OTOMATİK KİMLİK TANIMA SİSTEMLERİ:
OTOMOTİV SEKTÖRÜNDE BİR RFID
UYGULAMASI

Mustafa Demirel

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

Şubat-2013

ANADOLU ÜNİVERSİTESİ

JÜRİ VE ENSTİTÜ ONAYI

Mustafa Demirel'in "Otomatik Kimlik Tanıma Sistemleri: Otomotiv Sektöründe Bir RFID Uygulaması" başlıklı **Endüstri Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi **11.01.2013** tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

		Adı-Soyadı	İmza
Üye (Tez Danışmanı)	**:**	**Prof. Dr. MUSA ŞENEL**	**……..**
Üye	**:**	**Prof. Dr. REFAİL KASIMBEYLİ**	**……..**
Üye	**:**	**Doç. Dr. HAKAN G. ŞENEL**	**……..**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …….. tarih ve …….. sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

Otomatik Kimlik Tanıma Sistemleri: Otomotiv Sektöründe Bir RFID Uygulaması

Mustafa DEMİREL

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Musa Şenel

2013, 144 sayfa

RFID Teknolojisinin farklı sektörlerde kullanımının hızla artması özellikle iş süreçlerinin takibini kolaylaştırarak, işletim maliyetlerinin büyük oranda azaltmıştır. Otomatik kimlik tanıma ve veri toplama sistemlerinden olan RFID teknolojisi, giderek süreç otomasyonunda önemli bir yapı haline gelmektedir.

Bu çalışmada 10000'e yakın personeli bünyesinde barındıran bir üretim tesisinde, personel taşımacılığı operasyonlarının kayıt altına alınması, kontrolü ve bu operasyonlarının hakediş süreçlerinde merkezi bir yapıya geçmek için çözüm aranmış ve süreçlerde önemli mali kazanımlar ve kalite arttırımları sağlanmıştır. Tezde materyal olarak RFID teknolojisini meydana getiren; RFID okuyucu, RFID etiketler, kullanıcı bilgisayarları ve web tabanlı yazılım kullanılmıştır. Yazılım ile her lokasyonun kendi servis hareketlerini takip edebilmesi, bu hareket verilerinin merkezi bir veritabanında otomatik olarak kaydedilmesi, kontrol edilmesi ve raporlanması sağlanmıştır.

RFID teknolojisi ile oluşturulan bir servis takip otomasyonuyla; ilgili sürecin takibi için ayrılan personel maliyetlerinin önüne geçirilerek, ilgili bütün verilerin geriye dönük 5 yıllık olarak merkezi bir veritabanında saklanması sağlanarak, iç denetim faaliyetlerinde yaşanılan veya yaşanması muhtemel hataların önüne geçilmiştir.

Anahtar Kelimeler: RFID, Otomasyon, Otomatik Kimlik Tanıma, Web Tabanlı Yazılım

ABSTRACK

Master of Science Thesis

Automatic Identification System: A RFID Practice at Automotive Industry

Mustafa DEMİREL

Anadolu University

Graduate School of Sciences

Industrial Engineering Program

Supervisor: Prof. Dr. Musa ŞENEL

2013, 144 pages

Increasing usage of RFID technology in different industries, make process management more efficient and decrease operation costs. RFID, which is one of the Automatic Identification System, became mare important structure of process automation.

This work seek for a solution about a centralist structure which include whole personnel transportation operations, registering and progress payment processes for a company that have over ten thousand employees. Materials of the thesis are RFID readers, Active/passive RFID tags, web-based control software and user PC's. Web-Based control software provide ability of chasing their transportation vehicle movement data, control and report needed data from a safe database for ever location of company.

With transportation vehicle control automation which include RFID technology, company achieve important financial savings, quality improvement, for 5 years data storage ability and make provision about possible mistake at internal audits processes.

Keywords: RFID**,** Automation, Automatic Identification System, Web-Base Software

TEŞEKKÜR

Kişisel gelişimimde ve yaptığım bu tez çalışmasında her türlü konuda benden yardımlarını esirgemeyen aileme, Prof. Dr. Musa Şenel'e ve aynı zamanda çalışma arkadaşlarım olan Ford Otosan ailesine sonsuz teşekkürlerimi sunarım.

Mustafa Demirel

Şubat,2013

İÇİNDEKİLER

Sayfa

ÖZET ..... **i**
ABSTRACK ..... **ii**
TEŞEKKÜR ..... **iii**
İÇİNDEKİLER ..... **iv**
ŞEKİLLER DİZİNİ ..... **vii**
ÇİZELGELER DİZİNİ ..... **ix**

1. GİRİŞ **1**

2. RFID İLE KİMLİK TANIMA **3**
2.1 RFID'nin Tarihçesi 5
2.2. RFID Sistem Elemanları 9
2.2.1. Etiketler (Tags) (Transponders) 9
2.2.2. Okuyucular 19
2.2.3. Orta Katman Yazılımı (Middleware) 21
2.3 RFID Standartları 22
2.3.1 ISO Standartları 23
2.3.2 Elektronik Ürün Kodu (EPC) ve EPCglobal Standartları 25
2.4 Haberleşme Yapısı ve Çalışma Prensibi 29
2.4.1. Okuyucunun İşlevi 30
2.4.2. Okuyucunun Tasarım ve Performansı 31
2.4.3. Etiketin İşlevi 32
2.4.4. Endüktif Kuplaj-Pasif Etiketlere Güç Kaynağı 33
2.4.5. RFID Çalışma Prensibi 36
2.4.6. Sistemlerinde Kullanılan Modülasyon Yöntemleri 36
2.5 Çarpışma ve Çarpışma Önleyici Algoritmalar 42
2.5.1 Okuyucu Çarpışması 42
2.5.2. Etiket Çarpışması 43
2.5.3 Çarpışma Önleyici Algoritmalar 43

2.6 Uygulama Frekansları 44

3. WEB TABANLI YAZILIM 48

3.1 Web Tabanlı Uygulamalar 50

3.1.1. Web İstemcisi 52

3.1.2 Web Sunucusu 52

3.1.3 HTTP Protokolü 54

3.2 Web Tabanlı Uygulamaların Güvenliği 61

3.3 Web Uygulamalarında Sık Karşılaşılan Güvenlik Problemleri 63

3.3.1 Onaylanmamış Girdilerden Kaynaklanan Problemler 63

3.3.2 Komut Sızdırma veya SQL'e Sızma Açıkları 64

3.3.3 Çapraz Site Kod Çalıştırma Zayıflıkları 70

3.3.4 Hafıza Taşmaları 71

3.3.5 Kimlik Doğrulama ve Oturum Yönetimi Problemleri 72

3.3.6 Erişim Kontrolü Problemleri 73

3.3.7 Hata İşleme Problemleri 74

3.4 Web Uygulamalarının Güvenlik Problemlerine Karşı Test Edilmesi 74

3.4.1 Keşif Aşaması 75

3.4.2 Güvenlik Denetimi Aşaması 78

3.4.3 Raporlama Aşaması 83

3.4.4 Değerlendirme 83

4. RFID UYGULAMASI 85

4.1 Firma Tanıtımı 85

4.2 Süreç Kapsamı 86

4.2.1 Sürecin Amacı 87

4.2.2 Sürecin Mevcut Durumu 87

4.2.3 Girdi ve Çıktılar 88

4.2.4 Akış Diyagramı 89

4.2.5 Metrikler 90

4.2.6 Sorumluluk Alanları ve Roller 91

4.2.7 Sürecin gereksinimleri 91

4.2.8 Mevcut Süreci Destekleyen Araçlar 91
4.2.9 Mevcut Süreç Verim, Verimsizlik, Problem Noktaları 92
4.3 Yeni Prosesin Tanımı 92
4.3.1 Önerilen Sistem Özellikleri 93
4.3.2 Önerilen Web Tabanlı Yazılımın Güvenlik Unsurları 94
4.3.3 Verileri Vardiyalar Göre Gruplayarak Depolama Altyapısı 96
4.3.4 Otomatik Mail Desteği 98
4.3.5 Raporlama Opsiyonları 98
4.3.6 Arama Opsiyonları 98
4.4 Güvenlik Unsurları 99
4.4.1 Girdi ve Çıktılar 99
4.4.2 Akış Diyagramı 100
4.4.3 Akış Diyagramı Açıklaması 100
4.4.4 Metrikler 101
4.4.5 Sorumluluk Alanları ve Roller 101
4.4.6 Yeni Prosesi Destekleyecek Mantıksal Veri Akışı 101
4.4.7 Diğer Süreçler ve Sistemler ile Olan İlişkiler 102
4.5 Ara Birimlere Ait Çizimler, Rapor Tasarımları, Arayüzler ve Tarifler ... 102
4.5.1. Etiket Tanımlama Sayfası 104
4.5.2.Tanımlamalar 106
4.5.3. Raporlamalar 112
4.5.4. Otomatik Mail Desteği 114
4.5.5. Merkezi Kod Yapıları 114
4.5.6 Farklılık Analizi 130

5. SONUÇLAR **133**

KAYNAKLAR **137**

EK-1 Merkezi SQL Yapıları **140**

ŞEKİLLER DİZİNİ

1.1 Otomatik Kimlik Tanıma Sistemleri Auto-ID [29]……2
2.1. Etiket Okuyucu ve Kurumsal Sistemden Oluşan RFID Sistemi[6] 5
2.2. Pasif Etiket 16
2.3. Aktif Etiket 16
2.4. RFID Etiket Şekilleri 16
2.5. Disk Şekilli Etiket [13] 16
2.6. Anahtarlık Şeklindeki Etiket [13] 16
2.7. Kama Şeklindeki Etiket [13] 17
2.8. Cam Tüp Şeklindeki Etiket [13] 17
2.9. RFID Kimlik Kartı [13] 18
2.10. Yapışkanlı RFID Etiket [13] 18
2.11. Silindirik Etiket [13] 18
2.12. Okuyucu [13] 20
2.13. Elektronik Ürün Yapısı [6] 25
2.14. Elektronik Ürün Kodu ve Barkot Numarası İlişkilendirilmesi[6] 26
2.15. RFID Okuyucu Devre Yapısı 32
2.16. Okuycu Manyetik Alanı ile Enduktif Kuplajlanan Etiket 33
2.17. "L" Endüktansının Tanımı 34
2.18. İki Bobinin Oluşturduğu Karşılıklı Endüktans 35
2.19. Sinüzoidal Sinyal Modülasyonu (Taşıyı ve Yan Bandlar) 37
2.20. Mesaj (bilgi) İşareti ve ASK İşareti 37
2.21. İkili Kod Sinyali ile İki Durumda Anahtarlanan Taşıyıcı Genliği 38
2.22. ASK Modülasyonu Üretimi 38
2.23. FSK Üretimi 39
2.24. FSK Modülasyonu 39
2.25. FSK Modülasyonu Üretimi 40
2.26. BPSK İşaretinin Üretilmesi 40
2.27. BPSK İşareti 41
2.28. QAM(QPSK) Modülatörü 42
2.29. Okuyucu Çarpışması Protokolü 43
2.30. Pratikte Kullanılan RFID Frekansları 45
3.1. Web Uygulamasının Normal Kullanımı 66
3.2. Komut Sızdırma Açığından Etkilenen Uygulama 67
3.3. Geçerli Kullanıcı Adı ve Şifre İle Giriş 69
3.4. SQL'e Sızma Metoduyla Kimlik Doğrulamanın Aşılması 69
3.5. Çapraz Site Kod Çalıştırma Zayıflığından Etkilenen Web Sayfası 71
4.1. Anasayfa Giriş Ekranı 102
4.2. Ana Menü Ekranı 103
4.3. Araç Giriş-Çıkış Takip Ekranı 103

4.4. Tanımlama Ekranları 105
4.5. Taşıt Bilgileri Tanımlama Ekranı 105
4.6. Kart Atama Ekranı 105
4.7. Araç Bilgileri Ekranı 106
4.8. Şöför Sicil Bilgileri 107
4.9. Kullanıcı Yetkilendirme Altyapısı 108
4.10. Araç Tanımlma Ekranı 109
4.11. Vardiya Tanımlama Ekranı 110
4.12. Vardiya Zaman Aralıklarına Veri Tanımlama Ekranı 110

ÇİZELGELER DİZİNİ

2.1. EPC Global RFID Etiket Sınıfları 29
2.2. Frekans Bantları 46
2.3. Frekans Bantlarının Özellikleri 46
3.1. HTTP Yanıtı Durum Satırı 57
3.2. URL Formatı 58
3.3. Örnek HTTP İsteği 59
3.4. Örnek Sunucu Yanıtı 60
3.5. Komut Sızdırma Açığından Etkilenen Kod 65
3.6. SQL'e Sızma Açığından Etkilenen Kod 68
4.1. Vardiya Grupları 90
4.2. Vardiya Gruplama Aralıkları 97
4.3. Önerilen Ölçev Seti [36] 136

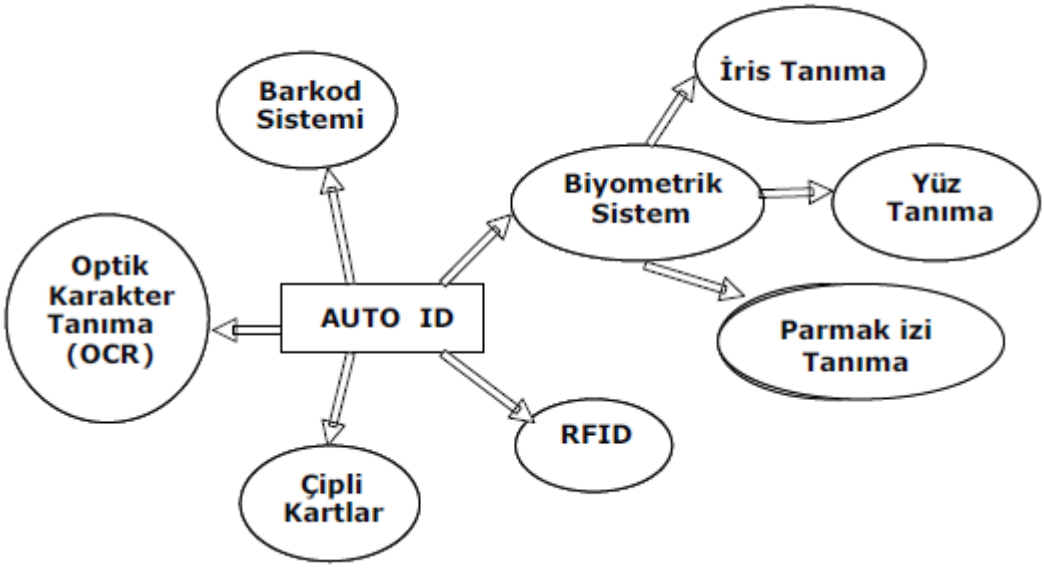
1. GİRİŞ

Gelişen teknoloji insanların hayatlarını kolaylaştırıcı sistemleri de beraberinde geliştirdi. Kimlik tanıma işlemlerinin otomatik olarak gerçekleşmesi, hayatı kolaylaştırıcı adımların başında gelir ve bunun için günümüze kadar pek çok sistem tasarlanmıştır. Bu sistemler çeşitli sektörlerde birçok uygulamada hayat bulmuştur. Otomatik kimlik tanıma sistemlerini (Auto-ID) incelediğimizde kendi içinde farklı teknolojileri içermekte olduklarını görürüz.

Geliştirilen otomatik tanıma sistemlerini temel olarak 5 grupta toplayabiliriz [1]:

1. OCR; optik karakter tanıma sistemleri
2. Biyometrik kimlik tanıma sistemleri
   - Yüz Tanıma
   - İris Tanıma
   - Parmak İzi Tanıma
   - Retina Tanıma
   - El İzi & Damar Tanıma

     Bir davranışın ölçülmesi ile veri elde edilen Davranışsal Biyometriler;
   - Ses Tanıma
   - İmza Tanıma
3. Akıllı Kart Sistemi
4. Barkod Sistemleri
5. RFID; Radyo Frekansı ile kimlik tanıma [1].



Şekil 1.1. Otomatik Kimlik Tanıma Sistemleri Auto-ID [29]

2. RFID İLE KİMLİK TANIMA

Radyo frekanslı tanımlama (RFID) radyo dalgaları kullanarak bir nesnenin veya insanın kimliğini (kendine has seri numarası formunda) kablosuz olarak ileten bir sistemi tanımlayan bir terimdir. RFID Otomatik Tanımlama (Auto-ID) Teknolojilerinin geniş kategorisi altındaki bir gruptur. Otomatik tanımlama teknolojileri barkotları, optik karakter tanıma sistemlerini ve retina tarama gibi biometrik teknolojileri içermektedir. Otomatik tanımlama teknolojileri veriyi manüel olarak girme ihtiyacı nedeniyle ortaya çıkan iş gücü ve zaman miktarını azaltmak ve veri doğruluğunu geliştirmek için kullanılır. Barkot gibi bazı otomatik tanımlama teknolojileri veriyi elde etmek için genellikle etiketi manüel taramak için bir insana ihtiyaç duyarlar. RFID'de ise okuyucuların etiketteki verileri elde etmesi ve bilgisayar sistemine iletmesi bir insana ihtiyaç duymadan tasarlanmıştır [3].

RFID şu anda otomatik kimlik tanıma teknolojilerinin en önemli ve ticari umut veren türüdür. En temel biçimde RFID bireysel ürünleri birbirinden ayırarak saptamayı ve ayrıca yerlerini ve hareketlerini takip etmeyi sağlar. RFID'nin ileri biçimlerinde, yalnızca ürünün kimliğinden ve yerinden başka ürünün fiziksel durumlarının bazı yönlerine de karar verme olası kılınmıştır. Örnek olarak bir gemi konteynırının iç sıcaklığı ve otobandaki kamyon motorunun devir hızı verilebilir. Takip edilen nesne depoya ulaşmış bir tüketici ürünlerinin tamamlanmış paleti, hastanelerde kullanılan pahalı medikal donanım, nakliye alanı içinde ve dışında hareket eden kamyonlar, kişisel ürünler, deterjan şişesi veya traş bıçağı paketi olabilir [4].

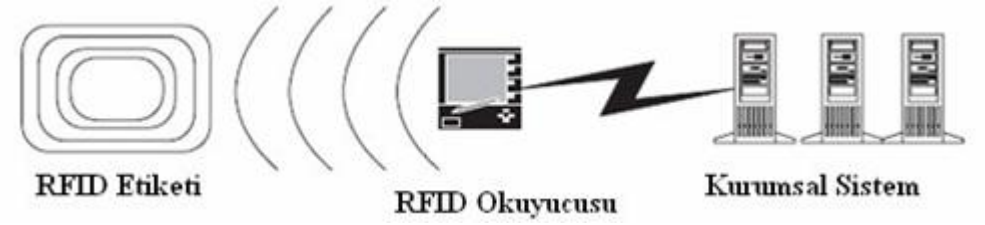
Barkotlar tedarik zincirinin her aşaması boyunca ürünlerin bilgisayar tabanlı takibini gerçekleştirmek için uzun süredir yardımcı olmaktadır ve bu teknolojinin bir gece içinde ortadan kalkması zordur. Çoğu uygulamada aşırı derecedeki ucuz fiyatı nedeni ile kullanılmaktadır. Ancak RFID bazı önemli avantajlar sunmaktadır. Barkot etiketi üzerindeki kodlanmış verinin elde edilmesi için etiketin kendisinin optik okuyucunun belirli uzaklığında ve direk görüş alanında olması gerekmektedir. Bu durum barkodun, kendisinin mürekkepli şeklinin odaklanmış lazer ışığı ışınları ile taranması sonucu

okunması nedeniyle zorunludur [4].

Buna karşılık RFID etiketlerinde etiketteki verinin elde edilmesi için yalnızca okuyucuya radyo dalgalarının ulaşması gerekir. Radyo dalgaları barkodun tarama lazerine göre daha büyük uzaklıklarda etkilidir. Bu uzaklıklar en azından 30- 60cm'den bazı durumlarda 9-10m'ye ve hatta daha fazlasına kadar büyümektedir. Radyo dalgaları kâğıt, karton, plastik, suni köpük, ağaç gibi tüketici ve endüstriyel ürünlerin paketlenmesinde veya üretilmesinde kullanılan genel maddelerin çoğunun içinden engellenmeden geçmektedir. Bu nedenle fabrika içinde veya araç yükleme durumlarında nesnenin pozisyonunun düzeltilmesi için çok daha az çaba harcayarak nesnelerdeki RFID etiketler okunabilir [4].

Radyo Frekanslı tanımlama Sistemi (RFID), çevresinde anten sarılı olan bir mikroçip ve bir okuyucudan oluşan otomatik tanımlama sistemidir. Veri ve enerji transferi, mikroçip ve okuyucu arasında herhangi bir temas olmadan sağlanmaktadır. Okuyucunun yaydığı elektromanyetik dalgalar etiketteki antenle buluşmakta ve mikroçipteki devreleri harekete geçirmektedir. Mikroçip, dalgaları modüle ederek okuyucuya yeni dalgayı sayısal veri halinde geri göndermektedir [5].

Radyo frekanslı tanımlama (RFID) entegre devreli mikroçip ve anteni içeren bir küçük etikettir ve bilgiyi gönderme, işleme ve depolamak için RFID okuyucusundan iletilen radyo dalgalarına cevap verme yeteneğine sahiptir. RFID sistemi Şekil 3,1'de gösterildiği üzere etiket, okuyucu ve kurumsal sistem olmak üzere 3 temel unsuru içermektedir. Etiket iliştirildiği ürünün tek (kendine has) tanımlama bilgisini içermektedir; okuyucu etiket içinde depolanmış bilgiyi okumak için radyo dalgaları yayar ve alır. Kurumsal sistem ise bütün toplanan verileri işler. Bu donanım bir kişisel bilgisayar kadar basit olabildiği gibi kurumsal yönetim bilgi sistemleri kadar karmaşık olabilir [12].



Şekil 1.1. Etiket Okuyucu ve Kurumsal Sistemden Oluşan RFID Sistemi [6]

2.1 RFID'nin Tarihçesi

RFID'nin kökleri erken askeri tanımlama sistemlerine dayanmaktadır ve 1940'ların başında başlamış bir dizi teknolojik yenilik ile temellenmiştir. Proceedings of the IRE'nin 1948 Ekim ayındaki sayısında yayınlanan Harry Stokcman'ın "Communication by Means of Reflected Power" adlı makalesi RFID'nin potansiyeli için ilk öngörü kabul edilmektedir. Bu makalede yazar yansıyan radyo sinyallerini, nesneden yansıyan sinyallere dayanarak uzaktaki bir nesnenin tanımlanması için bir yol olarak kullanılabileceğini tartışmaktadır. Bu makaleyi 1950'lerin başında D.B. Harris'in "Radio Transmission Systems with Modulatable Passive Responder" ve F.L. Vernon'un "Application of the Microwave Homodyne" adlı makaleleri izlemiştir. Bu iki makalede de iletilen radyo sinyallerinden tanımlanabilir, ölçülebilir ve fark edilebilir dönüş sinyalleri elde edilebileceği üzerinde durulmaktadır. Bu teknik, radar kullanımının olgunlaşması nedeniyle o zamanın iyi bilinen bir olgusu idi [6,7].

Radar RFID'nin teknolojik habercisidir ve radar başlangıcını radyonun dalga temelli doğasını saptayan ve üzerinde çalışan Alman araştırıcı Heinrich Hertz'e borçludur. Hertz, Guglielmo Marconi ve diğer erken dönem radyo araştırmacıları bu teknolojinin gelişmesinde önemli rol oynamışlardır [7].

Radar yankı kavramı üzerine dayanmaktadır. Radar baz istasyonu genellikle dönen anteni vasıtasıyla elektromanyetik enerjinin kısa yoğun ışınlarını yayar. Bu ışın gönderen aygıt daha sonra yayıcı durumundan alıcı durumuna geçer ve geri dönüş yankılarını dinler. Geri dönüş sinyallerine dayanarak radar nesnenin yeri kadar hızını ve yönünü tam olarak belirleyebilir [7].

İkinci dünya savaşı sırasında Almanlar eğer pilotlar uçaklarını üsse geri döneceklermiş gibi döndürürlerse bunun geri yansıyan sinyalleri değiştirdiğini

keşfettiler. Bu basit metot ile yerdeki radar görevlileri gelen uçakların yabancı uçaklar değil Alman uçakları olduğunu anlıyorlardı. Bu aslında ilk pasif RFID sistemidir. İskoç fizikçi Robert Alexander Watson-Watt'ın yönettiği gizli bir proje ile İngilizler ilk aktif dost ve düşman tanıma sistemini (Identify: Friend or Foe-IFF) geliştirmişlerdir. Böylece bütün İngiliz uçaklarına iletici konulmuştur. Bu iletici yerdeki radar istasyonlarından sinyaller aldığında, uçağın dost olduğunun saptanması için geri sinyal yaymaya başlamaktadır. RFID bu aynı temel kavram ile çalışmaktadır [8].

Etikete bir sinyal gönderildiğinde pasif sistemde etiket çalışmaya başlamakta ve geri sinyal yansıtmaktadır, aktif sistemde ise etiket sinyal yaymaktadır [8].

1950'ler bilimsel makalelerin yayınlandığı ve öncü araştırmalar ile RFID teknolojisinin teorik araştırmalarının yapıldığı yıllar olmuştur.1960'larda ise çeşitli araştırmacılar ve mucitler prototip sistemler geliştirmişlerdir. Sensormatic ve Checkpoint gibi bazı ticari sistemler çalınmayı engellemek için kullanılan "Elektronik Eşya İzleme (Elecronic Article surveillance-EAS)" aygıtlarını ortaya çıkarmışlardır [6].

EAS sistemleri, 1 bitlik bir sistem olup bir eşyanın varlığının ya da yokluğunun saptanmasında kullanılmaktadır. Bu teknolojinin geniş kullanım alanı, her bir malın etiketlendiği ve büyük bir anten okuyucunun her bir çıkışa konulduğu ve çalınmalara karşı kontrolün sağlandığı perakende mağazalarında kullanımındadır [5]. Eğer hırsız ilk önce fiyatını ödemeden bir malı mağazadan dışarı çıkarmaya çalışıyorsa, fiyatını ödemediği için kasada EAS devre dışı bırakılmadığından kapıdaki okuyucu, EAS'ın devrede olduğunu fark ederek alarmın çalması vasıtasıyla görevlileri uyarmaktadır.

Tüm EAS teknolojileri antenler, etiketler ve çözücüler olarak adlandırılan üç temel öğeden oluşmaktadır. Antenler tarafından yayılan sinyali bozarak alarmı devreye sokan etiketlerin, sert plastik veya kâğıt etiketler olmak üzere iki değişik çeşidi bulunmaktadır. Etiketler, plastik etiketlemede çivi veya tel ile kâğıt etiketlemede ise yapıştırılarak ürünler üzerine tutturulurlar [5].

Mario W. Cardullo 23 Ocak 1973'te yeniden yazılabilir hafızası ile bir aktif RFID etiketi için ilk Amerikan patentini aldığını duyurmuştur. Aynı yıl,

Kaliforniyalı girişimci Charles Walton anahtarsız bir kapının kilidini açmak için kullandığı pasif bir etiket için patent almıştır. Etiket ile güçlendirilmiş bir kart, kapının yanındaki bir okuyucu ile sinyal iletişiminde bulunur. Okuyucu RFID etiketi içinde depolanan geçerli kimlik numarasını fark edince, okuyucu kapıyı açar [8].

1970'li yıllar boyunca Los Alamos bilimsel laboratuarı modern RFID sistemleri geliştirmede rehberlik eden araştırmalar için merkez nokta olmuştur [16]. 1970'li yıllarda Amerikan Enerji bakanlığı Los Alamos Ulusal Laboratuarından nükleer maddelerin takibi için bir sistem geliştirilmesini istemiştir. Bu duruma bir grup bilim adamı kamyonlara etiket takılması ve güvenli tesislerin kapılarına da okuyucular yerleştirilmesini içeren bir plan ile çare bulmuşlardır [8].

Kapılardaki antenler kamyonlardaki etiketleri harekete geçirmekte ve bu etiketler kimlikleri ve sürücü kimliği gibi potansiyel diğer veriler ile cevap vermektedirler [8]. Bu sistem 1980'lerin ortalarında otomatik ücret ödeme sistemlerinin (automated toll payment system) geliştirilmesi ile ticari hale geçirilmiştir. Bu sistemler dünyanın her yerinde yollarda, köprülerde, tünellerde yaygın olarak kullanılmaya başlanmıştır [6,7,8]. Los Alamos Ulusal Laboratuarı ayrıca Amerikan Tarım Bakanlığının isteği üzerine ineklerin takibi için bir pasif RFID etiketi geliştirmiştir. Los Alamos bu duruma UHF radyo dalgalarını kullanan bir pasif RFID sistemi ile çare bulmuştur. Bu aygıt okuyucudan enerji çeker ve basitçe modüle edilmiş sinyali okuyucuya geri yansıtır.

Daha sonra şirketler alçak frekanslı (125 kHz) sistemler geliştirmişlerdir. Cam kapsüller içindeki etiket ineklerin derilerinin altına enjekte edilir. Bu sistem halen dünya çapında kullanılmaktadır.

Firmalar alçak frekanslı sistemleri ticari hale getirdikten sonra daha büyük okuma mesafesi ve daha hızlı veri transferi oranlarına sahip yüksek frekanslı (13,56 MHz) sistemlere geçmişlerdir. Şirketler Avrupa'da özellikle bu sistemleri yeniden kullanılabilen konteynırların ve diğer malların takibinde kullanmışlardır [8].

1990'lı yılların başında IBM şirketinin mühendisleri Ultra yüksek frekanslı (UHF) RFID sistemleri geliştirmişler ve patentini almışlardır. UHF

sistemleri iyi koşullarda 6 m'den daha fazla gibi uzun okuma mesafeleri ve hızlı veri transferi sunmaktadır. UHF sisteminin patenti 1990'lı yılların ortalarında IBM firmasında barkot sistemi sağlayıcısı olan Intermec firmasına geçmiştir.

Intermec RFID sistemi depo takibinden tarıma kadar sayısız farklı uygulamada kullanılmıştır. Fakat sistem o dönemde açık ve uluslar arası standartların olmaması ve düşük hacimdeki satışlar nedeniyle pahalıdır [8]. 1999 senesinde Uniform Code Council ve EAN International ile Gillette ve Procter & Gamble firmaları Massashusetts teknoloji enstitüsünde Auto-ID Center'ın kurulmasını desteklemişlerdir. Auto-ID Center 1999 ve 2003 yılları arasında 100'den fazla büyük şirketin desteğini kazanmıştır. Bu merkez tedarik zinciri boyunca takip edilecek tüm ürünlerin üzerine düşük maliyetli RFID etiketlerin konulması olasılığını araştırmış ve yüksek miktarda üretildiğinde fiyatı çok düşük olacak (amaç 5 senttir) RFID etiketleri üretmeyi kapsayan araştırmalar yapmıştır. Auto-ID Center'ın katkısı ucuz etiket üretmeyi denemenin ilerisine gitmiştir. Bu merkez Elektronik Ürün Kodunu (Electronic Product Code-EPC) geliştirmiştir. EPC her üretilen ürünün üzerine konulabilecek tek seri numarayı olası kılan bir sayısal şemadır. Auto-ID Center etiketlerin ve okuyucuların haberleşeceği bir yol (hava arayüzü protokolu) geliştirmiştir ve güvenli internet veritabanlarında bilgiyi depolamak için bir network altyapısı tasarlamıştır. Buna göre etiket numaralarıyla ilişkili tamamen sınırsız miktardaki veriler online olarak depolanabilecek ve verileri görmeye yetkili herhangi biri mevcut verileri istediğinde görebilecektir [3,7].

2003 yılının Ekim ayında Auto-ID Center'ın yönetimsel fonksiyonları bitmiş ve araştırma fonksiyonları Auto-ID laboratuarına geçmiştir. Auto-ID Center yerine EAN International ve Uniform Code Council ortaklığı olan EPCglobal kurulmuştur. EPCglobal hızlı, otomatik ve doğru ürün tanımlama için EPC teknolojileri ve networklarının global standartlar olarak kullanılmasını desteklemek amacıyla kurulmuş kar amacı gütmeyen endüstri temelli bir organizasyondur. EPCglobal networku EPC teknolojileri geliştirmek için Auto-ID Laboratuarlarıyla yakın çalışmayı devam ettirmektedir [8].

RFID teknolojisi günümüzde perakende sektöründen, ilaç sektörüne,

otomotiv sektöründen savunma sanayine kadar çoğu sektörde kullanılmaya başlamıştır. RFID eknolojisini oluşturan aygıtların fiyatı düştükçe ve teknolojinin yararları öğrenildikçe teknoloji gelecekte daha fazla kullanılamaya başlanacaktır.

2.2. RFID Sistem Elemanları

2.2.1. Etiketler (Tags) (Transponders)

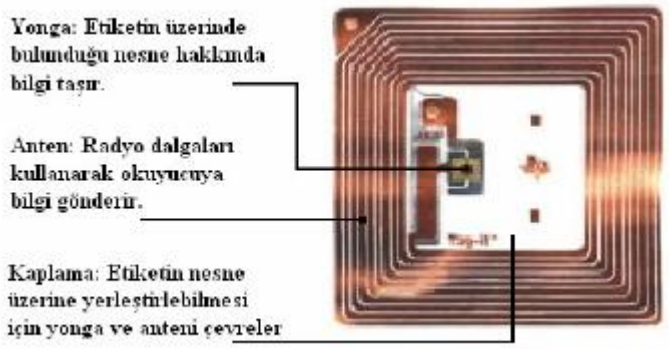
Transponders sözcüğü TRANSmitter/resPONDER sözcüklerinden türetilmiştir ve parçanın işlevini ifade etmektedir [14]. RFID etiketleri tanımlama bilgisini içeren bir mikroçip ve bu veriyi kablosuz olarak okuyucuya ileten bir antenden oluşmaktadır. En temelde mikroçipte bulunan ürünü kendine has olarak tanımlayan tanımlama verisi günümüzde kullanılan barkotlarınkine benzemektedir. Buna karşın en önemli farklılık RFID etiketinin barkoda göre daha yüksek veri kapasitesine sahip olmasıdır. Böylece ürünün üreticisi, lot numarası, ağırlığı, sahibi, ulaşacağı yer, ürünün bozulacağı sıcaklık aralığı gibi verileri etiket içerebilmektedir [12].

Etiketler uygulamasına özel olmak üzere çeşitli biçimlerde olmaktadırlar. Etiketler pasif veya aktif olmasına göre, şekline göre, frekanslarına göre çeşitli kategorilere ayrılabilir.

a. Pasif ve aktif etiketler:

RFID Etiketleri pasif veya aktif olabilirler. Pasif RFID etiketler kendilerine ait enerji kaynağına sahip değildirler. Onun yerine okuyucunun meydana getirdiği elektriksel alandan kendi işlem enerjilerini sağlarlar ve bu nedenle okuyucuya çok yakın mesafede olmadıkları sürece çalışmazlar. Pasif etiketler başarılı bir uygulama için çok fazla etiket gereken ve kütüphane giriş kontrol uygulamalarında olduğu gibi etiketlerin okuyucuya yakın yerleştirilebileceği durumlarda genellikle kullanılırlar. Şekil 15'de görüldüğü üzere pasif etiketler kolaylıkla bir kitabın kapağının içine veya küçük bir tüketim ürününe kolaylıkla takılabilir. Pasif etiketlerin içinde enerji kaynağı bulunmadığı

için küçük ve hafiftirler [7].



Şekil 2. 2. Pasif Etiket

Pasif RFID etiketler aktif etiketlere göre daha ucuzdurlar ve bakım istemezler. Bu nedenle büyük market zincirleri ve üreticiler pasif etiketleri tedarik zincirlerinde kullanmak istemektedirler. Fakat pasif etiketlerin okuma mesafeleri aktif etiketlere göre daha düşüktür [9]. Pasif etiketlerin sınırsız ömürleri vardır. Aktif etiketlere göre kötü yanları ise sınırlı veri depolama kapasiteleri, kısa okuma mesafeleri ve yüksek enerjili okuyucular istemeleridir.

Pasif etiketlerin performansları elektromanyetik gürültülü çevrelerde düşmektedir. Ayrıca mikroçip devresini çalıştırmak için kendi enerji kaynağını kullanan, fakat iletişim için okuyucudan aldığı enerjiyi kullanan yarı pasif etiketlerde mevcuttur [6].

Pasif etiketler antene, hafızaya, radyo frekans modülüne ve lojik/mikro işlem birimine sahiptir. Pasif etiketler okuyucudan etikete gönderilen radyo dalgalarından enerjilerini elde ederler. Anten doğru frekanstaki radyo dalgalarını alınca etiket, enerjiyi kendini uyandırmak ve okuyucuya bilgiyi göndermek suretiyle cevap vermede kullanır. Pasif RFID etiketler tanımlama numarasını depolayan küçük miktarda hafızaya sahiptirler. Radyo frekansı modülü anten vasıtasıyla gönderilen sinyali anlar ve anteni kullanarak okuyucuya geri bilgi gönderir. Etiketteki lojik/mikro işlemci birimi okuyucuya hangi bilginin geri gönderileceği ve çarpışmanın nasıl önleneceği konusundaki direktiflere cevap verir [4].

Aktif etiketler ise bir örneği Şekil 2.3'de görüldüğü üzere kendi içinde enerji kaynağına sahiptir ve okuyucuya sinyali kendileri yaydıkları için bazı

durumlarda 100 metrenin üzerinde olacak şekilde büyük okuma mesafelerine sahiptirler [7]. Bununla beraber pasif etiketlere göre önemli derecede büyüktürler ve böyle olmaları farklı tip uygulamalarda kullanılmasını sağlar. Aktif etiketler otomatik ücret ödeme uygulamaları (ülkemizdeki otomatik geçiş sistemindeki-OGS gibi) ya da depodaki ürün yüklü paletler gibi nesnelerin takibi için kullanılırlar. Pasif veya aktif etiketin kullanılması seçimi tamamen uygulamaya bağlıdır. Aktif etiketler az sayıda etiketin gerektiği (ürün seviyesinden çok palet veya büyük konteynır seviyesindeki gibi) ve etiket okuyucu arası mesafenin çok olması gereken uygulamalarda seçilir [7].

Şekil 2.3. Aktif Etiket [13]

Radyo sinyaline cevap veren (transponders) etiket ve yol, mevki gösteren işaretçi (beacons) etiket olmak üzere 2 çeşit aktif etiket mevcuttur. Radyo sinyaline cevap veren aktif etiketler okuyucudan sinyal aldıklarında çalışmaya başlarlar. Mesela radyo sinyaline cevap veren aktif etiket bulunan bir araç gişeye yaklaştığında gişedeki okuyucu arabanın ön canımda bulunan etiketin çalışmaya başlaması için bir sinyal gönderir ve sonra etiket kendisinin kendine has kimliğini okuyucuya yayar. Etiket yalnızca okuyucunun okuma mesafesinde kendi sinyalini yaydığı için pil ömrünü korur [9].

Yol ve mevki gösteren işaretçi aktif etiketler ise çoğunlukla takip ihtiyacı duyulan nesnenin doğru yerinin belirlenmesinde kullanılan tam zamanlı yer tayin istemlerinde (Real-Time Locating Systems-RTLS) kullanılır. RTLS'de işaretçi etiket önceden belirlenen zaman aralıklarında(bu nesnenin yerinin belirlenmesinin belli bir

sürede ne kadar önemli olduğuna bağlı olarak her 3 dakikada bir veya günde bir olabilir) kendisinin kendine has kimliğini içeren sinyali yayar. İşaretçi etiketin sinyali nesnenin takip edildiği alanın çevresinde yerleştirilmiş en azından 3 okuyucu anteni tarafından alınır [9].

Aktif etiketler ekstra okuma mesafeleri ve yetenekleri nedeniyle pasif etiketlere göre daha pahalıdırlar. Aktif etiketlerin maliyetleri hafıza miktarına, istenen pil ömrüne, üzerinde sıcaklık veya diğer sensorların olmasına, istenen sağlamlık düzeyine göre değişmektedir [9].

Aktif etiketler antene, hafızaya, radyo frekans modülü, lojik/mikro işlemci birimi, enerji birimi ve bazen sensorlar içerirler. Aktif RFID etiketler kendi enerji kaynaklarına sahiptirler. Aktif etiketlerdeki antenler çok değişik frekanslarda uzun mesafelerden radyo frekanslarını alır ve bu mesafelere bilgiyi gönderebilirler. Aktif etiketler, etiketlenmiş ürünün geçmişi hakkında etikete iletilen veriler veya sensorların verilerini kaydedecek kadar önemli miktarda hafızaya sahiptirler. Aktif etiketlerin radyo frekans modülü birçok frekansta radyo frekansı alabilir ve iletebilirler. Aktif etiketlerin lojik/mikro işlem birimlerinin yetenekleri sensorlar tarafından toplanan bilgilerin filtre edilmesine, ileri çarpışma yönetimi mekanizmasına, karışık komutlar setine izin verir. İşlem enerjisinin bu seviyesi etiketin akıllı bir aygıt gibi hareket etmesine ve yalnızca anlamlı durumların raporlanmasına izin verir. Sensorlar aktif etiketlere, etiketlenmiş ürünle ve onun çevresiyle ilişkili basınç, sıcaklık ve titreşim gibi nicelikler hakkında bilgi toplamalarına izin verirler [4].

Yeterli kendi işlem enerjisi ile RFID etiketler yalnızca iç verilerin depolanması ve raporlanması değil, ayrıca fiziksel sensorlardan gelen taze bilgilerin toplanmasını ve iletilmesini de sağlarlar. Elektronik etiketler içine yerleştirilebilen çeşitli sensorlar sıcaklık, nem, hava basıncı, ani şok, hızlanma ve hatta ışık yoğunluğu ve PH değerleri gibi parametreleri ölçerler. Bununla beraber bu durum maliyet konusu nedeniyle uygulanabilirlikten çok teoride kalmıştır. Sensorlu elektronik etiketler sensorların çok fazla enerji tüketmeleri nedeniyle çok fazla kullanılmamaktadır. Etiketteki sensorlar ve sensorlarla ilişkili elektronikler pasif etiketlere sağlanan enerjiden 10 ile 1000 kat arasında daha fazla enerji tüketmektedirler. Bu nedenle sensorlar kendine ait enerji

kaynağı olan aktif sistemlerde en çok kullanılırlar [4].

b. RFID sistemi frekansları:

Ülkelerde frekans tahsisi genellikle hükümetlerce kanunlar ve yönetmelikler aracılığı ile yönetilmektedir.

ISO ve benzeri organizasyonların standardizasyonları uyumlu hale getirme çalışmalarına rağmen uluslararası olarak RFID uygulamaları için frekans tahsisinde farklılıklar vardır [5,6].

RFID etiketleri ve okuyucuları her biri özel uygulama karakteristiklerine göre düşünülmüş birkaç farklı frekans aralıklarında işlem görürler. Alçak frekanslı sistemler (30-300KHz arası) genellikle çiftlik hayvanlarının tanımlanması gibi kısa mesafeli uygulamalarda ve pasif etiketler tarafından kullanılmaktadır. Alçak frekanslı aygıtlar özellikle 124 KHz, 125 KHz ve 134 KHz'lerde işlem görmektedir. Alçak frekanslı sistemler minimum işlem enerjilerine ihtiyaç duymaktadırlar, ucuzdurlar, iletim kapasiteleri kısa-orta mesafeye (alçak frekanslı pasif etiketler için 30cm. civarında) sahiptir, makul veri iletim hızını desteklemektedirler. Ayrıca çevreye duyarlı değillerdir, materyallerden ve sulu ortamlardan iyi geçebilmektedirler fakat elektromanyetik gürültüye duyarlıdırlar. Alçak frekanslı etiketler giriş kontrol, hayvan tanımlama, envanter kontrol ve araç bloke etme gibi uygulamalarda kullanılmaktadır [4,5,6,7,8].

Yüksek frekanslı sistemler (3-30 MHz arası) bagaj takip ve küçük ürün etiketleme gibi akıllı kart ve akıllı etiket uygulamalarında genellikle kullanılmaktadır.

Yüksek frekanslı sistemler özellikle 13,56 MHz'te işlem görmektedirler. Yüksek frekanslı sistemler alçak frekanslı sistemlere göre daha yüksek enerji istemektedirler ve daha pahalıdırlar. Ayrıca alçak frekanslılara göre daha büyük okuma mesafesi (Yüksek frekanslı pasif etiketler için 1 m civarında) ve daha yüksek veri iletim hızına sahiptirler fakat çevreye karşı duyarlıdırlar. Materyallerin içinden iyi geçememektedirler fakat alçak frekanslı sistemlere göre elektronik gürültülere daha dayanıklıdırlar. Yüksek frekanslı etiketler giriş

kontrol, akıllı kartlar, kütüphane kontrol gibi uygulamalarda kullanılırlar [4,5,-,7,8].

Ultra yüksek frekanslı (UHF) sistemler (300 MHz-3 GHz arası) öncelikle otoyol ücret ödeme sistemlerinde kullanılmıştır. ABD'de UHF sistemler genellikle 900MHz yada 2,45 GHz'de işlem görürken, benzer sistemler Avrupa'da 5,8 GHz'de işlem görmektedirler. UHF sistemler diğer frekans sistemlerine göre daha fazla enerji istemektedirler ve daha pahalıdırlar. Ayrıca diğer frekans sistemlerine göre daha büyük okuma mesafesi ve daha yüksek veri iletim hızına sahiptirler. UHF pasif etiketlerde okuma mesafesi 3-5 metre civarındadır. Konteynır takibi ve demiryolu uygulamaları gibi daha büyük okuma mesafesi isteyen uygulamalarda okuma mesafesini 100 metreye kadar çıkaran aktif etiketler kullanılır. UHF bandındaki radyo dalgaları su tarafından emilirler ve yüksek frekans sisteminde olduğu gibi materyallerin içinden iyi geçememektedirler. UHF etiketler demiryolu araç görüntüleme, palet ve konteynır takibi, araç takip gibi uygulamalarda kullanılır [5,6,7,9].

Şirketlerin alçak frekanslı ve yüksek frekanslı sistemler yerine UHF sistemlerini kullanmak istemelerinin en büyük nedeni okuma mesafesidir. Şirketler depolarında RFID'nin faydalı olabilmesi için etiketlerin en azından 3 metreden okunabilmesi gerekmektedir. Çünkü 3 metreden daha az mesafeden çıkış kapısından geçen bir palet üzerindeki bir etiketi okuma imkânı yoktur. Okuma mesafesi birçok faktör ile belirlenir, fakat en önemli faktör pasif etiketlerin okuyucuya veri iletirken kullandığı metottur [9].

Alçak ve yüksek frekanslı etiketler endüktif kuplaj-çiftler (inductive coupling) metodunu kullanırlar. Bu metotta okuyucu antenindeki bir bobin ile etiket antenindeki bir bobin bir elektromanyetik alan oluşturur. Etiket bu alandan güç emer ve bu gücü etiketteki devreyi çalıştırmak için kullanır ve sonra antendeki elektrik yükünü değiştirir. Okuyucunun anteni manyetik alandaki değişimi fark eder ve bu değişimi bilgisayarların anlayabileceği birler ve sıfırlara dönüştürür. Etiket ve okuyucu antenlerindeki bobinlerin manyetik alan oluşturma zorunluluğundan dolayı, etiket okuyucu antenine oldukça yakın olmalıdır, bu durum da bu sistemlerin okuma mesafesini düşürmektedir [7,9].

Pasif UHF etiketleri ise yayılma kuplaj-çiftler (propagation coupling)

metodunu kullanırlar. Okuyucu anteni elektromanyetik enerji (radyo dalgaları) yaymaktadır. Bu metotta elektromanyetik alan oluşmaz. Onun yerine, etiketler okuyucu anteninden enerji toplarlar ve mikroçip etiketin anteninde yükü değiştirmek için bu enerjiyi kullanır ve sonra değiştirilmiş sinyali geri yansıtır. UHF etiketleri birler ve sıfırlar ile 3 farklı yolla iletişim kurabilmektedirler. Etiketler geri gelen dalgaların genliklerini arttırabilmekte (genlik değiştirme anahtarı), faz dışındaki dalgaları değiştirilebilmekte (faz değiştirme anahtar) ya da frekansı değiştirebilmektedirler (frekans değiştirme anahtarı). Okuyucular sinyali alırlar ve değiştirilmiş dalgaları bir ve sıfıra dönüştürürler. Bu bilgi daha sonra ikili veriden bir seri numarasına veya etikette depolanan veriye dönüştürülmek üzere bilgisayara geçmektedir [7,9].

c. Şekillerine göre RFID etiketleri:

RFID etiketleri Şekil 2.4'de görüldüğü üzere birçok farklı uygulamalar için çeşitli şekil ve boyutta olmaktadırlar. Firmalar farklı fiziksel ürünlerin tanımlanma ihtiyaçlarını en iyi sağlayacak etiketin şeklini ve boyutunu çok dikkatlice seçmelidirler ya da birçok şekli kullanma yolunu seçebilirler.

Şekil 2.4. RFID Etiket Şekilleri [13]

En fazla kullanılan etiket şekli disktir. Şekil 2.5'de bir örneği görünen disk şekilli etiketlerin çapları 2,5 cm'den 10 cm'e kadar değişmektedir. Bu tip

etiketlerde etiketin bir palete veya diğer ulaşım konteynırlarına takı labilmesi için diskin ortasında bir monte etme deliği mevcuttur. Ayrıca ürün tanımlama amaçlı olarak giysilere dikilmek veya giysilerin etiketlerine takılmak amacıyla tasarlanmış gömlek düğmesi kadar küçük disk şekilli etiketlerde mevcuttur [7].

Şekil 2.5. Disk Şekilli Etiket [13]

İkinci olarak en fazla kullanılan etiketler kalıplanmış stiren (molded styrene) veya epoksi reçinelerdir(epoxy resin). Bu etiketler küçük ve dayanıklı tasarlanmışlardır ve sıklıkla anahtarlık şeklinde paketlenmektedirler. Şekil 2.6'da ilk olarak 1997 yılında Mobil Oil şirketi tarafından tanıtılan ve genellikle elektronik ücret ödeme için kullanılan anahtarlık şeklindeki (Speedpass olarak adlandırılan) RFID etiket görülmektedir [7].

Şekil 2.6. Anahtarlık Şeklindeki Etiket [13]

Şekil 2.7'de görünen kama şeklindeki (Wedge-shaped) etiketler özellikle fabrika çevrelerinde tedarik zinciri yönetimi uygulamalarını desteklemek için kullanılır [16,]. Kama şekilli etiket izlenen ürünlere birçok şekilde iliştirilebilecek şekilde

tasarlanmıştır. Kama şekli iletim alanındaki sinyal gücünün düşük olduğu veya bir metal tarafından engellendiği çevrelerde etiketin uyumunu garanti eder [13].

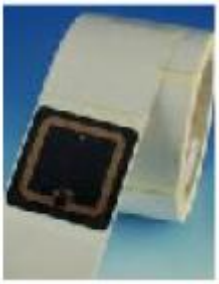
Şekil 2.7. Kama Şeklindeki Etiket [13]

RFID etiketlerinin en değişik şekillerinden biride Şekil 2.8'de görünen cam tüp şeklidir. Bu etiketler çok küçüktürler, aşağı yukarı çapları 3-4mm. civarında ve uzunlukları 2 cm'den daha küçüktürler. Bu etiketler özellikle izlenecek çiftlik hayvanlarının deri altına enjekte edilmek üzere tasarlanmışlardır [7].

Şekil 2.8. Cam Tüp Şeklindeki Etiket [13]

Hayvanlar için geliştirilmiş diğer etiketler ise tasma şeklindeki etiketler, küpe şeklindeki etiketler ve inek yuttuktan sonra devamlı midesinde kalan seramik etiketlerdir. Farklı amaçlar için kullanılan diğer etiketler ise Şekil 2.9'da gösterilen ve güvenli bina girişleri için kullanılan kimlik kartları şeklindeki etiketler, Şekil 2.10'da gösterilen yapışkanlı etiketlerdir [7].

Şekil 2.9. RFID Kimlik Kartı [13]



Şekil 2.10. Yapışkanlı RFID Etiket [13]

Bir diğer etiket ise Şekil 2.11'de görünen silindirik etiketlerdir. Bu etiketler otomobil boyama hatlarındaki gibi yüksek sıcaklık ve yakıcı, çürütücü ortamlar için tasarlanmıştır. Örnek olarak bu etiketler otomobil gövdelerine iliştirilir ve otomobil gövdesi geçerken etiket uygulanacak rengi ve şase sitilini hattaki okuyucuya iletir ve boyama robotu doğru renk ile boyamayı gerçekleştirir [13].

Şekil 2.11. Silindirik Etiket [13]

d. Veri programlama opsiyonlarına göre etiketler:

Bir etiket taşıdığı verinin niteliğine göre belleği salt okunabilir (Read Only Memory- ROM), bir kez yazılabilir çok kez okunabilir (Write Once/Read Many-

WORM), ve okunup yazılabilir (Read/Write-R/W) bellekler olabilir [5].

Salt okunabilir etiketler etiket üreticisi veya dağıtımcısı tarafından etiket üzerine önceden yazılmış sabit seri izleme numarası gibi verileri içerir. Bu etiketler tedarik zinciri boyunca hareket ederken hiçbir ek veri içeremedikleri için en ucuz etiketlerdir [5]. Ürünün hareketi ve aktivitesi izlenirken verilere gerekli güncelleme etiketin içine değil uygulama yazılımına yapılmak zorundadır.

Bir kez yazılabilir çok kez okunabilir etiketler kullanıcısına üretim veya dağıtım aşamasında etikete bir kerelik veri yazma olanağı verir. Bu veri tanımlama verisinin yanında lot numarası gibi verileri de içerebilmektedir.

Yazılabilir okunabilir etiketler gerektiği zaman etikete yeni veri yazılmasını sağlar. Bu yeni veri ürünün sahibine ulaşma saati ve günü, sabit bir aygıtın tamir geçmişinin güncellenmesi olabilir. Bu tür etiketler, 3 etiket türü arasından en pahalı olan etiket türüdür ve bu nedenle pahalı olmayan ürünlerin takibi için pratik değildir [5]. Bazı yazılabilir okunabilir etiketlerde hafıza birimi parçalara bölünmüştür, her parça bağımsız olarak güncellenebilmektedir. Örnek olarak böyle etiketleri kullanan tedarik zinciri içindeki firmalar diğer firmaların yazdıkları verileri okuyabilirken bunlarda değişiklik yapamazlar. Bu firmalar önceden kendilerine ayrılmış kısımlara veri yazabilirler [4].

2.2.2. Okuyucular

Başarılı bir RFID sistemi kurulumu için etiketler kadar önemli olan okuyucular pasif etiketleri etkinleştirmek için enerji verirler ve etiketlerden talep edilen veriyi elde ederler. Bazı durumlarda okuyucular etiketin hafızasındaki bilgileri eşleyebilmektedirler. Şekil 2.12'de örneği görünen okuyucuların iki sorumluluğu vardır. Birincisi ve öncelikli olanı okuyucunun, kendi kontrol alanındaki etiketleri harekete geçirmesi ve etiketlerin üzerlerindeki bilgiyi elde etmesidir. İkincisi ise etiketlerin bulunduğu alan ile tedarik zinciri içindeki harekete geçmiş etiketlerin oluşturduğu büyük miktardaki veriyi toplayan, analiz eden, dağıtan sistem arasında bir arayüz oluşturmaktır [7].

Şekil 2.12. Okuyucu [13]

Etiketler gibi okuyucu sistemlerin ve teknolojilerin de çok farklı çeşitleri vardır. Elle taşınabilen okuyucular, elle taşınabilen barkot tarayıcılarına benzer şekilde portatif veri toplama aygıtlarına bağlı olabileceği gibi uygulama yazılımına bilgi toplayan sabit terminal veya kişisel bilgisayarlara da bağlı olabilir. Elle taşınabilen okuyuculardan başka portatif veri toplama aygıtının içinde bulunan RFID okuyucular vardır. İçerisinde RFID okuyucu olan bu portatif veri toplama aygıtları genellikle depolarda ve fabrika içinde kullanılmaktadır. Bunlardan başka sabit RFID okuyucular ise ürün, okuyucunun yanından veya yakınından geçerken üründeki etiketi otomatik olarak okuyacak şekilde yerleştirilirler. Sabit RFID okuyucular konveyör ekipmanlarına, mağazaların arka depolarının giriş kapılarına, yüklenen ve firmaya gelen ürünlerin üzerindeki etiketleri otomatik olarak okumak üzere depo kapılarına ve malzeme taşıma ekipmanlarına takılırlar.

İşlevselliklerine göre de çeşitli okuyucular bulunmaktadır. Sessiz okuyucular (dumb readers) kısıtlı veri işleme, hesaplama gücüne sahiptirler. Bu okuyucular genellikle kendi üzerinde hesaplama gücüne sahip olan, verileri filtre edebilen, bilgileri depolayabilen ve komutları yerine getirebilen akıllı okuyuculara göre daha ucuz olmaktadırlar. Çevik okuyucular çeşitli protokolleri kullanan etiketler ile iletişim kurabilen okuyuculardır. Çoklu frekanslı okuyucular farklı frekansları kullanan etiketler ile iletişim kurabilen okuyuculardır. Ayrıca çevik ve çoklu frekanslı okuyucular verileri filtre etmek ve uygulamaları işletmek için kendi üzerlerinde hesaplama gücüne sahiptirler [10].

Okuyucular iç veya dış antenlere sahip olabilirler. Dış antenli okuyucular, antenler ile okuyucuyu bağlayan bir veya daha fazla girişe (ports) sahip olabilirler. En yeni okuyucular 8'den fazla anten girişlerine sahip olabilmektedirler. Ayrıca

okuyucular dış aygıtlar ile bağlantı kurabilmek için iç ve dış girişlere sahip olabilirler. İç giriş okuyucunun üzerindeki ve herhangi bir şey okuyucunun radyo dalgalarını kesince harekete geçen elektronik sensoru okuyucuya bağlayabilmektedir. Dış giriş ise okuyucu tarafından kontrol edilen bir program lojik kontrolcüsünü, konveyör ayırıcısını veya diğer aygıtları okuyucuya bağlamaktadır. Eski okuyucular seri girişleri kullanırken en yeni okuyucular Ethernet, Wi-Fi yada USB girişleri kullanmaktadırlar [10].

Eğer bir firma salt okunabilir etiketler kullanmıyorsa, etiketin tipine bağlı olmak üzere bir kere veya çok kez etikete veri yazma yeteneği önemli duruma gelmektedir. Okuyucular genellikle kendileri etikete veri yazma konusunda hizmet etmektedirler. Ancak çoğu firma hem okunabilir veri içeriği ve barkot yazabilen, hem de aynı zamanda RFID etiketine seri numarası gibi veriler yazabilen yeni nesil yazıcılar/kodlayıcılar kullanmaktadırlar. Bu yazıcılar barkot etiketindeki seri numarası ile RFID etiketindeki seri numarasının uyumlu olmasını sağlamaktadır.

2.2.3. Orta Katman Yazılımı (Middleware)

Orta katman yazılımı terimi RFID okuyucuları ile kurumsal uygulamalar arasında bulunan özel yazılımı tanımlamaktadır. Orta katman yazılımı her RFID sistemi için kritik bir unsurdur. Çünkü orta katman yazılımı okuyuculardan gerekli ham veriyi alır ve onları filtre ederek arka plandaki sistemlere kullanışlı verilerin geçmesini sağlar. Örnek olarak bir okuyucu aynı etiketi saniyede 100 kere okuyabilir ancak orta katman yazılımı bu durumu filtre eder ve kurumsal yazılıma bu etiketin bir okuma bilgisini iletir. Orta katman yazılımı doğru bilginin doğru zamanda doğru uygulama için elde edilmesinde anahtar bir rol oynar [10].

- RFID orta katman yazılımının birçok fonksiyonu yerine getirmektedir.
- Okuyucu arayüzü: Orta katman yazılımı çeşitli okuyucu üreticilerinin okuyucularından kurumsal yazılım için veri elde eder.
- Veri filtreleme: Bazen etiketler yanlış okunurlar. Orta katman yazılımı

etiket verilerinin toplanmasını, temizlenmesini ve filtre edilmesini sağlayarak düzeltilmiş veri ile kurumsal yazılımları besler.

- Okuyucu koordinasyonu: Orta katman yazılımı birçok okuyucuyu görüntüleyerek RFID etiketin bir okuyucunun okuma alanından diğerininkine geçerken hareketini izleyebilmektedir.
- Bu yönsel hareket izleme gerçekleştirilebilmekte ve kurumsal yazılıma envanter hareketi olarak geçebilmektedir.
- Sistem görüntüleme: Orta katman yazılımı etiketin okunmasının gerçek zamanlı görüntüsünün sağlanması için etiket-okuyucu ağının performansını görüntüleyebilmektedir. Böylece uygulama düzeni ve optimizasyonu için etiket okuma durumunun analizi ve geçmişi elde edilebilmektedir.

2.3 RFID Standartları

Standartlar ödeme sistemleri, açık tedarik zincirinde ürün ve yeniden kullanılabilir konteynırların takip edilmesi gibi uygulamalarda çok önemlidir. Son on yıl içinde farklı RFID frekansları ve uygulamaları için standartların geliştirilmesinde büyük çabalar gösterilmiştir. Etiketlerin ve okuyucuların birbiri ile iletişimini düzenleyen hava arayüzü protokolleri, verinin düzenini ve biçimini belirleyen veri içeriği, ürünlerin standartlara uyumunun test edilmesi (uyum) ve uygulamalar hakkında önerilmiş ve mevcut RFID standartları vardır [11].

RFID sistemi için uluslar arası standartlar geliştirmenin 3 önemli avantajı bulunmaktadır. Birincisi genel RFID standartları, farklı üreticiler tarafından üretilmiş etiket ve okuyucular arasında birlikte çalışabilirliği sağlayacak ve ülke sınırlarından geçişte kesintisiz birlikte işlem için izin verecektir. İkincisi uyumluluk ve değiştirilebilirlik sayesinde RFID elemanları ve ekipmanlarının talebi yükselecektir. Bu durumda maliyetleri düşürecektir. Son olarak uluslar arası kabul edilmiş RFID standartları dünya çapındaki RFID pazarının büyümesini kolaylaştıracaktır [12].

Halen UHF spektrumunda RFID teknolojileri için uluslar arası standartlar geliştirmek için çalışan başlıca iki organizasyon bulunmaktadır. Bu iki

organizasyon EPCglobal ve uluslar arası standartlar organizasyonu (The International organization for Standardization-ISO)'dur. EPCglobal 2004 yılının sonunda UHF bandı için kendi EPC sınıf 1 G2 protokolunu duyurmuştur, ve ISO 2004 yılının ağustos ayında kendi 18000-6 standardını duyurmuştur. Her iki standartta halen gelişmektedir ve birbirleri ile tam uyumlu değildir. Birleştirilmiş, küresel olarak birlikte çalışabilen RFID standardı RFID uygulamalarının tüm yararlarının anlaşılması için idealdir. Eksiksiz ve birleştirilmiş RFID standardının yokluğu birçok firmanın RFID sistemine adapte olmasında tereddüt etmelerine neden olmaktadır [12].

2.3.1 ISO Standartları

ISO otomatik tanımlama ve ürün yönetimi için RFID standartları geliştirmiştir. Bu standartlar ISO 18000 serisi olarak bilinir ve genellikle tedarik zincirindeki ürünlerin takip edilmesinde kullanılan sistemler için hava arayüzü protokolunu kapsamaktadır. Bu standartlar dünya çapındaki RFID sistemlerinde kullanılan başlıca frekansları kapsamaktadır. Ürün yönetimi için ISO'nun RFID standartları şunlardır [11]:

- ISO 15961: Veri protokolu: uygulama arayüzü
- ISO 15962: Veri protokolu: Veri kodlama kuralları ve mantıksal hafıza fonksiyonları
- ISO 15963: Radyo frekans etiketleri için tekil tanımlama
- ISO 18000-1: Referans mimarisi ve standartlaştırılacak parametrelerin tanımlanması
- ISO 18000-2: 135 KHz altı hava arayüzü haberleşmesi için parametreler
- Alçak frekans için ISO standardı
- ISO 18000-3: 13,56 MHz'de hava arayüzü haberleşmesi için parametreler
- Yüksek frekans için ISO standardı
- Okunup yazılabilir özellikteISO 18000-4: 2,45 GHz'de hava arayüzü haberleşmesi için parametreler
- ISO 18000-5: 5,8 GHz'de hava arayüzü haberleşmesi için parametreler

- ISO 18000-6: 860 MHz-960 Mhz arası hava arayüzü haberleşmesi için parametreler
- UHF frekansı için ISO standardı
- Okunup yazılabilir özellikte
- EPC standartları ile aynı pazarı amaçlıyor
- ISO 18000-7: 433 MHz'de hava arayüzü haberleşmesi için parametreler

ISO'nun ürün yönetiminin yanında diğer uygulama alanları içinde çeşitli standartları vardır. Bunlar [20]:

- ISO 10536: Tanımlama kartları-İletişimsiz bütünleşik devre(ler) kartları (10 cm'ye kadar)
- ISO 14443: Tanımlama kartları - İletişimsiz bütünleşik devre(ler) kartları (Proximity cards)
- ISO 15693: Tanımlama kartları-İletişimsiz bütünleşik devre(ler) kartları (Vicinity cards)
- ISO 11784: Hayvanların radyo frekans tanımlama ile tanımlanması-Kod yapısı
- ISO 11785: Hayvanların radyo frekans tanımlama ile tanımlanması-Teknik kapsam RFID kullanarak açık tedarik zincirinde ürün takip edilmesi oldukça yenidir ve bu konuda çok az standart mevcuttur. ISO taşıma konteynırları, paletler, taşıma birimleri, koliler ve tek ürünler için standartlar önermiştir.

  Tedarik zinciri için ISO'nun önerdiği RFID standartları şunlardır [20]:

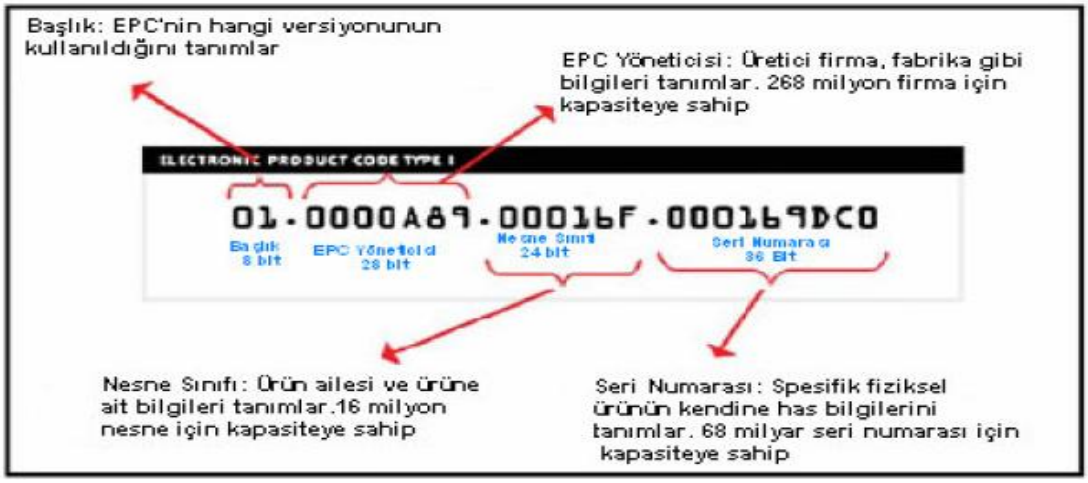
- ISO 17358: Uygulama gereksinimleri
- ISO 17363: Taşıma konteynırları
- ISO 17364: Geri dönüşümlü nakil ürünleri
- ISO 17365: Nakil birimleri
- ISO 17366: Ürün paketleme
- ISO 17367: Ürün etiketleme

2.3.2 Elektronik Ürün Kodu (EPC) ve EPCglobal Standartları

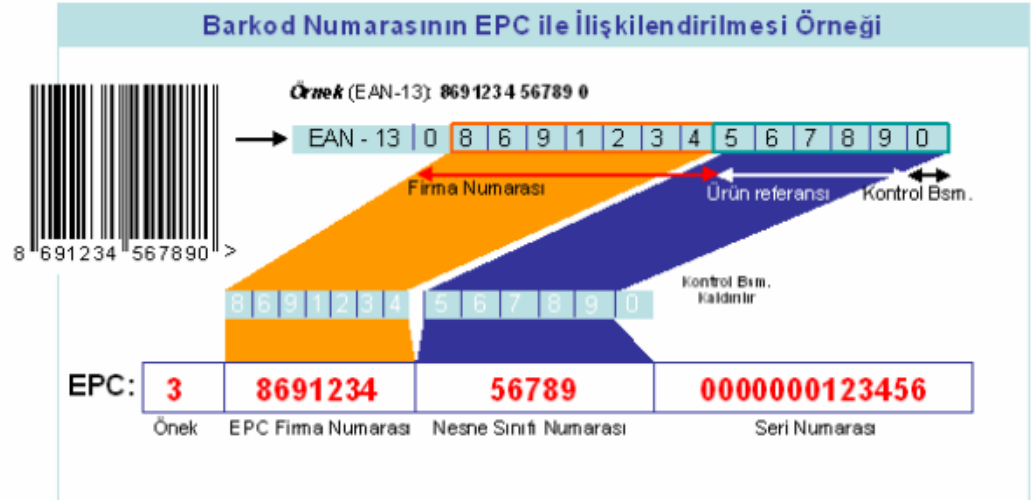
a. Elektronik ürün kodu (EPC):

Elektronik Ürün Kodu, Şekil 2.13'de gösterildiği gibi bir adet başlık bölümünden ve 3 adet veri bölümünden olmak üzere toplam 4 bölümden oluşan bir numaradır. Soldan sağa doğru birinci bölüm (başlık bölümü) EPC versiyon bölümünü tanımlar. Bu, ileride farklı uzunluk ve tiplerde yeni EPC kodları oluşturulmasına olanak sağlamak için tanımlanmıştır. İkinci bölüm EPC yöneticisini tanımlamaktadır. Bu genellikle ürünün üreticisidir. Örneğin 'The Coca-Cola Company'. Üçüncü bölüm nesne sınıfını tanımlamaktadır ve tam olarak ürün tipine karşılık gelir, genellikle stok saklama birimidir (Stock-Keeping Unit-SKU). Örneğin Diyet kola 330 ml kutuda, US versiyon. Dördüncü bölüm ise, ürünün tekil (kendine has) seri numarasını tanımlamaktadır. Tam olarak hangi 330 ml. kutu koladan söz edildiğini tanımlar [4,5,12].

Şekil 2.14'de gösterilen Elektronik Ürün Kodu 96 bit'liktir. Bu EPC 268 milyon adet şirket tanımlama kapasitesine sahiptir. Her üretici 16 milyon adet ürün sınıfı tanımlayabilir ve her ürün sınıfı için 68 milyar adet seri numarası tanımlama kapasitesi mevcuttur. Yılların getireceği daha fazla seri numarası ihtiyacını gelecekte tanıtılacak 128 ve 256 bit'lik kodlar karşılamayı garanti etmektedir [4,5].



Şekil 2.13. Elektronik Ürün Yapısı [6]



Şekil 2.14. Elektronik Ürün Kodu ve Barkot Numarası İlişkilendirilmesi[6]

b. EPCglobal standartları

Auto-ID Center küresel tedarik zinciri boyunca ürünleri tanımlamak ve takip etmek için elektronik ürün kodunu (EPC) ve bununla ilişkili teknolojileri geliştirmek için kurulmuştur. Auto-ID Center ayrıca kendi RFID sisteminin küresel ve açık standartlara dayanmasını istemiştir. 2003 yılında Auto-ID Center, Auto-ID laboratuarları ve EPCglobal adında iki ayrı organizasyona bölünmüştür. Auto-ID laboratuarları EPC teknolojileri ile ilgili birincil araştırmalarına devam etmektedir. EPCglobal ise EPC standartlarının resmi küresel standart olması için çalışmaktadır [20]. EPCglobal standartları aşağıdaki standartlardan oluşmaktadır [20]:

- EPC Etiket Verisi Standartları (Tag Data Standard-TDS, Version 1.3.1): Bu standart verinin etiket üzerine nasıl kodlanacağı ve EPC Sistemleri Ağının bilgi sistemleri katmanlarında nasıl kodlanacağı da dâhil, standart EPC etiket verisini tanımlamaktadır. Bu sürüm yayınlandığı zamanda yaygın kullanımda olan etiket çeşitleri içindir ve Gen 2 etiketleri için rehberlik sağlamamaktadır. GTIN (Küresel Ticari Ürün Numarası), SSCC (Taşıma Birimi Numarası), GLN (Küresel Lokasyon Numarası), GRAI (Küresel İade Edilebilen Varlık Tanımlayıcı), GIAI (Küresel Sabit Varlık Tanımlayıcı) ve GID (Genel Tanımlayıcı) için özel kodlama yöntemleri içermektedir.

- EPC Etiket Verisi Çevirimi Standardı (Tag Data Translation-TDT Standard, Version 1.0): Bu EPC Etiket Verisi Çevirimi Standardı EPC Etiket Verisi Standardı şartnamesinin makine-okuyabilir sürümü ile ilgilenmektedir. Makine-okuyabilir sürüm EPC biçimlerini doğrulamak için kullanılabileceği gibi farklı gösterim seviyeleri arasında tutarlı çevirim yapmak için de kullanılabilir. Bu şartname makine-okuyabilir sürümün nasıl yorumlanacağını açıklamaktadır ve makine-okuyabilir biçimlendirme dosyalarının yapısı ve öğeleri hakkında detaylar içermekte ve otomatik çevirim veya doğrulama yazılımlarında kullanımı ile ilgili rehber görevi görmektedir.
- Sınıf 1 Nesil 2 UHF Hava Arayüz Protokolu Standardı(Class 1 Generation 2 UHF Air Interface Protocol Standard, Version 1.1.0):"Nesil 2" (Gen 2) olarak bilinen bu standart 860 MHz-960 MHz frekans aralığında işlem gören pasif, ilk sorgucu(okuyucu) hareket eder (Interrogator talks first-ITF), radyo frekanslı tanımlama sistemleri için fiziksel ve mantıksal gereksinimleri tanımlamaktadır. Bu sistem okuyuculardan ve etiketlerden oluşur.
- Düşük Seviye Okuyucu Protokolu (LLRP) Standardı (Low Level Reader Protocol, Version 1.0.1): Bu standart, RFID okuyucuları ve alıcıları arasındaki arayüzü tanımlamaktadır. Bu arayüz protokolu düşük seviye olarak tanımlanmıştır çünkü RFID hava protokolu operasyon zamanlamasını ve hava protokolu komut parametrelerine erişimi kontrol etmektedir. Bu arayüzün tasarlanması bazı RFID sistemlerinde RFID hava protokolleri hakkında detaylı bilgiye ve RFID hava protokollerini uygulayan okuyucuları kontrol etme yeteneğine ihtiyaç duyulduğunu göstermektedir.
- Okuyucu Protokolu Standardı (Reader Protocol-RP Standard, Version Okuyucu protokolu, etiket okuma/yazma yetisi olan bir cihaz ile uygulama yazılımı arasındaki etkileşimi tanımlayan bir arayüz standardıdır.
- Okuyucu Yönetimi Standardı (Reader Management-RM, Version Bu standart yönetim yazılımı tarafından EPCglobal uyumlu RFID okuyucularının işlerlik durumunun izlenmesi için kullanılan protokolu tanımlar. Bu belge EPCglobal Okuyucu Protokolu Sürüm 1.1 şartnamesini tamamlar niteliktedir.
- Uygulama Seviyesi Olayları Standardı (Application Level Events-ALE Standard,Version 1.0): Bu standart, istemcilerin (müşterilerin) filtrelenmiş,

birleştirilmiş Elektronik Ürün Kodu (EPC) verisini çeşitli kaynaklardan edinebilecekleri bir arayüzü tanımlar.

- Elektronik Ürün Kodu Bilgi Servisleri Standardı (Electronic Product Code Information Services-EPCIS Standard, Version 1.0.1): Bu standart Elektronik Ürün Kodu (EPC) verilerinin toplanması ve paylaşılmasında kullanılan arayüzleri tanımlayan standarttır. EPC Bilgi Servislerinin (EPCIS) amacı Elektronik Ürün Kodu (EPC) kullanımı gerektiren çeşitli uygulamalar için EPC ile ilgili verilerin güvenli ve eşzamanlı olarak aynı organizasyon içinde veya farklı organizasyonlar arasında paylaşımını sağlamaktır. Böylelikle EPCglobal ağı üyelerinin EPC numarası taşıyan ürünlerle ilgili işlemlerde ortak bir yol izlemesi amaçlanmaktadır.
- Nesne İsimlendirme Servisi Standardı (Object Naming Service–ONS Standard, Version 1.0): Bu standart, Alan Adı Sisteminin (Domain Name System) belirli bir Elektronik Ürün Kodu (EPC) numarasının SGTIN bölümü ile ilişkili yönetimsel veri ve hizmetleri bulmak için nasıl kullanılacağını tanımlar. Hedef kitlesi Nesne İsimlendirme Servisi çözümleme sistemlerini uygulamalarında kullanacak olan geliştiricilerdir.
- EPCglobal Sertifika Profili Standardı (Certificate Profile Standard, Version 1.0):Bu standart, X.509 sertifikasının EPCglobal Ağı'ndaki öğelere tahsisi ve öğeler tarafından kullanımı için bir profil tanımlamaktadır. Bu belgede tanımlanan profiller İnternet Mühendislik İş Gücü'nün (IETF-Internet Engineering Task Force) Açık Anahtar Altyapısı (PKIX-Public Key Infrastructure) Çalışma Grubu'nda tanımlanan iki internet standardına dayanmaktadır.
- Elektronik Şecere Standardı (Pedigree Standard, Version 1.0): Bu belge, ilaç sektörü tedarik zincirindeki kullanıcılar için elektronik şecere belgelerini muhafaza ve değişimleri için gerekli yapıyı tanımlamaktadır.
- EPCglobal Mimari Çerçeve (Architecture Framework, Version 1.2): EPCglobal Mimari Çerçevesi, Elektronik Ürün Kodu (EPC) kullanımı ile tedarik zincirini iyileştirmeyi amaçlayan EPCglobal tarafından yürütülen temel hizmetleri ile birlikte, birbiriyle ilgili donanım, yazılım ve veri arayüzleri standartlarından oluşan bir derlemedir. Bu belgenin birkaç amacı bulunmaktadır.

EPCglobal, standartların yanında standartlar ile uyumlu farklı etiket sınıfları tanımlamıştır. Bu etiket sınıfları ve özellikleri Çizelge 2.1' de gösterilmiştir. Sınıf 0 ve Sınıf 1 arasındaki fark veri yapısında ve işlemlerindedir. Sınıf 0 etiketler salt okunurken, Sınıf 1 etiketler bir kez yazılabilir çok kez kunabilirdir [11]. Sınıf 0 etiketleri, Sınıf 1 etiketlerinden farklı bir protokol kullanmaktadır. Bu nedenle son kullanıcılar Sınıf 1 ve Sınıf 0 etiketlerinin her birinin okunması için multiprotokol okuyucular almak zorundadırlar. Sınıf 1 ve Sınıf 0, birlikte çalışamamalarının yanında ISO standartları ile de uyumlu değillerdir. 2004 senesi içinde EPCglobal bir 2. nesil protokol geliştirmeye başlamıştır. Bundaki amaç ISO standartları ile uyumlu olabilecek, basit ve küresel bir standart yaratmaktı. Sınıf 1 Nesil 2, 2004 yılının sonunda onaylanmıştır. RFID tedarikçileri Sınıf 1 Nesil2 'nin onaylanmasından sonra Sınıf 1 Nesil 2'de de çalışabilecek ISO UHF Standartları üzerinde çalışmışlardır.

Çizelge 2.1. EPC Global RFID Etiket Sınıfları

EPC Sınıfı	Özellikleri	Etiket tipi
Sınıf 0	Salt okunabilir	Pasif
Sınıf 1 Nesil 1	Bir kez yazılabilir Çok kez okunabilir	Pasif
Sınıf 1 Nesil 2	Bir kez yazılabilir Çok kez okunabilir. Küresel birlikte çalışabilirlik, arttırılmış veri iletim hızı, kill şifresi	Pasif
Sınıf 2	Okunup yazılabilir	Pasif
Sınıf 3	Okunup yazılabilir	Yarı Aktif
Sınıf 4	Okunup yazılabilir	Aktif

2.4 Haberleşme Yapısı ve Çalışma Prensibi

RFID sistemleri daha önce bahsettiğimiz gibi 3 ana bileşenden oluşurlar. Bunlar etiket, okuyucu ve denetleyicidir. Etiket, okuyucu tarafından gönderilen elektromanyetik alana girdiğinde aktif olur. Aktif olan etiket, sadece kendisine ait olan programlanmış kimlik bilgisini okuyucuya gönderir. Okuyucu, alıcı anteni yoluyla etiketin göndermiş olduğu bilgiyi alır, haberleşme için geliştirilmiş

yazılımı kullanarak bilgiyi işlenmek ve depolanmak üzere gerekli veri tabanına iletir.

Radyo frekans kimlik tanıma sistemlerinde okuyucu antenleri küçük yarıçaplı olarak seçildiklerinde anten merkezinde büyük manyetik alan yaratırken, büyük mesafelerde x>R büyük yarıçapa sahip antenler büyük alan şiddeti yaratmaktadır. Bunun sonucu olarak RFID sistemlerinde verici/sorgulayıcı anten yarıçapları bu nedenle büyük seçilmelidir. RFID sistemlerinde okuma mesafesini maksimum yapmak için anten yarıçapını büyütmek her zaman çözüm olmamaktadır. Maksimum okuma mesafesini hesaplarken sorgulama alan şiddeti değeri de bilinmelidir. Aksi halde anten yarıçapının büyütülmesi alan şiddetinin x=0 mesafesinde bile çok küçük olup sorgulama için gerekli enerjiyi sağlayamamasına sebep olur. Etiket okuyucu ile haberleşmesi için gerekli olan rezonans frekansından sapma gösterdiğinde haberleşme için büyük manyetik alan şiddetine ihtiyaç duyulur ki bu da haberleşme mesafesinin küçülmesine sebep olmaktadır.

2.4.1. Okuyucunun İşlevi

Okuyucu etiketle haberleşebilmek için gerekli enerjiyi radyo frekans kimlik tanıma sisteminin çalışma frekansına bağlı olarak KHz ve MHz frekanslarında zamanla değişen manyetik alan yaratarak sağlamaktadır. Okuyucu ürettiği zamanla değişen manyetik alanı genellikle dairesel çerçeve anten vasıtasıyla etikete gönderir. Dairesel çerçeve antenden akım aktığında çerçeve antene dik düzlemde oluşan manyetik alan şiddeti [7,9].

$$H = \frac{INR^2}{2(R^2+x^2)^{3/2}} \qquad (2.1)$$

Olarak hesaplanmaktadır. Burada:

I = Çerçeve antenden akan akım

N = Çerçeve anten sarım sayısı

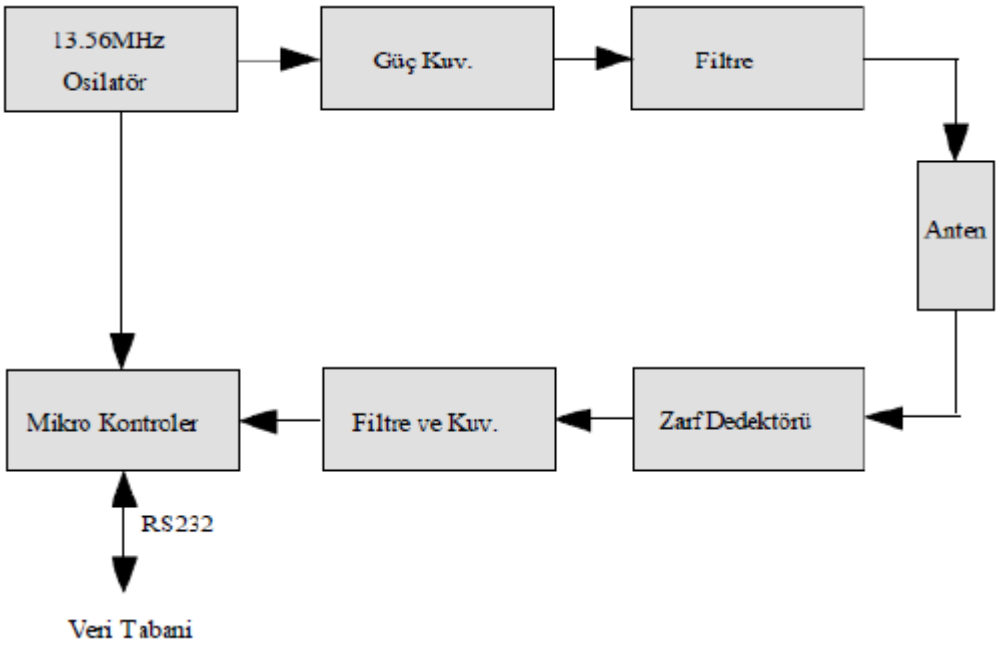
R = Anten yarıçapı

x = Anten düzlemine dik doğrultudaki alıcı uzaklığını tanımlar.

Denklemden de görüleceği üzere manyetik alan şiddeti mesafenin küpü ile ters orantılıdır. Endüktif bağlaşım prensibine dayanan radyo frekans kimlik tanıma sistemlerinde alanın mesafenin küpüyle ters orantılı olarak zayıflaması ana sınırlayıcı faktördür. Okuyucu tarafından gönderilen radyo frekans enerjisi etiketin fonksiyonlarını yerine getirebilmesi için taşıyıcı sinyal içermektedir. İncelediğimiz durumda söz konusu taşıyıcı frekansı 13.56MHz dir. Taşıyıcı sinyal etikete enerji sağlamasının yanı sıra, etiketteki bilgilerin okuyucuya gönderilmesini ve haberleşmenin senkronizasyonunu sağlar. Etiket okuyucu tarafından gönderilen sinyali alır ve modüle ederek tekrar okuyucuya gönderir. Etiket tarafından gönderilen okuyucu antenine gelen sinyaller geri şaçılım sinyalleri olarak adlandırılır. Okuyucu doğrultusunda geri saçılan sinyaller okuyucu tarafından şifresi çözülerek alınır.

2.4.2. Okuyucunun Tasarım ve Performansı

Pratikte kullanılan tipik bir yüksek frekans RFID okuyucu devre yapısı Şekil 2.15'de verilmektedir. Okuyucu aynı zamanda alıcı-verici olduğundan alıcı ve verici kısımlarını içermektedir. Verici 13.56MHz frekansında sinyali osilatörde üretir, kuvvetlendirir, filtreler ve akord devresi yardımıyla antenden etiket doğrultusunda gönderir. Alıcı kısımda ise etiketin göndermiş olduğu bilgiler zarf dedektörü ile işlenir, filtrelenir ve kuvvetlendirilerek mikro kontrolöre veri tabanına gönderilmek üzere iletilir. Denklem (2.1) ifadesine göre anten yarıçapı artırıldığında manyetik alan şiddeti de artmaktadır. Diğer taraftan NI da artırıldığında H da artacaktır. Manyetik alan şiddetinin artırılması için her iki durumda da sınırlamalar mevcuttur. Anten yarıçapı büyütüldüğü zaman okuyucu portatif özelliğini kaybedecek ve maliyeti artacaktır. NI değeri artırıldığında okuyucu anten endüktansı artacak, yüksek endüktans yükü de büyük oranda geriye yansıyan güce sebep olacaktır. Sonuç olarak NI çarpanını mümkün olduğu kadar küçük tutup haberleşme için gerekli manyetik alan şiddeti seviyesini elde edecek sistem tasarlanmalıdır.



Şekil 2.15. RFID Okuyucu Devre Yapısı

2.4.3. Etiketin İşlevi

Okuyucu etiketle Pasif radyo frekans kimlik tanıma etiketleri çalışması için gerekli gücü endüktif bağlaşımla antende indüklenen gerilimden alır. Antende indüklenen gerilim [9,10];

$$V = -N\frac{d\varphi}{dt} \tag{2.2}$$

Ve magnetik akı,

$$\varphi = \int B.dS \tag{2.3}$$

Olarak hesaplanır. Maksimum manyetik akı okuyucu anteni ile etiket anteninin paralel olması durumlarında elde edilir. Etiket anteninde indüklenen maksimum gerilim,

$$V = -\frac{\mu_0 N_1 N_2 a^2(\pi b^2)}{2(a^2+x^2)^{3/2}}\frac{di}{dt} \tag{2.4}$$

N_1= Okuyucu anten sarım sayısı

N_2= Etiket anteni sarım sayısı

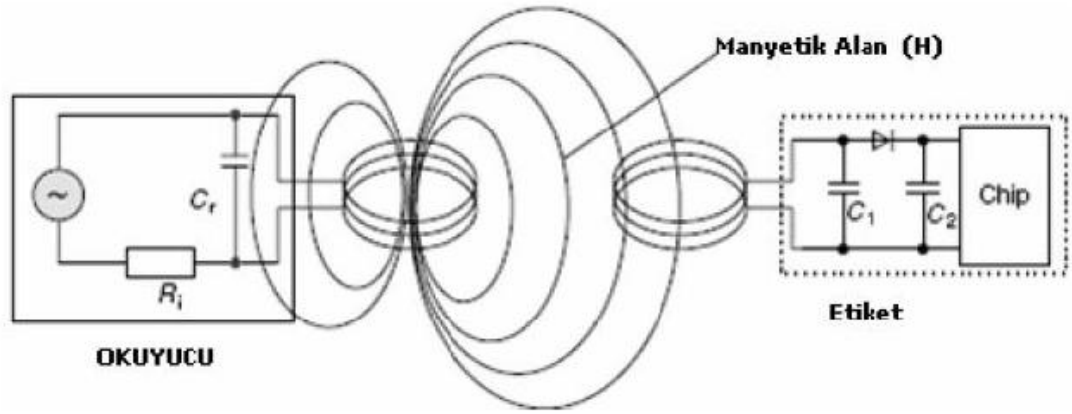
a= Okuyucu anten yarıçapı

b= Etiket anten yarıçapı

x= Okuyucu ile etiket arasındaki uzaklık

2.4.4. Endüktif Kuplaj-Pasif Etiketlere Güç Kaynağı

Endüktif kuplajlanan etiket, genellikle tek bir mikrochip ve anten görevi gören geniş yüzeyli bir bobinden oluşan bir elektronik veri taşıyıcısı olarak işlem görür. Endüktif kuplajlanan etiketlerin hemen hepsi pasif olarak çalışır. Bu da mikrochip için gereken enerjinin tamamının okuyucu tarafından sağlanması gerektiğini ifade eder. Bu amaçla, okuyucu etiket üzerindeki bilgiyi okuyabilmek için gerekli enerjiyi etiket doğrultusunda gönderir. Çünkü kullanılan frekans aralığındaki (<135 kHz: 2400 m, 13.56 MHz: 22.1m) dalga boyu, okuyucu anteni ve etiket arasındaki mesafeden birçok kat daha büyük olduğu için, elektromanyetik alan, etiket ve anten arasındaki mesafeye bağlı olarak, değişken manyetik alan gibi davranabilir. Bu işlem Şekil 2.16' da görülmektedir.



Şekil 2.16. Okuyucu Manyetik Alanı ile Enduktif Kuplajlanan Etiket

İletken döngülerinin geometrik yapılarına bağlı kuplajı hakkında nitelikli bir tahmin yapılabilmesi için kuplaj katsayısı tanımlanmıştır. Kuplaj katsayısı aşağıdaki gibi formulüze edilmiştir

$$k = \frac{M}{\sqrt{L_1 L_2}} \tag{2.5}$$

L1 ve L2 iki bobinin self endüktansıdır. M ise; iki iletken döngüsünün akı değişimi için tanımlanan karşılıklı endüktanstır. Kuplaj Katsayısı her zaman şu iki değer arasındadır: 0≤ k≤1.

k = 0: geniş mesafeye veya manyetik ekranlamaya bağlı tam ayrım

k = 1: toplam kuplaj

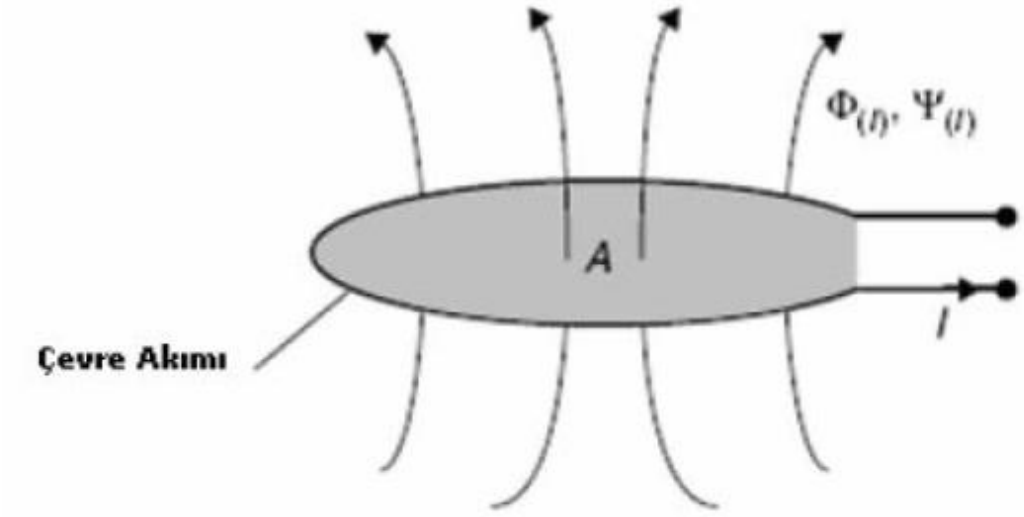
Endüktans(L), iletken bobinlerin karakteristik özelliklerinden biridir. Bilindiği gibi bir iletkenden akım geçirildiğinde, iletken etrafında bir manyetik alan oluşur. Böylelikle de bir manyetik akı üretilir.

$$\psi = \sum_N \varphi_N = N.\varphi = N.\mu.H.A \tag{2.6}$$

Bu akının geçen akıma oranı endüktansı formulüze eder;

$$L = \frac{\psi}{I} = \frac{N.\varphi}{I} = \frac{N.\mu.H.A}{I} \tag{2.7}$$

Bobinden geçen akımın oluşturduğu, bobin sargılarını çevreleyen bu manyetik alan Şekil 2.17'de görüldüğü gibi kâğıt üzerinde daireler şeklindeki kuvvet çizgileri ile sembolize edilir.

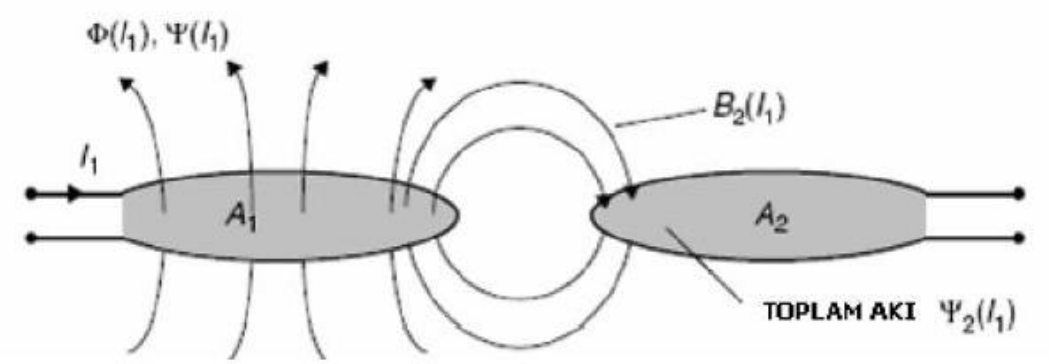


Şekil 2.17. "L" Endüktansının Tanımı

Akımın artıp azalmasına ve yön değiştirmesine bağlı olarak bobinden geçen kuvvet çizgileri artıp azalır ve yön değiştirir. Bobinin sarım sayısı ve kesit alanı ne kadar büyük olursa, "L" o kadar büyük olur. Böylelikle okuyucu anten endüktansı L1 ve etiket anteninin endüktansı L2'nin büyük olması durumunda, kuplaj katsayısı küçülür.

Karşılıklı Endüktans (M), aynı nüve üzerine sarılı iki bobinin birinden akım geçirildiğinde, bunun nüvede oluşturduğu kuvvet çizgileri diğer sargıyı da etkileyerek, bu sargının iki ucu arasında bir gerilim oluşturur. Bu gerilime "endüksiyon gerilimi" denir. Bu şekilde iletişim, karşılıklı (ortak) endüktans denen belirli bir değere göre olmaktadır. Şekil 2.18'de görülen karşılıklı endüktans M ile gösterilir ve su şekilde ifade edilir:

$$M = L_1 . L_2 \tag{2.8}$$



Şekil 2.18. İki Bobinin Oluşturduğu Karşılıklı Endüktans

$$M = M_{12} = M_{21}$$

$$M_{12} = \frac{\mu . N_1 . R_1^2 . N_2 . R_2^2 . \pi}{2\sqrt{(R_1^2 + x^2)^3}}$$

$$M_{21} = \frac{\mu . N_1 . R_1^2 . N_2 . R_2^2 . \pi}{2\sqrt{(R_2^2 + x^2)^3}} \tag{2.9}$$

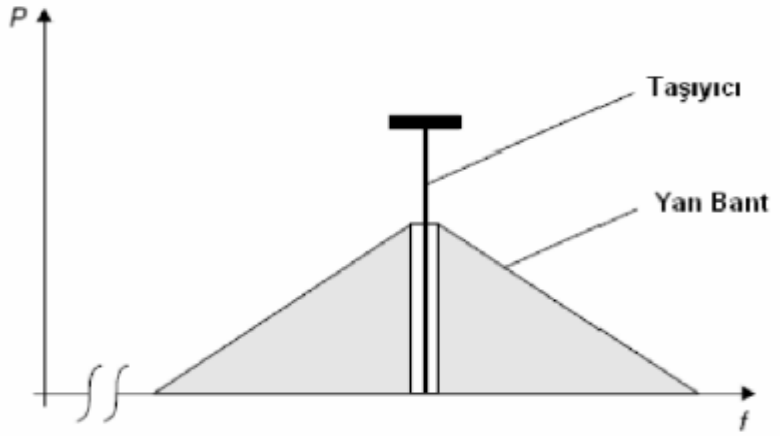
2.4.5. RFID Çalışma Prensibi

Radyo frekans kimlik tanıma sistem haberleşmesinde okuyucu radyo frekans sinyallerini gönderir. Okuyucunun radyo frekans alanına girmiş bulunan etiket, haberleşmesi için gerekli olan enerjiyi bu alandan alır. Etiket haberleşmesi için gerekli olan enerjiyi aldığında, üzerinde depolanmış bilgiye göre taşıyıcı sinyali modüle eder. Modüle edilmiş taşıyıcı etiketten okuyucuya gönderilir. Okuyucu modüle edilmiş sinyali dedekte eder, şifresini çözer ve okur. Son olarak alınan bilgi veri tabanının bulunduğu bilgisayara aktarılır.

Bir RFID sisteminin en önemli parçaları antenli bir chipten yapılan etiket (tag) ve antenli bir okuyucudur (reader). Okuyucu donanımı elektromanyetik dalgalar yayar. Etiket anteni bu dalgaları almak için ayarlanmıştır. Pasif bir RFID etiketi, okuyucudan yayılan dalgaları algılar ve bunu mikrochipin devrelerini harekete geçirmek için kullanır. Mikrochip bu dalgalardaki dijital bilgiyi değiştirir ve okuyucuya geri gönderir. RFID etiket ve okuyucuları iletişim kurabilmek için aynı frekansa ayarlanmalıdır.

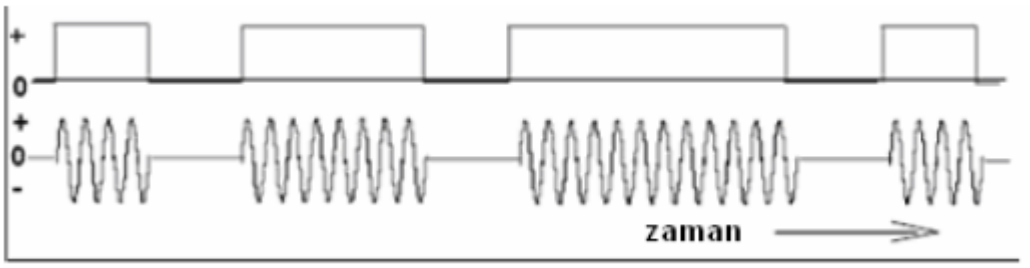
2.4.6. Sistemlerinde Kullanılan Modülasyon Yöntemleri

Genlik Kaydırmalı Anahtarlama (ASK), Faz Kaydırmalı Anahtarlama (PSK) ve Frekans Kaydırmalı Anahtarlama (FSK) olmak üzere 3 çeşit modülasyon vardır. Her modülasyon prosedüründe simetri olduğundan Şekil 2.19'da görüldüğü gibi taşıyıcının etrafında yan bandlar üretilir [2].



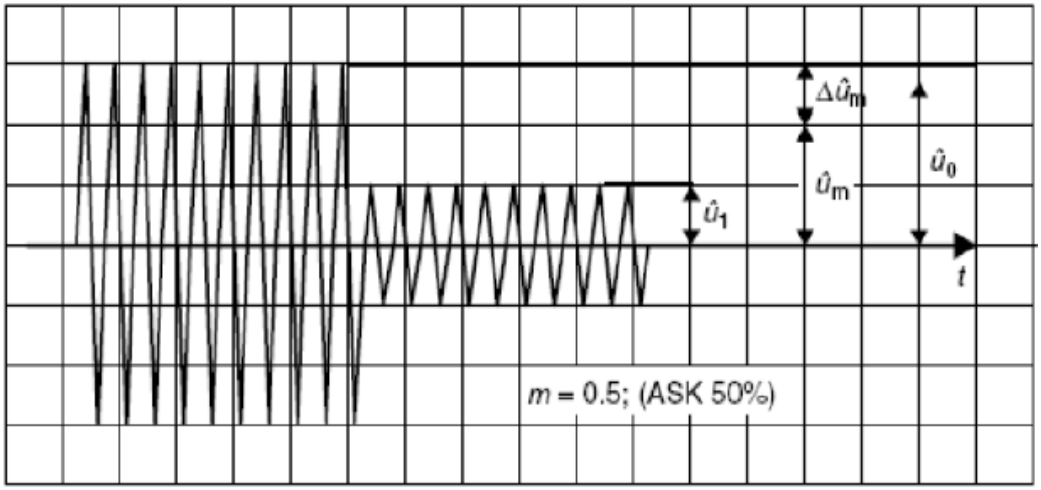
Şekil 2.19. Sinüzoidal Sinyal Modülasyonu (Taşıyı ve Yan Bandlar)

Genlik Kaydırmalı Anahtarlama (ASK): Bu Modülasyon tipinde, taşıyıcı işaretin genliği iki veya daha fazla değer arasında anahtarlanır. İkili durumunda genellikle var-yok anahtarlama kullanılır. ASK dalga biçimi "0" için boşluk, "1" için RF dalgalardan oluşur. ASK işareti Şekil 2.20' deki gibidir.



Şekil 2.20. Mesaj (bilgi) İşareti ve ASK İşareti

Genlik kaydırmalı anahtarlama modülasyonunda, taşıyıcı salınımının genliği u0 ve u1 durumlarında ikili kod sinyali ile anahtarlanır. "u1"; u0 ve 0 arasında değer alır. u0, u1 oranı "m", çalışma faktörü olarak bilinir. Bu durum Şekil 2.21'de görülmektedir.



Şekil 2.21. İkili Kod Sinyali ile İki Durumda Anahtarlanan Taşıyıcı Genliği

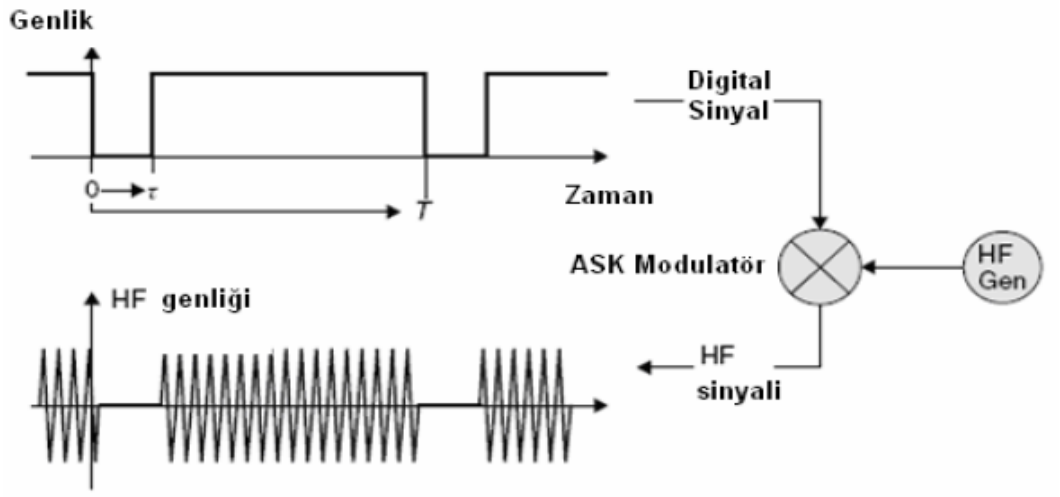
"m" çalışma faktörünü bulabilmek için, anahtarlanmış ve anahtarlanmamış taşıyıcı sinyal genliğinin aritmetik ortalaması hesaplanır.

$$\hat{u}_m = \frac{\hat{u}_0 + \hat{u}_1}{2} \tag{2.10}$$

Daha sonra û0-ûm genlik değişiminin, elde edilen ûm değerine oranı ile "m" çalışma faktörü, bulunur.

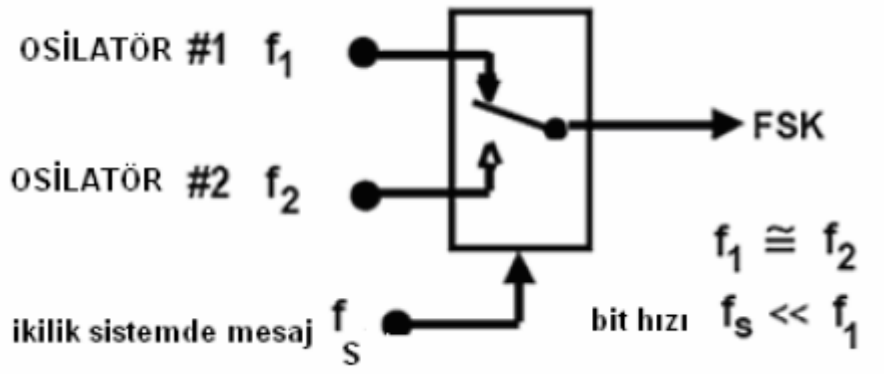
$$m = \frac{\Delta\hat{u}}{\hat{u}_m} = \frac{\hat{u}_{0m} - \hat{u}_m}{\hat{u}_m} = \frac{\hat{u}_0 - \hat{u}_{1m}}{\hat{u}_0 + \hat{u}_1} \tag{2.11}$$

Şekil 2.22'de ikili kod sistemi kullanan ASK modülatörü, HF generatöründen üretilen sinuzoidal taşıyıcı sinyali kullanarak HF sinyalini üretir.



Şekil 2.22. ASK Modülasyonu Üretimi

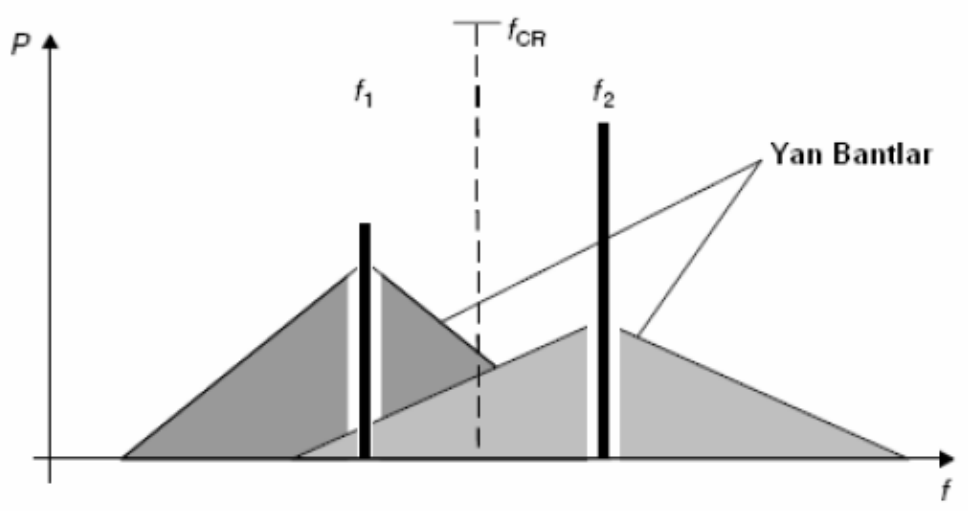
Frekans Kaydırmalı Anahtarlama(FSK): FSK modülasyonunda, taşıyıcı işaretin ani frekansı, sayısal işarete bağlı olarak iki veya daha çok değer arasında anahtarlanır. FSK işaretin üretimi Şekil 2.23 ‘de gösterilmiştir.



Şekil 2.23. FSK Üretimi

İki FSK modülasyonunun spektrumu, f1 ve f2 frekanslarının genlik kaydırmalı anahtarlama osilasyonlarının bireysel spektrumlarının eklenmesiyle elde edilir.

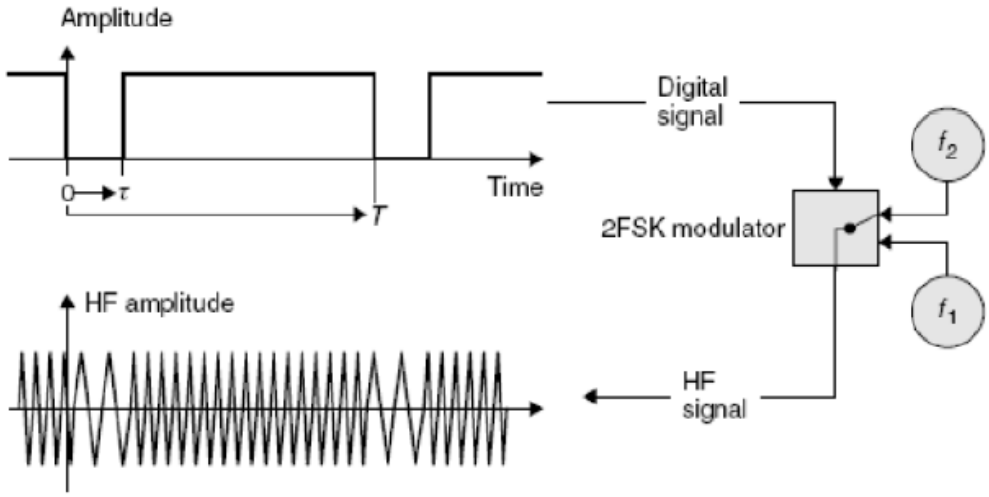
Taşıyıcı frekansı fCR, f1 ve f2 karakteristik frekanslarının aritmetik ortalaması alınarak hesaplanır. Karakteristik frekanslar ve taşıyıcı frekans arasındaki fark frekans sapması olarak adlandırılır. Bu durum Şekil 2.24‘de görüldüğü gibidir.



Şekil 2.24. FSK Modülasyonu

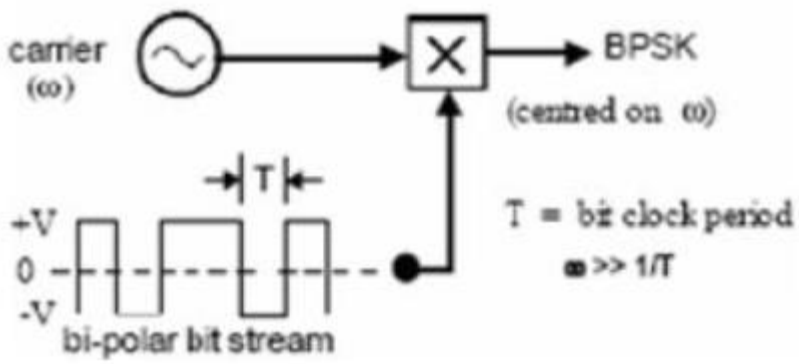
$$f_{CR} = \frac{f_1+f_2}{2} ; \Delta f_{CR} = \frac{|f_1+f_2|}{2} \quad (2.12)$$

İkili kod sinyali ile f1 ve f2 frekansları arasındaki anahtarlama ile iki FSK modülasyonunun üretimi Şekil 2.25'de gösterilmiştir.



Şekil 2.25. FSK Modülasyonu Üretimi

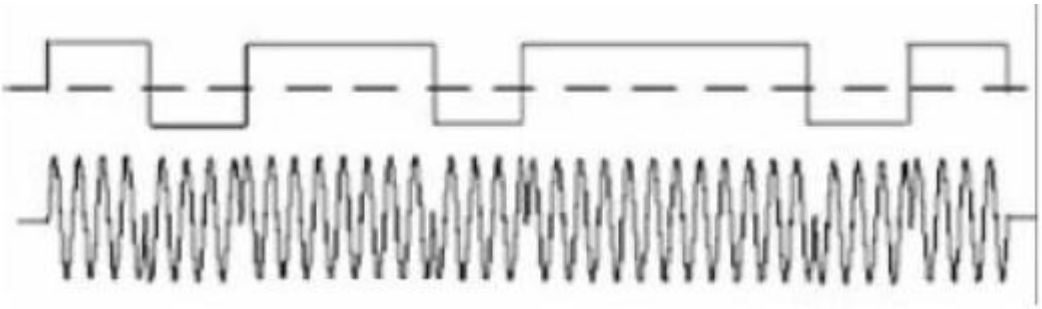
İki faz Kaydırmalı Anahtarlama(BPSK): Tek bir taşıyıcı frekansı için iki çıkış faz söz konusudur: (0 ve Π) veya (2 Π ve 23 Π). Girişteki sayısal işaret değiştikçe taşıyıcının fazı iki açı değeri arasında kayar. Eğer sinüzoidal taşıyıcı iki durumlu bit dizisi tarafından modüle edilecekse çıkış işaretinin polaritesi bit dizisinin olaritesinin değiştiği noktada değişecektir. Bunu Şekil 2.26'de görmek mümkündür.



Şekil 2.26. BPSK İşaretinin Üretilmesi

Bit dizisi, hakkında bilgi gönderilecek BPSK işaretinin fazının değişimlerini içermektedir. Bir senkron demodülatör bu faz değişimlerine duyarlı olması gerekmektedir.

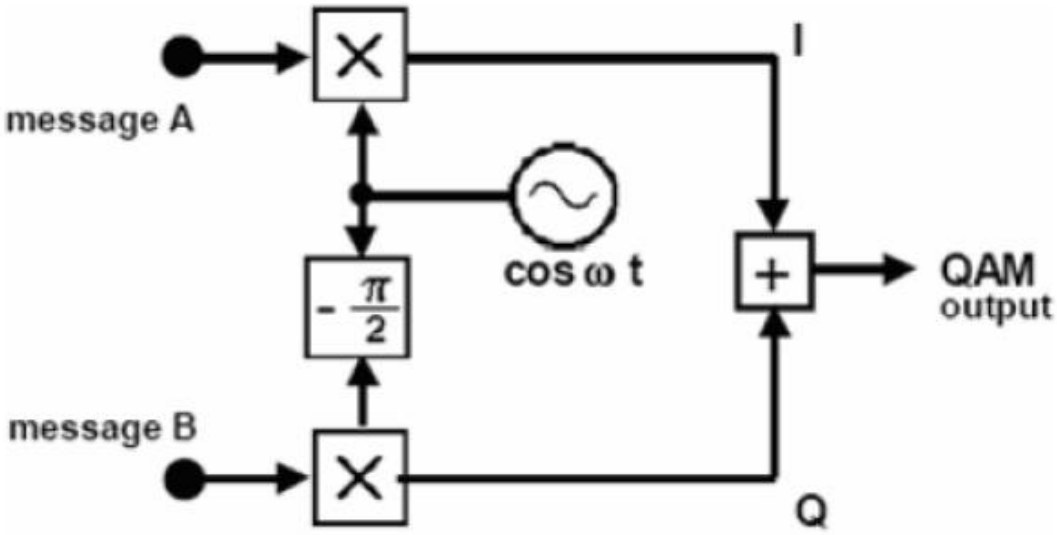
Şekil 2.27'deki, üstteki işaret ikili bilgi işaretidir. Dalga şekli her faz değişikliğinde simetrik bir yapı oluşturmaktadır. Bunun sebebi bit hızının taşıyıcı frekansının alt katları olmasıdır. Bu normalde özel bir durum olarak bakılabilir ve her zaman pratikte gözlemlenemeyebilir. Bu durum bize alınan işaretten kolay bir şekilde bit dizisinin elde edilmesini sağlamaktadır. Verimli haberleşme sağlamak için bir band sorunu olabilir, bu sorun (band sınırlanması) ise ya temel-bandda ya da taşıyıcı frekansında halledilmektedir.



Şekil 2.27. BPSK İşareti

a. Çeyrek faz kaydırmalı anahtarlama(QPSK):

Bu modülasyonda birbirinden bağımsız iki tane A ve B analog işaret modülatörün girişine uygulanmaktadır. Modülatör önceleri QAM modülatörü olarak adlandırılmıştır. Daha sonra QPSK modülatörü olarak adlandırılmaya başlanmıştır. Bu modülatör Şekil 2.28' da görüldüğü gibidir.



Şekil 2.28. QAM(QPSK) Modülatörü

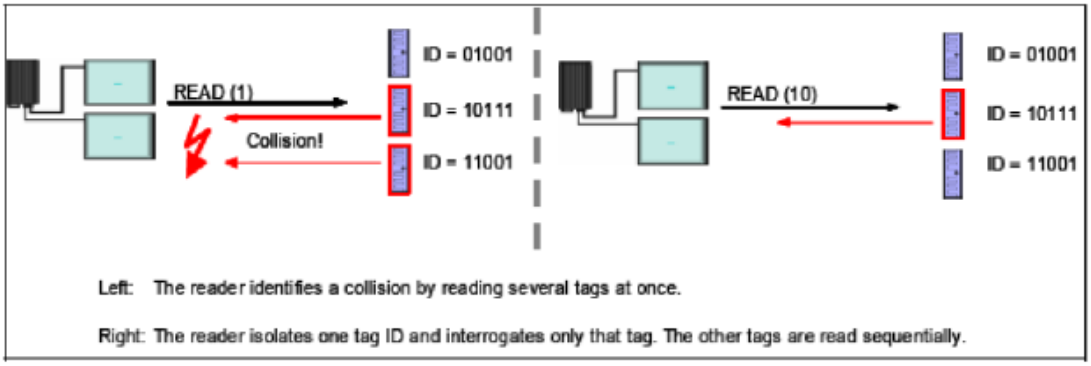
2.5 Çarpışma ve Çarpışma Önleyici Algoritmalar

RFID uygulamalarında, sistemin bazı fiziksel özelliklerinden dolayı da sorunlar ortaya çıkabilir. Bunlardan biri farklı okuyuculardan gönderilen sinyallerin çakışması diğeri de çok fazla sayıda etiketin okuyucunun etki alanına girmesidir. Ayrıca RFID sisteminin kurulduğu ortam da önem taşımaktadır. Örneğin yüksek frekanslı dalgalar su içinde absorbe olurken, düşük frekanslı dalgalar da metal nesnelerden etkilenmektedir. Bu nedenlerden dolayı sistemin tasarımı, uygulama başarısı için çok büyük önem taşımaktadır. "Çarpışma" farklı yönlerden gelen Radyo dalgalarının birbiri ile karışması olarak tanımlanabilir. RFID'deki bir problemde okuyucu çarpışması ve etiket çarpışmalarıdır.

2.5.1 Okuyucu Çarpışması

Okuyucu çarpışması, bir okuyucudan gelen sinyalin diğerinden gelen ile karışmasıdır. Bu problemi çözmenin bir yolu zamanı birçok geçiş için bölmektir. Bu basit olarak okuyucuların farklı zamanlarda etiket ile iletişim kurmasıdır. Bu birbirleri ile çatışmalarını engeller. Ancak bu aynı zamanda iki okuyucunun

çakıştığı bir yerde bir RFID etiketinin iki defa okunması anlamına da gelebilir. Bu yüzden sistem, bir etiket bir okuyucu tarafından okunduğu zaman diğer okuyucunun tekrar okumaması şeklinde kurulmalıdır. Bu durum Şekil 2.29'de görüldüğü gibi çözülmüştür.



Şekil 2.29. Okuyucu Çarpışması Protokolü

2.5.2. Etiket Çarpışması

RFID etiketi çarpışması aynı anda birden fazla etiket okuyucuya sinyal gönderdiği zaman meydana gelir. Üreticiler etiketin okuyucuya tek bir anda cevap vermesi için değişik sistemler geliştiriyorlar. Bu sistemler etiketleri tekilleştiren algoritmaları içerir. Her etiket saniyenin binde birinde okunduğu için, eş zamanlı okunuyorlarmış gibi görünür.

2.5.3 Çarpışma Önleyici Algoritmalar

Çarpışma önleyici algoritmalar, olasılıksal ve belirleyici algoritmalar şeklinde iki sınıfta ifade edilebilir.

Olasılıksal algoritmalarda (asenkron olarak da bilinir), etiket rastgele üretilmiş zamanlarda cevap verir. Olasılıksal algoritmaların birçok çeşidi vardır. Birçoğu, ağ üzerindeki ALOHA metodu tabanlıdır. Bu yöntem, bir veri paketi aldıktan sonra bir başka veri paketi iletilen noktalar içerir. Eğer bir çarpışma olursa, bu düğüm donar ve yeni bir paketi gecikmeli gönderir. Etiket üzerinden okuyan okuyucu, çarpışma olmadığı sürece data transferini sürdürür. Okuyucunun

cevap verdiği anlar, aralıklı (slotted) veya sürekli anlar olabilir. Bu mod özel veri paketlerini bağımsız transmisyonunda biraz kısıtlama yapmaktadır. Eğer, aralıklı anlarda (slotted ALOHA) modunda paket çarpışması oluştuysa, veri paketleri tamamen üst üste biner ve bu veri transferini şişirir.

Belirleyici algoritmalarda (senkron olarak da bilinirler), okuyucu, etiketlerin kendilerine has olan kimlik numaralarını tarar. En basit belirleyici algoritma çeşidi ikili ağaç/yürüyen ağaç (binary tree/tree-walking) yöntemidir. Bu yöntemde okuyucu, ağacı olası tüm kimlik numaraları için tarar. Ağaçtaki her düğüm noktasında okuyucu cevabı sorgular. Gelen cevap, bir sonraki okuma işlemini nerede yapacağının işaretini verir.

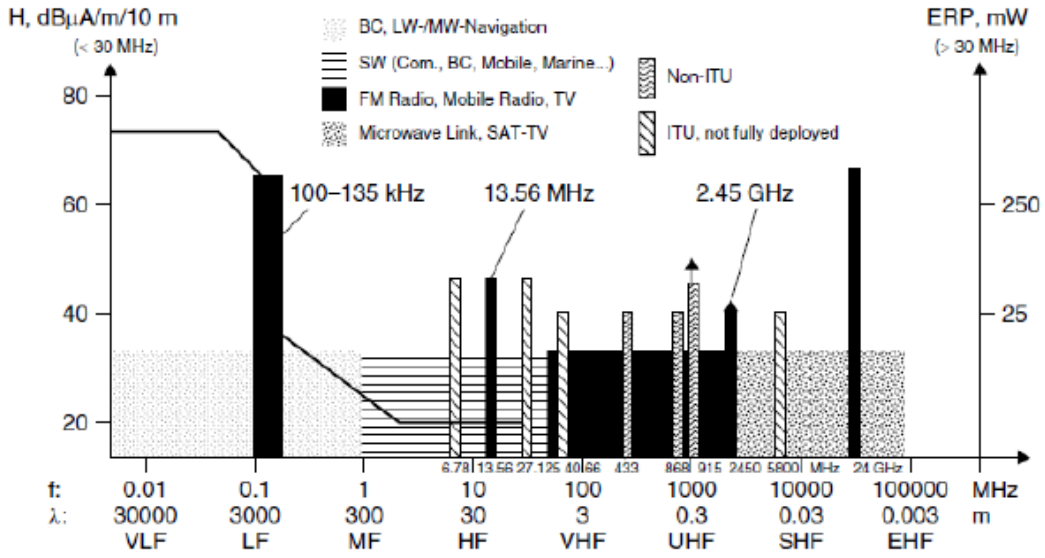
Bunlardan başka iki ayrı çarpışma önleyici algoritma çeşidi vardır; FMO ve Miller Subcarrier (alt taşıyıcı). FMO, günümüzde de ISO standartlarında kullanılmaktadır. Bu algoritma hızlı ve engellere karşı daha dayanıklıdır. Miller Subcarrier daha yavaştır ancak gürültülü ortamlarda RF de daha iyidir ve 2. nesil okuyucular tarafından desteklenmektedir. Bu algoritma, etiketlerin geri sinyallerini göndermek üzere dar bir band spektrumu kullanır ve bu sinyalleri okuyucu tarafından kullanılan kanallara uyarlar. Böylelikle okuyucudan gelen RF sinyalleri, etiketten gelen sinyallerle karşılaşmaz. Miller Subcarrier (alt taşıyıcı) algoritması, etiketin cevabını, okuyucunun transmisyon sinyallerinden ve diğer gürültülerden ayırmak için, FMO'ya nazaran daha gelişmiş filtreleme teknikleri kullanır.

2.6 Uygulama Frekansları

RFID sistemleri elektromanyetik dalgalar oluştururlar ve bunları yayarlar. Bu sebeple, frekans aralıkları kullanırlar ve yasal olarak radyo sistemleri olarak sınıflandırılırlar. RFID sistemlerinin çalışması diğer radyo hizmetlerini hiçbir koşul altında etkilememelidir ya da bozmamalıdır. Özellikle polis, güvelik hizmetleri, deniz ve havacılığa ait radyo hizmetleri, radyo ve televizyon servisleri ve mobil telefon hizmetleri ile karışmadığını garanti etmek için özel önlemler alınmalıdır.

Diğer sistemleri rahatsız etmeden RFID için kullanılabilir frekans aralıkları

Şekil 2.30'de gösterilmiştir. Bu sebeple, genellikle sadece endüstriyel, bilimsel ve ya tıbbi uygulamalar için özel olarak ayrılan frekans aralıklarını kullanmak mümkün olmuştur. Kullanılan bu frekanslar dünya çapında ISM(industrial-Scientific- Medical) olarak sınıflandırılan frekanslardır [4].



Şekil 2.30. Pratikte Kullanılan RFID Frekansları

ISM frekanslarına ek olarak, Amerika‟da 135Khz den, Japonya‟da da 400Khz den küçük frekanslarda yüksek manyetik alan kuvvetleriyle çalışmak mümkün olduğu için uygun görülmüştür. RFID için en önemli frekans aralıkları, 0-135khz, 6.78MHz civarındaki ISM frekansları(henüz Almanya‟da müsait değil), 13.56MHz, 27.125MHz, 40.68MHz, 433.92MHz, 869.0MHz, 915.0MHz (Avrupa‟da değil), 2.45GHz, 5.8GHz ve 24.125GHz dir. Ancak dünyada en yaygın kullanılanları, düşük frekans (125 KHz civarı), yüksek frekans (13.56 MHz) ve çok yüksek frekans ya da UHF (860–960MHz)‟dir. Ayrıca 2.45 GHz ve 5.8 GHz mikrodalgalar da bazı uygulamalarda kullanılmaktadır[2]. Radyo dalgaları farklı frekanslarda farklı özellikte olduklarından, uygulama için uygun frekansın seçilmesi gereklidir. Frekans badlarının listesi Çizelge 2.2'de verilmiştir.

Çizelge 1.2. Frekans Bantları

Band	Frekans Aralığı	Uygulama
Alçak Frekans (LF)	120-135 KHZ	Kısa mesafe uygulamalar
Yüksek Frekans (HF)	13,56 MHZ	Akıllı kartlar ve etiketler için kullanılan frekans
Ultra Yüksek Frekans (UHF)	433 MHZ	Aktif düşük güçlü etiketler
	860-960 MHZ	Tedarik zinciri uygulamaları
Mikrodalga	2450 MHZ	Aktif etiketlerde daha yüksek haberleşme mesafeleri ve daha yüksek haberleşme hızları

Frekans aralıkları Çizelge 2.3'de görüldüğü gibi değişik karakteristik özelliklere sahiptirler.

Çizelge 2.3. Frekans Bantlarının Özellikleri

Alçak frekans - (Low Frequency -LF <135 KHz)
Tipik olarak pasif, sadece okunabilir veya okunur-yazılır etiketler kullanılır. • Etiketler ucuz ama anten açısından sistem uzun ve pahalı bir bakır antene ihtiyaç duyar. • Metal ve sıvıların performansı düşüren etkilerinden en az etkilenilen frekanstır. • LF kısa okuma mesafesine ve düşük okuma hızına sahiptir. • Diğer frekanslara nazaran daha geniş boyutlara sahip etiket kullanılır.
Yüksek Frekanslar – (High Frequency -HF - 13.56 MHz)
Tipik olarak pasif, sadece okunabilir, okunur-yazılır veya bir kere yazılıp çok kez okunabilen WORM etiketler kullanılır. • Uzun okuma menziline ihtiyaç duyamayan çoklu etiket uygulamaları için iyi uyumlu
Çok yüksek Frekanslar – (Ultra-High Frequency UHF- 868 MHz- 915 MHz)
Aktif veya pasif sadece okunabilir, okunur-yazılır veya bir kere yazılıp çok kez okunabilen WORM etiketler kullanılır. • HF –Yüksek frekanslara nazaran daha yüksek menzil kapasitesi, daha fazla veri transferi ve hızlı tanımlama imkânı sunar. • Özellikli çoklu etiket okumada mesafe ve performans arasında iyi bir denge sağlar.
Mikrodalga -Microwave (2.45 GHz, 5.8 GHz)
Aktif veya pasif sadece okunabilir, okunur-yazılır veya bir kere yazılıp çok kez okunabilen WORM etiketler kullanılır. • UHF etiketlerine benzer karakter göstermekle beraber daha hızlı okuma oranına sahiptir. • Maliyeti alçak frekanslarınkinin bazen iki katı olabilmektedir.

Tabloyu inceleyecek olursak, örneğin düşük frekanslı etiketler daha az güç kullanırlar ve metal olmayan cisimleri algılamada daha iyidirler. Meyve gibi yüksek su içeren cisimlerde idealdirler, ancak okuma kapasiteleri düşüktür (0.33metre). Yüksek frekanslı etiketler metal cisimlerde daha iyidirler ve su içeren cisimler için de kullanılabilirler. 1 metreden çok rahat okunabilirler. UHF frekansı daha yüksek okuma kapasitesine sahiptir ve düşük ve yüksek frekanslara göre veri aktarımını daha hızlı yaparlar. Ancak çok fazla güç kullanırlar ve cisimle aralarında görüş engeli olmamalıdır. Bu yüzden etiket ile okuyucu arasında net bir iletişim yolu olmalıdır. UHF frekanslı etiketler bir deponun kapısından girişi yapılan kutuların taranması için daha iyidir. Uygulama için doğru frekansı seçilmesi çok önemlidir [28]. Günümüzde en sık kullanılan ve pazar tarafından en çok talep edilen frekans 13.56MHz ‘dir.

3. WEB TABANLI YAZILIM

İstemci-sunucu mimarisine dayanan uygulamalarda bir sunucu yazılım ve bu sunucu uygulamaya iş yaptıran istemci yazılımlar bulunmaktadır. Bu sunucu ve istemci yazılımlar yazılım ekipleri tarafından geliştirilmekte, sunucu yazılımlar sunucu bilgisayarlara kurulurken, istemci yazılımlar uygulamayı kullanacak tüm kullanıcıların bilgisayarlarına teker teker kurulmaktadır. Sunucu tarafında yapılacak bir değişiklik muhtemelen istemci yazılımların da buna göre güncellenerek değiştirilmesini gerektirecek ve uygulamaya erişen her yazılım yenisi ile değiştirilecektir.

Bu tip uygulamaların başlıca problemleri aşağıdaki gibi listelenebilir;

- Uygulamayı kullanacak her kullanıcı için istemci yazılımın kurulması gerekmektedir,
- İstemci yazılımların test edilmesi, kullanıcı bilgisayarlarına kurulması ve kurulu yazılımların güncellenmesi zahmetli ve zaman alıcı bir işlemdir,
- Teknik destek maliyeti yüksektir,
- İstemci tarafında eski yazılımların hala kullanılıyor olması, uygulama işleyişinde istenmeyen problemlere yol açacağından tüm istemci yazılımların güncellenmesi gerekmektedir. Örneğin yazılım ekibi uygulama tarafında gerekli kontrolleri yaptırmıyorsa, güncellenmemiş bir yazılım eksik veya hatalı veri göndererek veritabanında bulunan veri yapısının bozulmasına sebep olabilir,
- İstemci yazılımlar, standart olmayan kullanıcı bilgisayarlarında farklı problemlere yol açabilmektedir. Bu nedenle yazılım test maliyeti yüksektir. Örnek olarak; bir Microsoft Windows işletim sistemi kurulu bilgisayarda bölgesel ayarların farklı olması, gerekli yardımcı dosyaların sistemde bulunmaması, farklı sistemlerde var olan yardımcı dosyaların sürüm farklılıkları, işletim sistemlerinin farklı olması, veya aynı işletim sistemlerinin farklı sürümlerinin kullanılması verilebilir.

Web platformunun gelişmesi ve Internet'in sık kullanılır hale gelmesiyle birlikte bu uygulamalar, yerlerini web tabanlı uygulamalara bırakmaktadır. Bazı uygulamalar tamamen ortadan kalkmasa bile, çeşitli bölümlerine olan erişim, geliştirilen web tabanlı uygulamalar aracılığıyla sağlanmaktadır.

Web tabanlı uygulamalar organizasyonlara büyük kolaylıklar sağlamaktadır. Ancak organizasyonların bilgi güvenliği riskleri açısından bakıldığında, bir takım riskleri de beraberinde getirdiği görülmektedir. Gün geçtikçe uygulamalara erişimlere sağlanan bu kolaylıklar ve uygulamalara Internet dahil çeşitli yerlerden erişiliyor olması, bu risklerin gerçekleşme olasılığını arttırmaktadır. Yaklaşık olarak "saldırıların %75'inden fazlasının web protokolleri üzerinden olduğunu ve web uygulamalarının %90'ında en az bir güvenlik problemi olduğunu" [14] düşündüğümüzde, uygulamaların kullanıma açılmadan önce mutlaka güvenlik problemlerine karşı kontrol edilmesi gerektiği açıkça görülebilir.

Oluşabilecek güvenlik risklerini ortadan kaldırmak için yapılan güvenlik denetimleri, uygulama geliştirme, iyileştirme ve bakım süreçlerinin bir parçasıdır. Bu sayede, kullanıma açılan uygulamanın bilinen güvenlik problemlerinden etkilenmediği ve bilginin gizlilik, bütünlük ve erişilebilirliğinin sağlandığından emin olunmaktadır.

Web uygulamalarına uygulanan güvenlik denetlemelerinin insanlar tarafından yapılması uzun bir süreç olduğundan, bu güvenlik testlerini otomatikleştirme amacıyla, otomatik güvenlik denetim yazılımları oluşturulmuştur. "Web uygulamaları güvenlik denetimlerinin otomatikleştirilmesi ilk olarak 1999 yılında resmi kayıtlara geçmiştir" [14].

Otomatik güvenlik denetim yazılımları, güvenlik açıklarının tespit edilmesi sırasında harcanan emeği azaltmakta ve zaman açısından katkıda bulunmaktadır. Ancak el ile yapılan testlerle karşılaştırıldığında, bazı problemleri ve eksiklikleri olduğu görülmüştür. El ile ve otomatik olarak yapılan güvenlik denetimleri arasındaki farkı görme amacıyla, Watchfire firması tarafından 100 web sitesi üzerinde yapılan bir analizde [14];

- %36'sında, el ile ve otomatik olarak yapılan güvenlik denetimleri arasında

bir fark görülmemiş,

- %17'sinde otomatik güvenlik denetimleri hiç sonuç vermezken, el ile yapılan denetimlerde güvenlik problemleri tespit edilebilmiş,
- %46'sında ise, tüm güvenlik problemlerini tespit edebilmek için otomatik denetimlere ek olarak, el ile denetim yapmak gerekmiştir.

Bu analizler ve tecrübeler sonucunda, otomatik güvenlik denetim yazılımlarının iyileştirilmesi gerektiği görülmektedir. Bu tez çalışmasında, otomatik güvenlik denetim yazılımları incelenerek iyileştirilmeleri için öneriler yapılmış ve bu önerileri gerçekleme amaçlı bir uygulama çalışması hazırlanmıştır. Tezin bundan sonraki bölümlerinde, "web uygulamaları otomatik güvenlik denetim yazılımları" yerine "denetim yazılımları" ifadesi kullanılacaktır.

3.1 Web Tabanlı Uygulamalar

Kullanıcılara bir web sunucusu vasıtasıyla sunulan uygulamalar, web tabanlı uygulamalar olarak adlandırılmaktadır. Çalışma modeli olarak çok katmanlı yapıya sahip olan web uygulamaları, genelde üç katmanlı yapıya sahiptirler. 1.katmanda uygulamaya erişen web istemcileri, 2. katmanda web uygulamasının kendisi ve 3. katmanda da verilerin saklandığı veritabanı bulunmaktadır denilebilir [15].

Web tabanlı uygulamalara olan erişim bir ağ üzerinden, web istemcileri kullanılarak gerçekleşmektedir. Web istemcileri uluslararası standart olan HTML dilini işleyebilen yazılımlardır. Bu yazılımlar günümüz bilgisayarlarının ve diğer cihazların neredeyse hepsinde bulunmaktadır.

Web tabanlı uygulamalar, web istemcileri tarafından işlenebilen ve görüntülenebilen web sayfaları üretmektedir. Basitçe ele alınacak olursa, yaptıkları işlemler veritabanından alınan verileri işleyerek kullanıcıya göstermek ve kullanıcı tarafından değiştirilen veya gönderilen verileri belirli bir düzende veritabanına yerleştirmektir denilebilir.

Web tabanlı uygulamalara örnek olarak; bankaların Internet şubeleri, çeşitli kurumların Internet üzerinden erişilebilen fatura ödeme uygulamaları, bilgi paylaşımında kullanılan forumlar, üniversite not görüntüleme ve derslere kayıt

sistemleri gösterilebilir.

Web tabanlı uygulamaların kullanımı, özellikle Internet'in yaygın kullanımı ve web platformunun gelişmesi ile birlikte artmış ve standart istemci-sunucu mimarisine dayanan uygulamalar yerlerini web tabanlı uygulamalara bırakmıştır.

Bunun başlıca nedenleri, yani web tabanlı uygulamaların diğer uygulamalara göre avantajları:

- Web tabanlı uygulamaların farklı işletim sistemlerinde kullanılabilmesi,
- Gerekli standart yazılımların dışında kullanıcı tarafında ek bir istemci yazılım kurulumu gerektirmemesi,
- Yazılım güncellemelerinin ve bakımının kolaylığı,
- Yazılımda yapılan bir değişiklik sonrası kullanıcı tarafında bir güncelleme gerektirmemesi,
- Uygulamalara olan erişimin kolaylığı, dolayısıyla iş ortaklarının ve Internet üzerinden tüm dünyanın erişimine ve kullanımına kolayca açılabilmesi,
- İstemcilerde depolanan verileri azaltıldığından, veri kayıplarının azaltılması,
- Uygulamanın thin-client'lar dahil, güçsüz donanıma sahip bilgisayarlardan kullanılabilmesi,
- Bir web istemcisi olan herhangi bir yerden kullanılabilmesi, bu sayede istemci değiştirme kolaylığı da sağlaması, olarak sayılabilir.

Web tabanlı uygulamaların diğer uygulamalara göre dezavantajları ise:

- Web sayfalarında görüntülenen bileşenler üzerinde sınırlı kontrol imkanı,
- Performans problemleri,
- Web tabanlı olmayan uygulamalarla veya bir donanımla entegrasyon zorluğu,
- Belirli tipte uygulamaların web üzerinde yapılamaması (Örneğin tasarım veya grafik düzenleme yazılımları),
- Web erişim protokolü HTTP'nin getirdiği bazı güvenlik problemleri, olarak sayılabilir.

3.1.1. Web İstemcisi

Web istemcileri, kullanıcıların web servislerine erişmesini ve web sunucuları tarafından sağlanan içeriğin, HTML standartlarında işlenerek görüntülenmesini sağlayan programlardır. Web istemcileri sunucu tarafından gönderilen özel bileşenlerin kullanıcı tarafında işlenmesi görevini de üstlenmektedirler.

Web istemcilerinin pek çok alternatifi bulunmaktadır. Sıkça bilinen web istemcilerine örnek olarak Mozilla Firefox, Microsoft Internet Explorer, Netscape Navigator, Google Chrome ve Opera gibi motorlar verilebilir.

Web sunucularına istek yapabilmek için mutlaka bu programların varlığı gerekmez. HTTP protokolü standartlarını bilen bir kişinin web servisine bağlanarak yapacağı istekler de, web sunucuları tarafından işlenerek cevaplandırılacaktır [16].

3.1.2 Web Sunucusu

Web sunucuları, istemciler tarafından gönderilen HTTP isteklerini işleyerek, istemcilere HTTP yanıtları gönderme işlemi yapan servislerdir. İstemciler sunuculara HTTP protokolü standartlarında istek yapmaktadırlar. Aynı şekilde web sunucuları, istemcilere HTTP protokolü standartlarına göre yanıt verirler. İstemcinin istediği bilgiye göre sunucu tarafından istemciye sağlanan içerik, statik ve dinamik içerik olmak üzere ikiye ayrılabilir [16].

a. Statik İçerik:

Sunucu üzerinde duran bir dosya, sistemden alınarak istemciye direk gönderiliyorsa, bu içerik statik içerik olarak adlandırılır. Kullanıcıya gönderilen bu dosyalar web sunucusu tarafından herhangi bir ek işleme tabi tutulmazlar. Bu dosyalar genelde HTML, düz metin veya resim formatında olan dosyalardır.

HTML web sayfaları oluşturmak için tasarlanmış bir biçimleme dilidir [16]. Biçimleme dilleri, yazı ve yazı ile ilgili ek bilgiler içerir. Bu ek bilgiler,

yazının biçimlendikten, işlendikten sonra nasıl gözükeceği bilgisini barındırır.

HTML dili ile oluşturulan dosyalar, aslında belirli standartlardaki etiketlerle oluşturulmuş düz metin dosyalarıdır. Bu dosyalar, HTML dilini işleyebilen web istemcileri tarafından yorumlanarak kullanıcıya HTML dilinde belirtilen biçimde gösterilir. HTML formatında oluşturulmuş bir sayfa; yazı, grafik ve diğer biçimlerdeki bilgilerin bir arada görüntülenmesini sağlamaktadır. HTML dili ise bu bilginin bileşenlerini belirleyerek, HTML dosyası işlendiğinde kullanıcı tarafında sayfanın nasıl görüleceğini tanımlar.

b. Dinamik içerik:

Sunucu veya kullanıcı tarafında işlendikten sonra son haline ulaşan içerik, dinamik içerik olarak adlandırılır. Kullanıcı tarafında işlenen dinamik içerik, kullanıcının bilgisayarında işlenerek üretilir. Bu içerikte web sunucusunun herhangi bir rolü yoktur. Sunucu, önceden oluşturulmuş sayfayı, aynı statik içerikte olduğu gibi kullanıcıya gönderir. Kullanıcının kullandığı web istemcisi, sunucudan alınan bu sayfa içerisindeki kodu işleyerek kullanıcıya sayfanın son halini göstermektedir. Kullanıcı tarafında çalışacak bu kod için genelde JavaScript programlama dili kullanılmaktadır.

Sunucu tarafında işlenen dinamik içerikte ise, web istemcisinin yaptığı HTTP isteğine göre, web sunucusu dinamik içeriğin bulunduğu dosyayı harici bir programla işleyerek oluşan çıktıyı kullanıcıya göndermektedir [17].

Web uygulamalarının önemli bir parçasını oluşturan sunucu tarafında işlenen dinamik sayfalar, web sunucusunun desteklediği programlama dilleri ile hazırlanmaktadır.

Genelde web uygulamalarını geliştirirken kullanılan programlama dilleri PHP, ASP.NET, ASP, JSP ve Perl (Bkz. Kısaltmalar Tablosu) olarak sayılabilir. Sadece bunlar değil, web standartlarına uyacak şekilde çıktı üreten herhangi bir yazılım da web uygulaması olarak kullanılabilir. Ancak web uygulamaları geliştirilmesinde kullanılan programlama dilleri, içlerinde gelen fonksiyonların ve kütüphanelerin çokluğu nedeniyle web uygulaması geliştirme işlemlerini kolaylaştırmaktadır [17].

3.1.3 HTTP Protokolü

Web erişim protokolü HTTP, Web üzerinde bilginin taşınması ve iletilebilmesi için kullanılan ana protokoldür. İstemcinin web sunucusuna bağlanarak yaptığı isteklere HTTP isteği, sunucunun bu isteğe karşı olan cevabına ise HTTP yanıtı denilmektedir. HTTP istekleri ve yanıtları, HTTP sürüm numaraları içermektedir. HTTP'nin bugün kullanılan sürümü HTTP/1.1'dir [18].

Durum korumalı (stateful) protokollerde sunucular, kendilerine bağlanan istemciler hakkında bir durum bilgisi tutmaktadırlar. Bu bilgi kullanıcı durumunun korunması, kullanıcının hala ilk bağlanan kullanıcı olup olmadığının doğrulanabilmesi ile ilgili bilgidir.

Durum korumasız (stateless) bir protokol olan HTTP protokolü, istemciler için bir durum bilgisi tutmamaktadır. Yani web sunucuları bir istemci tarafından yapılan daha önceki istekleri hatırlamaz, her bir istek yeni/ayrı bir istekmiş gibi işlenerek yanıtlanır. Bu nedenle web uygulamaları durum bilgisini kendileri tutmak zorundadır (Ör: Sisteme giriş yapmış kullanıcının uygulamayı kullanırken tanınması).

Web uygulamaları bu işlemi yapabilmek için kullanıcılara bazı tanıtıcı bilgiler gönderirler. Kullanıcılar tarafından kullanılan web istemcileri de uygulamadan aldıkları bu tanıtıcı bilgileri, daha sonradan yaptıkları HTTP istekleri sırasında göndererek uygulama tarafından tanınırlar. Kullanıcılara atanan bu bilgiler tanımlama bilgisi (Cookie) olarak adlandırılmaktadır. Bu tanımlama bilgilerinde ek olarak; tanımlama bilgisinin geçerlilik tarihi, hangi alan adlarında ve hangi klasörlerde geçerli olacağı ve sadece güvenli bağlantı üzerinden gönderilmesi gerektiği bilgisi de bulunabilir [18].

Önceleri web uygulamaları tarafındaki kullanıcı yönetimi, sadece bu tanımlama bilgileri ile yapılmaktaydı. Örneğin web uygulamalarında kimlik doğrulaması yapıldıktan sonra uygulama, kullanıcıya tanımlama bilgisi olarak; kullanıcı numarası, kimlik doğrulamasından geçtiğine dair bilgi ve varsa uygulama içerisinde kullanıcının özel ayarlarıyla ilgili bilgiler göndermekteydi. Kullanıcı tarafından kullanılan istemci, bu bilgileri web sunucusuna yaptığı her istekte göndererek uygulama tarafından tanınmaktaydı.

Bir istemcinin tanımlama bilgilerini ele geçirebilen bir saldırgan, bu bilgileri yaptığı HTTP isteklerinde göndererek, web uygulaması tarafından tıpkı uygulamayı kullanan geçerli bir kullanıcı gibi algılanabilir. Ayrıca belirli bir zaman sonra bu bilgilerin otomatik olarak iptal olması gibi bir durum söz konusu değildir. Bu gibi güvenlik problemleri nedeniyle tanımlama bilgileri yerlerini oturum bilgilerine bırakmışlardır.

Oturum bilgisi (session information) aslında kullanıcılara atanan tek bir tanımlama bilgisidir. Yani web uygulaması, oturum bilgisi adı altında, kullanıcıya rastgele oluşturulmuş bir tanımlama bilgisi değeri gönderir. Kullanıcı bu tanımlama bilgisini her zaman olduğu gibi yaptığı tüm isteklerde sunucuya gönderecektir.

Bu tanımlama bilgisi aynı zamanda sunucu tarafında da tutulmakta, kullanıcı hakkındaki bilgiler sunucu tarafında bu tanımlama bilgisi ile saklanmaktadır [18].

Tanımlama bilgilerinde bulunan riskler oturum bilgilerinde de aynı şekilde bulunmaktadır. Ancak uygulamayı geliştiren ekip oturum bilgilerinde kullanıcıya ait özel bazı değerleri bulunduruyorsa, saldırganlar tarafından yapılabilecek yerine geçme saldırılarından kendilerini korumuş olurlar. (Örneğin uygulamaya bağlanan kullanıcının IP adresinin tutulması, Kullanıcının tam istemci bilgisinin tutulması vb.) Ayrıca oturum bilgilerinin zaman yönetimi, web uygulamalarının geliştirildiği programlama dilleri tarafından otomatik yönetilebilir. (Örneğin 15 dakika kullanılmayan bir tanımlama bilgisinin iptal edilmesi vb.)

Tanımlama ve oturum bilgileri sayesinde web uygulamaları oturum yönetimini gerçekleştirebilirler [18].

a. HTTP isteği:

İstemciler tarafından yapılan istekler HTTP isteği olarak geçmektedir. Bir http isteğinde:

- İstek satırı (Örneğin istenilen klasör veya dosya),
- Başlık bilgileri (Örneğin istek yapılan alan adı, web istemcisi ve sürüm bilgisi, tanımlama bilgisi),

- Boş bir satır,
- Yapılan istek gerektiriyorsa, istekle ilgili veri

bulunmaktadır. Gönderilen HTTP isteği web sunucusu tarafından işlenerek istemciye bir HTTP yanıtı gönderilmektedir.

HTTP isteğinde bulunan istek satırı standartı "[HTTP İstek Metodu] [İstenilen Dosya] [HTTP Protokolü Sürüm Numarası]" gibidir.HTTP standartı, istemciler tarafından yapılan isteklerde geçebilecek 8 adet metot tanımlamaktadır. İstek satırında kullanılan metotlar:

- OPTIONS: Sunucunun desteklediği metotları öğrenmek için kullanılmaktadır.
- GET: Sunucu tarafında bulunan bir kaynağı istemek için kullanılan metottur.
- HEAD: GET metodunda olduğu gibi sunucu tarafındaki bir kaynağı istemektedir. Ancak sunucu sadece HTTP yanıtının başlığını döndürmektedir. Bu metot genelde sunucu üzerinde bir dosyanın varlığını veya en son ne zaman değiştirildiğini kontrol etme amaçlı kullanılmaktadır.
- POST: Kullanıcı tarafından girilen veriyi sunucuya göndermekte kullanılan bir metottur. Örneğin kimlik doğrulama bilgileri veya foruma yazılan bir yazı genelde POST metoduyla gönderilmektedir.
- PUT: Sunucu tarafına bir dosya yazmada kullanılan metottur.
- DELETE: Sunucu tarafındaki bir dosyayı silmekte kullanılan metottur.
- TRACE: Kullanıcı tarafından gönderilen isteği aynı şekilde geri gönderen bir metottur.
- CONNECT: Sunucunun başka bir yere bağlanmasını sağlayan bir metottur.

Proxy sunucularında tünel oluşturmada kullanılmaktadır. Bu metotların dışında bazı web sunucularının desteklediği özel metotlar da bulunmaktadır. Ancak bunlar sadece o web sunucusu için özel olarak geliştirilmiş programlarda kullanılmakta, web istemcileri tarafından yapılan standart isteklerde kullanılmamaktadır.

Standart olan HTTP metotlarının web istemcileri tarafından kullanılanları GET ve POST'tur. Bazı istemciler bir dosyayı istemeden önce HEAD metodunu kullanarak dosyanın varlığını veya en son ne zaman değiştirildiğini kontrol edebilirler. Bu metotların dışındakiler standart web

kullanımının dışında olduğu için, güvenlik açısından kullanımları sunucu tarafında iptal edilmeli veya sınırlandırılmalıdır. HTTP istek satırındaki istenilen dosya veya klasör, "/" karakteri ile başlamaktadır. Bu işaret sunucu kök dizinini belirtmektedir [18].

b. HTTP yanıtı:

İstemci tarafından yapılan HTTP isteklerine karşılık sunucuların verdiği yanıtlar HTTP yanıtı olarak geçmektedir. HTTP yanıtları başlık ve içerik olarak iki kısımdan oluşmaktadır. Bir HTTP yanıtında:

- Durum satırı,
- Başlık bilgileri (Örneğin sunucu zamanı, sunucu yazılım ve sürüm bilgisi, gönderilen verinin uzunluğu, kullanıcı tanımlama bilgisi),
- Boş bir satır,
- Yanıt içeriği

bulunmaktadır. Durum satırı ve başlık bilgileri HTTP yanıt başlığı olarak, yanıt içeriği ise HTTP yanıt içeriği olarak adlandırılmaktadır. HTTP yanıt başlıkları istemci yazılım tarafından işlenir ve kullanıcıya gösterilmez. İstemci tarafından kullanıcıya gösterilen kısım, HTTP yanıtındaki yanıt içeriği kısmıdır.
HTTP yanıtında bulunan durum satırı standartı Çizelge 3.1'de görülebilir.

Çizelge 3.1. HTTP Yanıtı Durum Satırı

[HTTP Protokolü Sürüm Numarası] [Durum Kodu] [Yanıt Açıklaması]

Durum kodu istemci tarafından yapılan istek ile ilgili durumu belirten 3 haneli numerik bir değerdir. Bu 3 hanenin ilk hanesi, durumla ilgili mesajın kategorisini belirtmektedir:

• 1xx – Bilgilendirme Mesajları: İsteğin alındığını ve işleme devam edildiğini belirten mesajlar bu kategoride bulunur.

• 2xx – İstek Başarılı Mesajları: İsteğin başarılı olduğunu belirten mesajlar bu kategoride bulunur.

• 3xx – Yönlendirme Mesajları: Kullanıcının sunucu tarafından belirtilen yere gitmesi veya sunucunun belirttiği dosyaya veya klasöre istek yapması

gerektiği durumlarda kullanılır. (Örneğin dosya farklı bir yere taşındı, dosyayı verdiğim bu adresten iste vb.)

- 4xx – İstemci Hatasını Belirten Mesajlar: İstemcinin yaptığı istekte hata olması durumunda sunucu tarafından kullanıcıya gönderilen mesajlardır. (Örneğin dosya bulunamadı, kullanıcı tarafından yapılan istek hatalı vb.)
- 5xx – Sunucu Hatasını Belirten Mesajlar: Yapılan isteği işlemede bir hata oluşması durumlarında kullanıcıya gönderilen mesajlardır. (Örneğin HTTP sürümünün desteklenmemesi, işlemin zaman aşımına uğraması, sunucu hataları vb.) Yanıt açıklaması iste durum kodunu belirten kısa açıklamadır.

c. Web üzerindeki kaynaklar:

Web üzerinde bulunan kaynaklar, URL (Bkz. Kısaltmalar Tablosu) olarak adlandırılan standart bir formatta gösterilir. Bu formatın standart gösterimi ve web istemcileri tarafından kullanımı Çizelge 3.2'de görülebilir.

Çizelge 3.2. URL Formatı

"protokol"://"sunucu_adresi"[:"port_numarası"][istenilen_dosya [?"sorgu"]]

Bu formata göre:

- Protokol değeri http olmalı, eğer SSL(Bkz. Kısaltmalar Tablosu) ile şifrelenmiş güvenli bağlantı üzerinden çalışan HTTP ise, https olmalıdır.
- Sunucu adresi olarak geçen kısım, bir alan adı, sunucu adı veya IP adresi olabilir.
- Port numarasının girilmesi zorunlu değildir. Varsayılan HTTP port numarası 80'dir. Ancak web sunucuları o şekilde ayarlanırsa farklı porttan da çalışabileceği için, buraya sunucunun çalıştığı port girilebilir.
- İstenilen dosya girilmek zorunda değildir. Girilmezse, web istemcisi tarafından "/" olarak gönderilecektir.
- Sorgu kısmının girilmesi zorunlu değildir. Sorgu kısmındaki bu değerler, sunucu tarafında işlenen dinamik içerikteki değişkenlere verilen değerleri ifade etmektedir.

URL sonlarında "#referans_adı" şeklinde bir referans bulunabilir.

Referans, web istemcilerinin kullanıcıya HTML sayfalarında bazı etiketlerle oluşturulmuş olan referans noktalarını göstermesini sağlamaktadır. Standart bir URL'de web istemcisi sayfanın en üstünü gösterirken, URL'lerin sonuna koyulan bu etiketler, istemcinin sayfa içerisinde belirtilen noktaya gitmesini sağlar. Bu etiketler sunucuya yapılan isteklerde gönderilmez. Sadece istemci tarafında kullanıcıya nerenin başlangıç olarak gösterilmesi gerektiğini belirtir.
Yukarıda belirtilen açıklamalara göre URL örnekleri şu şekilde gösterilebilir.

Örnek 1:

http://www.anadolu.edu.tr/ .

http://www.anadolu.edu.tr/#duyurular, yine aynı sayfayı getirecektir ancak, sayfa içerisinde duyurular diye bir referans noktası varsa, istemcinin dokümanın en üstü yerine bu noktayı göstermesini sağlayacaktır.

Örnek 2:

http://www.anadolu.edu.tr/duyurular/duyuru_icerik.php?duyuru=rektor_bayram_msj&i mg=guncel_duyurular

Yukarıdaki URL örneklerine göre, web istemcisinin yapacağı HTTP isteği ve sunucu yanıtı aşağıdakilere benzer olacaktır:

Örnek 1 için yapılacak iletişim aşağıdaki şekilde gerçekleşecektir. Web sunucusunun 80. port'una bağlanıldıktan sonra istek Çizelge 3.3'teki gibi gönderilerek, istek sonu yeni boş bir satır ile sonlandırılır.
GET / HTTP/1.1

Çizelge 3.3. Örnek HTTP İsteği

Host: www.anadolu.edu.tr
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

İstemci www.anadolu.edu.tr web sunucusundan, HTTP protokolü sürüm 1.1 standartlarına uyarak yaptığı istekte, HTTP GET metoduyla, ana dokümanı (/ dosyasını) istemektedir. Buna karşılık sunucu Çizelge 3.4'te görülebileceği gibi,

yine HTTP protokolü sürüm 1.1 standartlarında, herhangi bir hata olmadığını iletmekte ve dosyayı kullanıcıya göndermektedir.

Çizelge 3.4. Örnek Sunucu Yanıtı

```
HTTP/1.1 200 OK
Date: Sat, 14 Jan 2006 22:40:48 GMT Server: Apache/2.0.54 (Fedora)
X-Powered-By: PHP/5.0.4
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-9
<html>
<head>
<title>Anadolu Universitesi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<meta http-equiv="Content-Language" content="tr">
... (Sunucu yanıtının geri kalan kısmı gerekli görülmediği için eklenmemiştir)
```

Örnek 2'deki URL için yapılacak istekteki fark, Çizelge 3.4'de görülen isteğin ilk satırın bahsedilen dosyayı değişkenlerle istemesi olacaktır. Bu değişkenler web uygulamalarında kullanılmaktadır.Bu durumda istemci isteğindeki ilk satır;
"GET/HTTP/1.1"yerine,
"GET/duyurular/duyuru_icerik.php?duyuru=rektor_bayram_msj&img=guncel_duyurular HTTP/1.1" olacaktır. Web uygulaması tarafında "duyuru" değişkeninin değeri "rektor_bayram_msj", "img" değişkeninin değeri ise "guncel_duyurular" olacaktır.

İstemcinin "User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" olarak gönderdiği değer, kullandığı web istemcisi ve işletim sistemi ile ilgili bilgi içermektedir (Microsoft Windows XP işletim sistemi üzerinde çalışan Internet Explorer 6.0 yazılımı). Ancak bu değer HTTP isteklerinin diğer bölümlerinde olduğu gibi kolaylıkla değiştirilebilir veya normal bir bağlantı sırasında bu değer gönderilerek, sunucunun istemciyi Internet Explorer olarak algılamasını sağlayabilir. Bu nedenle, kullanıcı tarafından gönderilen hiç bir

veriye güvenilmemeli ve tüm veriler yazılım içerisinde kullanılmadan önce kontrol edilmelidir.

3.2 Web Tabanlı Uygulamaların Güvenliği

Klasik istemci-sunucu uygulamalarının aksine web servisi standartlarında çıktı üreten ve yine bu standartlarda veri kabul eden web uygulamaları, bu protokol hakkında bilgisi olan zararlı kullanıcılar tarafından uygulama üzerindeki hataların denetimine açıktır. Web uygulamalarındaki hatalar, uygulamadan yararlanılarak farklı işlemler yapılabilmesine olanak sağlamaktadır.

Web uygulamalarının saldırganların hedefi haline gelmesindeki en önemli faktörler şu şekilde sıralanabilir:

• Erişim kolaylığı: Klasik istemci-sunucu tipi uygulamalar genelde organizasyon içerisinde kullanılmasına rağmen, web tabanlı uygulamalar dış ortaklara ve/veya Internet aracılığıyla herkesin erişimine açılabilmektedir.

• İstemci kolaylığı: Uygulamalar, kendi özel sunucularına erişim için kullanıcı tarafında istemci yazılımın kurulmasını gerekli kılarken, web tabanlı uygulamalarda bu erişim, standart HTML dilini işleyebilen tüm web tarayıcıları ile mümkün olmaktadır. Günümüzde web istemcileri, neredeyse tüm bilgisayarlarda ve diğer cihazlarda bulunmaktadır.

• Deneyimsiz programcılar: Bir çok programcı, güvenlik deneyimi olmadığından dolayı web uygulamalarını yazarken güvenli programlama ilkelerine dikkat etmemektedir. Ayrıca projelerin hayata geçirilmesindeki zaman sıkışıklığı da güvenlik kontrollerinin ikinci plana atılmasını sağlamaktadır.

• Platform bağımsız: Diğer uygulamalar gibi karşı tarafa platforma yönelik bir zararlı kod yüklemek gerekli değildir

Web uygulamaları güvenlik açıkları en sık tespit edilen problemler arasındadır. Web tabanlı uygulamaların güvenliğini arttırmak ve bu konu üzerine çalışmalar yapmak amacıyla bazı organizasyonlar oluşturulmuştur. Bunların başlıcaları “The Open Web Application Security Project (OWASP)” ve “Web Application Security Consortium (WASC)”dur. İki organizasyon da ticari amaç

taşımamakta ve belirli bir ürüne veya markaya hizmet etmemektedir. Organizasyonların kuruluş amaçları web uygulamaları güvenliğine katkıda bulunmak, uygulama güvenliği ile ilgili eğitici ve bilgi verici olmak ifadesiyle özetlenebilir.

Yapılan çalışmalar ve projeler ile web uygulamaları güvenliği gün geçtikçe artmaktadır. Web arabirimi kullanılarak yapılan saldırıların artmasıyla birlikte, web uygulamalarını barındıran, yöneten ve geliştiren kişiler güvenlik risklerine ve problemlerine karşı bilgilendirilmektedir. Bu da web uygulamalarının daha güvenilir ve sağlam hale getirilmesine yardımcı olmaktadır.

OWASP, herkesin katılımına açık bir organizasyondur. OWASP tarafından geliştirilen tüm projeler, yazılan tüm dokümanlar ve diğer yardımlar web uygulamaları güvenliği ile ilgilenen, web uygulamalarını daha güvenilir hale getirmek isteyen herkese açık ve ücretsizdir. OWASP tarafından geliştirilen başlıca projeler:

- OWASP Top Ten Project: Web uygulamalarında en sık karşılaşılan ve hiç bir web uygulamasında bulunmaması gereken güvenlik problemlerini listeleyen bir projedir.
- OWASP Guide Project: The OWASP Guide to Building Secure Web Applications olarak da bilinen güvenli web uygulamaları geliştirme kılavuzu, mevcut 2.0.1 sürümü 300 sayfaya yakın, web uygulamaları güvenliği ve bununla ilgili uygulama geliştiricilerinin ve bakımını yapan kişilerin nelere dikkat etmeleri gerektiğini içeren detaylı bir dokümandır.
- OWASP AppSec FAQ Project: Web uygulamaları güvenliği ile ilgili sıkça sorulan soruları içeren bir projedir.

WASC, web uygulamaları güvenliği konusunda standartları belirleme amacı başta olmak üzere, web uygulamaları güvenliği konularına katkı yapma amacı ile kurulmuş bir organizasyondur. Üyeleri uluslararası, web uygulamaları güvenliği konusunda uzman olan kişilerden oluşmaktadır. Bu kişilerden bazıları web uygulamaları güvenliği konularında ürünler üreten, web uygulamaları güvenlik denetimleri yapan firmalarda çalışmalarına rağmen, WASC organizasyonu üründen ve üreticiden bağımsızdır. Belirlediği standartlar kamuya

açık ve geniş bir kitle tarafından kabul görmüş web güvenlik standartlarıdır. WASC düzenli olarak teknik bilgi, dokümantasyon ve makaleler yayınlamaktadır.

Web Güvenliği Tehdit Sınıflandırması, web uygulamaları güvenliğinde standartlaştırılmaya ihtiyaç duyulan, önemli eksiklikleri ve karmaşıklıkları giderebilecek bir standart olarak oluşturulmuştur. Web uygulamalarını tehdit eden güvenlik problemlerine farklı üreticilerin farklı sınıflandırmalar yapması, ortak bir tehdit sınıflandırması ihtiyacını doğurmaktadır. WASC tarafından orjinali ingilizce olarak yayınlanmış bu dokümanın Türkçe tercümesi de bulunmaktadır.

Web güvenliği sözlüğü, web güvenliği ile ilgili dokümanlar vb. gibi diğer projelere, WASC Projects web sayfalarından ulaşılabilir.

3.3 Web Uygulamalarında Sık Karşılaşılan Güvenlik Problemleri

Bilgi güvenliği bir bütün olarak düşünülmelidir. Web uygulamalarında karşılaşılan tüm problemler aşağıdaki problemlerle sınırlı değildir. Web uygulamalarının barındırıldığı sunucu sistemlerin bulunduğu ağ yapısı, sunucu sistemlerin işletim sistemi, sunucu sistemlere kurulan yazılımlar ve diğer bileşenler, HTTP servisi üzerinde bulunan bazı ayarlar ve bileşenlerde olan güvenlik problemleri, web uygulamalarını da etkileyecektir.

3.3.1 Onaylanmamış Girdilerden Kaynaklanan Problemler

Kullanıcılar web uygulamalarını kullanırken HTTP istekleri yapmaktadırlar. HTTP protokolü standartlarına uygun yapılan tüm istekler, web sunucusu tarafından web uygulamasına iletilmektedir. Web uygulamaları da bu gelen isteklere göre bir sonuç üretmektedir. Bu sonuç üretilirken, istemci tarafından gönderilen bazı veriler yazılım içerisinde değişken olarak kullanılmaktadır.

HTTP protokolünde istemci, isteklerin tamamını değiştirebilir (Örneğin bir önce gelinen sayfanın farklı gösterilmesi, web istemcisinin farklı gösterilmesi, tanımlama bilgisinin farklı gösterilmesi). Bu nedenle kullanıcı tarafından gelen

tüm veriler zararlı veriler olarak düşünülmeli ve yazılımlarda kullanılmadan önce kontrol edilip onaylanmalıdır [18].

Kullanıcı tarafından gönderilen verilerin, onaylanmadan, direk yazılım içerisinde kullanılması web uygulamalarında karşılaşılan neredeyse tüm sorunların ana nedenidir. Kullanıcı tarafından gelen tüm veriler kullanılacağı yere göre belirli kontrollerden geçirilmelidir. Ancak bu kontrolleri kullanıcı tarafındaki mekanizmalarla (Ör: JavaScript kontrolleri) gerçekleştirmek geçerli bir çözüm değildir.

Onaylanmamış girdilerden kaynaklanacak güvenlik problemlerinden korunabilmek için, kullanıcı tarafından gelen tüm veriler yazılım içerisinde kullanılmadan önce, yazılım tarafında kontrol edilmelidir. Bu problemlerin çözümündeki temel yaklaşım, bilinen kötü verileri/karakterleri filtrelemek yerine, sadece bilinen ve beklenen iyi verileri/karakterleri kabul etmek olmalıdır.

Örneğin doğum tarihi gibi sayısal bir değer olabilecek alana sadece sayısal değer girildiğinden, soyad gibi alfabetik bir değer alabilecek alana sadece alfabetik değer girildiğinden, veri içerisinde SQL veritabanında kullanılan özel karakterlerin girilmediğinden emin olunmalıdır [18].

3.3.2 Komut Sızdırma veya SQL'e Sızma Açıkları

Web tabanlı uygulamaların dinamik içerikleri ve verileri veritabanı sistemlerinde saklanmaktadır. Ayrıca web tabanlı uygulamalarda, bazı işlevleri gerçekleştirmek için harici yazılımlar veya sistemler kullanılmaktadır.

Kullanıcı tarafından girilen veriler, kontrol edilmeden direk bu harici yazılımlara (Web uygulamasının kullandığı harici veritabanı, yazılım ve sistemler) gönderiliyorsa, kötü niyetli bir kullanıcı zararlı veri girerek harici yazılım üzerinde yazılımın elverdiği aktiviteyi gerçekleştirebilir, veya bu yazılımları kullanarak sunucu sistem üzerinde kod çalıştırabilir.

a. Komut sızdırma açıkları:

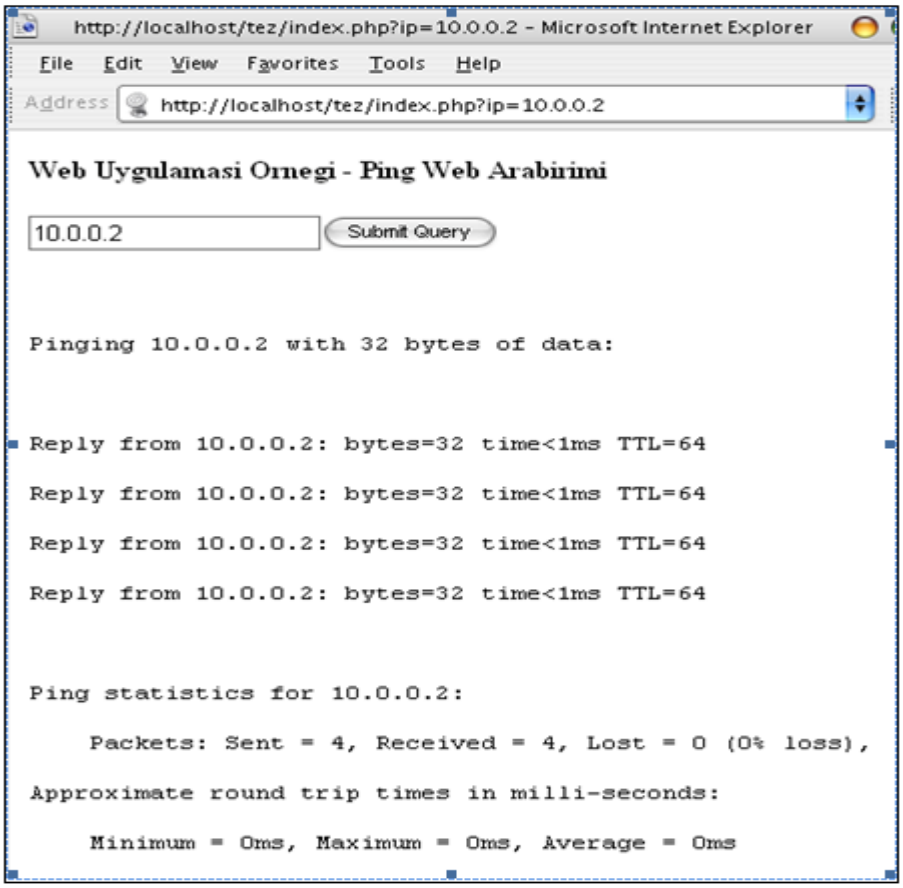
Parametre sonlarına veya aralarına belirli karakterler ekleyerek harici yazılımın sistem komutu çalıştırması mümkün olabilir.

Çizelge 3.5'de görülebilecek PHP kodu, kullanıcı tarafından girilen IP adresine ulaşılıp ulaşılamadığını gösteren Ping yazılımı için yazılmış basit bir web arabirimidir.

Çizelge 3.5. Komut Sızdırma Açığından Etkilenen Kod

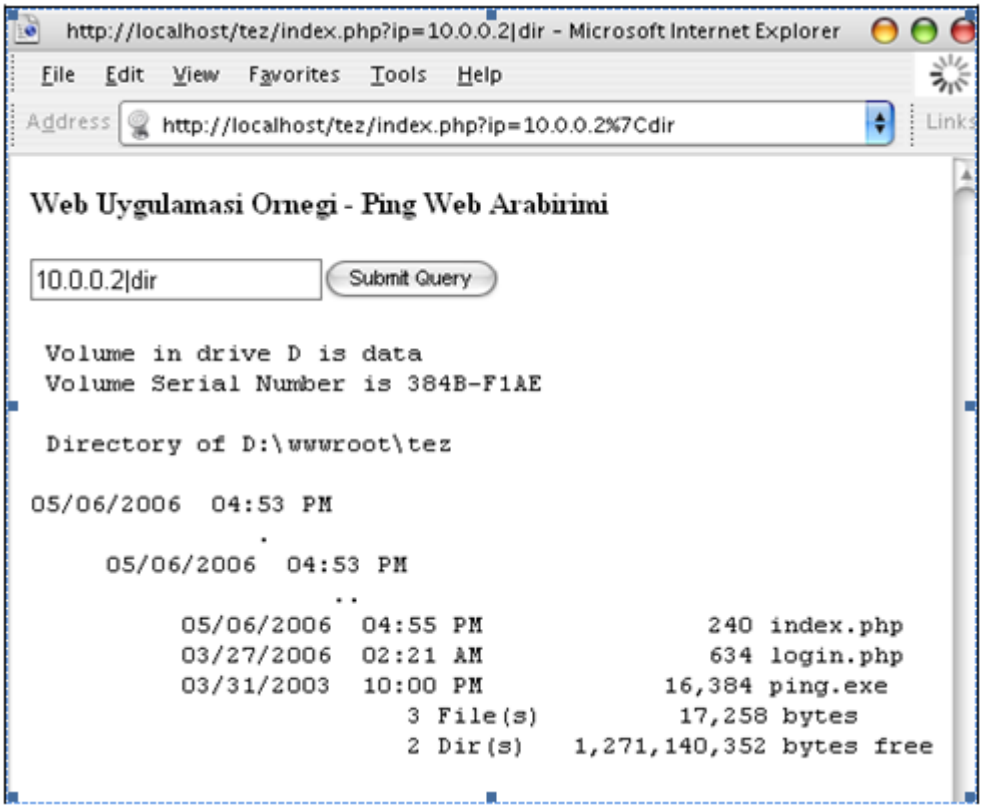
```
<b>Web Uygulamasi Ornegi - Ping Web Arabirimi</b>
<form method=get action="index.php">
<input type="text" name="ip" value="<?=$_GET[ip]?>"><input type="submit">
</form>
<pre>
<?
if ($_GET[ip])
    $l = system("ping $_GET[ip]", $rv);
?>
```

Bu sayfa bir web istemcisinde görüntülendiğinde kullanıcıdan IP adresi bilgisini isteyecek ve girilen IP adresi bilgisini Ping yazılımına göndererek, oluşan çıktıyı Şekil 3.1'de görülebileceği gibi kullanıcıya gösterecektir [18].



Şekil 3.1. Web Uygulamasının Normal Kullanımı

Komut sızdırma açığından etkilenen bu kodda kullanıcı, IP adresi girilmesi gereken yere ek parametreler girerek, web sunucusu üzerinde web servisinin hakları ile komut çalıştırabilir. Örneğin Şekil 3.2'de görülebileceği gibi, IP adresi bölümüne zararlı veri girilerek "dir" komutu çalıştırılabilmiştir. Microsoft Windows işletim sistemlerinde klasör içeriğini listeleyen bu komutun çıktısı, web istemcisi içerisinde görülebilmektedir.



Şekil 3.2. Komut Sızdırma Açığından Etkilenen Uygulama

Sistem üzerinde tanınan bir kullanıcıymış gibi sistem üzerinde komut çalıştırabilmek, sisteme sızılması işlemini çok basit bir hale getirecektir.

Komut sızdırma güvenlik açığının WASC Web Güvenliği Tehdit Sınıflandırması'ndaki karşılığı "Komut Çalıştırma: İşletim Sistemi Yönetme" olarak geçmektedir.

b. SQL'e sızma açıkları:

Web uygulamaları genelde SQL (Bkz. Kısaltmalar Tablosu) veritabanlarını kullandıklarından, en sık karşılaşılan zararlı veri girme zayıflığı SQL'e sızma (SQL Injection) açığıdır. SQL'e sızma; SQL veritabanı sistemine zararlı veri girilmesini mümkün kılan bir zayıflıktır. Bu güvenlik açığından etkilenen web uygulamaları kullanılarak, veritabanında bulunan veriler görülebilir veya SQL tabloları ve verileri üzerinde değişiklik yapmak mümkün olabilir. Bazı SQL veritabanlarının gelişmiş özelliklerini kullanarak sunucu sistem üzerinde kod çalıştırmak da mümkündür [18].

Çizelge 3.6'da görülebilecek PHP kodu, bir web uygulamasının kullanıcı adı ve şifre doğrulama işleminde kullanılan bölümüdür. Bu sayfa bir web istemcisinde görüntülendiğinde kullanıcıdan kullanıcı adı ve şifre bilgilerini isteyecek ve kullanıcı tarafından girilen bilgileri veritabanında sorgulayarak, bu bilgilerin doğruluğunu onaylayacaktır.

Çizelge 3.6. SQL'e Sızma Açığından Etkilenen Kod

```
<b>Web Uygulamasi Ornegi - Kullanici Adi ve Sifre Dogrulama</b>
<form method=post action="">
Kullanici adi: <input type="text" name="user"><br>
Sifre: <input type="text" name="pass"><br><input type="submit" value="Giris">
</form>
<pre>
<?
if ($_POST[user] and $_POST[pass]) {
    mysql_connect("localhost", "tez", "TEZpassword");
    mysql_select_db("tez");
    $sql="
    SELECT * FROM kullanicilar
    WHERE user = '".$_POST[user]."' AND pass = '".$_POST[pass]."'
    ";
    $result=mysql_query($sql);
    if ($result and mysql_num_rows($result)>0) {
        echo "<b>Giris basarili</b>\n";
    } else {
        echo "<b>Giris hatasi</b>";
    }
}
?>
```

Bunun gibi kullanıcı tarafından girilen verinin, SQL veritabanına özel karakterlere karşı filtrelenmediği durumlarda, SQL'e sızma metodunu kullanarak kimlik doğrulama işlemini aşmak mümkün olacaktır.

Kullanıcı tarafından girilen veri; kullanıcı adı "skolat" ve şifre "sifre" olduğu durumda, kodda görülen SQL sorgusu;

" SELECT * FROM kullanicilar WHERE user='skolat' AND pass='sifre' halini alacak ve veritabanı 1 kayıt döndüreceği için kimlik doğrulaması Şekil 3.3'te görülebileceği gibi gerçekleşecektir.

Geçerli kullanıcı adı ve şifre ikilisinin sağlanmadığı durumlarda kullanıcı girişi kabul edilmez [18].

Şekil 3.3. Geçerli Kullanıcı Adı ve Şifre İle Giriş

Ancak bu uygulamadaki SQL'e sızma açığından yararlanılarak, kimlik denetimini aşmak mümkündür. Bunun için kullanıcı tarafından; kullanıcı adı "mustafa" ve şifre "' OR 1='1" girilmelidir. Bu durumda kodda görülen SQL sorgusu " SELECT * FROM kullanicilar WHERE user='skolat' AND pass='' OR 1='1' " halini alacaktır. Bu sorgu kullanıcı adının "skolat" ve kullanıcı şifresinin boş veya 1'in 1'e eşit olduğu durumlarda geçerlidir. 1 her zaman 1'e eşit olacağından, sorgu hep geçerli olacaktır ve veritabanı sonuç olarak "mustafa" kullanıcısının kaydını döndürerek kimlik doğrulamasının Şekil 3.4'de görülebileceği gibi gerçekleşmesini sağlayacaktır.

Şekil 3.4. SQL'e Sızma Metoduyla Kimlik Doğrulamanın Aşılması

SQL'e sızma güvenlik açığının WASC Web Güvenliği Tehdit Sınıflandırması'ndaki karşılığı "Komut Çalıştırma: SQL Enjeksiyonu" olarak geçmektedir.

3.3.3 Çapraz Site Kod Çalıştırma Zayıflıkları

Uygulamaya gönderilen veri filtrelenmeden üretilen sayfada gösteriliyorsa, bu uygulama çapraz site kod çalıştırma (XSS (Bkz. Kısaltmalar Tablosu)) açığından etkileniyor denilebilir.

Saldırganlar bu açığı kullanarak, kullanıcılara hazırladıkları bağlantı adreslerini göndermek suretiyle, bu bağlantı adreslerine giden kullanıcıların bilgisayarında, oluşturdukları özel JavaScript kodunu çalıştırabilirler. Bu sayede kullanıcı tarafındaki tanımlama bilgisi, oturum bilgisi ve diğer hassas bilgiler saldırganın eline geçebilir veya saldırgan sayfayı değiştirilmiş gibi gösterebilir. Çapraz site kod çalıştırma saldırıları iki şekilde gerçekleşmektedir;

1. Saldırgan tarafından gönderilen zararlı veri, veritabanında saklanıp uygulamanın o bölümünü görüntüleyen kişilere filtrelenmeden gösterilir. (Ör: forum yazısı, ziyaretçi kayıtları, kullanıcı kimlik bilgilerinin tutulduğu sayfa)
2. Kullanıcının bu açıktan etkilenmesi için, saldırganın oluşturduğu özel URL'ye gitmesi gereklidir. (Ör: Farklı bir web sitesi aracılığıyla veya e-posta kullanılarak kullanıcıya sahte URL gönderme, veya sahte form doldurtma ile)

Yukarıda "**Sql Sızma Açıklarında"** görülebilecek PHP kodu kullanıcıdan aldığı veriyi filtrelemeden sayfa içerisine yazdığı için çapraz site kod çalıştırma saldırısından etkilenmektedir. Web uygulaması kullanıcı tarafından giriş yapılacak bölümde bir IP adresi yazılmasını beklemektedir. Ancak zararlı bir kullanıcı bu bölüme;

"><script>document.write('<h1>Sayfa%20X%20tarafindan%20degistirilmistir')</script> yazarak bu güvenlik zayıflığını tetikleyebilir. Çapraz site kod çalıştırma açıkları bu zararlı kodları içeren sayfaları görüntüleyen kullanıcıları etkileyeceğinden,saldırganın"http://localhost/tez/index.php?ip="><script>document.write('<h1>Sayfa%20X%20tarafindan%20degistirilmistir</h1>')</script>"adre

sini problemden etkilenmesini istediği kullanıcıya göndermesi ve kullanıcının gönderilen adresi açması gerekmektedir. Kullanıcı yukarıdaki adresi görüntülediğinde, göreceği sayfa Şekil 3.5'de ki gibi olacaktır.

Şekil 3.5. Çapraz Site Kod Çalıştırma Zayıflığından Etkilenen Web Sayfası

Çapraz site kod çalıştırma güvenlik açığının WASC Web Güvenliği Tehdit Sınıflandırması'ndaki karşılığı "İstemci Taraflı Saldırılar: Siteler Arası (ötesi) Betik Yazma" olarak geçmektedir.

3.3.4 Hafıza Taşmaları

Gönderilen verinin, bir girdiye ayrılan hafızadan daha uzun olduğu durumlarda hafıza taşması problemleri meydana gelebilir. Web uygulaması geliştirme dilleri genelde bu güvenlik açığından etkilenmese de, kullanılan harici uygulamalar bu açıklardan etkileniyor olabilir.

Genelde C/C++ programlama dilleri ile programlanmış yazılımlar hafıza

taşması güvenlik açıklarından etkilenmektedir. Bu hata durumu, uygulamanın belirli bir girdi için bellekte ayırdığı yere, daha uzun bir girdinin yazılması durumlarında oluşmaktadır. Hafıza taşması güvenlik probleminin tespit edildiği durumlarda, özel uzunlukta hazırlanmış özel girdiler, saldırganların bellek yönetimini ele geçirerek sunucu sistem üzerinde kod çalıştırmasına olanak sağlamaktadır

Hafıza taşmasından etkilenen bir web uygulaması bileşenini test etmek için uzaktan yapılan testler sırasında, uygulamaya web standartlarının kabul edeceği ölçüde uzun karakter gönderilerek, sunucunun hata mesajı döndürüp döndürmediğine bakılır. Ancak bu hata mesajından, test edilen sistemde bu güvenlik açığının olup olmadığını saptamak mümkün olmayabilir. Hafıza taşmalarına bakıldığında, web uygulamalarındaki diğer güvenlik açıklarına nazaran tespit etmesi ve bu açıklardan faydalanılarak zararlı aktivite gerçekleştirilmesi çok daha zordur.

Bu güvenlik açığından etkilenen web uygulamaları olsa da, web uygulamalarında tespit edilen güvenlik açıklarına oranla hafıza taşması açığından etkilenen web uygulamalarının oranı oldukça düşüktür.

Hafıza taşması güvenlik açığının WASC Web Güvenliği Tehdit Sınıflandırması'ndaki karşılığı "Komut Çalıştırma: Ara Bellek Taşması" olarak geçmektedir [19].

3.3.5 Kimlik Doğrulama ve Oturum Yönetimi Problemleri

Kimlik doğrulama ve oturum yönetimi bir web uygulamasının en kritik görevlerinden birisidir. Web uygulamalarında karşılaşılan kimlik yönetimi problemleri, izinsiz kullanıcıların farklı bir kullanıcının hesabı ile giriş yapabilmesine, yetkili bir kullanıcının oturumunun ele geçirilebilmesine ve sistem kullanıcılarının yetkilerinin üzerinde işlemler yapabilmesine olanak sağlayabilir. Web uygulamalarında kimlik yönetimi üç bileşenle sağlanmaktadır;

a. Kimlik Doğrulama,

b. Oturum Yönetimi,

c. Erişim Kontrolü / Yetkilendirme.

Kimlik Doğrulama ve Oturum Yönetimi işlemleri, uygulamayı kullanan kayıtlı kullanıcıların sistem tarafından tanınmasını ve kullanıcıların aktif bağlantılarının yönetimini; Erişim Kontrolü, yani Yetkilendirme ise hangi kullanıcının hangi bölümlere erişebileceğinin kontrolünü sağlamaktadır[19].

Web uygulamalarında kullanılan kimlik doğrulama işleminde genelde, kullanıcı adı ve şifre kullanılmaktadır. Tek kullanımlık şifre üreten cihazlar gibi güçlü kimlik doğrulama araçları olmasına rağmen, bunlar bütçesel bir yük oluşturmaktadır. Ancak kullanıcılar tarafından belirlenen şifrelere nazaran daha güçlü olması nedeniyle, günümüzde Internet bankacılığı gibi kritik web uygulamalarında bu cihazlar kullanılmaya başlanmıştır.

HTTP protokolünde oturum yönetimi özelliği bulunmadığı için web uygulamaları, kullanıcıların yaptığı istekleri bir oturum bilgisinde tutmak zorundadır. Oturum yönetiminin düzgün yapılmadığı durumlarda veya oturum bilgisinde kullanıcıya özel verilerin tutulmaması durumlarında kullanıcının oturum bilgisini ele geçiren bir saldırgan, bunu kullanarak kullanıcının yerine geçebilir [19].

Kimlik doğrulama ve oturum yönetimi problemleri WASC Web Güvenliği Tehdit Sınıflandırması'nda "Yetkilendirme: Yetki/Oturum Bilgisi Tahmin Etme" ve "Yetkilendirme: Yetersiz Oturum Sonlandırma" başlıkları altında standartlaştırılmıştır. Kullanıcı yetkilendirilmeleri ile ilgili diğer problemler, "Yetkilendirme" ana başlığı altında toplanmıştır [19].

3.3.6 Erişim Kontrolü Problemleri

Erişim kontrolü, yani yetkilendirme, kullanıcıların nerelere erişip nereye erişmemesi gerektiğinin kontrol edilmesidir. Oluşacak bir yetkilendirme problemi yetkisiz veya düşük yetkili bir kullanıcının erişmemesi gereken yere erişerek, eriştiği yerdeki fonksiyonları kullanabilmesini sağlayacaktır.

Erişim kontrolü problemlerinin WASC Web Güvenliği Tehdit

Sınıflandırması'ndaki karşılığı "Yetkilendirme: Yetersiz Yetkilendirme" olarak geçmektedir [19].

3.3.7 Hata İşleme Problemleri

Hata denetimi ve hata işleme, web uygulamalarının önemli bir parçasıdır. Kullanılan sistemin teknik detayları hakkında bilgi içerebilecek hata mesajlarının kullanıcı karşısına çıkması, hata denetim fonksiyonları ile kontrol edilmelidir. Uygulama tarafından üretilecek özel hata mesajları ise, kötü niyetli bir kullanıcıya bilgi vermeyecek şekilde basit, sade ve anlaşılır olacak şekilde oluşturulmalıdır [20]. Oluşan hataların sonradan incelenebilmesi için, oluşan hataların bir dosyaya kaydedilmesi önerilmektedir.

Hata denetimi düzgün yapılmayan uygulamalar, problem durumunda kritik bilgileri hata mesajları ile dışarı sızdırabilir. Bu hatalar, sistem hakkında bilgi içeren, sistem yapısını açığa çıkartabilen ve saldırganların bilmesi durumunda onlara avantaj sağlayacak mesajlar olabilir [21].

Uygulama içerisinde hata denetimi iyi yapılmalı, hata mesajları kullanılabilir bilgi vermemelidir.

Hata işleme problemleri WASC Web Güvenliği Tehdit sınıflandırmasın da "Bilgi Açığa Çıkarma: Bilgi Sızıntısı" konusu kapsamında standartlaştırılmıştır.

3.4 Web Uygulamalarının Güvenlik Problemlerine Karşı Test Edilmesi

Güvenlik denetimleri, bir ağ, bilgi sistemi, servis veya uygulamanın güvenlik problemlerine karşı incelenmesi işlemidir. Bu işlemler sırasında, kötü niyetli kullanıcıların ve saldırganların sisteme zarar verebileceği yollar denenerek, denetlenen sistemin güvenlik zaafiyetlerinin tespitine çalışılır. Güvenlik testleri kapalı kutu (Black box testing) ve açık kutu (White box testing) testleri olarak ikiye ayrılmaktadır. Açık kutu testlerinde, test edilen uygulamanın kaynak kodu, geliştirilme platformu, web sunucusu vb. bilgiler bilinmektedir. Kapalı kutu testlerinde ise uygulamanın kaynak koduna sahip olunmamakta, işleyişi ve yapısı bilinmemektedir. Bu nedenle kapalı kutu

testleri, kötü niyetli kullanıcıların veya saldırganların sisteme zarar verebileceği şekilde yapılmaktadır.

Web uygulamalarının güvenlik testlerine tabi tutulması, uygulama geliştirme aşamasının bir parçasıdır. Güvenlik testlerinin uygulama geliştirme aşamasında ve sonrasında yapılması organizasyonlar için büyük önem taşımaktadır. Bu güvenlik denetimleri için önceden belirlenmiş detaylı bir yol izlenmelidir. Web uygulamaları güvenlik denetimlerinin genel aşamaları aşağıdaki gibidir:

1. Keşif Aşaması
2. Güvenlik Denetimi Aşaması
3. Raporlama Aşaması [21].

3.4.1 Keşif Aşaması

Web uygulamalarının güvenlik açıklarına karşı test edilmesi için, diğer güvenlik testlerinde olduğu gibi, keşif aşamasının iyi yapılması gerekmektedir. Uygulamada hangi teknolojilerin kullanıldığı, dinamik içeriğin hangi dil ile geliştirildiği, klasör ve dosya yapısı, kimlik doğrulama ile erişilen içerikler, ne tip bir kimlik doğrulama yönteminin kullanıldığı, formlar, sorgu alanları, dışarıya veya kardeş sunuculara olan bağlantılar (link), tespit edilebiliyorsa veritabanı tipi ve bunun gibi diğer elde edilebilen bilgilerin, keşif aşamasında tespit edilmesi gereklidir [22]. Bu işlem uygulamanın güvenlik problemlerine karşı test edilmesi işlemini kolaylaştıracaktır.

Keşif aşamasının ardından edinilen bilgilere göre, uygulama güvenlik problemlerine karşı denetlenebilir. Aynı ağ üzerinde bulunan ve aynı ekip tarafından yönetilen sunucular, web sayfalarında veya web uygulamasında yer alan belirli dosyaların tutulduğu sunucular (Örneğin resim dosyalarının saklandığı sunucular)

a. Teknoloji tespiti:

Uygulamada hangi teknolojilerin kullanıldığı, dinamik içeriğin hangi dil ile geliştirildiği bilgileri şu yöntemlerle öğrenilebilir:

- Sunucu üzerinde bulunan dosyaların dosya uzantılarından,
- Web uygulaması tarafından gönderilen tanımlama veya oturum bilgisinden,
- Sunucu yanıtındaki öğelerden (Örneğin sunucu yanıtında "X-Powered-By: PHP/5.0.4", "X-Powered-By: ASP.NET" veya "Server: Apache/2.0.54 (Debian GNU/Linux) DAV/2 SVN/1.1.4 mod_python/3.1.3 Python/2.3.5 mod_ssl/2.0.54 OpenSSL/0.9.7e" benzeri satırların bulunması),
- URL formatından,
- Hazır kod kullanılmış olmasından,
- HTML çıktısındaki açıklama satırlarından (<!-- --> etiketleri arasında bulunan yazılar),
- Uygulamadaki URL'lerin bir kısmının eksik gönderilerek sonucun incelenmesinden,
- Standart giriş dosyalarına yapılacak istekler sonucu alınan yanıtların incelenmesinden (Ör: index.php, default.asp, index.html, index.jsp),
- Olmayan dosyalara yapılan istekler sonucu alınan yanıtların ve sayfaların incelenmesinden,
- Olmayan dosyalara, standart web uygulaması geliştirme dillerine ait uzantılarla yapılacak istekler sonucu alınan yanıtların incelenmesinden (Ör: x.jsp, x.asp, x.php, x.aspx),
- Web uygulamasına gönderilen değişkenlere ait değerlerin değiştirilerek yanıtların incelenmesinden [22].

b. Klasör tespiti:

Web uygulamasının işleyişini ve yapısını anlayabilmek ve bu alandaki zayıflıkları tespit edebilmek için klasör veya dosya yapısı incelenmeli, var olan klasörler tespit edilmelidir.

Klasör tespiti:

- Tahmin yoluyla,
- Uygulamada kullanılan resim, javascript veya diğer dosyaların bulunduğu klasörlerin tespiti ile,
- /robots.txt dosyasından yola çıkarak

yapılabilir. Tespit edilen klasörlerin klasör listeleme problemlerinden etkilenip

etkilenmediği denenmelidir. İçeriği listelenen klasörlerde unutulmuş olabilecek yedek dosyalar veya web uygulamasına ait eski sürüm dosyalar kontrol edilmelidir [22].

c. Hata sayfası kontrolleri:

Uygulama geliştiriciler tarafından kullanıcıya daha fazla bilgi vermesi amacıyla veya farklı bir amaçla oluşturulan özel hata sayfalarında da güvenlik problemleri olabilmektedir.

Sunucu üzerinde olmayan dosya ve klasörlerden yola çıkarak uygulamanın özel oluşturulmuş bir hata sayfası gösterip göstermediği kontrol edilmelidir. Buna göre, gönderilen URL içerisindeki bir parametre çıkan hata sayfasına direk yazılıyorsa web uygulamasında bir çapraz site kod çalıştırma açığı olabilir [22].

d. Yedek dosyaların kontrolü

Web uygulamasının belirli bir bölümünde değişiklik yapılmadan önce varolan dosyanın yedeği alınmış ve sunucu üzerinde bırakılmış olabilir. Yedek dosyalar bazı editörler tarafından otomatik olarak da oluşturulmuş olabilir.

Web uygulamasındaki güvenlik problemleri giderilmiş olsa da, sunucu üzerinde silinmesi unutulmuş bir yedek dosyada bir güvenlik problemi olabilir. Ayrıca dosya uzantısı ".bak", ".old" vb. olan sunucu tarafında işlenen dinamik içerik dosyalarına yapılan istekler sunucu tarafından işlenmeyeceğinden, o sayfanın kaynak kodlarına ulaşılabilecektir.

Bu nedenlerden dolayı, bulunan her dosya için yedek dosya olup olmadığı kontrol edilmeli ve bunun dışında olabilecek standart dosyaların varlığı kontrol edilmelidir [22].

d. Uygulama giriş noktalarının tespiti:

İncelenen sayfalar üzerindeki formlar, URL'lerde bulunan sorgu değerleri, HTML çıktısındaki açıklamalar ve diğer incelenen bileşenler neticesinde uygulama giriş noktaları tespit edilmelidir. Tespit edilen bu noktalar, onaylanmamış girdilerden kaynaklanan problemlere karşı test edilecektir. Örneğin

web uygulamasında “http://localhost/haberler/haber_goster.asp?id=3” şeklinde bir URL varsa; “haberler” bir klasör, “haber_goster.asp” bir dosya, “id” ise bu dosyaya gönderilen bir değişkendir. Bu değişkene verilen değer, uygulama içerisinde kullanılmaktadır. Dolayısıyla bu değişken uygulamaya olan bir giriş noktasıdır denilebilir.

Aynı şekilde kullanıcı tarafından gönderilen HTTP başlık bilgisi gibi diğer veriler de (Örneğin kullanıcının kullandığı web istemcisi) web uygulamasında kullanılıyor olabilir.

3.4.2 Güvenlik Denetimi Aşaması

Her bir güvenlik açığı için belirli zararlı karakterler veya özel parametreler gönderilerek sunucu tarafından gönderilen yanıtlar incelenmektedir. Yapılan isteklere verilen yanıtlardan, web uygulamasının denetlenen bölümünün, test edilen güvenlik açığından etkilendiğini söylemek mümkün olabilmektedir.

a. Komut sızdırma açığı denetimi:

Web uygulamasında bulunan sayfalarda, komut sızdırma açığından etkilenen değişkenleri bulmak için, değişkenlere “;”(noktalı virgül), “|”(boru –pipe– karakteri) veya “&”(ve karakteri) karakterlerinden birisi ve çalıştırılmak istenilen işletim sistemi komutu gönderilmelidir.

Web uygulaması bu problemden etkileniyorsa, gönderilen işletim sistemi komutun çıktısını kullanıcıya gösterecektir [23].

Bazı durumlarda uygulamada bu problem bulunmasına rağmen, komut çıktısını zararlı isteği yapan kullanıcıya göstermeyebilir. Bu gibi durumlarda ya farklı bir hata mesajı görülebilir, ya da sayfa eksik olarak görüntüleniyor olabilir. Bu nedenle işleme kesinlik kazandırmak için, işletim sisteminin işleyişini belirli bir zaman durdurabilen bir komut(Örneğin pause komutu) gönderilerek kontrol yapılabilir.

b. SQL’e sızma açığı denetimi:

Web uygulamasında bulunan sayfalarda, SQL’e sızma açığından etkilenen değişkenleri bulmak için, değişkenlere “;”(noktalı virgül), “ ’ ”(tekli

tırnak) veya “ “ ”(çift tırnak) karakterlerinden birisi gönderilmelidir. Web uygulaması bu problemden etkileniyorsa, uygulama oluşan SQL hata mesajının tamamını veya bir kısmını gösterecektir.

Bazı uygulamalar bu karakteri filtreliyor olabilir. Bu durumlarda varolan değişken değerinin yanına bir “ ”(boşluk) karakteri ve bununla birlikte bir takım rastgele oluşturulmuş harfler veya rakamlar girilerek, sunucunun yanıtı incelenmelidir.

Komut sızdırma açıklarında da olduğu gibi, bazı durumlarda uygulamada bu problem bulunmasına rağmen, zararlı isteği yapan kullanıcıya herhangi bir SQL hata mesajı gösterilmeyebilir. Bu durumlarda bu güvenlik problemine “Görmeden SQL’e Sızma (Blind SQL Injection)” da denilmektedir. Görmeden SQL’e sızma güvenlik problemini test edebilmek için aşağıdaki yöntemler uygulanabilir:

- SQL veritabanının belirli bir süre beklemesi için özel SQL komutu gönderilir.

Örneğin Microsoft SQL sunucusuna gönderilecek “WAITFOR DELAY '00:00:20'” SQL komutu, web uygulamasının kullanıcıya vereceği yanıtı 20 saniye geciktirecektir. Aşağıdaki gibi 3 farklı istek gönderilerek sonuçlar karşılaştırılır:

a. orjinal değer, boşluk karakteri ve “and 1=1”
b. orjinal değer
c. orjinal değer, boşluk karakteri ve “and 1=0”

Karşılaştırma sonucunda yukarıdaki a isteği b isteğine verilen sunucu yanıtı aynı olmalı, c isteğine verilen sunucu yanıtı ise a ve b isteğine verilen yanıtlarla aynı olmamalıdır [23].

c. Çapraz site kod çalıştırma açığı denetimi:

Web uygulamasına gönderilen veri filtrelenmeden üretilen sayfada gösteriliyorsa, bu uygulama çapraz site kod çalıştırma açığından etkileniyor denilebilir.

Sadece dosyalarda veya form alanlarında bulunan değişkenler değil, kullanıcı tarafından gönderilen herhangi bir verinin, kullanıcıya tekrar gösterildiği

her bölüm için bu testler yapılmalıdır. Web uygulamasında bulunan sayfalarda, çapraz site kod çalıştırma açığından etkilenen değişkenleri bulmak için, değişkenlere “ ” >”(çift tırnak ve büyüktür işareti) sonrasında JavaScript kodu veya direk JavaScript kodu gönderilmelidir [24]. (Örneğin: “<script>alert(‘çapraz site kod çalıştırma testi’)</script>”)

Bu güvenlik problemi 90’ın üzerinde yöntemle tetiklenebilir. Bu problemin varlığını denetlemek için sadece “<script>” etiketi ile JavaScript kodu çalıştırmayı denemek yerine, farklı HTML etiketleri ile de testler yapmak, güvenlik denetiminin doğruluğunu arttıracaktır. (Örneğin: “<body onload=“alert(‘çapraz site kod çalıştırma testi’)”>”)

d. Kimlik doğrulama problemlerinin denetimi

Web uygulamasının kullanıcı adı ve şifre ile giriş yapılan bölümlerindeki güvenlik açıklarını kullanarak geçerli bir kullanıcı adı ve şifre olmadan sisteme giriş yapılabilmektedir. Kullanıcı adı ve şifre ile giriş yapılan yerlerde bulunabilecek olan problemler:

- Hata Mesajlarından Kullanıcı Adı Tahmini: Web uygulamaları kullanıcı adı ve şifre ile giriş yapılan yerlerde “Geçersiz Kullanıcı Adı veya Şifre” benzeri hata mesajları üretmelidir. Ancak bazı web uygulamaları kullanıcı adının hatalı girildiği durumlarda “Geçersiz Kullanıcı Adı”, şifrenin hatalı girildiği durumlarda ise “Geçersiz Şifre” benzeri hata mesajları üretmektedir. Bu hata mesajları son kullanıcıya yardımcı olabilecek bir özellik gibi gözükse de, aslında sistemde bulunan kullanıcı adlarının tahmin edilmesine yol açmaktadır. Bu sayede deneme-yanılma yöntemiyle geçerli bir kullanıcı adı ve şifresi tespit etmek kolaylaşacaktır.

Web uygulamalarının kullanıcı adı ve şifre ile girilen bölümlerinin de güvenlik testlerine tabi tutulması için, güvenlik denetimini yapan kişinin elinde geçerli bir kullanıcı adı ve şifre olmalıdır. Bu güvenlik probleminin tespiti için sisteme geçerli bir kullanıcı adı ve hatalı bir şifre giriş yapılmaya çalışılır. Aynı şekilde geçersiz bir kullanıcı adı ve rastgele oluşturulmuş bir şifre gönderilip gelen yanıt incelenir. Web uygulamasının yapılan iki isteğe verdiği hata mesajları birbirinden farklıysa, bu web uygulamasının bilgi verici hata mesajı verdiği

söylenebilir [25].

- SQL'e Sızma Yöntemi ile Sisteme Giriş Denemesi: Web uygulamasının kullanıcı adı ve şifre ile giriş yapılan bölümü SQL'e Sızma saldırılarından etkileniyorsa, şifre değerine " ' OR 1='1" benzeri bir değer gönderilerek sisteme giriş yapılabilir. Gönderilen bu değer, uygulama içerisindeki SQL sorgusunun doğru olmasını ve en az 1 sonuç döndürmesini sağlayacağından bir çok web uygulamasında kişi sisteme giriş yapabilecektir.
- Zayıf Kullanıcı Adı ve Şifresinin Bulunması: Bazı durumlarda kullanıcı adı ve şifreler tahmin yoluyla bulunabilmektedir. Web uygulamasının kullanıcı adı ve şifre ile giriş yapılan bölümüne, sıkça kullanılan zayıf kullanıcı adı ve şifre bilgisi ile girilmesi işlemi denenmelidir [25].

Basit kullanıcı adı/şifre kombinasyonları: admin/admin, test/test, admin/12345, root/root, administrator/123456 vb. olarak örneklenebilir.

e. Oturum yönetimi ve erişim kontrolü problemlerinin tespiti:

Web uygulamasının kullanıcı adı ve şifre ile giriş yapıldıktan sonra erişilebilen bölümündeki sayfalara, bu sayfaların adreslerini bilen bir kullanıcı rastgele erişememelidir.

Uygulamaya kullanıcı adı ve şifre bilgisi ile giriş yapılmaksızın, bilinen adrese erişilerek, sadece kullanıcıların ulaşabileceği fonksiyonlar kullanılabiliyorsa, web uygulamasında bir oturum yönetimi problemi vardır denilebilir. Aynı şekilde bir kullanıcı kendi yetkilerinin dışındaki bildiği fonksiyonlara erişebiliyorsa, web uygulamasında erişim kontrolü problemi vardır denilebilir [25].

f. Tahmin edilebilir tanımlama ve oturum bilgisi problemlerinin tespiti

Kullanıcıya uygulama tarafından gönderilen tanımlama bilgisinde "yonetici_seviyesi=1" benzeri bir değer olabilir. Güvenlik denetimi sırasında bu bilgiyi değiştirerek yönetici seviyesinin yükseltilmesi denenmelidir.

Kullanıcıya uygulama tarafından gönderilen tanımlama bilgisinde "giris_yapti=evet" benzeri bir değer olabilir. Güvenlik denetimi sırasında bu bilgiyi göndererek, kullanıcı adı ve şifre bilgilerini kullanmaksızın uygulamaya

erişilip erişilemediği denenmelidir.

Kullanıcıya uygulama tarafından gönderilen tanımlama bilgisinde "kullanici=skolat" benzeri bir değer olabilir. Güvenlik denetimi sırasında bu bilgi değiştirilerek başka bir kullanıcının yerine geçme işlemi denenmelidir.

Web uygulamaları, tanımlama bilgisi kullanarak buna benzer değerleri kullanıcı tarafında tutmak yerine, oturum bilgisi kullanarak buna bağlı değerleri kendi taraflarında tutmaktadırlar.

Web uygulamasında özel oturum bilgisi üretme algoritmaları kullanılıyorsa, bu algoritmaların rastgele bir değer üretip üretmediği kontrol edilmelidir. Bu işlemi gerçekleştirmek için web uygulaması tarafından kullanıcıya gönderilen oturum bilgisi değerlerinden belirli sayıda örnek alıp, bunun belirli bir sırada artıp artmadığı kontrol edilmelidir. Sıralı bir artış varsa, kötü niyetli bir kullanıcı kendisine verilen oturum bilgisini değiştirerek başka kullanıcıların yerine geçebilir.

g. Hata işleme problemlerinin tespiti:

Web uygulamasının kendi içerisinde oluşabilecek hataları kullanıcıya göstermemesi gerekmektedir. Hata denetimi yapılmayan bir uygulama, yapısı hakkında detaylı bilgiyi açığa çıkartabilir.

Web uygulamasında bulunan sayfalarda, hata işleme problemlerinden etkilenen değişkenleri bulmak için, değişkenlere uygulamanın o değişkenden beklemediği bir değer gönderilmelidir. Yapılabilecek istekler aşağıdaki gibi sayılabilir:

- Varsayılan değeri sayısal olan değişkene harf içeren değerler göndermek,
- Varsayılan değeri harf olan değişkene sayısal veri içeren değerler göndermek,
- Form alanlarında bulunan maksimum değerli verilere, belirtilen değerden daha uzun veriler göndermek.

Web uygulaması bu problemden etkileniyorsa, kendi işleyişi ile ilgili bilgi içeren hata mesajlarını kullanıcıya gösterecektir.

3.4.3 Raporlama Aşaması

Yapılan testler sonucu elde edilen bulgular, yazılı olarak anlaşılır bir şekilde raporlanmalıdır. Rapor içerisinde tespit edilen güvenlik problemi, web uygulamasının hangi bölümünde tespit edildiği, problemin giderilmesi için gerekli öneriler ve sonuç detayları bulunmalıdır.

3.4.4 Değerlendirme

Web uygulamalarında sık karşılaşılan güvenlik problemlerinin denetlenmesi aşamasındaki testler sadece bunlarla sınırlı değildir. Yukarıda açıklanan güvenlik problemlerinin denetiminde yapılacak olan isteklerin birden fazla çeşiti bulunmaktadır.

Günümüzde kullanılan teknolojilerle birlikte web uygulamaları, fazlasıyla karmaşık hale gelmektedir. Kapsamlı bir web uygulamasında yüzlerce form alanı, kullanıcı tarafında işlenen dinamik içerik, fonksiyonlar ve diğer bileşenler bulunabilir. Bilgi güvenliği konusunda uzman kişilerin yaptığı testlerde daha fazla güvenlik problemi bulunabilmesi ihtimali olmasına rağmen, her bir bileşenin farklı güvenlik açıklarına karşı teker teker test edilmesi haftalar veya aylar sürebilir. Oysa ki bu işlemleri otomatikleştirebilecek bir yazılım, uygulamaların kısa sürede güvenlik testlerine tabi tutulmasını sağlayacaktır.

Denetim yazılımları bu işlemi yaparken, aynı zamanda uygulamadaki tüm parametrelerin test edildiğinden emin olunmasını, yani testlerin eksiksiz yapıldığını garantilemektedir.

Bu yazılımların otomatik, hızlı ve düşük maliyetli olmalarının yanında, bazı eksik yanları da bulunmaktadır. Bir uzman tarafından yapılan testlerle karşılaştırıldığında bu yazılımlar:

1. Daha az bulguya ulaşabilmektedirler,
2. Hatalı algılama oranları yüksektir,
3. Güncellenmediği taktirde, web uygulamalarını yeni bulunan saldırı metodolojilere karşı test etmekte yetersiz kalmaktadırlar,
4. Denetledikleri web uygulamasının yapısına göre başarıları değişebilir,

5. Bazıları, standart yanıtlar döndürmeyen web sunucularını veya web uygulamalarını test etmekte yetersiz kalmaktadır denilebilir [26].

4. RFID UYGULAMASI

4.1 Firma Tanıtımı

- Otomotiv sektöründe önemli bir yeri olan kurumun, tarihi 1950 yıllara dayanmakla birlikte, İnönü fabrikası için temel atma yılı 1979 yılı olarak gerçekleştirmiştir. Fabrika takip eden yıllarda;
- 1983: OHC & Dover Motor Üretimine Başlamıştır
- 1984: Ağır ticari araç(Cargo) montajı Başladı
- 1986: İlk yerli dizel Motor(ERK) geliştirildi.
- 1992: Hafif ticari araçlar için 2,5 DI Motor& Hipoid dişli üretimi başladı.
- 1995: Mt-75 Şanzıman üretimi başladı
- 1997: Transit arka aks üretimi başladı
- 2000: Puma motor&transit ön düzen üretimi başladı
- 2003: Yeni saç kabinli Cargo&7,3 Lt Eqotork Motor üretimi başladı
- 2006: Yeni transit(v347/8) güç aktarma parçaları üretimine başlandı
- 2007: Common-rail ve EFI Puma motor üretimine başlandı
- I5 Puma motor üretimine başlandı
- LT476 Cargo&9lt Eqotork motor üretimine başlandı
- 2009: 2.2 ltönden çekişli Transit için Puma motor üretimine başlandı
- 2010: 10,5 MY Cargo Euro-5 Motor Cargo İnsaat Serisi üretimine başlandı
- 2011 Global puma motor üretimlerine başlanmıştır.

Fabrika yıllık üretim kapasiteleri;

- 11000 adet Cargo Kamyon
- 55000 adet 4 ve 5 silindir Puma Motor
- 11000 adet 6 Silindir NHDD Motor
- 140000 adet Arka Aks
- 70000 adet Ön Düzen Talaşlı
- 210000 adet Öndüzen Montaj

Temel başlıkları altında gerçekleşmektedir.

Ayrıca kurum birçok farklı alanda onlarca ödül alarak üretim standartlarını belgelemiştir. Bunlardan bazıları;

- 1992: Ford Tedariçi Ödülü
- 1997: İSO 9001 Sertifikası
- 1998: İSO 14001 Sertifikası
- 1989-91-95-99-2005: Eskişehir Sanayi odası Teknoloji Ödülleri
- 2000: Ford Müşteri odaklı 6 Sigma Ödülü
- 2009: OHSAS 18001 Sertifikası
- 2011: Kocaeli Sanayi Odası- Sahabettin Bilgisu Çevre ödülü
- 2011: İstanbul Sanayi Odası Çevre Ödülü

Olarak özetlenebilir.

Tez için öncelikli plot uygulama lokasyonunda bugün itibariyle 1500'e yakın personel çalışmakta olup Kamyon pazarında önemli bir pazar payına sahiptir.

4.2 Süreç Kapsamı

Tez dahilinde planlanan çalışmada, fabrika müdürlüğü bünyesinde ihtiyaç duyulan RFID ile Personel Servis Araçları Takip için gerekli yazılım, donanım, veritabanı, lisans, hizmet ihtiyaçlarını içermektedir. Tez bünyesinde mevcut konu incelenerek, darboğazlar belirlenmiş ve web tabanlı bir yazılım aracılığıyla RFID sistem dinamikleri kullanılarak, araç tanıma sistemi geliştirilmiştir.

Süreç dinamikleri incelendikten sonra yazılım geliştirme kısmı C# programlama dilinde gerçekleştirilmiş olup, her başlık ayrı bir parametre yapısı olarak kodlanmış, sonrasında bu kodlamalar silverlight uygulamalarıyla işler hale getirilmiştir.

Aşağıdaki bölümlerde süreç kapsamında planlanan, uygulama isterlerinin amaçı, mevcut durumu ve sunulan çözümlerin detayları açıklamaları mevcuttur.

4.2.1 Sürecin Amacı

Fabrika bünyesinde, vardiya başına 4 ayrı güzergahtan 38'er araç girişi standart olarak gerçekleşmektedir. Standart üretimde Fabrika üç vardiya üretim gerçekleştiriyor olup 3*2*38 araç hareketi standart olarak gerçekleşiyor olup fazla mesai durumlarında bu sayı dahada artmaktadır. Bir aylık ortalama bi süreçte takip edilmesi gereken ortalama 5500-6000 arası servis araçı giriş çıkış işlemi gerçekleşmektedir. Süreçte outsource edilen personel taşıma işlemleri için, belirlenen rakamlar üzerinden fabrika ilgili firmaya ödeme yapmaktadır. Ödeme fabrika personelinin manuel olarak tespit ettiği araç-giriş çıkış adetleri ve zamanları temel alınarak gerçekleştirilmektedir. Mevcut süreçte ilgili personel her variya başlangıç ve bitiş zamanlarında araç hareketlerini, zaman ve ve plaka bazında kayıt altına alarak süreçin hakediş(ödeme) kısmı için gerekli doneyi sağlamaktadırlar.

4.2.2 Sürecin Mevcut Durumu

Fabrika yerleşkesinde personel servis araçları takibi manuel olarak güvenlik birimi tarafından görevlendirilen, hergün için farklı personeller tarafından yapılmaktadır. Vardiya giriş ve çıkışlarında o gün güvenlik biriminde uygun olan bir personel, manuel olarak araç giriş dakikalarını, güzergahlarını, plakalarını kayıt altına alarak ilk amirliğine raporlamaktadır. İlgili amirlikten onaylanan kayıtlar Endüstri İlişkiler ve İdari İşler Ekip Liderliğine raporlanıp, süreç devamı ve hakediş işlemleri gerçekleştirilmektedir. Mecvut personel taşımacılık takibinde yaşanan problemler;

- Servis giriş-çıkışlarında plaka bilgilerinin elle tutulması,
- Personelin dikkat etmediği veya araçların bir anda yoğunlaştığı durumlarda araç kayıtları eksik yapılması,
- Farklı güzergahlardan gelen araçların kayıtlarının yanlış tutulması,
- Araç geliş zamanlarının hatalı belirlenmesi,
- İşlemin hakediş kısmında, güvenlik ve sosyal hizmetler arasında sıkıntılar yaşanması,

- İç denetim ve holding denetimlerinde konu üzerinden problemler yaşanması,
- Kayıt işlemini yapan personelin ilgili birimin vardiya yapılarına ve personel akışlarına göre sürekli değişmesi, işin yapılışında profesyonelleşmeyi önleyerek, olası hataların oluşmasına zemin hazırlamaktadır.
- Geç kalma durumlarına itirazların yaşanması, işi takip eden personel ve taşıyı firmanın standart bir zaman üzerinden işlem yapamaması şeklinde özetlenmiştir.

4.2.3 Girdi ve Çıktılar

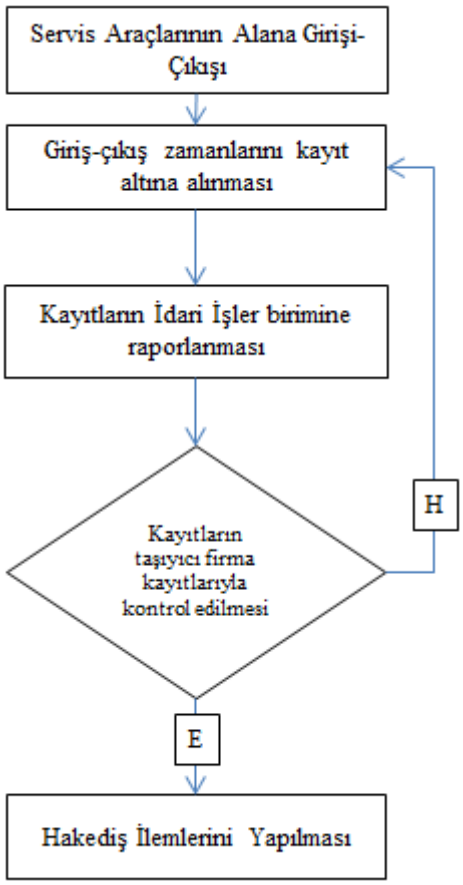
Sürece etki eden olaylar;

- Araç giriş çıkış saatleeri
- İlgili personelin yürüttüğü kayıt işlemleri
- Veri kayıtlarını manuel olarak tutulması ve geriye dönük veri sorgulamasında yaşanan güçlükler.

Girdiler; Personel servis araçlarının giriş-çıkış saatleri.

Çıktılar; Araç giriş-çıkış kayıtları.

4.2.4 Akış Diyagramı



Süreç servis araçlarının vardiya başlangıçlarında veya bitişlerinde alanan girişleriyle ve çıkışlarıyla başlamaktadır. Burada kritik olan kısım alandan ayrılışlarından daha çok alana girişlerdir.

Personelin iş başı yapacağı vardiya zamanından en az 15 dakika önceden fabrikaya giriş yapması, hem cezai işlemler hem de hakediş aşamalarında önem taşımaktadır.

Servislerin ilgili vardiya saatlerinde alan giriş, çıkış bilgileri güvenlik biriminden görevlendirilen bir personel tarafından, giriş-çıkış saati, güzergah bilgisi ve plaka bazında kayıt altına alınarak, Endüstri İlişkiler ve İdari İşler birimine raporlanır. İdari İşler biriminden görevli personel elindeki kayıtları, taşıyıcı firma kayıtları ile karşılaştırarak doğruluk kontrolü yapar. Teyitleme yapıldıktan sonra İlgili ekip lideri onayının ardından hak ediş işlemleri her ay yapılır.

Kayıltarın birbirini tutmadığı zamanlarda, firma verileri ve o ay içindeki hareket bilgileri geriye doğru taranarak uyumsızluk yaratan kayıtlar tespit edilir.

4.2.5 Metrikler

Mevcut süreçte, vardiya girişlerinde ve bazı vardiya çıkışlarında güvenlik biriminden görevli farklı personeller; servis araçlarının giriş ve çıkışlarını takip edip giriş-çıkış zamanalarını belirlemek için bir gün içinde aşağıdaki tabloda belirtilen sürelerde çalışma yürütmektedirler.

Çizelge 4.1. Vardiya Grupları

Gruplama İsimleri	ZamanAralıkları (dakika)
08:00 - 18:00 Vardiyası Giriş	50
16:00 - 00:00 Vardiyası Giriş	45
08:00 - 16:00 Vardiyası Çıkış	45
08:00 - 18:00 Vardiyası Çıkış	35
21:00 Fazla Mesai Çıkış	35
22:00 - 08:00 Vardiyası Giriş	30
22:00 Fazla Mesai Uzama Çıkış	20
00:00 - 08:00 Vardiyası Giriş	40
08:00 - 18:00 Vardiyası Çıkış	30
16:00 - 00:00 Vardiyası Çıkış	40
Toplam	370

4.2.6 Sorumluluk Alanları ve Roller

Güvenlik Birimi: Kayıtların tutulması

Endüstri İlişkiler ve İdari İşler E.L: kayıtların kontrolu, cezai işlemlerin takibi ve kayıtlar paralelinde hak ediş işlemleri.

4.2.7 Sürecin gereksinimleri

Süreçte en kritik faktör kayıtların; saat, giriş-çıkış adedi ve güzergah bazında doğru tutulup, raporlanmasıdır. Çünkü hakediş işlemleri güzergahlardan gelen araç sayılarına göre km bazında gerçekleştirilmektedir. Hakediş Km bazında gerçekleştirildiği için araçların, İnönü, Bozüyük, Çukurhisar ve Eskişehir güzergahlarından hangisinden geldiği veya hangisine gittiği hatasız olarak belirlenmelidir. Bu güzergahların içinde barındırdığı her bir rotanın Km bilgileri kayıt altında olup, bu kayıtlar paralelinde süreç işletilmektedir.

Diğer bir önemli faktör ise araçların alana, belirlenen zaman dilimleri içinde gelip gelmediğinin tespit edilmesidir. Belirlenen zaman dilimlerine göre ceza gerekliliklerin yerine getirilmesi kararlaştırıyor olup, taşıyıcı firma ile yaşanabilecek olası anlaşmazlıklarında önüne geçilmesi için giriş-çıkış zamnlarının standart ve hatasız tutulması gerekmektedir.

4.2.8 Mevcut Süreci Destekleyen Araçlar

Mevcut sistem işleyişi, yukarıdaki bölümlerde detaylı olarak anlatılmıştır.

4.2.9 Mevcut Süreç Verim, Verimsizlik, Problem Noktaları

Mecvut personel taşımacılık takibinde yaşanan problemler aşağıda özetlenmiştir;

- Servis giriş-çıkışlarında plaka bilgilerinin elle tutulması,
- Personelin dikkat etmediği veya araçların bir anda yoğunlaştığı durumlarda araç kayıtları eksik yapılması,
- Farklı güzergahlardan gelen araçların kayıtlarının yanlış tutulması,
- Araç geliş zamanlarının hatalı belirlenmesi, (Zaman zaman otobüs söförlerinin alana girdiklerini söyledikleri zaman dilimi ile ielgili personelin kayıtları arasıda farklılıklar oluşmakta, bu farklılıklardan kaynaklanan problemler oluşmaktadır)
- İşlemin hakediş kısmında, güvenlik ve sosyal hizmetler arasında sıkıntılar yaşanması,
- İç denetim ve holding denetimlerinde konu üzerinden problemler yaşanması,
- Kayıt işlemini yapan personelin ilgili birimin vardiya yapılarına ve personel akışlarına göre sürekli değişmesi, işin yapılışında profesyonelleşmeyi önleyerek, olası hataların oluşmasına zemin hazırlamaktadır.
- Geç kalma durumlarına itirazların yaşanması, işi takip eden personel ve taşıyıcı firmanın standart bir zaman üzerinden işlem yapamaması.
- En kritik unsurlardan biride, sistemin su anda tamamen insan eliyle yürütülerek, hataya açık olmasıdır.

4.3 Yeni Prosesin Tanımı

Yukarıda belirtilen problemlerin çözüme ulaştırılması ve gelecekte yaşanabilecek olası sorunların önüne geçilmesi maksadıyla sistemdeki insan unsurunun devreden çıkartılarak, veri yönetim sisteminin tamamen dijital ortama aktarılması gerekliliği ortaya çıkmıştır. Bu sistem için RFID etiketleri ve okuyucularıyla desteklenen dijital bir platform araştırılmıştır.

Sistemin, araçlara takılacak araç tanımlarını içeren RFID etiketlerinin; okuyucu antenler tarafından araç hareketlerini kayıt altına alması paralelinde çalışması planlanmıştır. Verilerin, depolanması, kontrolü, raporlanması vb. geliştirilmesi planlanan web tabanlı bir yazılım üzerinden yürütülecektir.

Planlanan sistemde verilerin toplanabilmesi için iki okuyucu anten giriş ve çıkışa olmak üzere yerleştirilecektir. Giriş için planlanan okuyucu konumlandırıldıktan sonra araçların okuyucunun önünden geçmesi için 20-30 metre arasında bir rota hazırlanıp; araçlar buralara yönlendirilerek personelin araçlardan indirilmesi işlemi gerçekleştirilecektir. Çıkış işlemi içinse çıkış noktasına yerleştirilen okuyucu anten çıkış yapan araçların verilerini depolayacaktır.

4.3.1 Önerilen Sistem Özellikleri

- Oluşturulan .Net 4.0 veya üstü ile, IIS üzerinde koşacak şekilde web tabanlı olarak yapılandırılmıştır.
- Sistem paralelinde oluşturulan altyapıda en az %95 güven seviyesinde veya üstünde hatasız veri okuma depolama yapılabilmektedir.
- Sistem, araçların giriş ve çıkış bilgilerini okuyacak ve depolayacaktır. Bu ihtiyaç paralelinde pasif/aktif RFID etiketleri, RFID Okuyucu Antenler ve Veritabanı entegrasyonu ile bilgisayar destekli tur kontrol sistemi planlanmıştır.
- Sistem 1 adet araç girişleri için 1 adet de araç çıkışları için olmak üzere 2 adet RFID okuyucuyu ve 100 adet RFID etiketi destekleyecek sekilde planlanmıştır.
- Sistemde etiket okuma hızı 0,5 saniyenin altındadır.

4.3.2 Önerilen Web Tabanlı Yazılımın Güvenlik Unsurları

1. Uygulamalar için kullanıcı doğrulama işlemi login.ford.com.tr üzerinden yapılmalıdır. Mevcut web sistemlerine entegrasyon işlemiyle gerçekleştirilmiştir.

2. DB şifresi, web server üzerindeki DBConfig dosyasında şifreli olarak tutulmaktadır. Uygulama tarafından buradan okunarak, şifresi çözülüp DB' e bağlantıda kullanılmalıdır. İki şekilde yapılabilir.

- Birinci yöntem, FO DBHelper nesnesini kullanmaktır.
- İkinci yöntemde şifre, BConfig e, yapılacak bir arayüz ile şifrelenerek atılacaktır. Bu arayüzü sadece fabrika(FO) personeli kullanacaktır.

3. Uygulamanın kullandığı veri tabanı ve şema ismi web.config'den alınmalı, kod içerisinde veritabanı ve şema ismi bulunmamalı.

4. Veri tabanı için *Uygulama_İsmi_APP* şeklinde kullanıcı tanımı yapılıp, bağlantıda bu kullanıcı kullanılacak. Bununla ilgili yetki ihtiyacı belirtilip kullanıcının yaratılması sağlanmıştır.

5. Uygulamada SQL Injection a karşı önlem alınmıştır. Harivi veri kutularıyla toplanan veriler bir parametreye bağlanarak işlemin devamı sağlanmıştır.

6. Bütün INSERT, UPDATE, DELETE işlemleri veritabanı prosedürleri (Stored Procedure) ile yapılmıştır. (SELECT komutları da prosedür kullanılarak REF CURSOR döndürülme yöntemiyle de yapılmıştır). Kod içerisinde string sql'ler kullanılmamıştır.

7. .Net uygulamalarında Hata Mesajlarının açık olarak kullanıcıya gösterilmemesi gerekmektedir. Hatalar yakalanıp kullanıcıya anlamlı mesajlar verilmektedir.

8. Güvenli cookie kullanılmıştır.

```
<system.web>

<httpCookies httpOnlyCookies="true" requireSSL="true" lockItem="true" />

</system.web>
```

9. Kullanıcıdan veya Db den gelen değerlerin değerlendirilmesi

- **Black List:** Net uygulamalarında standart olarak black list kontrolü vardır. Özel olarak sayfa içinden ValidateRequest=False veya web.config dosyasında *< pages validateRequest=false />* satırı eklenirse blacklist kontrolü kaldırılmış olur. Tezin uygulamasının yapıldığı fabrika uygulamalarında Black List kontrolü **açık** olmalıdır, dolayısıyla "ValidateRequest" ve "pages validate Request" parametreleri "true" olarak işaretlenmiştir.
- **.NET White list** : .Net Uygulamalarında kullanıcıdan gelen giriş bilgilerini bir white list'ten geçirmek gereklidir. Bunun için her server'a machine.config dosyası yardımı ile sabit bir white list eklenir. Uygulama bazında farklı bir whitelist'e ihtiyaç varsa uygulama ya özel liste web.config yardımıyla verilir. Çözüm:

1. Machine.config dosyasına aşşağıdaki şekilde bir whitelist eklenmelidir.

<addkey="WHITELIST"value="([\\\r\n]|([A-Z]|[a-z]|[0-9]|[_]|[:]|[$]|[-]|[(]|[)]|[€]|[?]|[/]|[]|[ı]|[ý]|[Ý]|[İ]|[ğ]|[ð]|[Ğ]|[ç]|[Ç]|[ş]|[Ş]|[þ]|[ü]|[Ü]|[ö]|[Ö]|[.]|[@]|[,]|[|]|[=] |[!]|[+]|[¦])*" />

2. Uygulama bazında whitelist isteniyorsa uygulamanın web.config dosyasına <add key="WHITELIST" value="([\\\r\n]|([A-Z]|[a-z]|[0-9]|[_]|[:]|[$]|[-]|[(]|[)]|[€]|[?]|[/]|[]|[ı]|[ý]|[Ý]|[İ]|[ğ]|[ð]|[Ğ]|[ç]|[Ç]|[ş]|[Ş]|[þ]|[ü]|[Ü]|[ö]|[Ö]|[.]|[@]|[,]|[|]|[=] |[!]|[+]|[¦])*" /> şeklinde yeni liste eklenmelidir.

- **Veri Tip Kontrolü:**Dışarıdan veya DB(database) den gelen, işlemlerde kullanılacak verinin tipi herzaman kontrol edilmelidir. Örneğin kullanıcıdan gelen ve oracle procedure içinde number alana gidecek değer mutlaka number olmalıdır.
- **Veri Uzunluk Kontrolü:**DB ye parameter olarak geçilen (oracle,com obje,function) değerlerin uzunlukları parameter olarak kullanıldığı yerdeki max. uzunluğundan fazla olmaması kontrol edilmelidir.

10. Ekrana Direk işlem yaparken

- Response.Write ile veya <%= ile direk yazılan dinamik alanlar *HttpUtility.HtmlEncode(TextBox1.Text)* şekinde HtmlEncode edilerek kullanılmalıdır.

Ör: Response.Write(“<script>alert(‘” + HttpUtility.HtmlEncode(TextBox1.Text) + ”’);</script>”)

Dinamik alanlar: Kullanıcı tarafından girilen veya database’den alınan, esas kaynağı kullanıcı olabilecek her türlü alan.

11. Response.Redirect ile giderken veya Ekrana link yazarken

- .Net ekranlarında Response.Redirecle giderken veya bir linki ekrana basarken UrlEncode kullanılmıştır

Ör **:**Response.Redirect("MBSUP-1.pdf?msg="+HttpUtility.UrlEncode(msg))

12. innerHTML’leri innerText yapmak.

- Web uygulamarında innerHTML yerine innerText kullanılmalıdır.

13. Uygulamanın genel parametreleri kodun içerisine gömüştür. Parametrelerin kullanım yerleri gözönünde bulundurularak parametreler konfigürasyon dosyasında (web.config, machine.config) veya veritabanı tablosunda saklanmıştır.

4.3.3 Verileri Vardiyalar Göre Gruplayarak Depolama Altyapısı

Burada planlanan belli zaman dilimleri arasında fabrikaya giriş-çıkış yapan araçların bilgilerinin ilgili vardiyaya kaydedilmesidir. Ayrıca planlanan gruplama altyapısında; belirtilen zaman aralıkları için giriş ve çıkışlarını yapmayan araçlar “ Gecikme” başlığı altında gruplanacaktır.

Gruplama kalemleri ve başlıkları aşağıda Çizelge 4.2 ‘ de belirtilmiştir.

Çizelge 4.2. Vardiya Gruplama Aralıkları

Gruplama İsimleri	Zaman Aralıkları
08:00 - 18:00 Vardiyası Giriş	7:30 - 7:55
08:00 - 18:00 Vardiyası Geç Giriş	7:45'den sonra
12:00 - 08:00 Vardıyası Çıkış	08:15 - 08:30
16:00 - 00:00 Vardiyası Giriş	15:30 - 15:55
16:00 - 00:00 Vardıyası Geç Giriş	15:45'den sonra
08:00 - 16:00 Vardiyası Çıkış	16:10 - 16: 30
08:00 - 18:00 Vardiyası Çıkış	18:15-18:30
21:00 Fazla Mesai Çıkış	21:15-21:30
22:00 - 08:00 Vardiyası Giriş	21:35 - 21:55
22:00 - 08:00 Vardiyası Geç Giriş	21:45'den sonra
22:00 - Fazla Mesai Uzama Çıkış	22:15 - 22:30
00:00 - 08:00 Vardiyası Giriş	23:30 - 23:55
00:00 - 08:00 Vardiyası Geç Giriş	23:45' den sonra
16:00 - 00:00 Vardiyası Çıkış	00:15 - 00:30

Yukardaki tabloda belirtilen vardia gruplamaları için vardiya aralıklarının ve gruplama isimlerinin kullanıcı tarafından dinamik olarak belirlenebileceği bir arayüz tasarlanmıştır.

4.3.4 Otomatik Mail Desteği

Yukarıdaki bölümde anlatılan "Gecikme" durumu, geç kalan araçların olması durumunda, yazılıma belirtilecek mail adreslerine otomatik olarak raporlama opsiyonunu içinde barındıran bir yapı oluşturulmuştur. Detaylar aşağıdaki bölümlerde belirtilmiştir.

4.3.5 Raporlama Opsiyonları

- Zaman dilimlerine göre (vardiyalar)
- Plaka bilgilerine göre
- Güzergah bilgilerine göre
- Şöför bilgilerine göre
- Geç Kalma durumuna göre
- Ay,yıl ve gün bazında sistemden rapor çekilebiliyor olmalıdır.

Sistem raporlar sayfası, Office 2010 excel ve word formatlarında kaydedilebilme özelliğini desteklemelidir.

4.3.6 Arama Opsiyonları

Sistem içi arama motoru uygulaması bulunmaktadır. Burada kasıt; belirlenen zaman aralıkları için herhangi bir aracın bilgilerinden bir tanesinin sisteme girilerek diğer bilgilerinin ve belirtilmiş aralık araç hareketlerinin listelenmesidir.

- Sistemde her bir etiketin belirlenen çevrim içerisinde bir kere okunmasına olanak sağlayan ve alandaki çarpışmasını engelleyen algoritmalar veya yapılar bulunmalıdır.

- Sistem paralelinde geliştirilen yazılım çoklu kullanım ve yetkilendirme altyapılarını sağlamalıdır.

Yetkilendirme yapıları aşağıda belirtilmiştir.

Admin: Yetkilendirme yapabilme ve tam denetim.

Tam Denetim: Programın bütün özelliklerini kullanabilme. (RFID etiket Tanımlama, söför tanımlama, güzergah tanımlama, vardıya adı ve gruplama zaman aralıkları tanımlama)

Kullanıcı: Programın tanımlama dışındaki, kullanım hakları.

4.4 Güvenlik Unsurları

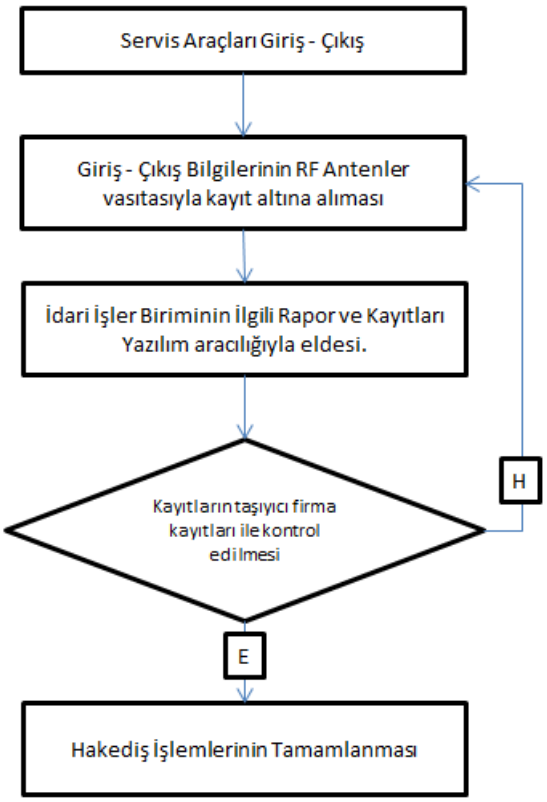
- Sistem Avrupa Telekominikasyon Standartları enstitüsünün izin verdiği bir frekans aralığıyla çalışmalıdır.
- Sistem tamamen kendi içinde bir frekans aralığında çalışarak fabrika ve çevresindeki diğer elektronik aletlerle etkileşim içinde bulunmamalıdır.
- Sistem paralelinde kullanılan etiketler takıldıkları yerden söküldüğü anda işlevini kaybetmelidir.

4.4.1 Girdi ve Çıktılar

Girdiler : Servis Araçları giriş- çıkışları

Çıktılar: Servis araçları giriş çıkış kayıtları, sistem raporları

4.4.2 Akış Diyagramı



4.4.3 Akış Diyagramı Açıklaması

Önerilen sistemde; süreç, mevcut durumda olduğu gibi servis araçlarının alan giriş- çıkışlarıyla başlamaktadır.

Servis araçlarına takılan RFID antenlere tanımlanan servis araçlarının bilgileri, Rf Antenler vasıtasıyla okunarak giriş- çıkış zamanlarıyla birlikte depolanacaktır.

Bu kayıtlar geliştirilen yazılım sisteminden eş zamnalı kontrol edilerek ve daha sonrasında raporlanarak, hakledіş işlemleri için gerekli bilgi eldesinden sonra, sistemden elde edilen kayıtlar taşıyıcı firma kayıtlarıyla karşılaşmalı olarak kontrol edilip onaylandıktan sonra, hak ediş sürecine geçilecektir.

4.4.4 Metrikler

Mevcut süreçte bütün vardiyalarda araç giriş/çıkış bilgilerini kontrol ve kayıt altına almak için güvenlik birimi tarafından görevlendirilen farklı personeller farklı zaman aralıklarında günde ortalama toplamda 370 dakika çalışma yürütmektedirler. Önerilen sistemde bu işlem için güvenlik birimi tarafından görevlendirilen herhangi bir personel ihtiyacı söz konusu değildir. Ayrıca önerilen sistem belirtilen vardiyaların, gün, ay ve yıl bazında araç giriş-çıkış bilgilerinin kayıt altına alacaktır.

4.4.5 Sorumluluk Alanları ve Roller

Önerilen sistemde, Servis araçları taşıma süreci tek merkezli hale getirilecektir. Daha öncesinde güvenlik birimiyle ortak olarak ilerletilen süreç, yeni yapıda sadece Endüstri İlişkiler ve İdari İşler E.L personelleri denetimine verilecektir.

4.4.6 Yeni Prosesi Destekleyecek Mantıksal Veri Akışı

Önerilen sistemde; araç giriş çıkış zaman kayıtları; Tablo 1. Vardiya Gruplama Aralıkları'da belirtilen zamanlara göre ilgili vardiya için özelleşmiş alanlarda kayıt altına alınacaktır. Giriş, çıkış kayıtları sisteme yapılan tanımlamalarla birlikte tutularak; bu kayıtlar ihtiyaçlar doğrultusunda raporlanarak, sürecin devamlılığı sağlanacaktır.

Bunun dışında Bölüm 6.8 de anlatılan otomatik mail yapısı ise, günü gelen hatırlatmaları veya geç kısmında kayıt altına alınan araç giriş/çıkış bilgilerini belirtilen kullanıcılara mail atmasını sağlayan bir yapıdır.

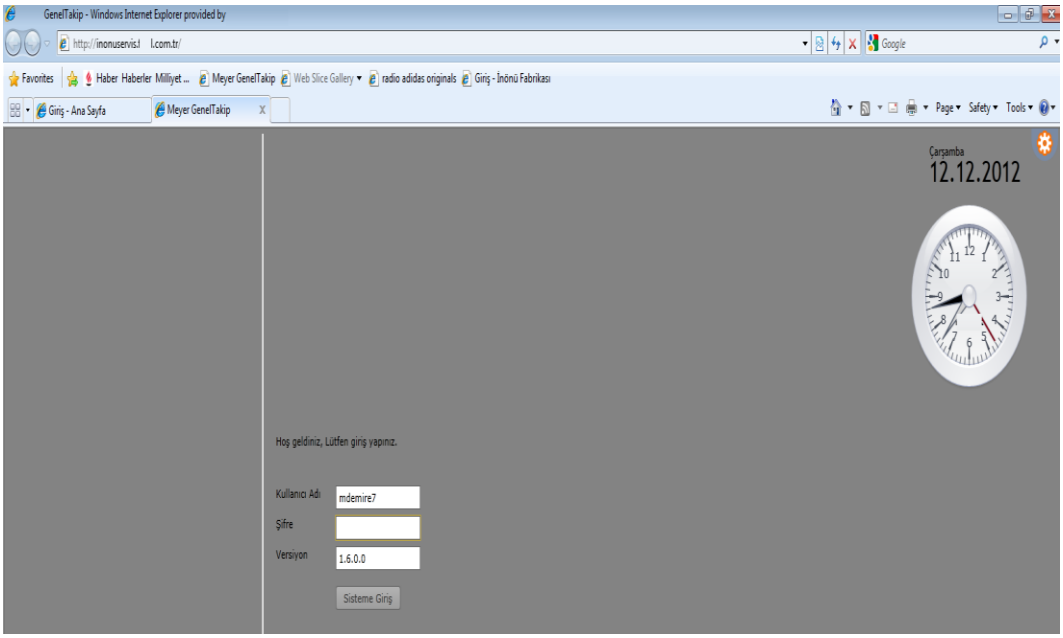
4.4.7 Diğer Süreçler ve Sistemler ile Olan İlişkiler

Uygulama içinde tanımlanacak kullanıcılar, Ford Otosan Active Directory de tanımlı olan kullanıcılar olup; uygulamanın kendi içinde ayrı bir kullanıcı kaydı tutmaması istenmemektedir.

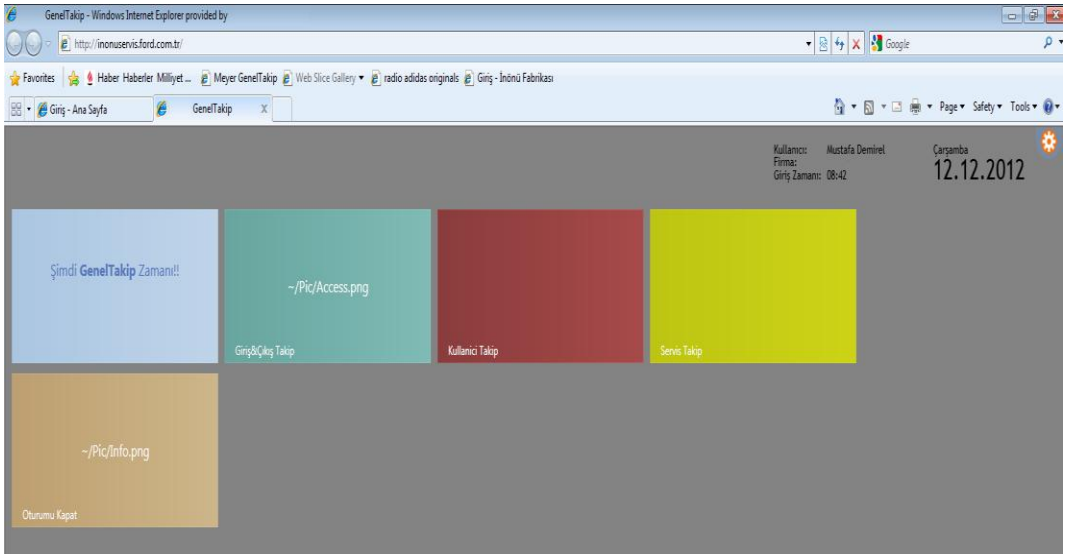
Ayrıca uygulamaya girişte kullanıcı şifre giriş işlemleri de Ford Otosan Login Sistemi üzerinden gerçekleşecektir.

4.5 Ara Birimlere Ait Çizimler, Rapor Tasarımları, Arayüzler ve Tarifler

Sistemde bir anasayfa ekranı üzerinden, bütün alt birimlere ulaşılabilen bölümler ve kısayollar bulunmaktadır.

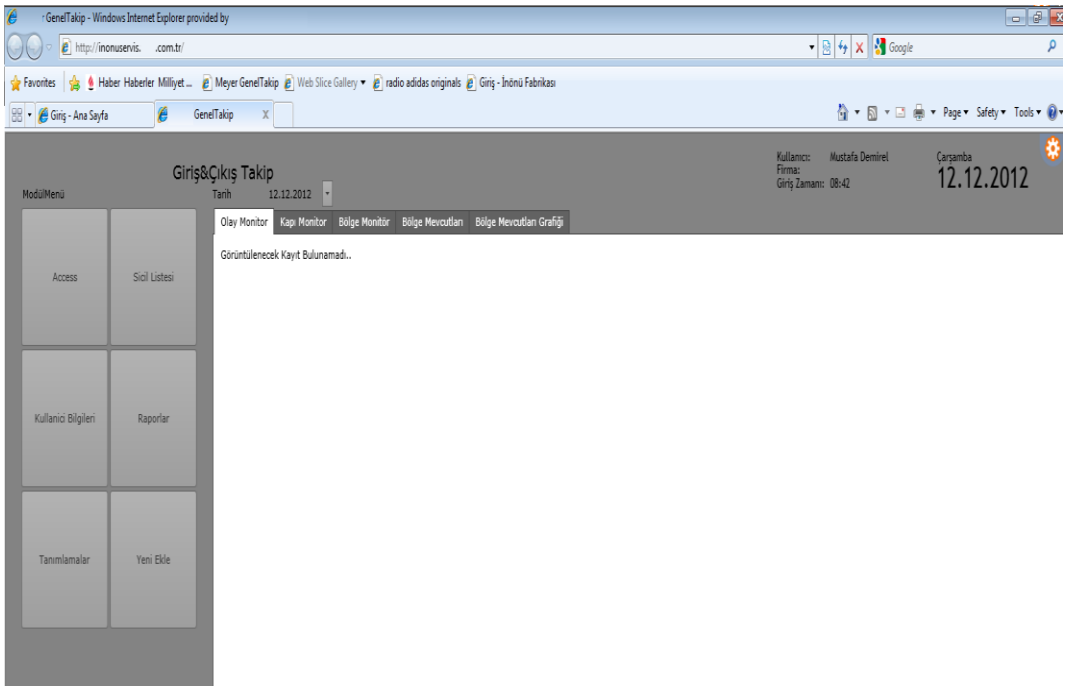


Şekil 4.1. Anasayfa Giriş Ekranı



Şekil 4.2. Ana Menü Ekranı

Sistem ana sayfasında araçların giriş çıkışlarını anlık olarak takip edebilen aşağıdaki formattaki gibi bir bölüm bulunmaktadır. Bu ekranda her vardiya aralığı için; tanımlanan vardiya grubu, plaka, güzergah, giriş-çıkış adet; giriş, çıkış tarihi bulunmaktadır.



Şekil 4.3. Araç Giriş-Çıkış Takip Ekranı

Anasayfada aşağıdaki bölümlerde tariflenen “Açıkta kalan Araç” araç listesine dair; Plaka, Güzergah, Giriş/Çıkış zamanı Şöför adının listelendiği bir uyarı ekranın bulunmaktadır.

Anasayfada aşağıdaki bölümlerde anlatılan “Geç Kalan Araç” listesine dair; Plaka, Güzergah, Giriş/Çıkış zamanı Şöför adının listelendiği bir uyarı ekranın bulunmaktadır.

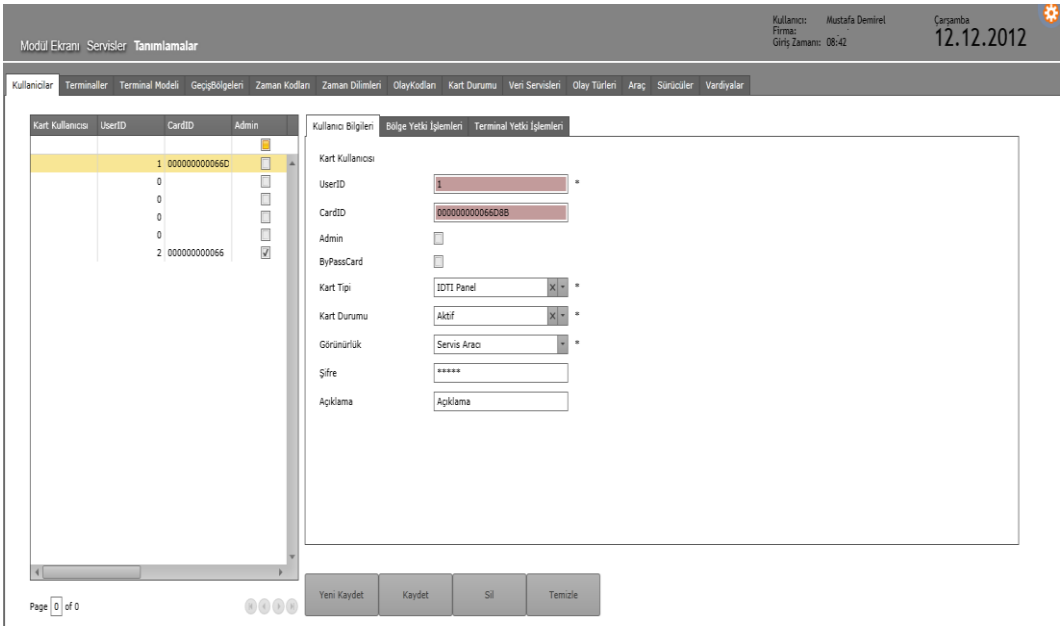
4.5.1. Etiket Tanımlama Sayfası

Bu birimden sisteme yeni tanımlanancak olan RFID etiketlerine ilgili başlıklarda veri girişi yapılarak sisteme dinamik olarak kayıt eklenebilmektedir.

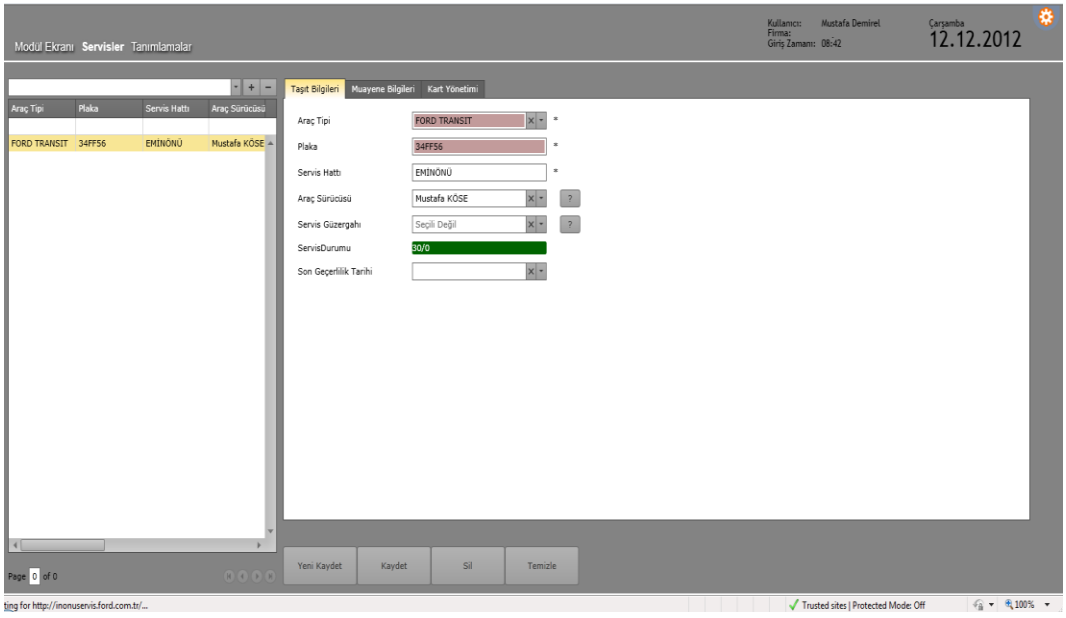
RFID Etiketlerine tanımlanabilmesi istenen başlıklar;

- Araç Plaka Bilgileri
- Araç Tipi
- Araç Modeli
- Sürücü Bilgisi
- Güzergah bilgisi olarak istenmektedir.
- Sigorta başlangıç bitiş Tarihleri
- Emülsiyon Başlangıç Bitiş Tarihleri
- Fenni Muayene Başlangıç Bitiş Tarihleri

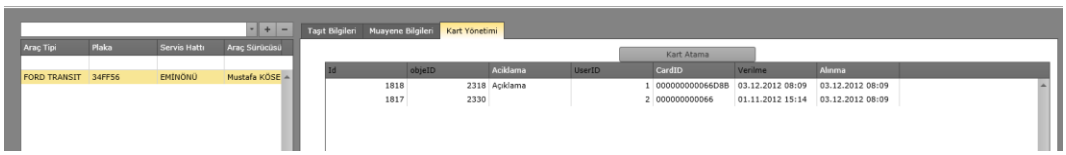
Sorgulama başlığında ise; minimum araç plakası ve/veya sürücü bilgilerine göre; tanımlanan diğer bilgiler sorgulanabilmektedir.



Şekil 4.4. Tanımlama Ekranları

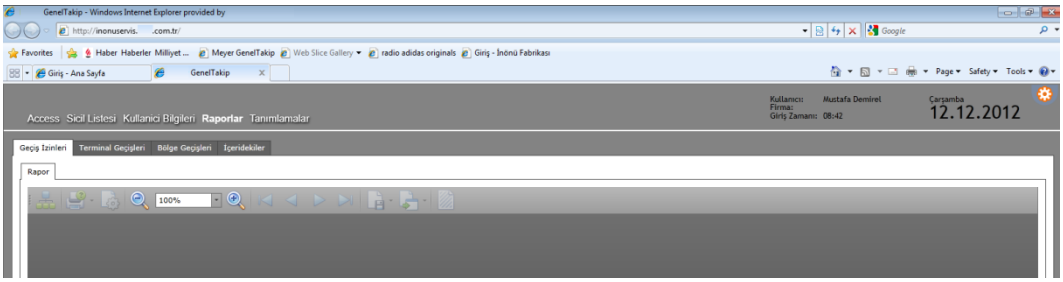


Şekil 4.5. Taşıt Bilgileri Tanımlama Ekranı



Şekil 4.6. Kart Atama Ekranı

Sistemde arayüzdeki gibi bir listeye mevcut kayıtlı araç bilgileri görünütlenebilyor ve aşağıdaki örnek format paralelinde raporlanabilmektedir.



Şekil 4.7. Araç Bilgileri Ekranı

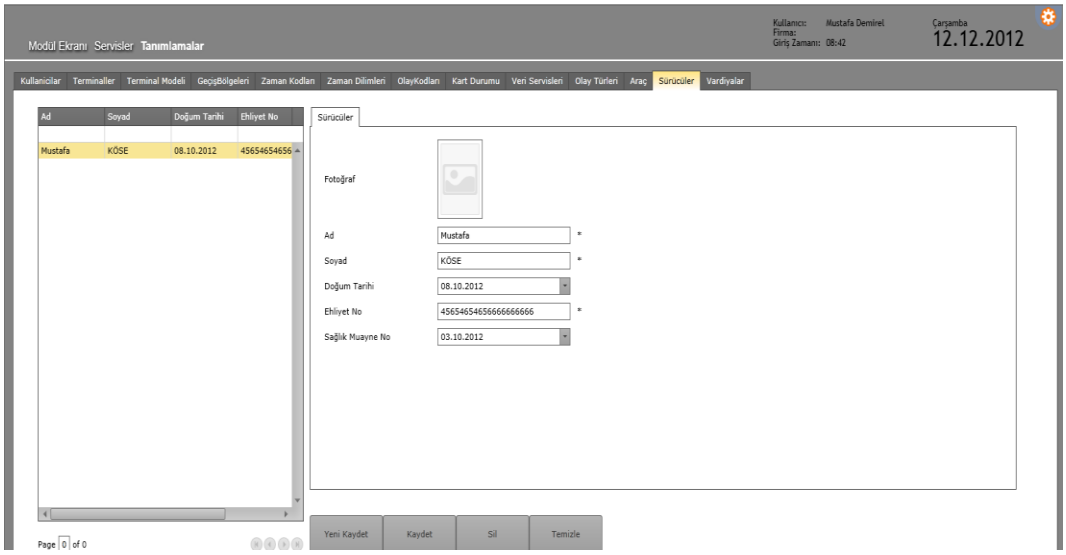
Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt edilebilp ve yazdırılabilir formattadır. Örnek rapor formatı aşağıdaki gibidir.

4.5.2.Tanımlamalar

Sistemde "Tanımlamalar" başlığı altında yapılan tüm tanımlamalar; ilgili yerlerde listelerden seçilebilir durumdadır.

a. Şöför Sicil Bilgileri:

Sistemde araç şöförlerinin Ad; Soyad; doğum tarihi, TC No; Ehliyet No bilgilerinin kayıt altına alınabildiği ve mevcut araç söförlerinin listesinin görülebildiği bir arayüz tasarımı bulunmaktadır.



Şekil 4.8. Şöför Sicil Bilgileri

Sistemde görülen kullanıcı kayıtları, ad , soyad, doğum tarihi, TC no, Sağlık Muayene Tarihi ve Ehliyet No sütünlarında raporlanabiliyor durumdadır . Sistem içerisinde ayrıca; minimum ad, soyad bilgilerine göre arama yapılabilme altyapısı bulunmalıdır.

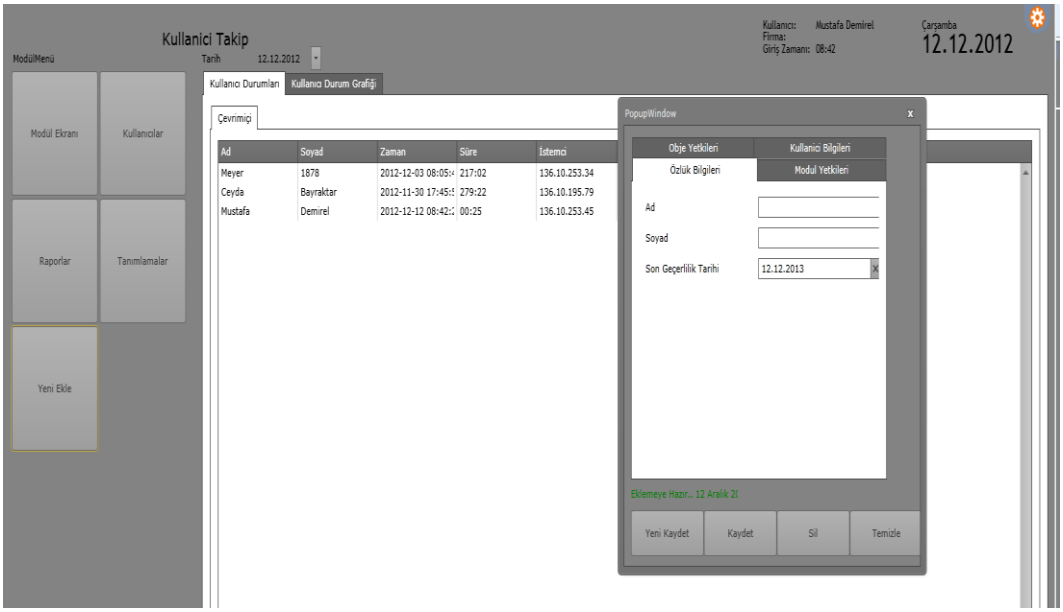
Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, Word, Pdf ve Excel olarak kayıt edilebilmektedir.

b. Kullanıcı yetkilendirme:

Sisteme, belirlenen kullanıcılara yetkilendirme yapılabilecek bir format bulunmaktadır. Yetkiler aşağıdaki şekilde sınıflandırılmıştır;

Tam Denetim: Programın bütün özelliklerini kullanabilme. (RFID etiket Tanımlama, söför tanımlama, güzergah tanımlama, vardıya adı ve gruplama zaman aralıkları tanımlama)

Kullanıcı: Programın tanımlama dışındaki, kullanım hakları. Burada kastedilen; normal kullanıcının bütün raporlama işlemlerinin ve manuel araç giriş işlemlerini yerine getirebiliyor olmasıdır. Admin yetkilendirme fabrika sistemleri üzerinden yapılmıştır.



Şekil 4.9. Kullanıcı Yetkilendirme Altyapısı

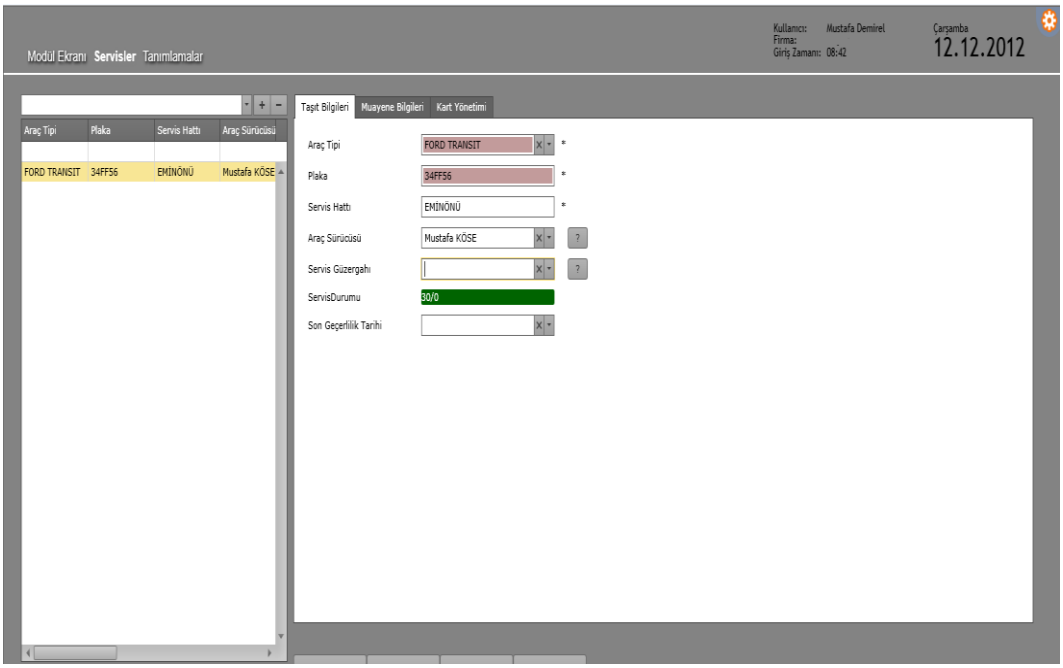
c. Güzergah Giriş Ekranı:

Sistemde, Yukarıdaki bölümlerde anlatılan "Tanımlamalar" Başlığı altında kullanıcıların güzergah bilgilerini tanımlayabilecekleri bir güzergah tanımlama arayüzü mevcuttur. Bu ekran üzerinden tanımlanan güzergahlar ilgili diğer raporlar ve alanlarda görüntülenebilmektedir.

Sisteme kayıt edilen güzergah bilgileri raporlanabilmektedir. Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt altına alanabiliyor durumdadır.

d. Araç tanımlama:

Sistemde, üzerine RFID etiket takılan araçları, Plaka, Araç tipi ve Modellerine göre tanımlama yapılabilecek; mevcut araç listesini görüntüleyebilecek arayüzler bulunmaktadır.

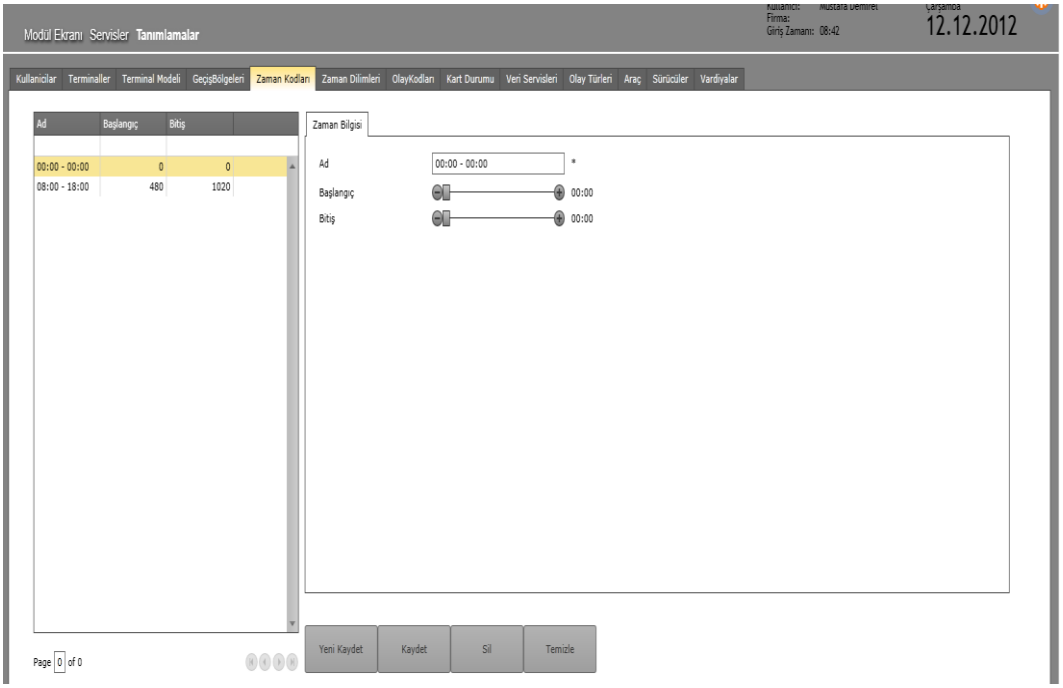


Şekil 4.10. Araç Tanımlma Ekranı

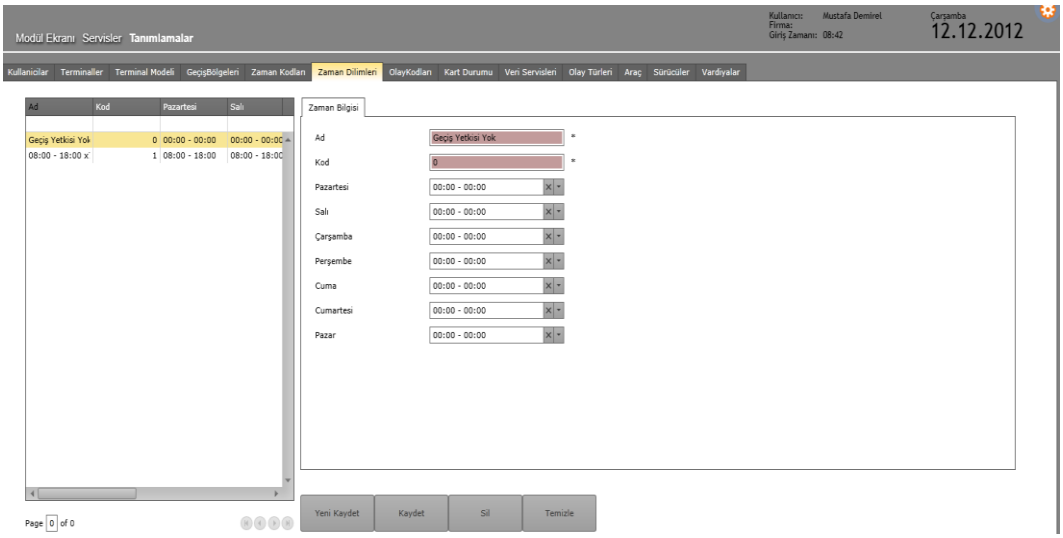
Bu ekran üzerinden tanımlanan araç bilgileri ilgili diğer raporlar ve alanlarda görüntülenebilir durumdadır.

e. Vardiya bilgileri ve zamana göre gruplama aralıkları:

Sistemde, önceki bölümde anlatılan vardiya bilgilerininin belirtilen saat aralıklarına göre gruplanması işlemi gerçekleştirilebil durumdadır. RFID etiket veri gruplama zamanları; sisteme dinamik olarak belirtilen zaman aralıklarına göre, o zaman aralıkları için tanımlanmış vardiya başlıklarına kayıt edilebilir durumdadır. Bu vardiya başlıkları ve gruplama zaman aralıkları şartların gerekliliklerine göre (mevsimsel vb) değiştirilebilir bir yapıya sahiptir.



Şekil 4.11. Vardiya Tanımlama Ekranı



Şekil 4.12. Vardiya Zaman Aralıklarına Veri Tanımlama Ekranı

Sistemde görülen vardiya kayıtları, vardiya tanımı ve gruplama aralıkları sütünlarıyla raporlanabilir durumdadır.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebilir, word, pdf ve excel olarak kayıt edilebilir durumdadır.

Sistemde Geç Kalma durumu mevsimsel sartlar ve o anki politikalar paralelinde değişmektedir. Yukarıdaki ekrandan her vardiya için tanımlanan geç kalma aralıkları için saat aralıkları tanımlanabilir formattadır. Bunun yanında geç kalma durumunda +5 dakikaya kadar olan kısım; bu zaman dilimine tanımlanmış en yakın vardiyaya eklenmeli, bu 5 dakikanın üzerinde geç kalan araçlar aşağıdaki bölümde anlatılan açıkta kalan araçlar listesine eklenmektedir.

Geç kalma ve açıkta kalma durumlarında oluşan listeler; yukarıdaki bölümlerde tariflendiği gibi; anasayfa üzerinde belirtilen alanlarda yayınlanmaktadır.

Bu tanımlama ekranıyla belirtilecek aralıklar; planlanan sistemde standart aralıklardır. Standart dışında bir araç giriş çıkışıyla karşılaşıldığı durumlarda veri kaybının önüne geçilmelidir. Bunun için aynı geç kalma durumlarında planlanıldığı gibi; belirlenen araç giriş çıkış zaman aralıkları dışında kalan araçlar giriş-çıkış verilerinin tutulması için " Tanımsız Vardiya" veya "Açıkta Kalan Araçlar" tarzında bir yapı düşünülmüştür. Bu yapı paralelinde sistem kullanıcısı tanımlanan aralıklar dışında giriş-çıkış yapan aracın verilerini inceleyerek; ilgili aracın hangi vardiyaya tabi olduğunu belirleyecek; sistemde oluşturulacak olan manuel veri giriş ekranından o aracı istenilen (daha önce tanımlanmış) vardiyaların birine ekleme-çıkarma opsiyonu bulunmaktadır.

Sistem; bir manuel kayıt giriş ekranıyla aynı zamanda; süreç döngüsü içinde yaşanabilecek olası hatalı durumlara çözüm getirebilir niteliktedir. Sisteme tanıtılmış mevcut üzerinde RFId etiket takılı olan araçlar dışında olası bir kaza, arıza vb durumlarda; taşıyıcı firma tarafından sürecin devam etmesi için ihtiyacı karşılamak üzere çevrime anlık olarak dahil edilen tanımsız araçların, giriş, çıkış, araç ve güzergah bilgilerinin girilerek; seçilecek olan ilgili vardiyanın veri bloguna kaydedilmesi sağlanmıştır.

Sorgulamadan elde edilen sonuçlar; rapor formatında tariflenen fonksiyonları yerine getirebiliyor durumdadır.

4.5.3. Raporlamalar

a. Zaman dilimlerine göre:

Sistemde, tanımlamalar kısmında dinamik olarak oluşturulan vardiya adlarına göre raporlama yapabilmektedir.

Burada istenen, ilgili vardiya tanımı, tarih ve saat aralığı sisteme belirtildiğinde; sadece belirtilen vardiyaya kayıt edilen tüm araçların, RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlamasıdır. Aynı şekilde; istenildiği takdirde bütün vardiya tanımlarına göre de yine belirtilen tarih arlıklarında raporlama yapılabilmektedir.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, Word, Pdf ve Excel olarak kayıt edilip ve yazdırılabilir durumdadır.

b. Plaka bilgilerine göre:

Sistem; belirtilen plaka bilgilerinden, belirtilen zaman ve tarih aralıklarına göre, RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlama yapabilmektedir. Mevcut plaka listesinden birden fazla plakanın aynı anda seçilip Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt edilip ve yazdırılabilir durumdadır.

c. Güzergah bilgilerine göre:

Sistem; mevcut tanımlı güzergah bilgilerine göre belirlenen tarih ve zaman aralıkları içinde ; RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlama yapabilmektedir. Mevcut güzergah listesinden aynı anda birden fazla güzergah seçilerek raporlama yapabilmektedir.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt edilip ve yazdırılabilir durumdadır.

d. Geç kalma durumuna göre:

Sistem; mevcut tanımlı araçların yukardaki bölümlerde belirtilen geç Kalma durumlarına göre, belirtilen tarih ve zaman aralıklarında, RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlama yapabilmektedir. Mevcut güzergah listesinden aynı anda birden fazla güzergah seçilerek raporlama yapabilmektedir.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt edilip ve yazdırılabilir durumdadır.

e. Söför bilgilerine göre:

Sistemde; mevcut tanımlı söför bilgilerine göre ad-soyad üzerinden belirtilen tarih ve zaman aralıkları paralelinde, RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlama yapabilmektedir. Mevcut güzergah listesinden aynı anda birden fazla güzergah seçilerek raporlama yapabilmektedir. Mevcut şöför listesinden birden fazla şöförün aynı anda seçilip raporlanabilmesi önemlidir.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, word, pdf ve excel olarak kayıt edilip ve yazdırılabilir durumdadır.

f. Ay, yıl ve gün bazında raporlama:

Sistem, belirtilen tarih aralıklarıyla; ay, yıl ve gün bazında tüm hareketleri genel raporlama yapabilmektedir. Rapor; RFID etiket kodlarını, araç plakalarını, araç tiplerini, sürücü ad-soyadlarını, vardiya tanımlarını, tarih ve giriş/çıkış zamanlarını ve adetlerini raporlama yapabilmektedir.

Rapor ekranı üzerinden veriler direk olarak mail gönderilebiliyor olup, Word, Pdf ve Excel olarak kayıt edilip ve yazdırılabilir durumdadır.

4.5.4. Otomatik Mail Desteği

Sistemde belirtilen geç kalma durumları; her tanımlanan vardiya giriş işlemi zaman aralıklarının bitiminde; sisteme belirtilen kullanıcılara ve sorumlulara otomatik olarak mail gönderebilir formattadır.

Ayrıca etiketlere tanımlı verilerde belirtilen araç muayene tarihleri; son günleriene 15 gün kala belirtilen mail adreslerine otomatik olarak gönderilmektedir. Otomatik mail işleminden beklene diğer bir yapı ise; günlük olarak her vardiya girişi sonrası; açıkta kalan listesindeki araç bilgilieri, giriş-çıkış zamanlarını yine belirtilen mail adreslerine mail atabilmesidir.

4.5.5. Merkezi Kod Yapıları

RFID etiketlere gerekli veri tanımlamak ve verilerin okuyucular aracılığıyla düzenli olarak kontrol panleline, kontrol panelinden serverlara ve en nihayetinde yazılımda ilgili arayüzlere taşımak projenin en kritik yapısıdır. Bu yazılanların bazı kritik unsurları aşağıdaki gibi anlatılmıştır.
Servis araçlarının rf etiketleriyle tanımlanması için 4 tabloya ihtiyacımız var;
Bu tablolar :
Sicil: araç bilgilerinin öz niteliklerinin bulunduğu tablo
Userlist : araçlara takılacak rf etiketlerin id numaralarının tutulduğu tablo
Firma : Sicil tablolarındaki araçların hangi firmaya ait olduklarının bilgisi
Pool : araçların giriş ve çıkış hareketlerinin anlık olarak tutulduğu tablo;

a. Sicil tablosu:

```
USE [ford]
GO
/****** Object:  Table [dbo].[Sicil]    Script Date: 12/13/2012
15:07:32 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
CREATE TABLE [dbo].[Sicil](
	[ID] [int] NOT NULL,
	[UserID] [nvarchar](8) NULL,
	[Firma] [int] NULL,
	[TerminalGrubu] [int] NULL,
	[Ad] [nvarchar](20) NOT NULL,
	[Soyad] [nvarchar](20) NOT NULL,
	[PersonelNo] [nvarchar](20) NULL,
	[GirisTarih] [smalldatetime] NULL,
	[CikisTarih] [smalldatetime] NULL,
	[SicilNo] [nvarchar](20) NULL,
	[Pozisyon] [int] NULL,
	[Bolum] [int] NULL,
	[Telefon1] [nvarchar](20) NULL,
	[Telefon2] [nvarchar](20) NULL,
	[CepTelefon] [nvarchar](20) NULL,
	[Adres] [nvarchar](100) NULL,
	[IL] [nvarchar](20) NULL,
	[Ilce] [nvarchar](20) NULL,
	[KanGrubu] [int] NULL,
	[FotoImage] [image] NULL,
	[Bilgi] [ntext] NULL,
	[MesaiPeriyodu] [int] NOT NULL,
	[PeriyodBaslangici] [smalldatetime] NULL,
	[SonDurum] [bit] NOT NULL,
	[ExpireDate] [smalldatetime] NOT NULL,
	[FazlaMesai] [bit] NOT NULL,
	[EksikMesai] [bit] NOT NULL,
	[EksikMesai_FM] [bit] NOT NULL,
	[ErkenMesai] [bit] NOT NULL,
	[EksikGun] [bit] NOT NULL,
	[MaasTipi] [int] NOT NULL,
	[Maas] [int] NOT NULL,
	[AylikCalismaSaati] [real] NOT NULL,
	[SonTasnifID] [int] NOT NULL,
	[SicilKilit] [tinyint] NULL,
	[DogumTarih] [smalldatetime] NULL,
	[OKod1] [nvarchar](50) NULL,
```

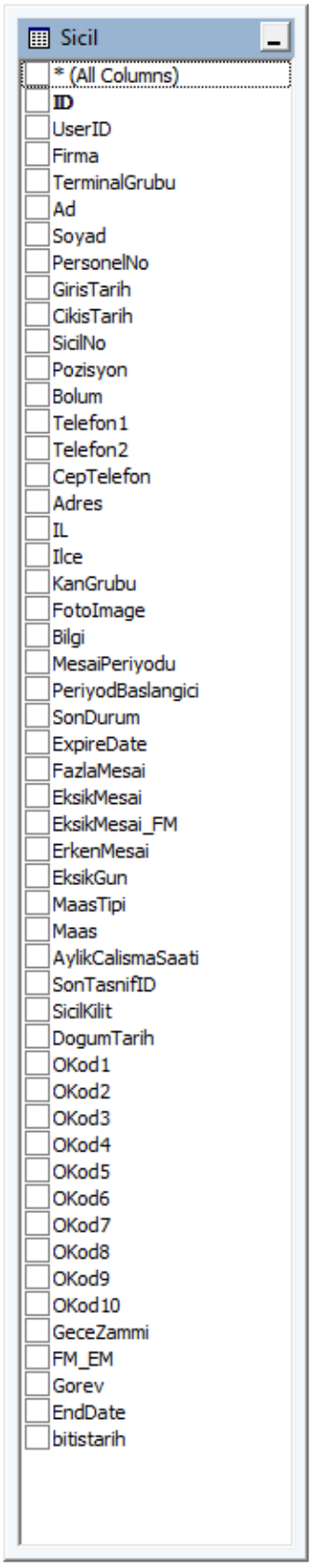
```
	[OKod2] [nvarchar](50) NULL,
	[OKod3] [nvarchar](50) NULL,
	[OKod4] [nvarchar](50) NULL,
	[OKod5] [nvarchar](50) NULL,
	[OKod6] [nvarchar](50) NULL,
	[OKod7] [nvarchar](50) NULL,
	[OKod8] [nvarchar](50) NULL,
	[OKod9] [nvarchar](50) NULL,
	[OKod10] [nvarchar](50) NULL,
	[GeceZammi] [bit] NOT NULL,
	[FM_EM] [bit] NOT NULL,
	[Gorev] [int] NULL,
	[EndDate] [datetime] NULL,
	[bitistarih] [datetime] NULL,
 CONSTRAINT [PK_Sicil] PRIMARY KEY CLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_cbo_Bolum] FOREIGN KEY([Bolum])
REFERENCES [dbo].[cbo_Bolum] ([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT [FK_Sicil_cbo_Bolum]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_cbo_Firma] FOREIGN KEY([Firma])
REFERENCES [dbo].[cbo_Firma] ([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT [FK_Sicil_cbo_Firma]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_cbo_Gorev] FOREIGN KEY([Gorev])
REFERENCES [dbo].[cbo_Gorev] ([ID])
```

```
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT [FK_Sicil_cbo_Gorev]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_cbo_Pozisyon] FOREIGN KEY([Pozisyon])
REFERENCES [dbo].[cbo_Pozisyon] ([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT [FK_Sicil_cbo_Pozisyon]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_TerminalGroup] FOREIGN KEY([TerminalGrubu])
REFERENCES [dbo].[TerminalGroup] ([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT
[FK_Sicil_TerminalGroup]
GO
ALTER TABLE [dbo].[Sicil] WITH CHECK ADD CONSTRAINT
[FK_Sicil_UserList] FOREIGN KEY([UserID])
REFERENCES [dbo].[UserList] ([UserID])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[Sicil] CHECK CONSTRAINT [FK_Sicil_UserList]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_ID] DEFAULT
((0)) FOR [ID]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_Firma]
DEFAULT ((0)) FOR [Firma]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_TerminalGrubu]
DEFAULT ((1)) FOR [TerminalGrubu]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_Pozisyon]
DEFAULT ((0)) FOR [Pozisyon]
GO
```

```
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_Bolum]
DEFAULT ((0)) FOR [Bolum]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_KanGrubu]
DEFAULT ((0)) FOR [KanGrubu]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_MesaiPeriyodu]
DEFAULT ((1)) FOR [MesaiPeriyodu]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT
[DF_Sicil_PeriyodBaslangici] DEFAULT ('2000-01-03') FOR
[PeriyodBaslangici]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_SonDurum]
DEFAULT ((0)) FOR [SonDurum]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_ExpireDate]
DEFAULT ('2000-01-03') FOR [ExpireDate]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_FazlaMesai]
DEFAULT ((0)) FOR [FazlaMesai]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_EksikMesai]
DEFAULT ((0)) FOR [EksikMesai]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_EksikMesai_FM]
DEFAULT ((0)) FOR [EksikMesai_FM]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_ErkenMesai]
DEFAULT ((0)) FOR [ErkenMesai]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_EksikGun]
DEFAULT ((0)) FOR [EksikGun]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_MaasTipi]
DEFAULT ((1)) FOR [MaasTipi]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_Maas] DEFAULT
((0)) FOR [Maas]
```

```
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT
[DF_Sicil_AylikCalismaSaati] DEFAULT ((225)) FOR
[AylikCalismaSaati]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_SonTasnifID]
DEFAULT ((0)) FOR [SonTasnifID]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_sicil_sicilkilit]
DEFAULT ((0)) FOR [SicilKilit]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT
[DF__Sicil__GeceZammi__324172E1] DEFAULT ((0)) FOR [GeceZammi]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT
[DF__Sicil__FM_EM__3335971A] DEFAULT ((0)) FOR [FM_EM]
GO
ALTER TABLE [dbo].[Sicil] ADD CONSTRAINT [DF_Sicil_Gorev]
DEFAULT ((0)) FOR [Gorev]
GO
```

Sicil tablosunun görüntüsü aşağıdaki gibidir:


Sicil
* (All Columns)
ID
UserID
Firma
TerminalGrubu
Ad
Soyad
PersonelNo
GirisTarih
CikisTarih
SicilNo
Pozisyon
Bolum
Telefon1
Telefon2
CepTelefon
Adres
IL
Ilce
KanGrubu
FotoImage
Bilgi
MesaiPeriyodu
PeriyodBaslangici
SonDurum
ExpireDate
FazlaMesai
EksikMesai
EksikMesai_FM
ErkenMesai
EksikGun
MaasTipi
Maas
AylikCalismaSaati
SonTasnifID
SicilKilit
DogumTarih
OKod1
OKod2
OKod3
OKod4
OKod5
OKod6
OKod7
OKod8
OKod9
OKod10
GeceZammi
FM_EM
Gorev
EndDate
bitistarih

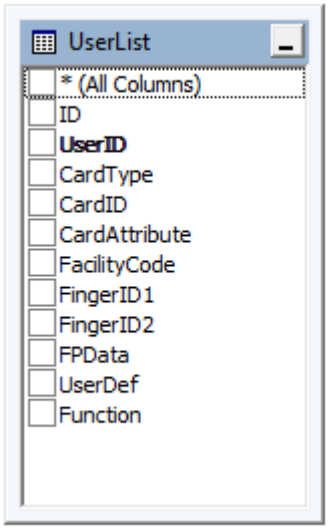
b. UserList tablosu:

RF etiketlerin kayıt edildiği ve daha sonra sicil tablosu ile ilişkilendirileceği tablodur.

```
USE [ford]
GO
/****** Object:  Table [dbo].[UserList]    Script Date: 12/13/2012
15:31:10 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UserList](
	[ID] [int] NOT NULL,
	[UserID] [nvarchar](8) NOT NULL,
	[CardType] [int] NOT NULL,
	[CardID] [nvarchar](15) NULL,
	[CardAttribute] [int] NOT NULL,
	[FacilityCode] [nvarchar](6) NULL,
	[FingerID1] [nvarchar](4) NULL,
	[FingerID2] [nvarchar](4) NULL,
	[FPData] [ntext] NULL,
	[UserDef] [int] NOT NULL,
	[Function] [int] NULL,
 CONSTRAINT [PK_UserList] PRIMARY KEY CLUSTERED
(
	[UserID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY],
 CONSTRAINT [IX_UserList] UNIQUE NONCLUSTERED
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

```
ALTER TABLE[dbo].[UserList]ADDCONSTRAINT[DF_UserList_ID]   DEFAULT
((0)) FOR [ID]
GO
ALTER TABLE [dbo].[UserList]ADD CONSTRAINT [DF_UserList_CardType]
DEFAULT ((0)) FOR [CardType]
GO
ALTER TABLE[dbo].[UserList]ADDCONSTRAINT F_UserList_CardAttribute]
DEFAULT ((0)) FOR [CardAttribute]
GO
ALTER TABLE [dbo].[UserList] ADD  CONSTRAINT [DF_UserList_UserDef]
DEFAULT ((1)) FOR [UserDef]
GO
```

UserList tablosunun görünümü aşağıdaki gibidir:

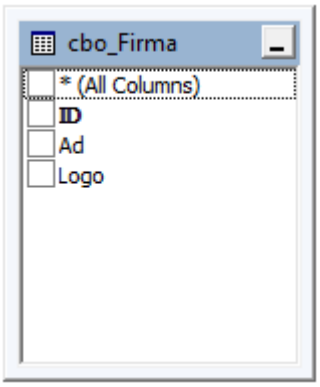


Firmalar tablosu : sicillerin hangi firmalara ait olduğu bilgisini verir

```
USE [ford]
GO
/******  Object:    Table  [dbo].[cbo_Firma]        Script  Date:
12/13/2012 15:34:44 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[cbo_Firma](
      [ID] [int] NOT NULL,
      [Ad] [nvarchar](50) NOT NULL,
      [Logo] [image] NULL,
 CONSTRAINT [PK_cbo_Firma] PRIMARY KEY CLUSTERED
```

```
(
	[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[cbo_Firma] ADD CONSTRAINT [DF_cbo_Firma_ID]
DEFAULT ((0)) FOR [ID]
GO
```

Firmalar tablosunun görünümü :



c. Pool tablosu:

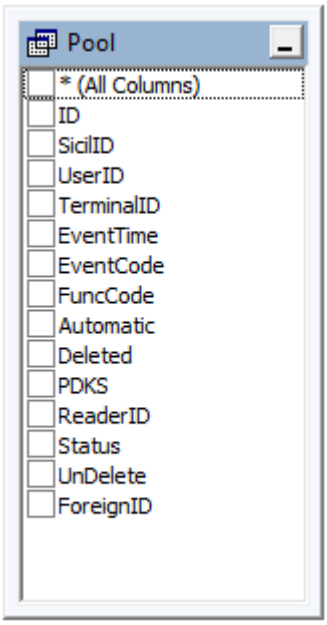
Bu tablonun işlevi antenlerden gelen verileri tutmaktır. Kart idleri burada tutulur.

```
USE [ford]
GO
/****** Object:  Table [dbo].[Pool]    Script Date: 12/13/2012
15:36:34 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Pool](
	[ID] [int] IDENTITY(1,1) NOT NULL,
	[SicilID] [int] NOT NULL,
	[UserID] [nvarchar](16) NOT NULL,
	[TerminalID] [int] NOT NULL,
	[EventTime] [datetime] NOT NULL,
```

```
	[EventCode] [tinyint] NOT NULL,
	[FuncCode] [int] NOT NULL,
	[Automatic] [bit] NOT NULL,
	[Deleted] [int] NOT NULL,
	[PDKS] [int] NOT NULL,
	[ReaderID] [int] NOT NULL,
	[Status] [text] NULL,
	[UnDelete] [bit] NOT NULL,
	[ForeignID] [bigint] NULL,
 CONSTRAINT [PK_Pool] PRIMARY KEY CLUSTERED
(
	[SicilID] ASC,
	[UserID] ASC,
	[TerminalID] ASC,
	[EventTime] ASC,
	[Deleted] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_SicilID]
DEFAULT ((0)) FOR [SicilID]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_TerminalID]
DEFAULT ((0)) FOR [TerminalID]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_EventCode]
DEFAULT ((0)) FOR [EventCode]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_FuncCode]
DEFAULT ((0)) FOR [FuncCode]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_Automatic]
DEFAULT ((-1)) FOR [Automatic]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_Deleted]
DEFAULT ((0)) FOR [Deleted]
GO
```

```
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT [DF_Pool_PDKS] DEFAULT
((0)) FOR [PDKS]
GO
ALTER TABLE [dbo].[Pool] ADD CONSTRAINT
[DF__Pool__ReaderID__3F466844] DEFAULT ((0)) FOR [ReaderID]
GO
ALTER TABLE [dbo].[Pool] ADD DEFAULT ('1') FOR [UnDelete]
GO
ALTER TABLE [dbo].[Pool] ADD DEFAULT ((0)) FOR [ForeignID]
GO
```

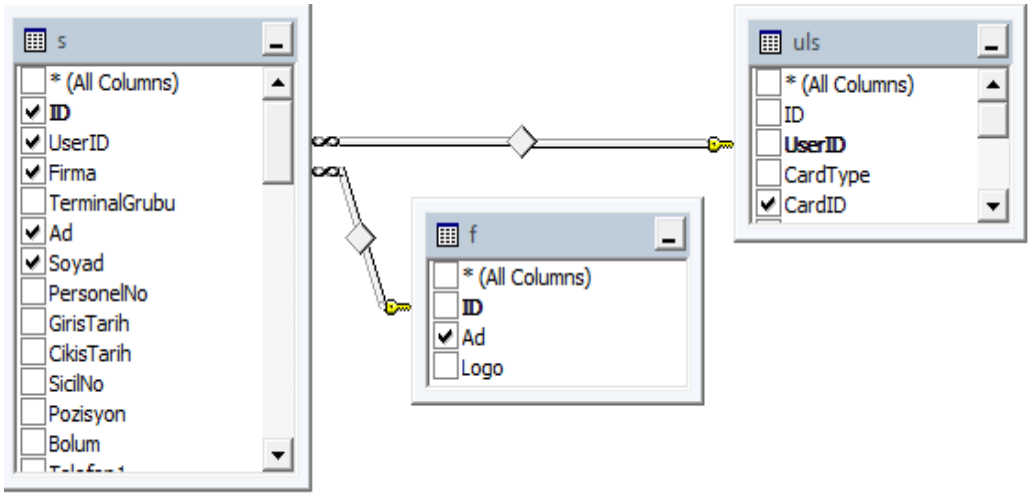
Pool tablosunun görüntüsü aşağıdaki gibidir:



d. Tabloların ilişkilendirilmeleri:

Hangi sicilin hangi firmada olduğu ve hangi rf karta sahip olduğunu aşağıdaki gibi sql script ile öğrenebiliriz;

```
SELECT s.ID, s.UserID, s.Firma, s.Ad, s.Soyad, f.Ad AS firmaadi,
uls.CardID FROM dbo.Sicil AS s INNER JOIN dbo.cbo_Firma AS f ON
f.ID = s.Firma INNER JOIN dbo.UserList AS uls ON uls.UserID =
s.UserID
```



Sorguyu çalıştırınca aşağıdaki gibi sonuç alırız.

```
select s.ID,s.UserID,s.Firma,s.Ad,s.Soyad,f.Ad firmaadi,uls.CardID
 inner join cbo_firma f on f.ID=s.firma
 inner join userlist uls on uls.UserID=s.UserID
```

Results | Messages

	ID	UserID	Firma	Ad	Soyad	firmaadi	CardID
1	1	00000001	1	Amavutköy	Esenyurt	Ford	000000000000000
2	2	00000002	1	Şile	Sultangazi	Ford	000000000000000
3	3	00000003	1	Beşiktaş	Gaziosmanpaşa	Ford	000000000018626
4	4	00000004	1	Avcılar	Ümraniye	Ford	000000000018606
5	5	00000005	1	Esenler	Kartal	Ford	000000000018607
6	6	00000006	1	Silivri	Beyoğlu	Ford	000000000018631
7	7	00000007	1	Kartal	Sancaktepe	Ford	000000000018605
8	406	00000007	0	Sancaktepe	Kağıthane	----------	000000000018605
9	43	00000043	1	Üsküdar	Esenler	Ford	000000000000042
10	44	00000044	1	Sultanbeyli	Sultanbeyli	Ford	000000000018603
11	45	00000045	1	Başakşehir	Gaziosmanpaşa	Ford	000000000000044
12	46	00000046	1	Gaziosmanpaşa	Bayrampaşa	Ford	000000000018630
13	47	00000047	1	Çatalca	Kağıthane	Ford	000000000018602
14	48	00000048	1	Sancaktepe	Beyoğlu	Ford	000000000018610
15	49	00000049	1	Güngören	Pendik	Ford	000000000018611
16	50	00000050	1	Tuzla	Çekmeköy	Ford	000000000018609
17	51	00000051	1	Beylikdüzü	Bağcılar	Ford	000000000018608
18	52	00000052	1	Bahçelievler	Fatih	Ford	000000000018614

Pool tablosu tek başına hareketlerin kaydedildiği tablodur. Select * from pool gibi bir sorguda aşağıdaki gibi bir sonuç döndürür:

```
select * from pool
```

Results | Messages

	ID	SicilID	UserID	TerminalID	EventTime	EventCode	FuncCode	Automatic	Deleted	PDKS	ReaderID	Status	UnDelete	ForeignID
1	281934	0	00000368	2	2012-08-24 12:24:35.000	0	255	1	0	0	0	NULL	1	0
2	283163	0	00000368	2	2012-08-28 11:41:58.000	0	255	1	0	0	0	NULL	1	0
3	284872	0	00000368	2	2012-08-31 09:55:00.000	0	255	1	0	0	0	NULL	1	0
4	284965	0	00000368	2	2012-08-31 14:20:53.000	0	255	1	0	0	0	NULL	1	0
5	285852	0	00000368	2	2012-09-03 15:20:16.000	0	255	1	0	0	0	NULL	1	0
6	286501	0	00000368	2	2012-09-04 16:33:41.000	0	255	1	0	0	0	NULL	1	0
7	297095	0	00000368	2	2012-09-12 09:31:42.000	0	255	1	0	0	0	NULL	1	0
8	305503	0	00000368	2	2012-09-21 16:12:27.000	0	255	1	0	0	0	NULL	1	0
9	306618	0	00000368	2	2012-09-24 10:33:59.000	0	255	1	0	0	0	NULL	1	0
10	315840	0	00000368	2	2012-10-03 13:20:28.000	0	255	1	0	0	0	NULL	1	0
11	317919	0	00000368	2	2012-10-04 08:03:20.000	0	255	1	0	0	0	NULL	1	0
12	322548	0	00000368	2	2012-10-05 09:47:14.000	0	255	1	0	0	0	NULL	1	0
13	325795	0	00000368	2	2012-10-08 13:41:40.000	0	255	1	0	0	0	NULL	1	0
14	340779	0	00000368	2	2012-10-17 11:55:17.000	0	255	1	0	0	0	NULL	1	0
15	342041	0	00000368	2	2012-10-19 09:35:29.000	0	255	1	0	0	0	NULL	1	0
16	343439	0	00000368	2	2012-10-22 17:05:10.000	0	255	1	0	0	0	NULL	1	0
17	344997	0	00000368	2	2012-10-23 11:17:07.000	0	255	1	0	0	0	NULL	1	0
18	345003	0	00000368	2	2012-10-23 11:27:45.000	0	255	1	0	0	0	NULL	1	0
19	979724	0	00000368	2	2012-11-01 16:41:23.000	0	255	1	0	0	0	NULL	1	0
20	1228...	0	00000368	2	2012-11-05 12:22:59.000	0	255	1	0	0	0	NULL	1	0
21	1238...	0	00000368	2	2012-11-14 12:26:23.000	0	255	1	0	0	0	NULL	1	0
22	1247...	0	00000368	2	2012-11-22 10:57:44.000	0	255	1	0	0	0	NULL	1	0
23	1248...	0	00000368	2	2012-11-23 08:35:13.000	0	255	1	0	0	0	NULL	1	0
24	281932	0	00000368	4	2012-08-24 12:24:22.000	0	255	1	0	0	0	NULL	1	0

Burada önceki ilişkili tablolarla pool tablosunu ilişkilendirirsek hangi servisin hangi saatlerde hareket ettiğini bulabiliriz. Burada bilmemiz gereken şey pool tablosundaki PDKS alanındaki 1 değerinin giriş 2 değerinin de çıkış olduğudur. Buna göre şu sorguyu hazırlayabiliriz.

```
select
s.ID,
s.UserID,
s.Ad as nereden,
s.Soyad as nereye,
f.Ad hatsahibi,
uls.CardID,
p.EventTime,
case when p.PDKS=1 then 'giriş' else 'çıkış' end hareket_yonu from pool p
inner join sicil s on s.UserID=p.UserID
inner join cbo_firma f on f.ID=s.firma
inner join userlist uls on uls.UserID=s.UserID
order by p.eventtime
```

Results | Messages

	ID	UserID	nereden	nereye	hatsahibi	CardID	EventTime	hareket_yonu
18123	90	00000090	Pendik	Çekmeköy	Ford	000000000018665	2012-05-14 09:17:49.000	giriş
18124	128	00000128	Gaziosma...	Çekmeköy	Ford	000000000044888	2012-05-14 09:33:09.000	giriş
18125	403	00000128	Başakşehir	Sultanbeyli	----------	000000000044888	2012-05-14 09:33:09.000	giriş
18126	287	00000287	Bahçelie...	Avcılar	Ford	000000000000286	2012-05-14 09:44:10.000	çıkış
18127	1	00000001	Amavutköy	Esenyurt	Ford	000000000000000	2012-05-14 10:17:41.000	giriş
18128	1	00000001	Amavutköy	Esenyurt	Ford	000000000000000	2012-05-14 10:17:43.000	giriş
18129	221	00000221	Amavutköy	Sultangazi	Ford	000000000018711	2012-05-14 10:22:49.000	giriş
18130	221	00000221	Amavutköy	Sultangazi	Ford	000000000018711	2012-05-14 10:23:06.000	giriş
18131	261	00000261	Amavutköy	Ümraniye	Ford	000000000018729	2012-05-14 10:28:30.000	giriş
18132	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:31:28.000	çıkış
18133	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:31:49.000	çıkış
18134	82	00000082	Esenler	Pendik	Ford	000000000018681	2012-05-14 10:36:10.000	giriş
18135	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:37:25.000	giriş

e. Bilgilerin C# ile web sayfasında gösterilmesi:

Bunun için C# içinde yeni bir web aplication projesi oluşturulur. Form1 'in using satırına using System.Data.SqlClient; satırı eklenerek sql bağlantı özelliklerinin projeye eklenmesi sağlanır. Örnek kod ve sonucu aşağıdaki gibidir:

Default.aspx içeriği :

```
<%@PageLanguage="C#"utoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.ford.com.tr/2012/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
         <asp:GridView ID="GridView1" runat="server" Width="100%">
        </asp:GridView>
       </div>
    </form>
</body>
</html>
```

Default.aspx sayfasının c# kodu ise;

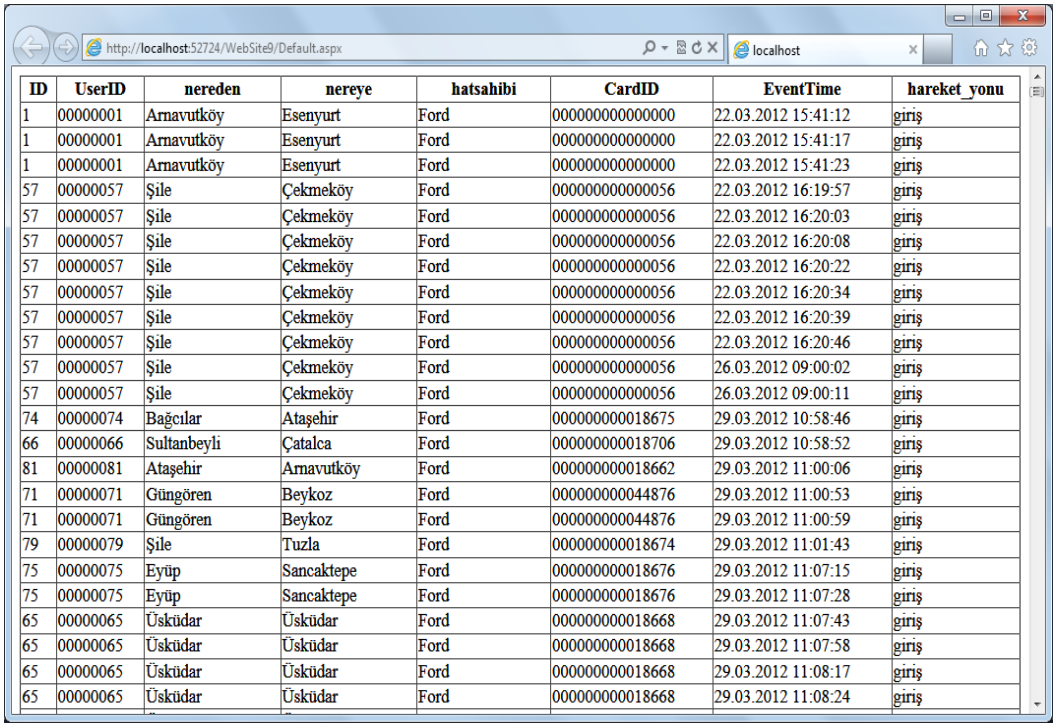
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
```

```
        bool sonuc = false;
        string soru = @"select
s.ID,
s.UserID,
s.Ad as nereden,
s.Soyad as nereye,
f.Ad hatsahibi,
uls.CardID,
p.EventTime,
case when p.PDKS=1 then 'giriş' else 'çıkış' end hareket_yonu from
pool p
inner join sicil s on s.UserID=p.UserID
inner join cbo_firma f on f.ID=s.firma
inner join userlist uls on uls.UserID=s.UserID
order by p.eventtime
";
            SqlConnection con = new SqlConnection();
        con.ConnectionString                =                "Data
Source=mustafa_lp\\sqlexpress;Initial   Catalog=fordotosan;Persist
Security Info=True;User ID=sa;Password=ford123";

        SqlCommand kontrol = new SqlCommand();
        kontrol.CommandText = soru;
        kontrol.CommandType = CommandType.Text;
        kontrol.Connection = con;
        con.Open();
        SqlDataReader dread = kontrol.ExecuteReader();
        GridView1.DataSource = dread;
        GridView1.DataBind();
        con.Close();
    }
}
```

Proje çalıştırıldığında açılan web sayfasında görüldüğü gibi sql sorgusu web sayfası üzerine aşağıdaki gibi yansımaktadır.



ID	UserID	nereden	nereye	hatsahibi	CardID	EventTime	hareket_yonu
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:12	giriş
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:17	giriş
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:23	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:19:57	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:03	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:08	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:22	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:34	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:39	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:46	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	26.03.2012 09:00:02	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	26.03.2012 09:00:11	giriş
74	00000074	Bağcılar	Ataşehir	Ford	000000000018675	29.03.2012 10:58:46	giriş
66	00000066	Sultanbeyli	Çatalca	Ford	000000000018706	29.03.2012 10:58:52	giriş
81	00000081	Ataşehir	Arnavutköy	Ford	000000000018662	29.03.2012 11:00:06	giriş
71	00000071	Güngören	Beykoz	Ford	000000000044876	29.03.2012 11:00:53	giriş
71	00000071	Güngören	Beykoz	Ford	000000000044876	29.03.2012 11:00:59	giriş
79	00000079	Şile	Tuzla	Ford	000000000018674	29.03.2012 11:01:43	giriş
75	00000075	Eyüp	Sancaktepe	Ford	000000000018676	29.03.2012 11:07:15	giriş
75	00000075	Eyüp	Sancaktepe	Ford	000000000018676	29.03.2012 11:07:28	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:07:43	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:07:58	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:08:17	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:08:24	giriş

4.5.6 Farklılık Analizi

Yeni sürecin eski süreç için belirtilmiş olan "olmazsa olmazlar"ını karşılayıp karşılamadığı, madde aşağıdaki bölümlerde belirtilen problem ve verimsizliklerin giderilip giderilemediği, varsa yeni süreçte ek bir problem ya da verimsizlik olup olmadığının analizi aşağıdaki şekiledir.

Mecvut personel taşımacılık takibinde yaşanan problemler ve sistemin getirileri aşağıdaki gibi özetlenmiştir;

- Servis giriş-çıkışlarında plaka bilgilerinin elle tutulması,

Önerilen Sistemde plaka bilgileri araçlara takılması planlanan RFID etiketlerinin üzerine tanımlanacağından elle herhangi bir kayıt tutulmasına gerek kalmamaktadır

- Personelin dikkat etmediği veya araçların bir anda yoğunlaştığı durumlarda araç kayıtları eksik yapılması,

RFID Okuma antenleri; 0-45 metre aralığından önünden geçen bütün tanımlı aktif etiketleri okumaktadır. Planalanan sistemde Okuma antenlerinden maksimum iki metre aralıklı araç geçişleri sağlanmıştır. Bu da yanyana geçişlerde bile bütün araçların eksiksiz olarak kayıt altına alınmasını sağlamış, kullanılan aktif etiketler antenlerin maksimum mesafede kayıt tutmasını sağlamıştır.

- Farklı güzergahlardan gelen araçların kayıtlarının yanlış tutulması,

  Önerilen sistemde, Aktif RFID etiketlerin üzerine araç bilgilerinin yanında araç güzergahlarıda tanımlıdır. Bu yüzden etiketin anten tarafından okunduğu her durumda güzergah bilgileride doğrulanmış olarak kayıt altına alınacaktır.
- Araç geliş zamanlarının hatalı belirlenmesi, (Zaman zaman otobüs söförlerinin alana girdiklerini söyledikleri zaman dilimi ile ielgili personelin kayıtları arasıda farklılıklar oluşmakta, bu farklılıklardan kaynaklanan problemler oluşmaktadır)

Önerilen sistemde; araç geliş gidiş zamnaları bilgisayar kontrollü tur kontrol yazılımı tarafından kayıt altına alınacağı için, zaman kayıtları tek merkezli olup ve kayıtlar arasındaki farklılaşmanın önüne geçilmiştir.

- İşlemin hakediş kısmında, güvenlik ve sosyal hizmetler arasında sıkıntılar yaşanması, Önerilen sistemde; kayıtlar dijital olarak tutulacağından dolayı güvenlik birimi artık bu süreç dahilinde olmayaktır. Süreç Artık tamamen tek departman tarafından kontrol edildiği için olaşı sorunların önüne geçilerek
- İç denetim ve holding denetimlerinde konu üzerinden problemler yaşanması, Verilerin manuel yöntemlerle dosyalarda depolanması yerine veritabınında tutulması, geriye dönük veri kayıplarının önüne geçecektir.
- Kayıt işlemini yapan personelin ilgili birimin vardiya yapılarına ve personel akışlarına göre sürekli değişmesi, işin yapılışında profesyonelleşmeyi önleyerek, olası hataların oluşmasına zemin hazırlamaktadır.

Önerilen sistemle beraber kayıt işlemleri için personele ihtiyaç kalmamıştır. Buda personel kaynaklı sorunların önüne geçmektedir.

- Geç kalma durumlarına itirazların yaşanması, işi takip eden personel ve taşıyıcı firmanın standart bir zaman üzerinden işlem yapamaması.

Önerilen sistemde; araç geliş gidiş zamnaları bilgisayar kontrollü tur kontrol yazılımı tarafından kayıt altına alınacağı için, zaman kayıtları tek merkezli olacak ve kayıtlar arasındaki farklılaşmanın önüne geçmiştir.

- En kritik unsurlardan biride, sistemin su anda tamamen insan eliyle yürütülerek, hataya açık olmasıdır. (Önerilen sistemde; en çok karşılaşılan hata tiplerine sebep olan süreçlerde insan unsuru ortadan kaldırılacaktır)

5. SONUÇLAR

Çok sayıda personelin çalıştığı büyük ölçekli üretim tesislerinde, personel taşımacılığı hizmetleri, iç müşteri memnuniyetini etkileyen unsurlardan olup, takibini, kalite standartlarında işleyişini sağlamak personel sayısıyla ve vardiya sistemiyle paralel olarak artan bir maliyet grafiğine sahiptir. Etkin takip ve kaliteli süreç yönetimi için süreç işleyişini belli ölçülerde otomasyona almak günümüzde biröok sektörde uygulanan bir yöntemdir.

Bu bağlamda; RFID teknolojisinin farklı sektörlerde kullanımının hızla artması, özellikle iş süreçlerinin takibini kolaylaştırarak, işletim maliyetlerini oldukça büyük oranda azaltmıştır. Otomatik kimlik tanıma ve veri toplama sistemlerinden olan RFID teknolojisi, giderek süreç otomasyonunda önemli bir yapı haline gelmektedir.

Bu tez çalışmasında bünyesinde 10000'e yakın personeli barındıran, farklı şehirlerde değişik loksayonları bulunan bir üretim tesisinde, personel taşımacılığı operasyonlarının, kayıt altına alınması, kontrolü, hak ediş süreçlerinde kalite iyileştirmesi ve merkezi bir yapıya geçilmesi için çözüm aranmıştır ve hem işletim zamanlarında hem de maliyet unsurlarında önemli kazanımlar sağlanmıştır.

Materyal olarak kullanılan RFID teknolojisinin unsurları; RFId okuyucu antenler, aktif RFID etiketler, bilgisayarlar, kontrol paneli ve bu sistemlerin yönetilmesini sağlayan bir web tabanlı yazılımdan oluşmaktadır. Web tabanlı yazılım ile her lokasyonun servis süreçleriyle ilgili olarak ihtiyaç duyduğu verileri merkezi bir veri tabanında geriye dönük 5 yıllık olarak depolaması ve süreci yönetmesi sağlanmıştır.

Bahsi geçen fabrikalardan 1500 personel için servis hakediş miktarları düşünüldüğünde yılda 5.000.000-7.000.000 TL arası bir operasyon söz konusu olup, sadece tek kapı üzerinden araçların fabrikaya geliş zamanlarının belirlenmesi için çeşitli vardiyalarda toplamda 405 dakika işgücü harcanmaktadır. Belirlenen bu sürelerin düzenlenip istenilen formatta ilgili birirme rapolanması için ise gün içerisinde çeşitli vardiyalarda toplamda ortalama 130 dakika işgücü harcanmaktadır. Bu çalışma paralelinde bu işgücü

kaybının önüne gecilerek gün içerisinde toplamda ortalama 435 dakikalık operasyon dakikalar içinde çözüme kavuşturulmuştur.

Sürecin raporlanması ve belirlenmesi, rapor üzerinden belirlenen veriler ile taşıyıcı firmanın verileri üzerinden mutabakat sağlanıp hakediş işlemlerinni gerçekleştirilmesi ve sürecin akışının sağlanması; araç geliş zamanları ve güzergah bilgilerinin toplanmasıyla başlayan ve bu bilgilerin kilometre bazlı değerleri üzerinden hakedişlerinin hesaplanıp, ödeme ve yasal gerekliliklerin sonlandırılmasıyla biten süreçte en temel darbogaz noktasıdır. Bu yüzden verilerin tek merkezli olarak toplanıp, hem taşıyıcı firma hem de süreç sahibi fabrika tarafından ortak bir yazılım paralelinde tek merkezli kontrol edilmesi bahsedilen bu darbogazı ortadan kaldırarak, süreci hızlandırmaktadır.

Ayrıca, fabrikaya işgücünün getirilmesi, bütün süreçlerin başlamasını sağlayan birincil aşama olduğu için; personelin servis araçlarıyla geliş gidiş zamanları, başta üretim ve sendikal süreçler olmak üzere birçok yapıyı birincil derecede ilgilendirmektedir. Taşıyıcı firmadan kaynaklı iş aksamalarında (personelin fabrikaya belirlenen zamanlarda getirilememe durumu), iş bedeli firmaya sözleşme geregi ceza olarak yansıtılır. Bu sistem bir yönüyle takip olayını otomatikleştirdiği için takip verilerinde ve cezai işlemlerde mutabakatın sağlanmasına olanak sağlamaktadır. Ayrıca, personelden kaynaklı gecikmelerin taşıyıcı firmaya yansımasına da engel olmaktadır. Sistem bir yönüyle personel hareketlerinide denetlediği için, işletme tahamüllerine uygunluk konusunda işçilerle dolayısıylada sendikalarla oluşabilecek olası sorunlarında önüne geçmektedir.

Servis takip için çalıştırılan personelin 435 dakikalık işgücünün fabrikaya maliyeti yan haklarıyla (yemek, ulaşım vb) beraber aylık 1580$ olarak belirtilmiştir. Bu değerde bir kapı için yılda 1580$ x 12 üzerinden 18.960 $ tutarındadır. Bu projenin 4 farklı lokasyonda 10 kapıda uygulandığı düşünüldüğünde yaklaşık yıllık maliyet; 18.960 $ x 10 üzerinden 189.600 $ olarak hesaplanmaktadır.

Kalite olarak sürecin sigma seviyesinde önemli kazanımlar sağlanmakla birlikte, manuel veri yönetiminde iç denetim süreçlerinde yaşanan olası sorunlarında, verileri merkezi bir dijital veritabanında saklanmasıyla önüne

geçilmiştir.

Sistemin toplam maliyeti 1 giriş-çıkış noktası için donanım olarak 4600€ , network ve elektrik kablolama olarak 100 metre başına 2200 TL ve sunucu masrafları için ise yaklaşık olarak 4200 $ cıvarında seyretmiştir. Yazılım geliştirme ve devreye alma iç süreçler olduğu için ekstra bir maliyet unsuru oluşmamıştır. Cocoma II yöntemine göre, Çizelge 12'deki başlıklar bazında hesaplanan yazılım maliyeti 30. 000 $ civarındadır. (KDV hariç)

Çizelge 5.1. Önerilen Ölçev Seti [36]

Ana Ölçev Ölçev (Metric)
A. İşin Büyüklüğü(Ürün)
1. Karmaşıklık
2.İşlev Puan (Function Point)
3.Önem
4. Ayrılan Bütçe
5. Ürünün beklenen özellikleri
B. Kaynak
6. Çalışanın yeterlilikleri
7 Çalışanların Projeye katılım oranları
8. Çalışan Kişi Sayıları
9. Donanım Durumu
C. Risk
10.Bütçe Değişme Riski
11. Çalışan Riski
12.Donanım Riski
13.Ürünün tanımının ve kapsamının değişme riski
D. Teknoloji
14.Yazılım Geliştirme araçlarının kullanım kolaylığı
15.Yazılım Geliştirme araçlarının kullanım tecrübesi
16. Yazılım Geliştirme Araçlarının Kullanımı
17. Modern Programlama Teknikleri
F.Ortam
18.Ortamın genel özellikleri
19.Sahiplenilme (Her bir payda_ türünün projeyi sahiplenmesi)
20.Baskı
21.Zaman kullanım durumu
22.Verimlilik durumu
F.Planlar ve Tahminler
23. Tahmin
24. Planlar

Ayrıca bu çalışma büyük ölçekli verilerin, RFID teknolojisi kullanılarak web tabanlı yönetilmesine bir örnek uygulama niteliğindedir.

KAYNAKLAR

[1] Manish, B. ve Shahram, M., *RFID Field Guide: Deploying Radio Frequency Identification Systems*, Prentice Hall PTR Upper Saddle River-NJ, A.B.D, 2005.

[2] Klaus, F. ve Dorte, M., *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, Wiley, *A.B.D.*, 2010.

[3] Anonim, *What is RFID?*, RFID Journal, 2005. http://www.rfidjournal.com/article/articleview/1339/1/129/

[4] Heinrich, C., *RFID and Beyond : Growing your business through real world awareness*, Wiley-Indianapolis, A.B.D, 2005.

[5] Malkoç, E., *Depo yönetim sistemlerinde kullanılan otomatik tanıma ve veri toplama teknolojileri ile rfid etiketleme*, Yüksek Lisans Tezi, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 2006.

[6] Roberts, C.M., "Radio frequency identification (RFID)," *Computers & Security*, **25**, 18-26, 2006.

[7] Shepard, S., *RFID: Radio Frequency Identification*, McGraw-Hill, New York, A.B.D, 2005.

[8] Mark Roberti, *The History of RFID Technology*, RFID Journal. http://www.rfidjournal.com/article/articleview/1338/1/129/

[9] Anonim, *The Basics of RFID Technology*, RFID Journal. http://www.rfidjournal.com/article/articleview/1337/1/129/

[10] Anonim, *RFID System Components and Costs*, RFID Journal. http://www.rfidjournal.com/article/articleview/1336/1/129/

[11] Anonim, *A Summary of RFID Standarts*, RFID Journal. http://www.rfidjournal.com/article/articleview/1335/1/129/

[12] Wu, N.C., Nystom, M.A. ve Yu, H.C., "Challenges to global RFID Adoption," *Technovation*, **26**, 1317-1323, 2006.

[13] Shepard, S., *RFID : The Promise of a Strategic Technology*, 2004. http://www.shepardcomm.com/RFID-whitepaper-wp.pdf

[14] Börklü, H.R., "Makine tasarım dili," *Mesleki ve Teknik Eğitim Sempozyumu*, Elazığ, 1995.

[15] Pahl, G. ve Beitz, W., *Engineering Design*, The Design Council, Springer- Londra, İngiltere, 1988.

[16] Börklü, H.R., "Computer-aided conceptual design based on design catalogues", *Politeknik Dergisi*, **4**, 77-78, 2001.

[17] Sağır, H.B., *Bilgisayar Destekli Kavramsal Tasarım*, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 1996.

[18] Deng, Y.M., Tor, S.B. ve Britton, G.A., "Abstracting and exploring functional design information for conceptual mechanical product," *Engineering with Computers*, **16**, 36-52, 2000.

[19] Kühnapfel, B., "Simulation-based evaluation in conceptual design," *International Conference on Engineering Design* , ICED 97, Tampere, Finland, 133-136, 1997.

[20] Gao, X. ve Li, Z., "Computer-aided conceptual design of mechanical products using polychromatic sets," *Int. Conf. on Mechatronics and Automation, China*, 1169-1174, 2006.

[21] Xu, L., Li, Z., Li, S. ve Tang, F., "A Polychromatic sets approach to the conceptual design of machine tools," *International Journal of Production Research*, **43**, 2397-2421, 2005.

[22] Guan, L., Wang, J. ve Wang, L., "Integrated approach for parallel machine tool conceptual design," I*nt. Conf. on Robotic, Intelligence Systems and Signal Processing*, China, 456–461, 2003.

[23] Ociepka, P. ve Swider, J., "Object-oriented system for computer aiding of the machines conceptual design process," *Journal Materials of Processing Technology*, **157**, 221-227, 2004.

[24] Pahl, G. ve Beitz, W., "Engineering Design," *The Design Council*, Springer- Veriag, London, 10-160, 1996.

[25] Kumar, A.S., Subramaniam, V. ve Teck, T.B., "Conceptual design of fixtures using machine learning techniques," *International Journal of Advanced Manufacturing Technology*, **16**, 176-181, 2000.

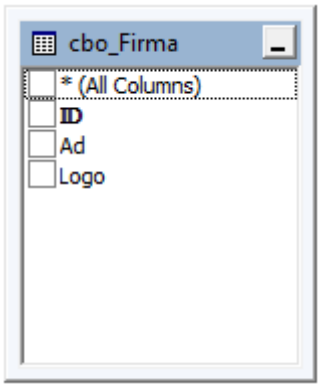
[26] Ira Campbell, M., *The a design invention machine: A means of automating and investigation conceptual design*, Doktora Tezi, Carneige Mellon University, A.B.D, 2000.

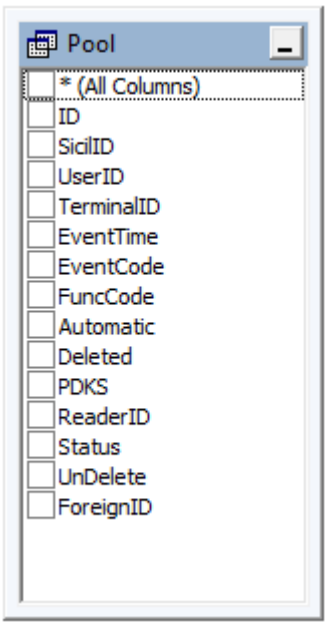
[27] Samson, B., Ellison, D. ve Dugard , P., "Software Cost Estimation using an Albus Perceptron(CMAC)," *Information and Software Technology*, **39**, 55-60, 1997.

[28] Burcu T. ve Aktül K., "Radyo Frekans Kimlik Tanıma Sistemleri ile Elektronik Para Uygulamasının Gerçeklenmesi," *III. İletişim Teknolojileri Ulusal Sempozyumu*, Çukurova Üniversitesi, Balcalı, Adana, 2007.

[29] Klaus, F., *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Wiley, *A.B.D.*, 2003.

EK-1 Merkezi SQL Yapıları





```
select * from pool
```

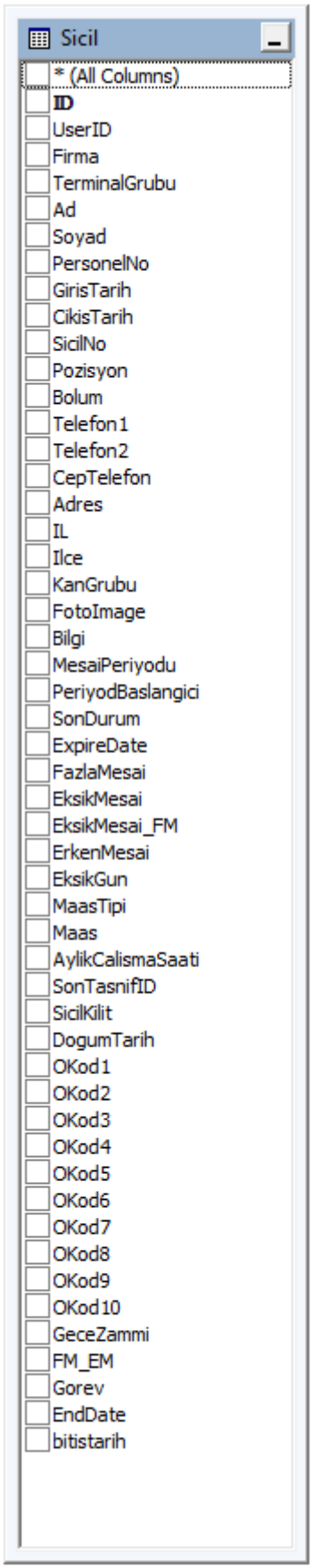
Results | Messages

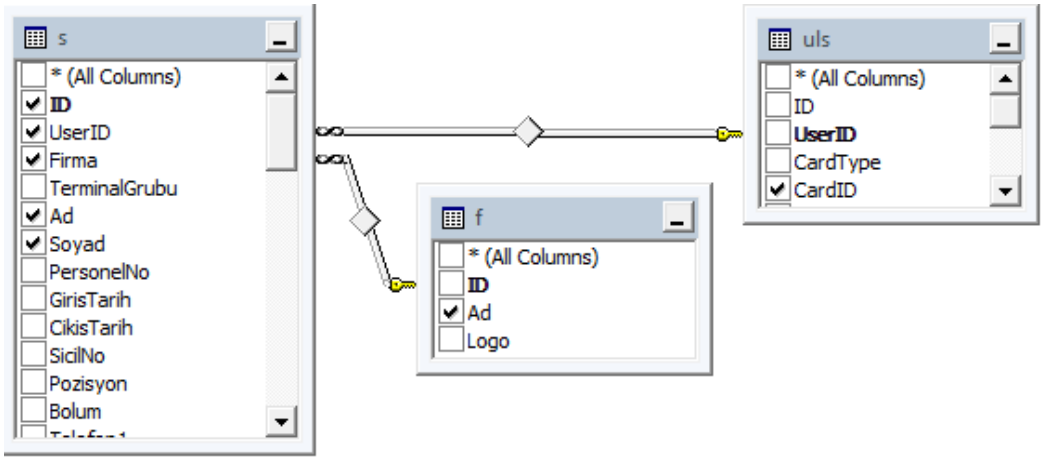
	ID	SicilID	UserID	TerminalID	EventTime	EventCode	FuncCode	Automatic	Deleted	PDKS	ReaderID	Status	UnDelete	ForeignID
1	281934	0	00000368	2	2012-08-24 12:24:35.000	0	255	1	0	0	0	NULL	1	0
2	283163	0	00000368	2	2012-08-28 11:41:58.000	0	255	1	0	0	0	NULL	1	0
3	284872	0	00000368	2	2012-08-31 09:55:00.000	0	255	1	0	0	0	NULL	1	0
4	284965	0	00000368	2	2012-08-31 14:20:53.000	0	255	1	0	0	0	NULL	1	0
5	285852	0	00000368	2	2012-09-03 15:20:16.000	0	255	1	0	0	0	NULL	1	0
6	286501	0	00000368	2	2012-09-04 16:33:41.000	0	255	1	0	0	0	NULL	1	0
7	297095	0	00000368	2	2012-09-12 09:31:42.000	0	255	1	0	0	0	NULL	1	0
8	305503	0	00000368	2	2012-09-21 16:12:27.000	0	255	1	0	0	0	NULL	1	0
9	306618	0	00000368	2	2012-09-24 10:33:59.000	0	255	1	0	0	0	NULL	1	0
10	315840	0	00000368	2	2012-10-03 13:20:28.000	0	255	1	0	0	0	NULL	1	0
11	317919	0	00000368	2	2012-10-04 08:03:20.000	0	255	1	0	0	0	NULL	1	0
12	322548	0	00000368	2	2012-10-05 09:47:14.000	0	255	1	0	0	0	NULL	1	0
13	325795	0	00000368	2	2012-10-08 13:41:40.000	0	255	1	0	0	0	NULL	1	0
14	340779	0	00000368	2	2012-10-17 11:55:17.000	0	255	1	0	0	0	NULL	1	0
15	342041	0	00000368	2	2012-10-19 09:35:29.000	0	255	1	0	0	0	NULL	1	0
16	343439	0	00000368	2	2012-10-22 17:05:10.000	0	255	1	0	0	0	NULL	1	0
17	344997	0	00000368	2	2012-10-23 11:17:07.000	0	255	1	0	0	0	NULL	1	0
18	345003	0	00000368	2	2012-10-23 11:27:45.000	0	255	1	0	0	0	NULL	1	0
19	979724	0	00000368	2	2012-11-01 16:41:23.000	0	255	1	0	0	0	NULL	1	0
20	1228...	0	00000368	2	2012-11-05 12:22:59.000	0	255	1	0	0	0	NULL	1	0
21	1238...	0	00000368	2	2012-11-14 12:26:23.000	0	255	1	0	0	0	NULL	1	0
22	1247...	0	00000368	2	2012-11-22 10:57:44.000	0	255	1	0	0	0	NULL	1	0
23	1248...	0	00000368	2	2012-11-23 08:35:13.000	0	255	1	0	0	0	NULL	1	0
24	281932	0	00000368	4	2012-08-24 12:24:22.000	0	255	1	0	0	0	NULL	1	0

```
select
s.ID,
s.UserID,
s.Ad as nereden,
s.Soyad as nereye,
f.Ad hatsahibi,
uls.CardID,
p.EventTime,
case when p.PDKS=1 then 'giriş' else 'çıkış' end hareket_yonu from pool p
inner join sicil s on s.UserID=p.UserID
inner join cbo_firma f on f.ID=s.firma
inner join userlist uls on uls.UserID=s.UserID
order by p.eventtime
```

Results | Messages

	ID	UserID	nereden	nereye	hatsahibi	CardID	EventTime	hareket_yonu
18123	90	00000090	Pendik	Çekmeköy	Ford	000000000018665	2012-05-14 09:17:49.000	giriş
18124	128	00000128	Gaziosma...	Çekmeköy	Ford	000000000044888	2012-05-14 09:33:09.000	giriş
18125	403	00000128	Başakşehir	Sultanbeyli	----------	000000000044888	2012-05-14 09:33:09.000	giriş
18126	287	00000287	Bahçelie...	Avcılar	Ford	000000000000286	2012-05-14 09:44:10.000	çıkış
18127	1	00000001	Amavutköy	Esenyurt	Ford	000000000000000	2012-05-14 10:17:41.000	giriş
18128	1	00000001	Amavutköy	Esenyurt	Ford	000000000000000	2012-05-14 10:17:43.000	giriş
18129	221	00000221	Amavutköy	Sultangazi	Ford	000000000018711	2012-05-14 10:22:49.000	giriş
18130	221	00000221	Amavutköy	Sultangazi	Ford	000000000018711	2012-05-14 10:23:06.000	giriş
18131	261	00000261	Amavutköy	Ümraniye	Ford	000000000018729	2012-05-14 10:28:30.000	giriş
18132	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:31:28.000	çıkış
18133	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:31:49.000	çıkış
18134	82	00000082	Esenler	Pendik	Ford	000000000018681	2012-05-14 10:36:10.000	giriş
18135	231	00000231	Gaziosma...	Kartal	Ford	000000000044849	2012-05-14 10:37:25.000	giriş

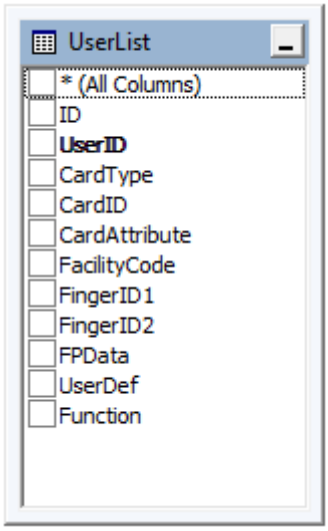





```
select s.ID,s.UserID,s.Firma,s.Ad,s.Soyad,f.Ad firmaadi,uls.CardID
 inner join cbo_firma f on f.ID=s.firma
 inner join userlist uls on uls.UserID=s.UserID
```

Results | Messages

	ID	UserID	Firma	Ad	Soyad	firmaadi	CardID
1	1	00000001	1	Amavutköy	Esenyurt	Ford	000000000000000
2	2	00000002	1	Şile	Sultangazi	Ford	000000000000000
3	3	00000003	1	Beşiktaş	Gaziosmanpaşa	Ford	000000000018626
4	4	00000004	1	Avcılar	Ümraniye	Ford	000000000018606
5	5	00000005	1	Esenler	Kartal	Ford	000000000018607
6	6	00000006	1	Silivri	Beyoğlu	Ford	000000000018631
7	7	00000007	1	Kartal	Sancaktepe	Ford	000000000018605
8	406	00000007	0	Sancaktepe	Kağıthane	----------	000000000018605
9	43	00000043	1	Üsküdar	Esenler	Ford	000000000000042
10	44	00000044	1	Sultanbeyli	Sultanbeyli	Ford	000000000018603
11	45	00000045	1	Başakşehir	Gaziosmanpaşa	Ford	000000000000044
12	46	00000046	1	Gaziosmanpaşa	Bayrampaşa	Ford	000000000018630
13	47	00000047	1	Çatalca	Kağıthane	Ford	000000000018602
14	48	00000048	1	Sancaktepe	Beyoğlu	Ford	000000000018610
15	49	00000049	1	Güngören	Pendik	Ford	000000000018611
16	50	00000050	1	Tuzla	Çekmeköy	Ford	000000000018609
17	51	00000051	1	Beylikdüzü	Bağcılar	Ford	000000000018608
18	52	00000052	1	Bahçelievler	Fatih	Ford	000000000018614



http://localhost:52724/WebSite9/Default.aspx — localhost

ID	UserID	nereden	nereye	hatsahibi	CardID	EventTime	hareket_yonu
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:12	giriş
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:17	giriş
1	00000001	Arnavutköy	Esenyurt	Ford	000000000000000	22.03.2012 15:41:23	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:19:57	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:03	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:08	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:22	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:34	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:39	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	22.03.2012 16:20:46	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	26.03.2012 09:00:02	giriş
57	00000057	Şile	Çekmeköy	Ford	000000000000056	26.03.2012 09:00:11	giriş
74	00000074	Bağcılar	Ataşehir	Ford	000000000018675	29.03.2012 10:58:46	giriş
66	00000066	Sultanbeyli	Çatalca	Ford	000000000018706	29.03.2012 10:58:52	giriş
81	00000081	Ataşehir	Arnavutköy	Ford	000000000018662	29.03.2012 11:00:06	giriş
71	00000071	Güngören	Beykoz	Ford	000000000044876	29.03.2012 11:00:53	giriş
71	00000071	Güngören	Beykoz	Ford	000000000044876	29.03.2012 11:00:59	giriş
79	00000079	Şile	Tuzla	Ford	000000000018674	29.03.2012 11:01:43	giriş
75	00000075	Eyüp	Sancaktepe	Ford	000000000018676	29.03.2012 11:07:15	giriş
75	00000075	Eyüp	Sancaktepe	Ford	000000000018676	29.03.2012 11:07:28	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:07:43	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:07:58	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:08:17	giriş
65	00000065	Üsküdar	Üsküdar	Ford	000000000018668	29.03.2012 11:08:24	giriş