REPUBLIC OF TURKEY
MINISTRY OF ENVIRONMENT
AND URBANISATION

unisystems

Türkiye Emisyon Ticareti Planı

Yazılım Tasarım Dokümanı

Sözleşme Development of the ETS Transactions Registry in Turkey
Ref.: TR-ETS_Sofware_Design_Document_v1.6.docx

Durum Bilgisi

		Durum :	Taslak
Versiyon No:	V1.6	İlk yayın tarihi :	01/04/2020
Hazırlayan:	Unisystems	Tarih:	16/09/2020
Doğrulayan:	Unisystems	Tarih:	
Onaylayan:	Unisystems	Tarih:	

Dağırım listesi

Ad	Adres	B/G	Ad	Adres	B/G
Engin Mert	CSB	A			I
		A			

(B = Bilgi için, G = Gereği için)

Döküman değişim kaydı

Versiyon	Tarih	Açıklama	Etkilenen bölümler
0.1	01/03/2020	İçindekiler oluşturuldu	Tümü
1.0	27/03/2020	Bölümler sonuçlandırıldı	Tümü
1.2	01/04/2020	Son yorumlar ve format değişiklikleri	Tümü
1.3	06/05/2020	CSB tarafından alınan uygulamalı yorumlar. Yazım hataları düzeltme ve yeniden ifade etme. E-R diyagramı ve Veri görünümü bölümünün güncellenmesi Şekil başlıklarının ve şekil tablosunun güncellenmesi	Tablo 1: tipleri, Tablo 3: matriksi 5.1.3, 5.1.4,5.1.4.2, 5.1.4.3, 5.1.4.14, 5.1.4.6,5.1.4.9, 5.2.3.1,5.2.3.1, 5.2.4.8, 5.2.5, 5.3.4
1.4	25/08/2020	Mimari Veri görünümü güncellendi	5.2.5
1.5	27/08/2020	Uygulanan ayrıntılarla senkronize edildi. Taahhüt Süresi için açıklama eklendi	Tümü §2.1.3
1.6	15/09/2020	Birebir / yazım hatası düzeltmeleri ve ayrıntılandırmaları	Tümü

TABLE OF CONTENTS

ŞEKİLLER LİSTESİ

1. SUNUŞ

1.1 Kapsam

Bu projenin amacı, Türkiye'de Emisyon Ticaret Sistemi İşlem Sicilinin geliştirilmesi ve dünya çapında mevcut örneklere dayalı danışmanlık hizmetlerinin sağlanmasıdır.

Emisyon Ticareti Planları, karbondioksit emisyonlarını kontrol etmek ve kademeli olarak azaltmak için uygun bir yol haritası sağlar. İşletme düzeyinde ve akademik düzeyde kanıtlandığı gibi, sınır ve ticaret programları kirlilikle mücadelede vergilendirme programlarından daha etkilidir, çünkü hem büyük kirleticileri 'cezalandırır' hem de yeşil teknolojilere yatırım yapan kirleticileri ödüllendirir ve artık daha az emisyon yayarlar.

Tasarlanan çözümün kapsamı aşağıdakileri içerir:

- Kullanıcıların sistemde güvenli bir şekilde oturum açabilmeleri ve rollerine ve hesaplarına göre uygun bilgilere erişebilmeleri için kimlik doğrulama ve yetkilendirme.
- Hesaplar arasında birim transferleri (tesisler ve ticaret kurumlarına ait)
- Fabrikalar tarafından emisyonlarına uyum sağlamak için birimlerin teslim edilmesi
- Birimleri silme
- Hesaplardaki birim varlıkları; her kullanıcı, hesabın tuttuğu birimleri gösteren kendi hesabına / hesaplarına erişime sahip olacaktır.
- İşlemler aranabilir ve işlem detayları sunulabime
- Kullanıcıları arama / kullanıcıların ayrıntılarını görüntüleme
- Çözüm, çeviri dosyaları aracılığıyla Türkçeye çevrilebilir. Böylece kullanıcı arayüzü her iki dilde, yani İngilizce ve Türkçe olarak mevcut olacaktır.

Spesifik Yazılım Tasarım Dokümanı (SDD), teknik gereklilikler ve sistem mimarisi ile birlikte İş gereklilikleri ve Kullanım senaryolarının yakalanmasına odaklanır. Belli bir belge için girdi toplanmıştır.:

- PMR- FCPF Kayıt Rehberi [1]. SDD'nin içindekiler tablosu, bu belgede sağlanan yönergelere dayanmaktadır.
- Haftalık analiz ve teknik toplantılar
- Çevre ve Şehircilik Bakanlığından alınan belgeler
- Yüklenicinin Avrupa Birliği Emisyon Ticareti Planının işletilmesi, geliştirilmesi ve desteklenmesindeki göreceli deneyimi ve uzmanlığı

1.2 Referanslar

1	EMISSIONS TRADING REGISTRIES: Guidance on Regulation, Development, and Administration (109027-WP-PUBLIC-12-10-2016-15-54-42-PMRFCPFRegistriesPosting.pdf)
2	A description of the methodology and work plan for performing the assignment (02. (b) A description of the methodology and work plan for performing the assignment-v3.pdf)

3	MoEU standards and Guidelines for Database (CSB.YTDB.01.07_Veri_Tabani_Standartlari(1).pdf)
4	MoEU Software Application standards and Guidelines (yazilim-gelistirme-ve-birlikte-calisma-esaslari-proseduru-20180307142636.pdf)

1.3 Tanımlar ve Kısaltmalar

Kısaltma	Açıklama
API	Application Programming Interface (uygulama programlama Arayüzü)
AR	Authorised Representative (Yetkili Temsilci)
AAR	Additional Authorised Representative (Ek Yetkili Temsilci)
CDM	Clean Development Mechanism (Temiz Geliştirme Mekanizmaları)
CIA	Confidentiality Integrity and Availability (Gizlilik Bütünlüğü ve Kullanılabilirliği)
CP	Commitment Period (Taahhüt Süresi)
EJB	Enterprise Java Beans
ETS	Emission Trading Scheme (Emisyon Ticaret Şeması)
GHG	Greenhouse Gas (Sera Gazı)
IS	Information System (Bilgi Sistemi)
ITL	International Transaction Log (Uluslararası İşlem Günlüğü)
JAXB	Java Architecture for XML Binding (XML Bağlama için Java Mimarisi)
JDBC	Java Database Connectivity (Java Veritabanı Bağlantısı)
JMS	Java Message Service (Java Mesaj Servisi)
JPA	Java Persistence API (Java Süreklilik API)
JVM	Java Virtual Machine (Java Sanal Makine)

Kısaltma	Açıklama
KP	Kyoto Protocol (Kyoto Protokolü)
LDAP	Lightweight Directory Access Protocol (Basit Dizin Erişim Protokolü)
MRV	Monitoring, Reporting and Verification of Emissions (Emisyonların İzlenmesi, Raporlanması ve Doğrulanması)
MoEU	Ministry of Environment and Urbanization of Turkey (Türkiye Çevre ve Şehircilik Bakanlığı)
MVVM	Model–view–viewmodel (Model-görünüm-görünüm modeli)
OLAP	Online Analytical processing (Çevrimiçi analitik işleme)
OLTP	Online transaction processing(Çevrimiçi işlem işleme)
OHA	Operating Holding Account (İşletme Holding Hesabı)
OWASP	Open Web Application Security Project (Açık Web Uygulama Güvenliği Projesi)
RBAC	Role Based Access Control (Rol Tabanlı Erişim Kontrolü)
RDBMS	Relational Database Management System (İlişkisel veritabanı yönetim sistemi)
REST	Representational State Transfer (Temsili Durum Transferi- Web hizmetleri oluşturmak için kullanılacak bir dizi kısıtlamayı tanımlayan bir yazılım mimari stili)
SOAP	Simple Object Access Protocol (Basit Nesne Erişim Protokolü)
SSO	Single Sign On (bir kullanıcının tek bir kimlik ve parola ile oturum açmasına izin veren bir kimlik doğrulama şeması)
WS	Web Service (Web Servis)
WSDL	Web Service Definition Language (Web servis tanımlama dili)

2. KULLANICI GEREKSİNİMLERİ

2.1 Fonksiyonel Gereksinimler

2.1.1 FR1 – Tesis ve Ticaret Kurumları

Sistem tarafından desteklenen iki ana işletme, Tesisler (fabrikalar) ve Ticaret kurumlarıdır (bankalar, yatırımcılar). Bunlar, sistem kullanıcılarının atandığı gerçek ticari varlıkları simüle eder.

Sistem, hesap yönetimi sunmaz, ancak bu kayıtları MRV sisteminden çevrimdışı olarak alır..

2.1.2 FR2 – Kullanıcılar ve Roller

Kullanıcıların Tesisler ve ticaret kurumları ile fiili yazışmalarının ardından iş kullanıcıları hesaplara atanır.

Yöneticiler sistemdeki tüm bilgilere erişebilir.

Sistem, kullanıcı yönetimi sunmaz, ancak bu kayıtları PMRV sisteminden alır.

Not: Sistem Yöneticisi profili: Bu profil de desteklenir, yalnızca sistem işlemlerine erişime sahiptir ve ETS ile ilgili işlevlere erişim sağlamaz

2.1.3 FR3 – Türkiye Emisyon Tahsisatı

Sistem, çalıştığı sürece Türk Emisyon Tahsisatlarını (birimleri) yönetir. İhraç, işlemler, holdingler sonraki gereksinimlerde açıklandığı gibi sistem tarafından desteklenir.

Uluslararası uygulamanın, Taahhüt Dönemlerinde veya Aşamalarında ETS sistemlerinin çalışmasını organize ettiği unutulmamalıdır. Bunlar, ödeneklerin işleyişini, verilmesini ve değişimini düzenleyen özel mevzuatla birlikte 8-10 yıl süren dönemlerdir. Daha sonraki bir dönemde, farklı kurallar ve kısıtlamalarla birlikte yeni mevzuat mevcut olabilir. Bu nedenle, bir Aşama veya Taahhüt Dönemindeki ödenek sorunları, diğer Aşamalar veya Taahhüt Dönemlerindeki ödenek sorunlarından ayırmak için bu mülkü onlara işaretler.

2.1.4 FR4 - Emisyonlar

Operatör holding hesapları (bunlar tesislere veya fabrikalara karşılık gelen hesaplardır) emisyonlara sahiptir. Sistem, emisyon yükümlülüklerini yönetmez, ancak bunları harici kaynaklardan (yani, MRV sistemi) alır.

Operatör sahibi hesapların kullanıcıları, hesaplarının emisyonlarını görüntüleyebilir ve böylece teslim edilmesi gereken miktarı bilebilir.

2.1.5 FR5 – Transfer işlemi

Tesis ve Ticaret kurumlarının sistemde hesapları vardır; iş kullanıcıları bu tür bir veya daha fazla hesaba bağlıdır; bu hesaplar, transfer adı verilen bir işlem türü aracılığıyla hesaplar arasında aktarılabilen birimleri tutar.

Sistem tarafından desteklenen transferler, aynı sistemde barındırılan iki hesap, TR-ETS arasında gerçekleştirildiği için dahili transfer olarak da adlandırılır.

2.1.6 FR6 – Teslim işlemi

Operatör holding hesaplarının kullanıcıları birimleri teslim edebilir, böylece emisyonlarına uyum sağlayabilir.

2.1.7 FR7 – Silme işlemi

Tüm hesapların kullanıcıları birimleri silebilir. Silinmeler, hesapların emisyon yükümlülüklerine dahil değildir.

2.1.8 FR8 – İhale işlemi

İhale, birimlerin Toplam Miktar Hesabı'ndan ihale teslimat hesaplarına aktarılmasıdır, böylece ihaleler bu sistem kapsamı dışında gerçekleştirilebilir.

Birimlerin Toplam Miktar Hesabından bir açık artırma teslimat hesabına aktarılmasıyla açık artırma gerçekleştirilebilir. Bu işlem yalnızca yöneticiler tarafından gerçekleştirilir.

Daha sonra ihale teslimat hesabı kullanıcıları sistem kapsamı dışında ihaleyi gerçekleştirir. Son olarak, ihale teslimat hesabı kullanıcıları, ihale teslimat hesabından ilgili ihalenin en yüksek teklif verenine transferleri sisteme manuel olarak ekler.

2.1.9 FR9 – Tahsis işlemi

Tahsis, birimlerin ulusal politikayı uygulayan işletmeci hesaplarına aktarılmasıdır.

Bir tahsis, birimlerin Toplam Miktar Hesabından bir operatör holding hesabına aktarılmasıyla gerçekleştirilebilir. Bu işlem yalnızca yöneticiler tarafından gerçekleştirilir.

2.1.10 FR10 - Hesaplarda tutulan ödenekler

Her hesap, hesabın tuttuğu birimleri Holding hesapları adı verilen bir ekranda sunar. Hesaba bağlı kullanıcılar birimleri görüntüleyebilir ve işlem önerebilir.

2.1.11 FR11 – İşlem listesi

Hesaplar arasındaki işlemler saklanır ve bunları çeşitli arama kriterleri ile gerçekleştiren kullanıcılar tarafından aranabilir.

2.1.12 FR12 – Kullanıcıları arama

Yöneticiler, sistemde tanımlı tüm kullanıcıları arayabilecektir.

2.1.13 FR13 – Hesap tipleri

Table 2 'de detaylandırıldığı üzere birimlerin yaşam döngüsünü ele almak için bir dizi iş ve idari hesap türü geliştirilecektir.

İdari hesaplara yalnızca yöneticiler erişebilir ve büyük miktarlarda birimi tutan muhasebe havuzları olarak hizmet ederler.

Operatör holding hesapları kurulumlara aittir ve sadece bağlı kullanıcıları tarafından erişilir.

2.1.14 FR14 - Komut dosyasıyla birimler oluşturma (yayınlama)

Bir komut dosyası çalıştırılarak yeni Türk emisyon tahsisatları (birimleri) oluşturulabilir (çıkarılabilir). Yeni üniteler, Bölüm §**Error! Reference source not found.**. 'te sunulan yaşam döngüsünü takip eder

2.1.15 FR15 - Uyumluluk

Operatör holding hesaplarının emisyon yükümlülüklerine eşit veya daha fazla birim miktarları teslim etmesi gerekir. Tüm Kurulumları Uyum durumlarıyla birlikte gösteren bir rapor mevcuttur. Bu rapor:

- A: Tesis emisyon sağladı ve bildirilen emisyonları karşılamaya yetecek kadar birim teslim etti.
- B: Kurulum emisyon sağladı ancak bu emisyonları karşılayacak kadar ünite teslim etmedi
- C: Kurulum hiç emisyon sağlamadı

Uyum politikaları çok yıllı dönemlere uzanabileceğinden, Uyum döngüsünün ayrıntıları analiz sırasında belirlenecektir [1].

[1] EUETS, grupları gruplara ayırmak için "Aşama" terimini kullanır. Örneğin 3. Aşama, 2013-2020 dönemidir. 4. aşama 2021-2030 olacak. EUETS'teki bir kurulum, Faz içindeki bireysel yıllar için değil, tüm Faz için uyumlu olabilir.

2.2 Fonksiyonel Olmayan Gereksinimler

2.2.1 NFR1 - Kimlik doğrulama ve yetkilendirme

Kullanıcılar sistemde güvenle oturum açabilir ve rollerine ve hesaplarına göre uygun bilgilere erişebilirler.

2.2.2 NFR2 – Yönetici erişimi

Yöneticiler özel bir rol, sistem tarafından mevcut tüm işlevlere ve tüm iş verilerine erişebilir.

2.2.3 NFR3 – Güvenli erişim

Yetkisiz hiçbir kullanıcı sisteme erişemez. İşletme kullanıcıları, yalnızca bağlı oldukları hesapların iş verilerine erişebilir. Diğer hesapların verilerine erişemezler.

Yöneticiler tüm bilgilere erişebilir. Yalnızca Yöneticiler yönetici hesaplarına erişebilir.

2.2.4 NFR4 – Türkçeye tercüme

Böylelikle kullanıcı ara yüzü İngilizce ve Türkçe olmak üzere iki dilde mevcut olacaktır. Bu, çeviri dosyalarıyla sağlanacaktır.

2.2.5 NFR5 - Standart, açık kaynaklı teknolojilerin kullanımı

Seçilen açık kaynak teknolojileri, Bölüm**Error! Reference source not found.**'te ayrıntılı olarak listelenmiştir.

2.2.6 NFR6 - Avrupa'daki en iyi uygulamalar

Sistemin tasarımı ve uygulanan tüm kuruluşlar Avrupa'nın en iyi uygulamalarına uyum gösterecektir.

2.2.7 NFR7 - Komut dosyaları aracılığıyla MRV'ye ara yüz

Çözüm, kullanıcıları, kurulum hesaplarını, emisyonları ve diğer gerekli verileri almak için komut dosyaları aracılığıyla MRV ile çevrimdışı ara yüz oluşturacaktır.

2.2.8 Performans ve Yük

Sistem, en az 700 operatör holding hesabını (kurulumlara karşılık gelen hesap), 30 yönetici hesabını ve 1000 kullanıcıyı destekleyebilmelidir.

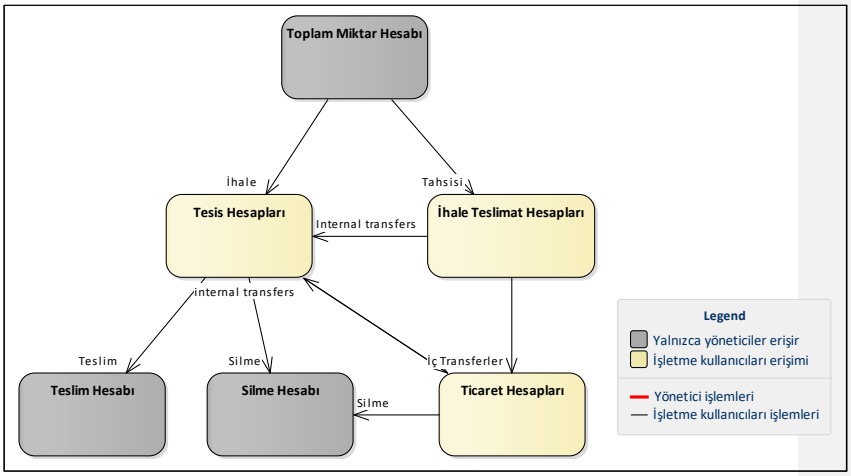
2.2.9 Kullanılabilirlik ve erişilebilirlik

Sistemin, UI / UX en iyi uygulamalarına göre kullanıcı dostu olması ve ayrıca yaygın olarak kullanılan tarayıcılar (en son sürüm) ve cihazlarda erişilebilir olması beklenmektedir.

2.3 Muhasebe Akış Şeması ve İlkeleri

Mevcut bölüm, sistem tarafından sunulacak işlem türlerini, hesap türlerini ve ilgili kullanıcıları analiz eder.

Şekil 2.3-1: Muhasebe Modeli



Ayrıntılı olarak, yukarıdaki muhasebe modeli aşağıdaki şekilde açıklanmıştır:

No	İşlem Adı	Aktör	Kaynak Hesap	Heded Hesap	Açıklama
1	İhraç	Yok	Toplam miktar hesabı	Toplam miktar hesabı	Komut dosyası aracılığıyla yeni tahsisatlar oluşturulur
2	Tahsis	Yönetici	Toplam miktar hesabı	Operator holding hesabı	Bunlar dahili transferlerdir, tahsisatları tesislere dağıtır, ulusal bir politika uygular.

3	İhale	Yönetici	Toplam miktar hesabı	İhale teslim hesabı	Bunlar, ödenekleri kurumsal hesaplara dağıtan dahili transferlerdir. Bu kurumlar, ihale bitiminden sonra, ihaleyi dış yollarla yapacak ve sonuçları bu sisteme iç aktarım olarak ekleyecektir.
4	Silme	İş kullanıcısı	Operator holding hesabı	Silme hesabı	Bunlar birimleri piyasadan kaldırarak "yok eden" işlemlerdir; kurulumun emisyon yükümlülüklerini kapsamazlar.
5	Teslim	İş kullanıcısı	Operator holding hesabı	Teslim hesabı	Bu işlemler, tesisin emisyon yükümlülüklerini kapsar.
6	(Dahili) Transfer	İş kullanıcısı	Operator holding hesabı veya Ticari hesap	Operator holding hesabı veya Ticari hesap	Bu işlemler TR-ETS içinde tanımlanan hesaplar arasında ödenek aktarımına izin verir.

Tablo 1: İşlem tipleri

Aşağıdaki hesap türleri desteklenmektedir:

Hesap Tipleri	Açıklama	Erişebilirlik
Operator holding hesapları[2]	Bunlar, sistemde kayıtlı her kurulum için bir tane karşılık gelen en yaygın hesap türleridir. Bu hesapların emisyon uyum yükümlülükleri vardır.	Yönetici tüm hesaplara erişebilir. İşletme kullanıcıları kendi hesaplarına erişebilir.

[2] Bunlar, diğer Emisyon Ticaret Planlarında Operatör Holding Hesapları olarak adlandırılır

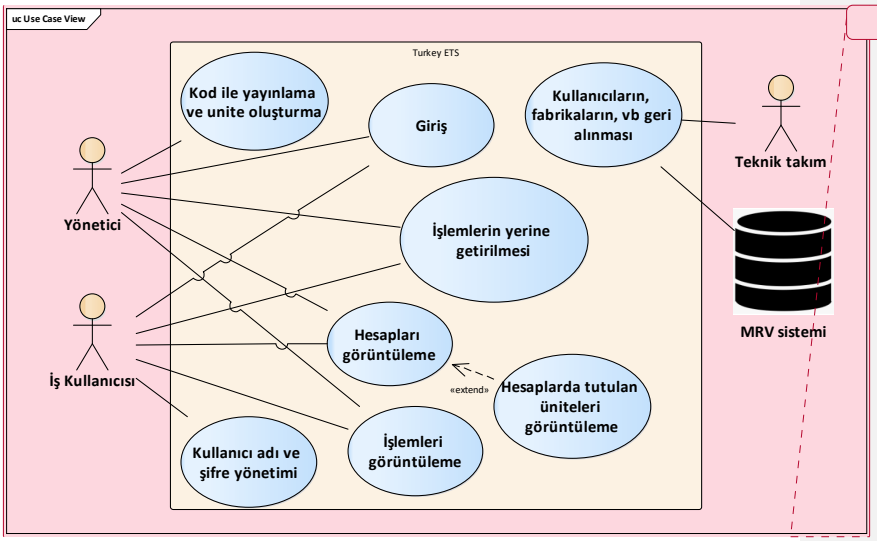
Hesap Tipleri	Açıklama	Erişebilirlik
Ticari hesaplar	Bunlar, Bankalara, yatırım şirketlerine veya bireysel yatırımcılara karşılık gelen yatırım hesaplarıdır. Bu hesapların uyum yükümlülüğü yoktur.	Yönetici tüm hesaplara erişebilir. İşletme kullanıcıları kendi hesaplarına erişebilir.
İhale teslim hesapları	Bunlar ihale, ihalelerin yapılmasından sorumlu kurumlardır. Bu hesaplar, her ihaleden önce ihaleye çıkarılacak büyük miktarlarda ödenek alır. İhaleler TR-ETS kapsamında yapılmaktadır. Her ihalenin kazananlarına (en yüksek teklif verenler) yönelik işlemler, her ihalenin bitiminden sonra, ihale teslimat hesabı ile miktarı veren ilgili operatör holding hesabı veya ticari hesap arasında dahili transferler olarak sisteme manuel olarak yazılır.	Yönetici tüm hesaplara erişebilir. Bu hesapların kullanıcılarının sisteme erişip erişmeyeceğine karar verilecektir.
Toplam miktar hesabı	Bu, yeni tahsisatların verildiği (oluşturulduğu) hesaptır.	Bu tür bir hesaba yalnızca yöneticiler erişebilir.
Silme hesabı	Bu, tüm silme işlemlerinin hedef hesabıdır.	Bu tür bir hesaba yalnızca yöneticiler erişebilir. İşletme kullanıcıları, yalnızca silme işlemleri yoluyla bu hesaba aktarabilirler. Bu hesaptan hiçbir transfer yapılmasına izin verilmez.
Teslim hesabı	Bu, tüm teslim işlemlerinin hedef hesabıdır.	Bu tür bir hesaba yalnızca yöneticiler erişebilir. İşletme kullanıcıları bu hesaba yalnızca teslim işlemleri yoluyla aktarabilirler. Bu hesaptan hiçbir transfer yapılmasına izin verilmez.

Table 2: Hesap tipleri

3. KULLANIM SENARYOLARI

Bu bölüm, TR-ETS'nin işlevselliğini tanımlayan kullanım durumlarını sunar.

İçerilen işlevsellik, Unisystems'in teklifinden kaynaklanmaktadır ([2] 'ye bakın) ve Çevre ve Şehircilik Bakanlığı tarafından incelemeye tabidir.



Commented [EM1]: Please add figure number

Sunulan işlevselliğin ima ettiği ekran akışı aşağıdaki diyagramda görülmektedir.:

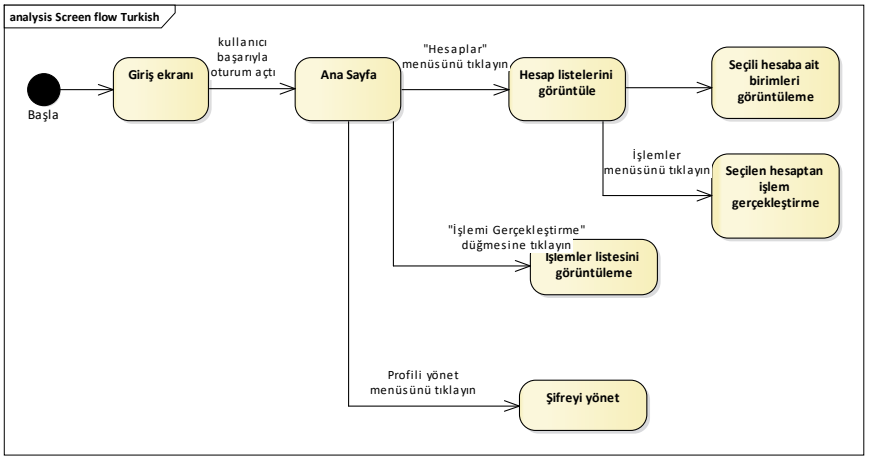


Figure 2.3-1: Ekran Akışı

3.1 Kullanım senaryosu UCS-01: Oturum açma

3.1.1 Kullanım senaryosu kısa açıklaması

Bu kullanım örneği, kullanıcıların sisteme nasıl bağlandığını açıklar.

3.1.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler
- İşletme Kullanıcıları

3.1.3 Ön koşullar

3.1.3.1 Kullanıcı sisteme kayıtlıdır

Kullanıcı kayıtlı olmalı ve kullanıcı adı ve şifresini bilmelidir

3.1.4 Temel akış

1. Kullanıcı, sistemin ana sayfasına gider. Şekil 3.2'ye bakın.
2. Kullanıcı, kullanıcı adını ve parolayı yazar.
3. Kullanıcı "Oturum Aç" düğmesine tıklar.
4. Kullanıcı sistemin özel alanına girer.

3.1.5 Alternatif akış

Yok

3.1.6 İstisna akışları

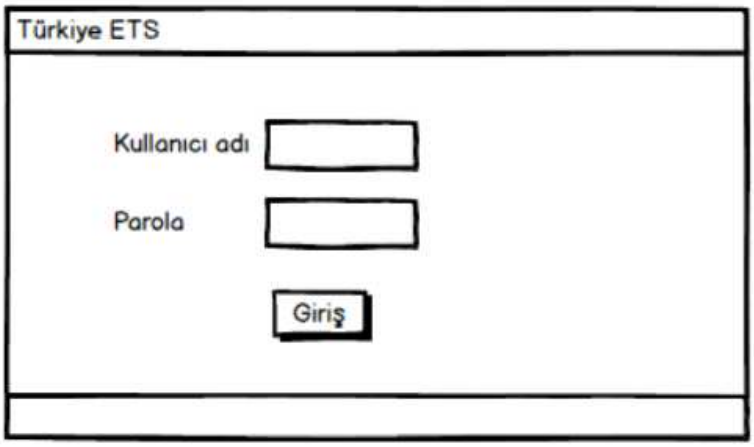
3.1.6.1 Kullanıcı hem kullanıcı adını hem de parolayı girmez

Temel Akışın 2. adımında, kullanıcı adını veya parolayı boş bırakır:

1. Kullanıcı "Giriş yap" düğmesini tıklar
2. Sistem, "Lütfen hem kullanıcı adı hem de şifre girin" şeklinde yanıt verir
3. Kullanım senaryosu sona erer.

3.1.7 İlave bilgi

3.1.7.1 Ekranlar



Şekil 3.1-1: Giriş ekranı

3.2 Kullanım senaryosu UCS-02: Şifre yönetimi

3.2.1 Kullanım senaryosu kısa açıklaması

Bu kullanım örneği, kullanıcıların kendi parolalarını nasıl yönetebileceklerini açıklamaktadır.

Bu, kısıtlamalara tabi teknik bir kullanım durumudur.

3.2.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler
- İşletme Kullanıcıları

3.2.3 Ön koşullar

3.2.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.2.4 Temel akış

1. Kullanıcı, "Profili Yönet" menü girişine tıklar.
2. Kullanıcıdan parolayı değiştirmesini isteyen bir ekran açılır. Şekil 3.3'e bakınız.
3. Kullanıcı, parola için mevcut ve yeni değerleri girer.
4. Kullanıcı yeni parolayı tekrar girer.
5. Kullanıcı "Değiştir" i tıklar.
6. Sistem sağlanan değerleri uygular.

3.2.5 Alternatif akış

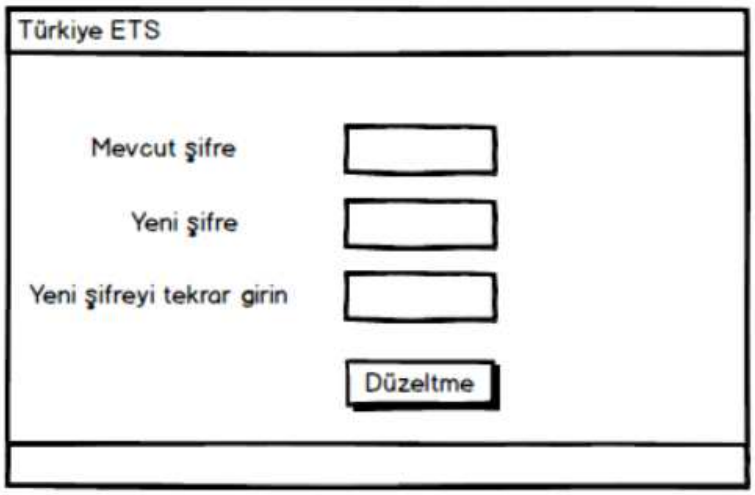
Yok

3.2.6 İstisna akışları

Yok

3.2.7 İlave bilgi

3.2.7.1 Ekranlar



Şekil 3.2-1: Parola değiştir ekranı

3.3 Kullanım senaryosu UCS-10: Hesapları görüntüleme

3.3.1 Kullanım senaryosu kısa açıklaması

Bu kullanım senaryosu, sistemde kimlik doğrulaması yapıldıktan sonra kullanıcıların hesapları nasıl görüntüleyebileceğini açıklar..

3.3.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler, tüm hesapları görüntüleyebilirler
- İşletme Kullanıcıları, yalnızca bağlı oldukları hesapları görüntüleyebilirler

3.3.3 Ön koşullar

3.3.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.3.4 Temel akış

1. Kullanıcı (bunun bir işletme kullanıcısı olduğunu varsayıyoruz) "Hesaplar" menü girişini tıklar.
2. Kullanıcı, kullanıcının bağlı olduğu hesapların bir listesini görüntüler.

Not: Listedeki hesap numarası, §3.4.'te analiz edilen işlevselliğe götüren bir köprüdür.

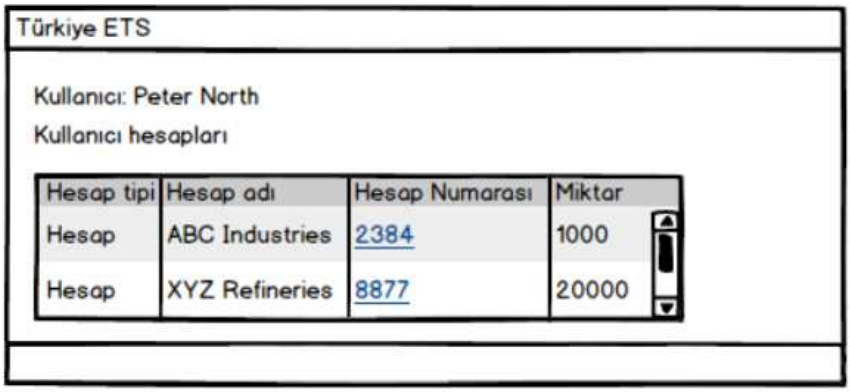
3.3.5 Alternatif akış

3.3.5.1 Yönetici tüm hesapları görüntüleyebilir

1. Yönetici, "Hesaplar" menü girişine tıklar.
2. Kullanıcı, sistemdeki tüm hesapları içeren bir listeyi görüntüler.

3.3.6 İlave bilgi

3.3.6.1 Ekranlar



Şekil 3.3-1: Hesap listesi

3.4 Kullanım senaryosu UCS-15: Hesaplardaki üniteleri ve emisyonları görüntüleme

3.4.1 Kullanım senaryosu kısa açıklaması

Bu kullanım senaryosu, kullanıcıların bir hesap seçtikten sonra hesaplarda tutulan hesapların ve birimlerin emisyonlarını nasıl görebileceklerini açıklamaktadır..

3.4.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler
- İşletme Kullanıcıları

3.4.3 Ön koşullar

3.4.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.4.3.2 Kullanıcı, hesap listesini görüntüler

Kullanıcı, hesapları §3.3.'te tanımlandığı şekilde görüntüler.

3.4.4 Temel akış

1. Kullanıcı, bir hesabın ayrıntılarını tıklar.
2. Hesapta tutulan emisyonları ve birimleri detaylandıran, belirli hesabın varlıklarını içeren bir liste görünür.

3.4.5 Alternatif akış

3.4.5.1 Hesabı etkinleştirme

Yalnızca yöneticiler için:

1. Kullanıcı, "Etkin" adlı onay kutusunu işaretler
2. Kullanıcı "Güncelle" düğmesini tıklar
3. Hesap etkinleştirilir ve bağlı işletme kullanıcıları tarafından erişilebilir hale gelir.

3.4.5.2 Hesabı devre dışı bırakma

Yalnızca yöneticiler için:

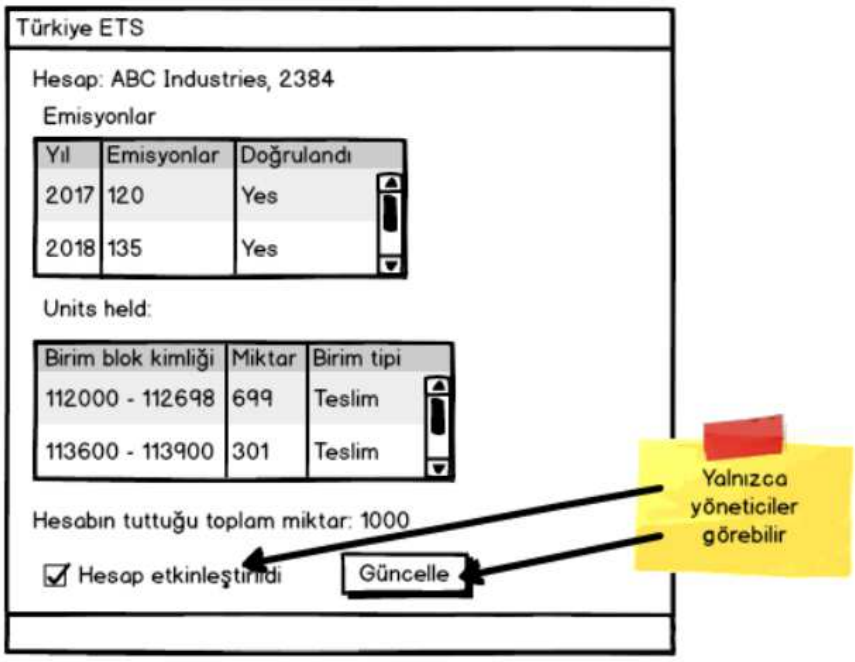
1. Kullanıcı, "Etkin" adlı onay kutusunun işaretini kaldırır
2. Kullanıcı "Güncelle" düğmesini tıklar
3. Hesap devre dışı kalır ve bağlı işletme kullanıcıları tarafından erişilemez.

3.4.6 İstisna akışları

Yok

3.4.7 İlave bilgi

3.4.7.1 Ekranlar



Şekil 3.4-1: Hesap tarafından tutulan birimler

3.5 Kullanım senaryosu UCS-20: İşlemleri görüntüleme

3.5.1 Kullanım senaryosu kısa açıklaması

Bu kullanım senaryosu, kullanıcıların kendi izinlerine göre sistemde gerçekleştirilen işlemleri nasıl görüntüleyebileceklerini açıklamaktadır.

3.5.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler
- İşletme Kullanıcıları

3.5.3 Ön koşullar

3.5.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.5.4 Temel akış

1. Kullanıcı (kullanıcının bir iş kullanıcısı olduğunu varsayıyoruz) "İşlemler" menüsünü tıklar.
2. Sistemde gerçekleştirilen işlemleri içeren bir liste görünür; İşlemler, kullanıcının bağlı olduğu bir hesabı aktarır veya alır.

3.5.5 Alternatif akış

3.5.5.1 Yönetici kullanıcı işlemleri görüntüler

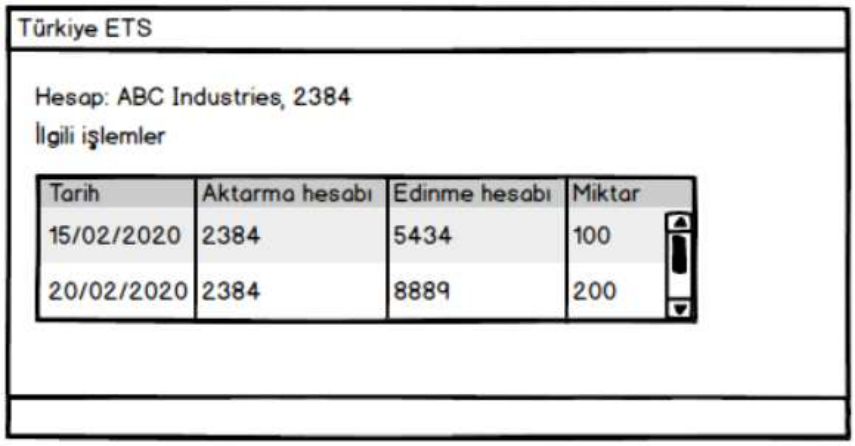
1. Kullanıcı (kullanıcının bir iş kullanıcısı olduğunu varsayıyoruz) "İşlemler" menüsünü tıklar.
2. Sistemde gerçekleştirilen tüm işlemleri içeren bir liste görünür.

3.5.6 İstisna akışları

Yok

3.5.7 İlave bilgi

3.5.7.1 Ekranlar



Şekil 3.5-1: İşlemler ekranını görüntüle

3.6 Kullanım senaryosu UCS-30: İşlemleri gerçekleştirme

3.6.1 Kullanım senaryosu kısa açıklaması

Bu kullanım örneği, kullanıcıların bağlı oldukları hesaplar için işlemleri nasıl gerçekleştirebileceklerini açıklamaktadır.

Alternatif olarak, yöneticiler sistemdeki herhangi bir hesap için işlem gerçekleştirebilir.

Sistemin bölüm §**Error! Reference source not found.**'te tanımlanan işlevselliği sunacağı ve gösterilen işlem türlerini sunacağı varsayılmaktadır

3.6.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır:

- Yöneticiler, bunlar herhangi bir hesaba erişebilir ve herhangi bir işlemi gerçekleştirebilir
- İşletme Kullanıcıları, bunlar yalnızca atandıkları hesaplara erişebilirler

3.6.3 Ön koşullar

3.6.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.6.3.2 Kullanıcı, bir hesabın ayrıntılarını görüntüler

3.6.4 Temel akış

1. Kullanıcı, "İşlemi Gerçekleştir" düğmesini tıklar
2. Zorunlu alanlar olarak sunulan bir ekran görünür:
    - o İşlemin hedef hesabı
    - o İşlem yapılacak birimlerin miktarı
3. Kullanıcı tüm zorunlu alanları girer ve "Gönder" i tıklar. to Şekil 3.6-1'ye bakın.
4. Sistem, işlemin tüm ayrıntılarını tekrarlayan bir onay penceresi sunar.
5. Kullanıcı "Onayla" düğmesine tıklar.
6. İşlem sistem tarafından saklanır ve işlenir

3.6.5 Alternatif akış

Yok

3.6.6 İstisna akışları

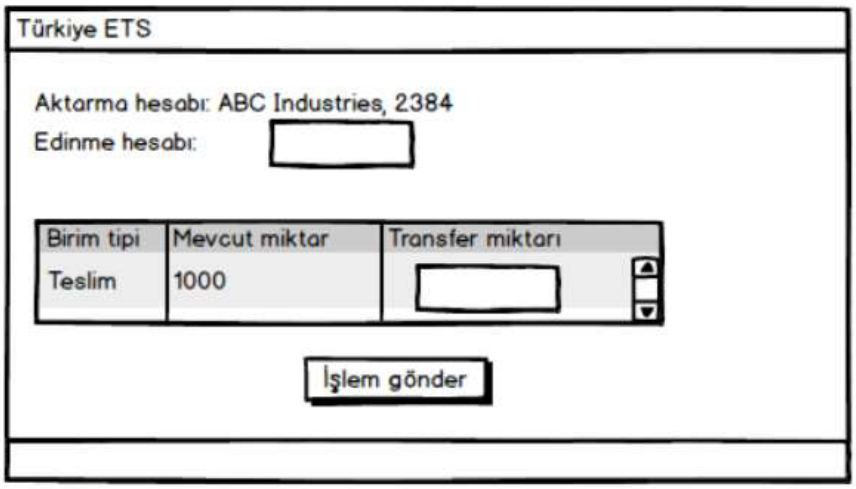
3.6.6.1 Yanlış işlem verisi

Temel akışın 2. adımında:

1. Kullanıcı yanlış veri türleri giriyor (örneğin, miktar alanına sayısal olmayan değer)
2. Hatayı ayrıntılandıran bir uyarı mesajı görünür.
3. Kullanıcı yanlış verileri düzeltir.
4. Temel akış 4. adımla devam eder.

3.6.7 İlave bilgi

3.6.7.1 Ekranlar



Şekil 3.6-1: İşlem gönder ekranı

3.7 Kullanım senaryosu UCS-35: Kullanıcıları arama

3.7.1 Kullanım senaryosu kısa açıklaması

Bu kullanım senaryosu, yöneticilerin kullanıcıları nasıl arayabileceğini açıklar.

3.7.2 Aktörler

Yöneticiler

3.7.3 Ön koşullar

3.7.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.7.4 Temel akış

1. Kullanıcı arama kriterlerini girer
2. Kullanıcı "Ara" düğmesini tıklar
3. Ekran, kriterleri karşılayan kullanıcıların bir listesini sunar

3.7.5 Alternatif akış

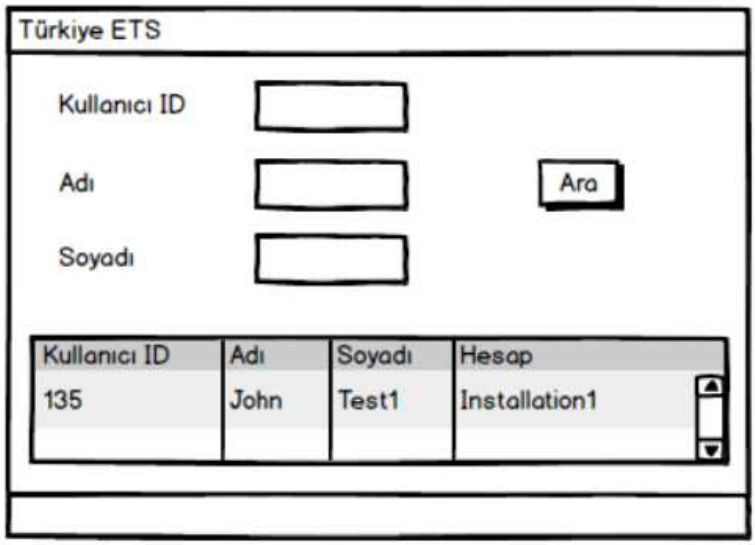
Yok

3.7.6 İstisna akışları

Yok

3.7.7 İlave bilgi

3.7.7.1 Ekranlar



Şekil 3.7-1: Kullanıcıları arama ekranı

3.8 Kullanım senaryosu UCS-40: Kullanıcıları, Kurulumları vb. Getirme

3.8.1 Kullanım senaryosu kısa açıklaması

Bu, bir program komut dosyasının manuel olarak yürütülmesi yoluyla teknik araçlarla ve ekransız olarak gerçekleştirilen teknik bir kullanım senaryosudur.

Bunun bu belgede görünmesinin nedeni, sistemin doğru çalışması için diğer sistemlerden (MRV sistemi gibi) veri alınmasının zorunlu olmasıdır.

3.8.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır::

- Yöneticiler

3.8.3 Ön koşullar

3.8.3.1 MRV sistemine ve TR-ETS sistemlerine veri bağlantıları mevcuttur

3.8.4 Temel akış

1. Kullanıcı komut dosyası MRV'den bir kullanıcı alma sorgusu gerçekleştirir
2. Geri Gönderme, TR-ETS'nin kullanıcı setiyle kontrol edilir.
3. TR-ETS'de bulunmayan herhangi bir kullanıcı TR-ETS'ye dönen

İşlem, aşağıdakiler gibi diğer iş verileri için tekrarlanır:

- tesisler,
- kullanıcıların kurulumlara bağlantıları,
- emisyonlar

3.8.5 Alternatif akış

Yok

3.8.6 İstisna akışları

3.8.6.1 Yanlış veri tipleri

Temel akışın 3. adımında:

1. Eklenecek veri türleri geçerli değilse, ekleme sorgusu başarısız olur.
2. Kullanıcı hatayı bulur.
3. Kullanıcı verileri düzeltir.
4. Kullanıcı TR-ETS'ye veri eklemeyi tekrarlar.

3.9 Kullanım senaryosu UCS-50: Komut dosyası aracılığıyla Düzenleme ve Birim oluşturma

3.9.1 Kullanım senaryosu kısa açıklaması

Bu, teknik araçlarla ve ekransız gerçekleştirilen teknik bir kullanım senaryosudur.

Sistem, ödenek oluşturmak için bir araç sunmasa da, bu, belirtilen miktarda yeni birimleri oluşturan bir veritabanı komut dosyası aracılığıyla mümkündür.

3.9.2 Aktörler

Bu işlevi gerçekleştirebilen kullanıcılar şunlardır::

- Yöneticiler

3.9.3 Ön koşullar

3.9.3.1 Toplam Miktar Hesabı oluşturuldu

Yeni ödenekleri tutan temel idari hesaba Toplam Miktar Hesabı denir. Bu, oluşturulmuş olmalıdır, çünkü komut dosyası bu hesapta yeni ödenekler yaratır.

3.9.4 Temel akış

1. Kullanıcı, 100 yeni tahsisat oluşturmak için veritabanı komut dosyasını yürütür
2. Toplam Miktar Hesabı tarafından tutulan ödenekler 100 artırılır.

3.9.5 Alternatif akış

Yok

3.9.6 İstisna akışları

3.9.6.1 Toplam Miktar Hesabı mevcut değil

Temel akışın 1. adımında:

1. Komut dosyası yürütürken, gerekli hesabın bulunamadığını belirten bir hata görünür.
2. Kullanıcı, Toplam Miktar Hesabını oluşturur.
3. Kullanıcı senaryosu tekrarlanır.

3.10 Kullanım senaryosu UCS-60: Uyumluluk görüntüleme

3.10.1 Kullanım senaryosu kısa açıklaması

Bu kullanım durumu, yöneticilerin sistemde tanımlanan kurulumların uyumluluğunu gözden geçirmesine olanak tanır.

3.10.2 Aktörler

Bu işlevi yalnızca yöneticiler gerçekleştirebilir.

3.10.3 Ön koşullar

3.10.3.1 Kullanıcı sistemde oturum açmıştır

Kullanıcı, §**Error! Reference source not found.**.'detanımlandığı gibi doğrulanmalıdır.

3.10.4 Temel Akış

1. Kullanıcı "Uyumluluk" raporunu seçer
2. Rapor hesaplanır ve sunulur

3.10.5 Alternatif akış

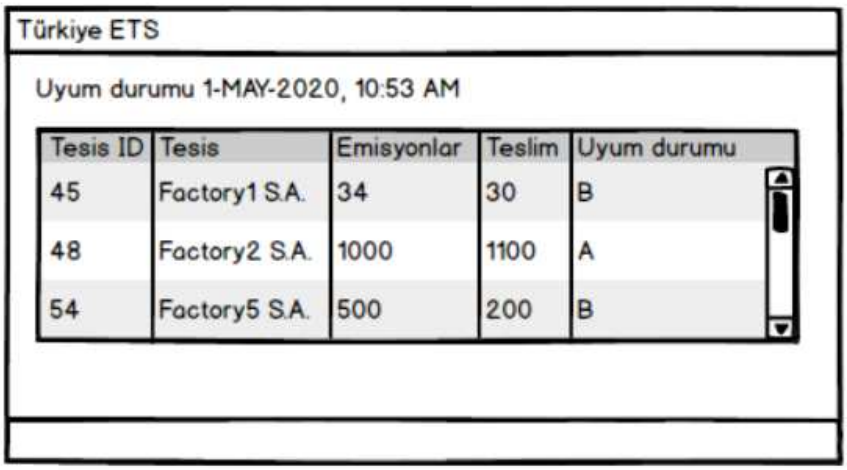
Yok

3.10.6 İstisna akışları

Yok

3.11 İlave bilgi

3.11.1 Raporlar



Şekil 3.11-1: Uyumluluk raporu

4. İZLENEBİLİRLİK MATRİSİ

Aşağıdaki tablo, kullanım senaryolarının işlevsel gereksinimlere uygunluğunu göstermektedir:

	UCS-01: Giriş	UCS-02: Hesapları görüntüleme	UCS-10: Hesapları görüntüleme	UCS-15: Hesaplardaki birimleri ve emisyonları görüntüleme	UCS-20: İşlemleri görüntüleme	UCS-30: İşlemleri gerçekleştirme	UCS-40: Kullanıcıları, Kurulumları vb. Alma	UCS-50: Komut dosyası aracılığıyla Düzenleme ve Birim oluşturma	UCS-60: Uyumluluk görüntüleme
FR1 – Tesisler ve ve Ticaret kurumları		X							
FR2 – Kullanıcılar ve roller							X		
FR3 – Türkiye emisyon tahsisatları				X					

FR4 - Emisyon yükümlülükleri				X					
FR5 – İşlem transferi					X				
FR6 – Teslim işlemi					X	X			
FR7 – Silme işlemi					X	X			
FR8 – İhale işlemi					X	X			
FR9 – Tahsisat işlemi					X	X			
FR10 – Hesaplarda tutulan tahsisatlar				X					
FR11 – İşlem listesi					X				
FR12 – Kullanıcıları arama							X		

FR13 – hesap tipleri			X						
FR14 - Komut dosyası ile birimler oluşturma (yayınlama)								X	
FR15 - Uyumluluk									X
NFR1 – Kimlik doğrulama ve yetkilendirme	X								
NFR2 – Yönetici eişimi	X								
NFR3 – Güvenli erişim	X								
NFR7 – Komut dosyaları aracılığıyla MRV'ye arayüz							X		

Tablo 3: İzlenebilirlik matrisi

Not: Tanımlanan tüm işlevsel olmayan gereksinimler, sistemin tüm kullanım durumları boyunca uygulanır.

5. SİSTEM ANALİZİ VE MİMARİ TASARIM

5.1 Üst Düzey Mimari

5.1.1 Vizyon / Mimari hedefleri

Bu Mimarinin Vizyonu, uluslararası yönergelere ve en iyi uygulamalara uygun olarak tutarlı, eksiksiz ve zamanında işleyen bir Emisyon Ticaret Sistemini oluşturmak ve işletmek için gereken tüm temel hususları kapsamaktır.

Mimari hedefler şunları içerir:

- Önerilen Mimari için güvenlik yönergelerine uyum sağlama ve güvenlik risklerini en aza indirme
- Yazılım geliştirme standartlarına ve en iyi uygulamalara uyum sağlama
- Teslim edilebilir eserler için ölçeklenebilirlik ve sürdürülebilirlik sağlama
- Son kullanıcı için yüksek düzeyde kullanılabilirlik sağlama

5.1.2 Mimari Prensipler

Mimari tasarım sürecini yönlendirmek için kullanılan bir dizi ilke burada listelenmiştir:

1. Endişelerin ayrılması: Bileşenler, aralarında karşılıklı bağımlılığı önlemek ve uygulamanın sürdürülmesine yardımcı olmak için bölünmelidir.
2. Tek Sorumluluk İlkesi: Her bileşenin, kullanıcının sistemi açıkça anlamasını sağlayan belirli bir sorumluluğu olmalıdır. Ayrıca, bileşenin diğer bileşenlerle entegrasyonuna da yardımcı olmalıdır.
3. Basit ve Açık Tut Prensibi: Gereksiz karmaşıklığı ve özelliklerin aşırı mühendisliğini azaltmak için görevleri yerine getiren en basit çözümler daha karmaşık olanlara tercih edilmelidir.
4. Kendinizi Tekrarlamama Prensibi: Bileşenlerin işlevselliği tekrar edilmemelidir. Bir uygulama içindeki işlevselliğin kopyalanması, değişikliklerin uygulanmasını zorlaştırabilir, netliği azaltabilir ve olası tutarsızlıklara neden olabilir.
5. Büyük Tasarımı Önceden Küçültme: Gereksinimleri değiştirme olasılığı varsa, tüm sistem için büyük bir tasarım yapmaktan kaçınılmalıdır.
6. Standart protokollerin ve formatların kullanımı: Çeşitli bileşenler arasındaki bilgi alışverişi, yanlış yorumlamaya yer bırakmayacak, iyi tanımlanmış ve standart protokoller üzerinden gerçekleştirilecektir.

5.1.3 Kısıtlamalar

Önerilen mimarinin riayet etmesi ve dikkate alması gereken bir dizi kısıtlama burada listelenmiştir:

1. ETS uygulamasının tüm çerçevesini ve operasyonel ayrıntılarını belirleyen mevzuat henüz yürürlükte değildir. Bu, belirli mimari kararların ve yönlerin benzer sistemlerdeki uluslararası deneyimlerden türetilen varsayımlara dayanması gerektiği anlamına gelir.
2. Harici uygulamalarla (ör. MRV sistemi, LDAP dizinleri vb.) Doğrudan (entegrasyon yoluyla) veri senkronizasyonu yapılmayacaktır - bunun yerine alternatif yaklaşımlar kullanılacaktır, ör. manuel veri tabanı içe aktarmaları.
3. Mimari ve Tasarım, Türkiye Çevre ve Şehircilik Bakanlığı tarafından sağlanan Standartlara, Yönergelere ve Sınırlamalara uygun olmalıdır. Bunun mümkün olmadığı haller gerekçelendirilecektir.
4. Bakanlık Altyapısına (Uzaktan Erişim, VPN, TeamViewer vb.) Harici erişime, çok katı sınırlamalar ve ön koşullar dışında izin verilmez.

5.1.4 Mimari Kararlar ve Varsayımlar

Burada sunulan, Mimari hedeflerini, ilkelerini ve kısıtlamalarını dikkate alan varsayımların ve kararların bir listesidir.:

5.1.4.1 Program platformu

Karar: Uygulama, Java çalıştırma ortamına ve daha özel olarak Java Standard Edition 8'e dayalı olacaktır. Java ortamı yaygın olarak dağıtılmış kurumsal uygulamalar için kullanılmaktadır. Belirli satıcılarla sıkı bir şekilde bağlantılı değildir. İlgili kitaplıkların ve araçların çoğu açık kaynaklıdır.

Tür: Altyapı

5.1.4.2 Yüklenici altyapısının kullanımı

Karar: Yüklenicinin altyapısı, uygulanan ETS Uygulamasının herhangi bir Geliştirme örneğini barındırmak için kullanılacaktır. Bu şunları çerir:

- Sanal sunucular
- Geliştirme araçları
- DevOps araç zinciri

Gerekli altyapı hakkında daha fazla ayrıntı, Bölüm 5.2.7.'de yer almaktadır.

Tür: Altyapı

5.1.4.3 CSB altyapısının kullanımı

Karar: ÇŞB'nin altyapısı, uygulanan ETS Uygulamasının herhangi bir UAT ve / veya Üretim örneğini barındırmak için kullanılacaktır. Bu içerir:

- Sanal Sunucular
- DevOps araç zinciri
- Gerektiğinde mesajlaşma hizmetleriyle entegrasyon, örn. Posta sunucusuyla
- Ağ üzerinden kullanılabilirlik
- İmzalı sertifikalar

- Gerekli altyapı hakkında daha fazla ayrıntı, Bölüm 5.2.7.'de yer almaktadır.

Tür: Altyapı

5.1.4.4 Mimari desen

Karar: Mimari, her katmanın belirli görevlerden sorumlu olması gereken Katmanlı Mimari yaklaşımını izleyecektir. Kabul edilen yaklaşım, mimarinin inşa edilme şekline esneklik sağlar, örn. diğer katmanlar üzerinde minimum etkiye sahip bir katmanda değişiklikler. Küçükten orta tarafa bir uygulama için, bir Bileşen yaklaşımı (çoğunlukla birçok uygulamada kullanılması gereken yeniden kullanılabilir bileşenleri hedefleyen) veya bir Mikro Hizmetler yaklaşımı yerine Katmanlı bir yaklaşımla gitmek daha uygun görünmektedir.

Tipik katmanlar **Sunum, Hizmet, Kalıcılık, Veri** olacaktır.

Tür: Uygulama Yapısı

5.1.4.5 Ek Modeller: Ön Yüz ve Arka Yüzün Ayrılması

Karar: Mimari bir Model-görünüş-görünüm modeli (MVVM) mimari modelini izleyecektir.

MVVM, grafik kullanıcı ara yüzünün geliştirilmesinin - bir biçimlendirme dili veya GUI kodu yoluyla - iş mantığının veya arka uç mantığının (veri modeli) geliştirilmesinden ayrılmasını kolaylaştırır. MVVM'nin görünüm modeli bir değer dönüştürücüsüdür, yani görünüm modeli, modeldeki veri nesnelerini nesnelerin kolayca yönetilebileceği ve sunulabileceği bir şekilde ortaya çıkarmaktan (dönüştürmek) sorumludur. Bu açıdan, görünüm modeli, görünümden daha modeldir ve görünümün görüntüleme mantığının tamamını olmasa da çoğunu idare eder.

En tipik MVVM uygulamaları Angular'dır.

Tür: Uygulama Yapısı

5.1.4.6 Ek Modeller: RESTful Web Hizmetleri

Karar: Mimari, REST Web Hizmetlerini / API'yi ortaya çıkaracaktır.

Temsili durum aktarımı (REST), Web hizmetleri oluşturmak için kullanılacak bir dizi kısıtlamayı tanımlayan bir yazılım mimari stilidir. RESTful Web hizmetleri adı verilen REST mimari spesifikasyonuna uyan web hizmetleri, İnternet üzerindeki bilgisayar sistemleri arasında birlikte çalışabilirlik sağlar. RESTful Web hizmetleri, talepte bulunan sistemlerin, tek tip ve

önceden tanımlanmış bir vatansız işlemler kümesi kullanarak Web kaynaklarının metinsel temsillerine erişmesine ve bunları değiştirmesine izin verir.

Tür: Uygulama Yapısı

5.1.4.7 Artifakt paketleme yaklaşımı

Karar: Sistemin dağıtım görünümünü, JVM kaynaklarındaki gereksinimleri basitleştirmek ve uygulamanın genel performansını iyileştirmek için uygulamanın tüm katmanlarını tek bir dağıtım yapıtında paketlemek mümkün olacaktır.

Tür: Uygulama Yapısı

5.1.4.8 İlişkisel verilerin depolanması

Karar: Bir RDBMS, İlişkisel verilerin tüm depolanması ve geri alınması için tercih edilen sistemdir.

Tür: Uygulama Yapısı

5.1.4.9 İkili verilerin depolanması

Karar: Dosya Sistemi, ikili verilerin, yani dosyaların tüm depolanması ve geri alınması için tercih edilen sistemdir. Her durumda

- Bu Mimari bağlamında herhangi bir Doküman Yönetim Sistemi çözümüne gerek yoktur
- Performans sorunlarını önlemek için ikili verilerin RDBMS'de depolanmaması şiddetle tavsiye edilir

Tür: Uygulama Yapısı

5.1.4.10 Kullanıcı Arabirimi aracılığıyla İşlevler

Karar: Uygulama, ortam konfigürasyonuna bağlı olarak HTTP veya HTTPS üzerinden bir tarayıcı aracılığıyla erişilebilecek bir Kullanıcı Arayüzü bileşeni aracılığıyla gerekli işlevlerin çoğunu ortaya çıkaracaktır.

Tür: Uygulama Yapısı

5.1.4.11 İdari işlemler aracılığıyla İşlevsellikler

Karar: Uygulama, Yönetici tarafından gerçekleştirilen manuel işlemlerle bazı işlevleri ortaya çıkaracaktır. Örn. komut dosyası yürütme

Tür: Uygulama Yapısı

5.1.4.12 MRV sistemi ile entegrasyon yok

Karar: Mimari, bir MRV (Emisyonların İzlenmesi, Raporlanması ve Doğrulanması) sistemiyle doğrudan entegrasyonun olmadığını varsayacaktır. ETS başvurusu için ihtiyaç duyulan MRV verileri, örneğin:

- Operatör Holding Hesapları ile ilgili veriler
- Emisyonlarla ilgili veriler

Yönetici tarafından gerçekleştirilen manuel bir işlem aracılığıyla ETS'de eşzamansız olarak içe aktarılacaktır. Böyle bir işlemin önerilen biçimi, bir komut dosyasının yürütülmesidir.

Bu nedenle, herhangi bir zamanda, MRV ile ilgili veriler uygulama RDBMS sisteminde saklanacaktır.

Tür: Entegrasyonlar

5.1.4.13 Harici Kullanıcı havuzuyla entegrasyon yok

Karar: Mimari, harici bir Kullanıcı Deposu (Veritabanı, LDAP sistemi) ile doğrudan entegrasyon olmadığını varsayacaktır. ETS uygulaması için gerekli olan böyle bir sistemden alınan veriler, örneğin:

- Kullanıcı bilgileri / kimlik bilgileriyle ilgili veriler
- Rol ile ilgili veriler

Yönetici tarafından gerçekleştirilen manuel bir işlem aracılığıyla ETS'de eşzamansız olarak içe aktarılacaktır. Böyle bir işlemin önerilen biçimi, bir komut dosyasının yürütülmesidir.

Bu nedenle, herhangi bir zamanda, uygulama kullanıcıları ve rol verileri, uygulama RDBMS sisteminde depolanacaktır.

Tür: Entegrasyonlar

5.1.4.14 SSO sistemiyle entegrasyon yok

Karar: Mimari, Kullanıcı Kimlik Doğrulamasının uygulama RDBMS sisteminde depolanan kullanıcıları destekleyeceğini varsayacaktır. Mevcut bir SSO sistemiyle hiçbir entegrasyon bu mimarinin parçası olmayacaktır.

Tür: Entegrasyonlar

Güvenlik hakkında daha fazla ayrıntı, bölüm 5.3.'te yer almaktadır

5.1.4.15 Adlandırma Kuralları

Karar: Bu Mimari bağlamında aşağıdaki adlandırma kuralları izlenecektir.

- Veri tabanı (RDBMS) nesnelerinin adlandırılması için İngilizce dili kullanılacaktır
- İngilizce dili, Uygulama Bileşenlerinin, Modüllerinin, Java sınıflarının, örn.
- Otomatik adlandırma önerilmemektedir, yani nesnelerin İngilizce dilinde bazı anlamlı isimleri olmalıdır
- Hem Veri tabanı nesnelerinin adlandırılmasında hem de Uygulama bileşenlerinde ön eklerin kullanılması önerilir

Tür: Standartlar ve Yönergeler

5.1.4.16 Esnek Yapılandırma

Karar: Uygulama sunucusu ve veri tabanı sunucusu bağlantı işlemleri yapılandırma dosyalarıyla yönetilecektir. Kaynak kodun içinde sabit kodlanmış yapılandırma bulunmamalıdır.

Tür: En İyi Uygulamalar

5.1.4.17 Çok dilli destek

Karar: Uygulama kullanıcı ara yüzü İngilizce ve Türkçe olmak üzere 2 dilde mevcut olacaktır. Ek diller için destek sağlanacaktır.

Tür: Kullanılabilirlik

5.1.4.18 Güvenli manuel yönetici eylemleri

Karar: 5.1.4.11'de belirtildiği gibi, uygulama Yönetici tarafından gerçekleştirilen manuel işlemleri destekler. Bu uygulama bağlamında,

- Yönetici güvenilir bir aktör olarak kabul edilir
- manuel işlemler güvenilir bir ortamda gerçekleştirilmelidir

Tür: Uygulama Yapısı

5.1.5 Büyük Resim

Mimari varsayımlara ve kararlara dayanan Büyük resmi gösteren bir şema aşağıda sunulmuştur:

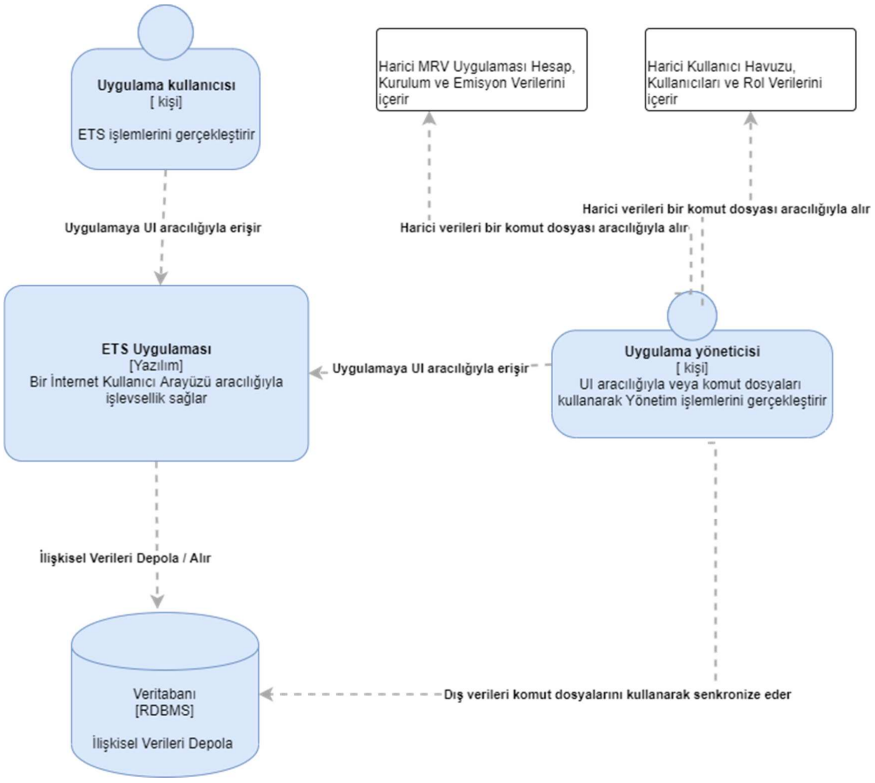


Figure 5.1-1: Büyük Resim

5.2 Mimari Görünümler

Aşağıda bu Mimarinin bir parçası olan ana mimari görünümler listelenmiştir.

5.2.1 Fonsksiyonel Görünümler

Fonksiyonel Görünümü oluşturan Kullanıcı Gereksinimleri bölüm 2.1'de yer almaktadır..1

5.2.2 Fonksiyonel Olmayan Görünümler

Fonksiyonel gereksinimlerin yanı sıra, yazılım mimarisi tarafından bir dizi Fonksiyonel olmayan gereksinim ele alınmalıdır.

Fonksiyonel olmayan gereksinimler bölüm 2.2'de yer almaktadır.

5.2.3 Mantıksal Görünüm

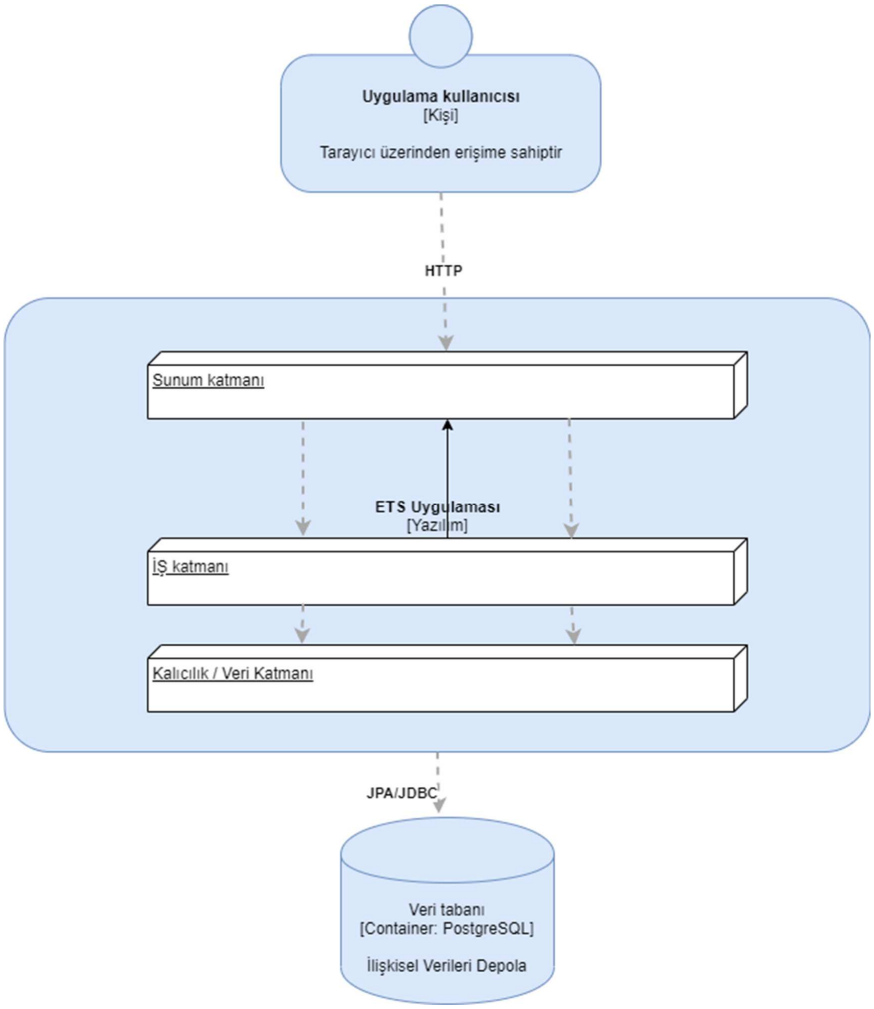
Aşağıdaki bölümler, önerilen çözümü oluşturan ana mimari bileşenlerin yüksek seviyeli bir görünümünü sunmaktadır.

Bölüm 5.1.4'te açıklandığı gibi, Mantıksal model için aşağıdaki varsayımlar geçerlidir:

- Kullanıcılar genellikle bir Web tarayıcısı kullanarak sisteme erişirler
- Sistem verileri bir RDBMS sisteminde saklanır

5.2.3.1 Katmanlı tasarım

Mimari İlkelere uygun olarak, ETS uygulaması tipik bir 3 katmanlı uygulama içeren Katmanlı bir sistem olarak tasarlanmıştır. Bu, aşağıdaki diyagramda sunulmuştur:



Şekil 5.2-1: Katmanlı Mimari diyagramı

Dahil edilen katmanlar:

- Sunum katmanı, uygulamanın en üst katmanıdır ve Kullanıcı Ara yüzünü içerir. Bu katmanın görevi, kullanıcıyla etkileşimi ele almak, kullanıcı girdisini almak, girdi doğrulaması yapmak ve kullanıcıya anlamlı yanıtlar sağlamaktır.
- İş / Mantık katmanı, işlemleri koordine etmekten, Sunum katmanından gelen komutları değerlendirmekten, Verileri işlemekten ve doğrulamaları gerçekleştirmekten, iş mantığını uygulamaktan ve Sunum katmanına uygun yanıtları sağlamaktan sorumludur.
- Veri / kalıcılık katmanı, İş katmanının istediği şekilde RDBMS'deki ilişkisel verileri depolamak ve almaktan sorumludur.

Prensip olarak, veri akışı yalnızca komşu katmanlar arasında desteklenir, örn.

- Kullanıcılar Sunum katmanıyla etkileşimde bulunur
- Sunum katmanı İş katmanıyla etkileşim kurar
- İş katmanı Veri katmanıyla etkileşim kurar
- Veri katmanı RDBMS sistemiyle etkileşime girer

5.2.3.2 Uygulama Bileşenleri

ETS uygulaması, aşağıdaki mantıksal bileşen ve modül gruplarını içerir:

5.2.3.2.1 Ön Yüz Bileşeni

Bu bileşen:

- Sunum katmanını uygular
- Bir Kullanıcı ara yüzünü açığa çıkararak uygulamaya bir giriş noktası sağlar
- MVVM tasarım modeline uygun olarak tasarlanmıştır
- Aşağıdaki mantıksal modülleri içerir:
    - Görünüm modülleri: Html şablonları, CSS ve farklı kullanıcı arabirimi kontrollerini temsil eden komut dosyalarından oluşur
    - Denetleyici modülleri: arka uçtan alınan verileri ve Görünüm modüllerini birbirine yapıştıran denetleyicilerden oluşur. Denetleyici, görünüm modelini başlatır ve görünümün model değişikliklerine nasıl tepki vermesi gerektiğini tanımlar ve bunun tersi de geçerlidir. Denetleyicinin temel sorumluluklarından biri, ön uç doğrulamaları yapmaktır.
    - Ön uç iletişim modülleri: arka uç hizmetleri ile etkileşime izin veren ve Denetleyici modülleri tarafından kullanılabilen bir dizi modül.
    - Güvenlik modülleri: görevi aşağıdaki gibi ön uç güvenlik özelliklerini uygulamak olan farklı modüllerden oluşur:
        - Siteler arası komut dosyasını (XSS) önleme
        - Siteler arası istek sahteciliğini önleme (CSRF veya XSRF)
        - Kullanıcı girdisini temizleme (komut dosyası yerleştirmeyi önlemek için)

5.2.3.2.2 Arka yüz bileşeni

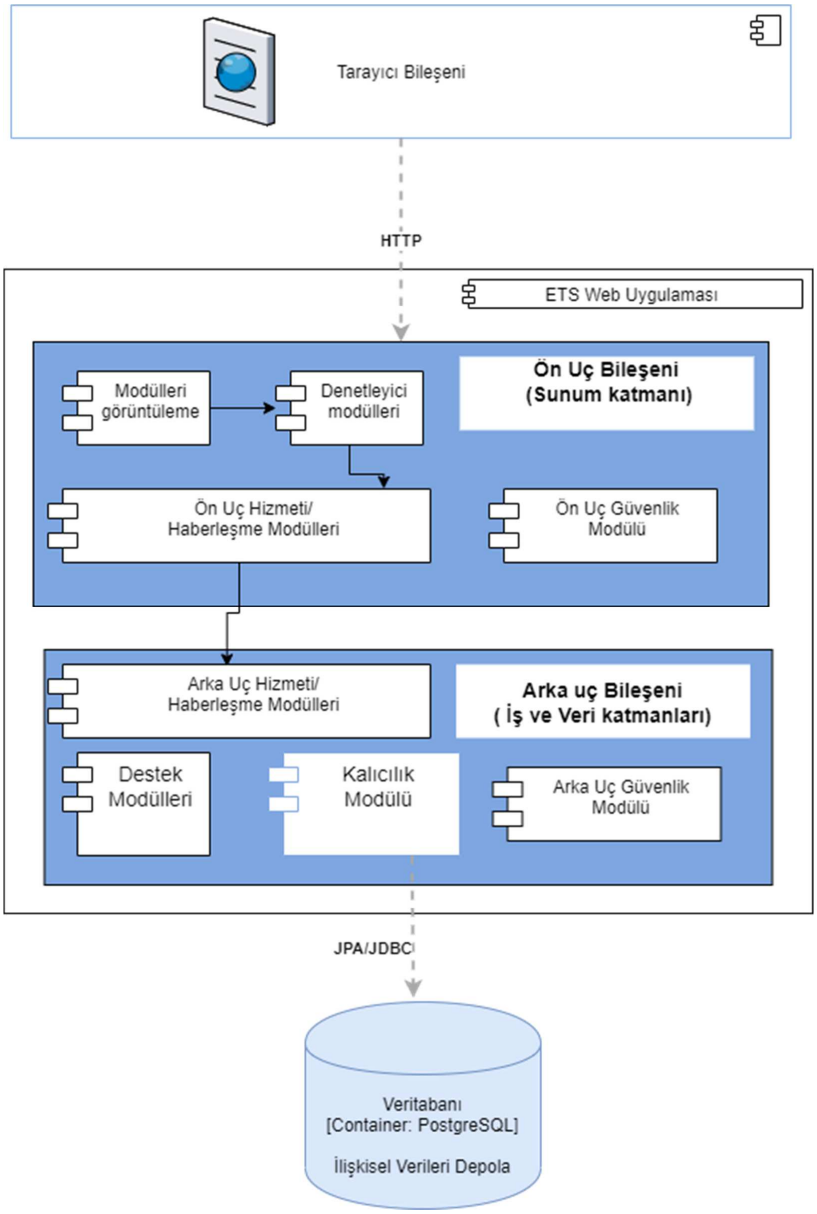
Bu bileşen:

- İş ve Kalıcılık katmanlarını uygular
- Ön uç bileşeninden kullanılabilen bir RESTFul Web Hizmetleri API'sini ortaya çıkarır
- RDBMS'de depolanan verilere özel erişime sahiptir
- Aşağıdaki mantıksal modülleri içerir:
    - Arka uç iletişim modülleri: Ön uç bileşeninden çağrılabilen bir dizi RESTful Web hizmeti uç noktası ve işleminden oluşur
    - Destek modülleri: Gereksinimlere göre yararlı destekleme işlemleri, doğrulamalar, veri değerlendirmeleri / hesaplamaları uygulayan bir dizi modül
    - Kalıcılık modülü: RDBMS ile verilerin depolanması ve geri alınması için soyut bir şekilde arabirim oluşturmaktan sorumlu bir dizi modül. Bu modül, veri bütünlüğünü ve RDBMS ile birlikte çalışabilirliği sağlar

- Güvenlik modülleri: görevi aşağıdakiler gibi arka uç güvenlik özelliklerini uygulamak olan farklı modüllerden oluşur:
  - Kimliği doğrulanmamış kullanıcıların sisteme erişimini önleme
  - Kaynaklara yetkisiz erişimi önleme
  - SQL yerleştirme ve Verilere maruz kalma gibi güvenlik tehditlerini yönetme

5.2.3.2.3 Bileşen Şeması

Yukarıda listelenen bileşenler ve modüller aşağıdaki şemada sunulmuştur



Şekil 5.2-2: Mantıksal bileşen diyagramı

5.2.3.3 Çevrimdışı Bileşenler / Komut Dosyaları

Bölüm 5.1.4.11'de belirtildiği gibi, uygulama Yönetici tarafından gerçekleştirilen manuel işlemleri destekler.

Bu nedenle ETS uygulaması, bir dizi çevrimdışı bileşen içerir.

- Katmanlı mimarinin parçası değildir
- Güvenli bir ortamda yönetici kullanıcılar (Veri tabanı yöneticisi, Destek yöneticileri) tarafından belirli iş işlevlerini uygulamak için sıkı bir şekilde kullanılırlar
- Doğrudan veri tabanı işlemlerini, komut dosyası yürütmelerini vb. Dahil edin.

Bu işlemler, uygulamanın Güvenlik modeline uymaz ve bu nedenle izlenebilirlik ve hesap verebilirlik özelliklerini desteklemez.

5.2.4 Teknoloji seçimi

Bu bölüm, bu Mimari bağlamında kullanılacak teknolojik seçimleri içerir.

5.2.4.1 İlişkisel Veri tabanı Yönetim Sistemi (RDBMS)

ETS Uygulaması için seçilen veri deposu PostgreSQL'dir

PostgreSQL, en karmaşık veri iş yüklerini güvenli bir şekilde depolayan ve ölçeklendiren birçok özellikle birlikte SQL dilini kullanan ve genişleten güçlü, açık kaynaklı bir nesne ilişkisel veritabanı sistemidir.

PostgreSQL kanıtlanmış bir mimariye, güvenilirliğe, veri bütünlüğüne, sağlam özellik setine, genişletilebilirliğe ve sürekli olarak yüksek performanslı ve yenilikçi çözümler sunmak için yazılımın arkasındaki açık kaynak topluluğunun bağlılığına sahiptir. PostgreSQL:

1. tüm büyük işletim sistemlerinde çalışır,
2. 2001'den beri ACID uyumludur,
3. ücretsiz ve açık kaynaklıdır,
4. yüksek oranda genişletilebilir

5.2.4.2 Java Program Ortamı

Uygulamanın seçilen Runtime ortamı OpenJDK 8'dir.

JDK 8, Java Topluluk Sürecinde JSR 284 tarafından belirtildiği şekilde Java SE Platformunun 8. sürümünün açık kaynaklı referans uygulamasıdır.

5.2.4.3 Uygulama Çerçevesi

Seçilen Uygulama Çerçevesi, Spring Framework 5.1 ile Spring Boot 2.1'dir

Spring Framework, modern Java tabanlı kurumsal uygulamalar için her tür dağıtım platformunda kapsamlı bir programlama ve yapılandırma modeli sağlar.

Spring Boot, Java geliştiricilerine otomatik olarak yapılandırılabilen, üretim düzeyinde bir Spring uygulamasıyla başlamaları için bir platform sağlayan açık kaynaklı bir mikro çerçevedir. Özellikler şunları içerir::

- Bağımsız spiring uygulamaları oluşturma
- Undertow'u doğrudan gömülü yapma (WAR dosyalarını dağıtmaya gerek yoktur)
- Spring ve 3. taraf kitaplıklarını mümkün olduğunda otomatik olarak yapılandırma
- Metrikler, sağlık kontrolleri ve harici konfigürasyon gibi üretime hazır özellikler sağlama
- Kesinlikle kod üretilmez ve XML yapılandırması gerekmez

5.2.4.4 Uygulama/Web Sunucu

Seçilen Uygulama / Web sunucusu, Spring Framework 5.1 ile Spring Boot 2.1'e gömülü olan Undertow'dur.

Undertow hakkında bazı ayrıntılar:

- NIO tabanlı hem engelleyen hem de engellemeyen API'ler sağlayan java ile yazılmış esnek performanslı bir web sunucusudur.
- Küçük tek amaçlı işleyicileri birleştirerek bir web sunucusu oluşturmanıza izin veren kompozisyon tabanlı bir mimariye sahiptir. Bu size tam bir Java EE servlet 4.0 konteyneri veya düşük seviyeli engellemeyen bir işleyici arasında seçim yapma esnekliği sağlar.
- Kullanımı kolay akıcı oluşturucu API'leri ile tamamen yerleştirilebilir olacak şekilde tasarlanmıştır. Undertow'un yaşam döngüsü tamamen yerleştirme uygulaması tarafından kontrol edilir.
- JBoss sponsorluğundadır ve Wildfly Uygulama Sunucusundaki varsayılan web sunucusudur.

Not: Spring Boot, 3 gömülebilir web sunucusunu destekler: Undertow, Tomcat ve Jetty. Bu seçenek, uygulama üzerinde genel bir etki olmaksızın bu Mimari bağlamında değiştirilebilir.

5.2.4.5 Nesne-İlişkisel Haritalama (ORM) Çerçevesi

Seçilen ORM, Spring Data JPA'dır.

Daha büyük Spring Data ailesinin bir parçası olan Spring Data JPA, JPA tabanlı depoların kolayca uygulanmasını kolaylaştırır. Bu modül, JPA tabanlı veri erişim katmanları için gelişmiş destekle ilgilenir. Veri erişim teknolojilerini kullanan Spring destekli uygulamalar oluşturmayı kolaylaştırır.

5.2.4.6 Veritabanı Sürümü Kontrol Çerçevesi

Seçilen DB SCM Liquibase'dir.

Liquibase, veritabanı şema değişikliklerini izlemek, yönetmek ve uygulamak için açık kaynaklı, veritabanından bağımsız bir kitaplıktır. Scriptler farklı formatlarda (XML, JSON, SQL) tanımlanabilir ve Java, Maven, Ant ile çalıştırılabilir. .

5.2.4.7 Birim Testi

Seçilen birim test teknolojisi JUnit çerçevesidir.

JUnit, bir endüstri lideridir, herhangi bir Java oluşturma aracıyla entegre olur ve çok sayıda kod kalitesi aracı içinde kullanılabilir.

5.2.4.8 İstemci tarafı teknolojiler

Ön Uç Kullanıcı Arayüzü için kullanılan seçili istemci tarafı teknolojileri HTML5, Javascript ve CSS'dir.

HTML5, World Wide Web'de içerik yapılandırmak ve sunmak için kullanılan bir biçimlendirme dilidir. HTML5, bir World Wide Web Consortium (W3C) önerisi olan HTML'nin beşinci ve son ana sürümüydür. Mevcut spesifikasyon HTML Yaşam Standardı olarak bilinir ve başlıca tarayıcı satıcıları (Apple, Google, Mozilla ve Microsoft), Web Köprü Metni Uygulama Teknolojisi Çalışma Grubu (WHATWG) konsorsiyumu tarafından sürdürülür.

JavaScript, World Wide Web'in temel teknolojilerinden biridir. [8] JavaScript, etkileşimli web sayfalarını etkinleştirir ve web uygulamalarının önemli bir parçasıdır.

CSS3, Basamaklı Stil Sayfaları dilinin en son evrimidir ve CSS2'yi genişletmeyi amaçlamaktadır. Yuvarlatılmış köşeler, gölgeler, gradyanlar, geçişler veya animasyonlar gibi birçok yeni özellik ve eklemenin yanı sıra çoklu sütunlar, esnek kutu veya ızgara düzenleri gibi yeni düzenler getirmektedir.

5.2.4.9 MVVM Çerçevesi

Ön uç UI için kullanılan seçili MVVM çerçevesi Açısaldır.

Angular, HTML ve TypeScript kullanarak tek sayfalı istemci uygulamaları oluşturmak için bir platform ve çerçevedir. Angular, TypeScript ile yazılmıştır. Uygulamalarınıza aktardığınız bir dizi TypeScript kitaplığı olarak temel ve isteğe bağlı işlevleri uygular.

Bir Angular uygulamasının mimarisi belirli temel kavramlara dayanır. Temel yapı taşları, bileşenler için bir derleme bağlamı sağlayan NgModules'dir. NgModules, ilgili kodu işlevsel kümeler halinde toplar; bir Angular uygulaması, bir dizi NgModül tarafından tanımlanır. Bir uygulamanın her zaman önyüklemeyi etkinleştiren en az bir kök modülü vardır ve genellikle daha fazla özellik modülüne sahiptir.

- Bileşenler, Angular'ın program mantığınıza ve verilerinize göre seçebileceği ve değiştirebileceği ekran öğeleri kümeleri olan görünümleri tanımlar.
- Bileşenler, görünümlerle doğrudan ilgili olmayan belirli işlevler sağlayan hizmetleri kullanır. Hizmet sağlayıcılar, bileşenlere bağımlılıklar olarak eklenebilir ve kodunuzu modüler, yeniden kullanılabilir ve verimli hale getirir.

Hem bileşenler hem de hizmetler, türlerini işaretleyen ve Angular'a bunları nasıl kullanacağını söyleyen meta veriler sağlayan dekoratörlere sahip basit sınıflardır.

- Bir bileşen sınıfının meta verileri, onu bir görünümü tanımlayan bir şablonla ilişkilendirir. Bir şablon, sıradan HTML'yi Angular yönergeleri ve bağlantı biçimlendirmesiyle birleştirerek, Angular'ın HTML'yi görüntüleme için oluşturmadan önce değiştirmesine olanak tanır.
- Bir hizmet sınıfının meta verileri, Angular'ın bağımlılık ekleme (DI) yoluyla bileşenlerin kullanımına sunması için ihtiyaç duyduğu bilgileri sağlar.

Bir uygulamanın bileşenleri genellikle hiyerarşik olarak düzenlenmiş birçok görünümü tanımlar. Angular, görünümler arasında gezinme yollarını tanımlamanıza yardımcı olmak için Yönlendirici hizmetini sağlar. Yönlendirici, gelişmiş tarayıcı içi gezinme yetenekleri sağlar.

5.2.4.10 Derleme Araçları

Java uygulaması için seçilen derleme aracı Maven'dir.

Apache Maven, bir yazılım proje yönetimi ve anlama aracıdır. Bir proje nesne modeli (POM) kavramına dayalı olarak Maven, bir projenin yapısını, raporlamasını ve dokümantasyonunu merkezi bir bilgi parçasından yönetebilir.

Ön uç için seçilen derleme aracı Webpack'tir.

Webpack, modern JavaScript uygulamaları için statik bir modül paketleyicidir. Webpack uygulamanızı işlerken dahili olarak projenizin ihtiyaç duyduğu her modülü eşleştiren ve bir veya daha fazla paket oluşturan bir bağımlılık grafiği oluşturur..

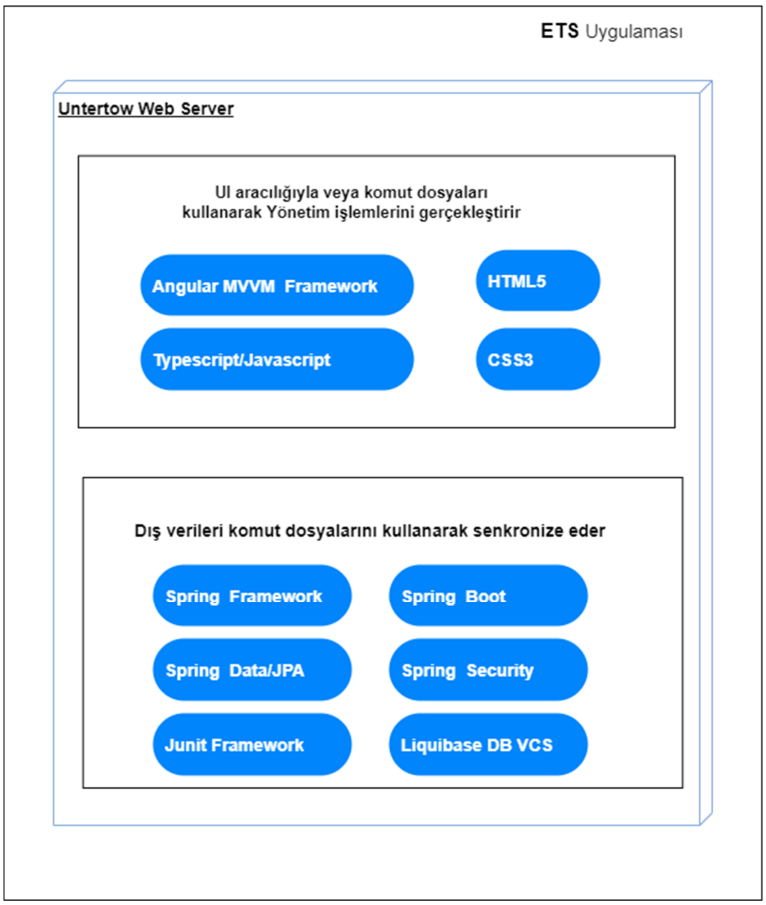
5.2.4.11 Geliştirme platformu

Geliştirmeyi önyüklemek için kullanılan seçilen geliştirme platformu JHipster'dır, ancak sürekli geliştirme için gerekli değildir.

JHipster, Spring Boot + Angular / React / Vue Web uygulamalarını ve Spring mikro hizmetlerini başlatmak, oluşturmak, geliştirmek ve dağıtmak için kullanılabilen bir geliştirme platformudur..

5.2.4.12 Teknoloji diagramı

Bu Mimari bağlamında kullanılacak en önemli teknolojiler aşağıdaki diyagramda gösterilmektedir.



Şekil 5.2-3: Teknoloji diagramı

5.2.4.13 Teknoloji referansı

Aşağıda, bahsedilen teknolojilere yönelik yararlı bağlantılar ve referanslar içeren bir liste bulunmaktadır.

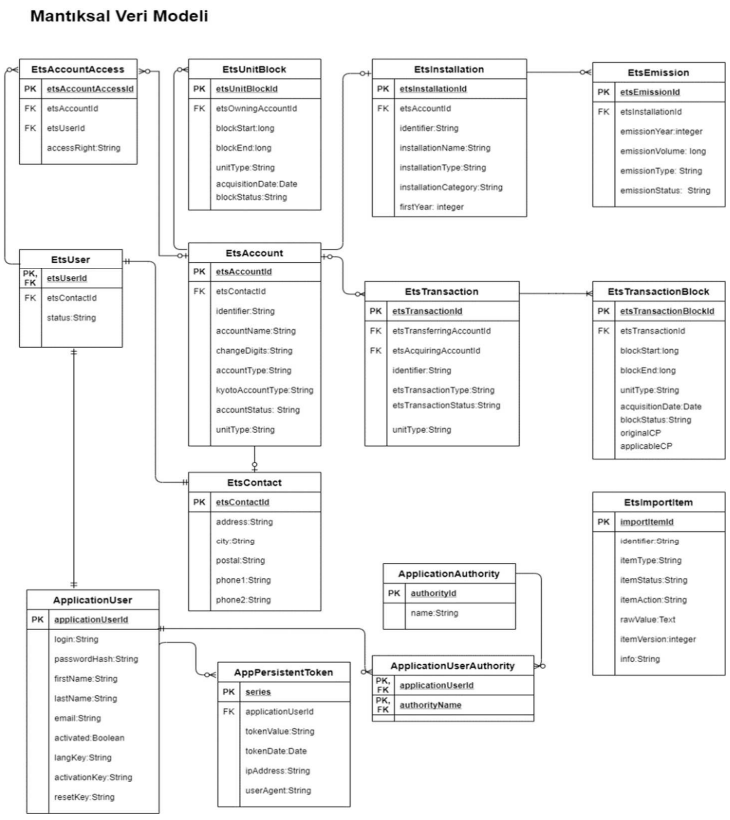
- https://openjdk.java.net/
- https://spring.io/projects/spring-boot
- https://spring.io/projects/spring-data
- https://spring.io/projects/spring-framework
- http://undertow.io/
- https://angular.io/
- https://www.liquibase.org/
- https://junit.org/junit5/
- https://maven.apache.org/
- https://webpack.js.org/
- https://www.jhipster.tech/

5.2.5 Veri Görünümü

Veri Görünümü, Etki Alanı Modellerini uygulamak için gerekli veri tabanı varlıklarının yanı sıra ETS Uygulamasının uygun şekilde çalışması için gerekli diğer verilerin depolanması hakkında ayrıntılar sunar.

5.2.5.1 Varlık-İlişki Şeması

Aşağıdaki Mantıksal veri diyagramı, ilişkileri dahil olmak üzere en önemli Ticari ETS varlıklarını özetler.



Şekil 5.2-4: Varlık ilişki diyagramı

5.2.5.2 Ticari İşletmelerin Tanımı

Aşağıdaki bölümler, ana varlıklar veya varlık grupları hakkında kısa bir açıklama sağlar..

5.2.5.2.1 ApplicationUser

Bu varlık, genel bir sistem aktörünü temsil eder ve esas olarak kimlik doğrulama amacıyla, yani kimlik bilgileri ve tanımlama amacıyla kullanılan bilgileri tutar.

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
login	String	Giriş için kullanılan kullanıcı adı
passwordHash	String	Şifrelenmiş Parola - bcrypt ile kodlanmıştır
Email	String	Email adresi
Activated	boolean	kullanıcının etkin olup olmamasına bağlı olarak doğru / yanlış
firstName	String	
lastName	String	
langKey	String	Kullanıcı için seçilen dil (Türkçe İngilizce), geçerli değerler "en" ve "tr" dir (ISO 639-1)
activationKey	String	Kullanıcının aktivasyonu için kullanılacak tek seferlik anahtar (Mevcut uygulamada kullanılmamaktadır)
resetKey	String	Kullanıcının şifre sıfırlamasını kullanmak için tek seferlik anahtar

5.2.5.2.2 ApplicationAuthority

Bu varlık genel bir sistem Rolünü temsil eder

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
name	String	Rolün adı. Mevcut roller şunları içerir: • ADMIN • ETS_ADMIN • USER

5.2.5.2.3 ApplicationUserAuthority

Bu varlık, bir Kullanıcı ve Rol arasındaki bir ilişkiyi temsil eder. Bir ApplicationUser girişi birden çok ApplicationAuthority girişine bağlanabilir.

5.2.5.2.4 EtsUser

Bu varlık, bir tescil Kullanıcısını temsil eder ve ETS işlemlerine özgü bilgileri tutar (örneğin, kimlik bilgilerine sahip değildir). Her giriş ile bağlantılıdır:

- tam olarak bir ApplicationUser girişi
- bir EtsContact girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
status	String	ETS sistemi içindeki kullanıcının durumunu gösteren bir durum. Durumlar arasında AKTİF ve AKTİF DEĞİL

5.2.5.2.5 EtsContact

Bu varlık, iletişim bilgilerinin bir girişini temsil eder ve diğer varlıklara (EtsUser, EtsAccount) bağlanabilir

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
address	String	
city	String	
postal	String	
Phone1	String	
Phone2	String	

5.2.5.2.6 EtsAccountAccess

Bu varlık, bir EtsUser girişini, belirli bir kullanıcının belirtilen hesaba erişim haklarına sahip olduğunu belirten bir EtsAccount ile ilişkilendirir.

Table columns

Kolon adı	Kolon tipi	Açıklama
accessRight	String	Erişim hakkı değerleri şunları içerir: OKUMA, YÖNET

5.2.5.2.7 EtsAccount

Bu varlık, işlem gerçekleştirebilen ve ödenekleri olan ana varlık olan bir Hesabı temsil eder. Her giriş ile bağlantılıdır

- bir EtsContact girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
identifier	String	Benzersiz tanımlayıcı, biçim aşağıdaki gibidir: TR-100-XXXXXXXX-0 Açıklama · TR, Türkiye'nin ISO 3166'daki iki harfli ülke kodudur. · 100, Holding hesabı Kyoto kodudur. Kyoto Protokolü ile gelecekteki entegrasyonu kolaylaştırmak için Türkiye sicilindeki tüm hesapların hesap sahibi olduğu varsayılacaktır. · XXXXXXXX, bir veritabanı dizisi tarafından üretilen kayıt defterindeki benzersiz bir sayısal değerdir. · 0, holding hesapları için sıfır Kyoto Protokolü taahhüt süresidir. Türkiye sicilindeki tüm cari hesaplar sıfır periyotlu olacaktır.
changeDigits	String	Gizli rakamlar, format şu şekildedir: XX Ve önceki alanlara göre hesaplanan iki basamak içerir. Hesap tanımlayıcısının geçerliliğini doğrularlar
accountName	String	Otomatik oluşturulan hesabın adı

accountType	String	Bölüm 2.3'te listelenen değerler aşağıdaki gibi kodlanmıştır: • OPERATOR_HOLDING("7"), • TRADING("12"), • AUCTION_DELIVERY("13"), • TOTAL_QUANTITY("16"), • DELETION("5")
accountStatus	String	Durumlar arasında AKTİF (1) ve AKTİF DEĞİL (0) bulunur
unitType	String	Ödenek birimi türü. Varsayılan "TRUNIT" dir
kyotoAccountType	String	HOLDING_ACCOUNT ("100") varsayılanı

5.2.5.2.8 EtsInstallation

Bu varlık bir Kurulumu temsil eder, ör. Bir Hesap tarafından yönetilen bir Fabrika veya Organizasyon. Her giriş ile bağlantılıdır

- hesap türünün OPERATOR_HOLDING olduğu bir EtsAccount girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
identifier	long	Bu kurulum için MRV sisteminden dışa aktarılan benzersiz tanımlayıcı
installationName	String	Fabrikanın adı. Bu bilgi, MRV sisteminden gelir
installationType	String	Değerler şunları içerebilir Çimento Cam etc Bu bilgi, MRV sisteminden gelir
InstallationCategory	String	Değerler, emisyon seviyesini gösteren A, B, C içerebilir. Bu bilgi, MRV sisteminden gelir
firstYear	integer	Emisyonların ilk yılı

5.2.5.2.9 EtsTransaction

Bu varlık, bir hesaptan diğerine ödenek birimlerinin transferini içeren iki hesap arasındaki bir işlemi temsil eder. Her giriş ile bağlantılıdır

- Aktaran hesap görevi gören bir EtsAccount girişi
- Alıcı hesap görevi gören başka bir EtsAccount girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
identifier	String	Biçimle birlikte otomatik oluşturulan benzersiz tanımlayıcı TRXXXX XXXX bir sıra numarasıdır
transactionType	String	Bölüm 2.3'te listelenen değerler
transactionStatus	String	Değerler arasında BEKLEMEDE, TAMAMLANDI olabilir. Tüm işlemler için varsayılan değer TAMAMLANDI
unitType	String	Ödenek birimi türü. Varsayılan "TRUNIT" dir

5.2.5.2.10 EtsUnitBlock

Bu varlık bir ödenek birimleri bloğunu temsil eder. Tahsisat birimleri tipik olarak seri numaraları atanır ve bloklar halinde gruplandırılır. Her giriş ile bağlantılıdır

- bir EtsAccount girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
blockStart	long	Bloktaki ilk ünitenin seri numarası
blockEnd	long	Bloktaki son birimin seri numarası

unitType	String	Ödenek birimi türü. Varsayılan "TRUNIT" dir
blockStatus	String	Bloğun mevcut olup olmadığını veya bir işleme dahil olup olmadığını gösterir. Varsayılan değer "KULLANILABİLİR" dir
acquisitionDate	Date	Bloğun edinildiği tarih

5.2.5.2.11 EtsTransactionBlock

Bu varlık, bir işleme katılan bir tahsisat birimleri bloğunu temsil eder. Tahsisat birimleri tipik olarak seri numaraları atanır ve bloklar halinde gruplandırılır. Her giriş ile bağlantılıdır

- • bir EtsTransaction girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
blockStart	long	Bloktaki ilk ünitenin seri numarası
blockEnd	long	Bloktaki son birimin seri numarası
unitType	String	Ödenek birimi türü. Varsayılan "TRUNIT" dir
blockStatus	String	Bloğun mevcut olup olmadığını veya bir işleme dahil olup olmadığını gösterir. Varsayılan değer "KULLANILABİLİR" dir
acquisitionDate	Date	Bloğun edinildiği tarih

5.2.5.2.12 EtsEmission

Bu varlık, belirli Tesis için yıllık emisyon bilgilerini tutmak için kullanılır. Bu bilgiler MRV sisteminden gelir. Her giriş ile bağlantılıdır

- bir EtsInstallation girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama

emissionYear	integer	Bu hesap için spesifik emisyon girişi ile ilgili yıl
emissionVolume	long	Emisyon miktarı
emissionType	String	Emisyon türleri. Varsayılan CO2'dir
emissionStatus	String	Bir durum. Varsayılan "DOĞRULANMIŞ" tır

5.2.5.2.13 EtsImportItem

Bu varlık, kayıt defterinde harici sistemlerden, yani İRD sisteminden içe aktarılan diğer varlıklar hakkındaki bilgileri tutmak için kullanılır. İçe aktarma, CSV dosyaları kullanılarak gerçekleştirilir, bu nedenle EtsImportItem'deki her giriş bir CSV satırını temsil eder. Desteklenen içe aktarılan varlık türleri şunlardır:

- KULLANICI türü: bir ETS Kullanıcısını temsil eder (ApplicationUser ve EtsUser varlıkları ile ilgili)
- KURULUM türü: bir Operatör Holding Hesabını temsil eder (EtsInstallation & EtsAccount varlıkları ile ilgili)
- EMİSYON tipi: bir Kurulum Hesabı için emisyon bilgilerini temsil eder (EtsEmission varlığı ile ilgili)
- OPERATÖR tipi: bir ETS Kullanıcısını Operatör olarak bir Operatör Holding Hesabı'na bağlama (EtsAccountAccess varlığı ile ilgili)

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
identifier	String	İçe aktarılan varlık için benzersiz bir tanımlayıcı. Varlığın türüne göre farklılık gösterir: • KULLANICI tipi: kullanılan tanımlayıcı <kullanıcı girişi> • KURULUM tipi: kullanılan tanımlayıcı, MRV sisteminden gelen <benzersiz kurulum kimliği> dir • EMİSYON: kullanılan kompozit tanımlayıcı <benzersiz kurulum kimliği> _ <emisyon yılı> • OPERATÖR tipi: kullanılan bileşik tanımlayıcı <benzersiz kurulum kimliği> _ <kullanıcı oturum açma>

itemType	String	İçe aktarılan varlığın türü: • USER • INSTALLATION • EMISSION • OPERATOR
itemStatus	String	İçe aktarılan varlığın durumu: • BEKLEMEDE: içe aktarma gerçekleştirilmemişse • ETKİN: içe aktarma gerçekleştirildiyse • HATA: içe aktarma bir hatayla sonuçlandıysa
itemAction	String	İçe aktarma işlemi: • İçe aktarılan varlık yeniyse EKLE • İçe aktarılan varlık zaten mevcutsa GÜNCELLE
rawValue	Text	JSON biçiminde içe aktarılan varlık için tüm verileri içeren ham değer
itemVersion	integer	Sürüm numarası - ADD eylemleri için varsayılan olarak 1'dir ve belirli varlık üzerindeki ardışık her GÜNCELLEME eylemi için artırılır
info	String	İçe aktarma işlemiyle ilgili bilgiler, ör. bir hata varsa hata mesajı

5.2.5.2.14 AppPersistentToken

Bu varlık, Kalıcı oturum açma tanımlama bilgileri ("Beni hatırla" özelliği) ile ilgili bilgileri tutmak için kullanılır. Temel mekanizma, Spring RememberMe hizmetlerine dayanmaktadır. Her giriş ile bağlantılıdır

- bir ApplicationUser girişi

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
series	String	Kullanıcı için benzersiz seri tanımlayıcı. Bu, kullanıcının ilk oturum açışını tanımlar ve kullanıcı otomatik olarak her oturum açtığında sabit kalır.

token_value	String	Rastgele alfanümerik belirteç
token_date	Date	Çerezin oluşturulduğu tarih. Bu aynı zamanda son kullanma tarihini oluşturmak için de kullanılır.
ip_address	String	Giriş için kullanılan IP adresi
user_agent	String	Giriş için kullanılan tarayıcı sürümü hakkında bilgi

5.2.5.3 Auditing-related Entities and Model

Denetim bilgilerini depolamak ve Denetim işlevlerini desteklemek için ticari varlıklara yatay olarak ek sütunlar eklenir. Bu sütunlar:

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
createdBy	String	Varlığı oluşturan kullanıcı
createDate	Date	Bu varlığın oluşturulduğu tarih
lastModifiedBy	String	Varlığı en son değiştiren kullanıcı
lastModifiedDate	Date	Bu varlığın son değiştirildiği tarih

Aşağıdaki varlıklar da Denetim amaçları için kullanılır

5.2.5.3.1 EntityAuditEvent

Bu varlık, denetim amacıyla tüm Ticari varlıklarda yapılan değişikliklerin geçmişini tutmak için kullanılır. Depolanan bilgiler şunları içerir:

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
entity_type	String	Değiştirilen varlık için tam java sınıfı adı, ör. EtsTransaction için

		tr.gov.moeu.ets.domain.EtsTransaction
entity_id	long	Değiştirilen varlığın birincil anahtarı
action	String	Değişikliğin türü, yani CREATE veya UPDATE
entity_value	Text	JSON biçiminde değiştirilen varlığın verileri
commit_version		Bir sürüm numarası - CREATE eylemleri için varsayılan olarak 1'dir ve belirli varlık üzerindeki her ardışık GÜNCELLEME eylemi için artırılır
modified_by	String	Spesifik değişikliği yapan kullanıcı
modified_date	Date	Spesifik değişikliğin tarihi

Etkilenen varlıklar:

- EtsTransaction
- EtsInstallation
- EtsEmission
- EtsUser
- EtsImportItem
- EtsTransactionBlock
- EtsUnitBlock
- EtsContact
- EtsAccount
- User

5.2.5.3.2 PersistentAuditEvent

Bu varlık, uygulamaya yönelik güvenlikle ilgili olayların geçmişini tutmak için kullanılır ve Springboot aktüatörünün Denetim mekanizmasına göre modellenir. Depolanan bilgiler şunları içerir:

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
principal	String	Olayla ilgili kullanıcı
Event_date	Date	Etkinlik tarihi
Event_type	String	Spring boot actuator Audit mekanizmasından döndürülen Audit olayının türü, ör.

		AUTHENTICATION_FAILURE, AUTHENTICATION_SUCCESS etc

5.2.5.3.3 PersistentAuditEventData

Bu varlık, güvenlikle ilgili olaylar için öznitelik ad-değer çiftleri olarak ek bilgileri tutmak için kullanılır. Farklı özellikler, farklı olay türlerine karşılık gelir. Her giriş bir PersistenAuditEvent girişine bağlıdır. Depolanan bilgiler şunları içerir:

Tablo kolonları

Kolon adı	Kolon tipi	Açıklama
name	String	Güvenlikle ilgili olay için özniteliğin adı, ör. "RemoteAddress" özniteliği, güvenlik olayını deneyen IP'yi depolamak için kullanılır
value	Date	Güvenlikle ilgili olay için özniteliğin değeri

5.2.5.4 Database Version control

Fiziksel veritabanı şeması, ticari varlıkları temsil etmeyen ancak bunun yerine Veritabanı sürüm kontrol çerçevesi Liquibase tarafından kullanılan 2 ek tablo içerir. Bu tablolar:

1. DATABASECHANGELOG
2. DATABASECHANGELOGLOCK

Ve yapıları Liquibase belgeleri tarafından belirlenir.

5.2.6 Dağıtım görüntüsü

Yazılımın talep ettiği işlevleri sağlamak için çeşitli mimari bileşenlerin birbirleriyle ve diğer sistemlerle arayüz oluşturması gerekir. Bu dahili arayüzler bu bölümde daha ayrıntılı açıklanacaktır..

5.2.6.1 ETS Back-end => ETS Database

ETS Arka uç katmanı, JDBC / Hibernate / Spring JPA kullanarak ETS veri katmanı (RDBMS) ile iletişim kurar.

5.2.6.2 ETS Front-end component => ETS Back-end component

ETS Ön uç katmanı (Angular UI), HTTP veya HTTPS üzerinden Restful Web Services / API kullanarak ETS Back-end katmanı (Business / Service katmanı) ile iletişim kurar.

5.2.7 Arayüz görüntüsü

Dağıtım Görünümündeki ana öğeler:

•Web/Mobile İstemcileri: HTTP / HTTPS üzerinden Web Sunucusuna veya Uygulama Sunucusuna giden erişim

•**İsteğe bağlı** Web Server:

- HTTP/HTTPS üzerinden gelen erişim
- HTTP / HTTPS / AJP üzerinden Uygulama Sunucusu düğümüne (veya kümesine) giden erişim

•Application Server node (or cluster)

- HTTP/HTTPS üzerinden gelen erişim
- I REST HTTP/HTTPS üzerinden gelen erişim
- JDBC üzerinden Veritabanı Sunucusuna giden erişim

Önerilen Dağıtım için diyagramlar şunları içerir:

- Uygulama Sunucusunun tek bir düğümde kurulu olduğu tek düğümlü bir topoloji:

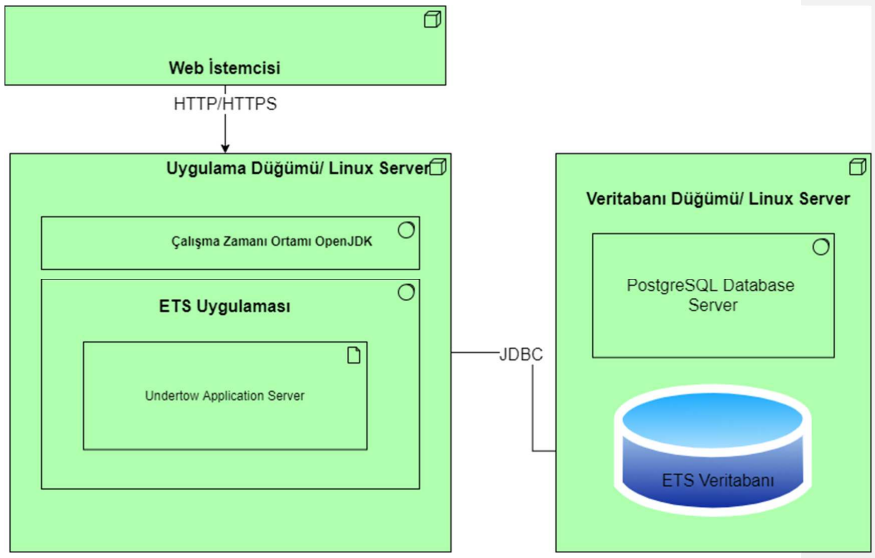
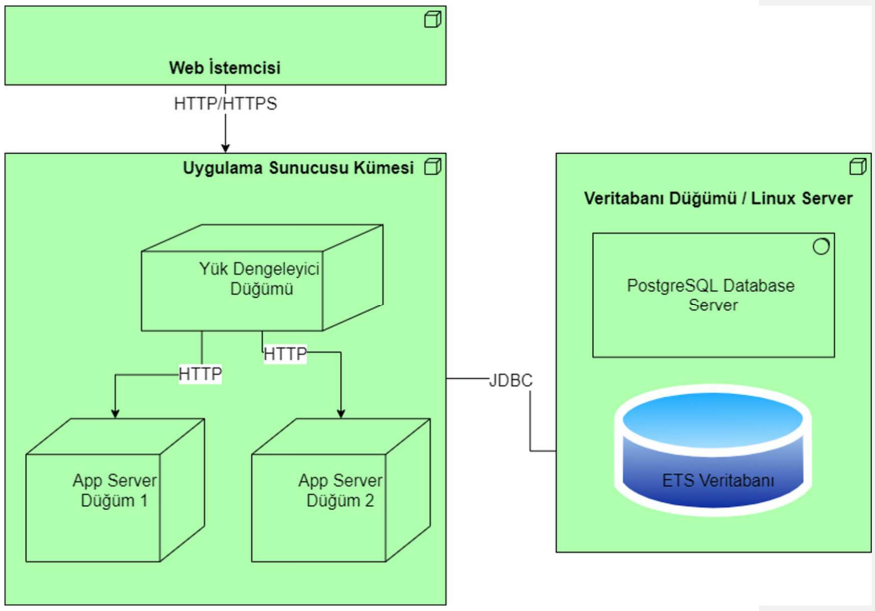


Figure 5.2-5: Dağıtım şeması 1

- Uygulama Sunucusunun iki düğümden oluşan bir küme olarak kurulduğu bir Yüksek kullanılabilirlik topolojisi. Load Balancer'a sahip üçüncü bir düğüm, istekleri iki Uygulama sunucusu düğümüne yönlendirir



Şekil 5.2-6: Dağıtım şeması 2

5.3 Güvenlik Tasarımı

This section contains the approach followed throughout this Architecture related to security concerns.

5.3.1 Güvenlik İlkeleri

Güvenlik mimarisi, bir uygulamanın normal mimarisinden ayrı değildir - en basit sistemlerin bile mimarisinin doğuştan gelen bir yönüdür.

Bilgi güvenliği genellikle aşağıdaki sütunlara dayanır:

- Gizlilik - yalnızca kullanıcıya izin verilen verilere erişime izin verme
- Bütünlük - verilerin yetkisiz kullanıcılar tarafından tahrif edilmemesini veya değiştirilmemesini sağlama
- Kullanılabilirlik - sistemlerin ve verilerin yetkili kullanıcılara ihtiyaç duyduklarında erişebilmelerini sağlama

Aşağıda, güvenlik özelliklerini şekillendirmek için dikkate alınan birkaç ilke listelenmiştir:

- Ekleme yüzey alanını en aza indirme: Bir uygulamaya eklenen her özellik, genel uygulamaya belirli bir miktar risk ekler. Güvenli geliştirmenin amacı, saldırı yüzey alanını azaltarak genel riski azaltmaktır. Örneğin, bir web uygulaması, bir arama işlevi ile çevrimiçi yardım uygular. Arama işlevi, SQL

enjeksiyon saldırılarına karşı savunmasız olabilir. Yardım özelliği yetkili kullanıcılarla sınırlıysa, saldırı olasılığı azalır.

- En az ayrıcalık: En az ayrıcalık ilkesi, hesapların iş süreçlerini gerçekleştirmek için gereken en az ayrıcalığa sahip olmasını önerir. Bu, kullanıcı haklarını, CPU sınırları gibi kaynak izinlerini, belleği, ağı ve dosya sistemi izinlerini kapsar. Örneğin, bir ara yazılım sunucusu yalnızca ağa erişim, bir veri tabanı tablosuna okuma erişimi ve bir günlüğe yazma yeteneği gerektiriyorsa, bu, verilmesi gereken tüm izinleri açıklar.
- Gizliliğe dayalı güvenlikten kaçınma: Belirsizlik yoluyla güvenlik, zayıf bir güvenlik kontrolüdür ve tek kontrol olduğunda neredeyse her zaman başarısız olur. Bu, sır saklamanın kötü bir fikir olduğu anlamına gelmez, basitçe anahtar sistemlerin güvenliğinin ayrıntıları gizli tutmaya bağlı olmaması gerektiği anlamına gelir. Örneğin, bir uygulamanın güvenliği, gizli tutulan kaynak kod bilgisine dayanmamalıdır.
- Güvenliği basit tutma: Saldırı yüzey alanı ve basitlik el ele gider. Bazı yazılım mühendisliği hevesleri, aksi takdirde görece basit ve basit olan koda aşırı karmaşık yaklaşımları tercih eder. Geliştiriciler, daha basit bir yaklaşımın daha hızlı ve daha basit olacağı durumlarda, çift negatiflerin ve karmaşık mimarilerin kullanımından kaçınmalıdır.
- En zayıf bağlantı: Bu güvenlik ilkesi, yazılımınızın hacker girişimlerine karşı dayanıklılığının, kod, hizmet veya ara yüz gibi en zayıf bileşenlerinin korunmasına büyük ölçüde bağlı olacağını belirtir.

5.3.2 Güvenlik Kaygıları

Bu Mimari, web uygulamaları için güvenlik risklerini belirlemek ve yönetmek için OWASP kaynaklarını kullanır.

Açık Web Uygulama Güvenliği Projesi (OWASP), yazılımın güvenliğini artırmaya odaklanmış, dünya çapında kar amacı gütmeyen bir hayır kurumudur. Misyon, yazılım güvenliğini görünür kılmaktır, böylece bireyler ve kuruluşlar bilinçli kararlar verebilir.

OWASP'ye göre, İlk 10 Web Uygulaması Güvenlik Riski:

- Enjeksiyon: SQL, NoSQL, OS ve LDAP enjeksiyonu gibi enjeksiyon kusurları, güvenilmeyen veriler bir komut veya sorgunun parçası olarak bir yorumlayıcıya gönderildiğinde ortaya çıkar. Saldırganın düşmanca verileri, yorumlayıcıyı istenmeyen komutları çalıştırması veya uygun yetkilendirme olmadan verilere erişmesi için kandırabilir.
- Bozuk Kimlik Doğrulama: Kimlik doğrulama ve oturum yönetimiyle ilgili uygulama işlevleri genellikle yanlış uygulanır ve saldırganların parolaları, anahtarları veya oturum belirteçlerini tehlikeye atmasına veya diğer kullanıcıların kimliklerini geçici veya kalıcı olarak üstlenmek için diğer uygulama kusurlarından yararlanmasına olanak tanır.
- Hassas Verilerin İfşası: Birçok web uygulaması ve API, finans, sağlık ve PII gibi hassas verileri düzgün bir şekilde korumaz. Saldırganlar, kredi kartı dolandırıcılığı, kimlik hırsızlığı veya diğer suçları işlemek için bu tür zayıf korunan verileri çalabilir veya değiştirebilir. Beklemede veya aktarım sırasında şifreleme

gibi ekstra koruma olmadan hassas veriler tehlikeye atılabilir ve tarayıcıyla değiştirildiğinde özel önlemler gerektirir.

- XML Harici Varlıklar (XXE): Birçok eski veya kötü yapılandırılmış XML işlemci, XML belgelerindeki harici varlık referanslarını değerlendirir. Harici varlıklar, dosya URI işleyicisini, dahili dosya paylaşımlarını, dahili port taramayı, uzaktan kod yürütmeyi ve hizmet reddi saldırılarını kullanarak dahili dosyaları ifşa etmek için kullanılabilir.
- Bozuk Erişim Kontrolü: Kimliği doğrulanmış kullanıcıların ne yapmasına izin verildiğine ilişkin kısıtlamalar genellikle düzgün bir şekilde uygulanmaz. Saldırganlar, diğer kullanıcıların hesaplarına erişmek, hassas dosyaları görüntülemek, diğer kullanıcıların verilerini değiştirmek, erişim haklarını değiştirmek gibi yetkisiz işlevlere ve / veya verilere erişmek için bu kusurları kullanabilir.
- Yanlış Güvenlik Yapılandırması: Yanlış güvenlik yapılandırması en yaygın görülen sorundur. Bu genellikle güvenli olmayan varsayılan yapılandırmaların, eksik veya geçici yapılandırmaların, açık bulut depolamanın, yanlış yapılandırılmış HTTP üstbilgilerinin ve hassas bilgiler içeren ayrıntılı hata mesajlarının bir sonucudur. Tüm işletim sistemlerinin, çerçevelerin, kitaplıkların ve uygulamaların güvenli bir şekilde yapılandırılması değil, aynı zamanda zamanında yamalanması / yükseltilmesi gerekir.
- Siteler Arası Komut Dosyası XSS: XSS kusurları, bir uygulama yeni bir web sayfasında uygun doğrulama veya çıkış olmadan güvenilmeyen veriler içerdiğinde veya HTML veya JavaScript oluşturabilen bir tarayıcı API'si kullanarak kullanıcı tarafından sağlanan verilerle mevcut bir web sayfasını güncellediğinde ortaya çıkar. XSS, saldırganların kurbanın tarayıcısında kullanıcı oturumlarını ele geçirebilecek, web sitelerini tahrif edebilecek veya kullanıcıyı kötü amaçlı sitelere yeniden yönlendirebilecek komut dosyaları yürütmesine olanak tanır.
- Güvensiz Serileştirme: Güvensiz serileştirme genellikle uzaktan kod yürütülmesine yol açar. Seri durumdan çıkarma kusurları uzaktan kod yürütülmesine yol açmasa bile, bunlar yeniden oynatma saldırıları, enjeksiyon saldırıları ve ayrıcalık yükseltme saldırıları dahil olmak üzere saldırıları gerçekleştirmek için kullanılabilir.
- Bilinen Güvenlik Açıklarına Sahip Bileşenleri Kullanma: Kitaplıklar, çerçeveler ve diğer yazılım modülleri gibi bileşenler, uygulama ile aynı ayrıcalıklarla çalışır. Savunmasız bir bileşenden yararlanılırsa, bu tür bir saldırı ciddi veri kaybını veya sunucunun ele geçirilmesini kolaylaştırabilir. Güvenlik açıkları olduğu bilinen bileşenleri kullanan uygulamalar ve API'ler, uygulama savunmalarını zayıflatabilir ve çeşitli saldırılara ve etkilere olanak sağlayabilir.
- Yetersiz Günlük Kaydı ve İzleme: Olay yanıtıyla eksik veya etkisiz entegrasyonla birlikte yetersiz günlük kaydı ve izleme, saldırganların sistemlere daha fazla saldırmasına, kalıcılığı sürdürmesine, daha fazla sisteme dönmesine ve verileri kurcalamasına, çıkarmasına veya yok etmesine olanak tanır. Çoğu ihlal araştırması, bir ihlali tespit etme süresinin 200 günden fazla olduğunu ve genellikle dahili süreçler veya izleme yerine harici taraflarca tespit edildiğini göstermektedir.

• Güvenlik Araçları

Yukarıdaki hususlar ve endişeler göz önüne alındığında, bu göreve yardımcı olacak araçlar şunlardır:

- Ön uç bileşen güvenliği için Angular çerçeve güvenlik özelliklerinin kullanımı
- Arka uç bileşen güvenliği için Spring Security çerçevesinin kullanımı
- OWASP yönergelerinin ve en iyi uygulamaların kullanımı

5.3.2.1 Ön Yüz Bileşenleri: Açısal güvenlik

Angular (Açısal), yaygın web uygulaması güvenlik açıklarına ve siteler arası komut dosyası çalıştırma saldırıları gibi saldırılara karşı yerleşik korumalara sahiptir.

Siteler arası komut dosyasını (XSS) önleme

Angular, siteler arası komut dosyası çalıştırma güvenlik modeli sağlar: XSS hatalarını sistematik olarak engellemek için Angular, tüm değerleri varsayılan olarak güvenilmez olarak değerlendirir. Özellik, öznitelik, stil, sınıf bağlama veya enterpolasyon yoluyla bir şablondan DOM'a bir değer eklendiğinde, Angular güvenilir olmayan değerleri temizler ve bunlardan kaçar.

Güvenilir olmayan değerleri DomSanitizer.sanitize yöntemi ve uygun SecurityContext ile sterilize etmek için yerleşik Açısal temizleme işlevleri kullanılabilir. Bu işlev aynı zamanda güvenilir olarak işaretlenmiş değerleri de kabul eder.

HTTP düzeyinde güvenlik açıkları

Angular (Açısal), iki yaygın HTTP güvenlik açığını, siteler arası istek sahteciliğini (CSRF veya XSRF) ve siteler arası komut dosyası dahil etmeyi (XSSI) önlemeye yardımcı olmak için yerleşik desteğe sahiptir. Bunların her ikisi de öncelikle sunucu tarafında azaltılmalıdır, ancak Angular (Açısal), istemci tarafında entegrasyonu kolaylaştırmak için yardımcılar sağlar.

5.3.2.2 Arka Yüz Bileşenleri: Spring Güvenlik çerçevesi

Spring Security, güçlü ve son derece özelleştirilebilir bir kimlik doğrulama ve erişim denetimi çerçevesidir. Yay tabanlı uygulamaların güvenliğini sağlamak için fiili standarttır. Spring Security, Java uygulamalarına hem kimlik doğrulama hem de yetkilendirme sağlamaya odaklanan bir çerçevedir. Tüm Spring projeleri gibi, Spring Güvenlik'in gerçek gücü, özel gereksinimleri karşılamak için ne kadar kolay genişletilebileceğinde bulunur

Özellikler şunları içerir::

- Kimlik Doğrulama: Kimlik doğrulama, belirli bir kaynağa erişmeye çalışanların kimliğini nasıl doğruladığımızdır. Kullanıcıların kimliğini doğrulamanın yaygın bir yolu, kullanıcının bir kullanıcı adı ve parola girmesini istemektir. Kimlik doğrulama yapıldığında kimliği biliriz ve yetkilendirme yapabiliriz.
- Yetkilendirme veya Erişim Kontrolü: Yetkilendirme, bir kullanıcının yalnızca izne sahip olduğu kaynaklara erişebilmesini sağlamakla ilgilidir.

- Oturum sabitleme, tıklama korsanlığı, siteler arası istek sahteciliği vb. saldırılara karşı koruma
- Servlet API entegrasyonu

Doğrulama

Spring Güvenlik, kimlik doğrulama için kapsamlı destek sağlar. Bir kullanıcının kimliğini doğrulamanın en yaygın yollarından biri, bir kullanıcı adı ve parolayı doğrulamaktır. Bu nedenle Spring Güvenlik, bir kullanıcı adı ve parola ile kimlik doğrulaması için kapsamlı destek sağlar. Aşağıdaki yerleşik mekanizmalar mevcuttur:

- Form Girişi
- Temel Kimlik Doğrulama
- Özet Kimlik Doğrulaması
- o Depolama Mekanizmaları

Bir kullanıcı adı ve parolayı okumak için desteklenen mekanizmaların her biri, desteklenen herhangi bir depolama mekanizmasından yararlanabilir.s:

- Bellek İçi Kimlik Doğrulamalı Basit Depolama
- JDBC Kimlik Doğrulamalı İlişkisel Veritabanları
- UserDetailsService ile özel veri depoları
- LDAP Kimlik Doğrulamalı LDAP depolama

Spring güvenliği ayrıca Kimlik Doğrulama için bir dizi ek özelliği destekler:

- Çoklu Parola şifreleme mekanizmaları
- Beni hatırla / kalıcı oturum açma kimlik doğrulaması, yani oturumlar arasında bir müdürün kimliğini hatırlayabilme

Yetki

Spring Güvenlik içindeki gelişmiş yetkilendirme yetenekleri, popülerliğinin en zorlayıcı nedenlerinden birini temsil etmektedir.

Yetkililer, Spring güvenliğinde yetkilendirmeyi yöneten ve iş Rollerine karşılık gelen ana varlıktır. Spring güvenlik, bir dizi gelişmiş yetkilendirme özelliğini destekler:

- İfade Tabanlı Erişim Kontrolü: Spring EL ifadelerini bir yetkilendirme mekanizması olarak kullanma yeteneği
- Yöntem düzeyinde güvenlik: yalnızca hizmet düzeyinde değil, aynı zamanda yöntem düzeyinde de çok ayrıntılı erişim kontrolleri tanımlama yeteneği

İstismarlara karşı koruma

Spring Security, yaygın istismarlara karşı koruma sağlar. Mümkün olduğunda, koruma varsayılan olarak etkindir. Daha spesifik olarak, Spring Güvenlik şunlara karşı korur::

- Spring, Siteler Arası İstek Sahteciliği (CSRF) saldırılarına karşı koruma sağlamak için kapsamlı destek sağlar
- Spring Güvenlik, web uygulamalarının güvenliğini artırmak için kullanılabilen birçok HTTP / HTTPS yanıt başlığı için açık destek sağlar. Gerekirse, Spring Güvenlik özel başlıklar sağlayacak şekilde de yapılandırılabilir.

5.3.3 Güvenlik Mimarisi Yaklaşımı

Elimizdeki hususlar, endişeler ve araçlar göz önüne alındığında, bu Mimari, ETS Web uygulamasının:

- güvenli kimlik doğrulama özelliklerini destekler
- yetkilendirme özelliklerini destekler
- hem ön uç hem de arka uç bileşenlere yönelik yaygın istismarlara karşı güçlü karşı önlemlere sahiptir

Bu açıdan aşağıdaki Mimari kararlar alındı:

1. Ön uç bileşenini sabitlemek için yukarıda açıklanan açısal çerçeve araçları ve mekanizmaları kullanılacaktır

2. Spring Güvenlik çerçeve araçları ve mekanizmaları, arka uç bileşeninin güvenliğini sağlamak için kullanılacaktır. Daha spesifik olarak:

   - ETS uygulaması için seçilen kimlik doğrulama yöntemi şudur: ETS İlişkisel Veritabanına JDBC Kimlik Doğrulaması ile Form Oturum Açma
   - Seçilen yetkilendirme yöntemi: Verilen Yetkiler (Roller) tanımlanacak ve Uygulama düzeyinde, Hizmet düzeyinde ve Yöntem düzeyinde belirli Erişim Kontrol noktaları eklenecektir
   - Yaygın saldırılara karşı koruma sağlamak için yerleşik Spring güvenlik bileşenleri yapılandırmada etkinleştirilecektir

3. 3. OWASP araçları, L1 yönergeleri ve karşı önlemler uygulama bağlamında kullanılacaktır:

   - Güvenlik kusurları olan üçüncü taraf kitaplıkları belirlenecek ve kaldırılacaktır
   - Yeterli izleme ve kayıt tutma
   - Yaygın kod kötü uygulamaları tanımlanır ve düzeltilir

6. KAYIT MERKEZİ YÖNETİMİ

Bu bölüm, Emisyon Kaydının idari gerekliliklerine göre rehberlik eder. Sunulan kılavuzlar, bir Emisyon Ticareti sisteminin işleyişinin altında yatan büyük risklerin azaltılmasını sağlamak için çeşitli kilit aktörler tarafından gerçekleştirilecek gerekli faaliyetleri sunmaktadır.

Bu bilgiler belgeye dayanmaktadır [1].

Sorumluluk	Örnek	Aktör
Kayıt defteri mimarisi	Sicilin yönetmeliğe uygun olmasını sağlamak..	Regulator
Kayıt BT sistemi	Kayıt sisteminin (donanım ve yazılım) amaca uygun olmasını sağlamak..	TRETS yönetici
Hesap yönetimi	Hesap açma, dondurma, kapatma; erişimin askıya alınması.	MRV sisteminin yöneticisi; bu veriler komut dosyası aracılığıyla TRETS'e kopyalanır
Müşteri ilişkileri yönetimi	Yardım hattı, yardım masası vb.	TRETS destek masası
Tesislerden yayılan emisyonlar	Tüm kurulumlar için doğru emisyonları girme	MRV sisteminin yöneticisi; bu veriler komut dosyası aracılığıyla TRETS'e kopyalanır
Kayıt faaliyetinin izlenmesi	Hataları ve anormallikleri tespit etmek, güvenlikle ilgili olanlar dahil olayları çözmek.	TRETS yönetici
Karbon birimlerinin ihracı	Yeni birimlerin verilmesi	İş yöneticisi of TRETS
Karbon birimlerinin tahsisi	Yeni birimlerin operatör holding hesaplarına tahsis edilmesi	İş yöneticisi of TRETS
Transferler	Genellikle operatör holding hesapları ve / veya ticaret hesapları arasında, hesaplar arasında birim transferleri	Transfer hesabına bağlı kullanıcılar
Birimlerin silinmesi	Birimleri gönüllü olarak silme	Transfer hesabına bağlı kullanıcılar

Sorumluluk	Örnek	Aktör
Teslim birimleri	Her bir operatör holding hesabının emisyonlarına uyma	operator holding hesabına bağlı kullanıcılar
BT / teknik yönler	Kullanılabilirlik; güvenlik; bilgisayarlı süreçlerin yürütülmesi; veri gizliliğinin yönetimi; işlem ve veri işleme yetkilerinin yönetimi, vb.	Sistem sahibi, TRETS yöneticisi
Yasal yükümlülükler	Operatör holding hesaplarının emisyon yükümlülüklerine uygunluğu	TRETS'in işletme yöneticisi, operatör holding hesaplarına bağlı kullanıcılar