Evrak Otomasyon Sistemlerinde Güvenlik Uygulamaları

Ömer DEPERLİOĞLU[1], Ayşen ÖLMEZ[2]

[1]Biyomedikal Mühendisliği Bölümü, Afyon Kocatepe Üniversitesi, Afyonkarahisar, Türkiye
[2]Afyon Meslek Yüksekokulu, Afyon Kocatepe Üniversitesi, Afyonkarahisar, Türkiye
odeper@aku.edu.tr, aolmez@aku.edu.tr

Özet- Gerek kişisel gerekse kurumsal faaliyetlerin sürdürülmesi için oluşturulan kağıt evraklar, teknolojik gelişmelerin etkisi ile yerini hızla elektronik evraklara bırakmaktadır. Gün geçtikçe sayısında artış gözlemlenen elektronik evrakların klasik yöntemlerle yönetimleri zorlaşmış, daha etkili, güvenilir ve elektronik ortama uygun çözümler geliştirilmiştir. Evrak yönetim sistemleri olarak tanımlanan bu sistemlerde, temel amaç evraka ait tüm işlemlerin sayısal olarak takibini sağlamaktır. Evrakın yaşam döngüsünü oluşturan evrak oluşturma, düzenleme ve saklama süreçlerinin gerçekleşmesinde önemli olan bir nokta da, sistemlerde olması gereken güvenlik unsurlarıdır. Programlama dili, kullanılacak veritabanı seçimi ve veritabanının tasarlanması gibi işlemlerden de etkilenen güvenlik fonksiyonu yazılım ile desteklenerek daha da güçlendirilebilir. Kullanıcıların sisteme erişimleri esnasında kullanıcı adı ve parola kontrolünden geçirilmeleri, yetkileri doğrultusunda dokümanları görebilmeleri ve düzenleyebilmeleri gibi güvenlik önlemlerinin sistemlerde bulunması zorunludur. Bu işlemler dokümanların orijinalliklerini kaybetmeden sistemde saklanmalarını sağlar. Bu çalışmada Afyon Kocatepe Üniversitesi evrak takibi için oluşturulan otomasyon sisteminin güvenlik yönü ele alınmıştır. Benzer uygulamalara ışık tutması amacıyla sistemde var olan güvenlik fonksiyonlarından bahsedilmiştir. Değinilen fonksiyonlardan bazıları sadece evrak takip uygulamalarında değil, birçok farklı uygulamada da uygulanabilir niteliktedir.

Anahtar Kelimeler- Evrak, Evrak Yönetimi, Yazılım Güvenliği, Yazılım Güvenliği Standartları

Security Applications in Document Management Systems

Abstract- With the effects of technological developments, paper documents, which are formed to maintain both personal and institutive activities, have given place to electronic documents rapidly. Management of electronic documents with classical methods has become difficult and more effective and dependable solutions, which are also suitable to electronic environment, have been developed. In these systems, which are defined as electronic document management systems, the main purpose is to ensure tracking all operations of the document with digital approaches. One of the most important points in document creating, editing and storing is to provide security elements, which must be included in the related systems. The security function, which can also be affected by programming language, database that will be used and designing the database, can be supported with software to become more powerful. Some security precautions like controlling user name and password during the login process and enabling users to see and edit documents according to their authorizations must be included in the systems. These operations allow storing the documents without loosing their authenticity. In this study, security aspects of the automation system, which was developed for Afyon Kocatepe University, are examined. In order to shed light on similar applications, security functions included in the system are explained. Some of the explained functions can be used in not only in document tracing applications but also in many different applications.

Keywords- Document, Document Management, Software Security, Software Security Standards

1.GİRİŞ

Bilginin her geçen gün artması sonucu günlük hayatımızda ve işyerlerinde oluşturulan belge sayısı da çoğalmaktadır. Belge sayısı arttıkça yönetilmeleri ve arşivlenmeleri daha zor hale gelmektedir. Oluşturulan belgelerin yönetiminde iki yaklaşım söz konusudur. Belgelerin kağıt üzerinde takibi ve geliştirilen bir yazılım sistemi ile takibi. Geleneksel yani kağıt üzerinde yapılan evrak takibinin güvenlik konusunda zayıf yönleri mevcuttur. Örneğin, kağıt olarak saklanan evrak kaybedilebilir ya da çoğaltılabilir. Ayrıca artan sayı sonucunda arşivleme işlemlerinde zorluk artmakta ve arşiv için ayrılan fiziki alan da her geçen gün yetersiz kalmaktadır. Her evrakın çıktısının alınması da belirli bir maliyete yol açmaktadır. Bu nedenle geleneksel yaklaşımın buna benzer eksikliklerini ortadan kaldırmak

için elektronik evrak yönetim sistemine ihtiyaç duyulmaktadır [1].

İlerleyen teknolojinin etkisiyle, kullandığımız belgelerin çoğunluğu elektronik ortamda oluşturulmaktadır. Bu belgelerin geçirdiği süreçlerin organizasyonlardaki takibini yapmak için birçok elektronik doküman yönetim sistemi geliştirilmiştir. Elektronik doküman yönetim sistemleri gerek kamuda gerekse özel şirketlerde çok fazla ilgi görmüştür. Oluşturulan bu sistemlerin birçoğu belgelerin yönetilmesini asıl hedef olarak belirlemiş, güvenli belge yönetiminde yetersizlikleri olduğu ortaya çıkmıştır. Güvenli bir elektronik doküman yönetimi sistemi kullanan her kullanıcı için aynı düzeyde güvenlik sağlamalıdır. Sisteme eklenen dokümana yetkili kişiler ulaşabilmeli ve yapılan değişiklikler anında takip edilebilmelidir. Kayıtlı dokümanlar merkezi bir alanda saklanmalı ve kaybedilme riskleri ortadan kalkmalıdır. Kullanıcının yetkisi yoksa kaydı görememeli ya da değiştirme yetkisi olmadan değişiklik yapamamalıdır [2].
Bu makalede ikinci bölümde, yazılım güvenliği ele alınmıştır. Üçüncü bölümde güvenlik konusu göz önüne alınarak geliştirilmiş Evrak Doküman Yönetim Sistemi'nin alt yapısı ile ilgili bilgi verilmiştir. Dördüncü bölümde ise sistemde var olan güvenlik fonksiyonları örneklerle açıklanmıştır.

2.YAZILIM GÜVENLİĞİ

Özellikle kurumlarda bir çok işin bilgisayar kullanılarak yapılması yazılım çeşitliliğine yol açmıştır. Evrak takibinin yanı sıra, muhasebe işlemleri, bordro hazırlama gibi kurum için hayati öneme sahip işlemlerin de yazılıma ve bilgisayara aktarılması, güvenliği kaçınılmaz bir ihtiyaç haline getirmiştir.

Bilişim tehditlerinin artması sonucu oluşturulan uygulamalarda güvenlik konusunun ortaya çıkması, yazılım güvenliği daha doğrusu "güvenilir bilişim" diye nitelendirilen kavrama önem kazandırmıştır. Yazılım güvenliği; bir uygulamanın gizliliğini, bütünlüğünü ve erişilebilirliğini sağlayan bir güvenlik mekanizmaları topluluğu olarak tanımlanabilir [3]. Trusted Computing Group tarafından belirlenmiş olan güvenilir bilişim; gizlilik, bütünlük, erişilebilirlik ve kurtarılabilirlik gibi 4 ana temel üzerinde durmaktadır. Gizlilik; bir verinin sadece yetkisi olan kişi tarafından erişilebilmesidir. Yetkilendirme, şifreleme, mühürlenmiş alanda çalışma gizlilik ilkesi kapsamındadır. Bütünlük; verinin izinsiz değiştirilememe ve alıcıya eksiksiz iletilme özelliğidir. Erişilebilirlik; istenilen verinin istenildiği zaman erişilebilir olmasını ifade eder. Kurtarılabilirlik ise herhangi bir veri kaybında söz konusu verilerin kurtarılabilmesi anlamına gelmektedir [1]. Yedekleme işlemi bu adım için en gerekli unsurdur. Birçok kurum ağ güvenliği için büyük çabalar harcamakta, ancak bunun yanında en hassas bilgilerimize ulaşmayı sağlayacak şifreleri içeren uygulama güvenliğini atlamaktadırlar [4].
Yazılım güvenliği ile kastedilen sadece yazılım kod denetimi değildir. Güvenlik unsurlarına temel yazılım geliştirme aşamalarında yapılması gereken tüm adımlar dahildir. Bu adımlar yazılım için gerekli planlama, analiz, tasarım, kodlama, test, kullanım ve iyileştirme süreçlerini içermektedir. Normalde, bir sistem kullanıma açıldığında güvenli olmalıdır. Bunu söylemek kolay ancak başarmak zordur. Bir yazılımda en önemli güvenlik işlemlerinden biri şifreleme kullanılmasıdır. Şifreleme başlı başına karmaşık bir yapıdadır. Her yazılımcı bu karmaşık yapı hakkında yeterli bilgiye sahip değildir, bu nedenle şifreleme işleminin şifrelemede uzman bir kişi tarafından yapılması gerekir. Veritabanında bulunan kullanıcı adı, şifre, kişisel bilgi, belge gibi birincil öneme sahip verilerin şifreleme algoritmaları ile tutulması gibi işlemler şifrelemeye örnek verilebilir. Şifreleme dışında uygulamanın test aşaması güvenlik için önemli bir aşamadır, ancak yeterli değildir [5]. Yazılımdaki açıkların testinde uygulama sunucuya atılmadan kullanıcı işlemlerinin sonuçları kaydedilir. Bu testler için kullanılan ve internetten erişilebilen ücretsiz yazılımlar mevcuttur. Bu yazılımlar test esnasında kişinin bir hacker gibi sistemin açıklarını tespit edebilmesini sağlar ve sonuçta sistem açıklarını listeleyen bir rapor sunar. Bu tür yazılımlara örnek olarak HP Uygulama Güvenliği Merkezi (HP Application Security Center) verilebilir. Yazılım, Owasp ilk 10 sıralamasında bulunan güvenlik açıklarını göz önünde bulundurarak sisteminizin açıklarını tespit etmenize yardımcı olur.

Yazılım güvenliği; ISO 27001, ISO 27002, ISO 15408, OWASP (Open Web Application Security Projects), OSSTMM (Open Source Security Testing Methodology Manual), COBIT (Control Objective fro Information and Related Techniques), WASC (Web Application Security Consortuim) gibi uluslararası standartlarca tanımlanmıştır. Bu standartlardan bazılarına değinmek istersek:

ISO 27001: Uluslararası ISO 27001 standardı; Bilgi Güvenliği Yönetim Sisteminin (Information Security Management System - ISMS) geliştirilmesi, uygulanması, bakımı için bir model tasarlanması amacıyla hazırlanmıştır. Kurumda uygulanacak ISMS kurumun ihtiyaçlarına bağlıdır ve farklılık gösterebilir. Kurumun çalıştığı alan ve büyüklüğü de ISMS'i etkiler. Bu standart Planla - Yap - Kontrol Et - Uygula(Plan-Do-Check-Act - PDCA) modelini benimsemektedir.

ISO 27002: ISO 27002 standardına göre veri çeşitli denetimler kullanılarak korunabilir. Teknik ve fonksiyonel olan bu denetimlere ahlaki ve mantıksal denetimler de eklenebilir. Kurumlar verilerini korumak için bazı denetim mekanizmaları geliştirmeli ve uygulamaya koymalıdır. Bu denetimlerin kayıt edilecek verinin geçerliliğini, işlem süreçlerini ve çıktı verisi gibi tüm süreci kontrol etmesi gerekir. Bunlar aşağıdaki gibi sıralanabilir:

- Çift girdi veya diğer girdi denetimleri, veri girişinde bazı alanlarda karakter sınırlamaları
- Aralık değerleri
- Veri alanında geçersiz karakterler

- Eksik veya tamamlanamamış veri
- Üst ya da alt sınırları aşan veri
- İzinsiz veya tutarsız veri
- Doğrulama hatalarına yanıt için prosedürler
- Giriş verisinin inandırıcılığını test için gerekli prosedürler

COBIT: Cobit; Bilgi Teknolojileri (Information Technology-IT) denetiminde kullanılan global bir standarttır. En yaygın bilinen güvenlik modelleri baz alınarak geliştirilmiştir. Doğruluk, güvenlik, bütünlük gibi geleneksel güvenlik kavramlarına etkinlik, etkililik, güvenilirlik ve uyumluluk gibi kavramları entegre ederek tanımlanmış bir modeldir. Planlama ve düzenleme (Planning-Organization/PO), edinme ve uygulama (Acqusition-Implementation/AI), teslim ve destek (delivery- support/DS) ve izleme (Monitoring) olarak 4 ana başlık altında 34 IT adımı içerir.

OWASP: Owasp'ın başlıca önemsediği durum; "Yazılım, tüm veri girişi biçimlerine karşı, özellikle kullanıcı girişlerinde alt yapısı, veritabanı sistemi ile tamamen güçlü olmalıdır." sözü ile ifade edilmiştir [3]. OWASP'ın tanımladığı güvenlik açıklarından ilk 10 sıralamasında aşağıdaki maddeler yer almaktadır.

1. **Kontrol Edilmemiş Girdi (Unvalidated Input):** İlk madde; Web kullanıcısından alınan ve uygulamada kullanılmadan önce kontrol edilmeyen veriyi içerir. İyi bir tasarım için önemli olan değer "Girdi Kontrol Yüzdesi (PercentValidatedInput)" dir. Bu değer; T değeri yani uygulamada bulunan giriş formlarının veya ara yüzlerin (Html formları, post metodu, get metodu sayıları) sayıları ile V yani girdi kontrolü mekanizmasını kullanan ara yüzlerin sayısı kullanılarak hesaplanır. V/T oranı uygulamanın sistem açığı hakkında bilgi verir. Oranın yüksek olması sistem açığının az olduğunu ifade eder.

2. **Kırık Erişim Kontrolü (Broken Access Control):** Sistemde yetkilendirilmiş kullanıcıların neler yapabileceği kesin sınırlar ile belirlenmediğinde ortaya çıkar. Saldırganlar; diğer kullanıcıların hesaplarına erişmek, özel belgeleri görüntülemek veya yetkiye bağlanmamış fonksiyonları kullanmak için sistemin bu zayıflığından yararlanabilirler.

3. **Kırık Yetkilendirme ve Oturum Yönetimi (Broken Authentication and Session Management):** Uygulama; kullanıcı hesabının önemli verilerini ve oturum bilgilerini tamamen koruyamadığında ortaya çıkar. Saldırganlar şifreler, oturum çerezleri veya diğer bilgileri kullanarak yetkilendirme sınırlarını aşabilir ve başka kullanıcıların kimliklerini kullanabilirler. Bu tür sistem açığına örnek olarak; 90 günden fazla kullanılmamış hesapların kullanımlarının sona ermemesi verilebilir. Bu tür hesaplar sistem için açık bir risk teşkil eder ve illegal erişimlere izin verir.

4. **Çapraz Site Betikleri (Cross Site Scripting - XSS):** Web uygulamasının kullanıcı bilgisayarında bir saldırı aracı olarak kullanılması durumudur. Başarılı bir saldırı kullanıcının oturum bilgilerini ortaya çıkarır, yerel bilgisayara saldırır.

5. **Tampon Taşması (Buffer Overflow):** Bir işlemin kontrolünü ele almak üzere uygulama bileşenlerinin ezilmesi ve böylelikle girdi kontrolünün başarısız olmasıdır.

6. **Sokuşturma Açıkları (Injection Flaws):** Uygulamada parametreler dışarıdan verilmemelidir. Aksi takdirde saldırgan, bu parametrelere çeşitli komutlar (Sql komutları gibi) gömebilir ve uygulamada bu komutlar çalıştırılır.

7. **Uygunsuz Hata Yönetimi (Improper Error Handling):** Uygulama kaynak kodunun, herhangi bir işlem esnasında hataları uygun şekilde kontrol etmediği ve ele almadığı durumlardır. Eğer saldırgan böyle bir hata ekranı ile karşılaşırsa sistem bilgilerini ele geçirebilir ve güvenlik mekanizmasının iptal olmasına veya sunucunun çökmesine neden olabilir.

8. **Güvensiz Depolama (Insecure Storage):** Web uygulamalarında veriyi ve kişi bilgilerini korumak için şifreleme fonksiyonlarının hangi sıklıkla kullanıldığıdır.

9. **Uygulama Servis Reddi (Application Denial of Service):** Saldırganlar uygulama kaynaklarını yok ederek yasal kullanıcıların erişimini ve uygulamayı kullanmalarını engelleyebilirler. Ayrıca kullanıcı hesaplarını kilitleyebilir ve tüm uygulamanın iptal olmasına neden olabilirler.

10. **Güvensiz Konfigürasyon Yönetimi (Insecure Configuration Management):** Son madde olan güvensiz konfigürasyon yönetimi, uygulamanın ne kadar güçlü bir sunucu ayarına bağlı olduğuna işaret eder. Sunucular güvenliği etkileyen bir çok konfigürasyon seçeneğine sahiptir ve varsayılan ayarlardan hiç biri güvenli değildir [6].

3. EVRAK OTOMASYON SİSTEMİ ALT YAPISI

Evrak Otomasyon Sistemi geliştirme aşamasında Java dili kullanılmıştır. Java programlama dili, uygulama geliştirmede taşınabilir ve güvenli bir ortam sunar [7]. Java Dili açık kodlu, nesneye yönelik, platform bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilen yorumlayıcı (interpreted) bir dildir [8]. En belirgin özelliklerinden biri platform bağımsız olmasıdır. Microsoft XP, Vista, Macintosh, Linux gibi birçok işletim sistemi üzerinde çalışabilirler. Ayrıca taşınabilir olması da istenilen ortama kolaylıkla aktarılabilmesini ve çalıştırılabilmesini sağlar [9].

Java Dili nesne yönelimli dillerden biridir ve bu özelliği en etkin kullanan programlama dili olarak sayılabilir.

Nesne yönelimli geliştirme özelliği, programlama esnasında gerçek sistem ile uygulama arasındaki uyuşmazlığı azaltmayı hedefler [10]. Veri ve veri üzerinde yapılacak işlem tek bir nesne olarak görülür. Günlük hayattaki varlıkların da nesne olarak tanımlanabilmesi, program yapısının daha anlaşılabilir olmasını sağlar [11].

Program geliştirme ortamı olarak Oracle Jdeveloper 10g kullanılmıştır. Oracle Jdeveloper 10g Java , Sql, Xml tabanlı uygulamalar ve web servisleri oluşturmak için kullanılan bütünleşik geliştirme ortamıdır(Integrated Development Environment-IDE). Uygulama geliştirme işleminin modelleme, kodlama, derleme, test etme gibi bütün adımlarını destekler. Bütün bu işlemler tek bir ortam kullanılarak yapılabilir [12].

Evrak Otomasyon sistemi geliştirme aşamasında veritabanı olarak Access 2007 kullanılmıştır. Odbc bağlantısı ile veritabanına bağlantı sağlanmıştır. Yazılım geliştirme süreci %80 tamamlandıktan sonra veritabanı, SQL Server 2008 veritabanı yönetim sistemine aktarılmıştır. SQL Server 2008, Microsoft tarafından geliştirilen ve pazarlanan ilişkisel veritabanı yönetim sistemidir. Kullanımı kolay bir sistemdir. Taşınabilir dizüstü bilgisayarlarda, tek işlemcili veya simetrik çok işlemcili sistemlerde çalıştırılabilir [13].

Sistemde veritabanı bağlantıları tek bir classta tanımlanmıştır. Bağlantı açma ve kapatma işlemleri bu class üzerinden yapılmaktadır. Böylece her sayfada tekrar tekrar bağlantı kodları yazılmamış, kullanılmak istenen bağlantının aktif hale getirilmesiyle veritabanı erişimi sağlanmıştır.

Access ve SQL Server bağlantılarının yanı sıra, Oracle veritabanı bağlantısı da gerçekleştirilmiş ve test edilmiştir. Yaygın kullanılan ilişkisel veritabanı yönetim sistemlerinden biri olan Oracle için Oracle reklamcılarının *"İnternet Oracle üzerinde Çalışıyor (The Internet Runs on Oracle)"* şeklindeki iddialı sözünün sadece bir iddia ile kalmadığı gözlenmektedir [14]. 1979 yılında geliştirilmeye başlanan Oracle'ın en son sürümü Oracle 11g'dir. Uygulamada Oracle veritabanına bağlantı sağlamak için Jdbc bağlantısı kullanılmıştır. Jdbc Java'da kullanılan ve ilişkisel veritabanlarına bağlantı kurmayı sağlayan standart bir Uygulama Programlama Arayüzü'dür (*Application Programming Interface (API)*). Bu API Sun Microsystems tarafından geliştirilmiştir. Oracle, Sybase gibi bir çok ilişkisel veritabanı ile bağlantı kurulmasını sağlar. Bağlantı tanımlamak için java.sql ve javax.sql kütüphanelerinin uygulamaya dahil edilmesi gerekir [15].

Sistemin çalıştırılması için açık kaynak kodlu bir uygulama olan Apache Tomcat kullanılmıştır. Tomcat programı herhangi bir kurulum işlemi gerektirmeden makineden makineye taşınarak kullanılabilir. Sadece bazı ayarların yapılmasıyla kolay bir şekilde kullanıma hazır hale gelmektedir. Güvenliği de içeren bu ayarlardan biri uygulamanın açık olduğu portun belirlenmesidir. Tomcat konfigürasyon ayarlarından yapılan bu değişiklik ile uygulamaya sadece tek bir porttan ulaşılması sağlanmıştır. Sunucuda dışarıya açık olmayan bir port uygulama için kullanılabilir. Böylece uygulamanın sadece kurum içinde kullanılması Tomcat ayarları ile de desteklenmiş olacaktır.

4. EVRAK OTOMASYON SİSTEMİ'NDE GÜVENLİK

Evrak Otomasyon sisteminin Afyon Kocatepe Üniversitesi'nde kullanılması hedeflendiği için güvenlik konusu üzerinde önemle durulmuştur. Sisteme Login sayfasından giriş yapılmaktadır (Bkz. Şekil 1).

Şekil 1. Evrak Otomasyon Sistemi Giriş Ekranı

Kullanıcı adı ile şifre kontrolü sonucunda her kullanıcı için oturum (session) açılmakta ve ekranda Gelen Evrak ya da Giden Evrak seçim sayfası görüntülenmektedir. Bu safhada seçilen sayfaya yönlendirilirken kullanıcının yetkisine göre güvenlik amaçlı bazı sınırlamalar göz önüne alınmaktadır. Bunlardan bazıları:

- Kişi yetkisi dahilinde sayfalara erişebilir ve sadece kendi bölümüne ait evrak detaylarına ulaşabilir.
- Sadece yetkili kişiler evrağa ait bilgileri güncelleyebilir ve evrağın ait olduğu bölüm adında değişiklik yapabilir.
- Kişi User yetkisine sahipse yeni evrak giriş sayfasını açamaz.
- Sadece evrağın ait olduğu bölüm personeli, yapılan değişiklikleri mesaj olarak yazabilir veya detay sayfasından bu mesajların tarihçesine ulaşabilir.
- Rapor ekranlarında her kullanıcı kendi bölümüne ait evrakların raporunu listeleyebilir.
- Dijital ortama kayıt edilen evrağa ilişkin belgeler, yetkisi olan kişiler tarafından görüntülenebilir. Sunucu üzerinden ulaşılan bu belgelerin güncellenerek kaydedilmesine izin verilmemiştir. Sunucuda bulunan klasöre erişim, tüm kullanıcılar için sadece okuma iznini kapsamaktadır.
- Kullanıcılar parola değiştirme işlemini yapabilirler ancak yeni kullanıcı ve yeni bölüm tanımlama sayfaları yetkiyle sınırlandırılmıştır.

Yukarıda bahsedilen kullanıcı izinlerini tanımlamak için sistemde kullanılan yetkiler ve gerçekleştirebildikleri işlemler Çizelge 1'de görülmektedir.

Çizelge 1. Yetkiler ve İşlemler

Yetkiler İzinler	Kayıt Girişi	Silme	Yanıt Mesajı	Rapor	Dosya Yükleme	Kurum İşlemleri	Güncelleme	Kullanıcı işlemleri
Yönetici (Admin)	Evet	Evet	Evet	Tüm Bölümler	Evet	Evet	Evet	Evet
Bölüm Yetkilisi	Hayır	Hayır	Sadece Kendi Bölümü	Sadece Kendi Bölümü	Hayır	Hayır	Hayır	Hayır
Kayıt Sorumlusu	Evet	Hayır	Hayır	Tüm Bölümler	Evet	Evet	Evet	Hayır
Yetkili Kullanıcı	Hayır	Hayır	Sadece Kendi Bölümü	Sadece Kendi Bölümü	Hayır	Hayır	Hayır	Hayır
Kullanıcı (User)	Hayır	Hayır	Sadece Kendi Bölümü	Sadece Kendi Bölümü	Hayır	Hayır	Hayır	Hayır

Sisteme giriş yapılmadan, sayfa adreslerinin adres çubuğuna yazılması ile sisteme erişim yapılamamakta, kişi Login sayfasına yönlendirilmektedir. Sistemde açılan oturumlar belirli bir süre işlem yapılmadığında sonlandırılmakta ve tekrar giriş yapılması gerekmektedir. Ayrıca sistem kurum içerisinde kullanılacağı için giriş sayfasına IP kontrolü eklenmiştir. Böylece sistemin, yazılımsal olarak dışarıdan erişimleri engellemesi sağlanmıştır.

```
public boolean IPControl(HttpServletRequest request)
throws Exception
 {
    boolean kontrol = false;
    String ip = request.getRemoteAddr();

    if ((ip.substring(0,7)).equals("192.168")) {
    kontrol= true;
    }
    return kontrol;
 }
```

Yukarıda belirtilen kod ile sisteme erişmek isteyen IP kontrol edilmekte ve IP'nin belirtilen değer ile başlaması durumunda sisteme girişe izin verilmektedir. Bu kontrolü güçlendirmek için kullanılan bir diğer ayarlama da, uygulamanın kullanılacağı port numarasının belirlenmesidir. Sistem sadece intranette kullanılacağı için uygulama için belirlenen portun internete kapatılması daha etkin güvenlik sağlamaktadır.

```
<Connector
className="org.apache.tomcat.service.PoolTcpConnecto
r">
        <Parameter name="handler"

value="org.apache.tomcat.service.http.HttpConnectionHa
ndler"/>
        <Parameter name="port"
          value="9999"/>
 </Connector>
```

Sistemin amacı, gelen ve giden evrakları dijital ortamda saklamaktır. Tarayıcı aracılığıyla dijital ortama aktarılan evraklar sistemdeki Upload ekranı kullanılarak sunucuda bulunan evrak klasörüne yüklenmektedir. Ancak aynı isimli yeni bir evrağın yüklenmesi ihtimaline karşı önlem alınmıştır. Her evrak klasöre iki kere yüklenmektedir. Bunlardan ilki birebir evrağın adını içerirken, ikincisinin uzantısına sistem tarihi eklenmektedir. Örneğin sisteme 29 Nisan 2010 tarihinde yüklenen "a.doc" isimli bir evrak klasörde hem "a.doc" hem de "a.doc20100429" şeklinde yer almaktadır. Her evrağın yüklendiği tarihe ilişkin yedeğinin olması da evrağın ilk haline bile ulaşılmasını sağlamaktadır.

Veritabanında yapılan her değişikliğin yedeğini almak için tabloların bir de yedekleme yani log tabloları oluşturulmuştur. Bu log tabloları veritabanı yönetim sistemleri tarafından oluşturulan log tablolarından farklıdır. Bu tablolar veritabanı tasarımında oluşturulmuş ve yapılan her değişikliğin ilgili tablo ile aynı anda log

tablosuna da eklenmesi sağlanmıştır. Bu sayede hatalı bir işlemden geri dönmek için veritabanının yedeğinin yüklenmesine gerek kalmamakta, hatta kayıt bazında geri dönüş işlemi yapılabilmektedir.

Bunun yanında kayıt silme işlemi Admin yetkisine sahip kişiye verilmiştir ve sil butonları diğer kullanıcılar tarafından görüntülenememektedir (Bkz. Şekil 2 ve Şekil 3). Kayıt silme işleminde kayıt tamamen silinmemektedir. Yeni bir evrak kayıt edilirken tabloda yer alan "sil" alanı hayır olarak girilmektedir. Silme işleminde ise silinmek istenen kayıt veya kayıtlar seçilmekte ve sil butonuna tıklandığında "delete (sil)" işlemi yerine "update (güncelle)" işlemi yapılmaktadır. İlgili kayıtlara ait sil alanı "evet" olarak güncellenmektedir. Sistemde yer alan kayıtların listelenmesinde de "select * from tablo where sil='hayır' " şeklinde bir sorgu kullanılmaktadır.

GELEN EVRAKLAR

Sıra	Evrak Tipi	Evrak Cinsi	Karşı Kurum Tarihi	Evrak Ek Sayısı	Kayıt Tarihi	Evrak Numarası + -	Görevlendirilen + -	Son Güncelleme	Detay	İlgili Giden Evrak	Dosyalama Tarihi	Dosya Numarası	Sil?
3147	Gelen Evrak	Aps	04/03/2010	5	04/03/2010	8437	AFYONMYO,BOLVADİNMYO,EĞİTİM FAK.		...				
3146	Gelen Evrak	Süreli	03/02/2010	9	03/02/2010	8436	AFYONMYO,BOLVADİNMYO	04/03/2010 11:02:14	...				
3145	Gelen Evrak	Süreli	03/02/2010	8	03/02/2010	8435	AFYONMYOBOLVADİNMYOEĞİTİM FAK.		...				
3143	Gelen Evrak	Süreli	01/02/2010		01/02/2010	8433	AFYONMYO	01/02/2010 14:58:59	...				
3142	Gelen Evrak	Süreli	01/02/2010	1	01/02/2010	8432	AFYONMYO	01/02/2010 10:55:23	...				
3141	Gelen Evrak	Aps	25/01/2010	55	25/01/2010	8431	AFYONMYO	01/02/2010 09:27:03	...				
3139	Gelen Evrak	Aps	25/01/2010	4	25/01/2010	8429	AFYONMYO	25/01/2010 10:44:13	...				
3138	Gelen Evrak	Süreli	25/01/2010	45	25/01/2010	8428	AFYONMYO		...				
3137	Gelen Evrak	Süreli	25/01/2010	4	25/01/2010	8427	AFYONMYO		...				
3136	Gelen Evrak	Süreli	25/01/2010	5	25/01/2010	8426	AFYONMYO		...				

SİL

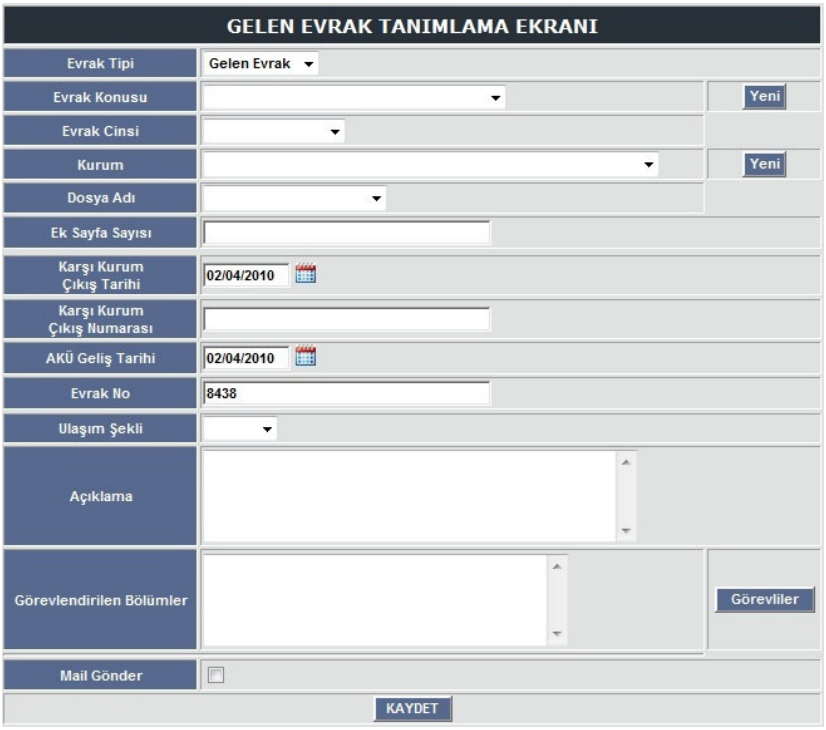
Şekil 2. Admin Evrak Liste Ekranı

GELEN EVRAKLAR

Sıra	Evrak Tipi	Evrak Cinsi	Karşı Kurum Tarihi	Evrak Ek Sayısı	Kayıt Tarihi	Evrak Numarası + -	Görevlendirilen + -	Son Güncelleme	Detay	İlgili Giden Evrak	Dosyalama Tarihi	Dosya Numarası	Sil?
3144	Gelen Evrak	Aps	03/02/2010	4	03/02/2010	8434	TIP FAK.,UEMYO	03/02/2010 12:34:53	...				
3144	Gelen Evrak	Aps	03/02/2010	4	03/02/2010	8434	TIP FAK.,UEMYO	03/02/2010 12:34:53	...				
3144	Gelen Evrak	Aps	03/02/2010	4	03/02/2010	8434	TIP FAK.,UEMYO	03/02/2010 12:34:53	...				

Şekil 3. User Evrak Liste Ekranı

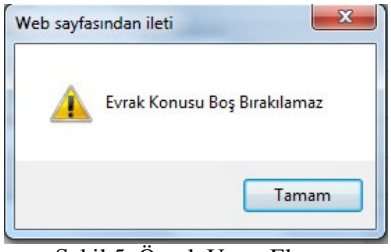
Kayıt ekranında mümkün olduğunca klavyeden veri girişinin az olması amaçlanmıştır. Tarih alanları için html sayfasından oluşan takvimler kullanılmıştır. Evrak adı, evrak cinsi, gönderilen kurum, ilgili bölüm gibi alanlar için de açılan kutular tercih edilmiştir. Gelen evrakta yer alan bilgilere göre alanlara veri girilmekte ve Şekil 4'te görüldüğü üzere sadece birkaç alanın veri girişi klavyeden gerçekleştirilmektedir. Klavyeden veri girişinin aza indirilmesinin sebebi veri tutarlılığının sağlanmasıdır. Diğer evraklar için de kullanılacak olan kurum adı, evrak konusu, evrak cinsi, dosya adı, ulaşım şekli gibi ortak alanların açılan kutudan seçilerek girilmesi hem kayıt girme süresinden tasarruf sağlamakta hem de izinsiz ve tutarsız veri girişini önlemektedir. Böylece rapor alma veya kayıt arama gibi işlemlerde daha doğru sonuçlar elde edilmektedir.



Şekil 4. Evrak Tanımlama Ekranı

Sistemde yeni evrağa ait Evrak No alanı otomatik artmakta ve alan salt okunur olarak belirlendiği için müdahale edilememektedir. Böylece aynı numaranın tekrar kullanılması, numara atlanması gibi durumların önüne geçilmektedir. Ayrıca kayıt işlemi esnasında, tekrarlı kayıt olasılığı da kontrol edilmektedir. Eğer kaydedilmek üzere girilen evrağın kurum adı, karşı kurum çıkış numarası ve karşı kurum çıkış tarihi veritabanında herhangi bir kayıt ile eşleşiyorsa "Bu Kayıt "8438" Evrak Numarası ile daha önce kaydedilmiştir, kontrol ediniz!" uyarısı ile yetkilinin kontrol etmesi istenmekte ve tekrarlı evrak girişi önlenmektedir.

Kayıt işleminde zorunlu alanlar boş bırakıldığında sistem Şekil 5'teki gibi uyarı vererek kullanıcıyı uyarmakta ve kaydetme işlemi yapılmadan, veri girilmesi için aynı ekrana geri dönülmektedir.

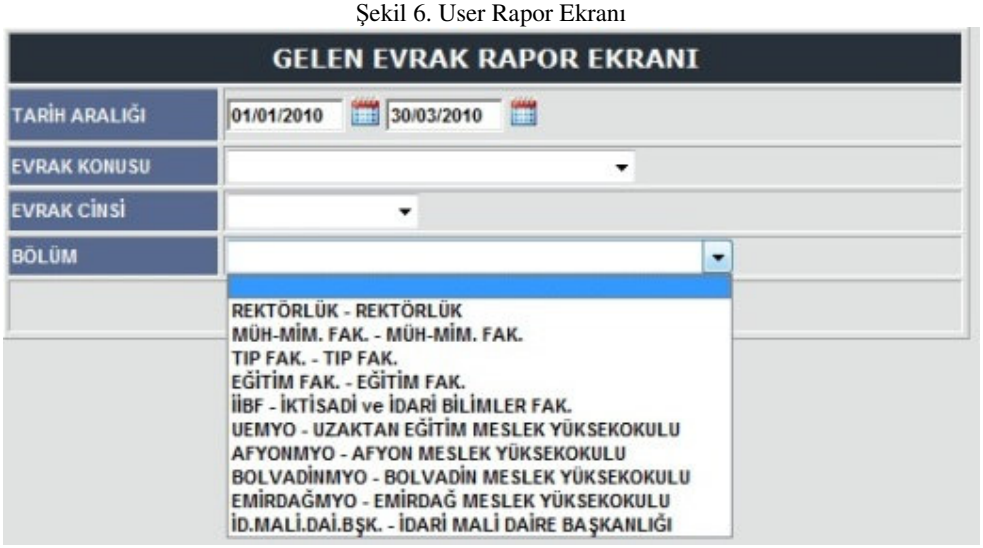


Şekil 5. Örnek Uyarı Ekranı

Kayıt ekranında zorunlu alanlar evrak konusu, karşı kurum çıkış numarası, ulaşım tipi, açıklama, görevlendirilen bölümler ve karşı kurum alanlarıdır.

Gelen Evrak ve Giden Evrak uygulamalarında girilen kriterlere göre raporlar alınabilmektedir. Rapor alma sayfasında da yetkilere göre bazı sınırlamalar yapılmıştır. Şekil 6'da görüldüğü üzere Admin ya da yönetici olmayan kişiler sadece kendi bölümlerine ait evrakların raporunu alabilirler. Bu sınırlama rapor kriterleri belirleme ekranında seçilen Bölüm alanı kullanılarak yapılmıştır. Admin yetkisindeki kullanıcı tüm bölümleri seçebilirken (Bkz. Şekil 7), User yetkisindeki kişi için Bölüm alanında sadece kendi bölüm adı yer almaktadır.

Şekil 6. User Rapor Ekranı



Şekil 7. Admin Rapor Ekranı

5.SONUÇ VE ÖNERİLER

Kamuda ve özel şirketlerde kullanılan uygulamaların arasında yerini alan evrak yönetim sistemleri gittikçe yaygınlaşmaktadır. Kullanılırlığının artması sistemlerin eksikliklerinin giderilerek, farklı yaklaşımların doğmasına sebep olmaktadır. Belgelerin sadece kayıtlarının tutulduğu sistemden ziyade, her belgenin yaşam döngüsünün takip edilebildiği sistemler haline gelmektedirler. Fonksiyonelliklerin artmasının yanında kaçınılmaz olarak güvenlik konusu da sistemler için zorunlu bir hal almaktadır. Evrak yönetiminde genel olarak üç farklı durum için güvenlik kavramı kullanılmaktadır.

- Evrağın orijinal olup olmadığı (reliable).
- Evrağın bir bütün olması, değiştirilmemiş olması, herhangi bir şekilde zarar görüp görmeyeceği.
- Gizli dokümanlara herkesin erişip erişemeyeceği.

Geliştirilen örnek evrak uygulamasında da temel olarak bu üç güvenlik unsurunun sağlanması hedeflenmiştir. Bu hedef doğrultusunda, kayıttan arşivleme sürecine kadar geçen tüm işlemlerde evrağın bütünlüğünü koruması için birçok güvenlik fonksiyonu eklenmiştir. Bu fonksiyonların başında gelen yetkilendirme işlemi ile kullanıcı hakları keskin sınırlar ile çizilmiş ve evrak gizliliği korunmaya çalışılmıştır. Ayrıca sisteme erişimdeki sınırlamalar ile de güvenlik güçlendirilmiştir.

Bu makalede, yukarıda bahsedilen güvenlik önlemlerinin benzer uygulamalara ışık tutması hedeflenmiştir. Bunların dışında uygulamaya daha farklı güvenlik önlemleri de eklenebilir. Dijital imzanın sisteme entegre edilmesi evrakların daha güvenli olmasını sağlayacaktır.

KAYNAKLAR

[1] S. Y. Na, S. Lee, "Design of Security Mechanism for Electronic Document Repository System", **International Conference on Convergence and Hybrid Information Technology**, Daejeon, 708-715, 2008.

[2] J. B. Liu, X. Q. Hu, Q. Li, X. M. Niu, "Design and Implementation of a PKI-Based Electronic Documents Protection Management System", **Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007)**, Kaohsiung, 87-92, 2007.

[3] S. Madan, "Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks", **2010 International Conference on Intelligent Systems Modelling and Simulation**, Liverpool, 226-230, 2010.

[4] S. King, "Appliying application security standards- a case study", **Computers &Security**, 23(1), 17-21, 2004.

[5] J.Zadeh, D. DeVolder, "Software Development and related Security Issues", **IEEE SoutheastCon,** Richmond, VA, 746-748, 2007.

[6] E. A. Nichols, G. Peterson, "A Metrics Framework to Drive Application Security Improvement", **IEEE Security & Privacy**, 5(2), 88-91, 2007.

[7] T. Cramer, R. Friedman, T. Miller, D. Seberger, R. Wilson, M. Wolczko, "Compiling Java Just In Time", **IEEE Micro**, 17(3), 36-43, 1997.

[8] Internet: Java (programlama dili),

http://tr.wikipedia.org/wiki/Java_dili, 30.08.2010.

[9] A. V. Hoff, "The Case For Java as A Programming Language", **IEEE Internet Computing**, USA, 1(1), 51-56, 1997.

[10] Ö. Karal, **Java Ortamında Bulanık Mantık Kontrol: Kamyon Yükleme–Boşaltma Uygulaması**, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, 2004.

[11] C. Erpolat, **Java Programlama Dilinin Bilgisayar Destekli Öğretimi**, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, 2006.

[12] S. Shmeltzer, **Oracle JDeveloper 10g Overview**, Oracle Corporation, U.S.A., 2004.

[13] D. Petkovic, **Microsoft Sql Server 2005**, N. G. Küçükler, Alfa Yayınları, İstanbul, 2006.

[14] C. Dawes, B. Bryla, J. C. Johnson, M. Weishan, **OCA(Oracle 10G Administration 1 Study Guide),** M. Lum, Sybex, U.S.A., 2005.

[15] R. M. Menon, **Expert Oracle Jdbc Programming,** T. Davis, Appres, U.S.A., 2005.