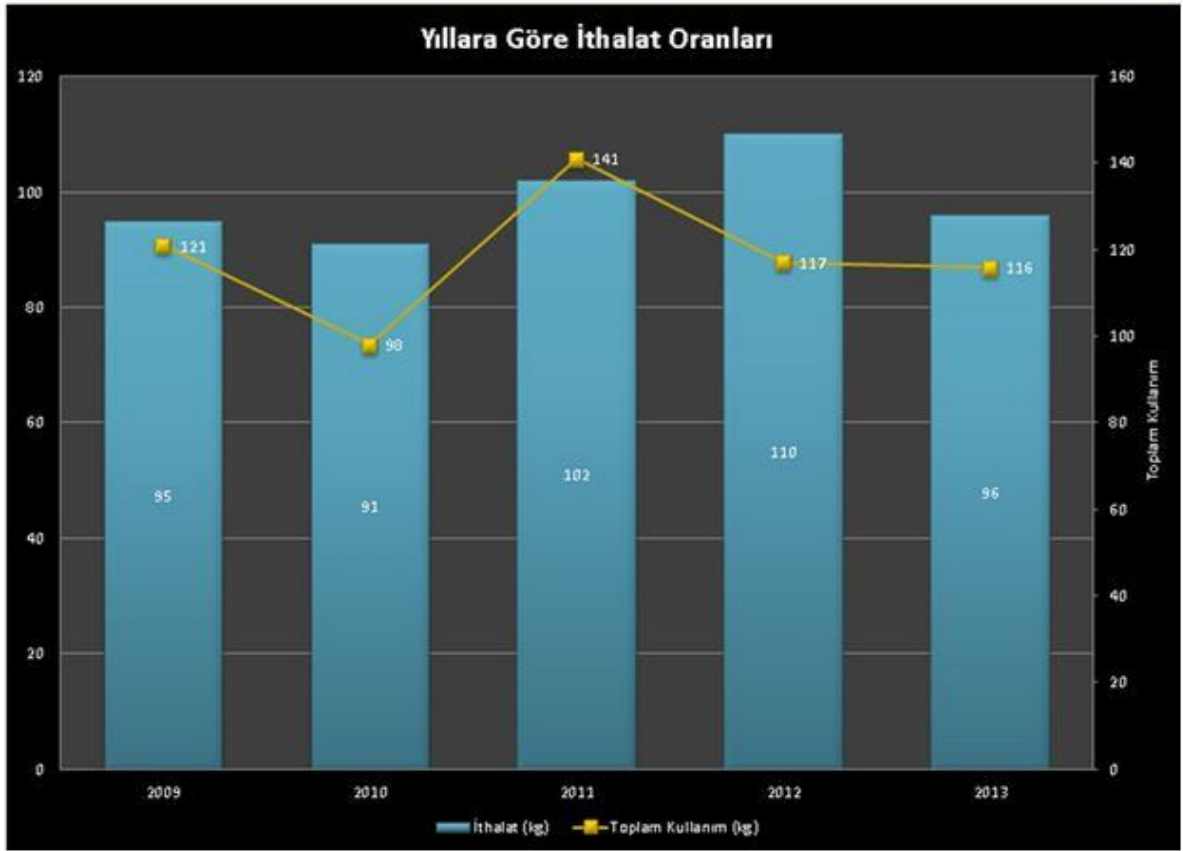


TEHLİKELİ KİMYASAL KARNESİ

Adı: Benzen
Rapor Tarihi: 15.04.2014
EC No: 200-753-7
CAS No: 71-43-2
Sınıflandırma: F;R11 Carc. Cat. 1;R45
Muta. Cat. 2;R46 T;
R48/23/24/25 Xn; R65
Xi;R36/38

Notlar
Kontrol Belgesi İşlemlerinde numune alınır.
İthalat miktar kontrolü bulunmaktadır.

Kimyasalın Yıllara Göre İthalatı ve Kullanım Oranları



Kimyasalın Kullanıldığı Faaliyet Alanları (Bildirimlere göre)

NACE Kodu	Açıklama	Tehlike Sınıfı	Yıllık Ortalama	Toplam Miktar	Yüzde
20.14.04	Odunun ve kömür katranının damıtılması (odun kreozotu, odun naftası, bitkisel zift, benzol, toluol, fenol, naftalin vb.)	Çok Tehlikeli	41.00	260.00	46.00
20.30.13	Diğer boya, vernik ve ilgili ürünlerin imalatı (resim ve tabelacı boyaları, matbaa mürekkepleri, hazır boya pigmentleri, matlaştırıcılar, renklendiriciler, astarlar, cam firit, camlaştırılabilir emay ve sırlar, solvent, inceltici (tiner))	Çok Tehlikeli	2.6.00	170.00	27.00
86.90.10	Tıbbi laboratuvarların hizmetleri (adli tıp ve dış laboratuvarlarının faaliyetleri hariç) (hastane dışı)	Çok Tehlikeli	11.00	39.00	8.00

Mazuriyet Kaynakları

Benzol üretimi,

- Benzolle her tür çalışma,
- Benzolün doldurulması, karıştırılması,
- Çözücü olarak ve ekstrasyon amaçlı kullanılması,
- Temizleyici olarak kullanılması,
- Yapıştırıcılarda çözücü olarak kullanılması,
- Boya, lak, cila, vernik gibi maddelerde çözücü ve inceltici olarak kullanılması,
- Metal parçaların temizliği, aşındırılma ve yağsızlaştırılması,
- Bazı kimyasal işlemler (kimya endüstrisinde).

Tehlikeler

Kolay alevlenir. Cildi tahriş eder. Uzun süreli solunması halinde sağlığa ciddi hasar tehlikesi. Doğmamış çocuğa zarar verme olası riski. Yutulması halinde akciğerde hasara neden olabilir. Buharları uyuşukluğa ve baş dönmesine neden olabilir.

Kaza Bildirimleri

Kaza Nedeni	Yıllara Göre Kaza Sayıları						Toplam	
	2012		2013		2014			
	İşyeri Sayısı	Kaza Sayısı	İşyeri Sayısı	Kaza Sayısı	İşyeri Sayısı	Kaza Sayısı	İşyeri Sayısı	Kaza Sayısı
10 13 Patlama	36	38	21	21	17	19	74	78
20 23 Gaz durumunda - buharlaşma, aerosol oluşum, gaz oluşumu	12	13	16	16	5	5	33	34
20 24 Toz halindeki madde - duman oluşumu, havadaki/yayılmış toz ve zerrecikleri	44	45	34	36	11	11	89	92
Genel Toplam	92	96	71	73	33	35	196	204

Meslek Hastalığı Bildirimleri

NACE KODU	Faaliyet Alanı	Hastalık Tanısı	Yıllara Göre Meslek Hastalığı Sayıları						Toplam	
			2012		2013		2014			
			İşyeri Sayısı	M. H. Sayısı	İşyeri Sayısı	M. H. Sayısı	İşyeri Sayısı	M. H. Sayısı	İşyeri Sayısı	M. H. Sayısı
20.14.04	Odunun ve kömür katranının damıtılması (odun kreozotu, odun naftası, bitkisel zift, benzol, toluol, fenol, naftalin vb.)	A15 Solunum yolları tüberkülozu, bakteriyolojik ve histolojik olarak kanıtlanmış C91 Lenfoid lösemi	0	0	8	9	7	7	15	16
86.90.10	Tıbbi laboratuvarların hizmetleri (adli tıp ve dış laboratuvarlarının faaliyetleri hariç) (hastane dışı)	C94 Lösemiler diğer, özel hücre tipli	4	4	2	3	0	0	6	7
			3	4	5	6	1	1	9	11
Genel Toplam			7	8	15	18	8	8	30	34

Korunma



Birincil korunma teknik koruma önlemlerini içerir. Kişinin etkenle temasını önlenmeye yöneliktir. Amaç kanserojen maruziyetin tanımlanması, mümkünse giderilmesi, değilse en aza indirilmesidir. Bunun için;

1. Yerine koyma (substitution). Örn: benzen yerine stiren, hekzan; asbest yerine sentetik lifler kullanımı gibi.
2. Kapalı sistem, otomatizasyon, uzaktan kontrol
3. Lokal ve genel aspirasyon ve havalandırma
4. Kişisel koruyucu malzeme kullanımı
5. Maruziyet süresinin kısaltılması
6. Dönüşümlü çalışma
7. Kanserोजenle çalışan kişi sayısının azaltılması
8. Düzenli bakım onarım
9. Eğitim
10. Kişisel hijyen önlemleri
11. İş ortamı ölçümleri yapılmalıdır.



İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

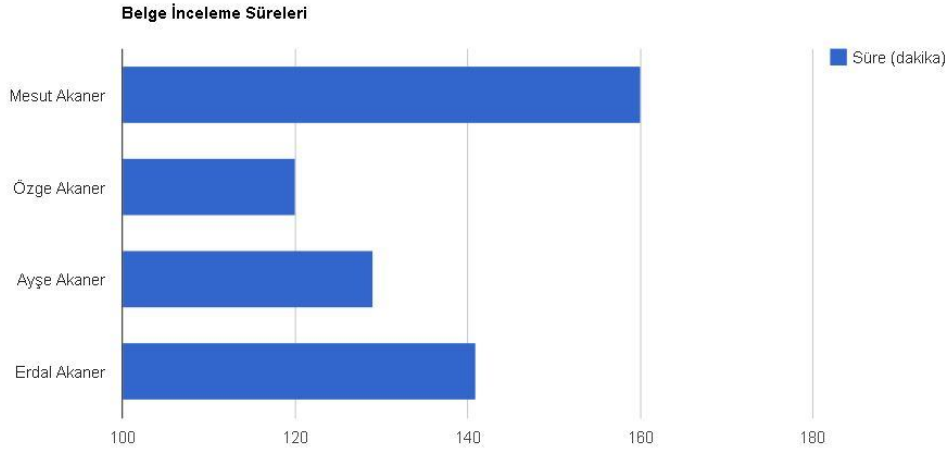
Belgelendirme Raporları İthalat Raporları Ürün Raporları **Personel Raporları** Çıkış

İSG Raporları

Personelin Ortalama İnceleme Süreleri

Aylara Göre İnceleme Sayıları

İncelenen Evrakların Personele Dağılımı



İŞ SAĞLIĞI VE GÜVENLİĞİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
ÇSGB
 İTHAL EDİLEN TEHLİKELİ KİMYASAL MADDELERİN GÖZETİMİ
 KONTROL BELGESİ BAŞVURU VE TAKİP SİSTEMİ

Belgelendirme Raporları İthalat Raporları Ürün Raporları Personel Raporları Çıkış

ISG Raporları

Kontrol Belgesi Teknik Ekip Performans Değerlendirmesi

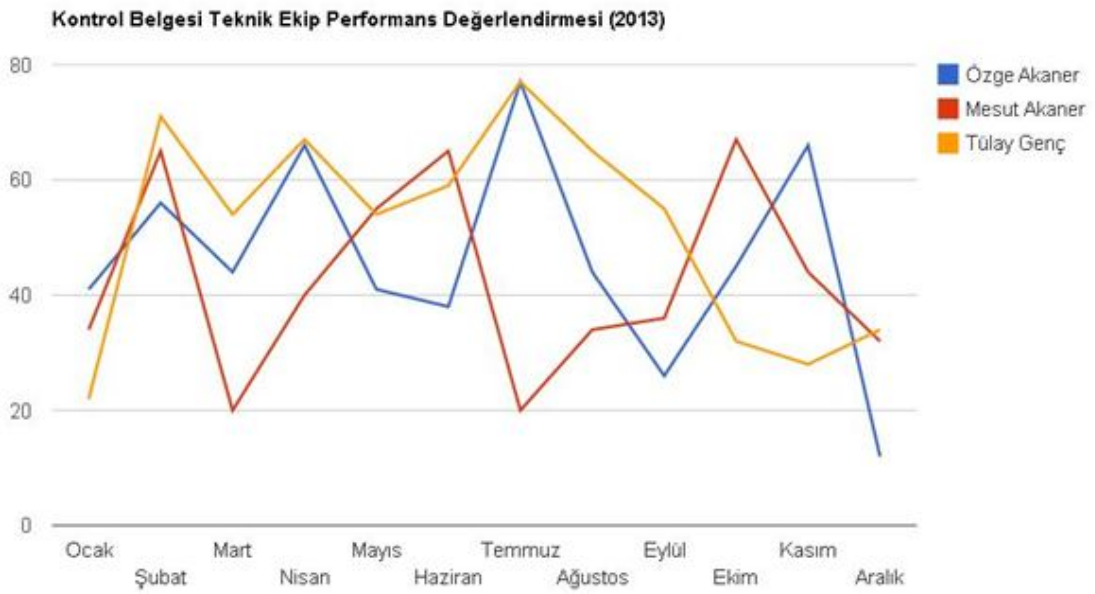
Değerlendirme Yılı: 2013

Değerlendirilecek Personel

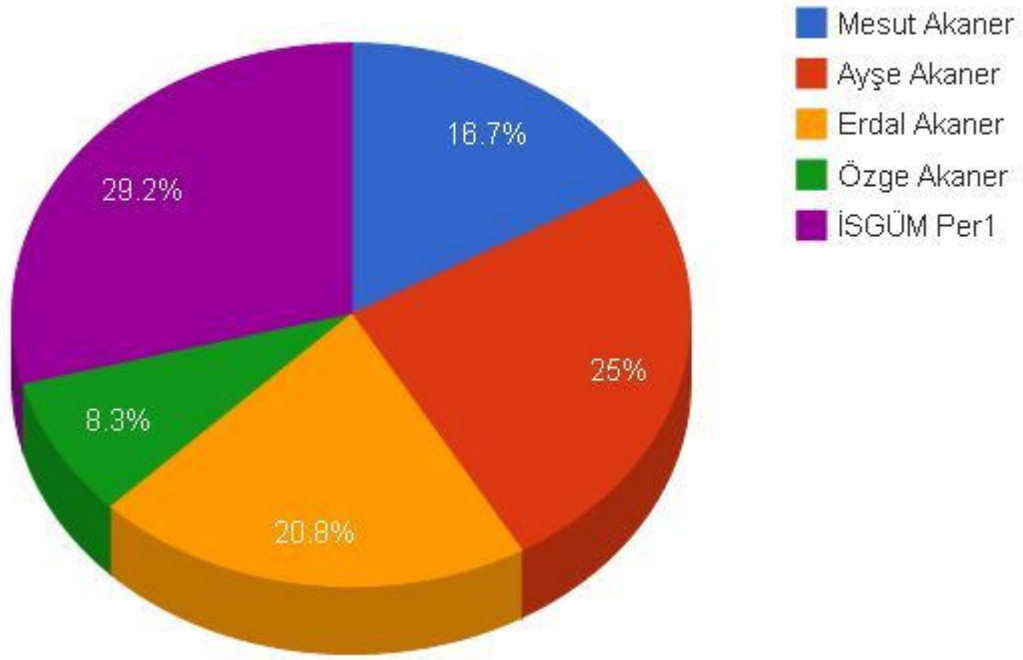
Mesut Akaner

Özge Akaner

Göster

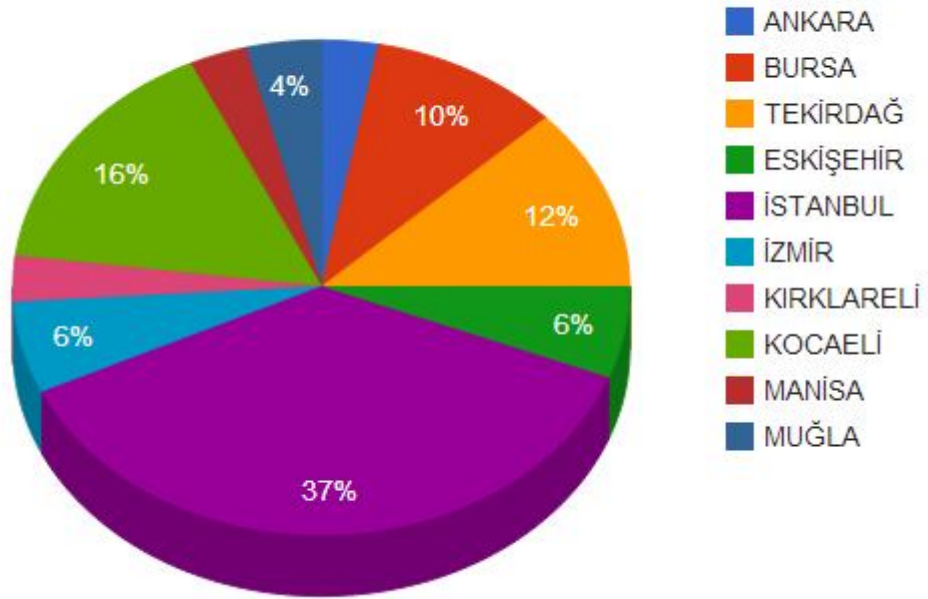


İncelenen Belgelerin Personele Göre Dağılımı



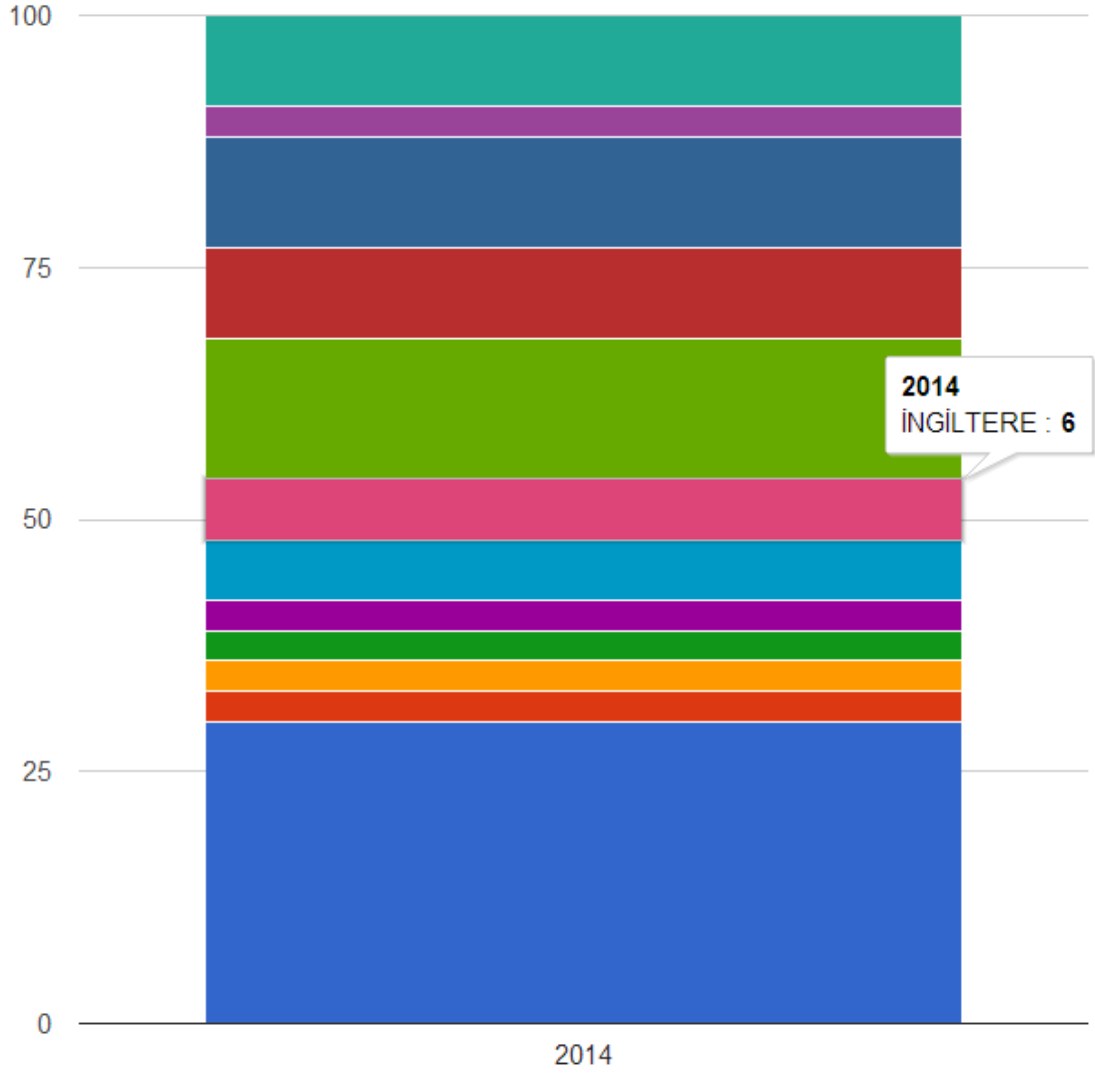
Örnek Veri Girişi Sonucu Elde Edilen Rapor Örnekleri

Başvuruların İllere Göre Dağılımı



İTHAL EDİLEN KİMYASALLARIN MENŞEİNE GÖRE DAĞILIMI

ALMANYA BELÇİKA ÇİN HIRVATİSTAN İSPANYA
İTALYA İNGİLTERE İSPANYA İSRAİL İTALYA
KORE MALEZYA



Değerlendirilen Kimyasallarda Risk İbarelerinin Görülme Oranları

