

BÜYÜKBAŞ HAYVANLARIN ZEKİ RF KARTLARLA İNTERNET ÜZERİNDEN KİMLİKLENDİRİLMESİ

(Yüksek Lisans Tezi)

Haydar TUNA

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ekim 2005

ÖZET

Bu çalışmada, kulak numarası ile takibi yapılan büyükbaş hayvanların vücuda enjekte edilen 128 bitlik RFID taşıyıcı yongaları kullanılarak internet üzerinden kimliklendirilmesi amaçlanmıştır. Bunun için 128 bitlik RFID taşıyıcılarına her hayvanda farklı olan kulak numaraları girilmiş daha sonra bu kulak numarası web üzerinde PHP programlama dili ile çalışan MySQL veritabanıyla eşleştirilmiştir. Veritabanında hayvanlarla ilgili yaş, cinsiyet, baba kulakno, anne kulakno, geçirdiği hastalıklar vb. bir çok bilgi tutulmuştur. Web tabanlı çalışan veritabanı ile RFID taşıyıcı yongasının haberleşmesi için ise Delphi programlama dili kullanılmıştır. Sonuçta Türkiye'deki her bir şehirde 387 420 489 tane büyükbaş hayvanın internet üzerinden kimliklendirilebileceği görülmüştür.

Bilim Kodu : 705
Anahtar Kelimeler : Pasif RFID, Web tabanlı hayvan takibi, RFID taşıyıcı, RFID web bağlantısı, hayvancılıkta RFID
Sayfa Adedi : 79
Tez Yöneticisi : Yrd. Doç. Dr. O. Ayhan ERDEM

**IDENTIFYING OF THE CATTLES VIA SMART RF CARDS ON THE
INTERNET**

(M.Sc. Thesis)

Haydar TUNA

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

October 2005

ABSTRACT

In this study was aimed that to identify on the internet the cattles have been following up with ear numbers, using by 128 bits RFID transporter chips which injected into their bodies before. RFID transporters consist of 128 bits was inserted databases different ear numbers for each cattle, after than these were matched with MySQL database working on the internet. On database was kept various knowledge about cattles age, sex, father and mother ear numbers, strain, illnesses having before etc. For communication between database working web based and RFID transporters chips was used Delphi Programming Language. At the result of this study was experienced that 387 420 489 number of cattles in each city of Turkey could be had identities.

Science Code : 705
Keywords : Pasive RFID, Web based animal tracking, RFID transponder, RFID web connection, RFID in the livestock
Page Number : 79
Adviser : Assist. Prof. Dr. O. Ayhan ERDEM

TEŐEKKÜR

Çalıőmalarım süresince yakın ilgi ve yardımlarını esirgemeyen danışmanım Sn. Yrd.Doç.Dr. O.Ayhan ERDEM'e, uygulamada her türlü desteęi saęlayan ETA Elektronik A.Ő yetkililerine böyle bir çalıőma için beni teşvik eden ve manevi desteklerini esirgemeyen bütün arkadaşlarıma içten teşekkürlerimi ifade etmeyi bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ÇİZELGELERİN LİSTESİ	viii
ŞEKİLLERİN LİSTESİ	ix
RESİMLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR LİSTESİ	xi
1. GİRİŞ	1
2. OTOMATİK TANIMA SİSTEMLERİ	5
2.1. Barkod Sistemleri	6
2.2. Optik Karakter Tanıma	7
2.3. Biyometrik Tanıma Sistemleri	7
2.3.1. Parmak izi ile tanıma sistemi	8
2.3.2. Yüz ile tanıma sistemleri	8
2.3.3. Ses ile tanıma sistemleri	8
2.3.4. İris ile tanıma sistemleri	9
2.4. Smart Kartlar	9
2.5. RFID Sistemleri	10
2.5.1. RFID sistemin çalışması	11
2.5.2. RFID taşıyıcı çeşitleri	13
2.5.3. RFID kullanım alanları	16
3. İNTERNET ÜZERİNDEN HAYVAN KİMLİKLENDİRME UYGULAMASI ..	24
3.1. Veritabanının Mantıksal Tasarımı	24
3.2. MySQL Veritabanına Mantıksal Tasarımın Taşınması	28
3.2.1. Linux üzerinde MySQL veritabanının yüklenmesi	28
3.2.3. MySQL’de büyükbaş hayvan veritabanının oluşturulması	31
3.2.4. MySQL’de büyükbaş hayvan veritabanı tablolarının oluşturulması	31
3.3. Apache Web Sunucu Desteği	33
3.3.1. PHP ile Apache’nin derlenmesi	34
3.4. PHP İle MySQL Veritabanının Web Tabanlı Olarak Haberleştirilmesi	36

3.5. Delphi İle Web Uygulamasının Birleřtirilmesi.....	40
4. SONUÇ VE ÖNERİLER.....	48
KAYNAKLAR.....	50
EKLER.....	53
EK -1. Büyükbaş Hayvanların Vücutuna Takılan Taşıyıcı Yongalar.....	54
EK -2. Büyükbaş Hayvan Veritabanında Yer Alan Tabloların Oluřturulması.....	55
Ek -3. PHP Uygulama Program Dosyaları.....	59
Ek -4. Delphi Program Kaynak Kodları.....	70
ÖZGEÇMİŐ.....	79

ÇİZELGELERİN LİSTESİ

	Sayfa
Çizelge 3.1 ISO 11784 standardına göre RFID taşıyıcı bilgi formatı.....	44
Çizelge 3.2 128 bitlik RFID taşıyıcısının bit tanımlaması.....	44

ŞEKİLLERİN LİSTESİ

	Sayfa
Şekil 2.1. Otomatik tanıma sistemlerinin sistematik gösterimi.....	6
Şekil 2.2. Tipik bir EAN barkod kodu	7
Şekil 2.3. RFID sisteminin çalışması	11
Şekil 2.4. Taşıyıcı devresi	12
Şekil 2.5. Cam RFID taşıyıcısı iç yapısı	14
Şekil 2.6. Smart kart şeklinde taşıyıcı.....	14
Şekil 2.7. RFID'nin elektronik gözetimde kullanılması	17
Şekil 2.8. İnsan takibinde RFID kullanımı	18
Şekil 2.9. Hasta takibinde RFID kullanılması.....	19
Şekil 2.10. Sportif aktivitelerde RFID kullanılması.....	21
Şekil 3.1. Bakanlık ve iltarım nesneleri arasındaki ilişki.....	25
Şekil 3.2. İltarım ve ilçetarım nesneleri ve aralarındaki ilişki	25
Şekil 3.3. İlçetarım ve işletme nesneleri ve aralarındaki ilişki	26
Şekil 3.4. Hayvan, hayvan_ işletme, işletme nesneleri ve aralarındaki ilişkiler.....	26
Şekil 3.5. Hayvan, hastalıklar, hayvan_ hastalık nesneleri ve aralarındaki ilişkiler...	27
Şekil 3.6. Hayvan, aşı, hayvan_ aşı nesneleri ve aralarındaki ilişkiler	27
Şekil 3.7. RFID uygulama geliştirme devresi	41
Şekil 3.8. MAX 232 yongasının iç devresi ve tipik uygulama devresi.....	41
Şekil 3.9. İlçe kodu ve şifre ile sisteme giriş.....	44
Şekil 3.10. İşletme listesinin görüntülenmesi	45
Şekil 3.11. Hayvan listesi ve hayvan girişi yapılan ekran.....	46
Şekil 3.12 Hayvan detaylarının alındığı ekran.....	47

RESİMLERİN LİSTESİ

	Sayfa
Resim 2.1. Disk ve para şeklinde RFID taşıyıcı	13
Resim 2.2. Cam RFID taşıyıcı	13
Resim 3.1. S2000 Mikro RFID okuyucu modülü	43

SİMGELER VE KISALTMALAR LİSTESİ

Kisaltmalar Listesi

CMOS	:	Complementary Metal Oxide Semiconductor
EAN	:	European Article Number
EEPROM	:	Electrically Erasable Programmable Read Only Memory
ER	:	Entity Relationship
FRAM	:	Ferroelectric Random Access Memory
OCR	:	Optical Character Recognition
RFID	:	Radio Frequency Identification
SQL	:	Structured Query Language
UPC	:	Universal Product Code

1. GİRİŞ

Klasik yöntemlerle çiftlik hayvanlarının takibinde kulak numarası kullanılmaktadır. Fakat kulak numarası kötü niyetli kişiler tarafından rahatlıkla kopyalanmakta ve bu durum çift hayvan numaralarının tutulmasına neden olmaktadır. Bu nedenle elektronik olarak bir taşıyıcı yonga kullanılarak çiftlik hayvanlarının takip edilmesi fikri ortaya atılmıştır. Otomatik tanıma sistemleri olarak anılan bu elektronik sistemler ilk olarak büyükbaş hayvanlar üzerinde 1970 yılında Hollanda ve Amerika’da denenmiştir (1).

Bu yeni teknoloji sayesinde hayvanlar radyo dalgaları kullanılarak otomatik tanıma sistemine dahil edilmiştir. Elektronik cihazların küçülmesi ile birlikte hayvanların taşıdığı bu yongalarda küçülmüştür. Günümüzde hayvanların vücuduna enjekte edilebilir hale gelmiştir. Vücuda enjekte edilebilen bu taşıyıcı yongalara pasif RFID (Radio Frequency Identification) denmektedir. Bu taşıyıcı yongalar vücuda yerleştirildiği için kaybolma riski yoktur ve pasif RFID ismi kendisine gelen radyo dalgalarından enerjisini almasından dolayı verilmiştir.

Caja ve arkadaşlarının 1998’deki “Evaluation of migratory distance of passive transponders injected in different body sites of adult sheep for electronic identification” isimli çalışmasında 32,5x3,8 mm’lik RFID taşıyıcısı koyunların vücuduna enjekte edilerek RFID taşıyıcısının hayvan vücudunda dolaşmasının bilgi okuma üzerinde etkisi araştırılmıştır. RFID taşıyıcısı 26 koyunun vücutlarının 8 farklı bölgesine enjekte edilmiş ve enjeksiyon daha kolay hale getirilerek hayvanların daha az tepki vermesi sağlanmıştır. Ayrıca RFID taşıyıcısı enjekte edilen bölgelerin belli olması için ameliyatlarda kullanılan kısaçlarla tutturulmuştur. Daha sonra örnek olarak seçilen 10 koyunun vücudunun değişik noktalarına yerleştirilen RFID taşıyıcılarının hareketleri radyoterapi yöntemi kullanılarak izlenmiş ve RFID taşıyıcısında bilgi okuma seviyelerine de bakılmıştır. Sonuçta koltuk altı ve kulak bölgesinin ön tarafındaki bir boşluğa yerleştirilen RFID taşıyıcıların daha az hareket ettiği ve daha iyi bilgi okumanın gerçekleştiği görülmüştür (2).

Benzer bir çalışmayı Klindtworth ve arkadaşları “Electronic identification of cattle with injectable transponders” isimli çalışmada gerçekleştirmiştir. Bu çalışmada ise büyükbaş hayvanların vücuduna RFID taşıyıcıları enjekte edilmiştir. RFID taşıyıcısının yerleştirildiği bölge olarak scutulum olarak adlandırılan kulağın ön tarafında bulunan boşluk seçilmiştir. Daha önce Caja ve arkadaşlarının 1998’deki yaptığı çalışmaya benzer bir şekilde hayvanın kulağının ön tarafındaki boşluğa yerleştirilen RFID taşıyıcılarının daha az hareket ettiği ve bilgi okumanın daha iyi olduğu sonucuna ulaşılmıştır (3).

Jansen ve Eradus’un 1999’daki “Future developments on devices for animal radio frequency identification” isimli çalışmasında RFID teknolojisinin geleceği üzerinde araştırma yapılmıştır. RFID taşıyıcıların boş yongalara kopyalanması ile ulusal çapta çift hayvan numaralarının tutulmasına neden olabileceğini ortaya konmuştur. Bu durumun RFID taşıyıcı yongalarının ISO 11784 ve ISO 11785 standartlarına uyması halinde önüne geçilebileceği vurgulanmıştır. Daha sonra RFID taşıyıcıları üç sınıfa ayrılmıştır. Birinci sınıfta doğrulama amaçlı kullanılan RFID taşıyıcıları, ikinci sınıfta hayvanların bir çok bilgisinin hafızada sayfalar halinde tutulduğu RFID taşıyıcıları ve üçüncü sınıfta hayvanlarla ilgili vücut sıcaklığı kalp atışı vb. sağlıkla ilgili bilgilerin anlık olarak alınabildiği sensör şeklindeki RFID taşıyıcıları yer almıştır. Son kısımda RFID taşıyıcılarının güvenlik seviyeleri beş seviyeye ayrılmış. Her seviyenin özellikleri ve nasıl geçilebileceği üzerinde durulmuştur (4).

Wismans’ın 1999’daki “Identification and registration of animals in the European Union” isimli çalışmasında Avrupa Birliğinde bulunan hayvanların RFID kullanılarak ulusal çapta nasıl tanımlanabileceğinin çerçevesi çizilmiştir. Oluşturulması gereken yönetmeliklerde olması gerekenler, hayvan tanımlamada kullanılacak metotlar, hayvan tanımlamada kullanılacak veritabanında olması gerekenler, kalite kontrol ve fiyatlandırmanın nasıl yapılacağı üzerinde durulmuş ve RFID’nin bu tarz büyük çaplı bir işte kullanılabileceği sonucuna ulaşılmıştır (5).

Jansen ve Eradus’un “Animal identification and monitoring” isimli çalışmasında şu anda geliştirilmekte olan üçüncü nesil RFID taşıyıcılarla anlık olarak bir hayvanın

sağlık durumunun nasıl takip edileceği ve karşılaşılan sorunlar üzerinde durulmuştur. Bir hayvanın kalp atış hızı, vücut sıcaklığı vb. sağlık bilgilerinin alınabilmesi için sensör şeklinde RFID taşıyıcıların kullanılması gerektiği ve bunların henüz küçük boyutlara indirgenemediği vurgulanmıştır. Sensör şeklindeki RFID taşıyıcılarla gelen bilgilerin yapay sinir ağları ve bulanık mantık kullanılarak işlenebileceği sonucuna ulaşılmıştır (6).

Bu çalışmada hayvanlara enjekte edilen RFID taşıyıcıları kullanılarak ulusal çapta web tabanlı olarak hayvanların kimliklendirilmesi amaçlanmıştır. Hayvanların ulusal çapta kimliklendirilmesinde tarım bakanlığı tarafından kullanılan kulak numarası standardından faydalanılmıştır. Bu kulak numarası ISO 11784 ve ISO 11785 standartlarına uygun olarak 128 bitlik RFID taşıyıcılarına yerleştirilmiş ayrıca bu kulak numaralarının web üzerinde çalışan veri tabanı ile eşleştirilmesi sağlanmıştır.

Çalışma dört bölümden meydana gelmektedir. Birinci bölümde büyükbaş hayvan takibinde RFID teknolojisinin kullanılması ile ilgili daha önceden yapılan çalışmalar, tezin amacı ve kapsamı üzerinde durulmuştur.

İkinci bölümde RFID dışında kalan otomatik tanıma sistemleri ve RFID'nin bu tanımlama sisteminin neresinde olduğu üzerinde durulmuş. Daha sonra RFID sisteminin çalışması, RFID taşıyıcı yonga çeşitleri ve RFID uygulamaları detaylı olarak anlatılmıştır.

Üçüncü bölümde Wismans'ın 1999'daki "Identification and registration of animals in the European Union" isimli çalışması neticesinde ortaya çıkan Avrupa Birliği hayvan veritabanı standardından faydalanılarak büyükbaş hayvan veritabanının mantıksal tasarımı gerçekleştirilmiştir. Mantıksal tasarımı gerçekleştirilen bu veritabanı MySQL veritabanı sunucusu ve PHP programlama dili yardımıyla internet ortamına taşınmıştır. Daha sonra internet ortamında web tabanlı olarak çalıştırılan uygulama ile RFID'nin haberleşmesi için Delphi'de bir arayüz hazırlanmıştır.

Dördüncü ve son bölümde RFID kullanılarak yapılan hayvan kimliklendirmenin daha güvenli olduğu ve daha çok sayıda hayvan tanımlamanın yapılabileceği gösterilmiştir.

2. OTOMATİK TANIMA SİSTEMLERİ

Son yıllarda otomatik tanıma sistemleri kısaca Auto-ID, üretim hatlarında endüstride, satış ve dağıtım sistemleri gibi birçok serviste çok geniş kullanım alanı bulmaktadır. Bu sistemler sayesinde insanlar, hayvanlar, taşınan eşya ya da ürün hakkında bilgi alınabilmektedir (7).

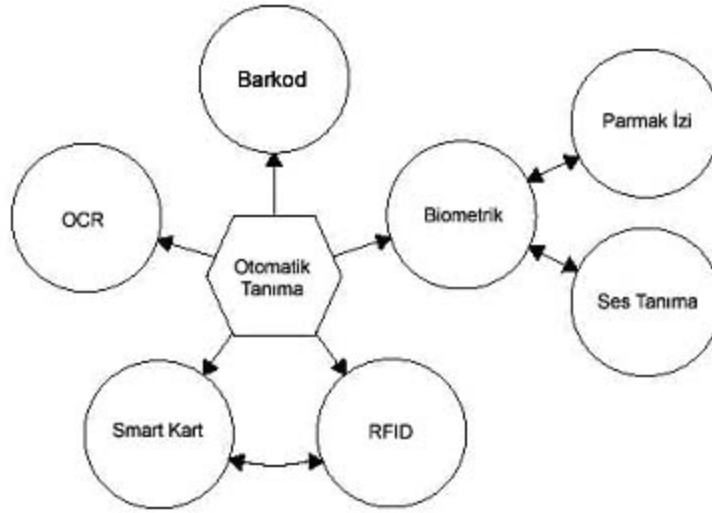
Otomatik tanıma sistemlerini gelişmesini tetikleyen en önemli olay çok yönlü barkod etiketlerinin kullanılmasıdır. Fakat barkodların, yeniden programlanmasının mümkün olmaması ve düşük miktarda veri taşımamasından dolayı kullanım alanları daralmıştır (7).

Barkodlardaki sorunların teknik olarak en iyi çözümü taşınabilir bir silikon yongaya verinin kaydedilmesidir. Bundan dolayı günlük hayatta sıklıkla elektronik ortamda veri taşımada kullanılan smart kartlar ortaya çıkmıştır. Smart kartlara örnek olarak banka ve telefonlarda kullanılan kartlar verilebilir. Smart kartlar, temelde temas yüzeyi esasına göre çalışır. Temas yüzeyi olarak mekanik kontaklar kullanılır. Mekanik kontaklarda aşırı okumadan dolayı zamanla bozulma meydana gelmektedir. Bundan dolayı okuma problemleri ortaya çıkabilmektedir. Fakat sistemler arasında verinin taşınması ve okunması barkod sistemlerine göre daha esnektir. Barkod sistemlerinin aksine smart kartlarda yeniden programlama yapılabilir.

Smart kartlarda mekanik kontakların aşınmasından dolayı ortaya çıkan okuma problemlerinin önüne geçmek için kontak yüzeyi olmadan radyo dalgaları ile çalışan smart kartlar geliştirilmiştir. Normalde, kontak kullanmadan çalışan bir veri okuyucunun elektronik olarak bilgi alıp vermesi için enerji gerekir. Fakat RFID olarak adlandırılan bu yeni teknoloji sayesinde okuyucudan gönderilen taşıyıcı radyo dalgaları ile silikon yongaya hem enerji verilmekte hem de bilgi alışverişi yapılmaktadır. Şu anda birçok şirket RFID sistemlerini geliştirmek ve pazarlamak için ciddi olarak çalışmaktadır. 2000 yılında tüm dünyada yaklaşık olarak 900 milyon dolar yıllık RFID satışının, 2005 yılında yılda 2650 milyona çıkması

beklenmektedir. Bu yüzden RFID pazarı cep telefonu ve telsiz telefon gibi cihazları içine alan mobil teknolojilerin hızla büyüyen sektörleri arasında yer almaktadır (7).

Bu çalışmadaki otomatik tanıma sistemlerinin sistematik gösterimi Şekil 2.1'de verilmiştir.



Şekil 2.1. Otomatik tanıma sistemlerinin sistematik gösterimi

2.1. Barkod Sistemleri

Barkod koyu renkli çubuklarla boşluk biçimindeki çubukların yan yana dizilmesiyle oluşan bir kod çeşididir. Barkod okuyucudan gelen ışınlar koyu renkli çubuklar tarafından emilir, boşluk şeklindeki çubuklar tarafından ise yansıtılır. Bunun neticesinde barkod etiketi yapıştırılan nesnenin bilgileri okunur. Bu işlem sırasında sağlıklı bir okumanın gerçekleşmesi için barkod okuyucu tarafından alınan bilginin en az %60 ile %70 arasında tutarlı olması gerekir (8).

Şu anda fiziksel olarak aynı özelliklere sahip olmasına rağmen kod dağılımları farklı olan 10 çeşit barkod sistemi mevcuttur. En bilinen barkod sistemi 1976 yılında kullanılmaya başlanan ve büyük alışveriş merkezlerinde kullanılan EAN (European

Article Number) barkod sistemidir. EAN barkod kodunun, A.B.D’de karşılığı 1973 yılında geliştirilen UPC (Universal Product Code) barkod kodudur. Günümüzde UPC barkod sistemi EAN sisteminin alt kümesi olduğu için bu iki kod sistemi birbirini desteklemektedir.

EAN kodu 13 karakterden oluşur. Bu karakterler ülke kodu, şirket ya da üretici kodu, üretilen malın kodu ve kontrol kodundan meydana gelir. Tipik bir EAN barkod kodu Şekil 2.2.’de görülmektedir.



Şekil 2.2. Tipik bir EAN barkod kodu

2.2. Optik Karakter Tanıma

Kısaca OCR (Optical Character Recognition) olarak adlandırılan Optik Karakter Tanıma Sistemleri ilk olarak 1960 yılında kullanılmaya başlanmıştır. OCR için özel fontlar geliştirilmiş bu yüzden insanlar tarafından ya da makinelerle yazılan yazılar, otomatik olarak okunabilmiştir. OCR sisteminin en önemli avantajı görsel olarak verinin okunabilmesidir.

Günümüzde OCR üretim, yönetim, bankacılık ve hizmet sektörü gibi değişik alanlarda kullanılmaktadır.

2.3. Biyometrik Tanıma Sistemleri

Biyometri, canlı varlıklar temel alınarak yapılan ölçme ve sayma bilimi olarak tanımlanabilir. Biyometri kullanılarak hataya meydan vermeden kişilerin fiziksel

karakteristiklerine göre tanıma işlemi gerçekleştirilebilir. Uygulamada tanıma işlemi insanların parmak izi, yüzü, sesi veya gözlerinde bulunan iris tabakası kullanılarak yapılmaktadır.

2.3.1. Parmak izi ile tanıma sistemi

Parmak izi benzersizdir ve yaşa göre ya da diğer özelliklere göre değişmez. Her insanın farklı parmak izi vardır (9). Bu yüzden parmak izi, en güvenilir kişisel doğrulama araçlarından biridir. Suçlu takibi erişim kontrolü ya da banka kartı doğrulaması gibi sivil ve adli bir çok uygulamada çok önemli rol oynamaktadır (10).

Bu sistemde yapılacak ilk iş kullanıcının optik tarayıcılar kullanılarak parmak izinin alınmasıdır. Bundan sonra alınan bu parmak izi ile ilgili detaylar (kenarlar ve çukurluklar gibi) bir algoritmadan geçirilerek veritabanına işlenir. Veritabanına detayları işlenen bu parmak izi daha sonra karşılaştırma maksadıyla kullanılır (11).

2.3.2. Yüz ile tanıma sistemleri

Yüz ile tanıma sistemleri, daha önceden veritabanına işlenen yüz resimleri ile bilinmeyen bir yüzün bilgisayar destekli olarak karşılaştırılması esasına göre çalışır. Bu sistemler erişim kontrolünde ya da kalabalık içerisinde kişilerin seçilmesinde kullanılır. Yüz ile tanıma sistemlerindeki temel sorun düşük kalitedeki resimlerin karşılaştırmada problemlere neden olmasıdır (12). Bunun dışında kişilerin gözlük vb. aksesuarlar takması da karşılaştırmada sorunlara neden olabilmektedir. Şu anda yapılan çalışmalarda bu sorunlar en aza indirilmiştir. A.B.D’de şu anda bazı havaalanlarında bu sistem denenmektedir.

2.3.3. Ses ile tanıma sistemleri

Ses perdesi her insan için farklı ton ve karakteristiğe sahiptir. Bu durum seslerin frekansa göre grafikleri incelendiğinde rahatlıkla görülebilir. Bu farklılıklardan faydalanarak ses ile tanıma sistemleri geliştirilmiştir (13). Diğer biyometrik tanıma sistemlerinde olduğu gibi bu otomatik tanıma sisteminde benzer şekilde tanımlama

sistemine dahil edilecek kişinin sesi kaydedilip veritabanına işlenir. Daha sonra veritabanına kaydedilen bu ses sinyali karşılaştırma amacıyla kullanılır.

Ses ile tanıma sistemleri, diğer tanıma teknolojilerinin aksine dile, ülkeye, kültüre göre değişiklik gösterir. Örneğin Japonca'yı baz alan bir ses ile tanıma sisteminde, kullanılan dilin yapısı çok zor olduğundan daha iyi teknolojinin kullanılması gerekir. Ses ile tanıma teknolojileri, genellikle telefon ya da diğer haberleşme sistemlerinde kullanım alanı bulmaktadır (14).

2.3.4. İris ile tanıma sistemleri

Gözde bulunan iris tabakası kişisel tanımlama için benzersiz ve tutarlı özelliklere sahip olduğu için son 10 yıldır biyometrik teknolojileri içerisinde en çok araştırma sahası bulan konulardan biridir (15).

İris ile tanıma teknolojilerindeki temel fikir gözün iris tabakasının resminin çekilerek üzerinde işlem yapılmasıdır. Alınan bu resim ile kişinin iris tabakasına uygun karakteristik değerler çıkartılıp bir veritabanına işlenir. Daha sonra veritabanındaki bu karakteristikler karşılaştırma amacıyla kullanılır.

2.4. Smart Kartlar

Fiziksel olarak smart kartlar, bir kredi kartı ve bu kartın sol üst köşesine yerleştirilen bir ya da daha çok yarıiletken meydana gelir. Smart kartlarda yarıiletken olarak bir mikroişlemci ve RAM (Random Access Memory), ROM(Read Only Memory), EEPROM(Electrically Erasable Programmable Read-Only Memory) gibi hafıza blokları kullanılır (16).

Mikroişlemci, smart kartların en önemli parçasıdır. Mikroişlemci, özellikle kart okuyucudan verileri almak ya da verileri göndermek için kullanılır. Hafıza blokları ise verileri sürekli ya da geçici olarak tutmak için kullanılır. Bu şekilde smart kartla okuyucu arasında çift yönlü haberleşme sağlanmış olur (17).

Smart kartların en önemli avantajı istenmeyen okuma ve erişimden verinin korunmasıdır. Bundan dolayı smart kartlar, bilgi teknolojileri kullanılan işleri ya da finanssal işleri kolaylaştırmak, güvenli hale getirmek ve bu işlemleri daha ucuza temin etmek için sıklıkla kullanılmaktadır.

2.5. RFID Sistemleri

RFID ilk olarak ikinci dünya savaşı sırasında radar kullanılarak dost ve düşman asker uçaklarını ayırmak için kullanılmıştır. Askeri alanda kullanılan RFID teknolojisi ticari olarak elektronik gözetim sistemlerinde 1960 yılında ortaya çıkmıştır. Bu tarihten sonra RFID teknolojisi hızlı bir gelişim izlemiş bugün çok değişik alanlarda kullanım olanağı bulmuştur (18).

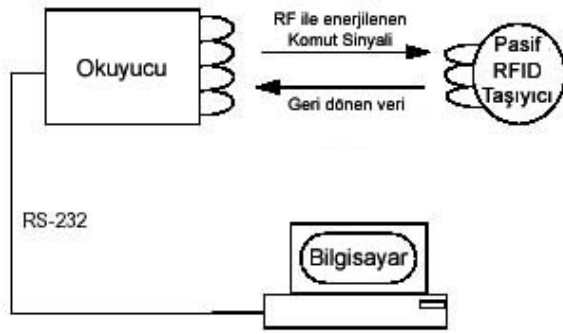
RFID sistemi temelde bir RF taşıyıcı yonga ile RF okuyucunun modüle edilmiş geriye yansımali dalgaları kullanarak kablosuz olarak haberleşmesi esasına göre çalışır. RFID'ye akıllı barkodlar da denilmektedir. Fakat barkodlara göre RFID'nin birkaç üstünlüğü vardır (19). Bu üstünlükleri şunlardır:

- Barkod okuyucuların aksine RFID'de okuyucunun doğrudan taşıyıcı yongayı görmesine gerek yoktur. Bu yüzden RFID kullanılarak birkaç metreden herhangi bir yönde okuma yapılabilir.
- RFID ile birden fazla yonganın tanımlaması yapılabilir.
- RFID barkod gibi kirden, nemden vb. dış etkilerden etkilenmez.

RFID taşıyıcıları pasif, yarı pasif ve aktif olmak üzere üç sınıfa ayrılır. Pasif RFID taşıyıcıları çalışmak için bataryaya ihtiyaç duymaz. Aktif ve yarı pasif RFID taşıyıcıları ise çalışmak için bataryaya ihtiyaç duyarlar. Bu taşıyıcıların mesafesi pasif RFID taşıyıcı yongalarına göre daha yüksektir. Yarı pasif RFID taşıyıcısında sadece enerji almak için batarya kullanılır. Aktif RFID taşıyıcı yongaları ise kablosuz telefon, küçük cep telsizleri gibi geleneksel haberleşme cihazlarının bir alt kümesidir (20).

2.5.1. RFID sistemin çalışması

RFID teknolojisi bilgi göndermek için elektromanyetik enerjiden faydalanır. RFID'nin daha önceden de söylendiği gibi okuyucu ve taşıyıcı olmak üzere iki temel elemanı vardır. Bunlardan okuyucu bir bilgisayara bağlanır. Okuyucu tarafından pasif RFID taşıyıcından alınan benzersiz numara bilgisayara aktarılır. Bilgisayarda ise RFID ile kimliklendirilen insan, hayvan ya da eşyanın diğer bilgileri tutulur (21).



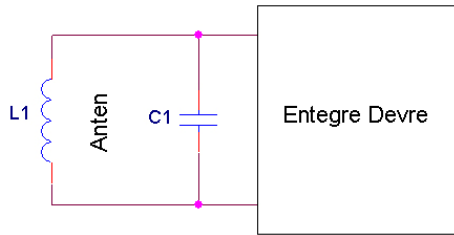
Şekil 2.3. RFID sisteminin çalışması

Tipik bir taşıyıcı devresi Şekil 2.4.'te görülmektedir. Taşıyıcı devresi, anten ve silikon yongadan meydana gelir. Silikon yonga içerisinde ise modülasyon devresi ve içeriğini kaybetmeyen hafıza bulunur. Taşıyıcı, okuyucu tarafından gönderilen elektromanyetik radyo frekans dalgasından enerjisini alır. Gönderilen bu radyo frekans dalgasına taşıyıcı sinyal denir. Taşıyıcı sinyal, RFID taşıyıcısının anteninden geçtiğinde, bobin uçlarında alternatif akım meydana gelir. Alternatif akım, cihaz için gerekli doğru akıma dönüştürülmek için doğrultulur. Doğru akım, istenilen seviyeye ulaştığında ise taşıyıcı çalışmaya başlayacaktır. Bu anda cihazda kaydedilen bilgiler okuyucuya geri gönderilir. Yapılan bu işleme daha önceden de söylendiği gibi geriye yansıma denir. Geriye yansıma sinyalinin alınmasıyla taşıyıcıya bilgi yüklenebilir ya da taşıyıcıdan bilgi alınabilir.

RFID taşıyıcıları, hafıza hücresine bağlı olarak sadece okunabilir ve hem okunabilir hem de yazılabilir olarak iki sınıfa ayrılır. Hafıza hücresi EEPROM ya da

FRAM'den (Ferroelectric Random Access Memory) yapılmış olabilir. EEPROM, CMOS (Complementary Metal Oxide Semiconductor) silikon yonga, FRAM ise ferroelektrik tabanlı yongadır. CMOS yonga teknolojisinin işlenmesi daha kolay olduğu için FRAM'e göre daha ucuzdur. Bu nedenle ekonomik açıdan ve kullanım kolaylığından dolayı, uzun süredir değişik uzaktan RF ile algılama sistemlerinde CMOS tabanlı yongalara sahip pasif RFID taşıyıcıları tercih edilmektedir. Özellikle hayvan takip ve kontrol sistemlerinde yoğun olarak bu tarz RFID taşıyıcılar kullanılır.

Pasif RFID taşıyıcıları, enerjisini uzaktan okuyucunun radyo frekans sinyali vasıtasıyla aldığı için mikro watt seviyelerinde çok küçük enerjiyle çalışır. Bu yüzden okuma seviyesi çok düşüktür. Okuma seviyesi, silikon yongadan çekilen akım, antenin büyüklüğü haberleşme protokolü, veri hızı, okuyucunun duyarlılığı, RF güç seviyesi ve frekans gibi değerlere bağlı olarak da değişir.



Şekil 2.4. Taşıyıcı devresi

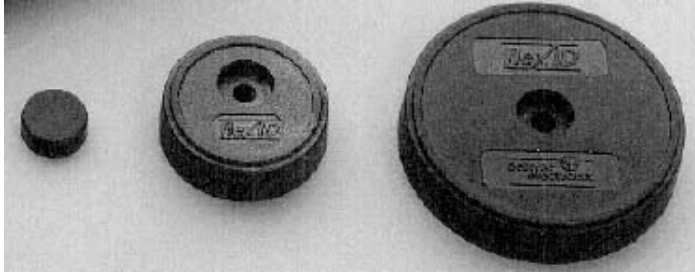
RFID uygulamalarında genelde düşük frekans aralığı (125 KHz – 400 KHz) kullanılır. Bu durum silikon yongadan kaynaklanmaktadır. Günümüzdeki uygulamalarda tipik taşıyıcı frekansı (Okuyucunun verici frekansı) 125 KHz ile 2,4 GHz arasındadır.

2.5.2. RFID taşıyıcı çeşitleri

RFID'nin uygulama alanlarına göre taşıyıcılar çeşitli şekillerde olabilir. Yapılan uygulamaya göre taşıyıcı seçimi önemlidir. Uygulamaya göre uygun taşıyıcının seçilmesi gerekir. Sıklıkla kullanılan RFID taşıyıcı çeşitleri şunlardır:

Disk şeklinde RFID taşıyıcı

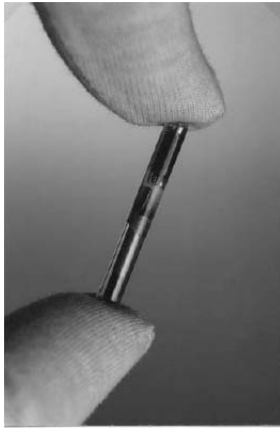
Disk şeklindeki taşıyıcıların birkaç mm'den 10 cm'ye kadar boyutları vardır. Bu taşıyıcıların, merkezinde vidalanması için bir delik bulunur ve çok geniş çalışma sıcaklığında polyester yada epoksi reçineli tipleri de bulunmaktadır. Disk şeklindeki değişik RFID taşıyıcıları Resim 2.1'de görülmektedir.



Resim 2.1. Disk şeklinde değişik RFID taşıyıcıları

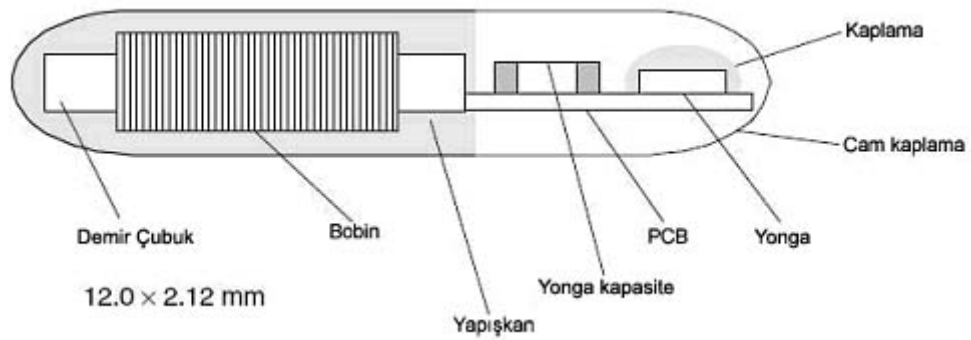
Cam RFID taşıyıcı

Cam RFID taşıyıcıları, hayvanları tanımlamak için deri altına enjekte edilmek üzere geliştirilmiştir. Resim 2.2.'de görülen taşıyıcı buna örnek olarak gösterilebilir.



Resim 2.2. Cam RFID taşıyıcısı

Şekil 2.5.'de Cam RFID taşıyıcısı yapısı görülmektedir. Bu şekle göre Cam RFID taşıyıcısı, bir baskı devre üzerinde yerleştirilen kapasitif eleman ve yonga, anten olarak görev yapan demir çubuk ve bobinden meydana gelir. Bobin, demir nüve üzerine 0,03 mm kalınlığında sarılan tel ile yapılır. Ayrıca cam RFID'de meydana gelen mekanik problemleri en aza indirmek için ince bir yapışkan malzeme de kullanılır.



Şekil 2.5. Cam RFID taşıyıcısı iç yapısı

Plastik RFID taşıyıcıları

Plastik RFID taşıyıcıları, özellikle yüksek mekanik hareketleri karşılamak için geliştirilmiştir. Plastik kılıf diğer ürünlere kolaylıkla tutturulabilir. Örnek olarak araba anahtarlarına takılabilir.

İçinde cam RFID taşıyıcısı ile aynı malzemeler kullanılır fakat çok daha uzun bobin kullanıldığı için etki alanı daha fazladır. Ayrıca daha büyük mikroişlemci içerir ve mekanik titreşimlere karşı daha az duyarlıdır. Bu nedenle otomotiv endüstrisinde sıklıkla kullanılır.

Alet ve gaz şişelerine monte edilen RFID taşıyıcılar

Metal yüzeylere taşıyıcıların endüktif olarak bağlanmasıyla elde edilir. Taşıyıcı bobini demir kalıp üzerine sarılır. Metal yüzeyler bu şekilde anten görevi yapar. Bu tarz sistemler özellikle metal meşrubat kutularında kullanılmaktadır.

Anahtarlık şeklinde RFID taşıyıcıları

Anahtarlık şeklinde RFID taşıyıcıları, yüksek güvenlik gereksinimlerini karşılamak amacıyla mekanik anahtarlarla birlikte kullanılabilir. Bu tarz taşıyıcılar anahtarlıkla birlikte plastik bir kılıf içerisine yerleştirilir. Ofis ya da işyerinde giriş ve çıkışta kullanılması oldukça yaygındır.

Saat biçiminde RFID taşıyıcıları

Saat biçiminde RFID taşıyıcılar, 1990'ların başında ilk olarak kayak pistlerinde geçiş kontrolü için kullanılmıştır. Saat içerisinde ince bir elektronik baskı devre vardır ve baskı devre üzerinde sargılarla elde edilmiş anten bulunur. Anten bobininin büyüklüğünden dolayı bu tarz sistemlerde okuma mesafesi diğer RFID taşıyıcılarına göre daha yüksektir (7).

Smart kart şeklinde RFID taşıyıcıları

Smart kart şeklinde RFID taşıyıcıları, telefon kartlarına ya da kredi kartlarına benzer ve RFID sistemlerin gelişmesinde önemli rol oynamıştır. Tipik bir smart kartın görünüşü Şekil 2.6.'da görülmektedir. Bu tarz sistemlerin en önemli avantajı çok geniş bobin yüzeyine sahip olmalarıdır. Bu yüzden smart kart şeklindeki RFID taşıyıcıları daha geniş etki alanına sahiptir.

Smart kart şeklinde RFID taşıyıcıları, 4 tabaka PVC tabaka arasına ince bir tabaka eklenmesiyle üretilir. Bu tabakalar, 100 C⁰ üstündeki yüksek sıcaklık ve yüksek basınç altında preslenir.



Şekil 2.6. Smart kart şeklinde taşıyıcı

Smart kart şeklindeki RFID taşıyıcılarının üzerinde taşınmasını kolaylaştırmak amacıyla telefon kartlarındaki gibi yazı ve resimler kullanılabilir. Örneğin geçiş yapan personelin kimlik bilgileri ya da resmi Smart kart şeklindeki RFID taşıyıcılara eklenebilir. Smart kart şeklinde RFID taşıyıcılarının kalınlığı ISO 7810 standardıyla belirlenir ve maksimum 0,8 mm'dir.

Etiket şeklinde RFID taşıyıcıları

Etiket şeklinde RFID taşıyıcıları, 0.1 mm kalınlığındaki plastik katmana bobin yapıştırılmasıyla elde edilir. Elde edilen bu RFID taşıyıcılarının, kendiliğinden yapışkanı vardır, değişik yerlere tutturulmasının kolaylaştırılması için esnek bir yapıya sahiptir, üzerine kolayca yazı da yazılabilir (7).

Yonga üzerine yerleştirilen RFID taşıyıcılar

Yonga üzerine yerleştirilen RFID taşıyıcılar, hibrit teknoloji olarak adlandırılır. Taşıyıcı üstündeki yonga anten görevini de üstlenmiştir.

Bobin, baskı devre şeklinde yonga üzerinde bulunan yollar kullanılarak elde edilir. Bu işlem normal CMOS bir yüzeye özel mikrogalvanik tabaka eklenerek yapılır. Bobin tek bir katman olarak spiral bir şekilde silikon yonganın izolatörüne doğrudan yerleştirilir. Daha sonra yonga ile bobin doğrudan birbirine bağlanır.

2.5.3. RFID kullanım alanları

RFID 1940 yılında ilk olarak askeri alanda kullanılmasından bu yana güvenlikten, üretim kontrolüne bir çok alanda kullanım alanı bulmuştur. Bu alanlar şunlardır:

RFID ile elektronik gözetim

Şekil 2.7'de görülen gözetim amacıyla bir mağazaya kurulan RFID sisteminin birinci şeklinde çok ucuz olan bir bitlik RFID taşıyıcısı elbiselere yerleştirilmiştir. İkinci şekilde kötü niyetli bir kişi kasadan geçmeden bir elbise almış, üçüncü şekilde kasadan geçmeden başka bir yoldan kaçmaya çalışan kötü niyetli bu kişi RFID

okuyucu tarafından fark edilmiş, dördüncü şekilde ise para ödemedi geçiş yapılan bu bölgede alarm çalışmıştır. Bu uygulama RFID'nin gözetimde kullanılmasının en temel uygulamalarından biridir (22).



Şekil 2.7. RFID'nin elektronik gözetimde kullanılması

RFID'nin elektronik gözetimde kullanılmasının diğer bir örneği ise doküman doğrulamada karşımıza çıkar. RFID taşıyıcıları bu uygulamada belli bir yerdeki dokümanı tanımlamak için kullanılır. Ocak 2004'te Golden Globe ödülleri dikkat çekmeden RFID kullanılarak taşınmıştır. Bununla ödüllerin, sahtesinin yapılmasının önüne geçilmiş, güvenlik artırılmış, hızlı ve kolay taşınma sağlanmış ve ödüller katılımcılara ulaştırılmıştır (22).

RFID ile erişim kontrolü

Araba, park ya da ofisler gibi alanlara RFID sistemi yerleştirilmiş kartlarla, anahtarlarla ya da benzer taşıyıcılarla güvenli bir erişim sağlanabilir. RFID taşıyıcısı üzerinde benzersiz bir numara ile tanımlanabilir ve bu suretle RFID taşıyıcısı ile çok güvenli geçişler sağlanabilir. Örneğin belli bir personelin ofis içerisine erişimi sağlanırken aynı personelin bilgisayar sunumcu odasına ya da gizli veri depolama alanına erişimi engellenebilir.

RFID'nin otomobillerde erişim kontrolü için kullanılmasına İngiltere'de 1992 yılında hırsızlık olaylarının aşırı artması neticesinde başlanmıştır. Bu uygulamada RFID taşıyıcısı anahtarlığa, RFID okuyucusu ise ateşleme sistemi içerisine monte edilmektedir. Sürücü arabayı çalıştırmak istediğinde öncelikle taşıyıcıdaki bilgisi kontrol edilmektedir. Eğer taşıyıcı yok ya da yanlış taşıyıcı varsa arabanın çalışmasına izin verilmemektedir (23).

Erişim kontrolünde kullanılan RFID taşıyıcısı ile çok az bir maliyetle güvenlik ve doğrulama sağlanır. Bu taşıyıcılar kolaylıkla çoğaltılamaz ve etkisiz hale getirilemez.

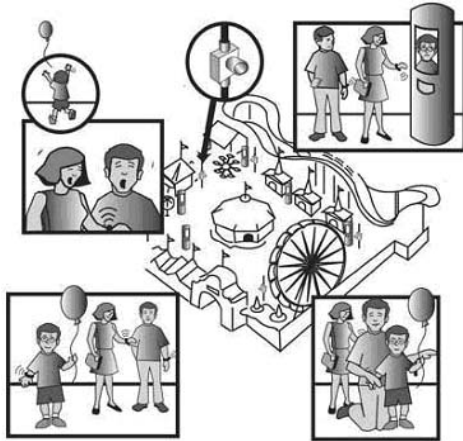
RFID'nin ilaç endüstrisinde kullanımı

İlaç dağıtımında yaşanan güvenlik sorunları ilaç veren firmaların ve ilaç dağıtımı yapan firmaların her ikisinin de temel sorunudur. İlaç dağıtımı ile ilgili güvenlik problemi ikinci bir toptancının ya da dışarıdan gelen toptancıların ilaçları yeniden paketlemesinden kaynaklanmaktadır.

Bu uygulamada ilaç paketlerine ve eczanelere dağıtılan ilaç broşürlerine RFID taşıyıcıları yerleştirilir. Daha sonra ilaçlar ve broşürler eczanelere dağıtılır. Eczanelerde ise RFID okuyucusu kullanılarak ilaç broşüründeki üretici bilgisi ile ilaç üzerindeki bilgi karşılaştırılır. Bu şekilde ilaçların sahte olup olmadığı anlaşılır (24).

RFID ile insan takibi

Dünya üzerindeki bazı eğlence parkları anne babalara çocuklarını takip etmek ve tam olarak yerlerini saptamak için bilezik şeklinde RFID kullanarak hizmet sunmaktadır. Şekil 2.8.'de görüldüğü gibi ebeveynler çocuklara verilen RFID taşıyıcılarının karşılığı ile donatılmıştır. İzleme istasyonları doğru ebeveynle çocuğun eşleştirmesini yapar. Parkta çocuğun yeri belirlendiğinde kamera oraya odaklanarak çocuğun o anki durumunu gösterir (22).



Şekil 2.8. İnsan takibinde RFID kullanımı

RFID okullarda da insan takibinde kullanılmaktadır. Bu uygulamada öğrencilere bilezik ya da kol bandı şeklinde RFID taşıyıcıları takılmaktadır. Bu sayede derse devam eden ve etmeyen öğrencilerin takibi otomatik olarak yapılmaktadır (25).

Diğer bir uygulama ise çok yaşlı ve sakat insanların yerinin ve yönünün izlenmesidir. Örneğin Alzheimer hastası olan yaşlı hastaya oda anahtarı şeklinde bir RFID taşıyıcısı takıldığında hastanın odayı terk etmesi durumunda alarm verdirilebilir.

RFID'nin hastanelerde kullanımı

RFID'nin hastanelerde bebek ünitelerinde kullanılmaktadır. RFID taşıyıcıları yeni doğmuş bebeklerin kollarına bileşik şeklinde takılmakta ve bebekleri hastaneden kötü niyetli kişilerin kaçırmaması durumunda sistem hastane personelini anında uyarılmaktadır (25).

Bilezik şeklindeki RFID taşıyıcıları hasta ile ilgili benzersiz bilgileri ve durumunu tanımlamakta da kullanılabilir. Bu şekilde kullanılan mobil veritabanı hasta ile ilgili ek bilgileri örneğin kan grubu, alerjileri ve önemli sağlık bilgileri taşınabilir. Ayrıca hasta ile ilgili çok önemli istatistiksel bilgiler ya da fizik tedaviye ne zaman gitti gibi bilgiler gerçek zamanlı olarak işlenebilir. Bunun neticesinde doktor ya da hemşireler hasta ile ilgili anlık bilgilere erişebilir. Şekil 2.9'da bunun tipik bir örneği görülmektedir.



Şekil 2.9. Hasta takibinde RFID kullanılması

RFID'nin hastanelerde kullanımı ile ilgili diğer bir uygulama tehlikeli hastalıklar taşıyan hastaların yalıtılmasıdır. Bu uygulama karşımıza, 2002 yılında Uzak doğuda ortaya çıkan ve binlerce kişinin ölümüne neden olan SARS (Severe Acute Respiratory

Syndrom) virüsüne yakalanmış hastaların yalıtılmasında çıkar. RFID'nin burada kullanılması ile virüs taşıyan kişilerin, alet ya da cihazların hastane yönetimi tarafından takibi sağlanmıştır. Hastane içerisinde bu şekilde bir enfeksiyon kontrol merkezi kurulmuştur (26).

RFID ile otomatik ödeme

RFID müşterilerin otomatik ödeme yapabilmesi için kartlara monte edilir. Genellikle köprü ve tünellerde kullanılır. Şu anda Türkiye'de otoyol otomatik yol geçişlerinde benzer sistemler kullanılmaktadır.

Bunun dışında benzin alımı sırasında bu kartlar kullanılabilir. Bu şekilde ödemenin daha hızlı bir şekilde yapılması sağlanır.

RFID ile müşteri takibi

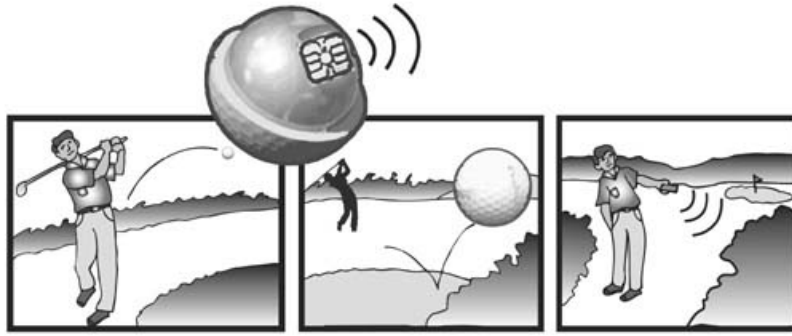
RFID ile müşterilerin takip edilmesi sürekli müşterilerin ödemelerini yapmak, belli servisleri sağlamak için kullanılabilir. Bu tarz uygulamalara kayak pistlerinde ve stadyumlarda rastlanmaktadır. Uygulama esnasında genellikle smart kart şeklinde RFID taşıyıcıları kullanılmaktadır.

Şu anda Barcelona'da bulunan Baja Plaj Kulübünün VIP üyeleri ise smart kart şeklinde RFID taşıyıcısı yerine deri altına yerleştirilen RFID taşıyıcıları kullanılmaktadır. Bu uygulamada RFID kartı taşınmak yerine sol kolda bulunan pazu altına lokal anestezi yapılarak enjekte edilir. Bu kartlar müşterinin içeri girişini çabuklaştırmak dışında ödeme işlemlerinde de kullanılmaktadır (22).

RFID'nin sportif aktivitelerde kullanılması

RFID taşıyıcıları, sportif aktivitelerde zaman ölçmede de kullanılabilir. Bu tarz uygulamalar son birkaç yıldır önemini artırmaktadır.

RFID'nin diđer bir uygulaması ise golf sporundadır. Burada iki tane kolay uygulama karřımıza ıkar. Birincisi Őekil 2.10.'da grldđ gibi kayıp golf toplarının bulunmasıdır. RFID tařıyıcısı golf topunun ierisine yerleřtirilir. Golf topu kaybolduđunda elde tařıyan bir RFID okuyucusu yardımıyla bulunabilir. Bu uygulama oyunu hızlandırmak ve yanlış vuruřları engellemek dıřında kayıp topların bulunmasını sađlayarak ekonomik ynden kazanç sađlar. İkinci uygulama ise golf topunun ne kadar mesafede olduđunu belirlemede kullanılır. Bu iřlem stratejik noktalara yerleřtirilen okuyucular yardımıyla gerekleřtirilir (22).

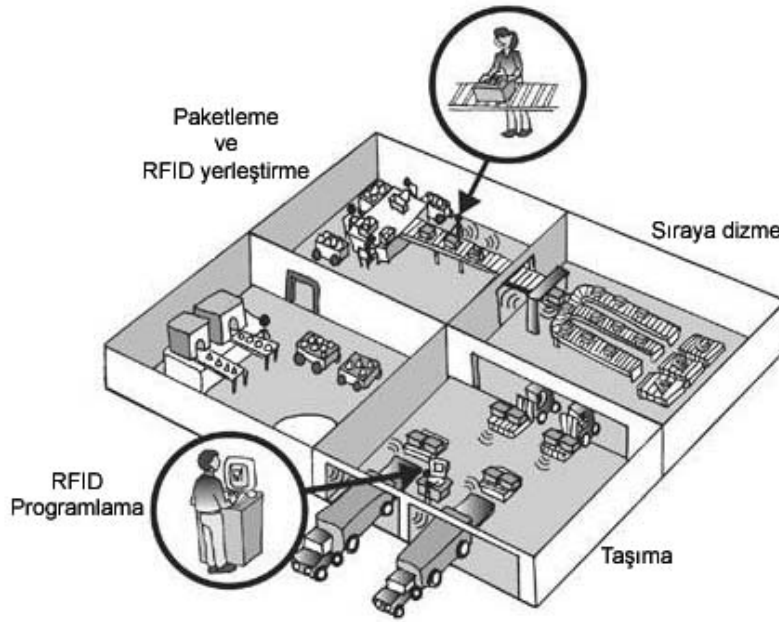


Őekil 2.10. Sportif aktivitelerde RFID kullanılması

RFID ile retim kontrol

Fabrikalarda ve depolarda envanter kontrol yapmak iin kullanılan izleme cihazları el ile yapılan iřleri ve iřlemlerin maliyetini azaltır. Bu izleme iřlemi RFID teknolojisi kullanılarak da yapılabilir.

Őekil 2.11'de basit bir paketleme sisteminin bir fabrikada uygulaması grlmektedir. Bu uygulamada RFID tařıyıcılıları paketleme esnasında malzemelerin zerine yerleřtirilir. Tařıma sırasında ise malzemelerin zerine yerleřtirilen tařıyıcılar RFID okuyucusu vasıtasıyla programlanır.



Şekil 2.11. Üretim kontrolünde RFID kullanımı

RFID ile bagaj izleme

Havaalanlarında bagaj izlemede bagajların sıralanması ve doğru yolcuya bagajın ulaştırılması işlemleri yapılır. Bu işlemler sırasında her bir bagajın etiketlenmesi gerekir. Etiketleme işlemini gerçekleştirmek için ise elektronik olarak işlenebilen sistemler kullanılır. Bugüne kadar en yaygın kullanılan teknoloji barkodlardı fakat RFID barkodlar gibi düşük miktarda bilgi taşımadığından ve okunması sırasında sorunlarla karşılaşılmadığından dolayı hava alanlarında tercih sebebi olmuştur (27).

RFID taşıyıcıları, yolcuların bagaj kontrolü sırasında devreye girer. Bu işlem ile taşıyıcı bant üzerinde giden bagajların sıralanması ve uygun müşteriye yönlendirilmesi daha basit hale getirilir. RFID ayrıca el ile yapılan işlemleri azaltır. RFID taşıyıcıların barkoddaki gibi tam okuyucu ile karşılıklı gelmesine gerek olmadığından bagajların yerinin değiştirilmesine de gerek kalmaz.

RFID kullanılarak hayvan kimliklendirilmesi

Elektronik tanıma sistemleri son 20 yıldır hayvan stoklarının takibinde kullanılmaktadır. RFID'nin otomatik besleme ve üretim kullanımlarına ek olarak RFID kullanılarak ulusal çapta hayvanların kimliklendirilmesi yapılmaktadır. Bu işlemi gerçekleştirmek için 1996 yılında 11784 ve 11785 sayılı ISO standartları çıkarılmıştır.

RFID kullanılarak hayvanların kimliklendirilmesinde, taşıyıcılar hayvanlara tasma ve küpe şeklinde takılarak, vücuda enjekte edilerek ya da gerdanlık boşluğuna yerleştirilerek kullanılabilir. EK 1'de görülen tasma şeklindeki taşıyıcılar bir hayvandan diğerine kolaylıkla taşınabilir. Bu tarz uygulamalar genellikle üreticiler tarafından kullanılır ve hayvanların otomatik olarak beslenmesi ya da süt miktarlarının ölçülmesinde etkilidir.

Kulak küpesi şeklindeki taşıyıcılar, standart barkod şeklindeki hayvan küpelerinden daha ucuzdur ve okuma mesafesi bir metredir.

Vücuda enjekte edilen taşıyıcılar ise 10 yıldır kullanılmaktadır. Bu taşıyıcılar hayvanın derisinin altına özel bir alet yardımıyla enjekte edilir. Bununla hayvanın vücudu ile okuyucu arasında bağlantı sağlanmış olur. Taşıyıcı ancak bir cerrahi operasyonla vücuttan çıkartılabilir. Vücuda enjekte edilen bu taşıyıcılarla ulusal çapta hayvan kimliklendirilmesi daha güvenli hale gelir.

3. İNTERNET ÜZERİNDEN HAYVAN KİMLİKLENDİRME UYGULAMASI

Bu uygulamanın gerçekleştirilebilmesi için öncelikle ilişkisel veritabanının mantıksal tasarımı yapılmıştır. Yapılan bu mantıksal veritabanı tasarımı MySQL veritabanına taşınmış daha sonra bu veritabanının PHP programlama dili ile bağlantısı yapılarak uygulamanın internet üzerinde çalışması sağlanmıştır. RFID okuyucu ile PHP programlama dili ile çalışan web uygulamasının haberleşmesi için Delphi programlama dilinden faydalanılmıştır. Neticede RFID taşıyıcıdaki kulak numarası ile veritabanına girilen kulak numarasının birebir eşleştirilmesi sağlanarak büyükbaş hayvanların internet üzerinden kimliklendirilmesi gerçekleştirilmiştir.

3.1. Veritabanının Mantıksal Tasarımı

Veritabanının mantıksal tasarımı gerçekleştirilirken ER (Entity Relationship- Nesne İlişki) olarak adlandırılan veritabanı modelinden faydalanılmıştır. ER veritabanı modeli ve bu modelle kullanılan ERD (Entity Relationship Diagram- Nesne İlişki Diyagramları) veritabanı tasarımında ve sistem analizinde oldukça yaygın olarak kullanılmaktadır. ERD modellenen veritabanının grafiksel gösterimidir. ERD diyagramlarında nesnelere, nesnelere özellikleri ve nesnelere aralarındaki ilişkiler olmak üzere üç temel eleman vardır. Nesnelere ERD diyagramlarının en temel elemanıdır. Farklı nesnelere özellikleri kullanılarak nesnelere arasında ilişkiler kurulur. Nesnelere birbiri ile ilişki kurmasını sağlayan özelliklerine de anahtar denir (28).

Bu çalışmada aşı, bakanlık, hastalık, hayvan, ilçetarı, iltarı ve işletme olmak üzere toplam yedi temel nesne kullanılmıştır. Bu nesnelere kullanılarak bir bakanlıktaki personelin, il tarım müdürlüklerinin ya da ilçe tarım müdürlüklerinin kendi yetkileri dahilinde bir hayvan ya da hayvan işletmesi ile ilgili bilgilere erişmesi hedeflenmiştir.

Çalışmada hayvanların kontrol edildiği bir tarım bakanlığının olduğu kabul edilmiştir. Bu tarım bakanlığının erişmesi gereken birden çok il tarım müdürlüğü vardır. Bu yüzden bakanlık nesnesi ile iltarım nesnesi arasındaki ilişki Şekil 3.1'deki gibidir. Bu iki nesnenin aralarındaki ilişki ulkekodu nesne özelliği kullanılarak kurulmuştur.



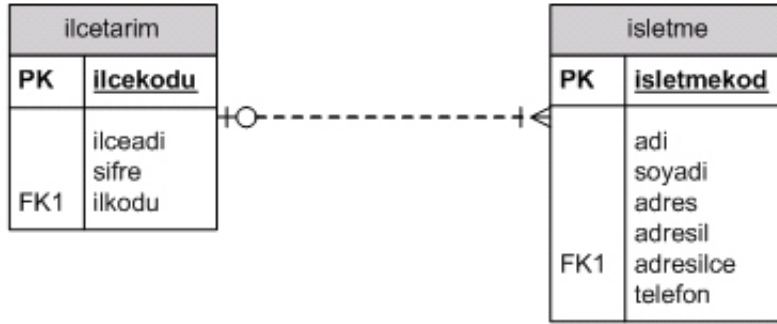
Şekil 3.1. Bakanlık ve iltarım nesneleri arasındaki ilişki

Her bir il tarım müdürlüğüne bağlı ilçe tarım müdürlükleri vardır. Bir il tarım müdürlüğünün kendisine bağlı ilçe tarım müdürlüklerine erişmesi için ilkodu anahtar nesne özelliğinin kullanılması gerekir. Bu özellikten faydalanılarak bir il tarım müdürlüğüne bağlı ilçe tarım müdürlüklerine erişilebilir. Buna göre iltarım, ilçetarım nesneleri ve aralarındaki ilişki Şekil 3.2'deki gibidir.



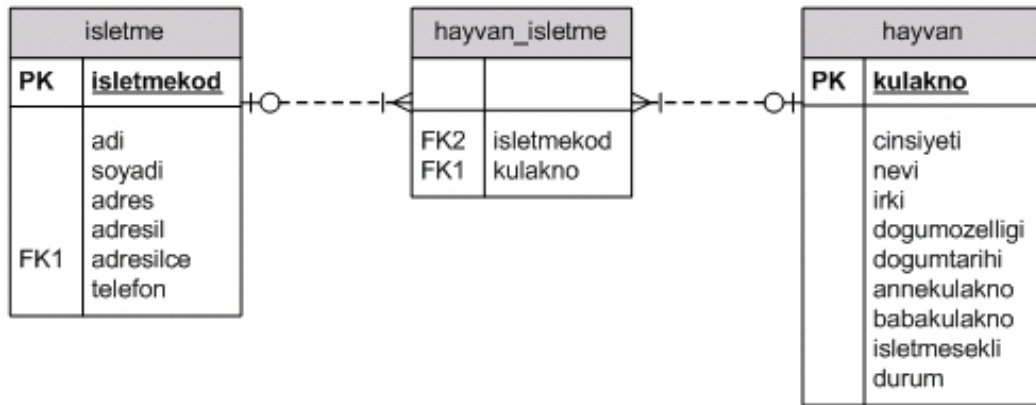
Şekil 3.2. İltarım ve ilçetarım nesneleri ve aralarındaki ilişki

Her bir ilçe tarım müdürlüğüne bağlı işletmeler vardır. Bu işletmeler birbirinden kendilerine verilen işletme kodları ile ayrılır. Bundan hareketle anahtar özelliği isletmekodu olan işletme nesnesi oluşturulmuştur. Bu nesne ile ilçetarım nesnesi arasında ilcekodu özelliği kullanılarak ilişki kurulur. İlçetarım, işletme nesneleri ve aralarındaki ilişki Şekil 3.3'de görülmektedir.



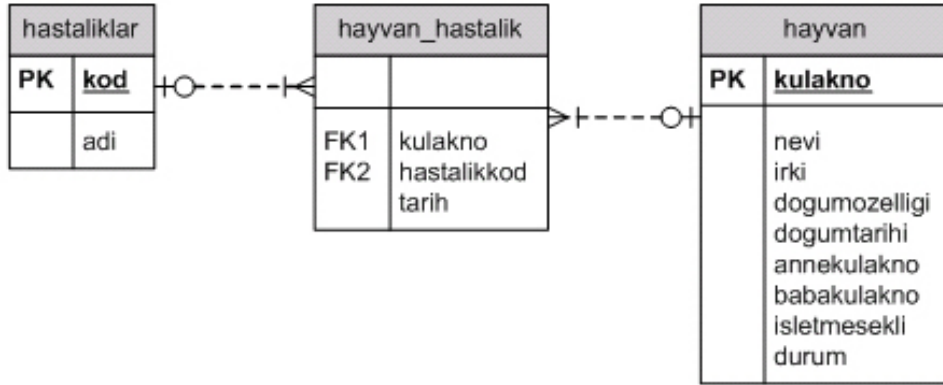
Şekil 3.3. İlçetarım ve işletme nesneleri ve aralarındaki ilişki

Bir ilçedeki her bir işletmede bir ya da birden fazla hayvan yetiştirilir. Her bir hayvan benzersiz olan kulak numarası ile birbirinden ayrılır. Bundan hareketle anahtar özelliği kulak numarası olan hayvan nesnesi oluşturulmuştur. Bu nesne ile işletme nesnesi arasında ilişki kurmak için hayvan ve işletme nesneleri arasında her iki nesnesinde benzersiz anahtar özelliklerini taşıyan hayvan_ışletme nesnesi oluşturulmuştur. Hayvan, hayvan_ışletme, işletme nesneleri ve bunların aralarındaki ilişkiler ise Şekil 3.4'deki olacaktır.



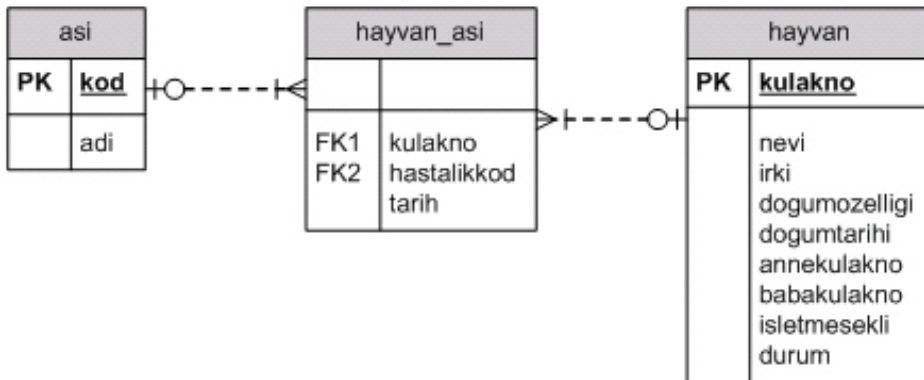
Şekil 3.4. Hayvan, hayvan_ışletme, işletme nesneleri ve aralarındaki ilişkiler

İşletme bazında erişimi sağlanan her bir hayvanın geçirdiği hastalıkların sisteme girilmesi için hastalıklar nesnesi oluşturulmuştur. Hayvan nesnesi ile hastalıklar nesnesinin ilişkilendirilmesi için ise her iki nesnenin anahtar özelliklerini taşıyan hayvan_hastalık nesnesi kullanılmıştır. Hayvan, hastalıklar, hayvan_hastalık nesneleri ve aralarındaki ilişkiler Şekil 3.5'de görülmektedir.



Şekil 3.5. Hayvan, hastalıklar, hayvan_hastalık nesneleri ve aralarındaki ilişkiler

Benzer şekilde her hayvan ile ilgili aşılar da tablolarda tutulmuştur. Burada öncelikle aşı nesnesi oluşturulmuştur. Daha sonra bu nesne ile hayvan nesnesinin anahtar nesnelere olan kulakno ve kod birleştirilerek hayvan_aşı nesnesi elde edilmiştir. Hayvan, aşı, hayvan_aşı nesneleri ve aralarındaki ilişki Şekil 3.6'da açıkça görülmektedir.



Şekil 3.6. Hayvan, aşı, hayvan_aşı nesneleri ve aralarındaki ilişkiler

Oluşturulan bu temel nesnelere dışında hayvan nesnesine veri kaynağı olacak cinsiyet, nevi, ırk, doğumozelligi, işletmeşekli ve durum nesnelere de oluşturulmuştur. Bu nesnelere ve hayvan nesnesi arasında birbirine ilişki oluşturulmuştur.

3.2. MySQL veritabanına mantıksal tasarımın taşınması

MySQL kolay kullanılabilir, kolay ayarlanabilir ve kolay yönetilebilir bir veritabanı uygulamasıdır. Unix türevi çok değişik işletim sistemleri altında ve Windows altında çalışabilir. Ayrıca MySQL tabanlı uygulamalar birçok programlama dilinde geliştirilebilir. MySQL veritabanı uygulamasına, sıklıkla bir web sunucu üzerinden çalıştırılan programlama dilleri kullanılarak erişilir (29).

3.2.1. Linux üzerinde MySQL veritabanının yüklenmesi

MySQL veritabanı Linux sistemler üzerine değişik yöntemler kullanılarak yüklenebilir. Bu uygulamada bütün Linux sistemlerde geçerliliği olan kaynak koddan derleme yöntemi üzerinde durulmuştur. Öncelikle Linux kurulu sisteme www.mysql.com sitesinden sıkıştırılmış olan kaynak kodun son halinin yüklenmesi gerekir. Bu işlemden sonra mysql kullanıcı grubu “addgroup mysql” komutu ile açılır. Daha sonra bu gruba mysql kullanıcısı “adduser -g mysql mysql” komutu ile dahil edilir ve sırasıyla aşağıdaki işlemler gerçekleştirilir:

1. Sıkıştırılmış olarak gelen paket bir klasöre açılır, açılan klasöre geçilir.

```
tar -zxvf mysql-5.0.1.tar.gz  
cd MySQL-5.0.1
```

2. ./configure çalıştırılır. Bu komut birçok seçeneğe sahiptir. Örneğin prefix=/usr/local/mysql ile MySQL'in ilgili dizine kurulması sağlanır. Ayrıca with-charset=latin5 ile de karakter setinin Türkçe olması sağlanır .

```
./configure --prefix=/usr/local/mysql --with-charset=latin5
```

3. Make komutu çalıştırılır. Bu komut ile derleme işlemi başlatılır.

4. Make install komutu ile de derlenerek oluşturulan dosyalar ikinci adımda belirtilen yere kopyalanır.

5. MySQL ayar dosyası /etc dizini altına kopyalanır.

```
cd support-files/my-medium.cnf /etc/my.cnf
```

6. Başlangıç veritabanı oluşturulur.

```
/usr/local/mysql/bin/mysql-install-db --user=mysql
```

7. MySQL veritabanının çalıştırılıp, durdurulmasını sağlayan script /etc/rcd/init.d altına kopyalanıp gerekli ayarlar yapılır.

```
cp support-files/mysql-server /etc/rcd/init.d/mysql
```

```
cd /etc/rc.d/init.d
```

```
chmod 755 mysql
```

```
ln -s mysql ../rc0.d/K85mysql
```

```
ln -s mysql ../rc3.d/S85mysql
```

```
ln -s mysql ../rc5.d/S85mysql
```

```
ln -s mysql ../rc6.d/K85mysql
```

8. Dizin hakları düzenlenir.

```
cd /usr/local/mysql
```

```
chown -R root .
```

```
chown -R mysql var
```

```
chown -R mysql
```

9. Daha sonra MySQL veritabanı sunucusu başlatılır.

```
/etc/rc.d/init.d/mysql start
```

MySQL veritabanı sunucuna bağlanabilmek için komut sisteminde mysql çalıştırılır. Bu komut çeşitli parametrelere sahiptir. Bu parametreler boş olabilir fakat genelde aşağıdakine benzer bir kullanım söz konusudur.

```
mysql -h bilgisayar adı -u kullanıcı adı -p
```

Sıklıkla kullanılan parametreler ve anlamları şunlardır:

-h bilgisayar adı:

Bağlanılacak sunucu bu parametre ile belirlenir. Sunucu kurulan makine üzerinde çalışılıyorsa bu parametre atlanabilir.

-u kullanıcı adı:

MySQL kullanıcı adıdır. Unix kullanıyorsanız ve MySQL kullanıcı adınız Unix kullanıcı adı ile aynı ise bu parametre atlanabilir.

-p parametresi:

Bu parametre ile MySQL'in şifre sorması sağlanır. İstenirse bu parametreden sonra şifrede girilebilir. Fakat şifre burada açık bir şekilde gözükecektir.

Başlangıçta mysql veritabanını yöneten root kullanıcısının şifresi boştur. Bu da güvenlik açığına neden olur. Bunu önlemek için aşağıdaki işlemler gerçekleştirilir.

```
mysql -u root
```

```
mysql>update mysql.user SET Password=PASSWORD('şifre')
```

```
-> where user='root'
```

```
mysql>FLUSH PRIVILEGES
```

```
mysql>exit
```

3.2.3. MySQL’de büyükbaş hayvan veritabanının oluşturulması

Yeni bir veritabanı oluşturmak için mysql komutu kullanılarak veritabanına bağlanılır. CREATE DATABASE sql ifadesi ile veritabanı oluşturulur.

```
mysql> CREATE DATABASE buyukbas
```

3.2.4. MySQL’de büyükbaş hayvan veritabanı tablolarının oluşturulması

Veritabanı oluşturulduktan sonra artık tablolar oluşturulabilir. Öncelikle USE veritabanı adı komutu ile veritabanı seçimi yapılır ve daha sonra CREATE TABLE ifadeleri ile tablolar oluşturulur. Bu uygulama ile aşı, bakanlık, cinsiyet, dogumozelligi, durum, hastalık, hayvan, hayvan_aşı, hayvan_hastalık, hayvan_isletme, ilcetarım, iltarım, irk, isletme, isletmesekli ve nevi olmak üzere 16 tane birbiri ile ilişkili tablo ERD diyagramlarından faydalanılarak oluşturulmuştur. Bu tablolardan en önemlisi olan büyükbaş hayvanların bilgisinin tutulduğu hayvan tablosunun CREATE TABLE sql ifadesi ile oluşturulmuş hali aşağıdaki gibidir.

```
CREATE TABLE hayvan (
    kulakno char(10) NOT NULL,
    cinsiyeti char(1) default NULL,
    nevi char(1) default NULL,
    irki char(2) default NULL,
    dogumozelligi char(1) default NULL,
    dogumtarihi date default '0000-00-00'
    annkulakno char(16) default NULL,
    babakulakno char(16) default NULL,
    isletmesekli char(1) default NULL,
    durum char(1) default NULL,
    PRIMARY KEY('kulakno')
);
```

Bu SQL ifadesinde char(10), date, varchar(10) gibi ifadeler kullanılabilir. Bu ifadeler tablo alanlarının tanımlanmasında işe yarar. Bir veritabanı oluşturulduğunda MySQL'e hangi veri tiplerinin kullanılacağını bildirilmesinde bu ifadeler kullanılır MySQL'de kullanılan veri tipleri şunlardır:

CHAR(uzunluk): Sabit uzunlukta string karakteri tanımlamada kullanılır.

VARCHAR(uzunluk) : Değişken uzunlukta string karakteri tanımlamada kullanılır. En uzun string uzunluk parametresi ile belirlenir. Bu uzunluk ise 1 ile 255 arasında olmalıdır.

TEXT : Maksimum uzunluğu 64KB olan değişken uzunlukta string tanımlamada kullanılır.

INT(uzunluk) : -2147483648 ile 2147483647 arasında değer alan tamsayıdır. Uzunluk parametresi ile sayının rakam adedi belirlenir. Örneğin uzunluk parametresi 4 olduğunda -999 ile 9999 arasında sayı girilebilir.

INT(uzunluk) UNSIGNED: 0 ile 4294967295 arasında değer alan tamsayıdır. Uzunluk parametresi ile sayının rakam adedi belirlenir. Örneğin uzunluk parametresi 4 olduğunda 0 ile 9999 arasında sayı girilebilir.

DECIMAL(uzunluk,rakam adedi) : Uzunluk ile ondalık sayının toplam uzunluğu, rakam adedi ile de ondalık sayının noktadan sonra kullanılacak rakam adedi belirlenir. Örneğin 12.34 sayısı için uzunluk 5, rakam adedi 2'dir.

DATE : Yıl, ay, gün kullanılan tarih şeklindedir. Tarih YYYY-MM-DD şeklinde gösterilir. İstenirse bu gösterim şekli sql komutları kullanılarak değiştirilebilir.

TIME : Saat, dakika ve saniye olarak zaman veri tipidir. Zaman HH:MM:SS şeklinde gösterilir.

DATETIME : Zaman ve tarihin birlikte gösterildiği veri tipidir. Bu veri tipi, YYYY-MM-DD HH:MM şeklinde gösterilir.

Veri tipleri temel alınarak diğer tablolar için oluşturulan CREATE TABLE ifadeleri EK-2’de verilmiştir.

3.3. Apache Web Sunucu Desteği

Apache web sunucusunun en güzel özelliği birçok bilgisayar platformunda kullanılabilir olmasıdır. Başlangıçta Apache Unix tabanlı bir web sunucusuydu. Fakat bu durum geçerliliğini yitirmiştir. Apache, şu anda sadece Unix tabanlı sistemlerde çalışmak dışında Windows 2000/NT/9x gibi Windows tabanlı işletim sistemlerinde hatta OS/2’ de bile çalışmaktadır (30).

Apache web sunucunun kurulumunda, yine MySQL sunucusunda olduğu gibi Linux üzerinde kaynak kodla derleme metodu kullanılmıştır. Apache kaynak kodunun en son sürümü www.apache.org sitesinden indirilmiş eklenebilir modül desteği verilerek PHP ile birlikte derlenmesi sağlanmıştır. Apache web sunucusu aşağıdaki adımlara göre kurulmuştur:

1. Sıkıştırılmış kaynak kodu bir klasöre açılır, açılan klasöre girilir.

```
tar --zxvf http-2.0.54.tar.gz
cd http-2.0.54
```

2. ./configure çalıştırılır. Bu komut birçok seçeneğe sahiptir. Örneğin prefix=/www ile Apache’nin kurulacağı klasör, enable-so ile eklenebilir modül desteği verilir.

```
./configure --prefix=www --enable-so
```

3. Make komutu çalıştırılır. Bu komut ile derleme işlemi başlatılır.

4. Make install komutu ile derlenerek oluşturulan dosyalar ilgili klasöre kopyalanır.

Bütün bu aşamalardan sonra /www dizini altına Apache web sunucusu kurulmuş olur.

3.3.1. PHP ile Apache'nin derlenmesi

PHP, server tarafında çalışan dinamik web sayfaları geliştirmek için kullanılan bir programlama dilidir. PHP, Hypertext Preprocessor kelimelerinden gelmektedir. C, Java ya da Perl programlama dillerini anımsatan bir yapıya sahip olduğu için bu tarz programlara aşina programcılar tarafından kolay öğrenilir. Ayrıca açık kaynak kodlu olarak tüm işletim sistemlerinde çalıştırılabilir (31).

PHP'yi Apache ile beraber çalıştırabilmek için eklenebilir modül olarak derlenmesi gerekir. Daha önceden kurulan Apache ve MySQL ile birlikte kaynak koddan derlenmek istenen PHP aşağıdaki adımları izlenerek kurulur:

1. Sıkıştırılmış dosya bir klasöre açılır, açılan klasöre gidilir.

```
tar -zxvf php-5.0.4.tar.gz
cd php-5.0.4
```

2. ./configure'ye parametre girilir. prefix=/www/php ile kurulacak klasör, with-apxs2=/www/bin/apxs ile Apache modülü, with-mysql=/usr/local/mysql ile MySQL dosyalarının yeri belirlenir. with-config-file-patch=/www/conf parametresi ile de PHP'nin ayar dosyası olan php.ini'nin yeri girilir.

```
./configure -- prefix=/www/php \
-- with-apxs2=/www/bin/apxs \
-- with-mysql=/usr/local/mysql \
-- with-config-file-path=/www/conf
```

3. Make komutu çalıştırılır. Bu komut ile derleme işlemi başlatılır.
4. Make install komutu ile derlenerek oluşturulan dosyalar ilgili klasöre kopyalanır.
5. PHP ayar dosyası ilgili dizine kopyalanır.

```
cp php-ini.dist /www/conf/PHP.ini
```

Apache'nin PHP dosyalarını PHP derleyicisi ile çalıştırılabilmesi için ona bu direktifin verilmesi gerekir. Apache'de böyle direktifler httpd.conf dosyasına yazılır. (31) Bu dosya buradaki kurulumunda /www/conf dizini altındadır. Apache web sunucusuna PHP desteği vermek için httpd.conf dosyasına aşağıdaki gibi bir yazılır.

```
AddType application/x-httpd-php .php
AddType application /x-httpd-php-source .phps
```

Bu işlemden sonra Apache Web Sunucusu aşağıdaki komut kullanılarak çalıştırılır.

```
/www/bin/apachectl start
```

Kurulumun test edilebilmesi için Apache Web Sunucusunun doküman klasörü olan htdocs klasöründe test.php dosyası oluşturulur (Bu kurulumda /www/htdocs dizini) ve bu dosyanın içine aşağıdaki PHP kodu girilir.

```
<?php
    phpinfo ();
?>
```

Bu dosya Internet Explorer ve Netscape gibi bir internet tarayıcılarında <http://IPadresi/test.php> şeklinde çalıştırıldığında PHP kurulumu ile ilgili detaylı bilgilerin verildiği bir sayfa görüntülenir.

3.4. PHP ile MySQL veritabanının web tabanlı olarak haberleştirilmesi

PHP ile hazırlanan arayüzle bakanlık personeli, il tarım müdürlükleri ve ilçe tarım müdürlükleri giriş yapabilir. RFID ile bilginin taşınması işlemini ise sadece ilçe tarım müdürlükleri gerçekleştirebilir.

PHP program dosyaları

PHP program dosyaları hem bankalık personeli, hem il tarım müdürlükleri ve hem de ilçe tarım müdürlükleri için ortak hazırlanmıştır. Program kaynak kodları EK-4.'de detaylı olarak gösterilmiştir. EK-4'de verilen program kaynak kodlarının yaptığı işler şöyledir:

common.inc.php:

Bu program dosyası, bütün dosyalar tarafından kullanılan ortak fonksiyonları üzerinde taşır. HTMLbasla() ve HTMLbitir() fonksiyonları ile HTML başlangıç ve bitiş tagları oluşturulur.

connecting() ve pconnecting() fonksiyonları veritabanına bağlantı için kullanılır. Connecting fonksiyonunun birinci, ikinci, üçüncü ve dördüncü satırında, \$hostname değişkeni ile bağlanılacak sunucu, \$username değişkeni ile veritabanı kullanıcısı, \$password değişkeni ile veritabanı kullanıcısının parolası ve \$database değişkeni ile bağlanılacak veritabanı belirlenir. mysql_connect fonksiyonu ile lokal makine üzerine root kullanıcısı ve deneme şifresi ile bağlantı sağlanır. Eğer bağlantı başarılı ise mysql_selectdb() fonksiyonu ile büyükbas veritabanı seçilir. Bu fonksiyonda başarıyla çalıştırdıktan sonra artık veritabanına sorgu gönderilebilir.

index.html:

Bu HTML dosyası başlangıçta kullanıcı adı ve şifrenin istendiği dosyadır. Bu dosya bakanlık, il tarım ve ilçe tarım için hemen hemen aynıdır. HTML formu doldurulup POST metodu ile `giris.php`'ye gönderilir.

giris.php:

`Giris.php` bakanlık, il tarım ve ilçe tarım müdürlükleri için sql sorguları ve session değişkenleri dışında hep aynıdır. Eklerde ilçe tarım müdürlükleri için gösterilen `giris.php` dosyasında öncelikle `index.html` dosyasından sisteme giriş butonuna basılıp basılmadığı kontrol edilir. Bu butona basıldı ise aşağıdaki SQL ifadesi çalıştırılır.

```
select ilcekodu, iladi from ilcetarım where ilcekodu='1806' and sifre='deneme'
```

Yukarıdaki SQL ifadesi örnek olarak verilmiştir. Bu sorgu ile ilçe kodu 1806 olan ve şifresi deneme olan kayıt getirilir. Kayıt sayısı `mysql_num_rows()` fonksiyonu ile döndürülür ve bu değer `$numresult` değişkenine aktarılır. Eğer bu değişkenin değeri sıfır ise kullanıcı adı ve şifresi hatalıdır. Bu yüzden sisteme giriş engellenir.

`$numresult` değişkeninin değeri sıfırdan farklı ise ilçe kodu ve şifre doğrudur ve sisteme giriş yapılır. Daha sonra oturum nesnesi olarak adlandırılan benzersiz bir numarayı her oturumda açmanızı sağlayan session nesnesinden faydalanılır. Session ile güvenliği sağlamak için `giris.php`'den sonra açılan her sayfada oturum, `session_start()` fonksiyonu ile başlatılır. Daha sonra session'ın kayıtlı olup olmadığı `session_is_registered()` fonksiyonu ile test edilir. Bu yöntemle kullanıcı sistemden çıkana kadar sistemin güvenliği sağlanır.

Session ile oturum açıldıktan sonra ilçe koduna ve il koduna diğer tüm sayfalar tarafından erişilebilmesi için `session_register()` fonksiyonu ile il ve ilçe kodları kayıt edilir. Yeni bir sayfaya geçilmesinde `header()` fonksiyonu kullanılır.

ana.php:

Ana.php dosyası ile sağ ve sol olmak iki farklı pencereye dosyalar yüklenir. Başlangıçta bu pencerelerden sol tarafta olana solmenu.php ve sağ tarafta olana bakanlıktaki personel giriş yaptıysa il listesinin, il tarım müdürlüğü personeli giriş yaptıysa ilçe listesinin, ilçe tarım müdürlüğü personeli giriş yaptıysa işletme listesinin getirildiği dosya yüklenir.

iltarim.php:

Bu program dosyası parametre olarak iltarim.php?ulkekodu=TR linkindeki ulkekodunu kabul eder. \$HTTP_GET_VARS ile bir HTML formu ile yapılan herhangi bir GET isteği bir diziye atılır. Bu uygulamada ulkekodu \$HTTP_GET_VARS ['ulkekodu'] değişkeni ile alınır. Ayrıca program dosyası içerisinde ilçe tarım müdürlüğü listesi için ilcetarim.php program dosyasına il kodu parametresini GET değişkeni olarak gönderen yapılar da vardır.

Program dosyasının başında session_start komutu ile oturum başlatılarak daha önceden yaratılan oturum nesnelерinin kontrolü yapılmasına zemin hazırlanır. Bu kontrol session_start()'tan hemen sonra session_registered() komutu ile yapılır.

ilcetarim.php:

Bu program dosyası parametre olarak ilcetarim.PHP?ilkodu=06 linkindeki il kodunu kabul eder. \$HTTP_GET_VARS['ilcekodu'] ile ilkodu alınır. Daha sonra bu sorguya dönüştürölüp ekrana yansıtılır. Ayrıca bu program dosyası ile işletmelerin listesini getirmek için isletme.php dosyasına ilçe kodu parametresi GET değişkeni olarak gönderilir.

Session_start() ve session_registered fonksiyonları ile oturum kontrolü yapılarak güvenlik sağlanır.

isletme.php:

Bu program dosyasında, ilçe tarım müdürlüğünden gelen gelen ilcekodu parametre olarak kabul edilir. \$HTTP_GET_VARS['ilcekodu'] ile alınan ilçe kodu sorguya dönüştürülerek ilçedeki işletmelerin listesi alınır. Listenin hemen altında yer alan giriş kutusu ve buton ile de isletmekle.php dosyasına POST değişkeni olarak işletme kodu gönderilir.

Bu program dosyası ile ayrıca bir işletmede bulunan hayvanların listesini almak için hayvanliste.php program dosyasına GET değişkeni olarak işletme kodu gönderilir.

isletmekle.php:

isletme.php program dosyası ile POST değişkeni olarak gönderilen işletme kodu \$HTTP_POST_VARS['isletmekodu'] ile alınır. Bu değişken öncelikle bir SELECT sql sorgusu ile sorgulanıp daha önceden işletmenin var olup olmadığı kontrol edilir. Yok ise INSERT sorgusu ile sisteme girilip ekrana yansıtılır. Bu ekranda işletme ile ilgili bilgiler UPDATE sorgusu ile güncellenip ilçede listelenen işletme listesine dönülür.

hayvanliste.php:

isletme.php program dosyası ile GET değişkeni olarak gönderilen işletme kodu \$HTTP_GET_VARS['isletmekodu'] ile alınır. Alınan bu değişken SELECT sql sorgusuna dönüştürülerek işletmede bulunan hayvan listesi görüntülenir. Listenin en altında hayvan eklemeye yarayan bir giriş kutusu bulunur. Bu giriş kutusuna Delphi kullanılarak seri porttan RFID taşıyıcısı ile yada el ile veri girilir. Girilen bu veri hayvanekle.php dosyasına POST değişkeni olarak kulakno parametresini gönderir.

Bu program dosyası ile ayrıca hayvan ile ilgili bilgileri görüp değiştirebileceğimiz, aşı ve hastalık görebileceğimiz hayvandetail.php dosyasına GET değişkeni olarak kulakno gönderilir.

hayvandetay.php:

hayvanliste.php program dosyası ile GET deęişkeni olarak gönderilen kulakno \$HTTP_GET_VARS['kulakno'] ile alınır. Alınan bu deęişken SELECT sql sorgusuna dönüştürülerek hayvan ile ilgili detaylı bilgiler ekrana yansıtılır ve bu bilgiler deęiştirilebilir. Bu ekranda hayvanın geçirdiđi hastalıklar ve aşılar da eklenebilir. Tekrar işletmeye bađlı hayvanların listesine de dönülebilir.

Delphiye bilgi gönderme olayı da bu dosya içindeki hidden deęişken olan kulakno ile yapılır.

hayvanekle.php:

hayvanekle.php program dosyası, hayvandetay.php ile hemen hemen aynıdır. Aradaki tek fark başlangıçta gelen kulakno deęişkeninin SELECT sorgusu ile kontrolünün yapılmasıdır. Bu kontrol ile aynı kulak no'ya sahip hayvan olup olmadığı kontrol edilir. Eđer yoksa sisteme INSERT sorgusu ile giriş yapılır.

hastalikliste.php:

hayvan.php tarafından çağrılan bu dosyada kulak numarası girilen bir hayvanın hastalık listesini alınıp, yeni hastalık girişı yapılabilir.

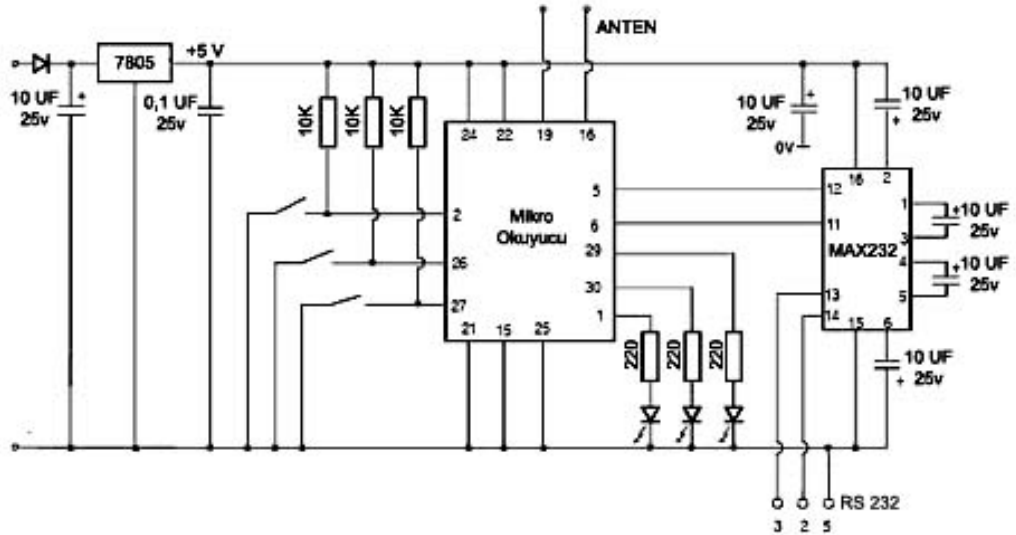
asiliste.php:

hayvan.php tarafından çağrılan bu dosya ile de kulak numarası girilen bir hayvanın aşı listesi alınıp, yeni aşı girişı yapılabilir.

3.5. Delphi ile web uygulamasının birleştirelmesi

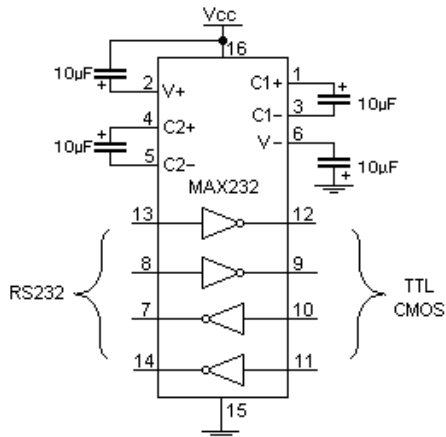
Delphi'nin buradaki rolü seri porttan RFID taşıyıcısı üzerindeki bilgiyi alıp bu bilgiyi web sayfasına aktarmaktır. Bu bilgiyi almada Texas Instruments'in RFID

okuyucu olarak S2000 Mikro okuyucu, RFID taşıyıcısı olarak da yine Texas Instruments'in 128 bit'lik RFID taşıyıcısı kullanılmıştır. Bu uygulama için örnek devre Şekil 3.6'da gösterilmiştir.



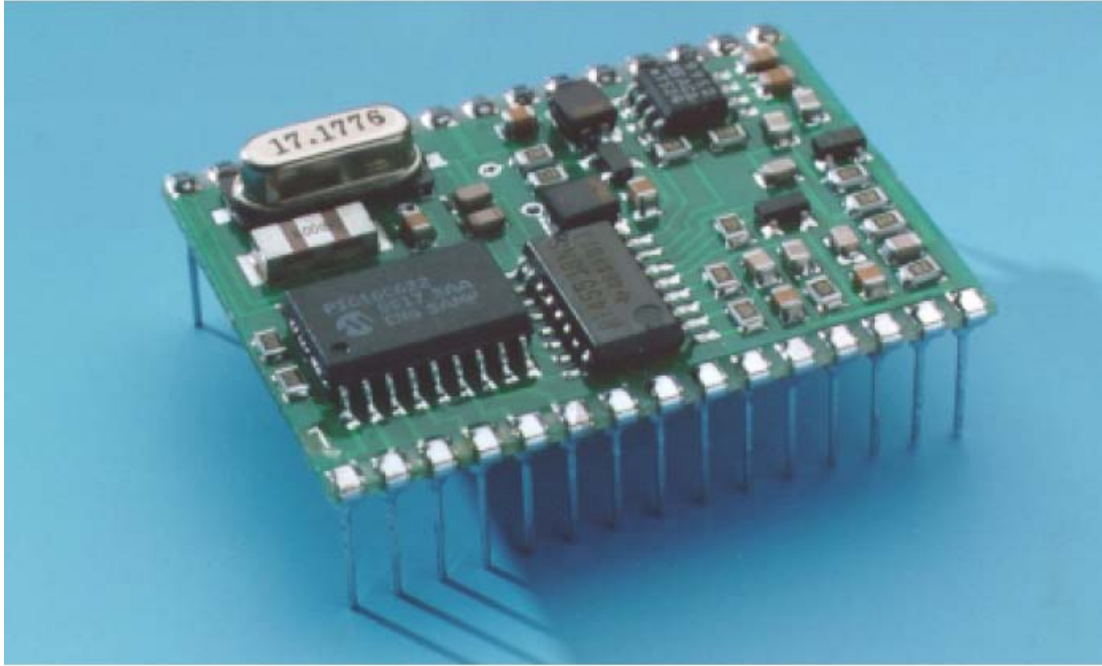
Şekil 3.7. RFID uygulama geliştirme devresi

Bu uygulama devresi 134.2 KHz'de çalışan RFID okuyucu modülü ve bu modüldeki TTL seviyesindeki sinyali bilgisayarın seri portuna çeviren MAX 232 yongasından meydana gelmektedir. MAX 232 yongası 5V'luk bir kaynakla beslenir ve seri porta +10V ya da -10V sinyal gönderebilir. MAX 232 yongasının iç devresi ve tipik uygulama devresi Şekil 3.8'de görülmektedir.



Şekil 3.8. MAX 232 yongasının iç devresi ve tipik uygulama devresi

Resim 3.1’de görülen S2000 RFID mikro okuyucusu 9600 baud hızında çalışır. Seri portla haberleşmede 1 başlangıç biti, 8 veri biti ve 1 bitiş biti kullanılır. Bu haberleşmede eşlenik biti kullanılmaz. RFID okuyucusuna seri porttan 19 sayısı gönderildiğinde o anda yaptığı işlemi ve seri porta veri göndermeyi durdurur ve 17 sayısı kendisine gönderilinceye kadar beklemede kalır. 17 sayısı seri porttan RFID okuyucuna gönderildiğinde daha önceden nerden kalındıysa oradan işleme devam edilir. Ayrıca bu bekleme periyodunda RFID okuyucusu seri porttan gelen komutları kabul eder, seri porttan 17 sayısı gönderildiğinde ise gönderilen bu komutlar çalıştırılır.



Resim 3.1. S2000 Mikro RFID okuyucu modülü

RFID'nin hayvan kimliklendirilmesinde kullanılmasında ISO 11784 ve ISO 11785 standartları kullanılmaktadır. Bu standartlara göre maksimum 64 bit uzunluğundaki RFID taşıyıcılarının kullanılması gerekir. ISO 11784 standardına göre 64 bitlik RFID taşıyıcı bilgi formatı şöyledir:

Çizelge 3.1. ISO 11784 standardına göre RFID taşıyıcı bilgi formatı

Bit Aralığı	Taşınan Bilgi Türü
1	Bayrak biti (Uygulamanın türünü belirlemede kullanılır)
2-15	Rezerve bittir.
16	Data bloğu olduğunda 1 yapılır.
17-26	ISO 3166 standardına göre 10 bitlik ülke kodu
27-90	38 bitlik data bloğu

Bu uygulamada ise 128 bitlik RFID taşıyıcısı kullanılmış ve kulak numaralarının bu taşıyıcılarda kullanılması amaçlanmıştır. ISO11784 ve 11785 standartlarına göre kullanılmayan 2-15 bitleri arasına şehir plaka kodu , 17-26 bitleri arasında Türkiye'nin ISO 3166 standardına göre kodu olan 792 sayısının ikilik sayma sistemindeki karşılığı, 27-38 bitler arasında ise hayvana verilen numara kodlanmıştır. Oluşan bu yeni hayvan numarasına göre 128 bitlik RFID taşıyıcısının bit tanımlaması Tablo 3.2'deki gibidir.

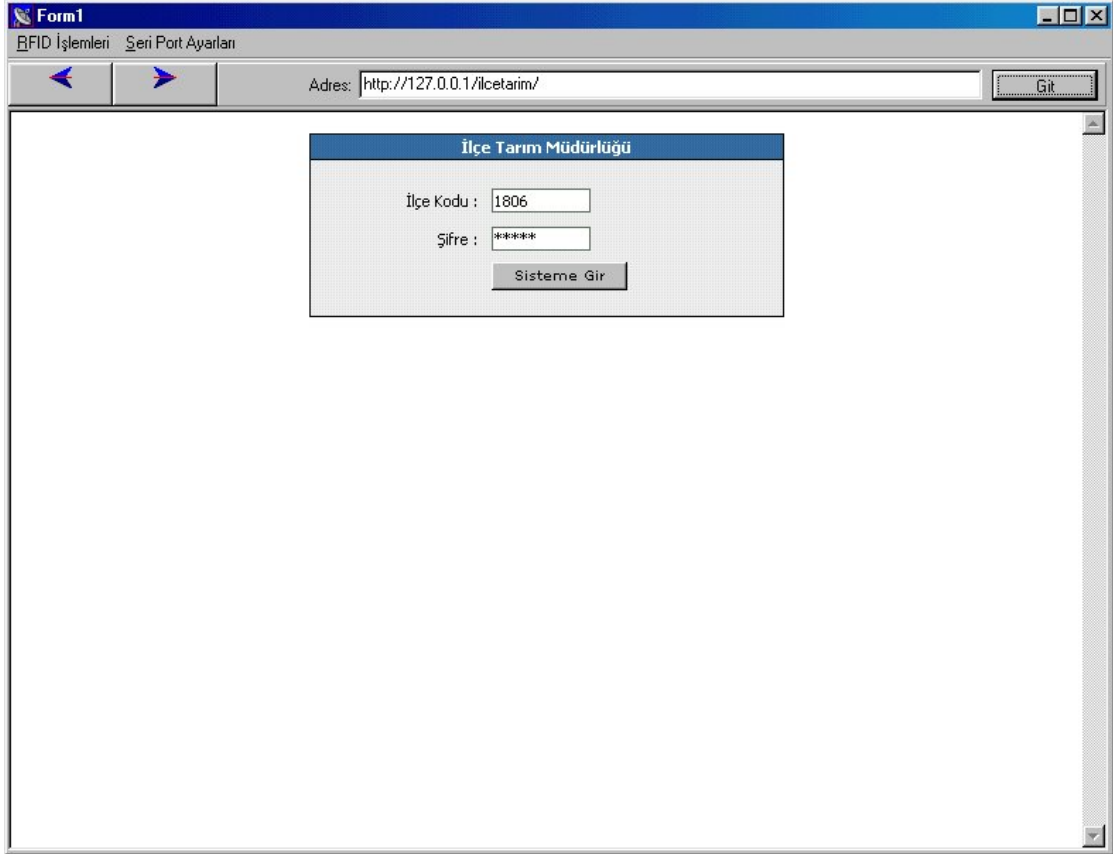
Çizelge 3.2. 128 bitlik RFID taşıyıcısının bit tanımlaması

Bit Aralığı	Taşınan Bilgi Türü
1	Bayrak biti hayvan uygulaması olduğu için 1 yapılır.
2-15	7 bit 0, 8 bit şehir plaka kodu (Örnek 0000000000110000)
16	Data bloğu olduğu için bu bit 1 yapılır.
17-26	ISO 3166 standardına göre Türkiye'nin ülke kodu 792 kodlanır.
27-38	2 bit 0, 36 bit 4'lü gruplanmış ikilik sayı sistemi.

Delphi ile RFID okuyucuyu haberleştirmek için mrp95lib.dll kullanılmıştır. Bu dll dosyasındaki fonksiyonları kullanarak RFID okuyucudan bilgi alınabilir ya da gönderilebilir. Bilgi alma ve gönderme işi web tabanlı olduğu için Delphi'de webbrowser nesnesi kullanılır. Bu nesnenin Oleobject özelliği ile web sayfalarında bulunan form nesnelere erişilebilir. Form nesnelere bu yöntemle RFID taşıyıcısı üzerindeki bilgi işlenebilir ya da RFID taşıyıcısı üzerine bu form nesnelere bilgi alınabilir.

Şekil 3.9'da görülen web sayfası Delphi programlama dilinin nesnesi olan webbrowser'a yüklenir. Bu yükleme işlemi Git butonunun onclick (Git butonuna tıklanması) olayına WebBrowser1.Navigate(Edit1.Text) kodunun yazılması ile

gerçekleştirilir. Yüklenen bu ilk web sayfası ilçe tarım müdürlüklerinin giriş yapacağı ana sayfadır. Bu sayfada ilçe kodu ve şifre yazılarak sisteme girilir.



The image shows a web browser window titled "Form1" with a menu bar containing "RFID İşlemleri" and "Seri Port Ayarları". The address bar shows "Adres: http://127.0.0.1/ilcetarim/" and a "Git" button. The main content area displays a login form titled "İlçe Tarım Müdürlüğü". The form has two input fields: "İlçe Kodu" with the value "1806" and "Şifre" with the value "*****". Below the fields is a "Sisteme Gir" button.

Şekil 3.9. İlçe kodu ve şifre ile sisteme giriş

Şekil 3.9'da görülen web sayfası ile doğru ilçe kodu ve şifresi girildiğinde Şekil 3.10'da görülen webbrowser nesnesine ana.php dosyası yüklenir. Bu web sayfası iki pencereden oluşur. Sol taraftaki pencereye solmenu.php, sağ taraftaki pencereye ise işletmelerin listesinin alındığı işletme.php yüklenir. İşletme.php sayfası vasıtasıyla yeni işletme girişi yapılabilir ya da işletmelere tıklandığında üzerlerinde görülen hayvanlar görüntülenebilir.

Form1

RFID İşlemleri Seri Port Ayarları

Adres: Git

İşletme Listesi
Sistemden Çıkış

İşletme Kodu	Adı Soyadı	Adresi
000006	AHMET DİNÇ	BEŞDUT KÖYÜ NO:9
000002	FATMA TUNA	GEVEREN KÖYÜ NO:7
000010	HACER ÇAKMAK	BÜYÜKBAHÇELİ KÖYÜ NO:1
000003	KEZBAN YILMAZ	KARADİBEK KÖYÜ NO:9
000005	MUSTAFA SÖYLEMEZ	DANABAŞ KÖYÜ NO:12
000008	OSMAN ŞAHİN	AĞZIBÜYÜK KÖYÜ NO:3
000004	REŞAT YILDIZ	BAYANPINAR KÖYÜ NO:8
000009	YUSUF YÜKSEL	BÜYÜKBAHÇELİ KÖYÜ NO:6

İşletme Kodu: İşletme Ekle

Şekil 3.10. İşletme listesinin görüntülenmesi

İşletmelerden herhangi birine tıklandığında Şekil 3.9.'da görülen hayvan listesi alınır. Şekil 3.11'de görülen bu listenin en altında hayvan girişini el ile ya da RFID kullanılarak yapmak için giriş kutusu ve buton bulunur. Hayvan girişini RFID ile yapmak için öncelikle üst tarafta bulunan seri port ayarları menüsünden bağlanılacak port'un seçimi yapılır. Bu seçim yapıldıktan sonra seri port ayarları menüsünde bulunan bağlan seçeneğine tıklanır. Bu şekilde bağlantı gerçekleştirildikten sonra RFID ile sisteme hayvan girişi RFID işlemleri menüsünden RFID taşıyıcı oku seçeneğine tıklanılarak gerçekleştirilir.

Form1
RFID İşlemleri Seri Port Ayarları

Adres:

İşletme Listesi
Sistemden Çıkış

Kulak No	Cinsiyeti	Nevi	Irkı	Doğum Tarihi
TR18000076	HERMAFRODİT	MANDA	ŞAROLE	2001-01-12
TR18000087	ERKEK	MANDA	DAK	2001-07-14
TR18000091	HERMAFRODİT	MANDA	ZAVOT	2001-08-07
TR18000092	HERMAFRODİT	MANDA	GAK	2001-03-14
TR18000094	ERKEK	SIGIR	YERLİ	2001-06-22
TR18000098	ERKEK	SIGIR	BOZ IRK	2001-08-06
TR18000105	DİŞİ	SIGIR	GAK	2001-08-18
TR18000116	ERKEK	SIGIR	YERLİ	2001-01-10
TR18000120	DİŞİ	SIGIR	DAK	2001-04-07
TR18000144	FREE	SIGIR	YERLİ	2001-02-01
TR18000147	DİŞİ	SIGIR	DAK	2001-04-23
TR18000161	DİŞİ	SIGIR	KB HOLŞTAYN	2001-06-02
TR18000165	FREE	MANDA	GAK	2001-01-19
TR18000169	FREE	SIGIR	BOZ IRK	2001-01-02
TR18000173	FREE	MANDA	DAK	2001-09-25
TR18000191	HERMAFRODİT	MANDA	DAK	2001-02-11
TR18000197	ERKEK	MANDA	HOLŞTAYN	2001-01-25
TR18000204	HERMAFRODİT	SIGIR	ŞAROLE	2001-06-25
TR18000212	HERMAFRODİT	MANDA	ZAVOT	2001-03-02
TR18000218	HERMAFRODİT	MANDA	JERSEY	2001-02-17
TR18000220	ERKEK	MANDA	KB HOLŞTAYN	2001-02-28
TR18000251	ERKEK	MANDA	KB HOLŞTAYN	2001-03-04
TR18000376	DİŞİ	SIGIR	MONTOFON	2001-01-11
TR18000384	HERMAFRODİT	SIGIR	YERLİ	2001-07-07

Kulak No:

Şekil 3.11. Hayvan listesi ve hayvan girişi yapılan ekran

Şekil 3.11’de görülen hayvan listesinde herhangi bir hayvanın kulak numarasına tıkladığında hayvan ile ilgili detaylı bilgiler görüntülenir. Şekil 3.12’de TR18000076 numaralı kulak numarasına sahip bir hayvanın detaylı bilgileri görüntülenmesi buna örnek olarak verilebilir. Eğer hayvan girişi RFID dışında el ile yapılmış ise bu web sayfası arayıcılığı ile hayvanın kulak numarası RFID taşıyıcısına yüklenir. Bu yükleme işlemi gerçekleştirilebilmek için öncelikle seri port ayarları yapıp RFID okuyucusuna bağlanılır daha sonra RFID işlemleri menüsünden RFID okuyucusu yaz seçeneğine tıklanır. Bu butona tıkladığında mnuRFIDYazClick olayı çalışır. Veritabanındaki kulakno ilk iki karakteri atlanır. TR ülke kodundan sonraki ilk iki rakam şehir plaka kodu ve dokuz rakam hayvan

numarasının her bir karakteri, 792 olan ülke kodu ISO11784 fonksiyonu kullanılarak onaltılık sayı sistemine çevrilir ve seri porta gönderilir

Form1

BFD İşlemleri Seri Port Ayarları

Adres: http://127.0.0.1/ilcetirim/ Git

İşletme Listesi
Sistemden Çıkış

Hayvan Bilgileri

Kulak No: TR1800000076
 Cinsiyeti: HERMAFRODİT
 Nevi: MANDA
 ırkı: ŞAROLE
 Doğum Özelliği: TEK
 Doğum Tarihi: 12-01-2001
 Anne Kulakno: TR1800001076
 Baba Kulakno: TR1800002076
 İşletme Şekli: BESİ
 Son Durumu: Hayatta

Hastalık Adı	Tarih
ENTEROTOXEMİA	25-05-2005
ENTEROTOXEMİA	Hastalık Ekle

Aşı Adı	Tarih
ENTEROTOXEMİA	25-05-2005
ENTEROTOXEMİA	Aşı Ekle

Listeye Dön Bilgileri Kaydet

Şekil 3.12. Hayvan detaylarının alındığı ekran

4. SONUÇ VE ÖNERİLER

Endüstriden, hayvancılığa, trafiğe, havaalanı işletmeciliğine vb. bir çok uygulamada akıllı RF kartlar olarak adlandırılan RFID cihazları kullanılmaktadır. Türkiye’de RFID otoyol geçişlerinde, üretim hatlarında halen kullanılmaktadır. Fakat hayvancılıkla ilgili uygulamalar yok denecek kadar azdır. Çoğunlukla hayvanlara takılan kulak numaraları üzerinden hayvan takibi yapılmakta bu da birden fazla kulak numarası verilmesi gibi sonuçları doğurmaktadır.

Bu çalışmada kulak numarası ile takibi yapılan hayvanların aynı numara kullanılarak RFID taşıyıcı yongaları kullanılarak ulusal bir veritabanında kimliklendirilmesi amaçlanmıştır. Kulak numarası 2 harfli ülke kodu, şehir plaka kodu ve hayvana verilen numaradan oluşur. Bu numara 128 bitlik RFID taşıyıcılarına ISO 11784 ve 11785 standardına göre yerleştirilmiştir. ISO 11784 standardına göre ayrılan 14 bitlik alana şehir kodu kodlanarak daha çok hayvan tanımlaması yapılmıştır. Ayrıca RFID taşıyıcı yongasının data bloğunda 38 bitlik alana 2 bitlik 0 ve 36 bitlik hayvan numarası da kodlanmıştır. Sonuçta her bir şehirde 36 bitlik hayvan numarası (ikilik sayı sisteminde 4’lü kodlanmış onaltılık sayı sistemi) kullanılarak 9⁹ tane hayvan tanımlaması yapılabilir. Türkiye’de 81 şehir olduğuna göre 81 x 9⁹ tane hayvan kimliklendirilebilir. Bu rakam dünyadaki tüm büyükbaş hayvanların sayısından daha büyük bir rakamdır. Ayrıca hayvan kimliklendirmede vücuda enjekte edilen RFID taşıyıcıları kullanıldığı ve standartlara uygun kulak numarası verildiği için çift kulak numarası verilmesi gibi olumsuz sonuçların önüne de geçilmiştir.

Tanımlaması yapılan hayvanlar veritabanına web tabanlı olarak işlenmiştir. Bu durum, sistemin seri portu bulunan herhangi bir bilgisayardan, dünyanın herhangi bir yerinden rahatlıkla çalışmasını sağlamıştır. Ayrıca hayvanlarla ilgili tutulan bir çok bilginin web tabanlı olarak ulusal çapta izlenmesinin hayvancılıkla ilgili politikaların geliştirilmesinde, karantina durumlarında acil önlem alınmasında, hayvan ırkları ve hastalıkları konusunda vb. bir çok çalışmada etkili olabileceği görülmüştür.

Web tabanlı uygulamalar sunumcular üzerinde çalıştırılır. Özellikle ülkemizde sunumcu olarak kamu sektöründe Windows tabanlı işletim sistemleri kullanılmaktadır. Bu da kamu sektöründe harcamaların artmasına neden olmaktadır. Türkiye’de sunucu olarak kamuda Linux kullanımı ise henüz yerleşmemiştir. Sunucularda Linux kullanılmasının işletim sistemi ve kullanılan açık kaynak kodlu yazılımlar sayesinde maliyetleri düşüreceği ve yazılımlarla ilgili güvenlik sorunlarının ortadan kalkacağı da tespit edilmiştir.

Günümüzde üçüncü nesil RFID taşıyıcıları geliştirilmektedir. Yakın bir zamanda da sensör tipindeki bu RFID taşıyıcıları hayvancılıkta kullanılmaya başlanılacaktır. Sensör şeklindeki bu RFID taşıyıcılar sayesinde hayvanların sağlık ile ilgili durumları takip edilebilecektir. Bu bilgiler ayrıca bu çalışmada geliştirilen web tabanlı uygulamalar ile de birleştirilebilecektir.

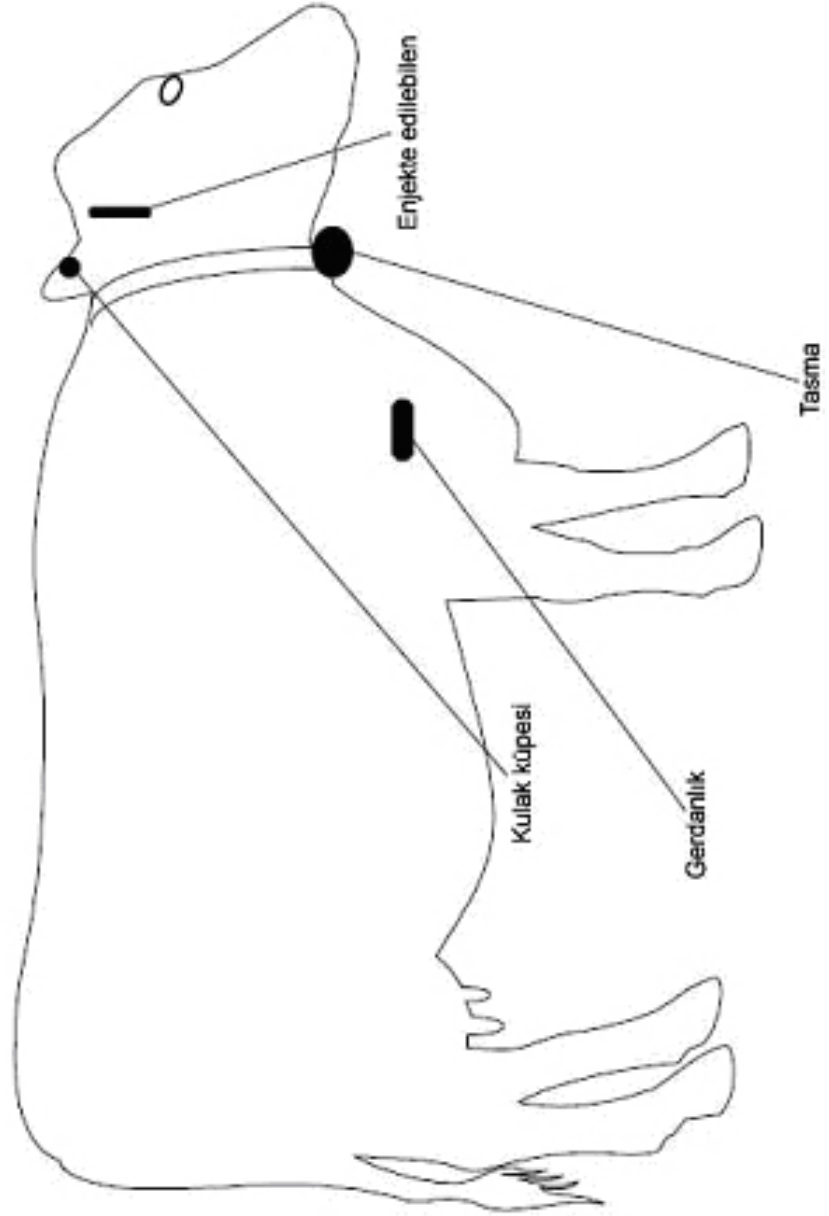
KAYNAKLAR

1. Spahr, S.L., "Methods and systems for cow identification in large dairy herd management", *Management Services*, 672-681 (1992).
2. Caja, G., Ribo, O. and Nehring, R., "Evaluation of migratory distance of passive transponders injected in different body sites of adult sheep for electronic identification", *Livestock Production Science*, 55 : 279-289 (1998).
3. Klindtworth, M., Wendl, G., Klindtworth, K. and Pirkelmann, H., "Electronic identification of cattle with injectable transponders", *Computers and Electronics in Agriculture*, 24 : 65-79 (1999).
4. Jansen, M. B. and Eradus, W. J., "Future developments on devices for animal radio frequency identification", *Computers and Electronics in Agriculture*, 24 : 109-117 (1999).
5. Wismans, W.M.G., "Identification and registration of animals in the European Union", *Computers and Electronics in Agriculture*, 24 : 99-108 (1999).
6. Jansen, M. B. and Eradus, W. J., "Animal identification and monitoring", *Computers and Electronics in Agriculture*, 24 : 91-98 (1999).
7. Finkenzeller, K., "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition", Rachel Waddington, John Wiley & Sons, *West Sussex*, 2-22 (1993).
8. Sriram, T., Vishwanatha, R. K., Biswas, S., Ahmed, B., "Applications of barcode technology in automated storage and retrieval systems", *Proceedings of the 1996 IEEE IECON 22nd International Conference*, Taipei, 641 – 646 (1996).
9. Younhee G., Dosung A., Sungbum P., Yongwha C., "Access control system with high level security using fingerprints", *32nd Applied Imagery Pattern Recognition Workshop*, Washington, 238 – 243 (2003).
10. Jain, A. and Lin H., "On-line fingerprint verification", *Proceedings of the 1996 IEEE IECON 22nd International Conference*, Vienna, 596 – 600 (1996).
11. Sanchez-Reillo, R. and Sanchez-Avila, C., "Fingerprint verification using smart cards for access control systems", *Aerospace and Electronic Systems Magazine*, 17 (9): 12 -15 (2002).
12. Sunde, J., Butavicius, M.A., Graves, I., Hemming, D., Ivancevic, V., Johnson, R., Kaine, A.K., McLindin, B.A., Meaney, K., "A methodology for evaluating the operational effectiveness of facial recognition systems", *4th EURASIP Conference*, Zagreb, 441-448 (2003).

13. Fujioka, M., Yamamoto, S., Nakamura, M., Mukasa, T., Inoue, N., "Experience and evolution of voice recognition applications for telecommunications services", *Global Telecommunications Conference GLOBECOM 98*, Sydney, 1338-1343 (1998).
14. Plaza-Aguilar, J.L., Baez-Lopez, D., Guerrero-Ojeda, L., Asomoza, J.R., "A voice recognition system for speech impaired people", *CONIELECOMP 14th International Conference*, Veracruz, 162-167 (2004).
15. Peng-Fei, Z., De-Sheng, L., Qi, W., "A novel iris recognition method based on feature fusion", *Machine Learning and Cybernetics Proceedings of 2004 International Conference*, Hollanda, 3661-3665 (2004).
16. Won J., S., Byung H., A., Won H., K., "Healthcare information systems using digital signature and synchronized smart cards via the Internet", *Coding and Computing Proceedings. International Conference*, Las Vegas, 177-182 (2002).
17. Cooke, J.C. and Brewster, R.L., "The use of smart cards in personal communication systems security", *Telecommunications Fourth IEEE Conference*, Londra, 246-251 (1996).
18. Garfinkel, S.L., Juels, A., Pappu, R., "RFID privacy: an overview of problems and proposed solutions", *Security & Privacy Magazine IEEE*, 3(3): 34-43 (2005).
19. K. V.S. Rao, "An overview of backscattered radio frequency identification system(RFID)", *1999 Asia Pacific Microwave Conference*, Singapore, 746-749 (1999).
20. Bridgelall, R., "Enabling mobile commerce through pervasive communications with ubiquitous RF tags", *Wireless Communications and Networking WCNC*, Louisiana, 2041-2046 (2003).
21. Tuttle, J.R., "Traditional and emerging technologies and applications in the radiofrequency identification (RFID) industry", *Radio Frequency Integrated Circuits (RFIC) Symposium*, Denver, 5-8 (1997)
22. Manish, B. and Shahram, M., "RFID Field Guide:Deploying Radio Frequency Identification Systems", *Printice Hall*, California, 7-22 (1995)
23. Ollivier, M.M., "RFID-a practical solution for problems you didn't even know you had!", *Wireless Technology IEEE*, 3 : 1-6 (1996).
24. Juels, A., "Yoking-proofs for RFID tags", *Pervasive Computing and Communications Workshops Proceedings of the Second IEEE Annual Conference*, Hawaii, 138-143 (2004).

25. Weinstein, R., "RFID: a technical overview and its application to the enterprise", *IT Professional*, 7(3): 27-33 (2005).
26. Cheng-Ju, L., Li, L., Shi-Zong, C., Chi, C. W., Chun-Huang H., Xin-Mei C., "Mobile healthcare service system using RFID", *Networking, Sensing and Control IEEE International Conference*, Taipei, 1014-1019 (2004).
27. Cerino, A. and Walsh, W.P., "Research and application of radio frequency identification (RFID) technology to enhance aviation security", *National Aerospace and Electronics Conference NAECON 2000*, Dayton, 127-135 (2000).
28. Il-Yeol S. and Froehlich, K., "Entity-relationship modeling", *Potentials, IEEE*, 13(5): 29-34 (1994).
29. DuBois, P., "Mysql Cookbook", O'Reilly, California, 1-46 (2002)
30. Kabir, M. J., "Apache Server 2 Bible", Hungry Minds Inc., Newyork, 3-13 (2002)
31. Naramore, E., Gerner, J., Scouarnec, Y. L., Stolz, J., Glass, M. K., "Beginning PHP5, Apache and Mysql Web Development", John Wiley & Sons, Indiana, 3-26 (2005)
32. Greenspan, J. and Bulger, B., "Mysql/PHP Database Applications", M&T Boks, California, 21-45 (2001)
33. Lerdorf, R. and Tatroe K., "Programming PHP", Kaehms B. and McGredy R., O'Reilly, California, 158-188 (2002)

EKLER

EK -1. Büyükbaş Hayvanların Vücuduna Takılan Taşıyıcı Yongalar

EK -2. Büyükbaş hayvan veritabanında yer alan tabloların oluşturulması

```
CREATE TABLE asi (  
    kod char(2) NOT NULL default "",  
    adi varchar(20) NOT NULL default "",  
    PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE bakanlik (  
    perno varchar(8) NOT NULL default "",  
    adi varchar(20) NOT NULL default "",  
    soyadi varchar(20) NOT NULL default "",  
    sifre varchar(8) NOT NULL default "",  
    ulkekodu char(2) NOT NULL default 'TR',  
    PRIMARY KEY (perno)  
) TYPE=MyISAM;
```

```
CREATE TABLE cinsiyet (  
    kod char(1) NOT NULL default "",  
    adi varchar(15) NOT NULL default "",  
    PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE dogumozellik (  
    kod char(1) NOT NULL default "",  
    adi varchar(15) NOT NULL default ""  
) TYPE=MyISAM;
```

```
CREATE TABLE durum (  
    kod char(1) NOT NULL default "",  
    adi varchar(15) NOT NULL default "",  
    PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE hastalik (  
    kod char(2) NOT NULL default "",  
    adi varchar(25) NOT NULL default "",  
    PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE hayvan (  
    kulakno char(10) NOT NULL default "",  
    cinsiyeti char(1) default NULL,  
    nevi char(1) default NULL,  
    irki char(2) default NULL,  
    dogumozelligi char(1) default NULL,  
    dogumtarihi date default '0000-00-00',  
    annukulakno char(10) default NULL,  
    babakulakno char(10) default NULL,  
    isletmesekli char(1) default NULL,  
    durum char(1) default '1',  
    PRIMARY KEY (kulakno)  
) TYPE=MyISAM;
```

```
CREATE TABLE hayvan_asi (  
    id double NOT NULL auto_increment,  
    kulakno char(10) NOT NULL default "",  
    asikod char(2) NOT NULL default "",  
    tarih date NOT NULL default '0000-00-00',  
    PRIMARY KEY (id)  
) TYPE=MyISAM AUTO_INCREMENT=110 ;
```

```
CREATE TABLE hayvan_hastalik (  
    id double NOT NULL auto_increment,  
    kulakno char(10) NOT NULL default "",  
    hastalikkod char(2) NOT NULL default "",  
    tarih date NOT NULL default '0000-00-00',  
    PRIMARY KEY (id)  
) TYPE=MyISAM AUTO_INCREMENT=111 ;
```

```
CREATE TABLE hayvan_isletme (  
  id double NOT NULL auto_increment,  
  kulakno char(10) NOT NULL default "",  
  isletmekod char(6) NOT NULL default "",  
  PRIMARY KEY (id)  
) TYPE=MyISAM PACK_KEYS=0 AUTO_INCREMENT=412 ;
```

```
CREATE TABLE ilcetarim (  
  ilcekodu varchar(4) NOT NULL default "",  
  ilceadi varchar(20) NOT NULL default "",  
  sifre varchar(8) NOT NULL default "",  
  ilkodu char(2) NOT NULL default "",  
  PRIMARY KEY (ilcekodu)  
) TYPE=MyISAM;
```

```
CREATE TABLE iltarim (  
  ilkodu char(2) NOT NULL default "",  
  iladi varchar(20) NOT NULL default "",  
  sifre varchar(8) NOT NULL default "",  
  ulkekodu char(2) NOT NULL default 'TR',  
  PRIMARY KEY (ilkodu)  
) TYPE=MyISAM;
```

```
CREATE TABLE irk (  
  kod char(2) NOT NULL default "",  
  adi varchar(15) NOT NULL default "",  
  PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE isletme (  
  isletmekod varchar(6) NOT NULL default "",  
  adi varchar(20) default NULL,  
  soyadi varchar(20) default NULL,  
  adres varchar(100) default NULL,  
  adresil char(2) default NULL,  
  adresilce varchar(4) default NULL,  
  telefon varchar(11) default NULL,  
  PRIMARY KEY (isletmekod)  
) TYPE=MyISAM;
```

```
CREATE TABLE isletmesekil (  
  kod char(1) NOT NULL default "",  
  adi varchar(15) NOT NULL default "",  
  PRIMARY KEY (kod)  
) TYPE=MyISAM;
```

```
CREATE TABLE nevi (  
  kod char(1) NOT NULL default "",  
  adi varchar(15) NOT NULL default "",  
  PRIMARY KEY (kod)  
) TYPE=MyISAM;
```



```

?>
<TABLE style="BORDER-COLLAPSE: collapse" borderColor=#111111 cellSpacing=0 cellPadding=0
width=400 border=0>
<TBODY>
<TR>
<TD class=thright width=400 height=26>
<SELECT class=InputActive size=1 name=asi>
<?php
    $sqlasi="select * from asi";
    $resultasi=mysql_query($sqlasi);
    $numresultasi=mysql_num_rows($resultasi);
    for ($i=0;$i<$numresultasi;$i++)
    {
        $bilgiasi=mysql_fetch_array($resultasi);
        echo("<option value=\".$bilgiasi[0].\">".$bilgiasi[1]."</option>");
    }
?>
</SELECT>
<INPUT class=fields type=submit value="Aþý Ekle" name=asiekle> &nbsp;
</TD>
</TR>
</TBODY>
</TABLE>
<br>

```

common.inc.php:

```

<?php
function HTMLbasla()
{
    echo("<HTML>\n");
    echo("<HEAD>\n");
    echo("<META http-equiv=Content-Language content=tr>\n");
    echo("<META http-equiv=content-type content='text/html; charset=iso-8859-9'\>\n");
    echo("<LINK href='style.css' type=text/css rel=stylesheet>\n");
    echo("<BODY>\n");
}

function HTMLbitir()
{
    echo("</BODY>\n");
    echo("</HTML>\n");
}

function connecting()
{
    $hostname="localhost";
    $username="root";
    $password="deneme";
    $database="buyukbas";
    @ $db=mysql_connect ($hostname, $username,$password)
    or die ('Sistemde bir hata oluþtu.<br>Lütfen daha sonra tekrar deneyiniz');
    mysql_select_db($database);
}

function pconnecting()
{
    $hostname="localhost";
    $username="root";
    $password="deneme";
    $database="buyukbas";
    @ $db=mysql_connect ($hostname, $username,$password)

```

```

    or die ('Sistemde bir hata oluþtu.<br>Lütfen daha sonra tekrar deneyiniz');
    mysql_select_db($database);
}
?>

```

giris.php:

```

<?php
require("common.inc.php");
if (empty($_HTTP_POST_VARS['sistemgir']))
{
    HTMLBasla();
    echo("<p class=tdcenter>Ýlçe Kodu yada Þifre boş bırakýlmamalýdır</p>");
    HTMLBitir();
    exit();
}
else
{
    connecting();
    $ilcekodu=$_HTTP_POST_VARS['ilcekodu'];
    $sifre=$_HTTP_POST_VARS['sifre'];
    $sql="select ilcekodu,ilkodu from ilcetarim where ilcekodu=" . $ilcekodu . " and sifre=" . $sifre ."";
    $result=mysql_query($sql);
    $numresult=mysql_num_rows($result);
    $bilgi=mysql_fetch_array($result);

if ($numresult==0)
{
    HTMLBasla();
    echo("<p class=tdcenter>Hatalý ilçe kodu yada þifre</p>");
    HTMLBitir();
    exit();
}
else
{
    session_start();
    session_register("ilcekodu","ilkodu");
    $_HTTP_SESSION_VARS["ilcekodu"]=$bilgi[0];
    $_HTTP_SESSION_VARS["ilkodu"]=$bilgi[1];
    header("Location: ana.php");
}
}
?>

```

hastalikliste.php:

```

<?php

    $sql="SELECT hastalik.adi,hayvan_hastalik.tarih FROM hayvan_hastalik,hastalik ".
        "WHERE hayvan_hastalik.hastalikkod=hastalik.kod and
hayvan_hastalik.kulakno=" . $kulakno ."";
    $result=mysql_query($sql);
    $numresult=mysql_num_rows($result);
    if ($numresult!=0)
    {
        echo("<center>\n");
        echo("<br>\n");
        echo("<table style='BORDER-COLLAPSE: collapse' borderColor=#111111 width=400
border=1>\n");
        echo("<TR>\n");

```

```

echo("<TD class=thleft width=300 height=18>Hastalýk Adý</TD>\n");
echo("<TD class=thcenter width=100 height=18>&nbsp;&nbsp;&nbsp;Tarih</TD>\n");
echo("<TR>\n");
for ($i=0;$i<$numresult;$i++)
{
    $bilgi=mysql_fetch_array($result);
    echo("<TR>\n");
    echo("<TD class=tdleft width=300 height=18>&nbsp;&nbsp;&nbsp;".$bilgi[0]."</a></TD>\n");
    echo("<TD class=tdcenter width=100 height=18>&nbsp;&nbsp;&nbsp;".$bilgi[1]."</TD>\n");
    echo("</TR>\n");
}
echo("</table>\n");
echo("</center>\n");
}
?>

```

```

<TABLE style="BORDER-COLLAPSE: collapse" borderColor=#111111 cellSpacing=0 cellPadding=0
width=400 border=0>
<TBODY>
<TR>
<TD class=tright width=400 height=26>
<SELECT class=InputActive size=1 name=hastalik>
<?php
    $sqlhast="select * from hastalik";
    $resulthast=mysql_query($sqlhast);
    $numresulthast=mysql_num_rows($resulthast);
    for ($i=0;$i<$numresulthast;$i++)
    {
        $bilgihast=mysql_fetch_array($resulthast);
        echo("<option value="."$bilgihast[0].">".$bilgihast[1]."</option>");
    }
?>
</SELECT>
<INPUT class=fields type=submit value="Hastalýk Ekle" name=hastalikekle> &nbsp;&nbsp;&nbsp;</TD>
</TR>
</TBODY>
</TABLE>
<br>

```

ilcetarim.php:

```

<?php

session_start();
if (!session_is_registered ('ilcekodu'))
{
    HTMLBasla();
    echo("<p class=tdcenter>Bifresiz sisteme girip yapýlamaz</p>");
    HTMLBitir();
    exit();
}

require('common.inc.php');

$ilkodu=$_HTTP_GET_VARS['ilkodu'];
connecting();
$sql="select ilcekodu,ilceadi from ilcetarim where ilkodu='".$ilkodu.'" order by ilceadi";
$result=mysql_query($sql);
$numresult=mysql_num_rows($result);

```


isletme.php:

```

<?php
    session_start();
    if (!session_is_registered ('ilcekodu'))
    {
        HTMLBasla();
        echo("<p class=tdcenter>Þifresiz sisteme girip yapýlamaz</p>");
        HTMLBitir();
        exit();
    }

    require('common.inc.php');

    $ilcekodu=$HTTP_SESSION_VARS["ilcekodu"];
    connecting();
    $sql="select isletmekod,concat(adi,' ',soyadi),adres from isletme where adresilce="."$ilcekodu." order
by adi";
    $result=mysql_query($sql);
    $numresult=mysql_num_rows($result);

    HTMLbasla();
    echo("<form name=form1 method=POST action=isletmeekle.php><center>\n");
    echo("<br>\n");
    echo("<table style=\"BORDER-COLLAPSE: collapse\" borderColor=#111111 width=540
border=1>\n");
    echo("<TR>\n");
    echo("<TD class=thcenter width=90 height=18>Ýpletme Kodu</TD>\n");
    echo("<TD class=thleft width=200 height=18>&nbsp;&nbsp;&nbsp;Adý Soyady</TD>\n");
    echo("<TD class=thleft width=250 height=18>&nbsp;&nbsp;&nbsp;Adresi</TD>\n");
    echo("<TR>\n");
    for ($i=0;$i<$numresult;$i++)
    {
        $bilgi=mysql_fetch_array($result);
        echo("<TR>\n");
        echo("<TD class=tdcenter width=90 height=18><a
href=hayvanliste.php?isletmekodu="."$bilgi[0].">".$bilgi[0]."</a></TD>\n");
        echo("<TD class=tdleft width=200 height=18>&nbsp;&nbsp;&nbsp;".$bilgi[1]."</TD>\n");
        echo("<TD class=tdleft width=250 height=18>&nbsp;&nbsp;&nbsp;".$bilgi[2]."</TD>\n");
        echo("</TR>\n");
    }
    echo("</table>\n");
?>

<TABLE style="BORDER-COLLAPSE: collapse" borderColor=#111111 cellSpacing=0 cellPadding=0
width=540 border=0>
<TBODY>
<TR>
<TD class=thright width=540 height=26>
    Ýpletme Kodu:&nbsp;&nbsp;&nbsp;
    <INPUT class=fields type=text value="" name=isletmekodu maxlength=6>
    <INPUT class=fields type=submit value="Ýpletme Ekle" name=isletmeekle>
    &nbsp;&nbsp;&nbsp;
</TD>
</TR>
</TBODY>
</TABLE>
<INPUT type=hidden value=<?php echo($ilcekodu);?> name=ilcekodu>
</center>
</FORM>
<?php
    HTMLbitir();
?>

```

hayvanekle.php:

```

<?php

    session_start();
    if (!session_is_registered ('ilcekodu'))
    {
        HTMLBasla();
        echo("<p class=tdcenter>İfresiz sisteme girip yapılamaz</p>");
        HTMLBitir();
        exit();
    }

    require('common.inc.php');
    connecting();
    $ilcekodu=$HTTP_POST_VARS['ilcekodu'];
    $isletmekodu=$HTTP_POST_VARS['isletmekodu'];

    if (isset($HTTP_POST_VARS['isletmeekle']))
    {
        $sql="select * from isletme where isletmekod='".$isletmekodu."'";
        $result=mysql_query($sql);
        $numresult=mysql_num_rows($result);

        if ($numresult!=0)
        {
            HTMLBasla();
            echo("<p class=tdcenter>Bu iletme sistemde zaten kayıtlı</p>");
            HTMLBitir();
            exit();
        }

        $sql="select ilkodu from ilcetarim where ilcekodu='".$ilcekodu."'";
        $result=mysql_query($sql);
        $bilgi=mysql_fetch_array($result);
        $ilkodu=$bilgi[0];

        $sql="insert into isletme values('".$isletmekodu."',null,null,null,'" . $ilkodu . "','" . $ilcekodu . "',null)";
        mysql_query($sql);
    }

    if (isset($HTTP_POST_VARS['kaydet']))
    {
        $sql="update issletme set
isletmekod='".$HTTP_POST_VARS['isletmekodu']."',adi='".$HTTP_POST_VARS['adi']."',
"soyadi='".$HTTP_POST_VARS['soyadi']."',adres='".$HTTP_POST_VARS['adres']."',
"adresil='".$HTTP_POST_VARS['ilkodu']."',adresilce='".$HTTP_POST_VARS['ilcekodu']."',
"telefon='".$HTTP_POST_VARS['telefon']."' where isletmekod='".$isletmekodu."'";
        mysql_query($sql);

        header("Location: http://127.0.0.1:1453/proje/isletme.php?ilcekodu=".$ilcekodu);
        exit();
    }

    $sql="select * from isletme where isletmekod='".$isletmekodu."'";
    $result=mysql_query($sql);
    $bilgi=mysql_fetch_array($result);
    HTMLbasla();
?>
<CENTER>

```



```

?>
</SELECT>
</TD>
</TR>
<TR>
<TD class=tdright width=147 height=27>&nbsp;Adres Ýlçe</TD>
<TD class=tdcenter width=3 height=27>:</TD>
<TD class=tdleft width=360 height=27>&nbsp;
<SELECT class=InputActive size=1 name=adresilce disabled>
<?php
$sqlilce="select ilcekodu,ilceadi from ilcetarim where ilkodu=".$ilkodu."";
$resultilce=mysql_query($sqlilce);
$numresultilce=mysql_num_rows($resultilce);
for ($i=0;$i<$numresultilce;$i++)
{
    $bilgiilce=mysql_fetch_array($resultilce);
    if ($bilgi[5]==$bilgiilce[0])
    {
        echo("<option value=".$bilgiilce[0]." selected>".$bilgiilce[1]."</option>\n");
    }
    else
    {
        echo("<option value=".$bilgiilce[0].">".$bilgiilce[1]."</option>\n");
    }
}
}
?>
</SELECT>
</TD>
</TR>
<TR>
<TD class=tdright width=147 height=27>&nbsp;Telefon</TD>
<TD class=tdcenter width=3 height=27>:</TD>
<TD class=tdleft width=360 height=27>&nbsp;
<INPUT class=InputActive size=10 name=telefon maxlength=11>
</TD>
</TBODY>
</TABLE>
</TD>
</TR>
</TBODY>
</TABLE>
<TABLE style="BORDER-COLLAPSE: collapse" borderColor=#111111 cellSpacing=0 cellPadding=0
width=512 border=0>
<TBODY>
<TR>
<TD class=tright width=596 height=26>
<INPUT class=fields type=submit value="Bilgileri Kaydet" name=kaydet>
&nbsp;
</TD>
</TR>
</TBODY>
</TABLE>
<INPUT type=hidden value=<?php echo($isletmekodu);?> name=isletmekodu>
<INPUT type=hidden value=<?php echo($ilkodu);?> name=ilkodu>
<INPUT type=hidden value=<?php echo($ilcekodu);?> name=ilcekodu>
</FORM>
</CENTER>
</BODY>
</HTML>

```

solmenu.php:

```

<?php
session_start();
if (!session_is_registered ('ilcekodu'))
{
    HTMLBasla();
    echo("<p class=tdcenter>İfresiz sisteme girip yapılamaz</p>");
    HTMLBitir();
    exit();
}

$ilcekodu=$HTTP_SESSION_VARS['ilcekodu'];
?>
<html>
<head>
<meta http-equiv="Content-Language" content="tr">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<LINK href="style.css" type=text/css rel=stylesheet>
<title>Yeni Sayfa 1</title>
</head>
<body leftmargin=0 topmargin=0>
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#000000" width="100%" id="AutoNumber1" height="620" bgcolor="#C0C0C0">
<tr>
<td width="170" height="77" valign="top">
<br>
<center>
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse" color="#111111"
width="94%" id="AutoNumber2" bgcolor="#C0C0C0" bordercolor="#000000">
<tr>
<td width="100%" height="20" class=tdcenter>
<a href="isletme.php?ilcekodu=<?php echo($ilcekodu);?>" target=sag>Yıpletme Listesi</a></td>
</tr>
</table>
</center>
</td>
</tr>
</table>
</body>
</html>

```

cikis.php:

```

<?php
session_start();
session_unset();
session_destroy();
?>
<html>
<body>
<script type="text/javascript">
window.top.location="index.html"
</script>
</body>
</html>

```

EK -4. Delphi program kaynak kodları

```

unit Demosrc;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, Menus, About, ImgList, OleCtrls, SHDocVw,
  ComCtrls, ToolWin;
type

  TInitCommunication = Function(PortNo:Byte;BaudRate:Word;
    Var hCOM:THANDLE):integer;
  TMRPCOM = Procedure (hCOM:THANDLE;Length,Command1,Command2:Byte;HexData:Pchar;
    Var CommErrorNo:Byte;Var InBuffer:Pchar);
  TCalc_DBCC = Procedure (Data:Pchar; var MSB_X:Byte; var LSB_X:Byte);
  TMRPProgData = procedure (hCOM:THANDLE; CardDat:PChar; var Response:PChar;
    var Stat:Byte);
  TTimeDelay = Procedure (Delay:LongInt);

  TForm1 = class(TForm)
    Timer: TTimer;
    MainMenu1: TMainMenu;
    ComportAyar1: TMenuItem;
    mnuBaglan: TMenuItem;
    mnuCOM1: TMenuItem;
    mnuCOM2: TMenuItem;
    mnuCOM3: TMenuItem;
    mnuCOM4: TMenuItem;
    mnuBagKes: TMenuItem;
    RFIDlemleri1: TMenuItem;
    mnuRFIDOku: TMenuItem;
    mnuRFIDYaz: TMenuItem;
    mnuRO: TMenuItem;
    mnuRW: TMenuItem;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    Panel1: TPanel;
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    WebBrowser1: TWebBrowser;
    ImageList1: TImageList;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure AboutClick(Sender: TObject);
    procedure ExitClick(Sender: TObject);
    procedure mnuCOM1Click(Sender: TObject);
    procedure mnuCOM2Click(Sender: TObject);
    procedure mnuCOM3Click(Sender: TObject);
    procedure mnuCOM4Click(Sender: TObject);
    procedure mnuBaglanClick(Sender: TObject);
    procedure mnuBagKesClick(Sender: TObject);
    procedure mnuRFIDOkuClick(Sender: TObject);
    procedure mnuRFIDYazClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure ToolButton2Click(Sender: TObject);
    procedure ToolButton1Click(Sender: TObject);
    procedure Panel1Click(Sender: TObject);
  end;

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  LibHandle:THandle;
  ComHandle : THandle;
  MRPCOM : TMRPCOM;
  InitCommunication : TInitCommunication;
  Calc_DBCC : TCalc_DBCC;
  MRPPProgData : TMRPPProgData;
  TimeDelay : TTimeDelay;
  MRPTxDat:PChar;
  MRPRxDat:PChar;
  ErrNo:Byte;
  MSB_DBCC, LSB_DBCC : Byte;
  ProgDat:PChar;
  PortErr: Boolean;
  x,y:integer;
implementation

{$R *.DFM}

Function HexToInt(c:Char):Byte;
Begin
  Case C Of
    '0','1','2','3','4','5','6','7','8','9':HexToInt:=StrToInt(C);
    'a','A':HexToInt:=10;
    'b','B':HexToInt:=11;
    'c','C':HexToInt:=12;
    'd','D':HexToInt:=13;
    'e','E':HexToInt:=14;
    'f','F':HexToInt:=15
    Else HexToInt:=0;
  End;
End;

function DecToBin(N: Integer): string;
var
  S: string;
  i: Integer;
begin
  if (N=0) then
    DecToBin :='0'
  else
    begin
    for i:=1 to SizeOf(N)*8 do
    begin
      if N<0 then S:=S+'1'
      else S:=S+'0';
      N:=N shl 1;
    end;
    Delete(S,1,Pos('1',S)-1);
    DecToBin:=S;
  end;
end;

function HexToBin(HexStr: string): string;
const
  BinArray: array[0..15, 0..1] of string =

```

```

    ('0000', '0'), ('0001', '1'), ('0010', '2'), ('0011', '3'),
    ('0100', '4'), ('0101', '5'), ('0110', '6'), ('0111', '7'),
    ('1000', '8'), ('1001', '9'), ('1010', 'A'), ('1011', 'B'),
    ('1100', 'C'), ('1101', 'D'), ('1110', 'E'), ('1111', 'F'));
var
  i, j: Integer;
begin
  result := "";
  for i:=1 to length(HexStr) do
    begin
      for j:=0 to 15 do
        begin
          if (copy(HexStr,i,1)=BinArray[j][1]) then
            result :=result+BinArray[j][0];
          end;
        end;
      end;
    end;
end;

function BinToHex(BinStr: string): string;
const
  BinArray: array[0..15, 0..1] of string =
    ('0000', '0'), ('0001', '1'), ('0010', '2'), ('0011', '3'),
    ('0100', '4'), ('0101', '5'), ('0110', '6'), ('0111', '7'),
    ('1000', '8'), ('1001', '9'), ('1010', 'A'), ('1011', 'B'),
    ('1100', 'C'), ('1101', 'D'), ('1110', 'E'), ('1111', 'F'));
var
  i, j: Integer;
begin
  result := "";
  for i:=0 to trunc(length(BinStr)/4) do
    begin
      for j:=0 to 15 do
        begin
          if (copy(BinStr,4*i+1,4)=BinArray[j][0]) then
            result :=result+BinArray[j][1];
          end;
        end;
      end;
    end;
end;

function ISO11784(ilkodu:string;hayvankodu:string):string;
var
  i:integer;
  hayvankodBin:string;
begin
  hayvankodBin:= "";
  for i:=1 to 9 do
    begin
      hayvankodBin :=hayvankodBin+HexToBin(copy(hayvankodu,i,1));
    end;
  result :='1000000'+HexToBin(copy(ilkodu,1,1))+HexToBin(copy(ilkodu,2,1))+ '1'+DecToBin(792);
  result := result+'00'+hayvankodBin;
  result:=BinToHex(result);
end;

function ilkoducoz(ISO11784:string):string;
var
  ilkoduBin:string;
begin
  ISO11784 := HexToBin(ISO11784);
  ilkoduBin :=copy(ISO11784,8,4)+copy(ISO11784,12,4);
  ilkoducoz :=BinToHex(ilkoduBin);
end;

```

```

function hayvankoducoz(ISO11784:string):string;
var
  hayvankoduBin:String;
begin
  ISO11784 := HexToBin(ISO11784);
  hayvankoduBin:=copy(ISO11784,29,36);
  hayvankoducoz:=BinToHex(hayvankoduBin);
end;

```

```

function Data_Valid(hexnumber:string):Boolean;
var
  txtlen, Ord_str : integer;
  tempstr : string;
  Valid: Boolean;
begin
  Valid := TRUE;
  txtlen:=Length(hexnumber);
  tempstr:=hexnumber;
  if txtlen = 0 then
    Valid:=FALSE
  else
    begin
      while (txtlen > 0) do
        begin
          Ord_str := Ord(tempstr[txtlen]);
          if (Ord_str < 48) OR (Ord_str > 102) OR
            ((Ord_str < 97) AND (Ord_str > 70 )) OR
            ((Ord_str < 65) AND (Ord_str > 57 )) then
            begin
              Valid:=FALSE;
              hexnumber:="";
              txtlen:=0; {quit}
            end;
            txtlen:=txtlen-1;
          end;
        end;
      Data_Valid:= Valid;
    end;
end;

```

```

function HandleResponse( Mode:Integer;var sonuc:String) : Byte;

var Len : Byte;
  Status : Byte;
  Data,msg:String;
  I : Word;
begin
  msg :=";

  Status:=HexToInt(MRPRxDat[4])*16+HexToInt(MRPRxDat[5]);

  if (Status AND $04) = $04 then
    begin
      if (Status AND $08) <> $08 then
        msg := 'RFID taşıyısı okuma mesafesi dışında';

      //Sadece okunabilir transporder

```

```

if (Status AND $03) = $00 then
begin
  Form1.mnuRO.Checked :=True;
  Form1.mnuRW.Checked :=False;
  Case Mode of
  1 : ;
  2 : begin
    msg :='Programlama hatası.'+chr(13)+'RFID taşıyıcısı sadece okunabilir';
    end;
  3 : ;
  4 : begin
    msg :='Programlama hatası.'+chr(13)+'RFID taşıyıcısı sadece okunabilir';
    end;
  else
  end;{case}

end
else if (Status AND $03) = $01 then
begin
  Form1.mnuRO.Checked :=False;
  Form1.mnuRW.Checked :=True;
end
else if (Status AND $03) = $02 then
begin
  Form1.mnuRO.Checked :=False;
  Form1.mnuRW.Checked :=False;
end
else if (Status AND $03) = $03 then
begin
  Form1.mnuRO.Checked :=False;
  Form1.mnuRW.Checked :=False;
  Case Mode of
  1 : ;
  2 : begin
    msg :='Programlama hatası'+chr(13)+'Lütfen RFID taşıyıcısını yeniden programlayın';
    end;
  3 : ;
  4 : begin
    msg :='Programlama hatası'+chr(13)+'Lütfen RFID taşıyıcısını yeniden programlayın';
    end;
  else
  end;{case}
end;

for l:=1 to (Len-1) do
  Data:= Data+MRPRxDat[22-2*l]+MRPRxDat[23-2*l];
sonuc :=Data;

end
else
begin
  msg :='RFID taşıyıcı okuma alanı dışında yada arızalı';
end;

if (msg<>'') then
  MessageDlgPos(msg, mtError, [mbOK], 0,Form1.Left+x,Form1.Top+y);
HandleResponse := 1;

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  x :=100;

```

```

y :=100;
LibHandle:=LoadLibrary('MRP95LIB.DLL');
if LibHandle <> 0 then
begin
  @MRPCOM:=GetProcAddress(LibHandle,'MRPCOM');
  @InitCommunication:=GetProcAddress(LibHandle,'InitCommunication');
  @Calc_DBCC:=GetProcAddress(LibHandle,'Calc_DBCC');
  @MRPPProgData := GetProcAddress(LibHandle,'MRPPProgData');
  @TimeDelay := GetProcAddress(LibHandle,'TimeDelay');

  if (@MRPCOM <> nil) AND (@InitCommunication <> nil) AND
    (@Calc_DBCC <> nil) AND (@MRPPProgData <> nil) then
  begin
    PortErr:=True;
    GetMem(MRPTxDat,263);
    GetMem(MRPRxDat,263);
    GetMem(ProgDat,20);

  end
  else
    if MessageDlgPos('MRP95LIB.DLL dosyasında bir hata oluştu',
      mtError, [mbAbort], 0 ,Form1.Left+x,Form1.Top+y) = mrAbort then
      Close;
  end
  else
    if MessageDlgPos('MRP95LIB.DLL dosyasında bir hata oluştu',
      mtError, [mbAbort], 0 ,Form1.Left+x,Form1.Top+y) = mrAbort then
      Close;
  end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  FreeMem(MRPTxDat);
  FreeMem(MRPRxDat);
  FreeMem(ProgDat);
end;

procedure TForm1.ExitClick(Sender: TObject);
begin
  Close;
end;

procedure TForm1.mnuCOM1Click(Sender: TObject);
begin
  mnuCOM1.Checked:=True;
  mnuCOM2.Checked:=False;
  mnuCOM3.Checked:=False;
  mnuCOM4.Checked:=False;
end;

procedure TForm1.mnuCOM2Click(Sender: TObject);
begin
  mnuCOM1.Checked:=False;
  mnuCOM2.Checked:=True;
  mnuCOM3.Checked:=False;
  mnuCOM4.Checked:=False;
end;

procedure TForm1.mnuCOM3Click(Sender: TObject);
begin
  mnuCOM1.Checked:=False;

```

```
mnuCOM2.Checked:=False;
mnuCOM3.Checked:=True;
mnuCOM4.Checked:=False;
end;
procedure TForm1.mnuCOM4Click(Sender: TObject);
begin
  mnuCOM1.Checked:=False;
  mnuCOM2.Checked:=False;
  mnuCOM3.Checked:=False;
  mnuCOM4.Checked:=True;
end;

procedure TForm1.mnuBaglanClick(Sender: TObject);
var
  msg:String;
begin
  if mnuCOM1.Checked = True then
  begin
    if InitCommunication(1,9600,ComHandle) = 0 then
    begin
      PortErr:=False;
      mnuCOM1.Enabled := False;
      mnuCOM2.Enabled := False;
      mnuCOM3.Enabled := False;
      mnuCOM4.Enabled := False;
      mnuBaglan.Enabled := False;
      mnuBagKes.Enabled := True;
      mnuRFIDOku.Enabled :=True;
      mnuRFIDYaz.Enabled :=True;
    end
    else
    begin
      PortErr:=True;
      msg :='COM1 kapalı!';
    end;
  end
  else if Form1.mnuCOM2.Checked = True then
  begin
    if InitCommunication(2,9600,ComHandle) = 0 then
    begin
      PortErr:=False;
      mnuCOM1.Enabled := False;
      mnuCOM2.Enabled := False;
      mnuCOM3.Enabled := False;
      mnuCOM4.Enabled := False;
      mnuBaglan.Enabled := False;
      mnuBagKes.Enabled := True;
      mnuRFIDOku.Enabled :=True;
      mnuRFIDYaz.Enabled :=True;
    end
    else
    begin
      PortErr:=True;
      msg :='COM2 kapalı!';
    end;
  end
  else if Form1.mnuCOM3.Checked = True then
  begin
    if InitCommunication(3,9600,ComHandle) = 0 then
    begin
      PortErr:=False;
      mnuCOM1.Enabled := False;
      mnuCOM2.Enabled := False;
```

```

    mnuCOM3.Enabled := False;
    mnuCOM4.Enabled := False;
    mnuBaglan.Enabled := False;
    mnuBagKes.Enabled := True;
    mnuRFIDOku.Enabled := True;
    mnuRFIDYaz.Enabled := True;
end
else
begin
    PortErr:=True;
    msg :='COM3 kapalı';
end;
end
else if Form1.mnuCOM4.Checked = True then
begin
    if InitCommunication(4,9600,ComHandle) = 0 then
    begin
        PortErr:=False;
        mnuCOM1.Enabled := False;
        mnuCOM2.Enabled := False;
        mnuCOM3.Enabled := False;
        mnuCOM4.Enabled := False;
        mnuBaglan.Enabled := False;
        mnuBagKes.Enabled := True;
        mnuRFIDOku.Enabled := True;
        mnuRFIDYaz.Enabled := True;
    end
    else
    begin
        PortErr:=True;
        msg :='COM4 kapalı';
    end;
end;

    if (PortErr) then
        MessageDlgPos('Seri haberleşmede hata oluştu', mtError, [mbOK], 0,Form1.Left+x,Form1.Top+y);

end;
procedure TForm1.mnuBagKesClick(Sender: TObject);
begin
    PurgeComm(ComHandle,PURGE_RXCLEAR);
    PurgeComm(ComHandle,Purge_TXCLEAR);
    SetCommBreak(ComHandle);
    TimeDelay(100);
    CloseHandle(ComHandle);
    mnuCOM1.Enabled := True;
    mnuCOM2.Enabled := True;
    mnuCOM3.Enabled := True;
    mnuCOM4.Enabled := True;
    mnuBaglan.Enabled := True;
    PortErr:=True;
    mnuRFIDOku.Enabled :=False;
    mnuRFIDYaz.Enabled :=False;
    mnuBagKes.Enabled := False;
end;

procedure TForm1.mnuRFIDOkuClick(Sender: TObject);
var
    sonuc,RFIDkulakno:String;
    formelements:OleVariant;
begin
    MRPCOM(ComHandle,$02,$08,$00,' 32',ErrNo,MRPRxDat);
    if ( ErrNo = 0 ) then

```

```

begin
  HandleResponse(1,sonuc);
  RFIDkulakno :='TR'+ilkoducoz(sonuc)+hayvankoducoz(sonuc);
  if(pos('ana.php',WebBrowser1.OleObject.Document.URL)<>0) then
  begin
    if (pos('hayvanliste.php',WebBrowser1.OleObject.Document.Frames.Item(1).Document.URL)<>0)
  then
    begin
      formelements:=WebBrowser1.OleObject.Document.Frames.Item(1).Document.Forms.Item('form1').elements;
      formelements.item('kulakno').value :=RFIDkulakno;
      formelements.item('hayvanekle').Click;
      end;
    end;
  end;
  else
    MessageDlgPos('Seri haberleşmede hata oluştu', mtError, [mbOK], 0,Form1.Left+x,Form1.Top+y);
  end;
  procedure TForm1.mnuRFIDYazClick(Sender: TObject);
  var
    sonuc,RFIDkulakno:String;
    formelements:OleVariant;
  begin
    if(pos('ana.php',WebBrowser1.OleObject.Document.URL)<>0) then
    begin
      if (pos('hayvandetay.php',WebBrowser1.OleObject.Document.Frames.Item(1).Document.URL)<>0)
  then
    begin
      formelements
:=WebBrowser1.OleObject.Document.Frames.Item(1).Document.Forms.Item('myform').elements;
      RFIDkulakno :=formelements.item(0).Value;
      RFIDkulakno :=ISO11784(copy(RFIDkulakno,3,2),copy(RFIDkulakno,5,9));
      end;
    end;

    if Data_Valid(RFIDkulakno) then
    begin
      MRPPProgData(ComHandle,StrPCopy(ProgDat,RFIDkulakno),MRPRxDat,ErrNo);
      if ( ErrNo = 0 ) then
        HandleResponse(2,sonuc)
      else
        MessageDlgPos('Seri haberleşmede hata oluştu', mtError, [mbOK],
0,Form1.Left+x,Form1.Top+y);
      end
    else
      MessageDlg('Hatalı veri girişi yaptınız!',mtError, [mbOK], 0);

  end;
  procedure TForm1.BitBtn1Click(Sender: TObject);
  begin
    WebBrowser1.Navigate(Edit1.Text);
  end;

  procedure TForm1.ToolButton2Click(Sender: TObject);
  begin
    WebBrowser1.GoForward;
  end;
  procedure TForm1.ToolButton1Click(Sender: TObject);
  begin
    WebBrowser1.GoBack;
  end;

end.

```

ÖZGEÇMİŞ

1978 yılında Ankara'da doğdu. İlk, Orta ve Lise öğrenimini Ankarada tamamladıktan sonra 1994-1998 yıllarında staj yaptığı şirket olan Aselsan A.Ş'de 4 sene Elektronik Teknisyeni olarak görev yaptı. 1998 yılında Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, Bilgisayar Sistemleri Anabilimdalına girdi, 2002 yılında mezun oldu. Aynı yıl Çankırı İli, Çerkeş ilçesi, 19 Mayıs Çok Programlı Lisesinde öğretmenlik görevine atandı. 2003 yılında Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi Anabilim Dalında yüksek lisansa başladı. 2004 yılında Ankara ili, Kalecik ilçesi, Kalecik Mesleki ve Eğitim Merkezine atandı. Aynı yıl görevlendirmeyeyle Milli Eğitim Bakanlığı, Eğitim Teknolojileri Genel Müdürlüğü, Bilgi İşlem Daire Başkanlığına geçti. Halen bu kurumda görevine devam etmektedir.