
GAZİ ÜNİVERSİTESİ
Gazi
1926

KURUMLAR ARASINDA WEB SERVİS ARACILIĞI İLE SUNULAN KİŞİSEL VERİLERİN YAZILIM GELİŞTİRİCİLERE KARŞI KORUNMASI

Mahmut ÇELİK

YÜKSEK LİSANS
BİLGİ GÜVENLİĞİ MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NİSAN 2021

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Mahmut ÇELİK

20/04/2021

KURUMLAR ARASINDA WEB SERVİS ARACILIĞI İLE SUNULAN KİŞİSEL VERİLERİN YAZILIM GELİŞTİRİCİLERE KARŞI KORUNMASI

(Yüksek Lisans Tezi)

Mahmut ÇELİK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Nisan 2021

ÖZET

Günümüzde bilgi teknolojilerinin getirmiş olduğu olanaklar sayesinde kurumlar vermiş oldukları hizmetlerin birçoğunu elektronik ortamda sunmaya başlamışlardır. Bu hizmetlerin doğası gereği bireylerin kişisel verilerine ihtiyaç duyulmaktadır. Bu ihtiyaçlar nedeniyle bireylerin kişisel verileri kurumların bilgi sistemlerinde depolanmakta ve işlenmektedir. Vatandaşlara etkin bir şekilde hizmet verilebilmesi amacıyla kurumların sahip oldukları kişisel verileri aralarında kanuna uygun bir şekilde paylaşılmasına izin verilmiştir. Kurumlar arasında veri paylaşımı genellikle web servisler aracılığı ile gerçekleştirilmektedir. Kurumların sahip oldukları yazılımların geliştirilmesinden sorumlu yazılım geliştiriciler geliştirdikleri yazılımlara web servisleri entegre edebilmek ve ortaya çıkan sorunları analiz edebilmek için web servis erişimine ihtiyaç duyarlar. Bu ihtiyaç dolayısı ile içerisinde kişisel verileri barındıran web servisler yazılım geliştiricilerin erişimine açılabilmektedir. Bu durum kişisel verileri insan faktörüne karşı savunmasız bırakmaktadır. Ulusal ve uluslararası düzenlemeler gereği kişisel verilerin korunması hukuki bir zorunluluktur. Bu zorunluluk sebebiyle kişisel verilerin insan faktörüne karşı korunabilmesi için bu tez kapsamında geliştirilen bir uygulama ile web servislerden gelen kişisel verilerin anonim hale getirilip sonrasında anonim verileri yazılım geliştiricilere sunarak kişisel verilerin korunması amaçlanmıştır. Geliştirilen bu uygulama ile yazılım geliştiricilerin çalışmalarına devam etmeleri engellenmeden kişisel verilere erişimlerinin önüne geçilmesi amaçlanmıştır. Tez kapsamında geliştirilen uygulama React ve Java programlama dilleri kullanılarak hazırlanmış web ara yüzünden ve kişisel verilerin anonimleştirilmesini sağlayan rest servislerinden oluşmaktadır.

Bilim Kodu : 92403
Anahtar Kelimeler : Bilgi güvenliği, kişisel verilerin korunması, java, maskeleme, anonimleştirme
Sayfa Adedi : 99
Danışman : Prof. Dr. Mustafa ALKAN

PROTECTION PERSONAL DATA SUBMITTED BY WEB SERVICES BETWEEN GOVERNMENT AGANCIES AGAINST SOFTWARE DEVELOPERS

(M. Sc. Thesis)

Mahmut ÇELİK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

April 2021

ABSTRACT

Thanks to the opportunities brought by information technologies, institutions have started to offer most of their services in electronic environment. Due to the nature of these services, individuals' personal data are needed. For this reason, personal data of individuals are stored and processed in the information systems of the institutions. In order to provide effective service to citizens, institutions are allowed to share their personal data in accordance with the law. Data sharing between institutions is usually carried out through web services. Software developers responsible for the development of the software owned by the institutions need web service access to integrate web services into the software they develop and to analyze the problems that arise. Due to this need, web services that contain personal data are opened to the access of software developers. This situation leaves the personal data vulnerable to the human factor. Protection of personal data is a legal obligation in accordance with national and international regulations. Due to this necessity, an application has been developed within the scope of this thesis in order to protect personal data against human factors. This application developed anonymizes personal data in web services. Anonymized personal data are presented to software developers, thus it is aimed to protect personal data. With this application, it is aimed to prevent software developers from accessing personal data without being prevented from continuing their work. The application developed within the scope of the thesis consists of a web interface prepared using React and Java programming languages and rest services that provide anonymization of personal data.

Science Code : 92403

Key Words : Information security, protection of personal data, java, masking, anonymization

Page Number : 99

Supervisor : Prof. Dr. Mustafa ALKAN

TEŞEKKÜR

Bu tezin hazırlanma sürecinde ve yüksek öğrenimim dönemi boyunca desteğini esirgemeyen saygıdeğer danışman hocam Prof. Dr. Mustafa ALKAN'a en içten teşekkür ve saygılarımı sunarım.

Tüm hayatım boyunca bilgisi ve tecrübeleriyle bana örnek olan annem Raziye ÇELİK'e ve babam Köksal ÇELİK'e, en iyi arkadaşım olan kardeşim Fatih ÇELİK'e, başarılı olmam için desteğini hiçbir zaman esirgemeyen değerli eşim Kübra ÇELİK'e ve hayatıma anlam veren, gözümün nuru, biricik kızım Hatice Meyra ÇELİK'e şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
A2A	Application-to-Application
AB	Avrupa Birliği
ACM	Association for Computing Machinery
AİHS	Avrupa İnsan Hakları Sözleşmesi
AK	Avrupa Konseyi
ANSI	American National Standards Institute
API	Application Programming Interface
ARPA	Advanced Research Project Agency
BDDK	Bankacılık Düzenleme ve Denetleme Kurumu
BGYS	Bilgi Güvenliği Yönetim Sistemi
BS	British Standard
BSI	British Standards Institute
BT	Bilgi Teknolojileri
CCTA	Central Computer and Telecommunications Agency
COBIT	Control Objectives for Information and Related Technology
CSS	Cascading Style Sheets
DES	Digital Encryption Standard
DNS	Domain Name System
FIPS	Federal Information Processing Standards
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
HTML	Hypertext Markup Language
IBM	International Business Machines
IEC	International Electrotechnical Commission
ISACA	Information Systems Audit and Control Association
ISO	International Organization for Standardization

Kısaltmalar	Açıklamalar
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSMF	Information Technology Service Management Forum
J2EE	Java 2 Platform Enterprise Edition
JDK	Java Development Kit
JPA	Java Persistence API
JPQL	Jakarta Persistence Query Language
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KVKK	Kişisel Verileri Koruma Kanunu
LAN	Local Area Network
MVC	Model-View-Controller
NIST	National Institute of Standards and Technology
OECD	Organisation for Economic Co-operation and Development
OGC	Office of Government Commerce
ORM	Object–Relational Mapping
REST	Representational State Transfe
RPC	Remote Procedure Call
SCCS	Source Code Control System
SDK	Software Development Kit
SQL	Structured Query Language
TBK	Türk Borçlar Kanunu
TCK	Türk Ceza Kanunu
TMK	Türk Medeni Kanunu
TS	Türk Standardı
TSE	Türk Standardları Enstitüsü
UI	User Interface
VCS	Version Control System
WSDL	Web Services Description Language
XML	Extensible Markup Language

1. GİRİŞ

Teknolojik gelişmeler ile birlikte bilgilerin elektronik ortamda aktarılması ve paylaşılması giderek artmaktadır. Ayrıca paylaşılan veriler dışında yeni üretilen veriler daha çok elektronik ortamda üretilmektedir. Paylaşılan bilgiler arasında bireylere ait veriler de bulunmaktadır. Bu bilgilerin kötü niyetli kişilerin eline geçmemesi ve korunması kişisel verilerin korunması kanunu gereği hukuki bir zorunluluk haline gelmiştir. Kurumlar hem dışardan gelecek saldıralara karşı hem de içeriden gelebilecek saldırılara karşı güvenlik önlemleri almaları gerekmektedir. Kurumlar sahip oldukları ve paylaştıkları verilerin güvenliği için veri korumaya yönelik önlemler almaktadırlar. Ayrıca kurumlar sahip oldukları kişisel verileri veri tabanlarında maskeleyip saklayarak ekstra tedbirler almaktadırlar.

Kurumlar vermiş oldukları hizmetlerin kalitesini artırmak, vatandaşlardan belge istemeyerek vatandaşların aldıkları hizmetlerdeki memnuniyetini artırmak, verilecek hizmetin hızını artırmak, belgelerde yapılacak sahteciliğin önüne geçmek için kurumlar birbirleriyle kişisel verileri paylaşmaktadır. Örneğin ehliyet alırken ikametgah belgesini vatandaşlardan değil, doğrudan Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü sisteminden otomatik getirilmesi gibi.

Kurumlar arasındaki veri paylaşımında genellikle web servis teknolojisi kullanılmaktadır. Web servisten getirilen veriler anlık olarak sistemler arasında sorgulanabilmekte bu sebeple kişilerin en güncel verilerine kurumlar erişebilmektedirler.

Kurumlar tarafından vatandaşların bilgilerinin işlendiği yazılımlar kullanılmaktadır. Yazılım geliştiriciler tarafından üretilen bu yazılımlar aracılığı ile vatandaşlara hizmet verilmektedir. Yazılım geliştiriciler üretmiş oldukları yazılımlara entegre edebilmeleri için web servislere erişim ihtiyacı duyarlar. Yazılım geliştiricilere gerçek verileri barındıran web servislere erişim yetkisi vermek kişisel verileri zafiyete açık hale getirmektedir. Ancak kurumların, yazılım geliştiricilere gerçek verilerin aktığı web servislere erişim yetkisi vermesinin birkaç sebebi vardır;

- Karşı kurumların her zaman test web servisi açmamaları,
- Bazı kişilerin verilerinden kaynaklı hatalar sebebiyle yazılım geliştiricilerin kişinin gerçek verileri ile test yapma ihtiyacı duyması,

- Test web servislerinin bütün olasılıkları kapsayacak nitelikte veri kümesi içermemesi.

Bu problemin önüne geçilebilmesi için bu tez kapsamında bir uygulama tasarlanmıştır. Tasarlanan bu uygulama ile kurumlardan alınan ve içerisinde kişisel verileri barındıran web servis cevap nesnesinin yapısı korunarak sadece aldığı değerlerin değiştirilmesi suretiyle kişisel veriler anonim hale getirilmiştir. Kişisel verilerin içeriği değiştirilirken geliştirilen uygulama ve kullanılan yöntem ile kişisel veriler değiştirilmekte böylece bir kişiye ait veriler her zaman aynı değeri almaktadır. Tezin ilerleyen bölümlerinde detaylı olarak anlatılan bu yöntemler ile kişisel veriler geriye döndürülemeyecek şekilde tasarlanmıştır.

Tasarlanan bu uygulama ile kişisel verilere kurum içerisinden gelebilecek olan tehditlere karşı korunmasında ekstra bir güvenlik önlemi alınmış olacaktır.

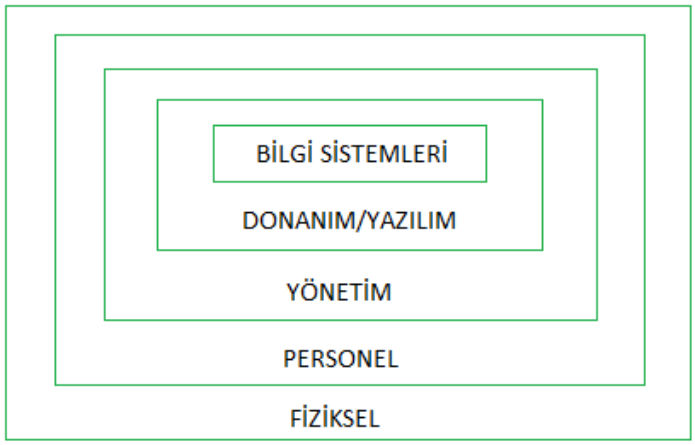
Bu tez çalışması altı bölümden oluşmaktadır. Birinci bölüm giriş bölümdür. İkinci bölümde bilgi güvenliğinin önemi ve bu konuda yapılan çalışmalardan bahsedilmiştir. Üçüncü bölümde tasarlanan uygulamada hangi teknolojilerin kullanıldığı ve neden bu teknolojilerin seçildiğinden bahsedilmiştir. Dördüncü bölümde tasarlanan uygulamanın bölümlerinde, kişisel veriler anonimleştirilirken kullanılan algoritmalardan bahsedilmiştir. Beşinci bölümde tasarlanan uygulamanın kullanımına dair bilgiler yer almaktadır. Altıncı bölümde uygulamanın detaylı kurulumundan bahsedilmiştir. Yedinci ve son bölümde ise tasarlanan uygulamanın kurumlara getireceği katkılardan bahsedilmiştir.

2. BİLGİ GÜVENLİĞİ VE KİŞİSEL VERİLERİN KORUNMASI

2.1. Bilgi Güvenliği

Kurumların sahip olduğu bilgi varlıklarının gizliliğine, bütünlüğüne ve erişilebilirliğine karşı oluşabilecek risklerin tanımlanıp, bu risklere karşı gerekli işlemlerin yapılması olarak tanımlanmaktadır [1]. Kurumların bilgi varlıklarına yetkisiz erişim, ifşa etme, kesintiye uğratma, değiştirme, inceleme, kaydetme veya imha olma gibi durumlara karşı koruma uygulamasıdır. Bilgi güvenliği amacı bilgi varlıklarını korumak olan bir süreçtir. Kurumların bilgi varlıklarını korumak amacıyla oluşturulmuş bilgi sistemleri güvenlik politikaları olmalıdır. Bilgi sistemleri güvenlik politikası, bir kuruluşun bilgi varlıklarını nasıl yönettiğini, koruduğunu ve bilgi sistemleri güvenlik altyapısı hakkında nasıl kararlar aldığını açıklayan, iyi tanımlanmış ve belgelenmiş bir kılavuzlar dizisidir [2]. Kurumlar karşılaşabilecekleri risklere karşı bilgilerin önemine, tehlikenin olabilirliğine ve gerçekleştiğinde etkisine göre aşağıdaki seçeneklerden birini tercih edilirler [1].

- Riskin kabul edilmesi,
- Riskin azaltılması
- Riskin transfer edilmesi
- Riskin kaynağının kaldırılması



Şekil 2.1. Bilgi güvenliğinin katmanları [3]

Bilgi güvenliği süreci iki ana unsuru içermektedir;

- Risk analizi
- Risk yönetimi

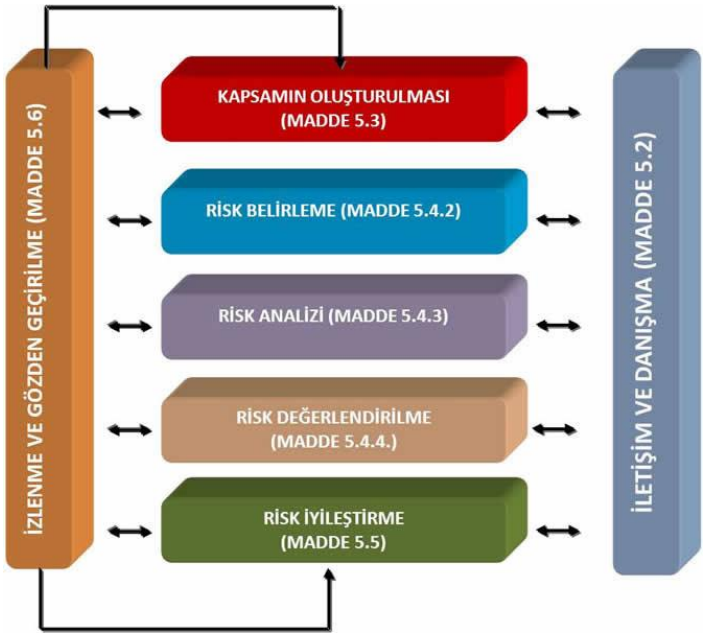
Risk analizi aşamalardan oluşmaktadır. Öncelikle tüm bilgi sistemlerinin envanteri çıkarılır. Her sistemin ayrı ayrı kurum için değeri belirlenir ve kurumun riske ne kadar maruz kalabileceği belirlenir.

Risk yönetimi, kurumların riske maruziyetinin kabul edilebilir bir seviyeye indirilebilmesi için kontrollerin ve güvenlik önlemlerinin seçilmesini içermektedir.
Bilgi güvenliğinin tanımı iki dizi parametreye dayanmaktadır [3];

- Tehditler, Zafiyet, Varlıklar ve Artık risk
- Gizlilik, Bütünlük ve Erişilebilirlik

Risk

Bilgi varlıklarının veya süreçlerinin zafiyetlerinden yararlanarak tehditlerin gerçekleşme olasılığına denir. Bilgi ve varlıkların korunabilmesi için risk yönetimi yapılmalıdır [4]. Risk yönetiminde yapılacak dört adım bulunmaktadır. Bunlar; Riskin kabul edilmesi, azaltılması, transfer edilmesi ve vazgeçilmesidir [5].



Şekil 2.2. Risk yönetimi [6]

Artık Risk

Risk kontrolleri ile varlığın doğasında var olan riskler azaltıldıktan sonra geri kalan eylem ve olaylarla ilişkili risk veya tehlike miktarıdır [7].

Varlık

Kurumlar için değeri olan, korunması gereken şeylerdir. Bilgi güvenliği özelinde belgeler, veriler, veri tabanları, yazılımlar, fiziksel bilgi teknoloji ürünleri, itibar varlık kategorisine girer. Kurumlar bilgi varlıklarını korumak için öncelikle varlık envanteri oluşturmaları gerekmektedir. Kurumlar bilgi varlıklarının kendileri için ne kadar değerli olduğunu ortaya çıkabilecek risklerden ne kadar etkileneceklerini hesaplaması gerekir. Bu sayede kaynaklar daha etkili bir şekilde kullanılabilir [8]. Aşağıdaki şekillerde varlıkların envanter çıkarımında kullanılabilecek örnek bir tablo ve varlıkların bilgi güvenliği için değerinin hesaplanma şekli yer almaktadır.

Sıra No:	Varlık Grubu	Varlık	Kategori	Varlık Sahibi	Varlık Kullanıcısı	Gizlilik Değeri	Bütünlük Değeri	Erişilebilirlik Değeri	Değer	Varlığın Eklenme Tarihi	Açıklama
1											
2											
3											

Şekil 2.3. Varlık envanter tablosu [8]

Güvenlik Hedefi	Düşük	Orta	Yüksek	Çok Yüksek
Gizlilik				
Bütünlük				
Erişilebilirlik				

Şekil 2.4. Varlık değeri tablosu [8]

Tehdit

Bir varlığa zarar verme olasılığı olan her şey tehdit olarak tanımlanmaktadır. Kurumların ortaya çıkabilecek veya çıkan tehditleri yakından takip etmesi ve bu tehditlere karşı risk analizi yaparak tehditleri tespit etmeleri gerekmektedir [8]. Kurumların sahip oldukları bilgilere karşı oluşabilecek tehditler bilgilerin değeri ile orantılıdır. Bilgiler değer kazandıkça oluşabilecek tehditlerin sayısı artacaktır [9]. Kurumların karşılaşabilecekleri tehditler aşağıdaki kategorilere ayrılabilir.

- İç kaynaklı tehditler (Kurum çalışanları)
- Dış kaynaklı tehditler (Saldırganlar)
- Doğal ve fiziksel tehditler (Sel, Deprem, Yangın vb.)
- Teknik iç tehditler (Bilgi sistemlerinde yaşanabilecek teknik sorunlar)

Zafiyet

Varlıkların ve sistemlerinin doğasındaki bazı zayıflıklara ve kötü niyetli kişilerce kötüye kullanılabilecek açıklıklara denir. Bilgi güvenliğini sağlayabilmek için zafiyetlerin yönetilmesi gerekmektedir [4].

Zafiyet Yönetimi: Kurumların sahip oldukları varlıkların zafiyetlerinin keşfedilmesi, tanımlanması, bu zafiyetlerle ilişkili risklerin tespit edilmesi ve değerlendirilmesi sonucu oluşabilecek risklerin düzeltilmesi ve ortadan kaldırılmasıdır. Zafiyetlerin yönetilmesinde tespit edilen riskler kurumun bilgi güvenliği politikasına aykırı değilse kurumun riski kabul etmesiyle zafiyet yönetimi sonuçlanabilir [10].

2.1.1. Bilgi güvenliği prensipleri

Gizlilik

Bilgiye sadece erişmesi gereken kişi, kurum ve sistemlerin erişimine izin verilmesi, diğer kişi, kurum ve sistemlerin erişememesidir [7]. Kurumların sahip olduğu bilgilerin kötü niyetli kişilerin eline geçmesinden dolayı oluşabilecek büyük zararlar dolayısı ile gizlilik büyük önem taşımaktadır.

Bütünlük

Verilerin yetkisiz kişilerce kötü niyetli veya kazara değiştirilmemesi, yok edilmemesi veya kaybolmaması olarak tanımlanır [11]. Bütünlüğün bozulmasıyla bilginin orijinalliği kaybolacağı için bilginin bir değeri kalmayacaktır.

Erişilebilirlik

Yetkili bir varlık tarafından ihtiyaç anında erişilebilir ve kullanılabilir bir sisteminin özelliğini tanımlar [11]. Bilginin gizlilik, bütünlük ve erişilebilirlik özelliklerinden biri bile sağlanamadığında bilgi varlıkları risk altındadır.

2.1.2. Bilgi güvenliği standartları

Bilgi güvenliği devamlılık gerektiren bir süreçtir kurumlarda bu sürecin sağlıklı sürdürülebilmesi için bilgi güvenliği standartları çerçevesinde yönetilmesi gerekir [12]. Kurumların bilgi güvenliğinde kullanabilecekleri uygulayabileceği çeşitli standart türleri Çizelge 2.1'de gösterilmiştir.

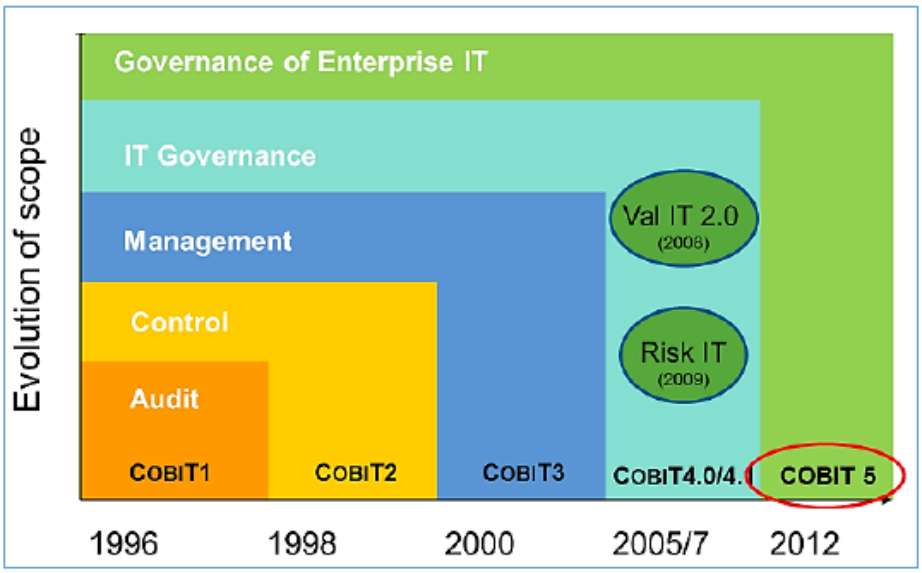
Çizelge 2.1. Bilgi güvenliği standartları [3]

İSİM	KAYNAK	TARİH	GEREKLİLİK
COBİT	ISACA	1996	YOK
ITIL	International	1989	YOK
NIS SP 800/30	NIST	2002	YOK
ISO 1335-2 (ISO 27005)	BT Güvenliği Yönetimi için Yönergeler	1996	Standart
ISO 15408	Yaygın Ölçütler	1996	Sertifika
ISO 27001	BS 7799-2'nun yeni versiyonu	2005	Sertifika
ISO 27002	17799 ve 7799-1'in yeni versiyonu	2007	Standart

Cobit (bilgi ve ilgili teknolojiler için kontrol hedefleri)

BT yönetimini iyileştirmek ve yönetimi kolaylaştırmak amacıyla kurumlar en iyi uygulama çerçevelerini kullanmaktalar. Bu uygulama çerçevelerinden biri olan Cobit, denetim faliyetlerini, süreçleri ve oprasyonların ölçümü ve dökümantasyon hazırlamada kurumlarda neler yapılması gerektiğine dair yönergeler sunmaktadır [13]. Ülkemiz de ise yaygın olarak kullanılamasına BDDK'nın Bankacılık Bilgi Sistemini denetlemek amacıyla kullanılması öncülük etmiştir. Cobit'in en güncel sürümü 5.0 sürümüdür. Ancak BDDK şu an da COBİT 4.1 sürümünü kullanmaktadır [8].
Yayınlanan bu standardın amaçları [4];

- BT yöneticilerine, kullanıcılarına ve denetçilerine iş hedeflerinin bilgi işlem hedeflerine dönüşümünü sağlamak,
- Gerekli kaynakları ve süreçleri düzenlemek
- BT altyapılarını etkin kullanılmasını sağlamak
- Güvenlik kontrol seviyelerine karar verilmesini kolaylaştırmak,
- Etkinliği denetlemek,
- Verimliliği/Efektifliği denetlemek,
- Bilginin bozulmaması ve bütünlüğünün korunmasını sağlamak
- Bilgi varlıklarını korumaktır.

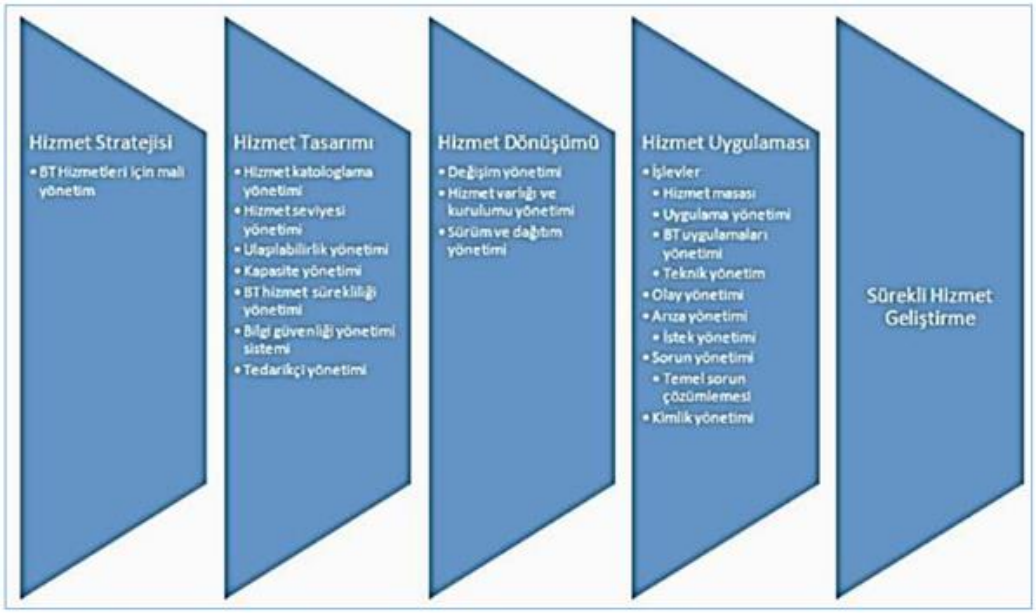


Şekil 2.5. Cobit tarihçesi [14]

Itıl (bilgi teknolojisi altyapı kütüphanesi)

Bilgi Teknolojileri Altyapı Kütüphanesi (ITIL), BT organizasyonunu geliştirmek amacıyla 1989'da Birleşik Krallığın eski Merkezi Bilgisayar ve Telekomünikasyon Kurumu (CCTA) tarafından oluşturulmuştur. Şuanda, İngiltere'nin Devlet Ticaret Odası (OGC) tarafından yönetilmekte ve BT Hizmet Yönetimi Forumu (ITSMF) tarafından desteklenmektedir [15]. Bu standart, BT hizmet yönetimi ve ilgili süreçler için kapsamlı ve tutarlı bir en iyi uygulama çerçevesini sunmaktadır. ITIL çerçevesinin amacı, kuruluşların kendi uygulamalarında kullanabilecekleri yaklaşımlar, işlevler, rol ve süreçleri tanımlamak ve genel olarak rehberlik etmektir [16]. ITIL'ın amaçları aşağıda sıralanmıştır.

- Hizmet maliyetlerini düşürmek
- Erişilebilirliği arttırmak
- Hizmet yönetimi odaklı tecrübeleri toplamak ve paylaşmak,
- Hizmet yönetiminde en iyi uygulamaları bir araya getirmek
- Ölçeklenebilirliği arttırmak
- Servis güvenliği, erişilebilirliği ve sürekliliğini sağlamak
- Kaynakların verimli kullanılmasını sağlamak,
- Kurumun amaçlarına ulaşmasında BT'ye yardımcı olmak



Şekil 2.6. Itil süreçleri [14]

Nist 800-26

Kurumların güvenlik uygulamalarını temel aldığı bir kontrol standardıdır. NIST, ABD Ticaret Bakanlığı İdaresi bünyesinde federal bir kurumdur. Nist, bilgi sistemleri güvenliğini iyileştirmek için hem Federal kurumlar hem de ticari kuruluşlar tarafından benimsenen çeşitli standartlar geliştirmiştir. NIST 800-3 / 26'nın amacı, federal hükümetin yürütme kurumlarını destekleyen bilgi sistemleri için güvenlik kontrollerinin seçilmesi ve belirlenmesi için kılavuzlar sağlamaktır. Yönergeler, federal bilgileri işleyen, depolayan veya ileten bir bilgi sisteminin tüm bileşenleri için geçerlidir [17].

ISO/IEC 27001

ISO 27001, 1993 yılında BSI (British Standarts Institution) tarafından geliştirilmeye başlanmış olan bu standart, ilk olarak BS 7799-1 olarak aynı yıl içerisinde ilk sürümü yayınlanmıştır. 1998 yılında BS 7799-2 isimli ikinci sürümü yayınlanmıştır. 2000 yılında ISO (The International Organization for Standardization) ve IEC (The Electrotechnical Commission) kurumları ortak bir ekip ile ISO/IEC 17799 standardını geliştirmişlerdir.
Bu standart içerisinde bilişim güvenliği ile alakalı 10 bölümden oluşan 127 kontrol maddesi yer almaktadır. Bu maddeler kurumlarda bilgi güvenliğinin sağlanabilmesi için gereken kuralların oluşturulmasını yardımcı olmayı amaçlamaktadır. 2005 yılında ISO/IEC 17799 ISO/IEC 27001:2005 sürümü olarak yayınlanmıştır. Ülkemizde TSE tarafından çevirisi yapılmış ve TS ISO/IEC 27001:2005 olarak Türk standardı olarak kabul edilmiştir. 2013 yılında son sürümü yayınlanmıştır.

ISO/IEC 27001 standardı kurumlarda bilgi güvenliği yönetim sistemlerinin kurulmasında ve oluşturulmasında kontrol maddelerinin bulunmasıdır. Kontrol maddelerinin ne olması gerektiği konusunda detaya girmeden cevap vermektedir. ISO/IEC 27001 standardının üç temel özelliği bulunmaktadır bunlar; Ölçülebilirlik, Tekrarlanabilirlik, Ölçeklenebilirlik'dir [8].

- Ölçülebilirlik: ISO/IEC 27001 standardının üçüncü taraflar tarafından değerlendirebilir olması özelliğidir.
- Tekrarlanabilirlik: Bilgi Güvenliği Yönetim Sistemi (BGYS) çok sayıda kontrol içermektedir bu kontrollere bağlı kalarak istenildiği kadar tekrar edebilir. PUKÖ (Planla- Uygula-Kontrol Et – Önlem Al) döngüsü sürekli tekrar edilebilir.

- Ölçeklenebilme: BGYS, kurumun belirli bölümlerinde oluşturulabilir daha sonra gerekli olması durumunda kurumun tamamı için oluşturulabilir. Kapsam ile ilgili istendiği zaman değişiklik yapılabilir. Ek denetimler eklenebilir veya kurumun ihtiyaçlarına göre azaltılabilir. Bu da ISO/IEC 27001'in ölçeklenebilir olma özelliğidir [8].

ISO/IEC 27001 Püko Yaklaşımı

ISO/IEC 27001 standardı kurumlardaki bilgi güvenliği yönetim sistemlerinin kurulmasında "Planla-Uygula-Kontrol Et-Önlem Al" yöntemi esas alınmaktadır.

- Planla: BGYS'nin kurulmasında, kurumun bilgi güvenliği ihtiyaçlarının belirlenmesinde ve hedeflerinin tespit edilmesi, bilgi güvenliği politikalarının oluşturulması, süreçlerin ve prosedürlerin belirlenmesi planla adımında gerçekleştirilir.
- Uygula: Bilgi güvenliği yönetim sisteminin oluşturulması, Güvenlik politikalarının, süreçlerin ve prosedürlerin işletilmesi ve denetimlerin yapılması bu adımda gerçekleştirilir.
- Kontrol Et: Bilgi güvenliği yönetim sistemi oluşturulduktan sonra güvenlik politikalarının ve alınan önlemlerin etkinliğinin tespit edilmesi bu adımda gerçekleştirilmektedir.
- Önlem Al: Kontrol çalışmaları ve denetimler sonucu ortaya çıkan eksikliklerin giderilmesi, BGYS'nin eksikliklerinin tamamlanması bu adımda gerçekleştirilmektedir.

ISO / IEC 27002: 2005 (bilgi güvenliği yönetimi için uygulama kodu)

British Standards Institute (BSI) tarafından oluşturulan BS7799-1'den yararlanılan uluslararası bir standarttır. ISO / IEC 27002: 2005, kurumsal güvenlik standartları oluşturmak ve etkili bir bilgi güvenliği yönetimi geliştirmek için ortak kılavuz olarak tasarlanmıştır.

ISO / IEC 27002: 2005 standardı, aşağıda belirtilen 10 alanda çeşitli yönergeleri ve en iyi uygulama örneklerini içerir [3].

- Güvenlik Politikası
- Bilgi güvenliği organizasyonu
- Varlık yönetimi
- İnsan kaynakları güvenliği
- Fiziksel ve çevresel güvenlik
- İletişim ve operasyon yönetimi
- Erişim kontrolü
- Bilgi sistemleri edinimi, geliştirilmesi ve bakımı
- Bilgi güvenliği olay yönetimi
- İş sürekliliği ve yönetimi
- Uyumluluk

Hipaa

1996 yılında Amerikan Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (HIPAA), sağlık hizmeti sunucularında saklanan hasta bilgilerinin; gizlilik, güvenlik ve elektronik işlem standartlarını belirleyen bir yasadır [18].

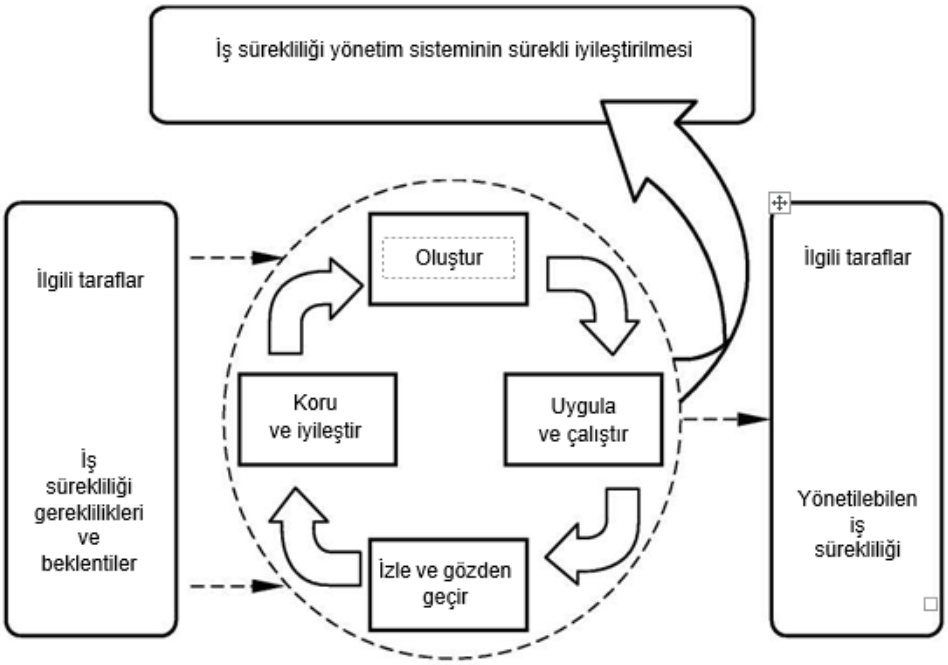
Bu yasaya göre;

- Yetkisiz erişimlerin ve kötüye kullanımların tespit edilmesi ve engellenmesi.
- Şüpheli işlemlerin izlemesi.
- Bilgi teknolojilerinde çıkabilecek sorunların gidermesinde ve ağ operasyonlarında kullanılmak üzere logların yönetilmesi gerekmektedir [19].

BS 25999

İş Sürekliliği Yönetim Sisteminde kullanılmak üzere uygulama kurallarını belirleyen standarttır. BS 25999 standardının 2007 yılında yayınlanan versiyonunda, kritik iş süreçlerinin felaket durumlarında sorun çıkmaması için bir yönetim sistemi sunmaktadır. İş

sürekliliği yönetim sistemi kurumlar için büyük etkileri olabilecek risklerin yönetilebilmesini hedeflemektedir. BS 25999 aynı ISO 27001 standardı gibi risk temelli bir standarttır [20].



Şekil 2.7. İş sürekliliği yaşam döngüsü [21]

2.2. Veri Kavramı

Veri ölçüm veya sayım yolu ile toplanan işlenmemiş gerçek enformasyon parçacığına verilen addır. Sayısal bir değer bildiren verilere nicel sayısal değer bildirmeyen verilere nitel veri olarak isimlendirilmektedir [22]. Bilişim sistemlerinde elektronik olarak toplanmış bilgi, işaret veya göstergeler anlamına gelmektedir [23].

2.2.1. Veri türleri

Günümüzde bilgisayarlar sorunlara çözüm üretebilmek amacıyla verilere ihtiyaç duyarlar. Bilgisayar programları verileri girdi olarak kullanarak bu verileri işlerler. İşleme sonucunda son kullanıcıya işlenmiş veri yani bilgi sunulur. Sayısal veri, Alfanümerik/Karakter Veri ve Mantıksal Veri olmak üzere üç çeşit veri türü bulunmaktadır.

- Sayısal Veri: Hesaplama işlemlerinde kullanılan pozitif negatif ve tüm reel sayıları kapsayacak şekilde tüm sayı tiplerine sayısal veri denilmektedir.

- Alfanümerik/Karakter Veri: Karakter veriler özel karakterler, harfler ve harflerin kombinasyonundan oluşan dizelere karakter verileri denilmektedir. Örneğin: “a”, “#”, “bilgisayar” gibi.

- Mantıksal Veri: Doğru ve Yanlıştan oluşan bu veri tipi karar verme süreçlerinde kullanılmaktadır. [24]

2.2.2. Veri yapı türleri

Teknolojik gelişmeler sayesinde genişletilebilir veri çeşitliliği ortaya çıkmıştır. Bu çeşitlilik yapısal özelliklerine göre yapısal, yarı yapısal ve yapısal olmayan veriler olmak üzere üç bölüme ayrılmıştır [23].

Yapısal veriler

Yapısal veriler önceden tanımlanmış ve etkili analizler için düzenlenmiş verilerdir. Genellikle ilişkisel veri tabanlarında saklanabilen veriler yapısal verilerdir. Yapısal veriler belirli bir veri modeline uygun tasarlandıkları için analiz edilmesi kolaydır [25].

Yapısal Verilerin Özellikleri:

- Belirli bir veri modeline uygun yapıdadır.
- Satırlar ve sütunlar biçiminde saklanır.
- Veri öğeleri adreslenebilir olması özelliği ile kolay analiz edilebilir.
- Sorgusu kolay olduğu için diğer programlar tarafından kullanılabilir.
- Aynı özelliklere sahip veriler gruplandırılır [23].

Yarı Yapısal Veriler

Herhangi bir veri modeline uymayan fakat belirli bir yapıda olan verilere yarı yapısal veri denir. Yarı yapısal verilerin bazıları veri tabanlarında saklanabilir ancak bazı yarı yapısal

veriler için bu işlem zor olabilmektedir. Genişletilebilir İşaretleme Dili (XML), e-postalar ve NoSQL veri tabanları yarı yapısal verilere örnek verilebilir.

Yarı Yapısal Verilerin Özellikleri:

- Satır ve Sütun şeklinde depolanmazlar.
- Verileri gruplandırmak için etiketler kullanılır.
- Benzer varlıklar birlikte gruplandırılır.
- İyi tanımlanmış bir yapıya sahip olmadıkları için diğer programlar tarafından kullanımı zordur [23].

Yapısal Olmayan Veriler

Önceden tanımlanmamış veya yapılandırılmamış veri modeline sahip verilere yapısal olmayan veri denir [25]. Belirli bir veri modeline sahip olmadıkları için yapısal olmayan veriler ilişkisel veri tabanları için uygun değillerdir. Bu tür verileri depolamak ve yönetmek için alternatif platformlar mevcuttur. Yapısal olmayan verilere görüntüler (JPEG, GIF, PNG, vb.), videolar ve sunumlar örnek verilebilir [23].

Yapısal Olmayan Verilerin Özellikleri

- Herhangi bir veri modeli ve tanımlanmış bir yapısı yoktur.
- Satır ve sütun şeklinde depolanmazlar.
- Veri öğeleri adreslenemediği için kolay analiz edilemezler.
- Erişim ve sorgu yapılabilir bir yapıda olmadığı için diğer programlar tarafından kullanımı zordur [23].

2.3. Hassas Veri Kavramı

Ulusal ve uluslararası düzenlemelerde kişisel verilerin işlenmemesi genel bir kural olarak kabul görmektedir ancak istisna olarak kişisel verilerin işlenmesi gerekir. Bu düzenlemelerde hassas veri adında bir kategori oluşturulmuştur [26]. Hassas veriler daha çok korunması gereken verilere verilen bir addır. Hassas verilerin daha çok korunmasının

sebebi kişisel verilerin işlenmesi sonucunda telafisi olmayan sonuçların ortaya çıkma riskidir [27].

Kişisel veriler ile alakalı hazırlanan düzenlemelerde genel olarak "hassas veri" kavramı kullanılmaktadır bunun yanında "özel kategorili kişisel veriler", "özel kişisel veriler" ve hatta "özel korumaya layık olan veriler" kavramları da kullanılmaktadır [28].

Çizelge 2.2'de hassas verileri KVKK, TCK 135. Madde, AB 95/46/EC Direktifi 8. Madde ve AK 108 sayılı Sözleşme 6. Madde karşılaştırmalı bir şekilde gösterilmiştir.

Çizelge 2.2. Hassas veri kavramı [27]

KVKKT 7. Madde	TCK 135. Madde	AB 95/46/EC Direktifi 8. Madde	AK 108 sayılı Sözleşme 6. Madde
Irk	Irki Köken	Irki, etnik köken	Irki Köken
Siyasi Düşünce	Siyasi Görüş	Siyasi Görüş	Siyasi Düşünce
Felsefi İnanç	Felsefi Görüş	Felsefi İnanç	Diğer İnançlar
Dernek,Vakıf, Sendika Üyeliği	Sendikal bağlantı	Sendika Üyeliği	
Sağlık	Sağlık Durumu	Sağlık Durumu	Sağlık Durumu
Özel Yaşam			
	Cinsel Yaşam	Cinsel Yaşam	Cinsel Yaşam
	Ahlaki eğilim		
Her turlu mahkumiyet		Ceza Mahkumiyeti	Ceza Mahkumiyeti

2.4. Kişisel Veri

Uluslararası yayınlarda kişisel veri; OECD'nin 1980 yılında Kişisel Verilerin Sınır aşan Trafiği ve Verilerin Korunmasına İlişkin Kılavuz'unda kişisel veri kavramını: "Belirli veya belirlenebilir bir gerçek kişiye ait tüm veriler" şeklinde tanımlamıştır [29]. Avrupa Konseyinin 1981 yılında yayınlamış olduğu Kişisel Verilerin Otomatik İşleme Tabi Tutulması Karşısında Şahısların Korunmasına Dair Sözleşmesinde "Kimliği belirtilen veya

belirlenebilen gerçek kişiye ait tüm veriler" şeklinde tanımlanmıştır. Avrupa Birliği tarafından 1995 yılında çıkarılan 95/46 sayılı yönergede "Bir gerçek kişinin belirli veya belirlenebilir olması koşulu, kişiye ait bir numara ya da fiziksel, kültürel, ekonomik veya sosyal kimliğini ifade eden bir veya daha fazla unsura doğrudan veya dolaylı olarak belirlenebilir olmasını ifade eder." [30].

Kişisel veri kavramlarına baktığımızda verinin sahibinin belirli olması veya belirlenebilir olması, gerçek kişi olması, her türlü veriyi içerebilir olması ve kişiye ait olması ortak kavramlardır. Kişinin belirlenebilir olması sahip olunan veriler ile kişiyi doğrudan göstermesi veya yapılacak çalışmalar ile kişiye ulaşıla bilinmesini ifade eder. Herhangi bir kişiyle ilişkilendirilemeyen hiçbir veri kişisel olarak kabul edilemez [27].

2.4.1. Kişisel veriler ve nitelikleri

Kişisel verilerin korunması kanununda korunması amaçlanan kişisel veriler, gerçek kişiye ilişkin verilerdir. Kişisel verilerin korunmasındaki amaç kişilerin ırkları fark etmeksizin kişilerin özel hayatın gizliliği başta olmak üzere temel hak ve hürriyetlerini korumaktır [31].

Kişinin kimliğinin belirlenebilir olması kişisel verilerin bir özelliğidir. Kişilerin kimliğinin belirlenebilmesi, diğer kişilerden ayırt edilmesini ve bir birey olarak tanımlanmasını sağlayan unsurlardır. Kişilerin ad, soyadı, uyruğu, doğum yeri, doğum tarihi, vatandaşlık numarası, fiziksel özellikleri, fotoğrafı ve biyometrik verileri kişinin kim olduğunun belli olmasını sağlayan verileridir [31].

Bilgi sistemlerine depolanan kişisel veriler birçok şekilde oluşabilmektedir. Bunlar [32];

- Kişilerin kendileri tarafından oluşturulmuş ve rızaları ile paylaştıkları veriler (Sosyal medya paylaşımları)
- Kişilerin eylemlerinin kaydedilmesiyle tutulan veriler (Cep telefonlarını kullanırken ki konum verileri)
- Kişilerden gönüllü veya gözlemlenerek elde edilen verilerin analizi ile ortaya çıkan veriler (Kişilerin kredi puanları gibi)

Kişisel verileri niteliklerine göre özel nitekli ve genel nitelikli veriler olmak üzere sınıflandırılabilir. Özel nitekli veriler genel nitelikli verilere göre daha hassas bilgiler içerdiği için KVKK ve GDPR gibi düzenlemelerde yer alması daha öncelikli ve önemlidir [23].

- Özel Nitelikli Veriler: KVKK/GDPR düzenlemelerinde kişiler için risk oluşturabilecek ve daha detaylı koruma gerektiren veriler olarak kategorize edilmiştir. Bu kategorideki veriler hassas veri olarak kabul edilmektedir. Etnik kökene ait bilgiler, Politik görüşler ve dini görüşler, sendika üyeliği bilgileri, biyometrik veriler, cezai mahkumiyet ile ilgili veriler, sağlık ile ilgili bilgiler bu kapsamda değerlendiren verilerden bazılarıdır.

- Çocuklara Ait Kişisel Veriler: Hem KVKK hem de GDPR da tüm bireylerin yaş ayrımı yapmaksızın kişisel verilerini güvence altına almıştır. Ancak yaşları itibari ile hakları üzerinde daha az farkındalığı olacağı için GDPR düzenlemelerinde çocuklara daha ayrıcalıklı ve güçlendirilmiş koruma sağlamaktadır.

- Cezai Mahkumiyet ve Suçlarla İlgili Kişisel Veriler: GDPR düzenlemelerinde bu tür verilerin işlenmesi *"resmi otoritenin kontrolü altında veya işlemin veri konularının hak ve özgürlükleri için uygun güvenceleri sağlayacak Birlik veya Üye Devlet yasaları tarafından yetkilendirilmesi"* şartına bağlanmıştır.

- Biyometrik Veriler: KVKK/GDPR düzenlemelerinde biyometrik verileri kişiye ait fiziksel, fizyolojik veya davranışsal özellikleri ile ilgili verilerin tamamı olarak tanımlanmıştır. GDPR biyometrik verilerin korunması ile ilgili düzenlemeler içermesine rağmen her ülke kendi ihtiyaçlarına göre düzenlemeler yapabilmektedir. Biyometrik verileri bedensel özelliklere ilişkin veriler ve davranışsal özelliklere ilişkin veriler olarak gruplandırabiliriz. Bedensel özelliklere ilişkin veriler parmak izi, yüz bilgisi gibi kişinin biyolojik özelliğine ilişkin verilerdir. Davranışsal veriler kişinin benzersiz tanımlanmasına neden olacak davranışsal özelliğidir.

2.4.2. Kimliği belirli veya belirlenebilir kişi kavramı

Verilerin korunmasındaki genel görüş kişisel verilerin korunması görüşü yaygındır ve bu doğrultuda ülkemizdeki mevzuatta bu yönde oluşturulmuştur. Tüzel kişilerinde kişisel verilerinin korunması gerektiğini savunanlar tüzel kişilerinde aslında gerçek kişiler ile ilişkilendirilebileceği görüşünü savunmuşlardır [33].

95/46/EC sayılı Direktif'inin 29. maddesinde, *"Koruma esasları, tespit edilmiş veya tespit edilebilir bir kişiye ilişkin herhangi bir bilgiye uygulanmalıdır; bir kişinin tespit edilebilir olup olmadığını belirlemek için, adı geçen şahsı tespit etmek için herhangi bir diğer kişi tarafından veya denetleyici tarafından kullanılabilecek makul tüm araçlar dikkate alınmalıdır; koruma esasları, veri öznesinin artık tespit edilebilir olmadığı bir biçimde anonimleşmiş verilere uygulanmayacaktır; madde 27'nin anlamı dâhilindeki davranış kuralları, verilerin anonimleşebilmesine ve veri öznesinin tespitinin artık mümkün olmadığı bir biçimde alıkonma biçimlerine dair rehberlik sağlamak için yararlı bir araç olabilir."* Eldeki imkanlar kullanılarak kişinin kim olduğu tespit edilebilen bütün bilgiler kişisel bilgi kabul edilir [34]. Başka bir ifade ile kişilerle ilişkili bilgilerin kişiyle doğrudan tanımlaması veya bilginin işlenmesi sonucu kişiyi tanımlayacak hale gelmesine sebep olabilecek bilgiler kişisel bilgi olarak kabul edilmektedir [34].

2.5. Kişisel Verilerin Korunması

KVKK/GDPR, korunması gereken maddi ve manevi her şeyi yazılı düzenlemeler ile koruma altına almayı amaçlamaktadır. Teknolojinin gelişmesiyle beraber kişisel verilerin daha kolay elde edilmesi, depolanması ve kullanılması kolaylaşmıştır. Bu durum kişisel verilen kullanımının hukuki çerçevede olduğu değerlendirilmiş, çoğu zaman ise kötüye kullanılarak verinin sahibine zarar verildiği değerlendirilmesinde bulunulmuştur. Teknolojinin kişisel verilerin kötüye kullanılmasını kolaylaştırması başta bireyler olmak üzere sivil toplum kuruluşlarını, hükümetleri, şirketleri ve uluslararası kuruluşları önlem almaya zorlamıştır [27].

Kişisel verilerin korunması ile kişisel verilerin sahibi kişilerin korunması ayrımı yapılmalıdır. Bu konuda yapılan düzenlemelerin çoğu kişilerin temel hak ve hürriyetini koruma altına almaya çalışmaktadır. Verilerin korunması ise hukuk düzeninde sadece araç

olarak görülmektedir. Veri güvenliği kişilerden çok verilerin korunmasını amaçlamaktadır. Verilerin korunması kişiler ile ilişkisi ölçüsünde kişinin korunmasına da hizmet etmektedir [35]. Örnek vermek gerekirse hakkında gizlilik kararı çıkarılan vatandaşların verilerinin korunması kişinin belki de hayatının tehlike altına girmesinin önüne geçecektir.

KVKK/GDPR uyumluğu sürecinde kurumlar sahip oldukları verilerden öncelikli kullanımda, hareketli ve beklemede olan verilerin içeriklerini analiz etmeleri gerekmektedir. Analiz işleminden sonra sahip olunan kişisel verilerin korunması sağlayacak teknik altyapılar ve iş süreçleri hayata geçirilmelidir [23].

Kişisel verilerin korunmasına yönelik alınabilecek önlemler;

- Kişisel verilerin şifrelenmesi veya bulanıklaştırılması,
- Veri işleme sistemlerinin gizliliğinin, bütünlüğünün, erişilebilirliğinin ve esnekliğinin sağlanması,
- Kişisel verilere erişimlerin etkin bir şekilde denetlenmesi,
- Kişisel verilerin güvenliğinin sağlanabilmesi amacıyla alınan teknik ve kurumsal önlemlerin etkinliğinin düzenli olarak test edilmesi ve değerlendirilmesi [23].

2.5.1. Kişisel verilerin korunmasının önemi ve amacı

Özel hayata ilişkin haklar, AİHS 8. Maddede "Özel ve aile hayatına saygı hakkı" başlığı altında düzenlenmiştir. Herkes özel ve aile hayatına, konutuna ve yazışmasına saygı gösterilmesi hakkına sahiptir. Anayasanın 20. Maddesine göre bu hak çekirdek hak olarak düzenlenmiştir [36].

Anayasada özel hayatın ve haberleşme gizliliğine dokunulmamasının tanımı, gizliliğin kapsamı açıkça düzenlenmemiştir. Sadece bu haklara sadece mahkeme kararı ile dokunulabileceği düzenlenmiştir [37]. Özel hayatın gizliliğine anayasa tarafından sağlanan güvence sayesinde kişilere yönelik ayrımcılığa sebep olacak bilgilere erişilmesinin önüne geçilecektir [38].

2.5.2. Kişisel verilerin korunmasında uygulanan kural ve ilkeler

KVKK'nın dördüncü ve beşinci maddesinde kişisel verilerin işlenme şartları ve ilkeleri düzenlenmiştir. Kişilerin açık rızası aranmaksızın kişisel verilerin işlenmesinin mümkün olduğu haller [39];

- Kanunlarda açıkça öngörülmesi: Kanunlarda kişisel verilerin işlenmesine cevaz veren hükümler bulunduğu takdirde kişisel veriler işlenebilecektir.
- Fiilen kişinin rızasını açıklayamayacak durumda olması veya rızası hukuken geçersiz olan kişilerin kendisi veya başkasının hayat bütünlüğünün korunması amacıyla.
- Sözleşmelerin kurulması veya yürütülebilmesi kaydıyla, sözleşme taraflarına ait kişisel verilerin işlenmesinin gerekli olması durumunda.
- Veri sorumlusunun hukuki gereklilikleri yerine getirebilmesi için zorunlu olması durumunda. Örnek olarak işveren çalışanına maaş ödemesi yapabilmesi için hesap numarası, medeni hali, bakmakla yükümlü olduğu kişiler gibi bilgileri işlemesi gerekliliği.
- Kişinin kendisi tarafından alenileştirilmiş olması durumunda.
- Hakların tesisi, kullanılması veya korunması için kişisel verilerin işlenmesinin zorunlu olduğu durumda. Örneğin temyiz kudreti olmayan kişilerin veya kısıtlı kişilerin yerine vasisinin o kişinin mali bilgilerinin tutulması gibi.
- Kişilerin temel hak ve hürriyetleri zarar görmemesi koşuluyla, veri sorumlusunun meşru menfaatleri için veri işlenmesinin zorunlu olması durumunda.

Özel nitelikli kişisel verilere ilişkin KVKK 6. Maddesine göre "*sağlık ve cinsel hayat dışındaki kişisel veriler, kanunlarda öngörülen hâllerde ilgili kişinin açık rızası aranmaksızın işlenebilir. Sağlık ve cinsel hayata ilişkin kişisel veriler ise ancak kamu sağlığının korunması, koruyucu hekimlik, tıbbî teşhis, tedavi ve bakım hizmetlerinin yürütülmesi, sağlık hizmetleri ile finansmanının planlanması ve yönetimi amacıyla, sır saklama yükümlülüğü altında bulunan kişiler veya yetkili kurum ve kuruluşlar tarafından ilgilinin açık rızası aranmaksızın işlenebilir.*" [39].

2.5.3. Kişisel verileri koruma ihtiyacı

6698 sayılı yasanın genel gerekçesinde, kişisel verilerin korunmasındaki ihtiyacın ne olduğuna dair bir ön giriş bulunmaktadır. Buna göre; *"...Kişisel verilerin korunması konusu 1980'li yıllardan itibaren uluslararası belgelerde yer almaya başlamıştır. İlk olarak, ülkemizin de üyesi bulunduğu, İktisadi İş birliği ve Kalkınma Teşkilatı (OECD) tarafından 23/9/1980 tarihin- de "Kişisel Alanın ve Sınır Aşan Kişisel Bilgi Trafiğinin Korunmasına İlişkin Rehber İlkeler" kabul edilmiştir. Avrupa Konseyi tarafından, tüm üye ülkelerde kişisel verilerin aynı standartlarda korunması ve sınır ötesi veri akışı ilkelerinin belirlenmesi amacıyla hazırlanan 108 sayılı "Kişisel Verilerin Otomatik İşleme Tabi Tutulması Karşısında Bireylerin Korunması Sözleşmesi", 28 Ocak 1981 tarihinde imzaya açılmış ve ülkemiz tarafından da imzalanmıştır. Avrupa Konseyi ayrıca, kişisel verilerin korunmasına yönelik, tıbbi veri bankaları, bilimsel araştırma ve istatistik, doğrudan pazarlama, sosyal güvenlik, sigorta, polis kayıtları, istihdam, elektronik ödeme, telekomünikasyon ve internet gibi çeşitli sektörlerde uygulanacak ilkeleri belirleyen tavsiye kararları da kabul etmiştir. Tasarının hazırlanması sırasında, söz konusu tavsiye kararları göz önüne alınmakla beraber, tasarının "çerçeve tasarı" niteliği korunmuştur. Tüm sektörlerle ilgili düzenlemelere yer verilmesi halinde, Tasarının hacminin çok genişleyeceği düşünülerek, söz konusu tavsiye kararları Tasarıya alınmamıştır. Bu tavsiye kararlarında yer alan ilkelere, ilerleyen süreçte, değişik sektörlerle ilgili yapılacak düzenlemelerde yer verilebileceği değerlendirilmiştir. Öte yandan, Avrupa Birliği, üye ülkelerin kişisel verilerin korunmasına ilişkin mevzuatı arasında uyum sağlamak üzere, 24 Ekim 1995 tarihinde "Kişisel Verilerin İşlenmesi Sırasında Gerçek Kişilerin Korunması ve Serbest Veri Trafiği Direktifi"ni (95/46/EC) yürürlüğe koymuştur. Bu Direktifle, üye ülkelerdeki bireylerin kişisel verilerinin üst düzeyde korunması ve kişisel verilerin Avrupa Birliği içerisinde özgür dolaşımını sağlayacak açık ve kalıcı bir düzenleme yapılması amaçlanmıştır. Kişisel verilerin korunmasına yönelik uluslararası belgeler göz önüne alındığında; bu konuya ilişkin hazırlanacak kanunda, kişisel verilerin işlenme şartlarının, bireylerin aydınlatılmasının, bu alanı denetleyecek ve düzenleyecek bir otoritenin oluşturulmasının, veri güvenliğine ilişkin gerekli tedbirlerin alınmasının temel ilkeler olarak kabul edildiği görülmektedir. Uluslararası belgeler, mukayeseli hukuk uygulamaları ve ülkemiz ihtiyaçları göz önüne alınmak suretiyle hazırlanan Tasarıyla, kişisel verilerin çağdaş standartlarda işlenmesi ve koruma altına alınması amaçlanmaktadır"* denilmektedir [40]. Belirtilen metinde ülkemizde kişisel verilerin

düzenlendiği bir yasaya olan ihtiyacı belirtmektedir. Kişisel verilerin korunmasındaki asıl amaç verinin sahibini korumaktır.

2.5.4. Kişisel veri koruma teknikleri

Gizliliğin korunması GDPR'ın tanımına göre tanımlanmış veya tanımlanabilir gerçek bir kişiye ait herhangi bir bilgi olarak tanımladığı kişisel verilerle doğrudan ilgilidir. Kişisel verilerin korunmasında anonimleştirme ve takma adlandırma(pseudonymization) GDPR tarafından önerilen iki tekniktir. Çünkü bu iki teknik risk düzeyini düşürür ve kurumların veri koruma yükümlülüklerini yerine getirmesinde yardımcı olurlar. Bu iki tekniğin temel özelliği birey ile veriler arasındaki bağlantıyı azaltmalarıdır [41].

Anonimleştirme

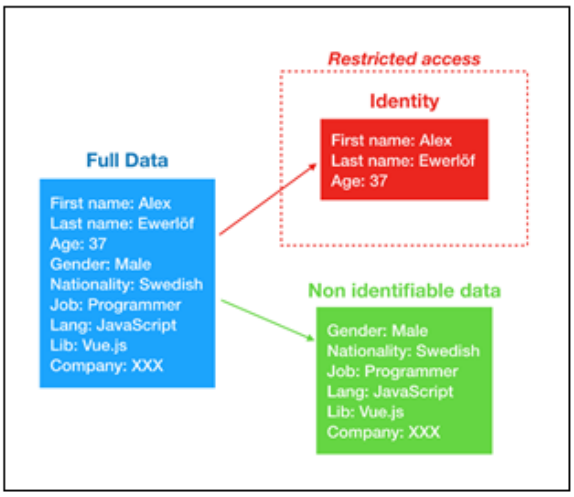
Tanımlayıcı görevi görebilecek herhangi bir bilginin kalıcı olarak kaldırılmasıdır. Bir veri seti anonim hale getirildikten sonra bireylere o veri setinden tanımlanması imkânsız olması gerekir [41]. Anonim verilerin paylaşılmasında yasal bir sakınca bulunmamaktadır. Anonim veriler genellikle istatistiksel çalışmalarda kullanılmaktadır.

Takma adlandırma (Pseudonymization)

Takma adlandırma, bir veri kaydındaki kişisel olarak tanımlanabilir bilgi alanlarının takma adla değiştirildiği bir veri yönetimi ve kimlik gizleme prosedürüdür. Değiştirilen her alan için tek bir takma ad, veri kaydını daha az tanımlanabilir hale getirirken, veri analizi ve veri işleme için uygun kalmasını sağlar. Pseudonymization, kişisel verilerin gizliliğinin korumak için güvenlik uzmanlarını tarafından kullanılan ayrıca GDPR tarafından önerilen bir tekniktir [42].

Pseudonymization tekniği uygulanmasında ilk aşamada gizlenmesi istenen veri seti incelenir. Veri setinde yer alan kişisel veriler belirlenir. Şekil 2.8 de görüldüğü üzere veri sahibine ulaşılabilecek verilerin tespiti gerçekleştirilir. Sonrasında bu kişisel veriler maskelenebilir, rastgele değer atanabilir veya orijinal veriye tekrar ulaşılmak isteniyorsa şifreleme teknikleri kullanılarak kişiye ait bilgiler değiştirilir. Böylece verileri veri

sahipleri ile ilişkilendiremeyecek böylece veriler kötü niyetli kişilerin eline geçmesi durumda kişisel verilerin güvenliği sağlanmış olacaktır.



Şekil 2.8: Pseudonymization Tekniği

K - Anonimlik (K – Anonymity)

K-anonimlik, bir veri kümesindeki bazı alanlarla, birden fazla kişinin tanımlanmasıyla, kombinasyonlarda tekil özellikler gösteren kişilere ait bilgilerin belirlenmesini engellemek için geliştirilmiştir bir yöntemdir. Bir veri kümesindeki değişkenlerden bazılarının bir araya getirilerek oluşturulan kombinasyonlara ait birden fazla kayıt bulunması halinde, bu kombinasyona denk gelen kişilerin kimliklerinin saptanabilmesi olasılığı azaltılmaktadır. Örneğin; Çizelge 2.3 de kişilere ait bilgiler ve sahip oldukları hastalık bilgisi yer almaktadır. Kişilerin isim ve soy isimleri maskelenerek anonimlik sağlanmaya çalışılmıştır. Ancak bu şekilde yapılan anonimleştirmede tabloda aynı değerleri içeren sadece bir kayıt olması durumunda bu tekil kayıtları gerçek kişiler ile ilişkilendirmek mümkün olacaktır. Tabi ki bu kayıtların gerçek kişiler ile ilişkilendirilebilmesi için kişilere ait bilgilerin olduğu veri setine bir şekilde erişim sağlanması durumunda mümkün olacaktır. Örneğin; Çizelge 2.3 de 1982 yılında doğmuş cinsiyeti kadın ve posta kodu 3440 ile başlayan tek bir kayıt olduğu için bir şekilde kişi bilgileri elde edilmiş belgeye bakılarak o kişinin kim olduğu ve hastalığının ne olduğu tespit edilebilecek ve kişisel veriler korunamamış olacaktır. K-Anonimlik bu durumun önüne geçilebilmesi için kullanılan bir tekniktir. K-Anonimliğin nasıl uygulanacağını göstermek adına Çizelge 2.3 de 1983 yılında doğan cinsiyeti erkek olan posta kodu 3440 ile başlayan 5 farklı kayıt

bulunmaktadır. Aynı özelliklere sahip 5 farklı kayıt olduğu için o kişilere ait kimin hangi hastalığa sahip olduğunun tespit edilmesi mümkün olmayacaktır [43].

Çizelge 2.3: K-Anonimlik Örneği

Ad Soyad	Doğum Tarihi	Cinsiyet	Posta Kodu	Hastalık Adı
*	1983	E	3440*	Soğuk Algınlığı
*	1982	E	3440*	Hepatit-B
*	1983	E	3440*	Astım
*	1980	E	3440*	Baş Ağrısı
*	1982	K	3440*	Beyin Tümörü
*	1983	E	3440*	Yüksek Tansiyon
*	1983	E	3440*	Baş Ağrısı
*	1980	K	3440*	Grip
*	1983	E	3440*	Akciğer Kanseri

L – Çeşitlilik (L-Diversity)

K-Anonimlik modelinin bir uzantısıdır. Bir veri temsilinin ayrıntı düzeyini azaltarak veri kümelerindeki gizliliği korumak için kullanılan grup tabanlı bir anonimleştirme biçimidir. Bu azalma, bir miktar gizlilik kazanmak için veri yönetimi veya madencilik algoritmalarında bir miktar etkinlik kaybına neden olabilmektedir [44]. L-Çeşitlilik yöntemi K-anonimlik uygulandıktan sonra veri ihlallerinin oluşabilme olasılığından ötürü ortaya konulmuş bir yöntemdir. Örneğin; Çizelge 2.4 de kişilere ait hastalık bilgilerinin olduğu veriler K=4 anonimleştirilmesi uygulanmıştır. Yani her kayıttan 4 adet olacak şekilde anonimleştirme yapılmıştır. Ancak son gruba bakıldığında posta kodu 130 ile başlayan 30 lu yaşlarında olan kişiler uyruğundan bağımsız olacak şekilde Kanser hastaları olarak gruplanmıştır. Bu iki bilgiye sahip olan bir kullanıcı tanıdığı bir kişinin kesin olarak kanser hastası olduğu bilecek böylece o kişiye ait kişisel veri ihlal edilmiş olacaktır. Bu durumun önüne geçilebilmesi için K-anonimlik uygulanan her grubun içerisinde belirli bir çeşitliliğin sağlanmasını savunan tekniğe L-çeşitlilik olarak adlandırılır. Örneğin; Çizelge 2.5 de her grupta en az 3 farklı L=3 çeşit hastalığın bulunması sağlanmıştır. L-çeşitlilik sayesinde anonimleştirme işlemi güçlendirilmiştir [43].

Çizelge 2.4: K=4 Anonimleştirme Uygulanan Veri Kümesi

Posta Kodu	Yaş	Uyruk	Hastalık
130**	< 30	*	Kalp
130**	< 30	*	Kalp
130**	< 30	*	Viral Enfeksiyon
130**	< 30	*	Viral Enfeksiyon
1485*	≥ 40	*	Kanser
1485*	≥ 40	*	Kalp
1485*	≥ 40	*	Viral Enfeksiyon
1485*	≥ 40	*	Viral Enfeksiyon
130**	3*	*	Kanser
130**	3*	*	Kanser
130**	3*	*	Kanser
130**	3*	*	Kanser

Çizelge 2.5: K=4 Anonimlik ve L=3 Çeşitlilik Uygulanan Veri Kümesi

Posta Kodu	Yaş	Uyruk	Hastalık
1305*	≤ 40	*	Kapl
1305*	≤ 40	*	Viral Enfeksiyon
1305*	≤ 40	*	Kanser
1305*	≤ 40	*	Kanser
1485*	> 40	*	Kanser
1485*	> 40	*	Kalp
1485*	> 40	*	Viral Enfeksiyon
1485*	> 40	*	Viral Enfeksiyon
1306*	≤ 40	*	Kalp
1306*	≤ 40	*	Viral Enfeksiyon
1306*	≤ 40	*	Kanser
1306*	≤ 40	*	Kanser

Maskeleme (masking)

Kişisel verilerin belirli alanlarının silinmesi veya maskelenmesi kişisel bilginin belirsiz hale getirilmesi işlemidir. Kurumların kişisel verilere erişim ihtiyacı olabilmektedir ancak kişisel verilere erişim istismarlara neden olmaması gerekmektedir. Ayrıca, birçok kuruluş uygulama geliştiricilerin kapsamlı testler gerçekleştirebilmeleri için üretim veya canlı verilere erişim sağlaması gerekebilmektedir. Bu gibi ihtiyaç olduğu durumlarda verilerin paylaşılmasında maskeleme gerekebilir. Veri tabanı yönetim sistemlerinin veri maskeleme yetenekleri bulunmaktadır [45]. Örnek olarak Çizelge 2.5 de bulunan “Uyruk” alanına maskeleme işlemi uygulanmıştır.

Karıştırma (Fixing)

Veri seti içerisindeki değerlerin karıştırılarak toplam faydaya zarar verilmeden kişilerin tespit edilebilirliğinin ortadan kaldırılması anlamına gelir. Örneğin yaş ortalamaları üzerine yapılacak bir çalışmada veri setindeki kişilerin yaşları bir birbirleriyle değiştirilerek verilerin karıştırılması gibi [45].

Yaklaşıklık (approximation)

Belirli kişisel verileri daha genel değerlerle saklanması yöntemidir. Örneğin kişilerin doğum tarihlerini 01.09.1990 yerine 09.1990 veya 1990 şeklinde saklanması gibi [45].

Şifreleme (encryption)

Kişisel verilerin gizli anahtar ile şifrelenmesi işlemidir. Şifrelenmiş veriler sadece uygun anahtar ile okunabilir hale getirilebilir. Kuruluşlar itibarlarını korumak için giderek daha fazla kullandıkları bir yöntemdir. GDPR'da zorunlu bir teknik olmamasına rağmen önerilen bir tekniktir [45]. Veri gönderilmesi gereken kurum bir açık anahtar oluşturur ve bunu yayınlar kuruma ait gizli anahtar sadece kendisinde bulunur. Gönderilmek istenen veri kuruma ait açık anahtar ile şifrelenir ve karşı kuruma gönderilir. Şifrelenmiş veri sadece gönderilmek istenen kurumun gizli anahtarı ile açılabileceği için gönderilmek istenen verinin güvenliği sağlanmış olacaktır.

Gürültü ekleme (adding noise)

Gürültü ekleme kişisel verileri içeren veri kümesinde rastgele önemsiz değişiklikler yapma sürecidir. Gürültünün eklenmesi verileri büyük ölçüde değiştirebildiği ve verilerin güvenilirliğini azaltabileceği için çok sık kullanılan bir teknik değildir [45].

Tokenization

Anlamlı verilerin anlamlı olmayan token adı verilen rastgele oluşturulmuş karakter dizisine dönüştürülme işlemidir. Tokenlar, gerçek verilere erişimde kullanılmazlar sadece gerçek verilere erişimde referans görevi görürler. Hassas veriler ile tokenların ilişkileri bir veri

tabanında tutulmaktadır. Veri tabanındaki veriler genellikle şifreleme yoluyla ayrıca güvence altına alınmaktadır [46].

Tekniklerin karşılaştırılması

Veri işlemede gizliliği sağlayabilmek için kullanılan teknikler kuruluşların sahip oldukları verilerin yapısı, boyutu ve amacına göre teknikler seçilmelidir. Bir sorunun çözümünde tüm tekniklerin kullanılması mümkün değildir. Teknikler veri güvenliği, değiştirilen veri miktarı ve sağlanan anonimlik seviyesi olmak üzere üç faktöre göre karşılaştırılabilir [45].

Çizelge 2.6. Kişisel veri koruma teknikleri karşılaştırması

Teknik	Veri Güvenilirliği	Veri Değişikliği	Veri Anonimliği
Anonymization	***	**	***
k-Anonymity	***	-	**
l-Diversity	***	-	***
Masking	***	-	**
Derivation	**	**	**
Mixing	**	***	**
Approximation	**	**	**
Encryption	***	-	***
Tokenization	***	-	***
Adding Noise	*	***	*
(-: Yok, *: Düşük, **: Orta, ***: Yüksek)			

2.6. Kişisel Verilerin Korunmasında Uyulması Gereken İlkeler

Kişisel verilerin korunması konusunda referans alınması gereken ilk düzenleme KVKK olmalıdır. KVKK'nın 4. Maddesinde kişisel verilerin işlenmesinde uyulması gereken ilkeler açıkça belirtilmiştir. Bu ilkeler: Hukuka ve dürüstlük kurallarına uygun olma; doğru ve gerektiğinde güncel olma; belirli, açık ve meşru amaçlar için işlenme; işlendikleri amaçla bağlantılı, sınırlı ve ölçülü olma ve ilgili mevzuatta öngörülen veya işlendikleri

amaç için gerekli olan süre kadar muhafaza edilme şeklinde düzenlenmiştir [39]. Kişisel verilerin korunması konusundaki ilkeler hem KVKK'dan hem de uluslar arası düzenlemelerden faydalanılarak incelenecektir.

2.6.1. Hukuka ve dürüstlük kurallarına uygun olma ilkesi

Kişisel verilerin işlenmesinde kanun ve hukuki düzenlemelere uygun işlenmesi gerekliliği hukuka uygun olmaz ilkesinin gereğidir. KVKK'nın 4. Maddesi 2-a bendinde "Hukuka ve dürüstlük kurallarına uygun olma" şeklinde düzenlenmiştir. Avrupa Konseyi Szöleşmesinin 5. Maddesinde kişisel verileri yasal olarak elde edilmesi ve işlenebilmesi konusunda düzenlemeler yapılmıştır.[47] Medeni kanunun 2.maddesinde düzenlenmiş olan dürüstlük kuralı kişisel verilerin işlenmesi sırasında da ihlal edilmemesi gerekmektedir [48].

Bu ilke doğrultusunda Anasaya'nın 20. Maddesinde kişisel verilerin kanunda belirtilen durumlar haricinde veya kişinin izni alınmadan kullanılamayacağı belirtilmektedir. Kanuni bir dayanak olmadan kişisel verilerin kullanılması, işlenmesi veya depolanması bu ilkeye aykırıdır. Ayrıca kişisel verilerin işlenmesi veya depolanması özel hayata müdahale niteliği taşımasından dolayı Anayasadaki temel hak ve hürriyetlerin sınırlandırılmasına ilişkin düzenlemelere de ayrıca uygun olması gerekmektedir [49].

2.6.2. Amaca bağlı olma

KVKK'nın 4. Madde 2-c bendinde "*belirli, açık ve meşru amaçlar için işlenme",* 2-ç bendinde *"işlendikleri amaçla bağlantılı olma"* şeklinde düzenlenmiştir. 2016 /679 sayılı Genel Veri Koruma Tüzüğü'nün b bendinde, kişisel verilerin *"belirlenmiş, açık ve kanuni bir amaca dayalı"* şeklinde düzenlenmiştir [56]. KVKK'nın 12/4. Maddesinde kişisel verileri işleyenlerin sahip oldukları kişisel verileri Kanunlara aykırı şekilde başkaları ile paylaşmayacakları ve işlenme amacı dışında kullanamayacakları düzenlenmiştir. Avrupa Konseyi Sözleşmesinin 5/b maddesinde kişisel verilerin belirlenmiş kanuni işlemler için kaydedileceği ve bu amaçlarına aykırı kullanılamayacağı düzenlenmiştir [30].

Kişisel veriler işlenirken; kanunlarda verilen yetkileri en az şekilde kullanılması ve yapılan işlemlerin amacına ulaşılmasında asgari ölçüde kişisel verilerin kullanılması amaca bağlı

olma ilkesine uygun olacak davranış biçimi olacaktır. Kişisel verilerin ilk kez işlenmesi sırasında kişilerden alınan açık rızanın sonradan işlenme amacının değişmesi durumda kişilerden yeniden açık rıza alınması gerekmektedir.

2.6.3. Kişisel verilerin işlenmesinin gerekli olması

KVKK'nın 4. Maddesi 2-d bendinde "*ilgili mevzuatta öngörülen veya işlendikleri amaç için gerekli olan süre kadar muhafazaedilme"* şeklinde düzenlenmiştir. 108 sayılı Avrupa Konseyi Sözleşmesinin 5/e maddesinde [30] kişisel verilerin işlenmesindeki amaca göre gerekli süreyi aşmamak kaydı ve kişilerin kimliklerinin tespiti sağlanabilecek şekilde saklanabileceği belirtilmiştir [50].

Gereklilik ilkesi doğrultusunda kişisel verilere ihtiyaç duyulduğunda toplanması anlamına gelmektedir. İşlenen kişisel veri ile işlenme amacı arasında mantıklı bir bağ olması gerekmektedir [51]. İşlenecek kişisel verinin amacı doğrultusunda yeterli miktarda olmalı ve amacına ulaşılınca kişisel veriler yok edilmelidir.

2.6.4. Kişisel verilerin doğru ve güncel olması (unutulma hakkı)

KVKK'nın 4. Maddesinin 2. Fıkrasının "b" bendinde; *"doğru ve gerektiğinde güncel olma"* şeklinde düzenlenmiştir. Veri işleme sürecinin son aşamasına kadar devam etmesi gereken bir ilkedir. Veri sahibi kişisel verilerinin güncel tutulmasını talep etme ve kendine ait verilerin güncelliğini periyodik olarak kontol etme hakkı vardır [50].

Kişisel verilerin doğru ve güncel olması kapsamında kişilerin unutulma hakları da vardır. Bu hak 016/679 sayılı Genel Veri Koruma Tüzüğünün 17. Maddesinde [52] ve Anayasa'nın 20. Maddesinin ikinci fıkrasında kişilerin kendilerine ait verilerin silinmesini talep etme haklarının olduğu belirtilmiştir. Unutulma hakkı güvence altına alınmıştır [50].

2.6.5. Kişisel verilerin işlenmesinde ölçülü olma

KVKK'nın 4. Maddesinin 2-ç bendinde *"işlendikleri amaçla bağlantılı, sınırlı ve ölçülü olma"* şeklinde düzenlenmiştir. 108 sayılı Avrupa Konseyi Sözleşmesi 5/c maddesinde kişisel verilerin kaydedilme amacına uygun olması gerektiği ve aşırı olamayacağı

düzenlenmiştir [47]. 2016/679 sayılı Genel Veri Koruma Tüzüğü'nde "veri minimizasyonu" şeklinde ifade edilmiştir [52]. Kişilere ait verileri hedeflenen amaçlar doğrultusunda en az sayıda veri toplanması, veri ekonomisi olarak isimlendirilen bu ilke gereğince fazla kişisel veri işlenmemeli, verisi işlenen kişilerin verileriyle ilgili karşılaşılabileceği risklerin en aza indirilmesi amaçlanmalıdır.

TCK'nın 138. Maddesinde, kişisel verilerin elde edilmesinden belirli bir süre geçtikten sonra sistemden silinmemesi suç olarak düzenlemiştir, kişisel verilerin gereklilikleri ortadan kalktığı zaman silinmesi veya anonim hale getirilmesi ölçülülük ilkesi ile korunmuş olacaktır [53].

2.6.6. İşlemenin aleni olması

Kişisel verilerin işlenip işlenmediğinin kamuya açıklanmasına işlemenin aleni olması denir. Aleniyet ilkesi sayesinde kişiler kendilerine ait verilerin işlenip işlenmediğini öğrenebilmesi verilerini işleyenin kim olduğunu bilmesi kendine ait verileri düzelttirme, sildirme ve itiraz etme haklarını kullanabilir [54].

2.6.7. İlgili kişilerin bilgilendirilmesi

KVKK'nın 10. Maddesinde *"Aydınlatma Yükümlülüğü"* başlığında veri sorumlularının bilgilendirme yükümlülüğü düzenlenmiştir. Kişisel verilerin toplanmasında veri sorumlusu veya yetkilendiren kişiler veri sorumlusunun ve varsa kanunu temsilcisinin kimliği, kişisel verinin işlenme amacını, işlenecek olan kişisel verilerin kimlerle neden paylaşılabileceği, kişisel verinin elde edilme yöntemi konularında bilgilendirmekle yükümlülerdir [51].

2.6.8. İlgili kişilerin bilgi edinmesi

KVKK'nın 11. Maddesinde *"İlgili kişinin hakları"* maddesinde herkes veri sorumlularına başvuruda bulunduğunda aşağıdaki haklara sahip olduğu belirtilmiştir [50].

- Kişisel verilerinin işlenip işlenmediğini öğrenme
- Verileri işlenmişse bununla ilgili bilgi talep etme
- Yurt içinde ya da yurt dışında kişisel verilerin aktarıldığı üçüncü kişileri öğrenme

- Kişisel verilerin eksik ya da yanlış işlenmiş olması durumunda bunların düzeltilmesini talep etme
- 7. maddede düzenlenen şartlar çerçevesinde verilerin silinmesini ya da yok edilmesini isteme
- Kişisel verilerin eksik ya da yanlış işlenmesi nedeniyle bunların düzeltildiğinin, kişisel verilerin aktarıldığı üçüncü kişilere bildirilmesini isteme
- İşlenen kişisel verilerin münhasıran otomatik sistemler aracılığıyla analiz edilerek kişinin kendisi aleyhine bir sonucun ortaya çıkması sonucuna itiraz etme
- Kişisel verilerin kanuna aykırı olarak işlenmesi nedeniyle zarar görmesi hâlinde zararın giderilmesini talep etme

2.7. Türk Hukukunda Kişisel Verilerin Korunması

Kişisel veriler Türk hukukunda çeşitli mevzuatlarla korunmaktadır. Başta anayasa olmak üzere 6698 sayılı Kişisel verilerin korunması kanunu, Türk Medeni Kanunu, Türk Borçlar kanunu, Türk Ceza kanunu ile korunmaktadır.

2.7.1. Kişisel verilerin korunması kanunu

Kişisel verilerin işlenmesi, kişilerin temel hak ve hürriyetlerini korumak amacıyla 23 Mart 2016 tarihinde 6698 sayılı Kişisel Verilerin Korunması Kanunu kabul edilmiştir.

Kanunun kapsam başlıklı 2'nci maddesine göre, *"kişisel verileri işlenen gerçek kişiler ile bu verileri tamamen veya kısmen otomatik olan ya da herhangi bir veri kayıt sisteminin parçası olmak kaydıyla otomatik olmayan yollarla işleyen gerçek veya tüzel kişiler"* hakkında bu Kanun hükümleri uygulanır. Bu ifadeden yola çıkılarak kişisel verileri gerçek kişiler bu kanun kapsamda değerlendirilir. Hem kamu hem de özel sektörde veri işleyen gerçek veya tüzel kişiler hakkında bu kanun hükümleri uygulanır [47].

KVKK'nın 28. Maddesinin birinci fıkrasında belirlenen diğer bir durum, *"milli savunmayı, millî güvenliği, kamu güvenliğini, kamu düzenini, ekonomik güvenliği, özel hayatın gizliliğini veya kişilik haklarını ihlâl etmemek ya da suç teşkil etmemek kaydıyla, sanat tarih, edebiyat veya bilimsel amaçlarla ya da ifade özgürlüğü kapsamında işlenmesi*dir."

Verinin işlenmesinin kanun kapsamında ihlal olup olmadığına hâkim tarafından karar verilmelidir.

Kişisel verilerin hukuka aykırı olarak işlenmesi sebebiyle ortaya çıkan zararın giderilmesi talep edilebilecektir. *"Kişisel veri işlemenin suç işlenmesinin önlenmesi veya suç soruşturması için gerekli olması, ilgili kişinin kendisi tarafından alenileştirilmiş kişisel verilerin işlenmesi, kişisel veri işlemenin kanunun verdiği yetkiye dayanılarak görevli ve yetkili kamu kurum ve kuruluşları ile kamu kurumu niteliğindeki meslek kuruluşlarınca, denetleme veya düzenleme görevlerinin yürütülmesi ile disiplin soruşturma veya kovuşturması için gerekli olması ve kişisel veri işlemenin bütçe, vergi ve mali konulara ilişkin olarak devletin ekonomik ve mali çıkarlarının korunması için gerekli olması"*ndan ibarettir [47].

2.7.2. Anayasa

Anayasamızda yer alan hak ve hürriyetler doğrudan veya dolaylı olarak kişisel verilerle ilişkilidir. Anayasamızda kişisel verilerin korunması ile ilgili düzenlemelere Hukuk Devleti İlkesi (m.2), Bireyin Maddi ve Manevi Varlığını Serbestçe Geliştirme Hakkı (m.17), Özel Hayatın Gizliliği Hakkı (m.20), Konut Dokunulmazlığı (m.21), Haberleşmenin Gizliliği (m.22), Dini Ve Vicdani Kanaatleri Açıklamaya Zorlanamama (m.24), Düşünce ve Kanaatleri Açıklamaya Zorlanamama (m.25) hakkı örnek olarak verilebilir. Anayasamızın 20. Maddesi 3. Fıkrasında kişisel verilerin korunmasına ilişkin düzenlemeler yer almaktadır. Bu düzenleme ile kişisel verilerin korunması anayasal bir temele kavuşmuştur [36].

2.7.3. İş kanunu

İş kanununda işçilerin kişisel verilerinin korunması ile ilgili hükümler yer almaktadır. 4857 sayılı iş kanunu'nun 75. Maddesi gereği işçinin özlük dosyasında yer alan bilgilerin gizli tutulması gerekliliği düzenlenmiştir. İkinci fıkrada "*gizli kalmasında işçinin haklı çıkarı bulunan bilgileri açıklamamakla yükümlüdür*" şeklindeki açıklama ise işverene sır saklama yükümlülüğü getirilmiştir [55].

2.7.4. Türk borçlar kanunu (TBK.)

Türk Borçlar Kanunu‟nun 419. Maddesinde özel kanun hükümleri saklı tutulmak şartıyla; *“ İşveren, işçiye ait kişisel verileri, ancak işçinin işe yatkınlığıyla ilgili veya hizmet sözleşmesinin ifası için zorunlu olduğu ölçüde kullanabilir. Özel kanun hükümleri saklıdır.”* şeklindeki düzenleme ile işçiye ait kişisel veriler sadece işçinin işe yatkınlığı ve işi ifa etmeye yönelik konuları ile sınırlandırılmıştır başka amaçlar ile işçinin kişisel verilerinin kullanılması kısıtlanmıştır [27].

2.7.5. Türk medeni kanunu (TMK.)

Kişilik haklarının korunması amacıyla kişisel verilerin korunması gerekmektedir. Kişesel verilerin korunmasında yaşanacak ihlaller kişilik haklarının ihlali anlamı taşıyacaktır. Kişinin “kişilerin vücut bütünlüğüne, onuruna(şerefine), sırlarına, manevi dünyasına ilişkin haklarına kişilik hakları denir [56].

Medeni kanunun 24. Maddesine göre “Kişilik hakkı zedelenen kimsenin rızası, daha üstün nitelikte özel veya kamusal yarar ya da kanunun verdiği yetkinin kullanılması sebeplerinden biriyle haklı kılınmadıkça, kişilik haklarına yapılan her saldırı hukuka aykırıdır.” Bu maddeye göre kişinin kişilik haklarının ihlali kişinin rızasına bağlanmıştır. Rızası haricindeki işlemlerde kanunun 25. Maddesine göre dava açma hakkına sahiptir.

2.7.6. Türk ceza kanunu (TCK.)

Türk ceza kanunda kişisel verilerin ihlal edilmesi durumunda şikayete bağlı bir suç olup cezai yaptırımı bulunmaktadır. TCK’nin 134. Maddesine göre “*kişilerin özel hayatının gizliliğini ihlal eden kişi, hapis veya adli para cezası ile cezalandırılır*” maddesi gereğince cezalandırılır. Eğer bu ihlal görüntü veya seslerin kayıt altına alınması ile yapılmışsa ceza artırılır. TCK’nin 136. Maddesi “*kişisel verileri hukuka aykırı olarak bir başkasına veren, yayan veya ele geçiren kişinin eyleminin hapis ile cezalandırılacağı*” göre kişisel verileri ihlaline sebep olan kişi kamu görevlisi olması durumda cezanın ağırlaştırıcı sebep olarak kabul edilir ve görevi kötüye kullanma suçu olarak kabul edilir [57].

3. KVKSIS UYGULAMASINDA KULLANILAN TEKNOLOJİLER

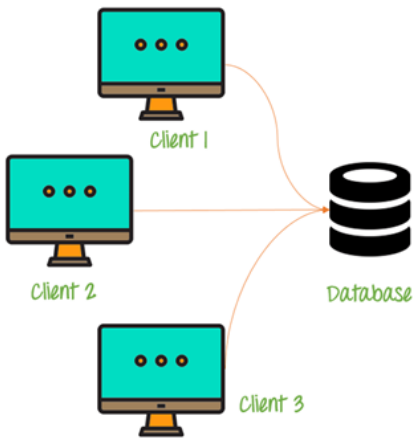
3.1. Yazılım

Farklı ve çeşitli görevlerin yapılabilmesi için tasarlanmış elektronik aygıtların gerekirse birbirleriyle iletişim kurarak kendilerine verilen görevleri yerine getirmesini sağlayan makine komutlarıdır. Yazılımlar var olan bir sorunu çözmek amacıyla bilgisayar dili kullanılarak oluşturulan anlamlı bilgisayar komutları bütünüdür. Pascal, .NET ve Java bu dillere örnek verilebilir [58].

3.1.1. Yazılım mimarileri

İstemci – sunucu mimarisi

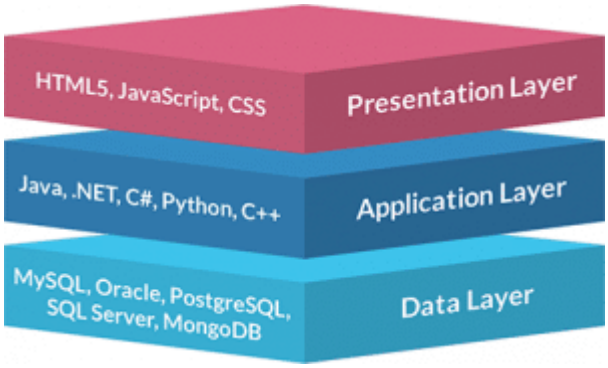
İstemci- sunucu mantığı ile çalışan sistemler iki katlı mimari üzerine kurulmuşlardır. İstemci ve sunucu bölümlerinden oluşmaktadır. Uygulamanın yapması gereken işlerin büyük çoğunluğu istemci tarafından gerçekleştirilirken sunucu bölümü çoğu zaman sadece veri tabanı işlemlerini gerçekleştirmektedir. Uygulamanın yapması gereken işlerin çoğu istemci tarafından gerçekleştirildiği için uygulamada yapılması gereken küçük bir değişiklik sonucu son kullanıcıların uygulamayı güncellemesi gerekmektedir. Uygulamanın yoğun kullanımı halinde sistemin ölçeklenebilirliği arttırılamadığı için donanımı artırmak tek çözüm yoludur. Donanım artırmak maliyetli bir işlemdir [59]. Şekil 3.1'de 2 katmanlı mimari yapısı gösterilmektedir [60].



Şekil 3.1. İki katmanlı mimari

Üç katmanlı mimari

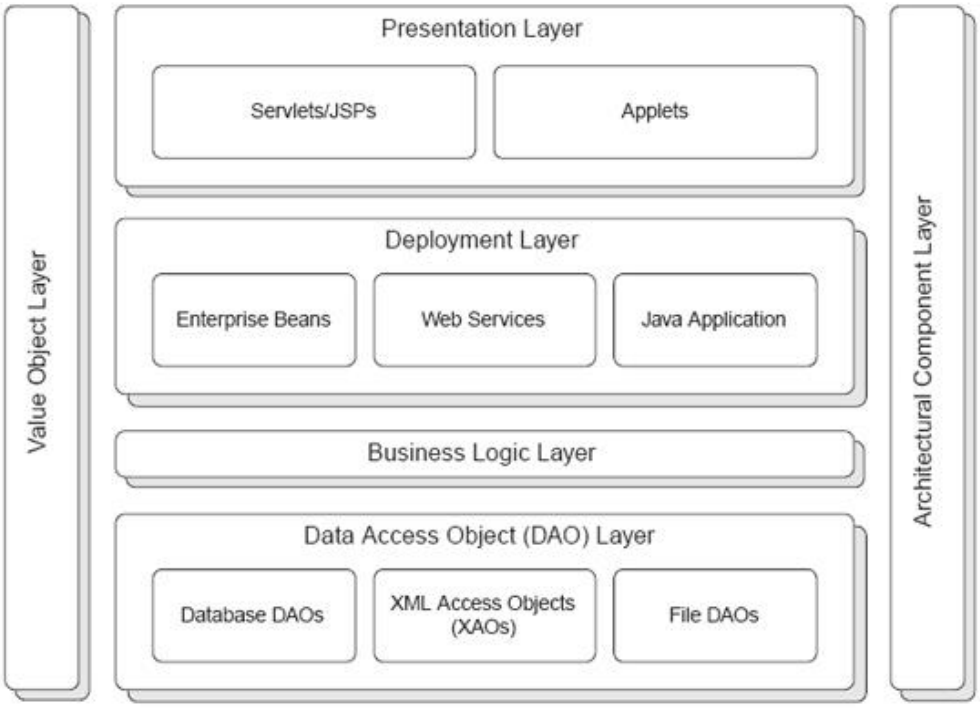
Üç katmanlı mimariler sunum, uygulama ve veri tabanı katmanlarından oluşmaktadır. Sunum Katmanı, bir uygulamanın en üst katmanıdır. Sunum katmanı yazılımı (Arayüz, web sayfaları) kullanırken görülen katmandır ve ana işlevi uygulama katmanıyla iletişim kurmaktır. Bu katman, kullanıcı tarafından klavye hareketleri ve fare tıklamaları açısından verilen bilgileri uygulama katmanına aktarır [61]. Bu katman genellikle HTML5, JavaScript, CSS gibi web teknolojileri veya diğer popüler web geliştirme çerçeveleri aracılığıyla oluşturulur ve API çağrıları aracılığıyla diğer katmanlarla iletişim kurar [62]. İş katmanı olarak da adlandırılan uygulama katmanı sunum ve veri tabanı katmanlarından alınan verileri işleyen katmandır. Sunum ve veri tabanı katmanı arasında bir aracı görevi görür. Veri tabanı katmanı veri tabanına bağlanan ve gerekli eylemleri gerçekleştiren katmandır. Bu eylemler ekle, sil ve güncelleme örnek olarak verilebilir [61]. Şekil 3.2'de 3 katmanlı mimari yapısı gösterilmektedir [62].



Şekil 3.2. Üç katmanlı mimari

Çok katmanlı mimari

Genellikle n katmanlı mimari olarak adlandırılan çok katmanlı mimari mimari, sunum, uygulama işleme ve veri yönetimi işlevlerinin fiziksel olarak ayrıldığı bir istemci-sunucu mimarisidir. Çok katmanlı mimarinin en yaygın kullanımı üç katmanlı mimaridir [63]. Şekil 3.3'de J2EE'nin çok katmanlı mimari yapısı gösterilmektedir [64].



Şekil 3.3. J2EE çok katmanlı mimari yapısı

3.1.2. Kullanılan yazılım dilleri

Java programlama dili

Java dili Sun Microsystem’s tarafından 1995 yılında geliştirilmiştir. Java C ve C++ dillerinin yazım kurallarının türevlerini kullanmakta ancak bu dillerle karşılaştırıldığında daha basit bir nesne modeline ve daha üstün yeteneklere sahiptir.

Java, olabildiğince az uygulama bağımlılığına sahip olacak şekilde tasarlanmış, sınıf tabanlı, nesne yönelimli bir programlama dilidir. Uygulama geliştiricilerin bir kez yazmasına, her yerde çalışacak şekilde tasarlanmış bir programlama dilidir. Bu özelliği ile derlenmiş Java kodunun Java'yı destekleyen tüm platformlarda yeniden derlemeye gerek olmadan çalışabilmektedir. Java uygulamaları genellikle temel bilgisayar mimarisinden bağımsız olarak herhangi bir Java sanal makinesinde (JVM) çalışabilecek şekilde bayt koduna göre derlenir [65]. 2020 itibarıyla Java, GitHub'a göre özellikle istemci-sunucu web uygulamaları için kullanılan en popüler programlama dillerinden biridir [66].
Java’nın kurucusulan Sun Microsystems tarafından açıklanan java’nın en önemli 3 özelliği;

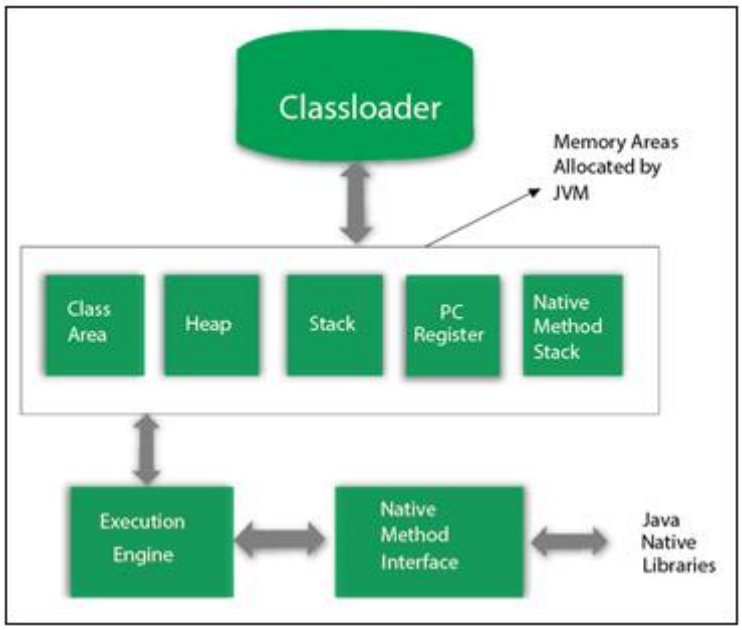
- Daha zengin kullanıcı deneyimi
- İş birleştirme özelliğidir
- Web servisleri için daha uygun uygulama ortamı'dır [67].

Language	2020 Ranking	2019 Ranking	2018 Ranking
JavaScript	1	1	1
Python	2	2	3
Java	3	3	2
TypeScript	4	7	4
C#	5	5	6
PhP	6	4	4
C++	7	6	5
C	8	9	8
Shell	9	8	9
Ruby	10	10	10

Şekil 3.4. GitHub 2018-2020 yılları arası programlama dili sıralaması [68]

Java sanal makinası (JVM)

JVM (Java Virtual Machine), sunucuların Java programını çalıştırmasını sağlayan soyut bir makinedir. Java programı çalıştırıldığında, Java derleyicisi önce Java kodunuzu bayt koduna derler. Ardından, JVM, bayt kodunu yerel makine koduna çevrilir ve çalıştırılır [69].



Şekil 3.5. JVM çalışma şekli

JVM'in yazılan kodları yorumlamanın yanına da bazı görevleri bulunmaktadır;

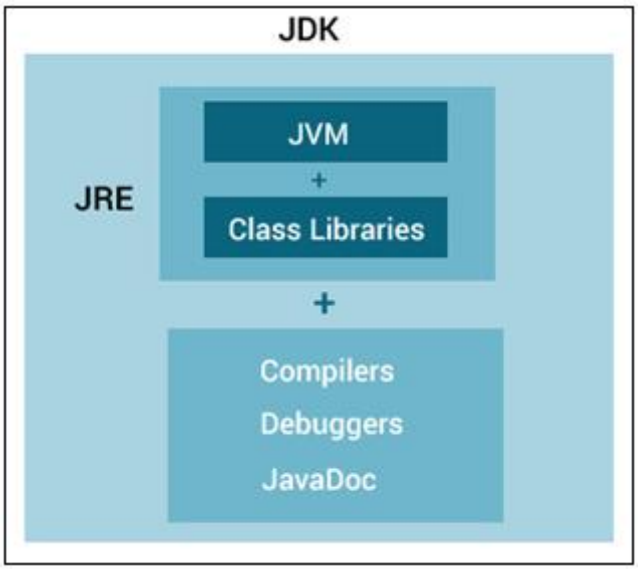
- Kodları belleğe yüklerken güvenlik denetimi yapması.
- .class uzantılı dosyaların çalışması sırasında bellekte oluşabilecek gereksiz dosyaları (çöp) toplamasıdır [68].

Java geliştirme seti (JDK)

Java dilinde program geliştirebilmek için gerekli olan JVM, JRE, Java Interpreter, debugger, Java api içeren bir pakettir. Java dilinde yazılan, hataları tespit eden, derleme işlemini gerçekleştiren ve çalıştıran bütün paketleri içerir [68].

Java çalışma zamanı ortamı (JRE)

Java Runtime Environment veya JRE, bir bilgisayarın işletim sistemi yazılımının üzerinde çalışan ve belirli bir Java programının çalıştırması gereken sınıf kitaplıklarını ve diğer kaynakları sağlayan bir yazılım katmanıdır [70].



Şekil 3.6. JVM, JDK ve JRE arasındaki ilişki

JPA (Java Persistence API)

Java nesnelerini veri tabanı tablolarıyla eşleme ve bunun tersi Nesne ilişkisel eşleme (ORM) olarak adlandırılır. Java Persistence API (JPA), ORM'ye yönelik olası bir yaklaşımdır. JPA aracılığıyla geliştirici, ilişkisel veri tabanlarından Java nesnelerine ve tersi yönde verileri eşleyebilir, depolayabilir, güncelleyebilir ve alabilir. JPA, Java-EE ve Java-SE uygulamalarında kullanılabilir. JPA, geliştiricinin SQL ifadeleri yerine doğrudan nesnelerle çalışmasına izin verir. Çoğu JPA kalıcılık sağlayıcısı, meta verilere göre veri tabanı şemasını otomatik olarak oluşturma seçeneği sunar [71].

```
@Entity
@Table(name = "METOT")
public class Metot implements Serializable{

    //private static final long serialVersionUID = -3009157732242241606L;

    @Id
    @Column(name = "METOT_ID")
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long metotId;

    @Column(name = "SERVIS_ADI")
    private String servisAdi;

    @Column(name = "METOT_ADI")
    private String metotAdi;

    @Column(name = "EKLENME_TARIHI")
    private Date eklenmeTarihi;

    @OneToMany(mappedBy="metot", fetch = FetchType.EAGER)
    private List<ParametersPath> ParametersPathList;
```

Şekil 3.7. JPA Metadata Açıklayıcı Notları

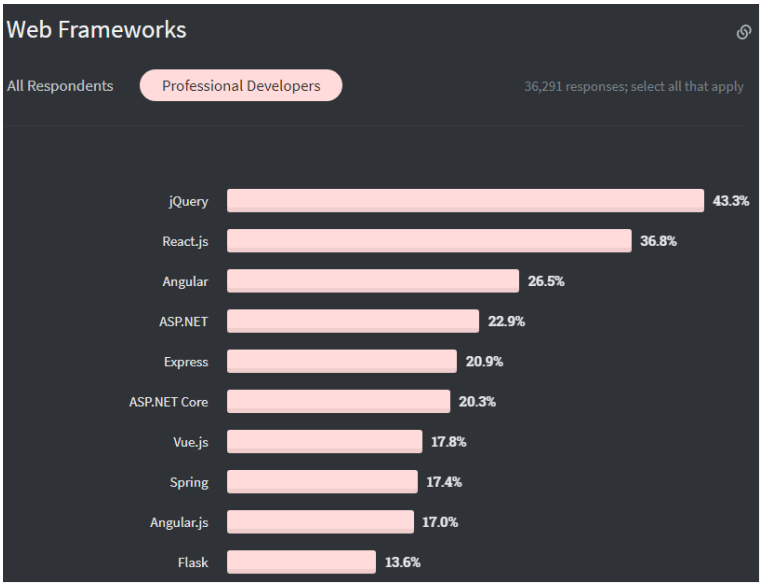
Kullanılan açıklayıcı notların bazıları Çizelge 3.1'de gösterilmektedir.

Çizelge 3.1. Açıklayıcı not örnekleri ve açıklamaları [67]

@Entity	Bir sınıfın kalıcı olduğunu belirtir.
@Inheritance	İlgili sınıfı kalıcı yapıcak kalıtım modelini belirtir.
@IdClass	İlgili sınıfiçin belirleyici (Id) özelliği gösterir.
@NamedQuery	İlgili kalıcı birim üzerinde belirlenen bir JPQL'i tanımlamak için kullanılır.
@Table	İlgili sınıfın eşleştitileceği veritabanı tablosunu tanımlar.
@Transient	Kalıcı olmayacak alanları tanımlama için kullanılır.
@OneToOne	Başka bir kalıcı alanla 1-1 ilişkisi olacak alanı tanımlar.
@OneToMany	Başka bir kalıcı alanla 1-N ilişkisi olacak alanı tanımlar.
@ManyToMany	Başka bir kalıcı alanla M-M ilişkisi olacak alanı tanımlar.
@ManyToOne	Başka bir kalıcı alanla N-1 ilişkisi olacak alanı tanımlar.
@GeneratedValue	İlgili alanın bir yaratıcı kullanarak tanımlanacağını belirtir.
@OrderBy	İlgili alanın belirtilen özelliğe göre sıralanması gerektiğini belirtir.

ReactJS

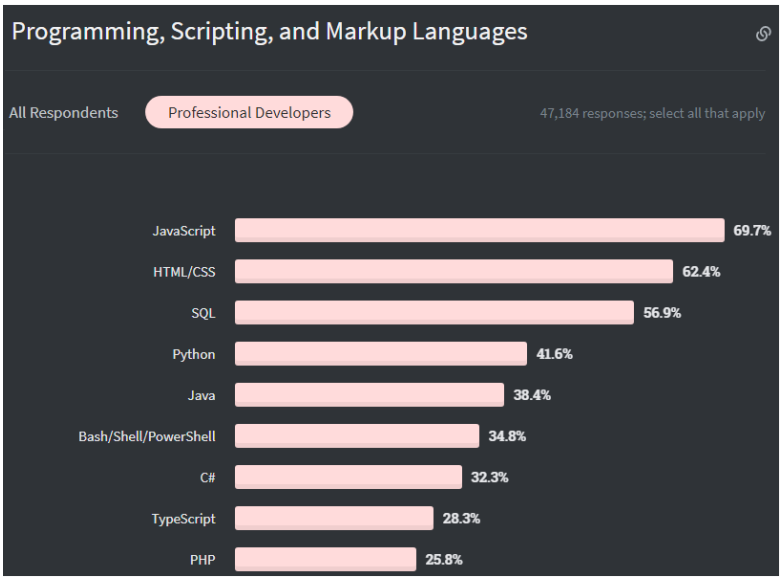
ReactJS, yeniden kullanılabilir kullanıcı arabirimi (UI) bileşenleri geliştirmek için oluşturulan JavaScript kütüphanesidir. React, temel olarak, sonraki sayfa yenilemeleri olmadan verilerini değiştirebilen büyük ve karmaşık web tabanlı uygulamaların geliştirilmesini sağlar. Model-Görünüm-Kontrolör yapısında (MVC) Görünüm (V) olarak kullanılır. React çoğunlukla NodeJS kullanılarak sunucu tarafında oluşturulur ve yerel mobil uygulamalar için destek React Native kullanılır [72].



Şekil 3.8. Stackoverflow'un 2020 yılında "Web Frameworks" kategorisinde yaptığı araştırma sonucu [73]

Javascript

Javascript kapsamlı kullanıcı ara yüzleri içeren web uygulamaları ve çeşitli sistemler oluşturulmasında kullanılan modern web uygulamalarının temel parçasıdır [74]. Javascript, web, gömülü sistemler ve sunucu sistemleri dahil olmak üzere çeşitli uygulamaların geliştirilmesinde kullanılabilen programlama dilidir [75].



Şekil 3.9. Stackoverflow'un 2020 yılında "Programlama, Script Yazma ve Biçimlendirme Dilleri" kategorisinde yaptığı araştırma sonucu [73]

3.2. Web Servis

İnternet yaygınlaştıkça uygulamalar arası veri alışverişine talep artmıştır. Mevcut protokollerin uygulamalar arası iletişim (application-to-application, A2A) için kullanılan http protokolü uygulamalar arası veri transferlerinde ortaya çıkacak karmaşık işlemler için uygun yapıda değildi. Http protokolü sunucular ile son kullanıcılar arasında verilerin nasıl aktarılacağına dair kural ve yöntemleri düzenlemekteydi.

Ortaya çıkan ihtiyaçlar sebebiyle 1999 yılında, Microsoft tarafından uygulamalar arası veri alışverişinde kullanılabilen XML tabanlı SOAP adında bir protokol yayınlanmıştır. SOAP kullanılarak servislerin tanımlamalarının daha net yapılabilmesi ve bulunabilmesi için IBM, Microsoft ve Ariba'nın beraber geliştirdiği web servis tanımlama dili (WSDL) geliştirmişlerdir. Sistemler arasında veri transferlerinde ortaya çıkacak sorunları çözmenin ideal yollarından birisi belirli bir standardın oluşturulmasıdır. Web servisler farklı sistemler arasındaki iletişimi sağlamanın uygun bir yöntemidir. Web servisler arasındaki iletişim XML ile sağlanır. Bütün iletişim XML üzerinden kurulduğu için web servisler donanım, işletim sistemi ve programlama dillerinden bağımsızdırlar [76].

3.2.1. Soap (simple object access protocol)

SOAP (Basit Nesne Erişim Protokolü) uygulamalarda ve web servislerinin haberleşmesinde kullanılmak üzere tasarlanan, RPC (Remote Procedure Call) modelini kullanan, istemci/sunucu mantığına dayalı bir protokoldür. Daha genel olarak SOAP, web üzerinden fonksiyonları kullanmak için geliştirilmiş bir sistemin XML tabanlı kurallar topluluğudur. SOAP ile ilgili bütün mesajlar XML formatında iletilir. Soap mesajı envelope, header ve body bölümünden oluşmaktadır [77].

```
<SOAP-ENV:Envelope xmlns:SOAPENV="
    http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
    ....
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
    ....
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Şekil 3.10. Soap mesajı yapısı

3.2.2. Restful

Roy Fielding tarafında 2000 yılında yayınlanan doktora tezi ile dağıtık sistemler tasarlanmasında kullanılabilecek mimari bir tarz olan REST kavramını ortaya koymuştur [78]. REST, http protokolü üzerinde çalışan bir mimaridir. İstemci-sunucu arasında XML veya JSON verilerin taşınarak uygulamanın haberleşmesini sağlanır. RESTful servisler REST mimarisi kullanılarak oluşturulur [79].

RESTful web servislerin http metotlarının görevi;

- GET metodu: Veri listeleme ve görüntülemek.
- POST metodu: Yeni bir kaynak oluşturmak.
- PUT metodu: Bir kaynağın veya verinin güncellenmesi
- DELETE metodu: Bir kaynağın veya verinin silinmesi.

3.2.3. Restful ve soap karşılaştırması

Restful ve Soap web servis teknolojilerinden hangisinin seçilmesi gerektiğine karar verme sürecinde bir birlerine olan üstünlükleri göz önüne alınarak seçim yapılmalıdır. Bu kapsamda iki teknolojinin bir birlerine olan avantajları şunlardır;

Rest servisin soap servise göre avantajları

- RESTful servislerde önbelleğe alma özelliği sayesinde cevap verme süresi kısalmaktadır. RESTful servisler de SOAP servislerinde bulunan zarf yapısı olmadığı için veri transferinde gönderilen ve alınan verilerin boyutları daha azdır.
- RESTful servisler daha küçük mesaj formatları kullandığı için daha verimlidir.
- Kapsamlı işlem gerektirmediği için hızlıdır [80].

Soap servisin rest servise göre avantajları

- SOAP servisler bütünleşme odaklı olması dolayısı ile sistemler arasındaki uyumluluk sağlandığı için uygulamalar açısından avantajlıdır.

- Güvenlik, kayıt işlemi ve bazı iletişim modelleri (senkron, asenkron, istek/geri çağırma, uyarı vb.) sunduğundan dolayı geliştirilmesi ve tasarımı REST servise göre daha karmaşıktır.
- Dil, platform ve iletişimde bağımsızlık sağlar. Rest servisler http kullanımı gerektirir.
- Dağıtılmış kurumsal ortamlarda iyi çalışır. Rest doğrudan noktadan noktaya iletişim kurar.
- Dahili hata yönetimine sahiptir [80].

Çizelge 3.2. RESTful ve SOAP web servislerinin karşılaştırılması [79, 81]

SOAP	REST
SOAP bir protokolüdür. Web servisin bulunduğu yere ek olarak web servisin ne yaptığı hakkında gerekli bilgilerin bulunduğu bir WSDL dosyası içerir.	REST bir mimari tarzdır. İçeriisnde herhangi bir bilgi içeren dosya bulunmamaktadır.
REST mimari bir tarz olduğu için SOAP Rest'i kullanamaz.	REST, SOAP'ı web servisleri için temel protokol olarak kullanabilir.
SOAP tabanlı yaklaşımlar kendi içerisinde bir dizi standartlar içerir.	REST güncel internet standartlarını kullanacağını vaat etmektedir.
SOAP güvenlik konusunda hala geliştirilmekte olan bir servistir.	REST mimarisinde var olan güvenlik altyapsını kullanıldığı için RESTful sistemlerin daha güvenli olduğu ileri sürülmektedir.

3.3. Veri Tabanı

Kullanım amaçları doğrultusunda belirli formatlara uygun olarak hazırlanmış veriler topluluğuna veri tabanı denir. Veri tabanlarında kapsamı belirli ve bu doğrultuda birbiriyle ilişkili kayıtların bir arada tutulduğu ve digital ortamda muhafaza edildiği ortam olarak kabul edilir. Veri tabanları kullanım alanlar; mühendislik, hukuk, eğitim, genetik bilimi, sosyal medya gibi bir çok alanda kullanılmaktadır. Veri tabanları çeşitli ihtiyaçlar doğrultusunda muhafaza ettiği veriler üzerinde okuma, yazma, güncelleme ve silme işlemlerinin yapılmasına olanak sağlar.

3.3.1. İlişkisel veri tabanı

1970 yılında Edgar Frank Codd tarafından tasarlanan ilişkisel veri tabanı, birbirleriyle ilişkili verilerin belirli kritirlere göre tablolarda saklanması ve tablolar arasında ilişki kurularak veri bütünlüğünün sağlanmasını hedefleyen tasarım biçimine denir. Codd tarafından veri tabanlarında tutulan veriler üzerinde işlemlerin gerçekleştirilebilmesi için (ekleme, silme, güncelleme gibi) SQL (Structural Query Language) dili geliştirilmiştir. SQL 1983 yılında veri tabanlarında kullanılmak üzere standart olmuştur. ISO ve ANSI tarafından 1987 yılında ilişkisel veri tabanlarında kullanılacak standart dil olarak kabul edilmiştir [82].

İlişkisel veri tabanlarının avantajları

- Veri tekrarını azaltır.
- Veri tabanı hakimiyetini artırır.
- Veri güvenliği sağlanır.
- Veri bağımsızlığı sağlanır.

İlişkisel veri tabanlarında veri bütünlüğünün sağlanabilmesi için veri tutarlılığının sağlanması gerekmektedir. Veri tutarlığının sağlanabilmesi için ACID yani Bölünmezlik (Atomicity), Tutarlılık (Consistenct), İzolasyon (Isolation), Dayanıklılık(Durability) ilkelerine bağlı kalınmalıdır.

Bölünmezlik: Bölünmezlik ilkesi veri tabanlarında birbirini izleyen ve etkileyen işlemler bir bütün olarak görülmekte işlemlerin tamamı ya başarı ile tamamlanacak aksi takdirde işlemlerin tamamının başarısız olması durumudur.

Tutarlılık: Veri tabanında yapılan her işlemden sonra alınan girdi, çıktı ve yapılan işlemlerin tanımlanan kurallara uygun olması durumudur.

İzolasyon: Veri tabanlarında aynı anda aynı veri üzerinde işlem yapılması gereksinimi söz konusu olduğunda işlemlerin bir birini etkilememesi için işlemler seri olarak yapılır.

Yapılması hedeflenen işlemlerde sorun olmaması amacıyla veri setleri üzerinde gerçekleşen işlemler başarılı veya başarısız sonuç dönene kadar kitlenmesine izolasyon denir.

Dayanıklılık: Veri tabanında yapılacak işlemler esnasında fiziksel veya işlemsel bir hata olması halinde sistemin kendisini bir önceki geçerli duruma döndürebilme yeteneğine denir [83].

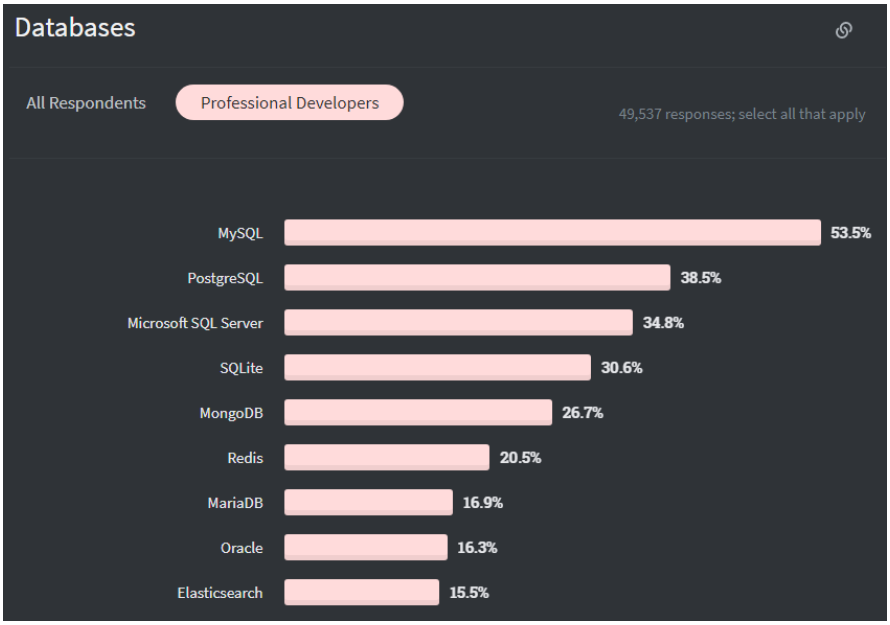
PostgreSQL

PostgreSQL karmaşık veri iş yüklerini saklayan, ölçekleyen açık kaynaklı bir ilişkisel veri tabanı sistemidir. PostgreSQL Kaliforniya üniversitesinde bulunan POSTGRES projesinin bir parçası olarak 1986 yılında SQL dilini kullanacak şekilde oluşturulmuştur. PostgreSQL 30 yıldan fazla bir süredir geliştirilmeye devam etmektedir. Kanıtlanmış mimarisi güvenilirliği, veri bütünlüğü, genişletilebilirliği ve açık kaynak topluluğu performans ve yenilikçi çözümler üretmesi postgresql'in güçlü taraflarıdır. Tüm yaygın işletim sistemleri ile çalışabilen postgresql, 2001 yılından beri ACID ile uyumlu çalışabilmekte ve bu özelliği ile PostGIS gibi jeo uzamsal veri tabanları gibi güçlü eklentilere sahiptir [84].

PostgreSQL tercih edilme sebepleri;

- Açık kaynak kodlu
- Ücretsiz
- Platform bağımsız
- Dünyanın birçok yerinde geliştiricisinin bulunması
- ANSI SQL uyumlu

Stackoverflow'un 2020 yılında yaptığı araştırmaya göre veri tabanı kategorisinde en popüler olan teknolojiler arasında %38,5 ile PostgreSQL 2. Sırada yer almaktadır. Ayrıca projemizde kullandığımız Elasticseach ise %15. ile 9. Sırada yer almaktadır [73].



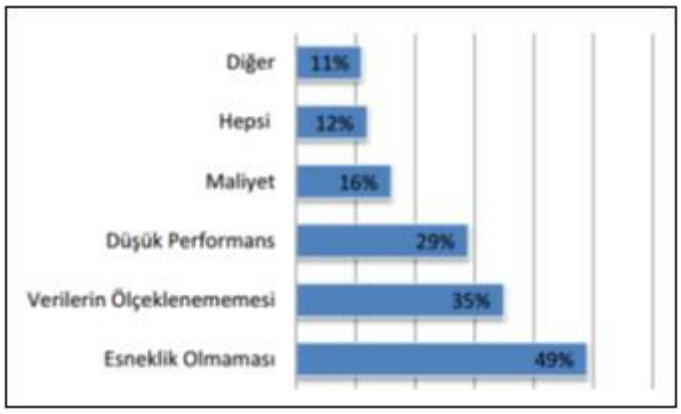
Şekil 3.11. Stackoverflow 2020 yılı araştırma sonucu

3.3.2. İlişkisel olmayan veri tabanı (NoSQL)

NoSQL ilişkisel veri tabanı sistemlerine alternatif olarak 1998 yılında Carlı Strozzi tarafından ortaya konulan bir kavramdır. NoSQL yatay olarak ölçeklendirebilen veri depolama sistemleridir [85]. NoSQL sistemlerinin yatay ölçeklendirilebilme özelliği ile bir çok sunucuya dağıtılabilmektedir bu sayede veriler birden fazla sunucuda yedeklenebilmektedir. NoSQL sistemler arayüzleri ve protokoller ile kolayca entegre olabilmekte ve dağıtık indeksleme özelliği ile veri depolama da daha verimli ram kullanımı sağlamaktadır [86].

NoSQL sistemlerin kullanılma sebepleri;

- Kaydedilmek istenen verilerin değişebilen veri yapısı olması durumda
- Denormalizayson yapılması gerekliliği durumlarında
- Sistem gereksinimleri değişkenlik göstermesi durumlarında tercih edilmelidir [85].
- Ayrıca Şekil 3.12'de gösterilen sebepler ile NoSQL kullanılmaktadır.

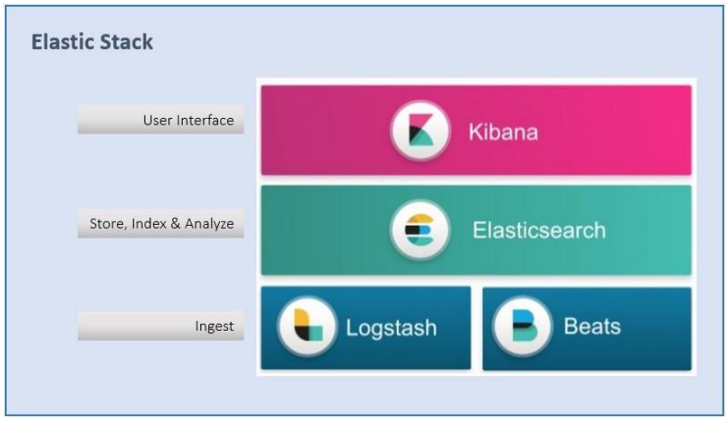


Şekil 3.12. NoSQL tercih edilme sebepleri [85]

Elastic stack

ELK yığını üç araçtan oluşur bunlar; Elasticsearch, Logstash ve Kibana. Her ne kadar Logstash ve Elasticsearch ayrı olarak çalışabilsede, üç ürün şu anda Elastik Yığın olarak adlandırılan entegre bir çözüm olarak kullanılmak üzere tasarlanmıştır [87]. Elasticsearch bir arama ve analitik motorudur. Logstash, aynı anda birden fazla kaynaktan veri alan, dönüştüren ve ardından Elasticsearch gibi bir yığına gönderen bir sunucu tarafı veri işleme hattıdır. Kibana ise logstash'den dönüştürülen verilerin görselleştirildiği bölümdür [88]. ELK yığınına Beats isimli ürün sunuculardan Logstash'a veri gönderen hafif bir nakliyeci olarak yığına dördüncü ürün olarak eklendi [87].

ELK yığını şu anda Elastik adlı şirket tarafından açık kaynak lisansı altında tutulmakta ve aktif olarak desteklenmektedir. Ayrıca gönüllüler tarafından geliştirilen bir dizi topluluk eklentisi de bulunmaktadır [87].



Şekil 3.13. Elastic stack projeleri

Elasticsearch

2010 yılında Shay Banon Java ile geliştirilmiş olan lucene tabanlı dağıtık tam-cümle arama motoru olan elasticsearch'ü yayınlamıştır. Elasticsearch verileri JSON formatında tutmaktadır [89].

Elasticsearch içerisine dokümanlar kaydedilirken, dokuman içerisindeki alanlar "Apache Lucene" altyapısı kullanılarak indexlenmektedir. Böylece bir kelimenin hangi dokumanda geçtiği bilgisi tutulur. Daha sonra bir kelime Elasticsearch içerisinde arandığında büyük veri kümesinde aranmaz direk oluşturulan index listesi içerisinde aranır böylece arama sonucu çok hızlı bir şekilde sonuçlandırılır.

Bir örnek ile açıklamak gerekirse aşağıdaki gibi bir index içerisinde 4 farklı dokuman bulunmaktadır. Her dokuman içerisindeki veriler ayrı ayrı indexlenmektedir ve index tablosuda yan tarafta gösterilmiştir. Örneğin "nal" kelimesi 1 ve 2. Dokümanlarda bulunmaktadır. Nal kelimesi elasticsearchde arandığında aranan kelime büyük veri kümesi içerisinde değil index tablosunda hızlıca aranır ve sonuçlar getirilir.

Doküman

1	Bir mıh bir bir nal kurtarır
2	Bir nal bir at kurtarır
3	Bir at bir yiğidi kurtarır
4	Bir yiğit bir memleketi kurtarır

İndex Listesi

Terim	Doküman
bir	<1>,<2>,<3>,<4>
mıh	<1>
nal	<1>,<2>
kurtarır	<1>,<2>,<3>,<4>
at	<2>,<3>
yiğidi	<3>
yiğit	<4>
memleketi	<4>

Şekil 3.14. Elasticsearch indexleme yapısı

Elasticsearch Avantajları;

- Elasticsearch, Java'da geliştirildiği için her platformda çalışacak şekilde uyumludur.
- Bu gerçek zamanlı bir arama motorudur.
- Büyük organizasyonlarda ölçeklendirmeyi kolaylaştıran yapısı sayesinde geliştiriciler sistemi ölçeklendirerek herhangi bir büyük kuruluşa kolayca entegre edebilir.
- Çoklu dil desteği bulunmaktadır.

- Elasticsearch açık kaynaklıdır. Bu nedenle, indirmek için herhangi bir lisans ücreti ödemeye gerek yoktur.
- Yalnızca metin belgelerini değil tüm belge türlerini destekler [90].

Logstash

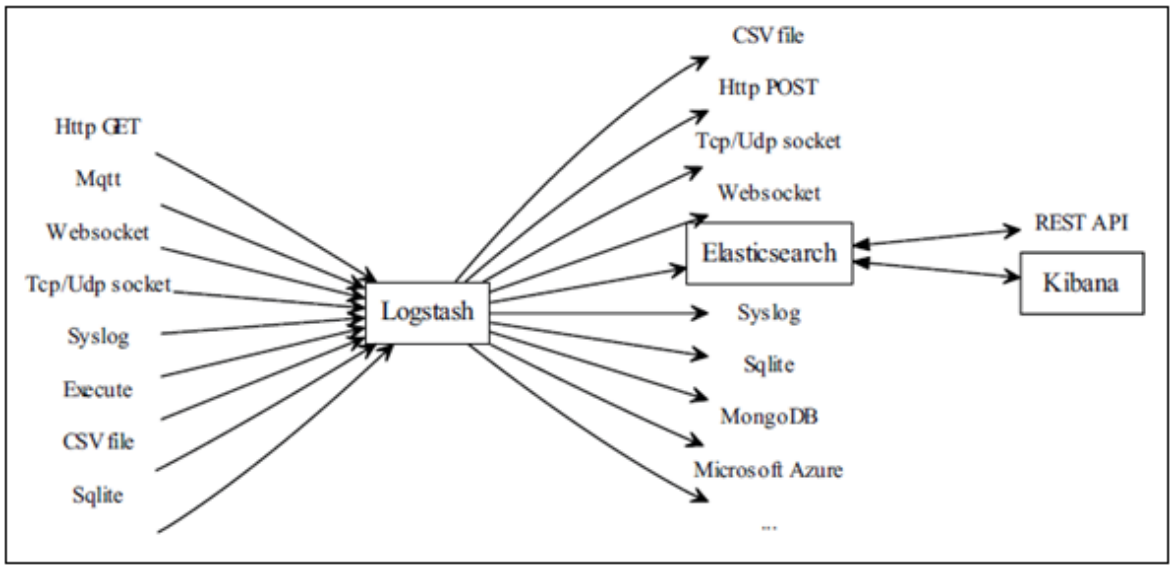
Logstash, biçimi ve karmaşıklığına bakmaksızın verileri dinamik olarak alır, dönüştürür ve gönderir. Logstah birden fazla kaynaktan gelen verileri eş zamanlı işleyebilir. Logstash işlemler giriş, filitreleme ve çıkış olarak üç aşamalı gerçekleşir. Elasticsearch kaynak olarak çeşitli türleri destekler bunlar .txt, .csv gibi, hedef olarak ise Elasticsearch seçilebilmektedir [91].

Kibana

Kibana, Elasticsearch için bir görselleştirme platformu olarak tasarlanmıştır. Kibana ile Elasticsearch içerisine kaydedilen verileri görüntüleyebilir, çeşitli dashboardlar oluşturulabilmektedir ayrıca elasticearch'e kaydedilen indexler yönetilebilmektedir.
Kibana'nın Avantajları;

- Verileri kolayca görselleştirme imkanı sunar.
- Elasticsearch ile entegre olarak çalışır.
- Gerçek zamanlı analiz, grafik oluşturma, özetleme ve hata ayıklama yetenekleri sunar.
- Kullanıcı dostu bir ara yüze sahiptir.
- Aranan logların anlık görüntülerinin paylaşılmasını sağlar.

Şekil 3.15'deki gibi logstash çeşitli dosya türlerinden verileri alır tanımlanması halinde çeşitli dönüştürmeler yaparak Elasticsearch dahil olmak üzere çeşitli dosya türlerine verileri gönderebilmektedir. Logstash tarafından elasticsearch'e kaydedilen veriler kibana aracılığı ile görüntülenebilmektedir.



Şekil 3.15. ELK Stack genel görünümü

3.4. Versiyon Kontrol Sistemleri

Versiyon kontrol sistemleri belgeler üzerinde yapılan tüm değişiklikleri izlemek, saklamak ve yönetmek için kullanılmaktadır. Versiyon kontrol sistemlerinin destekledikleri işlemler.

Yaratma (create): Dosyaları saklamak için yeni bir boş repository oluşturma işlemidir.
Teslim alma (checkout): Üzerinde çalışma yapmak için repository'den herhangi bir sürümü alma işlemine denir.
İşlemek (commit): Repository'den alınan çalışmanın üzerinde yapılan değişiklikleri repository'e kaydetme işlemidir.
Fark (dif): Üzerinde işlem yapılan bir çalışmanın iki sürümü arasındaki farkların gösterilmesi işlemidir.
Dal (branch): Ana sürümden farklı ve bağımsız bir sürüm oluşturma işlemidir.
Birleştirme: Oluşturulan iki dal yapılan değişiklikleri yeni bir sürüm elde etmek için yapılan birleştirme işlemine denir [92].

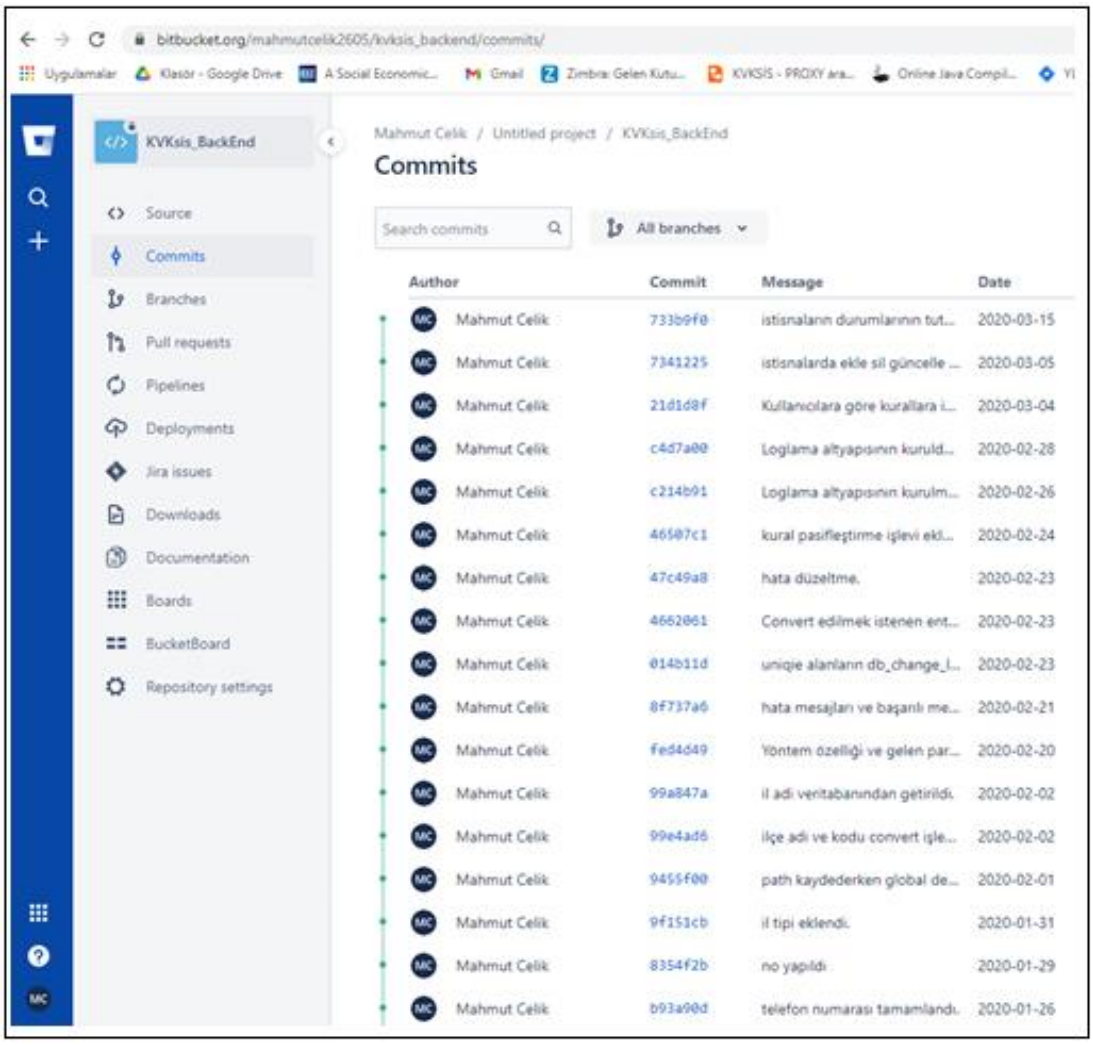
Versiyon kontrol sistemlerinde verileri saklamak için iki temel teknik kullanılmaktadır. Birincisi her yeni revizyonun tam bir kopyası saklanır ikinci teknik sadece revizyonlar arasındaki farklar tutulur. Yalnızca aradaki farkları depolayan sürüm kontrol sistemlerine örnekler Visual SourceSafe ve Kaynak Kod Kontrol Sistemi (SCCS) iken Git gibi modern bir VCS (Version Control System) revizyonların anlık görüntülerini saklar.

3.4.1. Bitbucket

Bitbucket, profesyonel ekipler için tasarlanan Git veri deposu yönetim çözümüdür. Git depolarını yönetmek, kaynak kodları üzerinde birlikte yazılım geliştirme boyunca rehberlik etmek için merkezi bir çözüm sunar. Bitbucket sunduğu özellikler;

- Kaynak kodlara erişimi kısıtlamak için erişim kontrolü.
- Bir proje veya ekip iş akışına uyması için iş akışı kontrolü.
- Jira entegrasyonu
- Bitbucket'da bulunmayan ihtiyaçlarımızı karşılayabilmek için REST API özelliği bulunmaktadır [93].

Bitbucket'ı projemizde hem versiyon kontrol sistemi olarak hem de talep yönetim sistemi özelliğini kullanılmıştır. Projenin Bitbucket'taki gelen görünümü Şekil 3.16'daki gibidir.



Şekil 3.16. Bitbucket

3.5. Geliştirme Ortamları

Tasarlanan uygulamanın geliştirme aşamasında yazılım geliştirme araçları kullanılmıştır. Bu araçlar bir nevi yazılım geliştirme yazılımlarıdır. Bu çalışma kapsamında geliştirilen uygulamanın geliştirilme aşamasında kullanılan yazılımlar anlatılmaktadır.

3.5.1. Eclipse

IBM tarafından 2001 yılında başlatılan ve Borland, Oracle, SAP, OMG gibi 40'ı aşkın yazılım geliştirme araçları üreten firmaların üye olduğu Eclipse Birliği (Eclipse consortium) tarafından yürütülen açık kaynak kodlu bir projedir. Eclipse projesi, Eclipse platformu java programlama dli ile entegre geliştirme ortamı SDK (Software Development

Kit) üreterek Eclipse tabanlı araçların üretimi için bir geliştirme ortamı sunmaktadır. Bu özelliği ile eclipse platformu genişletilebilir bir platformdur [94].

3.5.2. Visual studio code

Visual Studio Code, masaüstünde çalışan ve Windows, MacOS ve Linux için kullanılabilen güçlü bir kaynak kodu editörüdür. JavaScript, TypeScript ve NodeJS dil desteği sunmaktadır [95]. Visual Studio Code'u özellikleri;

- Git gibi versiyon kontrol sistemleri ile entegre olabilmesi
- Platform bağımsız kod geliştirilebilir olması
- Code debugger özelliği
- Visual Studio Code Live Share özelliği geliştirilen kodların paylaşılabilir olması
- Kod tanıma özelliği (code intellisense) [96].

3.5.3. Postman

Postman Hindistanlı Abhinav Asthana tarafından web servis testlerinin gerçekleştirilebilmesi için oluşturulmuştur. Postman google chrome tarayıcı içerisinde eklenti olarak hem de masaüstü uygulaması ile kullanılabilmektedir. Tasarlanan sistemin backend'inde hazırlanan web servislerin test edilmesinde kullanılmıştır. Postman'in özellikleri;

- Kolay kullanılabilir bir ara yüze sahiptir.
- Geçmiş istekleri kaydetme ve otomatik tamamlama önerileri sunmaktadır.
- Postman sayesinde herhangi bir ek kurulum gerektirmeden web servisler testleri gerçekleştirilebilir.
- Postman'in kimlik doğrulama desteği bulunmaktadır.
- API çağırılarını dokümante edebilme ve paylaşma özelliği bulunmaktadır [97].

3.5.4. PgAdmin

PgAdmin PostgreSQL veritabanı için geliştirilen en popüler açık kaynak yönetim ve geliştirme platformudur [98]. PgAdmin sunucu tarafında pyhton, uygulama framework'ü olarak Flask, micro-framework, istemci tarafında Javascript/JQuery/Backbon, HTML düzenlerinde Bootstrap teknolojileri kullanılarak bir çok şirketin ve bireysel katılımcıların oluşturduğu PostgreSQL küresel gelişim grubu tarafından geliştirilmiş ve geliştirilmeye devam etmektedir [99]. PgAdmin tasarlanan sistemin PostgreSQL veri tabanı tasarım ve yönetim ara yüzü olarak kullanıldı. Veri tabanında tablo oluşturma, veri ekleme, silme ve sorgulama gibi işlemlerinin gerçekleştirilmesine olanak sağlamıştır.

4. KVKSIS UYGULAMASININ GENEL TANIMI

4.1. Uygulamanın Yapısı

Bu çalışma kapsamında kişisel verilerin yazılımcılara (insan faktörüne) karşı korunması amacıyla bir uygulama tasarlanmıştır. Tasarlanan bu uygulamaya KVKSIS (Kişisel Verileri Koruma Sistemi) adı verilmiştir. Tezin ilerleyen bölümlerinde bu şekilde isimlendirilecektir. KVKSIS uygulamasında web servis sorgulaması yazılım geliştirici tarafından yapılıyorsa web servisten gelen veriler içerisindeki kişisel veriler belirli bir düzende değiştirilip yazılım geliştiriciye anonim hale getirilmiş veriler sunulacaktır. Ancak web servis sorgulaması uygulama sunucusu üzerinden yapılmışsa web servisten gelen verilerde herhangi bir değişiklik yapılmadan sunulacaktır. Böylece kişisel verilerin kötüye kullanılmasının önüne geçmek amacıyla alınan önlemlere ek bir önlem alınmış olacaktır.

KVKSIS 3 bölümden oluşmaktadır. Birinci bölüm, web servis cevap nesnesine ait bütün parametrelerin tespit edilip daha sonra kullanılmak üzere veri tabanına kayıt işleminin yapıldığı bir rest servistir.

İkinci bölüm, KVKSIS arayüzü kullanılarak birinci bölümde veri tabanına kaydedilen web servis cevap nesnesi içerisinde yer alan parametrelere kural tanımlandığı bölümdür. Tanımlanan bu kurallar KVKSIS uygulamasının web servis cevap nesnesi içerisinde kişisel veri barındıran parametrenin nasıl değiştirileceğine karar verebilmesi için kullanılacaktır.

Üçüncü bölüm ise web servis cevap nesnesi içerisindeki kişisel verilerin ikinci bölümde girilecek kurallara göre içeriğinin değiştirildiği bölümdür.

4.1.1. Uygulamanın birinci bölümü

Uygulamanın ilk bölümünde Şekil 4.1'de gösterildiği üzere yazılım geliştiricilere karşı korunmak istenen web servis cevap nesnesi KVKSİS uygulamasının "SaveJsonObject" metoduna gönderilir. "SaveJsonObject" metodu çağırılırken body kısmında cevap nesnesi ve cevap nesnesinin hangi servis metoduna ait olduğu bilgisi gönderilir. "SaveJsonObject"

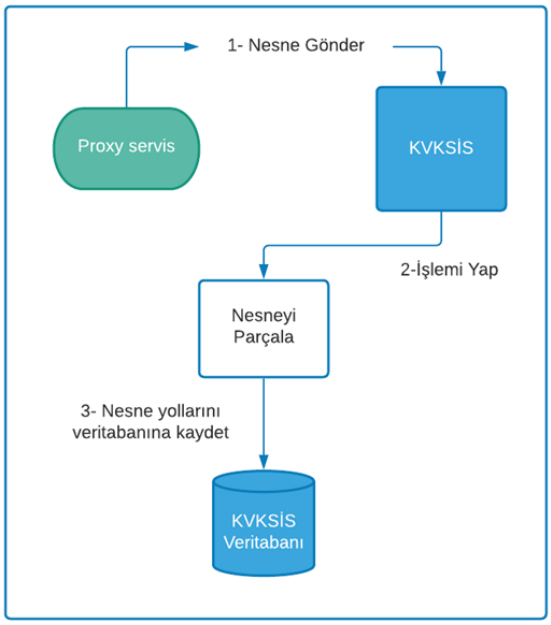
metodu cevap nesnesi içerisindeki bütün parametreler özyineli fonksiyon ile gezilir ve kırılımları ile birlikte veri tabanına kaydedilir. “SaveJsonObject” metodunun çalışma prensibi;

- Gönderilen web servis metodunun sistemde olup olmadığı kontrol edilir. Yani Şekil 4.2’de yer alan metotId KVKSIS uygulamasında bulunup bulunmadığı kontrolü yapılır.
- Web servis nesnesinin içinde yer alan bütün parametrelerinin tüm kırılımları daha sonra kural girilebilmesi ve içeriğinin değiştirilmesi için tespit edilir. Örneğin aşağıdaki gibi bir nesne metoda gönderildiğinde;

```
{
"TcKimlikNo":"54670252582",
"adres":"yol mahallesi 8425 SOKAK 365/52 yenimahalle Ankara",
"aile":[
        {"annead": "ayşe" }
    ]
}
```

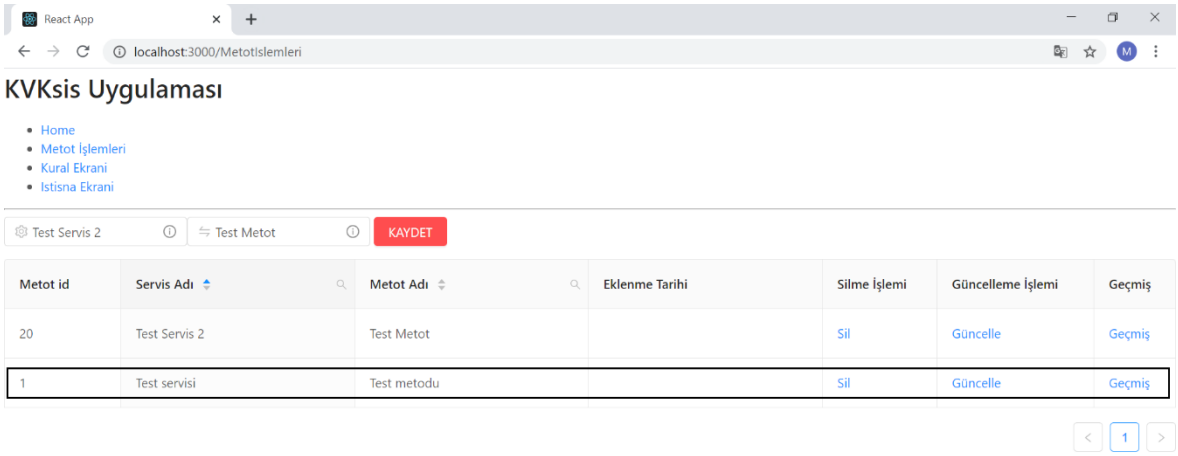
- Web servis nesnesinin içerisinde özyineli fonksiyon (**recursive**) ile gezilir ve pathler çıkarılır. Sonuç olarak veri tabanına aşağıdaki haliyle kaydedilir.

1- TcKimlikNo
2- Adres
3- aile.annead

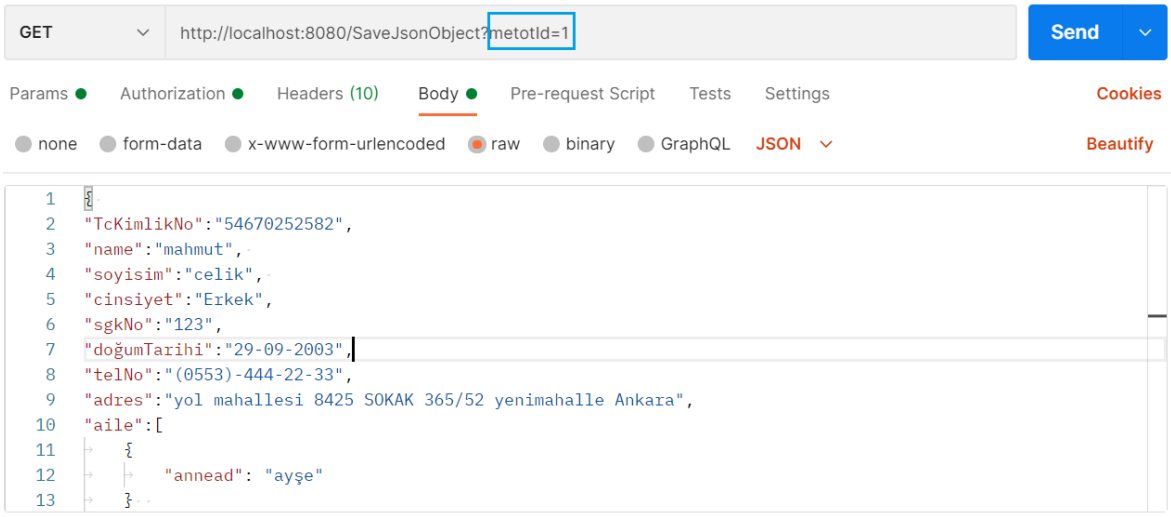


Şekil 4.1. Web servis cevap nesnesinin KVKSİS'e gönderilmesi

Bir örnek ile açıklamak gerekirse KVKSIS uygulamasına tanımlanmış web servis metodunun "id" bilgisi Şekil 4.2'de gösterildiği üzere KVKSIS uygulamasının "Metot İşlemleri" bölümünden bu bilgi alınır. Şekil 4.3 de gösterildiği üzere postman uygulaması kullanılarak (farklı araçlar kullanılabilir) içeriği anonim hale getirilmek istenen web servis cevap nesnesi KVKSIS uygulamasına gönderilir. Şekil 4.3'de gösterildiği üzere Postman aracılığı ile KVKSIS uygulamasının sunmuş olduğu "SaveJsonObject" isimli metoda cevap nesnesi gönderilir. KVKSIS uygulması içerisinde yazılan "SaveJsonObject" metodu içerisinde cevap nesnesi parametreleri çıkarılır ve KVKSIS veri tabanına kaydedilir.



Şekil 4.2. KVKSIS uygulaması metot id öğrenme



Şekil 4.3. Cevap nesnesinin “SaveJsonObject” metoduna gönderilmesi

4.1.2. Uygulamanın ikinci bölümü

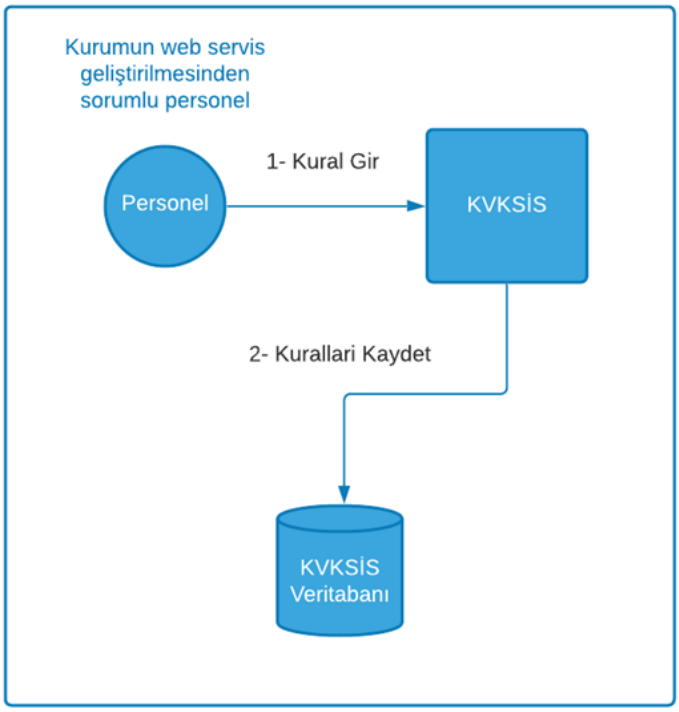
Yazılım geliştiricilerden korunmak istenen web servisin cevap nesnesinin bütün parametreleri birinci bölümde anlatıldığı üzere KVKSIS uygulamasına kayıt işlemi yapıldıktan sonra ikinci bölüme geçilir. İkinci bölüm Şekil 4.4’de tarif edildiği üzere KVKSIS uygulamasının ara yüzü kullanılarak cevap nesnesinin hangi parametresi ne şekilde değiştirileceğine dair kuralların girildiği bölümdür. Örnek vermek gerekirse cevap nesnesinin “ad” parametresinin kişi ismi olduğu yazılıma kural olarak girilmeli ki KVKSIS uygulaması o alanı değiştirirken kişi ismi atayabilsin. KVKSIS uygulaması kişisel verileri değiştirirken belirli bir düzende yeni değer atayacak şekilde tasarlanmıştır.

Şekil 4.5’de gösterildiği üzere KVKSIS uygulamasının ara yüzünde “Kural Ekranı” seçilir. Daha sonra Servis adı, Metot adı, kural girilecek parametre, yöntem ve parametrenin ne olduğu seçilir ve KVKSIS veri tabanına kural kaydedilir.

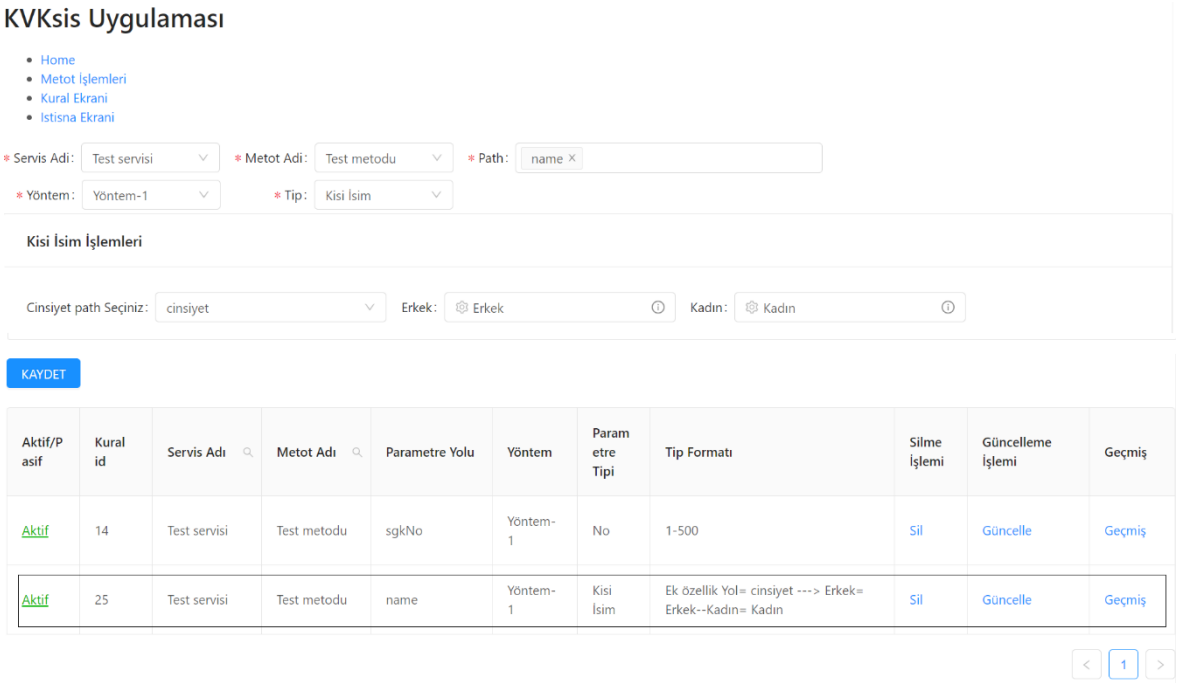
KVKSIS uygulamasında ekranlar da kural girilirken;

- Önceden KVKSIS uygulamasına kaydedilen servislerden kural girilecek servis seçilir.
- Servise ait metot seçilir.

- Seçilen web servisin metoduna birinci bölümde anlatıldığı üzere cevap nesnesi gönderilmiş ise “path” alanına cevap nesnesinin bütün parametreler açılan kutuya otomatik getirilir. Kural tanımlanmak istenen parametre buradan seçilir.
- “Yöntem” alanına her parametre için birden fazla yöntem seçilebilir. Bu yöntemlere göre parametre değiştirilir. Bu yöntemler arasında “***” şeklindeki yöntem seçilirse o parametre “******” şeklinde gösterilir. Diğer yöntemlerin farklı algoritmalar kullanılarak farklı yöntemlerle yeni değer atanır. Karmaşıklığı artırmak için bu alan uygulamaya eklenmiştir.
- “Tip” alanında ise seçilen parametrenin hangi tipte olacağı seçilmelidir. Örneğin parametreye kişi ismi, soyisim, tarih, telefon numarası gibi seçeneklerden uygun olan seçilmelidir. Seçilen bu tipe göre KVKSIS uygulaması seçilen parametreye yeni değer atanır.
- Son olarak eğer seçilen “Tip” için ekstra özellik tanımlanması yapılabilir. örneğin kişilerin isimleri değiştirilmek istendiğinde erkeklere erkek adı kadınlara kadın adı atanacak şekilde düzenleme yapılabilir.
- Son olara “Kaydet” tuşuna basarak kuralı KVKSIS veri tabanına kaydedilmektedir.



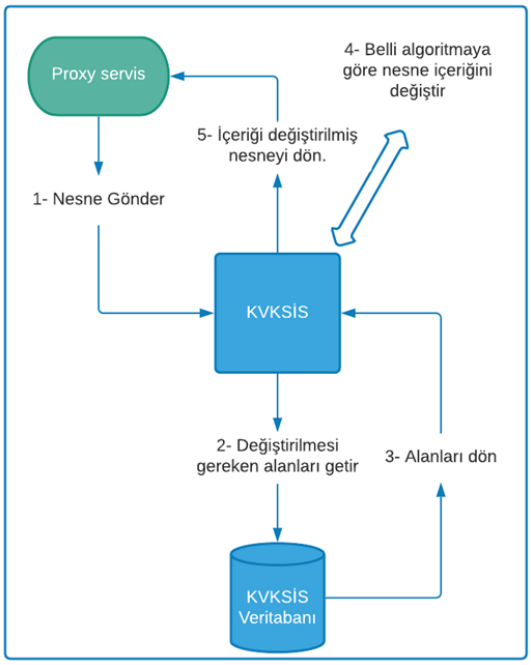
Şekil 4.4. KVKSIS ikinci bölümü



Şekil 4.5. KVKSIS uygulamasına kural girilmesi

4.1.3. Uygulamanın üçüncü bölümü

KVKSIS uygulamasının son bölümünde gizlenmesi gereken web servis cevap nesnesinde ikinci bölümde girilen kurallar ışığında belli bir algoritmaya göre parametrelerin değiştirildiği bölümdür. Şekil 4.6'da gösterildiği üzere web servis cevap nesnesi KVKSIS uygulamasına gönderilir. Uygulama veri tabanından web servis metoduna ait önceden tanımlanmış kurallar getirilir. Veri tabanından getirilen kurallara göre web servis cevap nesnesi içerisindeki kişisel veriler anonim hale getirilir. Anonim hale getirilmiş web servis cevap nesnesi proxy servise geri iletilir. Proxy servisten web servis cevap nesnesi yazılım geliştiriciye iletilir. KVKSIS uygulamasına ikinci adımda girilmiş olan kurallar veri tabanından çekilir ve hangi parametrelere kural girilmişse bu doğrulta da veriler belli bir algoritmaya göre değiştirilir. Bu adımda dikkat edilmesi gereken nokta, gönderilen nesnenin her gönderildiğinde aynı şekilde değiştirilmesi gerekir. Çünkü yazılım geliştiriciler bir kişiyi web servisten sorguladığında her defasında aynı kişi için aynı ismi görmek ister aksi takdirde yazılım geliştirirken mantıksal hatalar alabilir. KVKSIS uygulamasının web servis cevap nesnesinin değiştirilmesi için hizmete sunduğu "ConvertEntitybyRule" metodu çağırıldığında aşağıdaki işlemler gerçekleştirilir.

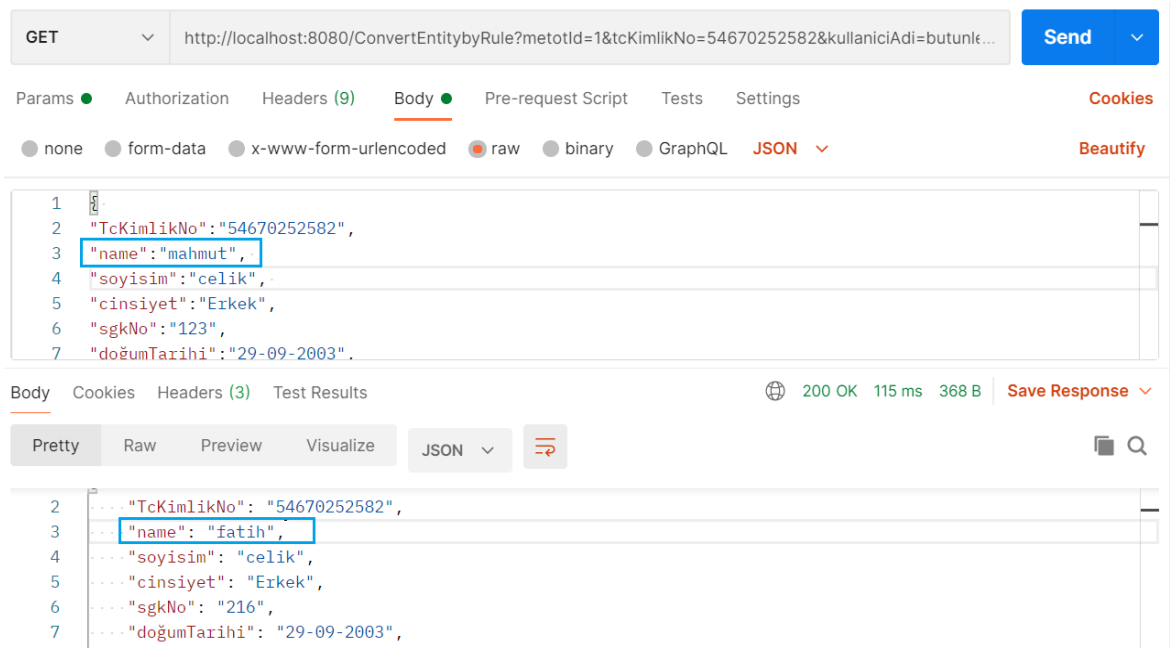


Şekil 4.6. KVKSIS parametre değiştirme adımı

- "ConvertEntitybyRule" metoduna Şekil 4.7'de gösterildiği gibi metotId ye ait kurallar veri tabanından çekilir.
- Çekilen kurallar tek tek işleme alınır.
- Web servis cevap nesnesinin hangi parametresine kural konulmuş ise özyineli fonksiyon (**recursive**) ile cevap nesnesinin içinde gezilir ve parametrenin o kırılımına gelinir.
- Kural da tanımlanan "Yöntem", "Tip" ve varsa "Ek özelliklere" göre (geliştirilen bir algoritma kullanılarak) parametreye yeni bir değer atanır.
- Gönderilen web servis metoduna ait girilen tüm kurallar uygulandıktan sonra "ConvertEntitybyRule" metodu gelen cevap nesnesinin içeriği değişmiş ancak yapısı korunmuş şeklide geri döner.
- KVKSIS uygulamasının görevi tamamlanmış olur.

Örnek olarak birinci ve ikinci bölümde gönderilen web servis cevap nesnesinin manuple edilme işlemini postman kullanılarak gösterilmiştir. KVKSIS uygulamasının web servis nesnesinin değiştirilmesi için hizmete sunulan "ConvertEntitybyRule" isimli rest servis metoduna değiştirilmek istenen web servis nesnesi gönderilir. KVKSIS uygulamasının "ConvertEntitybyRule" metodu içerisinde ikinci bölümde tanımlanan kurallara göre gerekli parametrelerde değişiklik yapılır. Kural ekranında sadece "name" alanına kural

girildiği için Şekil 4.7'de gösterildiği üzere sadece web servis nesnesinin "name" alanı değiştirilmiştir.



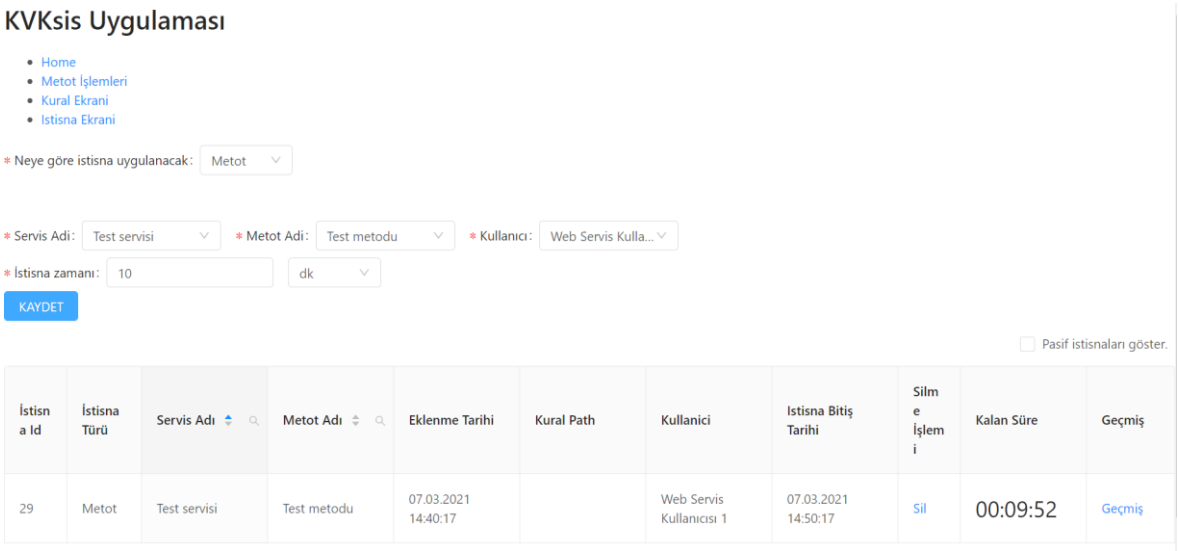
Şekil 4.7. Web servis nesnesinin değiştirilmesi

4.1.4. KVKSIS uygulamasının ek bölümü (istisna bölümü)

Kurumlardaki yazılım geliştiricilerin bazı durumlarda web servis sorgularının gerçek verilere erişim sağlaması gerekebilmektedir. Örneğin vatandaşlara verilen bir hizmette genel bir hata olmamakla birlikte bir vatandaşın işleminde hata alınması sebebi ile yazılım geliştirici debug yaparak sorunu tespit etmesi gerektiği durumda web servisten gelen vatandaşa ait gerçek bilgilere ihtiyaç duyacaktır. Bu gibi durumlara istinaden KVKSIS uygulamasına "Istisna" tanımlama özelliği eklenmiştir. İstisna işleminde kullanıcılara metot bazlı ayrıcalıklar tanımlanabilmektedir. Tanımlanan bu ayrıcalık zaman kısıtı konulacak şekilde tasarlanmış ve kullanıcıya tanımlamanın süresi dolduğunda istisna otomatik olarak iptal edilmektedir. Böylece kullanıcılara sınırsız erişim açılmasının önüne geçilmektedir. İstisna özelliği Şekil 4.8'de gösterildiği gibi ekranlardan girilmektedir.

- İstisnalar metotta yer alan bütün kurallara veya metodun belirli parametrelerine konulabilmektedir. Yani sadece bazı alanlara orijinal veriler erişim izni verilebilir veya o metoda ait bütün verilerin orijinalleri yazılım geliştiricilere eriştirilebilir.

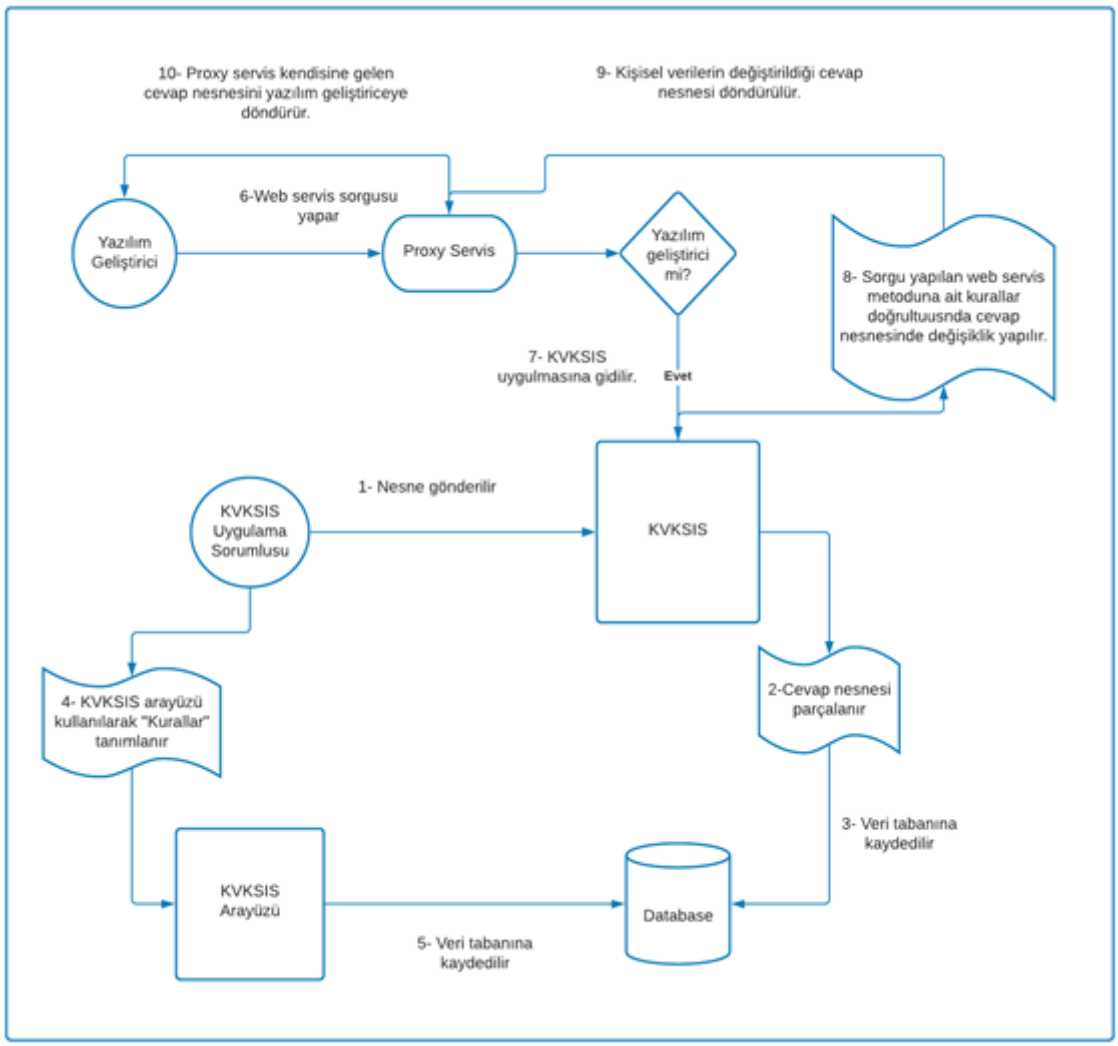
- İstisna konulmak istenen Servis ve Servis Metodu seçilir.
- İstisna uygulanacak web servis kullanıcısı seçilir. (Kurumdaki bütün kullanıcılara istisna konulamaz sadece ihtiyacı olan kişilere istisna konulabilir.)
- Yazılım geliştiriciden kişilere ait orijinal verileri ne kadar süre görmesi gerektiği öğrenilir ve belirli bir zaman verilir. Verilen zaman dolduğunda tanımlanan istisna otomatik kaldırılır ve yazılım geliştirici yeniden gizlenmiş verileri görür.



Şekil 4.8. KVKSIS uygulamasında istisna ekranı

4.2. Uygulama Genel Akış Şeması

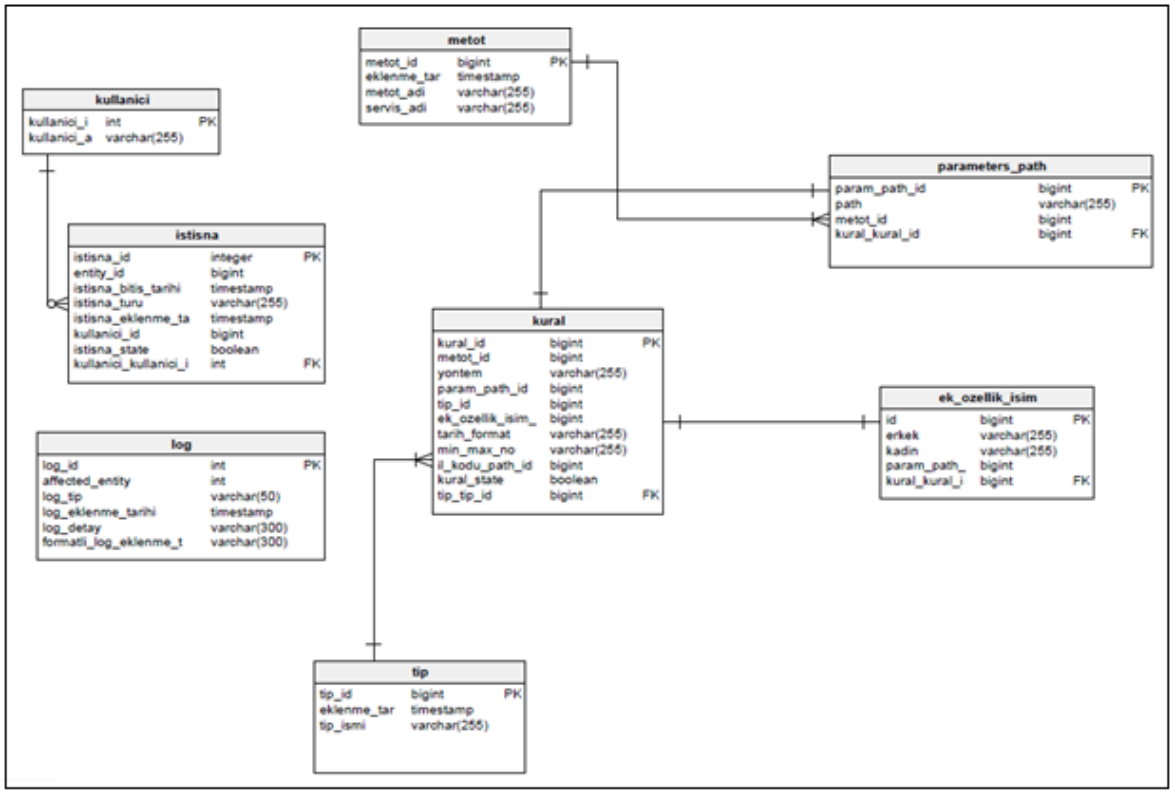
KVKSIS uygulamasının asıl amacı uygulamaya girilen kurallara göre web servislere ait cevap nesnelerinin içeriğini değiştirmektir. Verileri değiştirirken bir kişinin verisini her sorgulandığında aynı şekilde değer verilmesini sağlayacak şekilde kurgulanmıştır. KVKSIS uygulamasına ait cevap nesnesi kaydetme, kural tanımlama ve cevap nesnesi içeriği değiştirme adımları Şekil 4.9'da birlikte gösterilmiştir.



Şekil 4.9. KVKSIS uygulamasının genel akış şeması

4.3. Veri Tabanı Yapısı

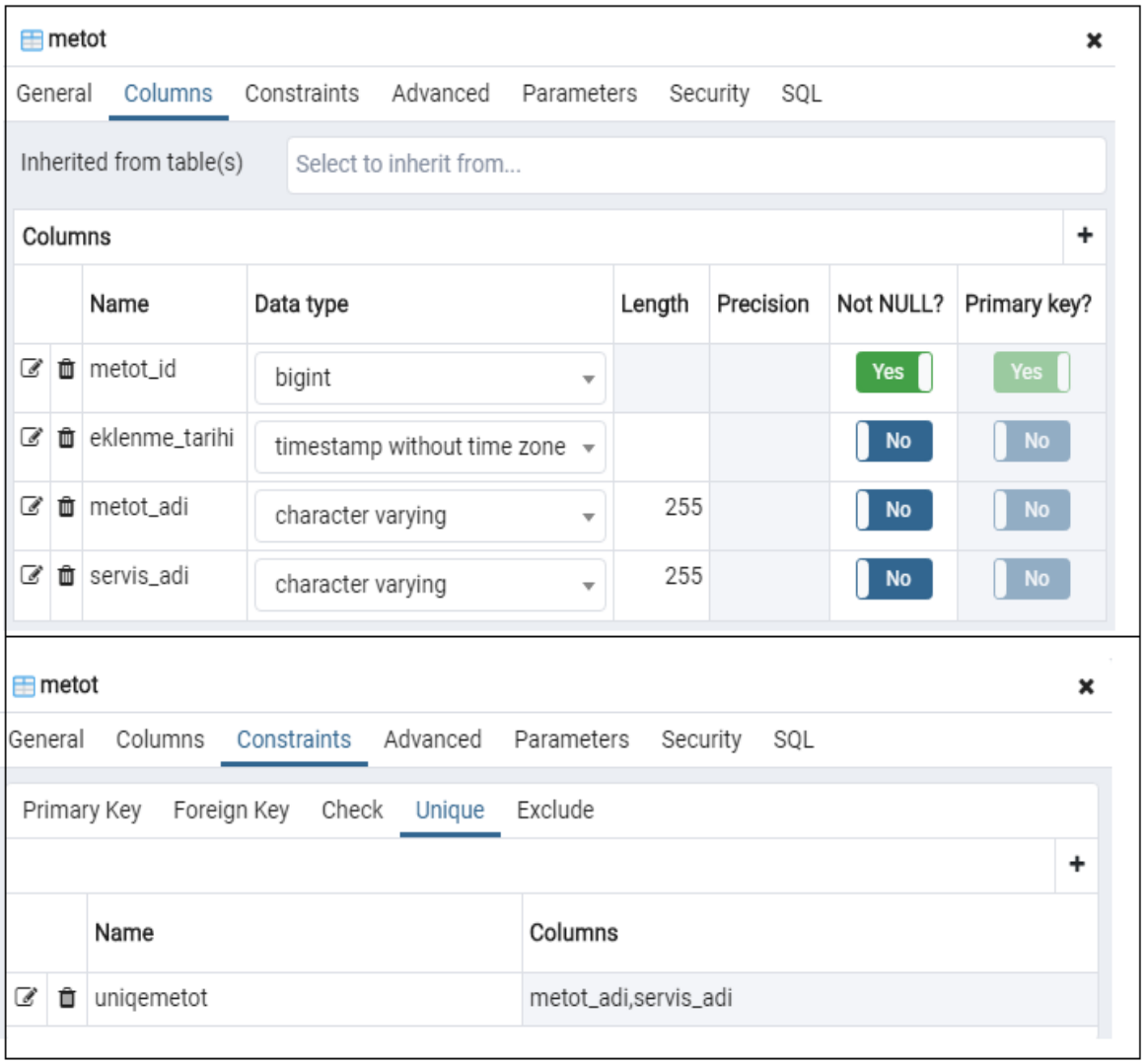
KVKSIS uygulamasınında kullanılan metot, kullanici, kural, parameters_path, tip, istisna ve log tabloları postgreSQL veri tabanının genel yapısı Şekil 4.10'de görüldüğü gibi tasarlanmıştır.



Şekil 4.10. KVKSIS uygulamasının veri tabanı şeması

4.3.1. Metot tablosu

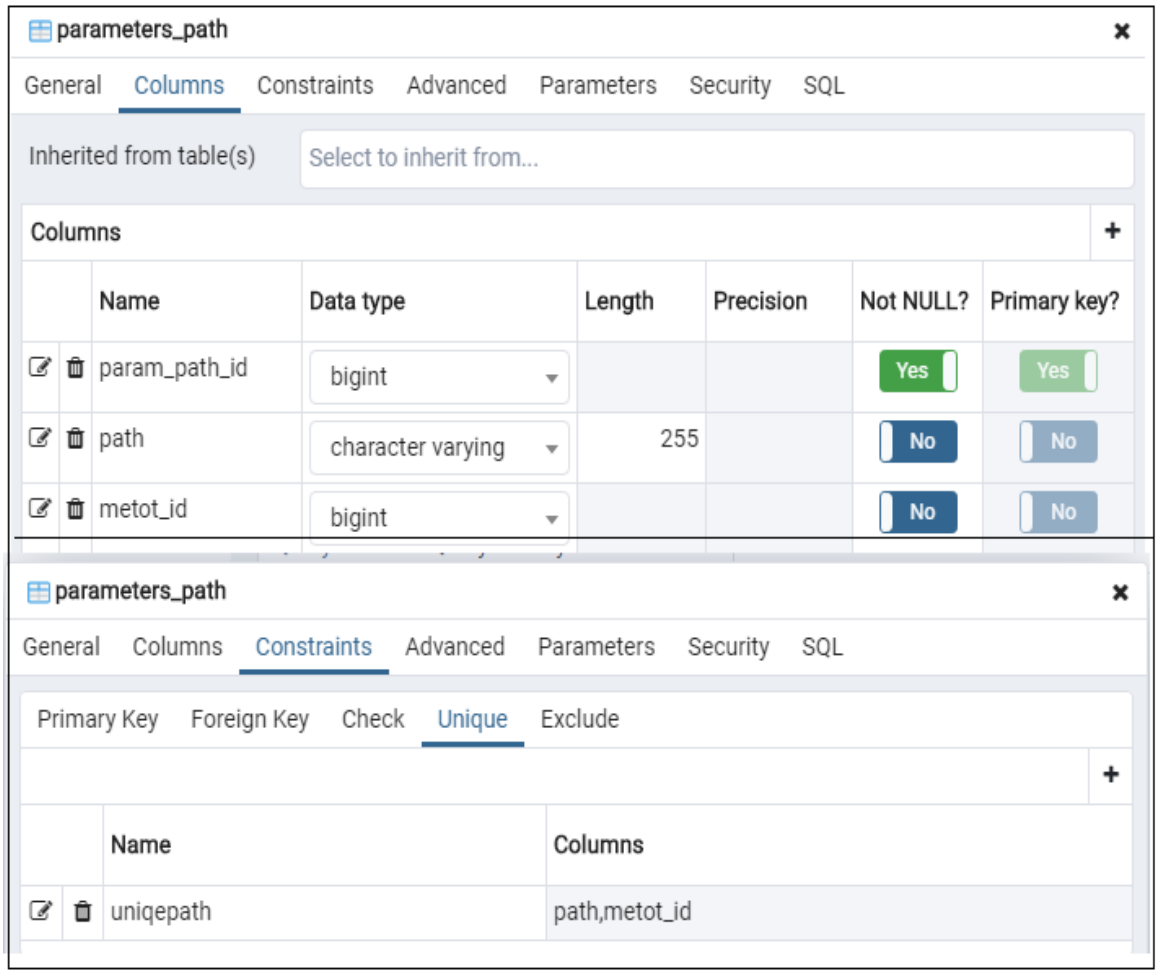
KVKSIS uygulamasının cevap nesnesi üzerinde işlem yapacağı servislerin bilgilerinin yer aldığı tablodur. İçerisinde işlem yapılacak web servis metotlarının cevap nesnelerine ait parametrelerin yollarının tutulduğu parameters_path tablosu ile metot_id alanı üzerinden aralarında bağlantı kurulmuştur. Bir metoda ait birden fazla parametre olacak şekilde (one to many) ilişki kurulmuştur. Şekil 4.11'de gösterildiği üzere metot tablosunda metot_adi ve servis_adi beraber bir adet (unique) olacak şekilde kısıt konulmuştur.



Şekil 4.11. Metot tablosu ve uygulanan kısıtlamalar

4.3.2. Parameters path tablosu

Üzerinde işlem yapılacak web servise ait cevap nesnelerinin içerisinde yer alan parametrelerin yollarının tutulduğu tablodur. Buradaki yollara göre KVKSIS uygulaması hangi parametrede işlem yapacağını belirlemektedir. Metot tablosu ile (one to many) ilişki kurulduğundan bahsedilmişti. Ayrıca Cevap nesnelerine ait parametrelerin nasıl değiştirilmesine karar verildiği kural tablosu ile parameter_path_id alanı üzerinden bağlantı kurulur. Bir parametreye birden fazla kural girilecek şekilde (many to one) ilişki kurulmuştur. Şekil 4.12'de gösterildiği üzere parameter_path tablosu üzerinde path ve metot_id beraber bir adet(unique) olacak şekilde kısıt konulmuştur.



Şekil 4.12. Parameter path tablosu ve uygulanan kısıtlamalar

4.3.3. Kural tablosu

Üzerinde işlem yapılacak web servis metoduna ait cevap nesnesinin parametrelerinden hangileri gizlenmesi gerekiyorsa KVKSIS uygulamasına ait web ara yüzünde girilecek kuralların kaydedildiği tablodur. Kural tablosu cevap nesnesi parametresinin hangi türde olduğunun tutulduğu Tip tablosu ile tip_id üzerinden (Many to One) ilişki kurulmuştur. İsim türündeki parametrelere ek özellik verilmek istediğinde örneğin kadın erkek olarak ayırmak istendiğinde özelliklerinin tutulduğu ek_özellik_isim tablosu ile (One to One) ilişki id parametresi üzerinden kurulmuştur. Şekil 4.13'de gösterildiği gibi metot_id,param_path_id alanları aynı anda tek (unique) olacak şekilde kısıt konulmuştur.

Şekil 4.13. Kural tablosu ve uygulanan kısıtlamalar

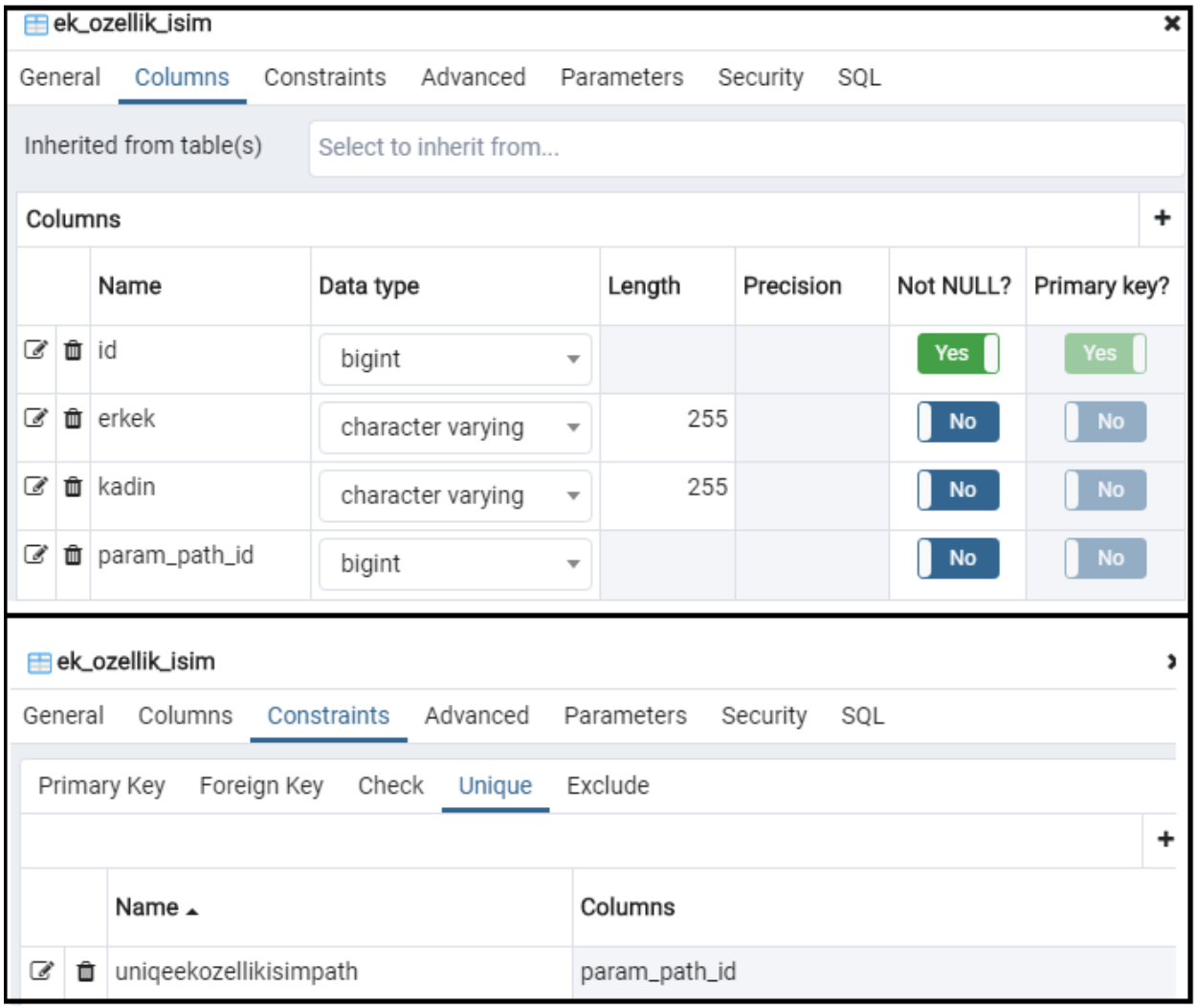
4.3.4. Tip tablosu

Cevap nesnesinin parametrelerinin hangi türde olduğunun tutuldu tablodur. Kural tablosu ile (One to Many) ilişki tip_id alanı üzerinden kurulmuştur.

4.3.5. Ek özellik isim tablosu

Cevap nesnesinin parametresinin türü isim ise ve ek özellik girilmek istenirse örneğin kadın kişilere kadın erkek kişilere ise erkek ismi atanması istenirse kadın ve erkek belirleyicisinin ne olduğu bu tabloda tutulmaktadır. Kural tablosu ile (one to one) ilişki ek_ozellik_isim_id üzerinden kurulmuştur. Şekil 4.14'de gösterildiği üzere her

parametreye sadece bir tane ek özellik girilebilmesi için “param_path_id” alanı tek (unique) olarak kısıtlanmıştır.



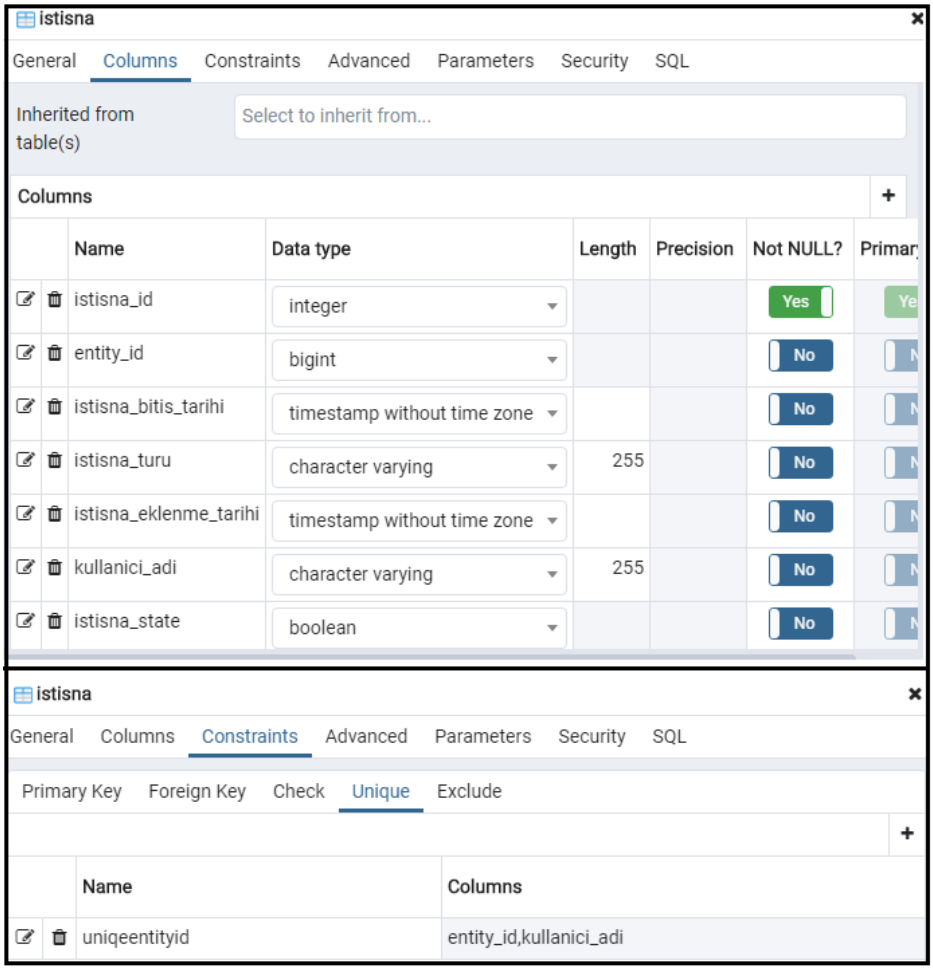
Şekil 4.14. Ek özellik isim tablosu ve uygulanan kısıtlamalar

4.3.6. Kullanıcı tablosu

Proxy servisinden web servis sorgusu yapan web servis kullanıcılarının bilgisinin tutulduğu tablodur. Kuralın uygulanıp uygulanmayacağı yani tanımlanan kullanıcıya bir istisna tanımlanıp tanımlanmadığının anlaşıldığı istisna tablosu ile her kullanıcıya birden fazla istisna tanımlanabilecek şekilde (one to many) kullanıcı_id üzerinden ilişki kurulmuştur.

4.3.7. İstisna tablosu

Yazılım geliştiricilerin bazı durumlarda web servisten gelen gerçek verilere ihtiyaç duymaktadır bu durumlara karşı bazı kişilere bazı parametre veya metotlara istisna konulabilir ve konulan bu istisnalar bu tabloda tutulmaktadır. Şekil 4.15'de gösterildiği üzere "entity_id,kullanici_adi" alanlarının beraber tek (unique) olarak kısıtlanmıştır.



Şekil 4.15. İstisna tablosu ve uygulanan kısıtlamalar

4.3.8. Log tablosu

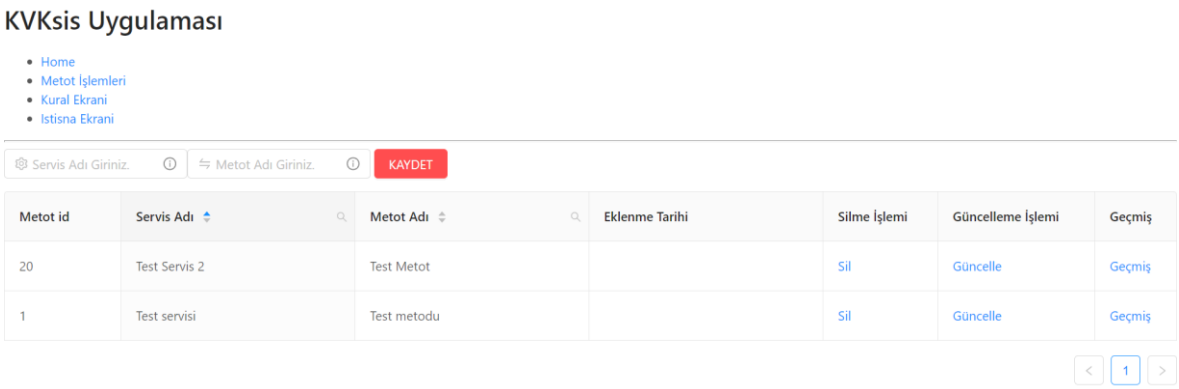
KVKSIS uygulamasında yapılan her işlemin kaydı tutulmaktadır. Kimin kural girdiği kimin istisna girdiği şeklindeki kayıtlar bu tabloda tutulmaktadır.

5. KVKSIS UYGULAMASININ KULLANIMI

Kişisel Verilerin Korunması Sistemi uygulaması kurumlardan alınan web servislerdeki kişisel verilerin yazılım geliştiricilere karşı korunması için tasarlanmıştır. KVKSIS uygulamasının ara yüzü web servise ait bilgileri, web servis cevap nesnesinin parametrelerini, bu parametrelerin nasıl değiştirilmesi gerektiğine dair kuralları ve son olarak bazı olaylar karşısında bazı kullanıcılara istisna tanımlanması işlemlerinin yapılabilmesi adına tasarlanmıştır. Bu sistemin nasıl kullanılacağı bu bölümde açıklanmıştır.

5.1. Metot İşlemleri

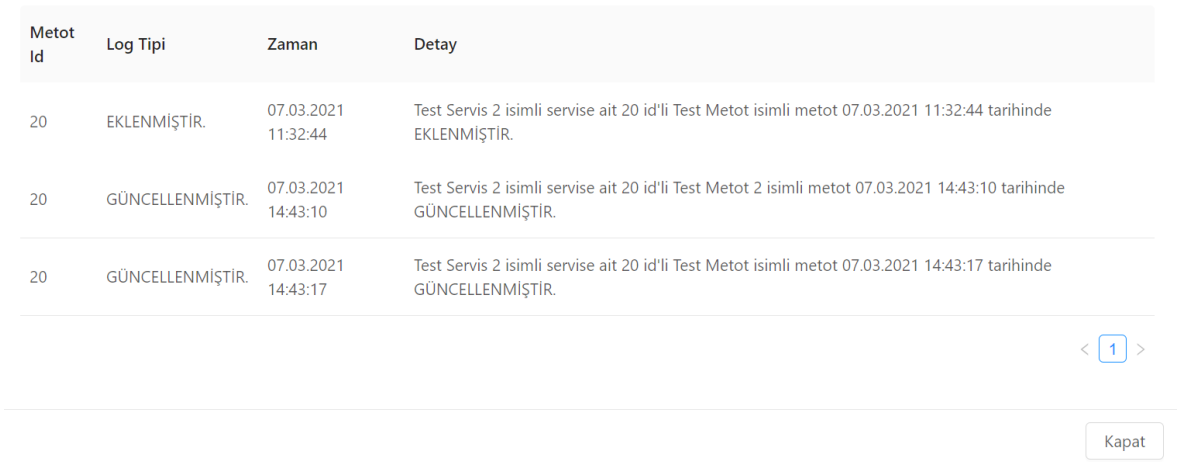
Metot işlemleri ekranında yazılım geliştiricilerden korunması gereken web servisin bilgilerinin kaydetme, silme ve güncelleştirme işlemlerinin yapıldığı bölümdür. Şekil 5.1'de görüldüğü üzere web servisin adı ve metot adı girildikten sonra "KAYDET" tuşuna basılarak web servis bilgileri kaydedilmektedir. Kayıt işlemi yapılan web servisin adında veya metot adında güncelleştirme işlemi gerekirse Şekil 5.1'de gösterildiği üzere "Güncelle" tuşuna basılır ve datagrid'de o satır yazılabilir hale gelir ve gerekli güncelleştirmeler yapıldıktan sonra "Güncelle" tuşunun yerine gelecek olan "Kaydet" tuşuna basıldıktan sonra gerekli değişiklikler veri tabanına yansıyacaktır. Web servise ihtiyaç kalmadığı zaman "Sil" tuşuna basarak web servis sistemden kaldırılacaktır.



Şekil 5.1. Metot işlemleri ekranı

Ayrıca web servis üzerinde yapılan değişikliklerin "Güncelleme" gibi log kayıtları tutulmakta ve bu kayıtlara "Geçmiş" tuşuna basıldığında Şekil 5.2'de gösterildiği gibi bir

popup aracılığı ile bütün değişiklikler görüntülenebilmektedir. Yapılan güncellemeler sistemin çalışmasını etkileyeceği ve her şeyin kontrol altında tutulabilmesi adına bu adım eklenmiştir.



Şekil 5.2. Metot işlemleri geçmiş ekranı

5.2. Kural İşlemleri

KVKSIS uygulamasının amacı olan web servislerin cevap nesnelerinin içeriğinin nasıl değiştirileceğine dair kuralların girildiği ekrandır. Şekil 5.3'de görüldüğü üzere;

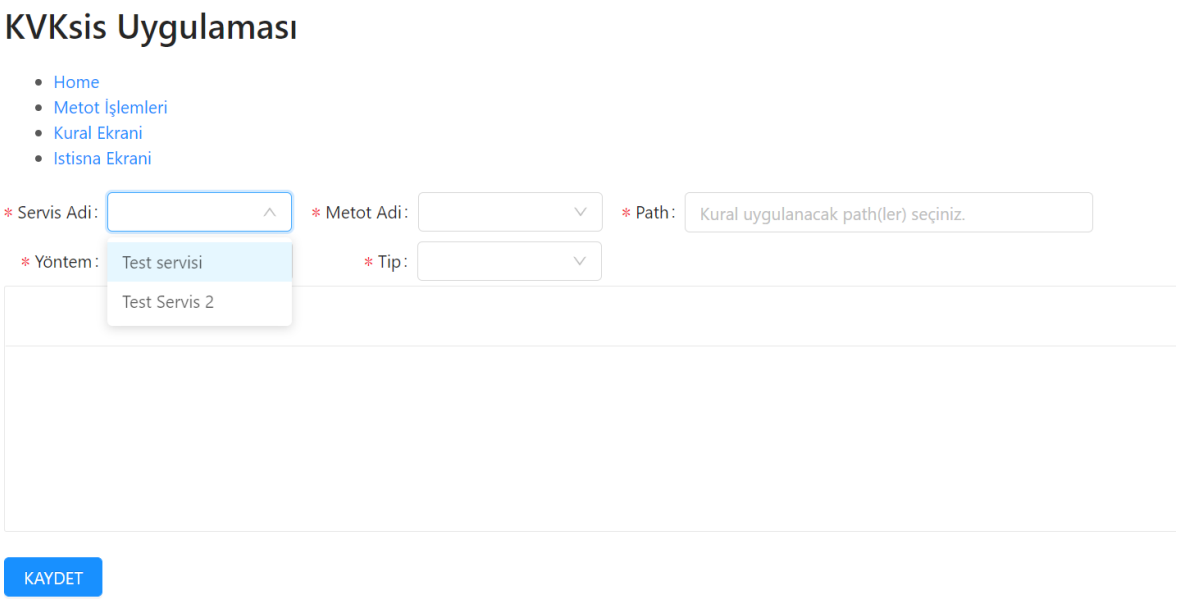
- "Servis Adı" alanına metot işlemleri ekranları aracılığı ile eklenen servislerin isimleri otomatik gelmektedir. Bu servislerden hangisi üzerinde işlem yapılacaksa o servis seçilir.
- Seçilen servise ait metot isimleri "Metot Adi" alanına otomatik olarak gelmekte ve hangi metot üzerinde işlem yapılacaksa o metot seçilmelidir.
- 4.1.1. bölümde anlatıldığı üzere web servis cevap nesnelerine ait parametrelerinden hangisine kural girilecekse o seçilmelidir.
- "Yöntem" alanında ise web servisin parametresinin hangi yöntem ile gizleceğinin seçildiği bölümdür. Parametrenin içeriğinin değiştirilmesini karmaşıklaştırmak için birden fazla yöntem eklenebilmekte veya bazı alanların değiştirilmesi mümkün değilse "Yöntem-**" seçilerek o parametreye "*********" atanması sağlanabilmektedir.

- “Tip” alanı ise “Path” bölümünde seçilen parametrenin ne türden olduğuna karar verildiği bölümdür. Kişi Adı mı yoksa şehir mi gibi seçimlerin yapıldığı bölümdür. Seçilen bu tipe göre KVKSIS uygulaması değer atayacaktır.
- Ayrıca bazı alanlara ek özellik girilebilmesi gerekebilmektedir. Örneğin kişi isim alanlarına erkek bireylere erkek kadın bireylere kadın ismi tanımlanması talep edilebilmektedir. Bu durumda Şekil 5.4’de gösterildiği gibi. “Tip” bölümü “Kişi İsim” olarak seçilmesi durumunda aşağıda yer alan bölümde “Cinsiyet path seçiniz” kısmında web servis içerisinde kişilerin cinsiyetlerinin verildiği parametre seçilmelidir. Daha sonra her servisten cinsiyet alanı farklı şekillerde gelebileceği için “K , KADIN, Kadın, Kadin” gibi seçilen serviste olanların nasıl geldiği KVKSIS uygulamasına tanıtılır. KVKSIS uygulaması da buradaki tanımlamalara göre ilgili parametreye cinsiyetine uygun isimler atayabilecektir.

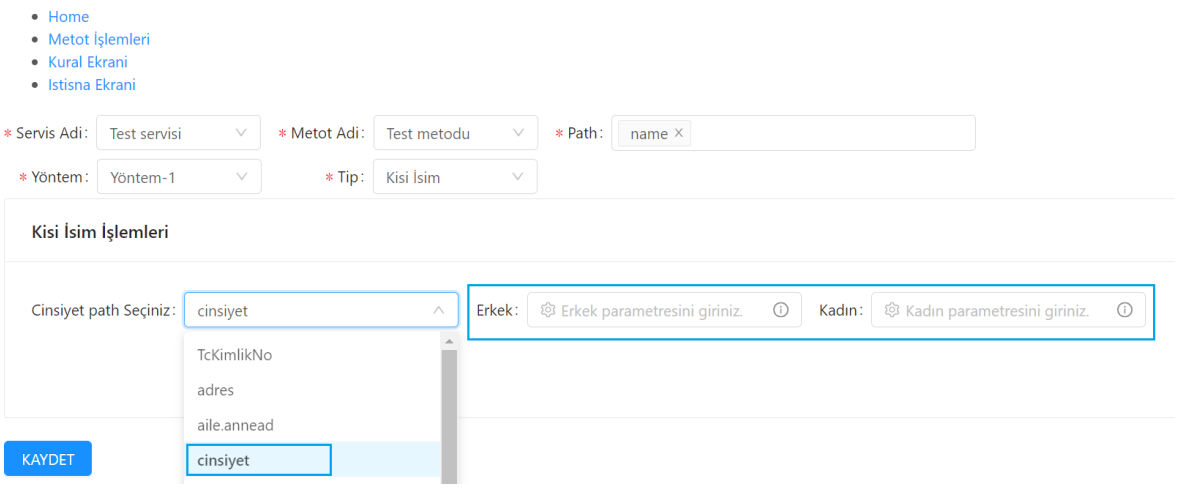
Tanımlanan kurallar Şekil 5.5’de gösterildiği gibidir. Tanımlanacak kurallar bir sorun olduğu zaman veya talep edildiğinde pasif/aktif yapılabilmektedir. Pasife alınan kurallar aktif edilmeyene kadar KVKSIS tarafından göz ardı edilecek ve herhangi bir işlem yapmayacaktır. Tanımlanan kurallarda herhangi bir güncelleme işlemi yapılmak istendiğinde “Güncelle” tuşuna basılır ve güncellenmek istenen kurala ait bütün bilgiler Şekil 5.6’da gösterildiği gibi kural ekleme bölümüne yansıtılmakta ve burada istenilen değişiklikler tanımlanır ve “GÜNCELLE” tuşuna basıldığında kuralda gerekli değişiklikler KVKSIS uygulamasına eklenmiş olur.

Ayrıca kurallar üzerinde yapılan bütün değişiklikler kayıt altına alınır ve olası zafiyet durumlarında veya hata durumlarında bu log kayıtlarına bakılmaktadır. Bu log kayıtları Şekil 5.5’de gösterildiği gibi “Geçmiş” tuşuna basılır ve Şekil 5.7’deki gibi o kayda ait yapılan bütün değişikliklerin dökümü görüntülenmiş olur.

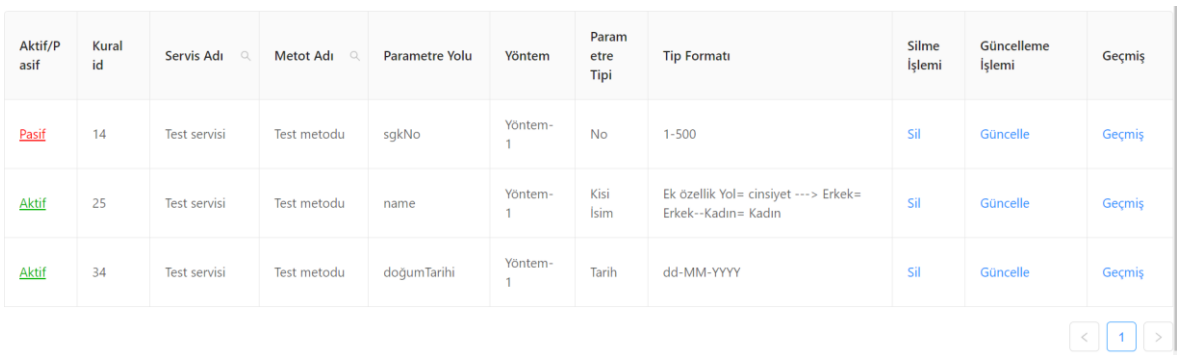
Tanımlanan kurallar istenildiği takdirde Şekil 5.5’de gösterilen “Sil” tuşuna basılarak kural silinebilmektedir.



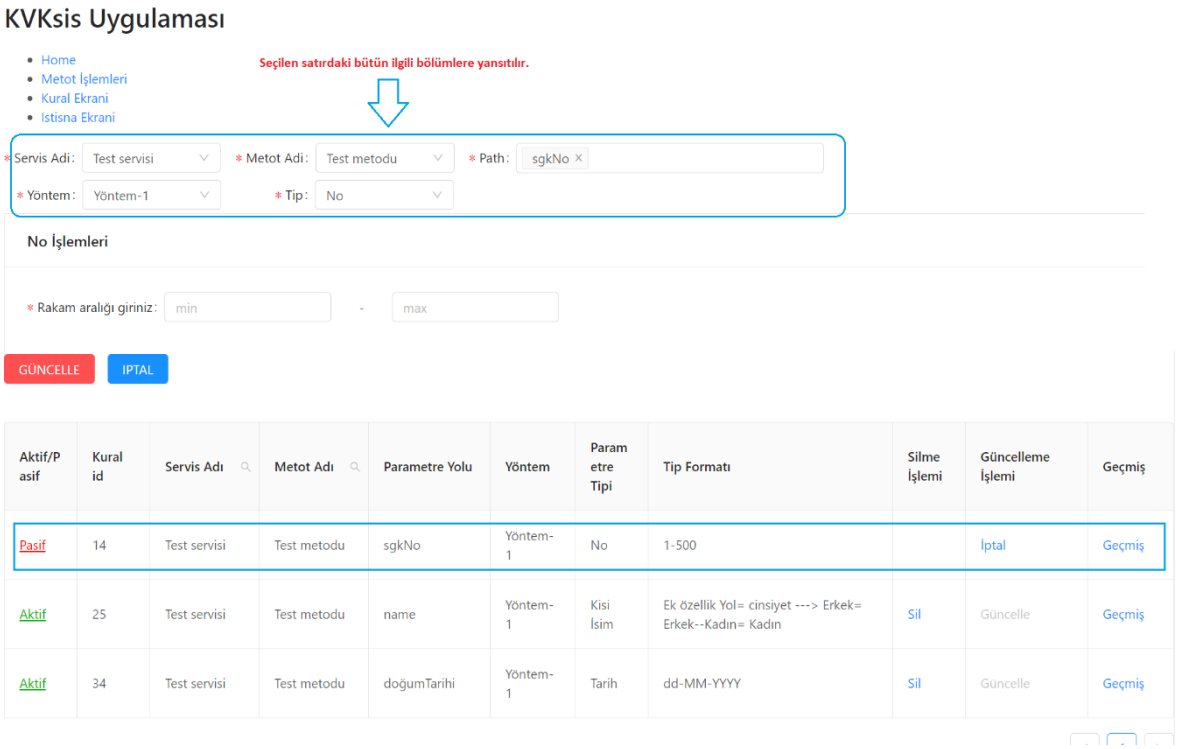
Şekil 5.3. Kural ekranı



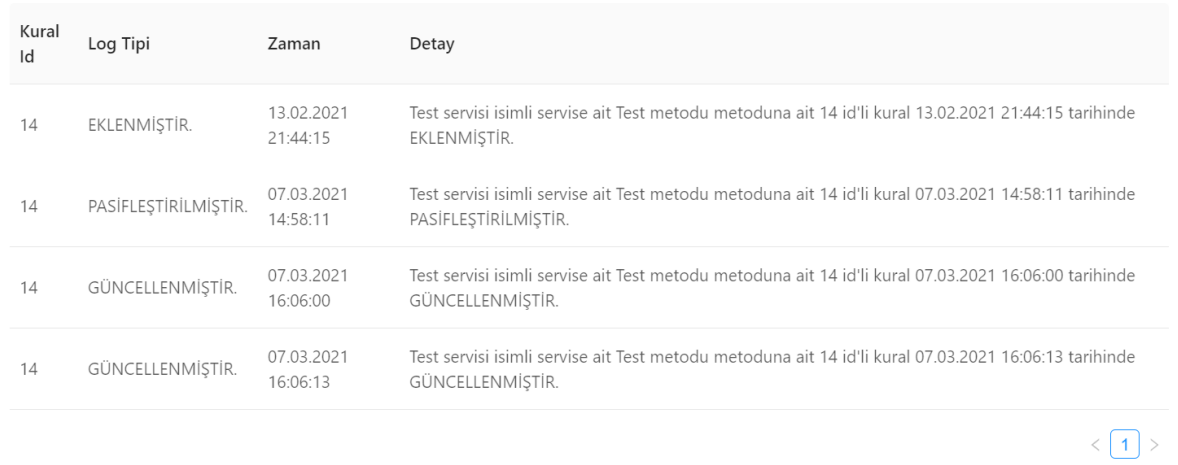
Şekil 5.4. Kişi isimlerinin cinsiyete göre verilebilmesi için tasarlanan ek özellik bölümü



Şekil 5.5. Eklenen kuralların gösterimi



Şekil 5.6. Kuralların güncellenmesi işlemi



Şekil 5.7. Geçmiş sekmesi

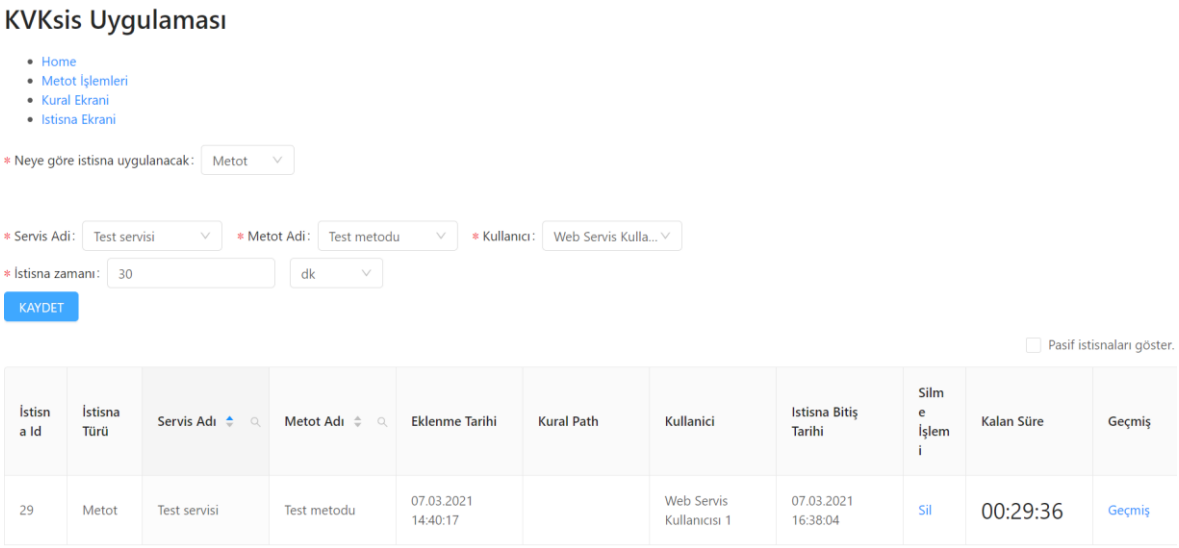
5.3. İstisna Ekranı

Yazılım geliştiriciler bazı durumlar karşısında web servislerden gelen gerçek verilere ihtiyaç duyabilmekteler. Bu gibi durumlarda yetkililerden gerekli izinler alındıktan sonra KVKSIS uygulamasında kullanıcı bazlı istisnalar tanımlanabilmektedir. Bu istisnalar metot veya parametre bazlı konulabilmektedir. İstisnalar belli bir süreliğine verilmektedir

tanımlanan süre bittiğinde istisnada otomatik olarak kalkacaktır. Şekil 5.8'de görüldüğü üzere;

- İstisna uygulanacak metoda ait parametrelerin tamamına erişim ihtiyacı varsa "Neye göre istisna uygulanacak" alanı "Metot" seçilmeli eğer metot parametrelerinden bazılarına ihtiyaç duyuluyorsa "Kural" seçilir ve sadece seçilecek kurallara istisna uygulanabilmektedir.
- İstisna uygulanmak istenen "Servis Adi" ve "Metot" seçilir.
- Eğer "Neye göre istisna uygulanacak" alanı "Kural" seçilmişse istisna uygulanacak "Kural Path"'i seçilir.
- İstisnaya ihtiyaç duyan web servis kullanıcısı "Kullanıcı" alanında seçilir.
- İstisnanın süresi seçilir ve "KAYDET" tuşuna basılır.

İstenildiğinde zamanı dolmadan istisna "Sil" tuşuna basılarak kaldırılabilmektedir. Ayrıca istisnalara ait log kayıtları tutulmaktadır. Şekil 5.9'da gösterildiği üzere istisnanın eklenme, kaldırılma zamanları gibi bilgilerin detayları görüntülenebilmektedir. Bu log kayıtları önemlidir çünkü istisna konulması kişisel verilerin gizliliğini kaldıran bir hamle olduğu için istisnanın ne zaman, kimin, kime istisna tanımladığının kayıt altına alınması önemlidir.



Şekil 5.8. İstisna ekranı

Şekil 5.9. İstisna ekranı geçmiş sekmesi

6. KVKSIS UYGULAMASININ KURULUMU

Tasarlanan kişisel verilerin korunması sistemi sunucu (backend) ve önyüz (frontend) olmak üzere 2 ayrı bölümden oluşmaktadır. Bu nedenle sunucu ve önyüz uygulamalarının kurulumları ayrı şekilde yapılmakta ve uygulamalar aralarında web servisler aracılığı ile iletişim kurmaktadır.

6.1. Ön Yüz Kurulumu (React)

React ile geliştirilmiş olan ön yüz Tomcat uygulama sunucu üzerine kurulumu gerçekleştirilmiştir. Tomcat uygulama sunucusu indirildikten sonra gerekli ayarlamaları yapmak için "conf" klasöründe yer alan "tomcat-users.xml" dosyası içerisinde Şekil 6.1'de gösterilen yetkili kullanıcıya ait bilgilerin tanımlamaları yapılır.

```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<!--
  NOTE:  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE:  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
 <role rolename="tomcat"/>
  <user username="admin" password="12345>" roles="tomcat"/>
</tomcat-users>
```

Şekil 6.1. Tomcat kullanıcı bilgileri düzenleme işlemi

Tomcat sunucusu standart olarak 8080 portunu kullanmaktadır. 8080 portu çok kullanılan bir port olduğu ve KVKSIS uygulaması ile herhangi bir çakışmanın olmaması için Tomcat sunucusunu 8084 portunu kullanarak çalışabilmesi için "server.xml" dosyası içerisinde Şekil 6.2'de gösterildiği gibi gerekli tanımlamalar gerçekleştirilir.

```
<!-- A "Connector" represents an endpoint by which requests are received
     and responses are returned. Documentation at :
     Java HTTP Connector: /docs/config/http.html
     Java AJP  Connector: /docs/config/ajp.html
     APR (HTTP/AJP) Connector: /docs/apr.html
     Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8084" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

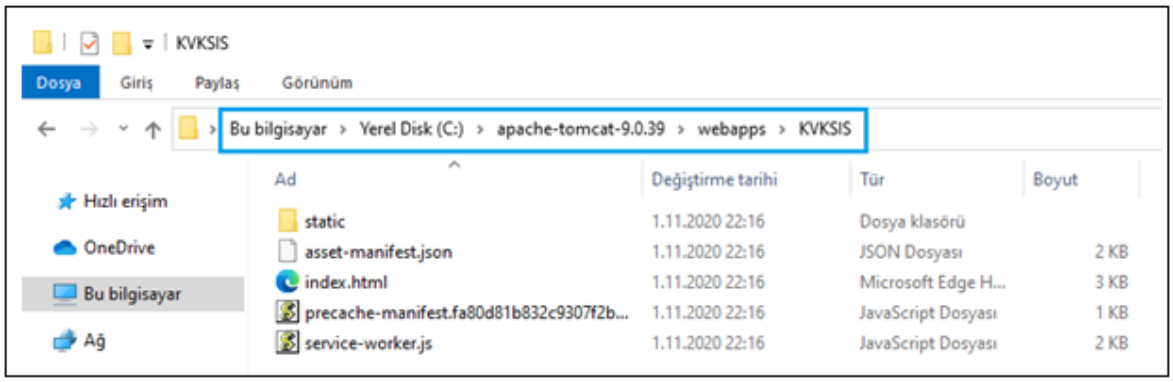
Şekil 6.2 Tomcat sunucusunun port ayarı

Tomcat uygulama sunucusunun ayarları gerçekleştirildikten sonra ön yüz uygulamasının yazıldığı "Visual Studio Code" editörü ile kişisel verilerin korunması sistemi uygulamasının kurulum dosyalarını üretmek için Şekil 6.3'de gösterildiği gibi "npm run build" kodu editörün terminalinde çalıştırılmaktadır.

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\mahmutPC\Desktop\YAZILIM_CALISMASI\React> npm run build

> maximillan_egitim@0.1.0 build C:\Users\mahmutPC\Desktop\YAZILIM_CALISMASI\React\maximillan_egitim
> react-scripts build

Creating an optimized production build...
```

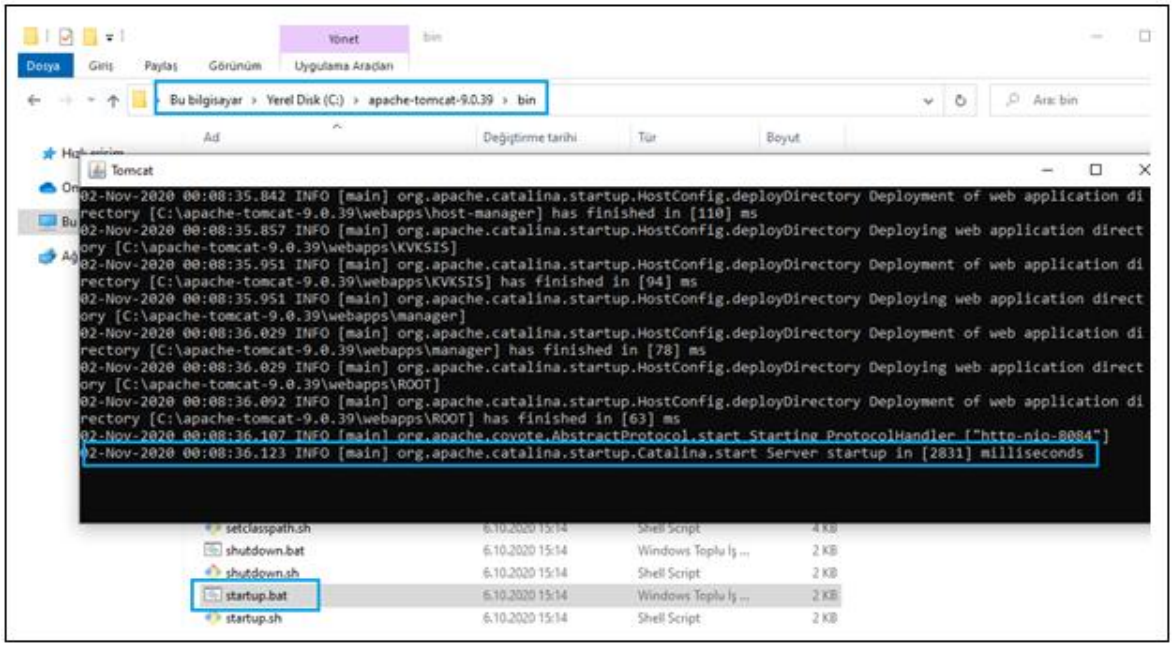
Şekil 6.3. React projesi kurulum dosyası üretme

Kişisel verilerin korunması sistemi uygulamasının tomcat sunucusu üzerinde çalıştırılabilen dosyaların proje içerisinde yer alan "build" klasöründe oluşmaktadır. Oluşturulan bu dosyalar tomcat sunucusu içerisinde yer alan "webapps" klasörünün altında "KVKSIS" isminde bir klasör oluşturup Şekil 6.4'de gösterildiği gibi kurum dosyaları "websis" klasörü içerisine eklenir.



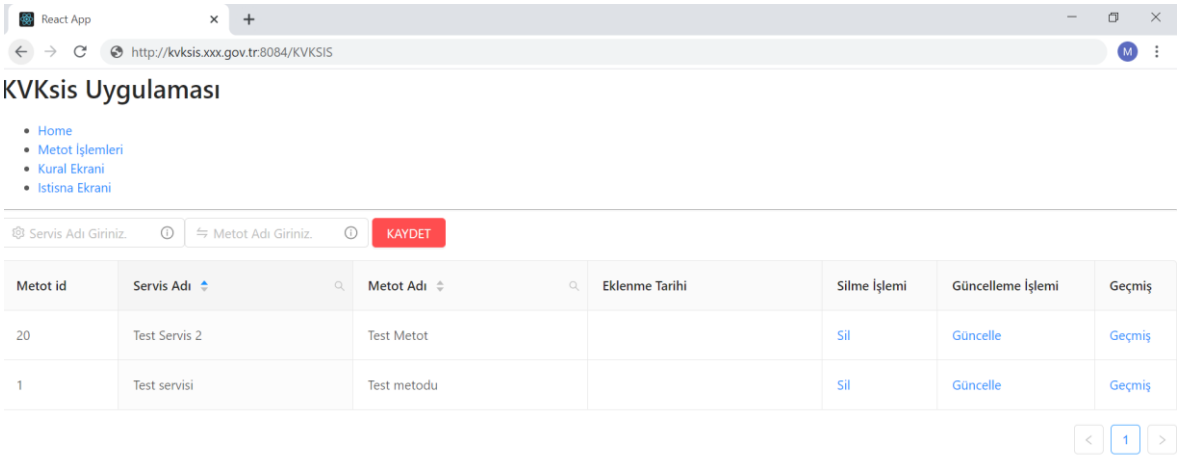
Şekil 6.4. Tomcat sunucusuna kurulum dosyalarını ekleme

Kurulum dosyaları eklendikten sonra Şekil 6.5'de gösterildiği gibi tomcat sunucusunun bulundu dizinde yer alan "bin" klasörünün içerisindeki "startup.bat" dosyasını çalıştırılarak tomcat ayağa kaldırılır.



Şekil 6.5. Tomcat sunucusunu çalıştırma

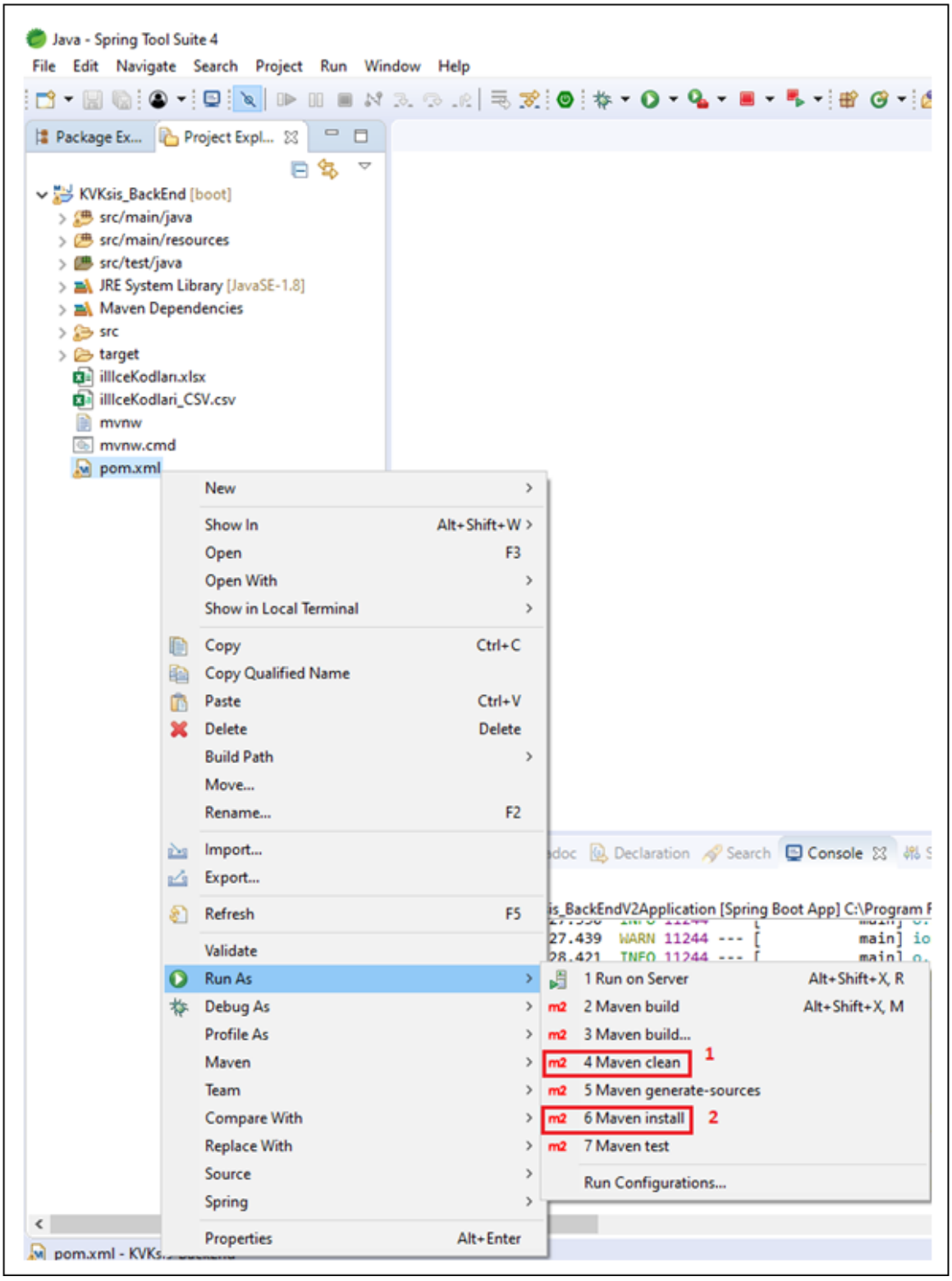
Son olarak kişisel verileri koruma sistemi için "kvksis.xxx.gov.tr" isminde bir DNS name oluşturuldu. Tomcat sunucusu üzerinden web servis altyapı sistemi uygulamasının kurulum işlemi tamamlanmaktadır. Uygulamayı test etmek için "http://kvksis.xxx.gov.tr:8084/KVKSIS" adresi herhangi bir tarayıcıda çalıştırıldığında Şekil 6.6'da görüldüğü gibi uygulama ekranına erişilir.



Şekil 6.6. KVKSIS uygulaması

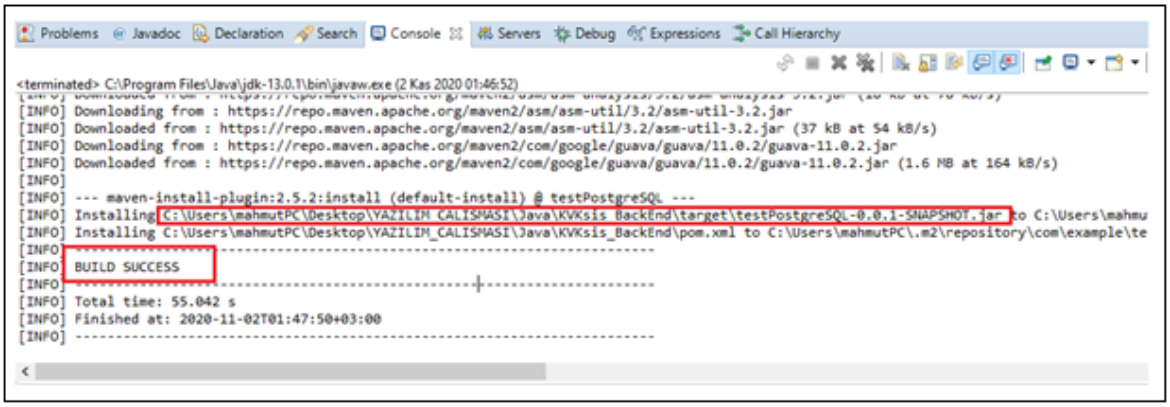
6.2. Server Kurulumu

Java spring boot ile server kısmı geliştirilen kişisel verileri koruma sisteminin kurulumu için “Spring Tool Suite 4” editöründe yer alan “pom.xml” dosyasını Şekil 6.7’de görüldü gibi “Run As” seçeneği seçilir sonra “maven clean” seçeneğini seçerek öncelikle projenin “clean” ardından “maven install” seçeneğinin seçilmesi gerekir bu şekilde projenin kurulum dosyaları oluşturulur.



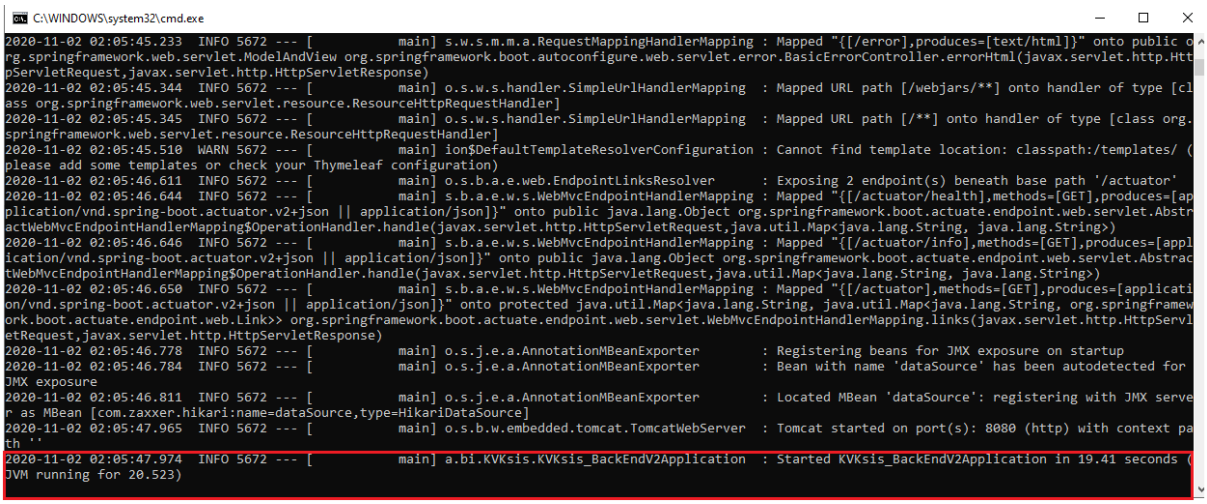
Şekil 6.7. Sunucu kurulum dosyası üretme

“Maven install” işlemini gerçekleştirdiğimiz de kurulum dosyaları Şekil 6.8’de gösterilen dizine oluşturulur.



Şekil 6.8. Server uygulaması kurum dosyası oluşturma

Şekil 6.8'de gösterilen dizine gidilir ve oluşturulan "jar" dosyasını çalıştırmak için "bat" uzantılı bir dosya hazırlanır. "bat" uzantılı dosya içerisine uygulamanın çalıştırılabilme için "java -Dfile.encoding=UTF-8 -jar testPostgreSQL-0.0.1-SNAPSHOT.jar" kodu eklenir ve çalıştırılır. Şekil 6.9'da gösterildiği gibi server uygulaması tomcat uygulama sunucu üzerinde ayağa kaldırılır.



Şekil 6.9. Server uygulamasının çalıştırılması

7. SONUÇ VE ÖNERİLER

Kurumlar vermiş oldukları hizmetlerin kalitesini ve hızını artırmak, vatandaşların verilen hizmete erişimini kolaylaştırmak amacıyla bilgi sistemleri üzerinden hizmet vermektedirler. Bu hizmetleri daha etkili sunabilmek için sahip oldukları verileri kanunlara uygun olarak işlemekte, depolamakta ve bir birleriyle paylaşmaktadırlar. Kurumlar arasında paylaşılan veriler içerisinde kişisel veriler de bulunmaktadır. Paylaşılan bu veriler genellikle web servisler aracılığı ile kurumlar arasında aktarımı sağlanmaktadır. Kurumların sahip oldukları yazılımları idame ettiren yazılım geliştiricilere içerisinde kişisel verileri barındıran web servislere erişim yetkisi verebilmektedirler.

Yazılım geliştiriciler ile içerisinde kişisel verilerin bulunduğu web servislere erişim yetkisi verilmesinin bazı sebepleri;

- Kurumların her zaman test web servisi açmaması,
- Sunulan test web servislerinde bütün olasılıkları kapsayacak veri kümesinin olmaması,
- Bazı kişilerin verilerinden kaynaklı hatalardan dolayı yazılım geliştiriciler o kişilerin gerçek verileri ile yazılımlarında test yapması gerekliliği.

Gerçek verilerin bulunduğu web servislerin yazılım geliştiriciler ile paylaşılması kişisel verileri zafiyete açık hale getirmektedir. Bu tez kapsamında bu zafiyetin önüne geçilebilmesi için bir uygulama geliştirilmiştir. Geliştirilen uygulama ile web servisten sorgulama yapan yazılım geliştirici ise web servisten gelen verilerden kişisel veri olanları belirli bir düzende değiştirilip yazılım geliştiriciye sunulması amaçlamaktadır. Böylece kişisel verilerin kötüye kullanılmasının önüne geçmek amacıyla mevcut uygulamalara ek bir güvenlik önlemi alınmış olacaktır. Ancak web servis sorgusunu yapan uygulama sunucu ise web servisten dönen verilerde herhangi bir değişiklik yapılmadan uygulama sunucularına veriler iletilmektedir.

Yazılım geliştiricilere kişisel veri barındıran web servislere erişim yetkisi verilmesi durumunda geliştirme ortamı veri tabanlarına gerçek veriler kaydedilmektedir. Geliştirme ortamı veri tabanları kişisel verileri barındıracağı için kişisel verilerin korunmasına yönelik

bu veri tabanları üzerinde çalışmaların yapılması gerekecektir. Bu durum kurumlara ekstra iş yükü getirmektedir.

Bu gibi sorunların önüne geçilebilmesi amacıyla bu tez kapsamında tasarlanan uygulama üç bölümden oluşmaktadır. Birinci bölüm web servis metotlarının cevap olarak döndüğü nesnenin bütün parametrelerini belirleyip veri tabanına kayıt işlemi yapan rest servis olarak hizmet veren bölümdür. Bu rest servis kendisine gönderilen cevap nesnesinin bütün elemanlarının belirlenip veri tabanına kayıt edilmektedir. Kayıt işlemi ikinci adım için bir ön hazırlıktır.

İkinci bölümde, uygulama ara yüzü kullanılarak web servis cevap nesnesi parametrelerinin kuralların girildiği bölümdür. Bu tanımlamalar sayesinde tasarlanan uygulama cevap nesnesi içerisindeki hangi alanların ne şekilde değiştirileceğine karar vermesi için gereklidir. Cevap nesnesi içerisindeki parametreleri anlamlı bir şekilde değiştirilmesi amaçlanmıştır. Örnek olarak kişi ismi içeren bir alana kişi ismi verilmesi, doğum tarihi içeren bir alana doğum tarihi verilmesi gerekmektedir. Aksi takdirde içeriğini değiştirip gönderdiğimiz cevap nesnesi üzerinde işlem yapan yazılım geliştirici yazılımını geliştirirken hatalar ile karşılaşacaktır. Bu hataların önüne geçmek için cevap nesnesi parametrelerine kural girilmesi ve bu kurallar neticesinde içeriğinin mantıklı bir şekilde değiştirilmesi gerekmektedir.

Üçüncü bölümde web servis cevap nesnesi tasarlanan uygulamanın ikinci bölümünde belirlenmiş olan kurallara göre cevap nesnesi içerisinde kişisel verilerin değerlerini değiştirerek kişisel verilerin anonim hale getirildiği bölümdür.

Tasarlanan bu uygulama ile her servis için gerekli tanımlamalar yapıldıktan sonra başka herhangi bir çalışma yapılmasına gerek kalmadan kişisel verilerin yazılım geliştiricilere karşı korunmasında ek bir önlem alınmış olacaktır.

Bu uygulama ile yazılım geliştiriciler kişilerin web servis sorgulamalarını gerçekleştirdiklerinde anonim hale getirilmiş verilere erişeceği için kişisel verilere erişmesi engellenmiş olacaktır. Ancak bu engelleme web servise erişmemesi değil anonim hale getirilen kişisel verileri içeren web servis sunularak yazılım geliştiricilerin hem yazılımı geliştirmeye devam edebilmesi hem de kişisel verilerin korunması sağlanmış olacaktır.

Yazılım geliştiricilerin geliştirdikleri yazılımın geliştirme ortamlarının veri tabanında anonim hale getirilmiş veriler olacağı için kurum personelleri bu veri tabanlarında kişisel verileri korumak için ekstra çalışma yapmalarına gerek kalmayacaktır. Böylece kurumlar personellerini daha verimli kullanabilecektir.

Bu uygulama her servise entegre edilebilmesi amacıyla esnek bir yapıda tasarlanmıştır. Cevap nesnesinin içeriği değiştirilirken yapısı korunmaktadır. Böylece web servis sorgulaması yapan yazılım geliştiriciler cevap nesnesini karşı kurumdan geldiği haliyle aldıkları için (yapısı korunmuş ancak içeriği anonim hale getirilmiş şekilde) ekstra bir çalışma yapmadan kullanabileceklerdir.

Gerekli izinler alındıktan sonra bazı kullanıcılara belli süreli canlı verileri görme yetkisi tanımlanabilmektedir. Ancak süre dolduğunda otomatik olarak yetki geri alınmakta böylece yazılım geliştiricilerin kişisel verilere sınırsız erişim sağlamasının önüne geçilmektedir. Bu uygulama geliştirilirken gerçek hayatta karşılaşılabilecek olası bütün durumlar göz önünde tutularak geliştirilmeye çalışıldığı için uygulamaya bu özellik konulmuştur.

Çizelge 7.1. Kişisel verilerin korunması sisteminden önce ve sonrasının karşılaştırması

KVKSIS Uygulamasından Önce	KVKSIS Uygulamasından Sonra
Web servislerden gelen kişisel veriler yazılım geliştiricilere karşı savunmasızdır.	KVKSIS uygulaması ile web servislerden gelen kişisel verilerin korunmasında ek bir güvenlik önlemi alınmış olacaktır.
Yazılımların geliştirme ortamı veri tabanlarında gerçek kişisel veriler olacaktır.	KVKSIS uygulaması yazılım geliştiricilere kişisel verileri değiştirip sunduğu için yazılımların geliştirme ortamı veri tabanlarında gerçek kişisel veriler olmayacaktır.
Veri tabanı ekibi yazılımların geliştirme ortamlarında yer alan verilerin korunması için çaba sarf etmesi gerekecektir.	Geliştirme ortamlarında gerçek kişisel veriler olmayacağı için ekstra bir çaba gerekmeyecektir. Kurumlar personellerini daha verimli kullanabileceklerdir.
Web servis sorgularında arada ekstra herhangi bir uygulama olmadığı için sorgu süresi nispeten kısadır.	Web servis sorgularında arada KVKSIS olduğu için sorgu süresi nispeten uzundur.
Kurumlardaki veri ihlalleri ve saçılmaları kolaydır.	Kurumlardaki veri ihlalleri ve saçılmalar zorlaşarak daha güvenli hale gelecektir.

KAYNAKLAR

1. YILDIZ, B. (2007). *Bilgi güvenliği ve e-devlet kapsamında kamu kurumlarında bilgi güvenliği yönetimi standartlarının uygulanması*. Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, Gebze.

2. İnternet: What is Information Security?, URL: http://www.demop.com/articles/information-security.html, Son Erişim Tarihi: 24.12.2020.

3. MOĞOL, Ş. H. (2016). *İmportance of ınformatıon securıty awareness*. Yüksek Lisans Tezi, Yıldırım Beyazıt Üniversitesi, Ankara.

4. KÖSEOĞLU, Ç. (2018). *Sosyal yardım ve sosyal hizmet alanındaki bilişim uygulamalarında iso/iec 27001 bilgi güvenliği yönetim sistemi uyumunun değerlendirilmesi*. Uzmanlık Tezi, Aile, Çalışma ve Sosyal Hizmetler Bakanlığı, ANKARA.

5. Macedo, F. and Da Silva, M. M. (2012). *Comparative study of information security risk assessment models*. Instituto Superior Técnico, UniversidadeTécnica de Lisboa, Lisboa, Portugal.

6. İnternet: ISO 31000 Risk Yönetim Sistemi, URL: https://www.vericert.com.tr/turkish/ISO_31000.html, Son Erişim Tarihi: 25.12.2020.

7. Baykara, M., Daş, R. ve Karadoğan, İ. (2013). Bilgi güvenliği sistemlerinde kullanılan araçların incelenmesi. *1st International Symposium on Digital Forensics and Security* (ISDFS'13).

8. Çek, E. (2017). *Kurumsal bilgi güvenliği yönetişimi ve bilgi güvenliği için insan faktörünün önemi*. Yüksek Lisans Tezi, İstanbul Bilgi Üniversitesi, İstanbul.

9. Johnson, M. E. and Goetz, E. (2007). Embedding information security into the organization. *IEEE Security & Privacy*, 5(3), 16-24.

10. İnternet: EKS Zafiyet Yönetimi, URL: https://www.barikat.com.tr/teknolojiler/eks-zafiyet-yonetimi, Son Erişim Tarihi: 25.12.2020.

11. Gelbstein, E. and Kamal, A. (2002). *Information Security*. November, 17, 18-19.

12. Vural, Y. ve SAĞIROĞLU, Ş. (2008). Kurumsal bilgi güvenliği ve standartlari üzerine bir inceleme. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 23(2).

13. Radovanović, D., Radojević, T., Lučić, D. and Šarac, M. (2010). IT audit in accordance with Cobit standard. *The 33rd International Convention MIPRO*, 1137-1141.

14. İnternet: ITIL & COBİT & ISO 27001, URL: https://mustafabayramblog.wordpress.com/2016/03/04/itil-cobit-iso-27001/, Son Erişim Tarihi: 02.01.2021.

15. Sallé, M. (2004). *IT Service Management and IT Governance: review, comparative analysis and their impact on utility computing*. Hewlett-Packard Company, 10.

16. Năstase, P., Năstase, F. and Ionescu, C. (2009). *Challenges generated by the implementation of the IT standards CobiT 4.1, ITIL v3 and ISO/IEC 27002 in enterprises*. Economic computation & economic cybernetics studies & research, 43(3), 9.

17. Rezakhani, A., Hajebi, A. and Mohammadi, N. (2011). Standardization of all information security management systems. *International Journal of Computer Applications*, 18(8), 5.

18. Volonino, L. and Robinson, S. R. (2003). *Principles and practice of information security*. Prentice.

19. McMillan, M. (2010). Logging and Auditing in a Healthcare Environment, *OCR/NIST HIPAA Security Rule Conference, 9.*

20. Boehmer, W. (2010). Performance, survivability and cost aspects of Business Continuity Processes according to BS 25999. *The International Journal on Advances in Security*, 2, 312-324.

21. İnternet: BS 25999 İş Sürekliliği Yönetimi Standardı, URL: https://www.cozumpark.com/bs-25999-is-surekliligi-yonetimi-standardi/amp/, Son Erişim Tarihi: 02.01.2021.

22. İnternet: Veri, URL: https://tr.wikipedia.org/wiki/Veri#cite_note-1, Son Erişim Tarihi: 27.02.2021.

23. Alkan, M., Menteş, T. ve İnceefe, M.A. (2020). Kişisel verileri koruma el kitabı. Ankara: Nobel Akademik Yayıncılık.

24. İnternet: Veri Türleri Nelerdir, URL: https://www.dilimiz.gen.tr/veri-turleri-nelerdir/, Son Erişim Tarihi: 27.02.2021.

25. İnternet: Difference between structured, semi-structured and unstructured data, URL: https://www.geeksforgeeks.org/difference-between-structured-semi-structured-and-unstructured-data/, Son Erişim Tarihi: 27.02.2021.

26. Başalp, N. (2004). *Kişisel verilerin korunması ve saklanması*. Ankara: Yetkin Yayınları.

27. Boz, A. (2014). *Kişisel Verilerin Korunması: Türkiye, ABD ve AB Örnekleri*. Yayınlanmamış Yüksek Lisans Tezi, TC Polis Akademisi Güvenlik Bilimleri Stratejileri ve Yönetim Anabilim Dalı, Ankara, 23-25.

28. Kaya, C. (2011). Avrupa birliği veri koruma direktifi ekseninde hassas (kişisel) veriler ve işlenmesi. *İstanbul Üniversitesi Hukuk Fakültesi Mecmuası*, 69(1-2), 318.

29. İnternet: OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, URL: http://www.oecd.org/digital/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflowsofpersonaldata.htm, Son Erişim Tarihi: 02.01.2021.

30. İnternet: 95/46/EC Kişisel Verilerin İşlenmesi ve Bu Tür Verilerin Serbest Dolaşımına Dair Bireylerin Korunması Direktifi, URL: https://kisiselveri.com/9546ec-turkce, Son Erişim Tarihi: 02.01.2021.

31. Gürsel, İ. (2016). *İşçinin kişisel verilerinin korunması hakkı*. Ankara: Adalet Yayınevi.

32. Personal Data: The Emergence of a New Asset Class, An Initiative of the World Economic Forum, January 2011 in Collaboration with Bain & Company, Inc.

33. Dülger, M. V. (2018). İnsan Hakları ve Temel Hak ve Özgürlükler Bağlamında Kişisel Verilerin Korunması. *İstanbul Medipol Üniversitesi Hukuk Fakültesi Dergisi*, 5(1), 73.

34. Lloyd, I. (2020). Information technology law. *Oxford University Press*.

35. Küzeci, E. (2010). *Kişisel verilerin korunması*. Ankara: Turhan Kitabevi.

36. DURSUN, Y. (2019). *6698 Sayılı kişisel verilerin korunması kanunu kapsamında işçinin korunması*. 69.

37. HAFIZOĞULLARI, Z. ve Muharrem, Ö. (2009). Özel hayata ve hayatın gizli alanına karşı suçlar. *Ankara Barosu Dergisi,* (4), 9-22.

38. Sevimli, A. (2008). İşçinin Özel Yaşam Hakkı Bağlamında İşçi-İşveren İlişkisi. *Sicil Dergisi*, 10, 53-79.

39. Kişisel Verilerin Korunması Kanunu, 29677 (2016). URL: https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5, Son Erişim Tarihi: 02.01.2021.

40. Kişisel Verilerin Korunması Kanunu Tasarısı (1/541) ve Adalet Komisyonu Raporu (2016). 4,5,6.

41. Ribeiro, S. L., & Nakamura, E. T. (2019). Privacy protection with pseudonymization and anonymization in a health iot system: Results from ocariot. *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), 904-908.*

42. İnternet: Pseudonymization, URL: https://en.wikipedia.org/wiki/Pseudonymization, Son Erişim Tarihi: 02.01.2021.

43. İnternet: Kişisel verilerin silinmesi, yok edilmesi veya anonim hale getirilmesi rehberi, 28,29 (2017). URL: https://www.kvkk.gov.tr/yayinlar/K%C4%B0%C5%9E%C4%B0SEL%20VER%C4%B0LER%C4%B0N%20S%C4%B0L%C4%B0NMES%C4%B0,%20YOK%20ED%C4%B0LMES%C4%B0%20VEYA%20ANON%C4%B0M%20HALE%20GET%C4%B0R%C4%B0LMES%C4%B0%20REHBER%C4%B0.pdf, Son Erişim Tarihi: 02.01.2021.

44. İnternet: l-diversity, URL: https://en.wikipedia.org/wiki/L-diversity, Son Erişim Tarihi: 02.01.2021.

45. Saatci, C. and Gunal, E. S. (2019). Preserving Privacy in Personal Data Processing. *2019 1st International Informatics and Software Engineering Conference (UBMYK),* 2-3.

46. İnternet: Tokenization vs Encryption, URL: https://www.mcafee.com/enterprise/en-us/security-awareness/cloud/tokenization-vs-encryption.html, Son Erişim Tarihi: 02.01.2021.

47. Kişisel Nitelikteki Verilerin Otomatik İşleme Tabi Tutulması Karşısında Şahısların Korunmasına Dair Sözleşme (2016).

48. Türk Medeni Kanunu, 24607 (2001). URL: https://www.mevzuat.gov.tr/MevzuatMetin/1.5.4721.pdf, Son Erişim Tarihi: 02.01.2021.

49. Ketizmen, M. (2008). *Türk ceza hukukunda bilişim suçları*. Ankara: Adalet Yayınevi.

50. SERT, Ş. (2018). *Kişisel verilerin 5237 sayılı Türk Ceza Kanunu kapsamında korunması.* Yüksek Lisans Tezi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü, Erzurum, 73.

51. Solove, D. J. (2008). The new vulnerability: Data security and personal information. 63, 111-136.

52. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. 2016.

53. Akdağ, H. (2015). '*Türk Ceza Hukukunda Kişisel Verilerin Korunması İlkeleri*. Prof. Dr. Nevzat Toroslu'ya Armağan, 1(459), 37.

54. Şimşek, O. (2008). *Anayasa Hukukunda kişisel verilerin korunması*. İstanbul: Beta Yayıncılık.

55. İş Kanunu, 25134 (2003). URL: https://www.mevzuat.gov.tr/MevzuatMetin/1.5.4857.pdf, Son Erişim Tarihi: 02.01.2021.

56. Yılmaz, E. (1992). *Hukuk sözlüğü* (Vol. 2). Ankara: Yetkin Yayınları.

57. Türk Ceza Kanunu, 25611 (2004). URL: https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=5237&MevzuatTur=1&MevzuatTertip=5, Son Erişim Tarihi: 02.01.2021.

58. İnternet: Software, URL: https://en.wikipedia.org/wiki/Software, Son Erişim Tarihi: 02.01.2021.

59. Kurt, K. (2005). *Java teknolojisi kullanarak internet tabanlı öğrenci kayıt sistemi*. Yüksek Lisans Tezi, Muğla Üniversitesi, Fen Bilimleri Enstitüsü, Muğla, 77.

60. İnternet: What is Database Architecture?, URL: https://www.guru99.com/dbms-architecture.html, Son Erişim Tarihi: 02.01.2021.

61. İnternet: Two-tier Vs Three-tier Architecture, URL: https://medium.com/@gacheruevans0/2-tier-vs-3-tier-architecture-26db56fe7e9c, Son Erişim Tarihi: 02.01.2021.

62. İnternet: What is a 3-Tier Architecture?, URL: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/, Son Erişim Tarihi: 02.01.2021.

63. İnternet: Multitier architecture, URL: https://en.wikipedia.org/wiki/Multitier_architecture, Son Erişim Tarihi: 02.01.2021.

64. İnternet: The J2EE Architect's Handbook, URL: http://www.kurumsaljava.com/2009/01/25/the-j2ee-architects-handbook/, Son Erişim Tarihi: 02.01.2021.

65. İnternet: Java (programming language), URL: https://en.wikipedia.org/wiki/Java_(programming_language), Son Erişim Tarihi: 02.01.2021.

66. İnternet: 10 Most Popular Programming Languages on GitHub, URL: https://insights.dice.com/2020/12/03/10-most-popular-programming-languages-on-github/, Son Erişim Tarihi: 02.01.2021.

67. Elibol, M. Ç. (2008). *e-Hastane sistemlerinin incelenmesi ve Java teknolojileri ile e-hastane uygulaması geliştirilmesi*. Yüksek Lisans Tezi. Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

68. KARA, Ş. E. (2014). *C# ve java nesne yönelimli programlama dillerinde collectıon framework'lerin karşılaştırmalı performans analizleri*. Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara.

69. İnternet: Java JDK, JRE and JVM, URL: https://www.programiz.com/java-programming/jvm-jre-jdk, Son Erişim Tarihi: 02.01.2021.

70. İnternet: JRE (Java Runtime Environment), URL: https://www.ibm.com/cloud/learn/jre, Son Erişim Tarihi: 02.01.2021.

71. İnternet: Java persistence API - Tutorial, URL: https://www.vogella.com/tutorials/JavaPersistenceAPI/article.html, Son Erişim Tarihi: 02.01.2021.

72. Aggarwal, S. (2018). Modern web-development using reactjs. *International Journal of Recent Research Aspects*, 5(1), 2349-7688.

73. İnternet: 2020 Developer Survey, URL: https://insights.stackoverflow.com/survey/2020, Son Erişim Tarihi: 02.01.2021.

74. Ramos, M., Valente, M. T. and Terra, R. (2017). AngularJS performance: A survey study. *IEEE Software*, 35(2), 72-79.

75. Oya, M., & Kashiwakura, A. (2016). JavaScript language extension for non-professional programmers: Sharable own variables. *2016 IEEE 5th Global Conference on Consumer Electronics.*

76. DURAL, D. (2012). *Servis yönelimli mimarilerde güvenlik çözümleri için web servis temelli bir alt yapı modeli*. Yüksek Lisans Tezi, Sakarya Üniversitesi, Sakarya.

77. İnternet: Simple Object Access Protocol(SOAP) Nedir?, URL: https://www.muratoner.net/internet/soap-simple-object-access-protocol-nedir, Son Erişim Tarihi: 02.01.2021.

78. Maboçoğlu, A. (2010). *Restful web servisleri ile ontoloji sorgulama*. Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

79. Çiçek, A. N. (2009). *Restful web servisleri ile e-sağlık sistemleri gerçekleştirimi*. Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

80. İnternet: SOAP vs REST. What's the Difference?, URL: https://smartbear.com/blog/test-and-monitor/soap-vs-rest-whats-the-difference/, Son Erişim Tarihi: 02.01.2021.

81. İnternet: SOAP Vs. REST: Difference between Web API Services, URL: https://www.guru99.com/comparison-between-web-services.html, Son Erişim Tarihi: 02.01.2021.

82. İnternet: SQL Server ve İlişkiselVeritabanı Modeli, URL: https://www.academia.edu/9330264/SQL_Server_ve_%C4%B0li%C5%9Fkisel_Veritaban%C4%B1_Modeli, Son Erişim Tarihi: 02.01.2021.

83. İnternet: ACID Nedir?, URL: https://medium.com/cloud-and-servers/acid-nedir-53f729f2bbb2, Son Erişim Tarihi: 02.01.2021.

84. İnternet: What is PostgreSQL?, URL: https://www.postgresql.org/about/, Son Erişim Tarihi: 02.01.2021.

85. ÖZTÜRK, S. ve ATMACA, H. E. (2017). İlişkisel ve ilişkisel olmayan (NoSQL) veri tabanı sistemleri mimari performansının yönetim bilişim sistemleri kapsamında incelenmesi. *Bilişim Teknolojileri Dergisi*, 10(2), 199-209.

86. GÖKŞEN, Y. ve Hakan, A. (2015). *Veri büyüklüklerinin veritabanı yönetim sistemlerinde meydana getirdiği değişim*: NOSQL. INTERNATIONAL JOURNAL OF INFORMATICS TECHNOLOGIES, 8(3), 125.

87. Bajer, M. (2017). Building an IoT data hub with Elasticsearch, Logstash and Kibana. *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 63-68.*

88. İnternet: ELK Stack Tutorial: Learn Elasticsearch, Logstash, and Kibana, URL: https://www.guru99.com/elk-stack-tutorial.html, Son Erişim Tarihi: 02.01.2021.

89. İnternet: Elasticsearch, URL: https://en.wikipedia.org/wiki/Elasticsearch, Son Erişim Tarihi: 02.01.2021.

90. İnternet: Advantages and disadvantage of Elasticsearch, URL: https://www.javatpoint.com/advantages-and-disadvantages-of-elasticsearch, Son Erişim Tarihi: 02.01.2021.

91. İnternet: Logstash, URL: https://www.elastic.co/logstash, Son Erişim Tarihi: 02.01.2021.

92. Shatnawi, A. S. (2017). *Enhanced Version Control for Unconventional Applications*.

93. İnternet: Bitbucket: What is Bitbucket?, URL: https://confluence.atlassian.com/confeval/development-tools-evaluator-resources/bitbucket/bitbucket-what-is-bitbucket, Son Erişim Tarihi: 02.01.2021.

94. BAKTIR, O. (2006). *Eclipse Platformu İçin Yüksek Düzeyli Mimari Yazılım Modelleme Aracı*. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara .

95. İnternet: Visual Studio Code, URL: https://code.visualstudio.com/docs, Son Erişim Tarihi: 02.01.2021.

96. İnternet: My Top VS Code Tips and Features, URL: https://scotch.io/bar-talk/my-top-8-visual-studio-code-tips-and-features, Son Erişim Tarihi: 02.01.2021.

97. İnternet: Postman Client Makes RESTful API Exploration a Breeze, URL: https://www.programmableweb.com/news/review-postman-client-makes-restful-api-exploration-breeze/brief/2014/01/27, Son Erişim Tarihi: 02.01.2021.

98. İnternet: Pgadmin, URL: https://www.pgadmin.org/, Son Erişim Tarihi: 02.01.2021.

99. İnternet: Why pgAdmin 4?, URL: https://wiki.postgresql.org/images/7/71/Why_pgAdmin_4%3F.pdf, Son Erişim Tarihi: 02.01.2021.

GAZİ GELECEKTİR...