

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**İTÜKART
KULLANICI OTOMASYONU**

**Y.Ü. YÜKSEKÖĞRETİM ANKARA
DOKÜMANTASYON MERKEZİ**

**YÜKSEK LİSANS TEZİ
Müh. E. Ayşe ÇÖLOĞLU
(504971603)**

104059

**Tezin Enstitüye Verildiği Tarih : 11 Haziran 2001
Tezin Savunulduğu Tarih : 29 Haziran 2001**

Tez Danışmanı :

Doç.Dr. A. Coşkun SÖNMEZ

Diğer Jüri Üyeleri

Prof.Dr. Bülent ÖRENCİK

Yrd.Doç.Dr. Selçuk PAKER

HAZİRAN 2001

ÖNSÖZ

İçerdiği bellek ve mikroişlemci sayesinde veri depolama ve işleyebilme yeteneğine sahip, kredi kartı boyutlarındaki akıllı kartlar özellikle sundukları yüksek güvenlik taşınabilirlik ve kolay kullanılabilirlik nedeniyle günümüzde oldukça geniş uygulama alanı bulmaktadırlar.

Ülkemizdeki akıllı kart kullanımının hızla yaygınlaşmasına katkıda bulunmak amacıyla, bir örnek teşkil edeceği düşünülerek üniversitelere yönelik bir kimlik kartı uygulaması prototipi olan İTÜKart projesinin hazırlanmasına karar verilmiştir.

Tez çalışması boyunca yardımlarını esirgemeyen değerli danışmanım Sayın Doç. Dr. A. Coşkun SÖNMEZ'e, uygulamanın geliştirilmesi aşamasında destek veren Sayın Güney SEYHAN nezdinde KoçSistem Bilgi ve İletişim Hizmetleri A.Ş.'ye, sevgili ablam Banu PAMİR'e, Murat TÜRK'e, Süleyman SATAL'a ve bu zorlu çalışma sırasında her zaman yanımda olan aileme katkılarından dolayı teşekkür ederim.

Haziran, 2001

Ayşe ÇÖLOĞLU

İÇİNDEKİLER

KISALTMALAR	V
TABLO LİSTESİ	VII
ŞEKİL LİSTESİ	IX
SEMBOL LİSTESİ	X
ÖZET	XI
SUMMARY	XIII
1. GİRİŞ	1
2. AKILLI KARTLAR	4
2.1. Neden Akıllı Kartlar?	4
2.2. Akıllı Kart Tipleri	6
2.3. Güvenlik Mekanizmaları	8
2.4. Akıllı Kartların Uygulama Alanları	9
3. İTÜKART UYGULAMASI	11
3.1. Yazılım ve Donanım Gereksinimleri	13
3.2. Kullanılan Teknolojiler ve Diller	13
3.3. İşlevler	14
3.4. Veritabanı	17
3.4.1. Tablolar	17
3.4.2. Bağımlılıklar	18
3.5. Güvenlik	18
3.6. ITUCard.dll Sınıfları	20
3.6.1. Class: DB	20
3.6.2. Class: Ders	21
3.6.3. Class: Genel	24
3.6.4. Class: Kullanici	28
3.6.5. Class: Parametre	33
3.7. Web Uygulaması Sayfaları	34
3.7.1. Giriş Sayfası	35
3.7.2. Ana Sayfa	37
3.7.3. Başlık Sayfası	39
3.7.4. Menü Sayfası	40
3.7.5. Açılış Sayfası	41
3.7.6. Ders Kayıt Sayfası	42
3.7.7. Ders Bilgileri Detay Sayfası	47
3.7.8. Ders Kaydı Ekleme Sayfası	50

3.7.9.	Ders Kaydı Silme Sayfası	51
3.7.10.	Haftalık Ders Programı Sayfası	53
3.7.11.	Kullanıcı Bilgileri Detay Sayfası	56
3.7.12.	Not Çizelgesi Sayfası	60
3.7.13.	Ödeme Özeti Sayfası	63
3.7.14.	Rezervasyon Yaptırma ve İzleme Sayfası	65
3.7.15.	Randevu Kaydı Ekleme Sayfası	68
3.7.16.	Rezervasyon Kaydı Silme Sayfası	69
3.7.17.	Ödeme Yapma Sayfası	71
3.7.18.	Ödeme Kaydı Ekleme Sayfası	73
4.	SONUÇLAR VE TARTIŞMA	75
	KAYNAKLAR	78
	EKLER	80
	Ek.A. GemClub Akıllı Kartı	80
	Ek.B. Veritabanı Yapısı	114
	ÖZGEÇMİŞ	120

KISALTMALAR

ADO	: ActiveX Data Object
APDU	: Application Protocol Data Unit
ASN.1	: Abstract Syntax Notation 1
ASP	: Active Server Pages
BBS	: Blok Bekleme Süresi
BKZ	: Blok Koruma Zamanı
CAD	: Card Accepting Device
CEN	: Comité Européen de Normalisation
COM	: Component Object Model
CPU	: Central Processing Unit
CRC	: Cyclic redundancy check
CTC	: Card Transaction Counter
DES	: Data Encryption Standard
EEPROM	: Electrically Erasable Programmable Read Only Memory
EPROM	: Electrically Programmable Read Only Memory
ETSI	: European Telecommunications Standards Institute
ETU	: Elementary Time Unit
IC	: Integrated Circuit
ICC	: Integrated Circuit Card
IEC	: International Electrotechnical Commission
IFD	: Interface Device
IIS	: Internet Information Server
JTC1	: Joint Technical Committee
KBS	: Karakter Bekleme Süresi
LRC	: Longitudinal redundancy check
MAC	: Message Authentication Code
MİB	: Merkezi İşlem Birimi
OSI	: Open System Interconnection
OTP	: One-time Programmable
PIN	: Personal Identification Number
PKB	: Protokol Kontrol Byte'i
PROM	: Programmable Read Only Memory
PSE	: Payment System Environment
PVC	: Poli vinil klorür
RAM	: Random Access Memory
ROM	: Read Only Memory
SC	: Sub Committee
SFI	: System File Identifier
SIM	: Subscriber Identification Number

SIO	: Serial Input Output
TF	: Task Force
TLV	: Tag Length Value
TPDU	: Transport Protocol Data Unit
WG	: Working Group



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1. Giriş Sayfası Bileşenleri.....	35
Tablo 3.2. Ana Sayfa Bileşenleri.....	37
Tablo 3.3. Başlık Sayfası Bileşenleri.....	39
Tablo 3.4. Menü Sayfası Bileşenleri.....	40
Tablo 3.5. Açılış Sayfası Bileşenleri.....	41
Tablo 3.6. Ders Kayıt Sayfası Bileşenleri.....	43
Tablo 3.7. Ders Bilgileri Detay Sayfasının Bileşenleri.....	48
Tablo 3.8. Haftalık Ders Programı Sayfası Bileşenleri	54
Tablo 3.9. Kullanıcı Bilgileri Detay Sayfasının Bileşenleri.....	57
Tablo 3.10. Not Çizelgesi Sayfasının Bileşenleri.....	61
Tablo 3.11. Ödeme Özeti Sayfasının Bileşenleri.....	63
Tablo 3.12. Rezervasyon Sayfasının Bileşenleri.....	66
Tablo 3.13. Ödeme Yapma Sayfasının Bileşenleri.....	72
Tablo A.1. Dosya Tipleri	82
Tablo A.2. Sistem Dosyası Yapısı.....	84
Tablo A.3. Sayaç Dosyası Yapı Alanı.....	85
Tablo A.4. Sayaç Dosyası Yapısı.....	85
Tablo A.5. Sayaç Dosyası Seçimlik Özellikler.....	86
Tablo A.6. GemClub Tarih Dönüşümü.....	87
Tablo A.7. Kural Dosyası Yapısı.....	88
Tablo A.8. Kural Dosyasına Başka Makrokomut Eklenmesi.....	88
Tablo A.9. Makrokomut Yapısı.....	89
Tablo A.10. Gizli Kod Dosyasının Yapısı.....	90
Tablo A.11. Gizli Anahtar Dosyasının Yapısı.....	91
Tablo A.12. Kayıt Dosyasının Yapısı.....	93
Tablo A.13. EMV_DIR Dosyasının Yapısı.....	94
Tablo A.14. Sistem Dosyası Başlangıç Parametreleri.....	96
Tablo A.15. Anahtar Dosyası Parametreleri.....	96
Tablo A.16. Erişim Koşulları.....	96
Tablo A.17. EEPROM Büyüklüğü.....	97
Tablo A.18. Erişim koşulları için kullanılan fonksiyonlar.....	99
Tablo A.19. Erişim koşullarında yer alan kodlar için kurallar.....	100
Tablo A.20. Dolaylı Erişim Koşuluna Sahip Veri Elemanları.....	100
Tablo A.21. Ödül işlemlerinde kullanılan MAC değerleri.....	107
Tablo A.22. İndirim işlemlerinde kullanılan MAC değerleri.....	108
Tablo A.23. Kural Kullanma işlemlerinde kullanılan MAC değerleri.....	109

Tablo A.24.	Diğer işlemlerde kullanılan MAC değerleri.....	110
Tablo A.25.	Önceden Tanımlanmış Etiket Değerleri.....	111



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Akıllı Kartların Boyutları.....	1
Şekil 2.1. En popüler olan kimlik doğrulama yöntemlerinin güvenlik/maliyet açısından karşılaştırılması.....	5
Şekil 2.2. Kontaklı kart konnektörleri.....	7
Şekil 3.1. İTÜKart Genel Sistem Mimarisi.....	12
Şekil 3.2. Yazılım fonksiyonlarının genel giriş-çıkış şeması.....	16
Şekil 3.3. İTÜ Veritabanı Yapısı.....	19
Şekil 3.4. Giriş Sayfası.....	35
Şekil 3.5. Ana Sayfa	37
Şekil 3.6. Ders Kaydı Sayfası.....	42
Şekil 3.7. Ders Bilgileri Detay Sayfası.....	47
Şekil 3.8. Haftalık Ders Programı Sayfası.....	53
Şekil 3.9. Kullanıcı Bilgileri Sayfası.....	56
Şekil 3.10. Not Çizelgesi Sayfası.....	60
Şekil 3.11. Ödeme Listesi Sayfası.....	63
Şekil 3.12. Rezervasyon Sayfası.....	65
Şekil 3.13. Kurum – Ödeme Sayfası.....	71
Şekil A.1 Dosya Yapısı.....	81
Şekil A.2 GemClub Tarih Gösterimi.....	87
Şekil A.3 Değişken uzunluklu kayıtlar içeren doğrusal dosya.....	92
Şekil A.4 Başlangıçtaki Dosya Yapısı.....	95
Şekil A.5 Seri Numarası.....	97
Şekil A.6 Erişim koşulu yapısı.....	99
Şekil A.7 Parametre Güncelleme için güvenli mesajlaşma.....	102
Şekil A.8 İşlem Sınama.....	104
Şekil A.9 3DES Şifreleme / Şifre Çözme Algoritması.....	106
Şekil A.10 MAC Hesabı.....	107

SEMBOL LİSTESİ

- V_{CC} : Kart besleme gerilimi
 V_{SS} : V_{CC} besleme geriliminin ölçümünde kullanılan toprak seviyesi gerilimi
 V_{PP} : Karttaki EPROM belleğin programlanması sırasında kullanılan yüksek gerilim.



İTÜKART KULLANICI OTOMASYONU

ÖZET

Akıllı kart, kredi kartlarınıninkine benzer fiziksel özelliklere sahip olan ve kart gövdesinde yer alan tümleşik mikroişlemci sayesinde içerisinde bilgi saklayıp işleyebilen kartlara verilen genel isimdir. Akıllı kartların içindeki bu kırımlarda değişik amaçlarla, değişik tipte veriler saklanabilir; alışverişte kullanılmak üzere para taşınabileceği gibi, sağlık ile ilgili bilgiler ve uygulamalar da güvenli bir şekilde tutulabilir.

Günümüzde akıllı kartlar kimlik tanıma, bankacılık, elektronik cüzdan, ulaşım gibi pek çok alanda yaygın olarak kullanılmaktadır. Ülkemizde de elektronik cüzdan ve ulaşım alanlarında kullanılmaya başlanan akıllı kartların giderek yaygınlaşması beklenmektedir.

Akıllı kartlar, manyetik kartlardan bin kat daha fazla bilgi depolayabilirler. Daha güvenilir oldukları gibi, birden fazla fonksiyonu gerçekleyebilmektedirler. Ayrıca ileri seviyedeki kontrol mekanizmaları sayesinde daha güvenlidirler.

Ülkemizde gittikçe yaygınlaşan kullanıma sahip akıllı kart uygulamalarına bir örnek oluşturması amacıyla tez kapsamında İTÜKart projesi hazırlanmıştır. İTÜKart, üniversitelerde öğrenci ve öğretim üyelerine yönelik kimlik ve elektronik cüzdan uygulamasının bir prototipi olarak gerçekleştirilmiştir. Sistemin birimlerine erişim için İTÜKart sahibi olmak gereklidir. Proje iki alt bölümden oluşmaktadır: İTÜKart Yönetim Otomasyonu ve İTÜKart Kullanıcı Otomasyonu [1].

İTÜKart Yönetim Otomasyonu'nda, İTÜKart sisteminin çalışması için gerekli olan ve sadece yetkili kişilerce yapılabilecek tanımlamalar yer almaktadır. Bu program ile girilen veriler, Kullanıcı Otomasyonu ile ortak kullanılan veritabanında tutulur [1].

İTÜKart Kullanıcı Otomasyonu'nda ise, kullanıcıların web ortamında, üzerinde kart okuyucu bulunan herhangi bir makinedeki tarayıcı ile erişebileceği ders, randevu ve ödeme işlemleriyle ilgili ekranlar bulunmaktadır. Bu program, öğrenciler için

- Derse kayıt yaptıрма,
- Ders programı görüntüleme,
- Geçmiş ve mevcut ders durumlarını izleme,
- Laboratuvar ve sınıf rezervasyonu yapma,

imkanı sağlayacak şekilde geliştirilmiştir. Ayrıca tüm kullanıcılar için,

- Özlük bilgilerini görüntüleme,
- Ödeme bilgilerini izleme,
- Şifre değiştirme

fonksiyonları yazılmıştır. Bunların yanısıra, kurumlarda kullanılmak üzere, ödeme işlemlerinin yapılabildiği bir uygulama da aynı otomasyon sistemi içinde ele alınmıştır.

Bu programlardaki akıllı kart ile kart okuyucunun iletişimini sağlamak üzere bir arabirim yazılımı da gerçekleştirilmiştir. Bu yazılımdaki fonksiyonlar çağırılarak karttaki bilgiyi okumak ve istenen bilgiyi karta aktarmak mümkündür.

İTÜKart uygulamasında kullanılan GemClub kartı üzerinde kullanıcı numarası ve kullanıcının o an sahip olduğu kredi tutarı saklanmaktadır. Verilerin güvenliği için kartın kendisi tarafından sağlanan PIN (Personal Identification Number) mekanizmasından yararlanılmıştır. Kartta tutulan her iki verinin okunması işlemi PIN, güncellenmesi işlemi ise hem PIN, hem de ilgili kayıt dosyasına ait gizli anahtar (secret key) ile korunmaktadır. Ayrıca PIN'in değiştirilmesi için de farklı bir gizli anahtar kullanılmaktadır.

Kart güvenliğinin yanısıra, veritabanında tutulan, kart sahibine ait bilgilerin güvenliği de şifrelenmiş olarak saklanan kullanıcı parolaları ile sağlanmaktadır. Böylece uygulamadaki iki katmanlı koruma mekanizması ile veri güvenliğinin artırılması amaçlanmıştır.

Modüler olarak tasarlanan İTÜKart uygulaması bir prototip olarak hazırlandığından sadece örnek teşkil edecek birkaç temel fonksiyon ele alınmıştır. Bu yapı üzerinde, ileride ortaya çıkabilecek yeni ihtiyaçlar doğrultusunda kapsamın kolaylıkla genişletilmesi mümkündür.

İTÜKART USER AUTOMATION

SUMMARY

Smart Card is the general term, which is used for cards physically similar to credit cards, having an embedded microprocessor enabling it to store and process data. Different types of data, as well as applications, can be stored in a secure way in these chips for different purposes such as keeping credit for purchase or information about the cardholder's background concerning health.

Nowadays, smart cards are widely used in many areas like identification, banking, electronic purse and transportation. In Turkey, applications in electronic purse and transportation have started to show up in market, therefore smart card usage is expected to accelerate.

Smart cards can store approximately a thousand times more data compared to magnetic cards. They are more reliable and can support multiple functions. Moreover they are more secure with their advanced security mechanisms.

The İTÜKart project is developed as a part of this thesis for providing an example to smart card applications. İTÜKart is a prototype for a university card to be used by students, faculty, and other academic personnel as an id card and an electronic purse. It is required to have an İTÜKart card to access the units in the system. The project consists of two parts: İTÜKart Administration Automation and İTÜKart User Automation [1].

With İTÜKart Administration Automation, authorized users can access the program using their İTÜKarts and can make and maintain the definitions required for the system. The data is stored on a common database also used by the User Automation [1].

İTÜKart User Automation consists of screens, which are accessed through internet via terminals with smart card readers and internet browsers. With the help of these screens the students can

- Select lessons,
- View their schedules,
- Display previous or current status and grades,
- Reserve a lab or a classroom, etc.

Moreover all users can

- Display their own user information,
- View payment information,
- Change their passwords.

Apart from these functions, the payment application, which will be used by the corporations within the campus, is a part of the user automation.

To provide a means of communication between the card and the reader an interface software is also developed during this thesis work. It is possible to read or update the data on the card by calling the methods of this interface.

In the İTÜKart application, the user id of the cardholder and the amount of credit that the cardholder has are stored on the GemClub card. The PIN mechanism of the card itself is used for guaranteeing data security. Read operation on these data is protected by PIN whereas the update operation is protected by both the PIN and the secret key related to the record file. Furthermore a different secret key is required for updating the PIN on the card.

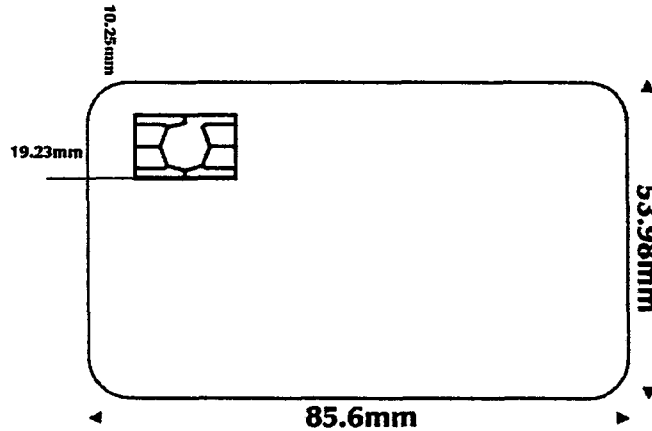
The security of the data in the database, which is directly related to the cardholder, is accomplished through the encrypted user password in the user definition. This two-layered protection mechanism used in the application ensures data security.

Because the system is designed as a prototype on a modular basis, only some basic functions of university cards are implemented. This structure can easily be expanded to meet future needs.

1. GİRİŞ

Her geçen gün gelişen bilgi teknolojileri, günlük hayatı hızla elektronik platforma taşımaktadır. Türkiye’de de örneklerini sıkça görmeye başladığımız elektronik ticaret, elektronik bankacılık, elektronik bilet (Akbil, OGS, vb), elektronik kimlik gibi uygulamaların hayata geçmesi bir çok avantajı beraberinde getirmiştir. Hız, kolaylık, yüksek erişilebilirlik, düşük maliyet gibi faydalarının yanında bu tür sayısal uygulamalarda en büyük zorluk kullanıcı kimliğinin güvenilir bir şekilde doğrulanmasında ortaya çıkmaktadır.

Elektronik uygulamalarda güvenlik gerektiren her hangi bir işlem için talepte bulunan kişinin yetkili olup olmadığının anlaşılması amacıyla manyetik ve biometrik tanıma yöntemleri kullanılmaktadır. Kredi kartı gibi manyetik ürünler ucuz ve yaygın olmalarına rağmen yeterli güvenlik sağlayamamaktadırlar ve sahtekarlığa açıktırlar. Parmak izi, retina tarama gibi yüksek teknoloji ürünü olan biometrik doğrulama



Şekil 1.1. Akıllı Kartların Boyutları

yöntemleri çok daha güvenli olmalarına rağmen, yüksek maliyetleri nedeniyle tercih edilmemektedir.

Mevcut kimlik doğrulama yöntemleri içinde güvenlik/maliyet oranı açısından en verimli çözümü akıllı kartlar sunmaktadır. Şekil 1.1’de de görüldüğü üzere, fiziksel olarak manyetik kartları andıran ve benzer PIN mekanizması kullanan akıllı kartlar, içerdikleri bellek ve işlemci sayesinde sadece veri depolama ile sınırlı kalmayıp işlem yapma yeteneğine de sahiptir. Akıllı kartların bu özelliği kullanılarak hem depolanan veri, hem kartı kullanmak için gerekli olan PIN, hem de okuyucu ile kart arasındaki iletişim şifrelenir. Böylelikle kart üzerinde yapılabilecek sahtekarlık neredeyse imkansız hale gelmiş olur.

Akıllı kartların, yüksek güvenlik sağlayan diğer yöntemlere göre daha yaygın kullanılmalarının ana sebebi alternatiflerine göre çok düşük maliyetli olmalarıdır. Akıllı kart uygulamaları, kart, okuyucu ve uygulama maliyetleri göz önüne alındığında biometrik yöntemlere göre çok daha az bir bütçe ile gerçekleştirilebilirler.

Tez kapsamında geliştirilmesi düşünülen uygulama konusu olarak, dünyada hızla yaygınlaşmasına rağmen Türkiye’de çok fazla örneği bulunmayan üniversite kartı seçilmiştir. Üniversite bünyesindeki herkesi kapsayan bu uygulama ile elektronik kimlik, elektronik cüzdan, öğrenci durum takibi, akademik takvimin ders programı detayına kadar belirlenebilmesi, sınıf/laboratuvar rezervasyonu gibi işlemlerin gerçekleştirilebildiği çok işlevli bir bütünün prototipi olarak İTÜKart sistemi hazırlanmıştır.

İTÜKart uygulaması ortak veritabanı üzerinde çalışan ve kart okuma-yazma işlemlerini aynı arabirim aracılığı ile gerçekleştiren bağımsız iki bölümden oluşmaktadır:

İTÜKart Yönetim Otomasyonu : Enstitü görevlisi, öğretim üyesi gibi yetkili kişilerce gerçekleştirilecek, İTÜKart sisteminden beklenen işlevler için gerekli olan tanımlama ve giriş işlemlerinin yapıldığı programdır. Sadece sistemde tanımlanmış yetkili kişiler

tarafından kullanılması amacına dayandığından istemci-sunucu mimarisi esas alınarak hazırlanmıştır [1].

İTÜKart Kullanıcı Otomasyonu : Kart okuyucusu, internet bağlantısı ve tarayıcısı bulunan herhangi bir terminalden, İTÜKart kullanıcılarının 7 gün, 24 saat erişim imkanıyla, kolaylıkla, hızla ve ucuza ulaşabildiği bu uygulama kullanılarak ders, randevu ve ödeme takibi ile ilişkili işlemler yapılabilmektedir.

Bu uygulamalarda akıllı kart ile kart okuyucu arasındaki iletişimi sağlamak üzere gerekli olan arabirim yazılımı da bir ActiveX kontrolü (ActiveX control) olarak tez kapsamında gerçekleştirilmiştir. Söz konusu kontrolün metodları sayesinde PIN kontrolü yapmak, PIN değiştirmek, karttaki bilgileri okumak veya istenen bilgileri karta aktarmak mümkündür. Genel amaçlı hazırlanan bu kontrol, farklı uygulamalar için de yeniden kullanılabilir niteliktedir.



2. AKILLI KARTLAR

Plastik kredi kartı ile aynı boyutlarda olan akıllı kartlar, kredi kartlarından farklı olarak tümleşik devreli kırımlar içerir. Akıllı kartlar, IC (Integrated Circuit) kartları veya ICC (Integrated Circuit Chip) kartları olarak da adlandırılırlar. Akıllı kartların fiziksel özellikleri ISO 7816 – 1 standartı tarafından belirlenmiştir [2, 3].

Tümleşik devreli kırımla sadece bellek bulunabileceği gibi, kırıml PIN kotrollü erişim güvenliğini sağlayan bir mantıksal devre tarafından da denetlenebilir [3]. Üçüncü bir seçenek de akıllı kartın mikroişlemci içermesidir. Bazı çevrelerde, mikroişlemcili kırıml içeren kartlara ‘Akıllı Kart’ denilse de tanım tam olarak açıklığa kavuşturulmadığından herhangi bir tümleşik devre içeren kartlar ‘Akıllı Kart’ olarak kabul edilebilir.

2.1. Neden Akıllı Kartlar?

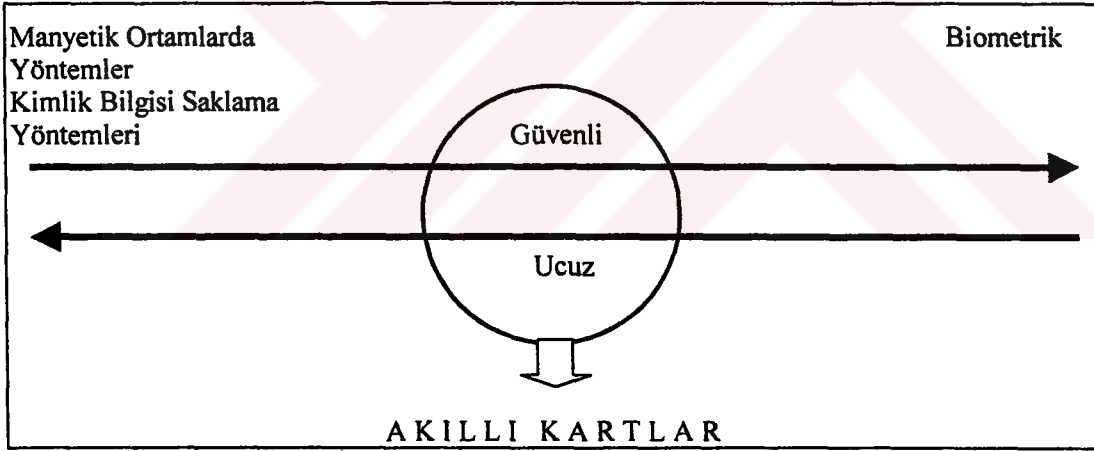
Özellikle son bir kaç yıl içinde hızla yaygınlaşan elektronik-iş uygulamalarında karşılaşılan en önemli problemlerden birisi olan kimlik doğrulama işlemi için güvenlik/maliyet açısından verimli bir çözüm bulma amacıyla yola çıkıldığında piyasadaki mevcut yöntemler içinde en tatmin edici olanların akıllı kart tabanlı sistemler olduğu görülmektedir.

Çözüm olacak araç veya teknolojinin aşağıda listelenmiş olan kriterleri karşılaması gerektiği düşünülürse eğer bulunan optik, manyetik, elektromanyetik, biometrik temele dayalı yöntemler gibi bir takım çözümlere rastlanmaktadır.

- Taşınabilir olma: Son zamanlarda elektronik uygulamaların mümkün olduğunca mobil platformlara taşınmasıyla birlikte kullanılacak teknolojilerin kullandığı araçların da taşınabilir olması gerekliliği doğmuştur.

- Kişiyi özgü ayırtedici özelliklere sahip olma: Kimlik doğrulama işleminin doğal gereksinimlerinden biri olarak, bulunacak çözümün kişiyi özgü ayırtedici bir takım özellikler sunması bellendirilmiştir.
- Hızlı olma: Çözüm, mevcut iletişim yolları üzerinde darboğaza yol açmayacak kadar hızlı yanıt verebilecek durumda olmalıdır.
- Kolay kullanılabilir olma: Araç veya teknoloji her ne olursa olsun mutlaka kolay kullanılabilir olmalıdır; çünkü varolan e-iş uygulamalarında geçişle birlikte yeni uygulamaların geliştirilme hızına da bağlı olarak kısa süre içinde yaygınlaştırılması ihtiyacı söz konusudur.

Sözü geçen kriterler gözönünde bulundurulduğunda karşılaşılan mevcut teknolojiler güvenlik ve maliyet açısından karşılaştırıldıklarında iki uç noktada Şekil 2.1’de görüldüğü gibi manyetik ortamlarda kimlik bilgisi saklama yöntemleri ile biometrik yöntemlerin yer aldığından bahsedilebilir.



Şekil 2.1. En popüler olan kimlik doğrulama yöntemlerinin güvenlik/maliyet açısından karşılaştırılması

Şekil 2.1’de görülen sembolik skalada sadece manyetik ve biometrik yöntemlerin gösterilmiş olmasının nedeni, güvenlik ve maliyet kriterlerine göre iki uç noktayı temsil edebilme özelliklerinin yanında halihazırda kullanılmakta olan en popüler yöntemlerden olmalarıdır. Bunların dışında kimlik doğrulama amacına yönelik olarak barkod ve optik karakter tanıma (OCR – Optical Character Recognition) yöntemlerin yanısıra radyo

frekasıyla kimlik doğrulama gibi elektromanyetik yöntemlerden de bahsetmek mümkündür.

Manyetik yöntemler düşük maliyetleri ile dikkati çekmekte ancak sundukları güvenliğin bir çok profesyonel uygulama için yetersiz kalması nedeniyle yavaş yavaş piyasadaki yerlerini kaybetmektedirler. Biometrik yöntemler ise yüksek maliyetleri nedeniyle ancak ciddi seviyede güvenlik ve çok küçük hata paylarıyla çalışma gerektiren ortamlarda tercih edilmektedirler. İki uç noktanın arasında yer alan akıllı kart çözümleri ise manyetik kartlara göre sundukları yüksek güvenlik ve biometrik yöntemlere göre sahip oldukları düşük maliyetler nedeniyle son yıllarda hızla yaygınlaşmaktadır. Akıllı kartlar, diğer yöntemlerden farklı olarak yüksek miktarda veri depolayabilme ve işlem yapabilme yetenekleri sayesinde de avantajlı duruma geçmektedirler.

2.2. Akıllı Kart Tipleri

Akıllı kart, kredi kartlarınıninkine benzer fiziksel özelliklere sahip olan ve plastik kart gövdesinde yer alan tümleşik devre sayesinde bilgi saklayıp işleyebilen kartlara verilen genel isimdir.

Akıllı kartları içeriklerine ve iletişim şekillerine göre iki ayrı kategoride incelemek mümkündür.

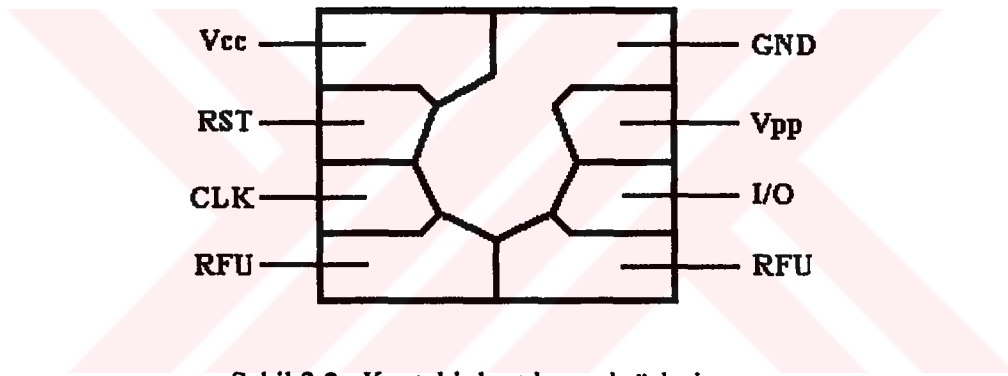
İçeriklerine Göre Akıllı Kartlar:

- **Bellek Kartları:** Bellek kartlarında plastik kart gövdesi içinde sadece bir bellek yer alır. Bu kartlar veri depolama amaçlı kullanılırlar.
- **Lojik Devreli Kartlar:** Lojik devreli kartlarda belleğin yanısıra mantıksal bir devre de yer alır. Böylelikle basit işlemleri gerçekleştirme yeteneğine sahip duruma gelirler.
- **İşlemcili Kartlar:** Şifreleme gibi karmaşık işlemleri yerine getirebilecek kapasiteye sahip, zaman zaman işletim sistemi bile barındırabilen bu kartlar en

çok tercih edilen akıllı kart türüdür. ISO 7816 – 4’te işlemcili kart yapısı fazla detaylandırılmadan anlatılmaktadır [4, 5].

İletişim Şekillerine Göre Akıllı Kartlar:

- **Kontaklı Kartlar:** Plastik gövde içinde barındırdıkları kırılgının dış dünyayla iletişiminin kart yüzeyinde yer alan iletken konnektörler vasıtasıyla sağlandığı kartlardır. Bu işlem için, kart üzerindeki plakanın ISO 7816 – 2 standardında belirtilen konumda olması gerekir [3, 10]. Kontaklı kartlara göre avantajı, iletişim hattının dinlenmesi suretiyle verilerin yasal olmayan yollardan ele geçirilmesi ihtimalinin daha düşük olmasıdır. Şekil 2.2’de kontaklı bir akıllı kartın yüzeyindeki kontaklar gösterilmiştir.



Şekil 2.2. Kontaklı kart konnektörleri

Kontakların işlevleri aşağıda açıklanmıştır:

- **Vcc**, kırılgı besleyen güç kaynağıdır. Genellikle 5 voltur. İlerleyen teknoloji sayesinde tümleşik devre tarafından daha düşük akım çekilmesi mümkün oldukça bu gerilimin 3 volta çekilmesi düşünülmektedir.
- **GND** ise Vcc’nin ölçümünde kullanılan toprak seviyesini belirlemektedir.
- **Albaştan (Reset)** ile, karta ilk güç verildiğinde (power on) tümleşik devrenin başlangıç durumuna çekilmesi için gereken sinyal verilir.

- **Saat işareti (clock)**, tümleşik devrenin sürülmesinde ve seri haberleşme bağlantısında referans olarak kullanılır. Genelde kullanılan iki saat frekansı vardır: 3.5795 MHz ve 4.9152 MHz. Avrupa'da genelde 3.5795 MHz kullanılmaktadır; ama bu durum değişebilir. Bu şekilde küsuratlı frekansların seçilmiş olmasının altında yatan neden, televizyonlarda olduğu gibi, tümleşik devrelerde de kullanılmakta olan ucuz kristallerdir.
 - **Karttaki EPROM belleği** programlamada kullanılan yüksek gerilim V_{PP} kontağı ile sağlanır.
 - **Seri giriş/çıkış (SIO – Serial Input/Output)** konnektörü sayesinde kırkık dış dünyadan komut alabilir ve veri alışverişinde bulunabilir.
- **Kontaksız Kartlar:** Kontaksız kartların içinde yer alan tümleşik devrenin dış dünya ile iletişimi yine kart gövdesinin içinde yer alan anten vasıtasıyla sağlanır. Dolayısıyla bu kartların okuyucu/yazıcı terminale teması gerekli değildir. Toplu taşımacılık gibi hızlı akış gerektiren yerlerde avantaj sağlamaktadırlar. Günümüzde kontaksız kartların özellikleri ISO 10536 standartı ile belirlenmektedir [7].

2.3. Güvenlik Mekanizmaları

Akıllı kartlar içerdikleri bellek birimlerinde tuttukları verileri şifreyebildikleri gibi, iletişim sırasında aktarılan veriyi de şifreleme yeteneğine sahiptirler. Böylelikle, iletişim hattının dinlenmesi suretiyle bilgi ele geçirme ve sahtecilik büyük oranda engellenmiş olmaktadır.

Akıllı kartlarda her kart için bir özel bir gizli anahtar olan PIN kodunun tanımlanması yoluyla kartı koruma altına almak mümkündür. Böylelikle şifreye sahip olan kişiden başkasının kartın fonksiyonlarına erişmesi neredeyse imkansız hale gelir. Bunun dışında kullanılan ikincil anahtarlar ise karttaki kayıt dosyalarını korumak amacıyla

kullanıldıkları gibi, gerekli durumlarda, PIN kodunun korunması işlevini de yerine getirirler. Bu servisleri sağlamak işletim sisteminin görevidir. ISO 7816 – 7’de güvenlik için sunulan bu mekanizmalar detaylandırılmıştır [3, 8, 9, 11].

Akıllı kartların güvenlik açısından sunduğu bir avantaj, fiziksel yapıları gereği, kırmığının kart gövdesi içine gömülmüş olması sebebiyle kopyalama ve deneme-yanılma yöntemleriyle şifrelerin kırılması ve bilgilerin ele geçirilmesinde ortaya çıkardığı zorluktur.

Akıllı kartların güvenliğinin artırılması için biometrik yöntemlerle birleştirilerek karma yapıların kurulması mümkündür. Örneğin, bir akıllı kartın verilerine PIN koduyla değil, sahibinin parmak iziyle ulaşılması kısıtı getirildiğinde şifrenin kart sahibi dışındaki kişiler tarafında ele geçirilecek şekilde yayılabilmesi ihtimali de ortadan kalkmış olur. Güvenliğin artırılmasının bir başka yolu olarak da sayısal imzaların kullanılması yönteminden bahsetmek mümkündür.

2.4. Akıllı Kartların Uygulama Alanları

Akıllı kartlar ilk olarak telefon kartlarında ve bankacılık uygulamalarında kullanılmış olmakla birlikte, günümüzde aşağıdaki listede yer alan tüm uygulama türlerinde kullanım alanı bulmaktadırlar. Bu listenin zaman içinde daha da uzaması ve kullanım yoğunluğunun artması beklenmektedir [10, 11, 12].

- Telefon kartları
- Bankacılık işlemleri
- Mobil haberleşme
- Perakende ticaret
- Elektronik cüzdan
- Elektronik bilet

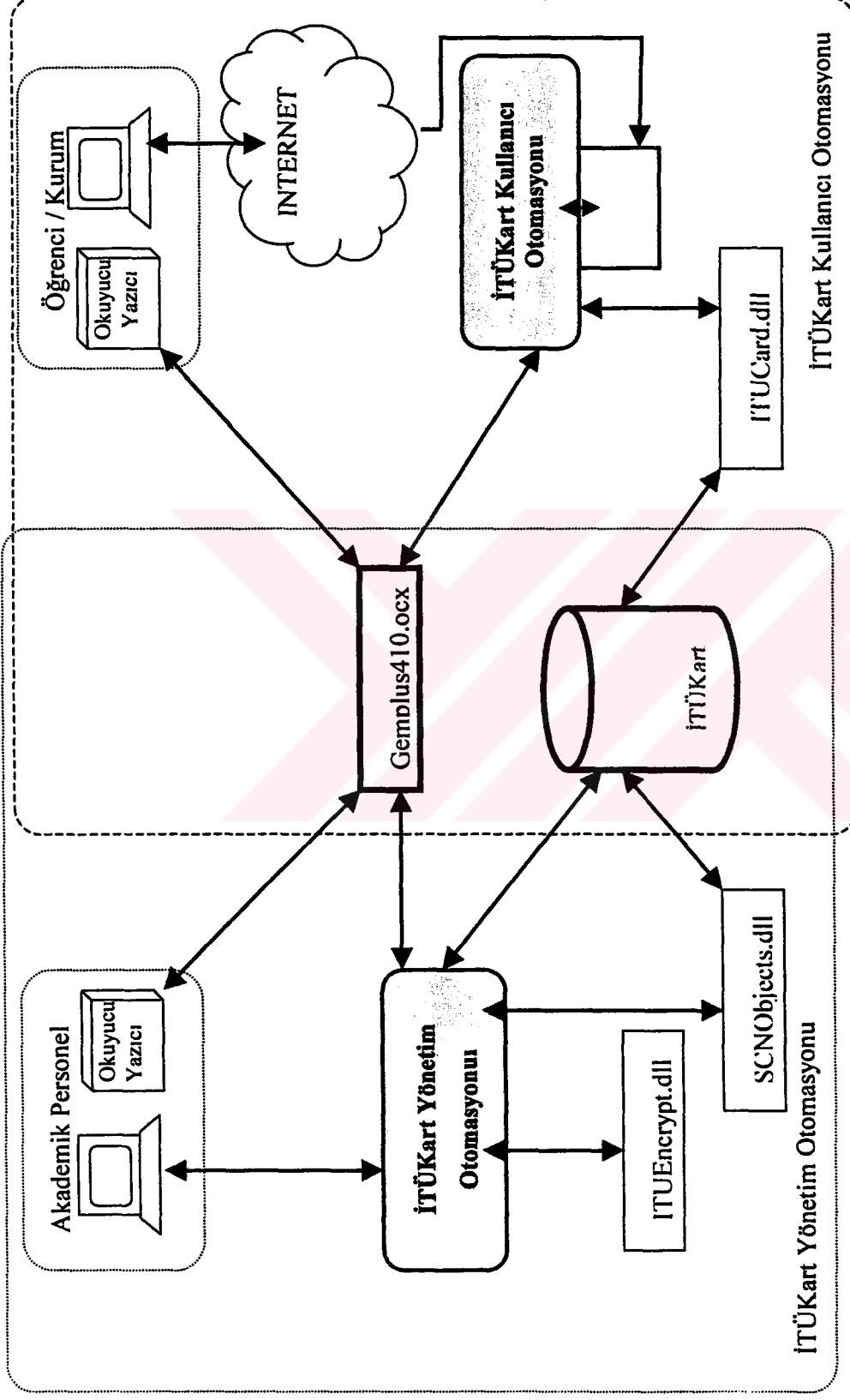
- Otoparklar
- Sağlık
- Ağ erişimleri



2. İTÜKART UYGULAMASI

Yüksek Lisans tezi kapsamında geliştirilen İTÜKart Projesi, üniversitelerde kullanılabilecek akıllı kart otomasyon sisteminin bir prototipidir. Öğrencilerle birlikte tüm üniversite kullanıcılarının üzerinde sadece bir akıllı kart okuyucu/yazıcı ve internet tarayıcı bulunan herhangi bir terminalden ders, sınav, rezervasyon, ödeme bilgileri ve kişisel bilgiler gibi sistemde kayıtlı olan kendilerine ait tüm verilere kolay, anında ve güvenli bir şekilde ulaşabilmelerini ve işlem yapabilmelerini sağlamak amacıyla geliştirilmiştir.

Bu tez kapsamında Kullanıcı Otomasyonu bölümü ele alınmıştır. Kullanıcı Otomasyonu iki kısımdan oluşmaktadır. Bunlardan birisi, akıllı kart okuyucu/yazıcı içeren terminaller üzerinde öğrencinin akıllı kartında tutulan numara ve şifresi ile sisteme giriş yaparak notlarını ve ders programını görmesi, laboratuvar rezervasyonu yaptırması, derslere kayıt olması, ödeme bilgilerine ulaşması ve şifresini değiştirebilmesi için gerekli işlemlerin gerçekleştirildiği öğrencilere yönelik ekranlardır. Diğeri ise sadece üniversite dahilindeki ticari kurumlarda kullanılacak olan, kartlarda tutulan kredi üzerinden eksiltme yapmak suretiyle ödeme fonksiyonunu gerçekleştiren kurumlara yönelik ekranlardır. İTÜKart Kullanıcı Otomasyonu, sadece fakülte ve enstitü yetkilileri tarafından kullanılabilecek Yönetim Otomasyonu ile birlikte çalışabilecek şekilde geliştirilmiştir. Bu tez kapsamında ele alınmamış olan ancak hazır bulunan Yönetim Otomasyonu'nda ise öğrenci tanımlama, öğretim üyesi tanımlama, ders tanımlama, ders notu girişi gibi tüm İTÜKart sisteminin çalışması için gerekli olan tanımlama ve veri girişlerinin yapılabileceği ekranlar yer almaktadır [1].



Şekil 3.1. İTÜKart Genel Sistem Mimarisi

Programlardan kart okuyucu/yazıcıya erişim, yine tez kapsamında geliştirilen GemPlus410 yazılımındaki fonksiyon ve prosedürlerin çağırılmasıyla gerçekleşmiştir

3.1. Yazılım ve Donanım Gereksinimleri

Web uygulamasının çalışacağı istemci bilgisayarların minimum 100 MHz hıza sahip Pentium (veya dengi) işlemcili, en az 64 MB RAM ve akıllı kart okuyucu/yazıcı kurulu bir harici com port içermesi gerekmektedir. Tez kapsamında akıllı kart okuyucu/yazıcı olarak GemPlus firması ürünlerinden GemCore 410 modeli kullanılmıştır. Kartlar ise 1K'lık bellek içeren GemClub kartlarıdır. Prototip için merkezi veritabanının bulunduğu bilgisayar ise Pentium (veya dengi) işlemcili, 166 MHz ve üstü işlemci hızına sahip, en az 128 MB RAM'e sahip olmalıdır. Bu makinada MS SQL Server 7.0'ın kurulması için 150 MB civarında ve bunun dışında bütün verileri tutabilecek büyüklükte sabit disk alanı gerekmektedir. MS SQL Server 7.0 kurmak için bilgisayarda Windows NT Server 4.0 veya Windows 2000 işletim sistemlerinden biri yüklü olmalıdır. MSSQL Server 7.0 yanında IIS (Internet Information Server)'ın en az 4.0 versiyonu bulunmalıdır. Sunucu makineye aynı zamanda akıllı kart üreticisi tarafından sağlanan kütüphane fonksiyonları ile veritabanı işlemleri için hazırlanan ActiveX DLL ve kart okuyucu/yazıcı arabirimi olan ActiveX Component (GemPlus410) yüklenmiş olmalıdır. Sistemin uygulamaya alınmasına karar verilirse, uygulama sunucusunun konfigürasyonu üniversitenin ihtiyaçları, veri miktarı, eklenen fonksiyonlara ve beklenen performansa bağlı olarak değişiklik gösterebilir.

3.2. Kullanılan Teknolojiler ve Diller

Akıllı kart okuyucu/yazıcı ve internet tarayıcı bulunduran terminallerde çalışacak Kullanıcı Otomasyonu (Öğrenci Ekranları ve Kurum Ekranları) web ortamında, ASP, HTML ve Dynamic HTML ile yazılmıştır. Sunucu tarafta ASP ve VBScript kullanılırken, istemci tarafta HTML ve JavaScript kullanılmıştır. Böylelikle uygulamaya yaygın olarak kullanılan tarayıcılardan erişim mümkün kılınmıştır.

Uygulamada iki seviyeli bir hiyerarşi kurulmuştur. Kullanıcı arayüzü ASP ile yazılırken, veritabanı işlemleri ve sık kullanılan fonksiyonlar seviyesinde yine bu tez kapsamında ActiveX DLL olarak hazırlanan COM nesnesinden yararlanılmıştır. COM teknoloji sayesinde fonksiyonların paketlenmesi sağlanarak, modülerlik, yeniden kullanılabilirlik ve güvenlik artırılmıştır. Fonksiyonlarda herhangi bir güncelleme gerektiğinde sadece ilgili COM nesnesinin değiştirilmesi yeterli olduğundan ileride yapılabilecek değişiklikler için kolay kullanılabilir bir alt yapı hazırlanmıştır. Bununla birlikte, paketlenmiş COM nesnesini benzer her türlü uygulamada da kullanmak mümkün hale gelmiştir. COM nesnesi üzerinde tanımlanabilecek yetki seviyeleri ile de eğer gerekirse güvenlik düzeyini kontrol etmek imkan dahilindedir. Veritabanına erişimler ADO (ActiveX Data Objects) teknolojisi ile sağlanmıştır.

İstemci makinelerde akıllı kart okuma/akıllı karta yazma işlemleri için kullanılan ve terminal ile kart okuyucu/yazıcı arasındaki arabirim görevini yerine getiren yazılım ise Visual Basic 6.0'da ActiveX Component olarak geliştirilmiştir. Bu bileşenin istemciden sunucuya istek iletildiği anda kullanıcının izni dahilinde istemciye yüklenmektedir ve bu işlemin bir kez gerçekleşmesi sonraki tüm kullanımlar için yeterli olmaktadır.

3.3. İşlevler

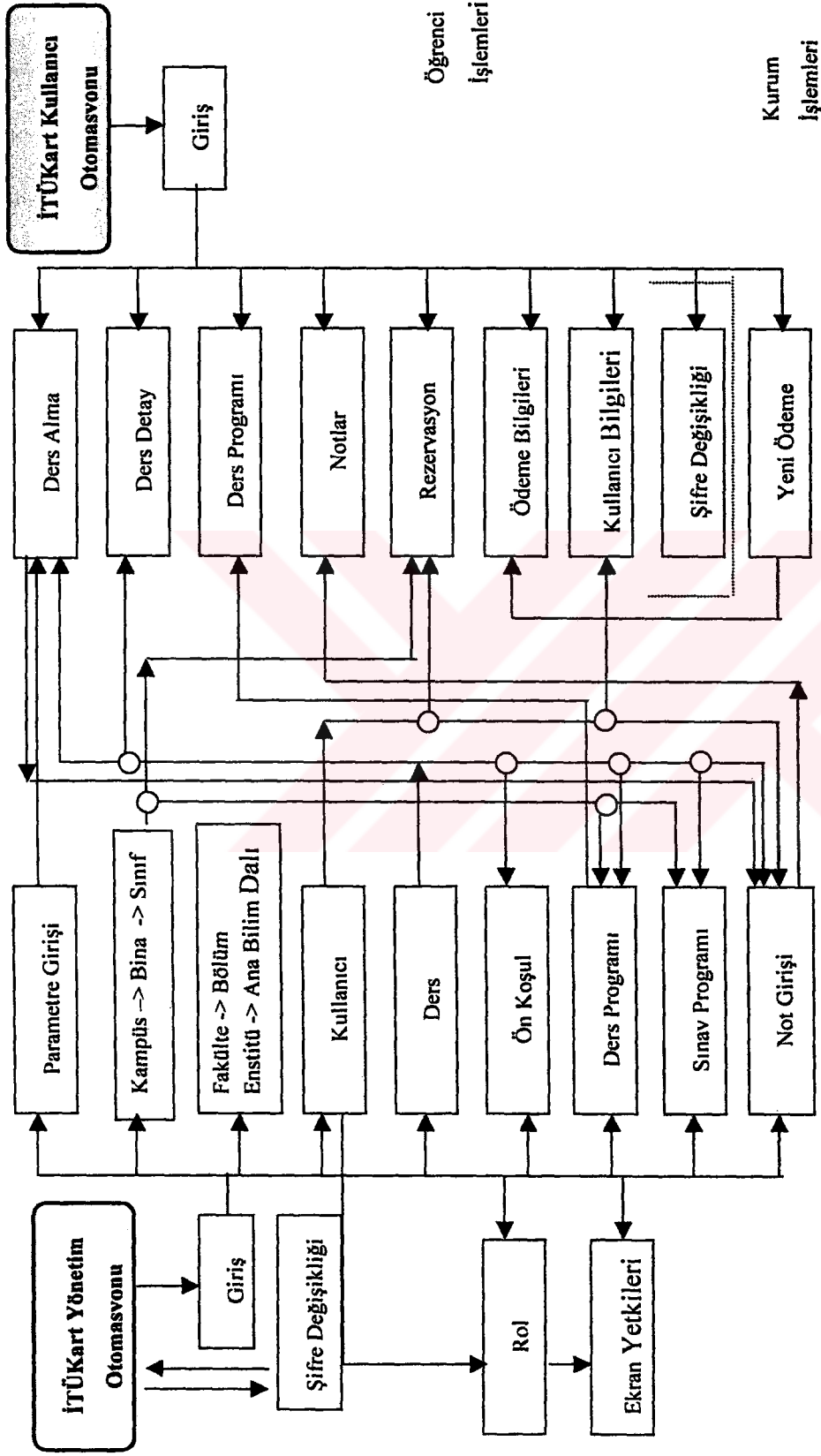
İTÜKart Kullanıcı Otomasyonu Öğrenci Ekranları'nda sunulan işlevler aşağıda sıralanmıştır:

- **Giriş Ekranı:** Siteye giriş sırasında güvenliğin artırılması amacıyla iki seviyeli bir kontrol mekanizması kullanılmaktadır. Kullanıcı öncelikle kartındaki PIN numarasını kartıyla birlikte sunar ve eğer verdiği PIN numarası tutarlıysa daha sonra veritabanında tutulan şifresini vermek suretiyle siteye giriş yapar.
- **Ders Kaydı:** Öğrencinin bulunduğu döneme kadar olan ve öğrencinin henüz almamış olduğu veya başarısız tamamladığı tüm zorunlu ve seçmeli dersler listelenerek öğrencinin ders seçmesine ve derse kayıt yaptırmaya olanak tanınır. Ders seçildiğinde, öğrencinin o anki programı ile ders programı karşılaştırılarak

herhangi bir akışmanın olmaması saėlanır. Bununla birlikte kayıt dnemi iinde bulunulup bulunmadıėına dair kontroller de yapılmaktadır.

Buradan aynı zamanda derslerin detay bilgilerine de ulaşılabilmektedir. Ders detay bilgilerinde dersin zellikleri yanında ėrencinin o derse iliřkin gemiři de grntlenmektedir. Bylelikle ėrenci dersi her alışına iliřkin hem bařarı durumunu hem de not detaylarını takip edebilmektedir.

- **Ders Programı:** ėrenci mevcut ders programını grebilmektedir. Buradan da ders bilgileri detay ekranına geiř mmkndr.
- **Notlar:** ėrenci mevcut dnemde almıř olduėu tm derslerine iliřkin notlarını buradan takip edebilmektedir.
- **Rezervasyon:** Bu iřlev sayesinde ėrencinin herhangi bir sınıfta veya laboratuarda ileri tarihli tek kiřilik rezervasyon yaptırmaya olanak tanınmıřtır. Bir gn ncesine kadar iptal etme hakkı da bulunduėundan, laboratuvar rezervasyonları bylelikle otomatize edilmiřtir. Rezervasyon gerekleřtirilmeden nce ders programları ile akışmalar ve kontenjan sınırları da kontrol edilmektedir.
- **demeler:** Bu iřlev ėrenciler tarafından olabileceėi gibi her niversite kullanıcısı tarafından da kullanılabilmektedir. Kullanıcı niversite dahilindeki ticari kurumlarda akıllı kartı ile yaptıėı tm harcamalarını buradan takip edebilmektedir.
- **Kullanıcı Bilgileri:** Kullanıcı sistemde kayıtlı kendisine ait zlk bilgilerini grntleyebilmektedir.
- **PIN/řifre Deėiřtirme:** Kullanıcının hem veritabanında tutulan řifresini hem de kart zerindeki PIN numarasını deėiřtirmesi mmkn olmaktadır.
- **ıkıř:** Uygulamadan gvenli bir ıkıř saėlamaktadır. ıkıř butonuna tıkladıėında, aktif kullanıcıya iliřkin baėlamda bulunan tm bilgiler silinip Giriř Sayfasına



Şekil 3.2. Yazılım fonksiyonlarının genel giriş-çıkış şeması

yönlendirme yapıldığından aynı tarayıcı penceresi bir sonraki kullanıcı tarafından güvenle kullanılabilir.

İTÜKart Kullanıcı Otomasyonu Kurum Ekranları'nda sunulan işlevler ise aşağıda verilmiştir:

- **Giriş Ekranı:** Siteye giriş sırasında güvenliğin artırılması amacıyla iki seviyeli bir kontrol mekanizması kullanılmaktadır. Kurum yetkilisi öncelikle kartındaki PIN numarasını kartıyla birlikte sunar ve eğer verdiği PIN numarası tutarlıysa daha sonra veritabanında tutulan şifresini vermek suretiyle siteye giriş yapar.
- **Yeni Ödeme:** Ödemeyi yapacak olan kullanıcı PIN numarasını girerek kartını okutur. Eğer verdiği PIN numarası geçerliyse kullanıcı numarası ve kartındaki mevcut kredi miktarı görüntülenir. Ardından veritabanında tutulan şifresini, alışveriş tutarını ve açıklamayı yazarak işlemi gerçekleştirir. Tutar, kartındaki kredi meblağından düşülür ve veri tabanında da ödemeye ilişkin bir kayıt yapılır.
- **PIN/Şifre Değiştirme:** Kurum yetkilisi hem veritabanında tutulan şifresini hem de kart üzerindeki PIN numarasını değiştirmesi mümkün olmaktadır.
- **Çıkış:** Uygulamadan güvenli bir çıkış sağlamaktadır. Çıkış butonuna tıklandığında, aktif kullanıcıya ilişkin bağlamda bulunan tüm bilgiler silinip Giriş Sayfasına yönlendirme yapıldığından aynı tarayıcı penceresi bir sonraki kullanıcı tarafından güvenle kullanılabilir.

3.4. Veritabanı

İTÜKart Yönetim Otomasyonu ve İTÜKart Kullanıcı Otomasyonu tarafından ortak olarak kullanılan merkezi veritabanı MS SQL Server 7.0 üzerinde yapılandırılmıştır.

3.4.1. Tablolar

İTÜKart projesinde Yönetim Otomasyonu [1] ile Kullanıcı Otomasyonu aynı veritabanı üzerinde çalışmaktadır. Otomasyonlar arası veri akışı veritabanında yer alan tablolar

üzerinden sağlanmaktadır. Ortak veritabanında Kullanıcı Otomasyonu tarafından kullanılan tabloların yapısı Ek.B’de verilmiştir.

3.4.2. Bağımlılıklar

İTÜKart Kullanıcı Otomasyonu tarafından kullanılan veritabanı dosyaları arasındaki bağımlılıklar Şekil 3.3’te verilmiştir.

3.5. Güvenlik

Uygulamada çok katmanlı bir güvenlik yapısı kurularak güvenlik arttırılmış, yasa dışı kullanım olasılığı minimum düzeye indirilmiştir.

Tasarlanan kart yapısında kayıt dosyalarını korumak amacıyla PIN kodundan faydalanılmakta, bunun yanısıra PIN değeri de bir gizli anahtar ile korunmaktadır. Dolayısıyla, karttaki bilgiyi okumak veya karta yazmak suretiyle PIN koduna ulaşmak ancak ve ancak bu gizli anahtar değerinin bilinmesi ile mümkündür.

Kartta tutulan PIN ve gizli anahtar şifrelerinin yanısıra uygulama sitesine giriş yapabilmek için kullanıcının ikinci bir şifreyi de girmesi zorunlu kılınmıştır. Bu şifre kart üzerinde değil veritabanında saklanmaktadır. Dolayısıyla, kartta herhangi bir şekilde (çok zor olsa da) değişiklik yapmayı başaran bir kişi bu sekiz karakterli ikincil şifreyi de ele geçirmelidir; aksi takdirde uygulama fonksiyonlarına erişmesi mümkün değildir.

Veritabanında şifreler açık şekilde tutulmamaktadır. Tez kapsamında tasarlanan DES benzeri simetrik bir algortima ile tüm şifreler ilgili tablolara şifrelenmiş şekilde yazılmaktadır. Bunun sonucu olarak, veritabanındaki şifreleri ele geçiren birisi, bir de

şifreyi gerçek değerlere ulaşabilmek için bir de şifrelenmiş parolaları çözecek algortimayı bulmak zorundadır. Uygulamada kullanılan şifreleme algoritması örnek teşkil etmek amacıyla hazırlanmıştır; güvenliği arttırmak için algortima değiştirilebilir veya şifre uzunluğu artırılabilir. Böylelikle, deneme yöntemi ile şifreleri çözmek daha da zorlaştırılabilir.

Tez kapsamında geliştirilen uygulama, sadece bir prototip niteliğine sahip olduğundan SSL gibi internet tarayıcısı seviyesinde güvenlik mekanizmaları kullanılmamıştır; ancak sisteme bu tarzda bir güvenlik katmanının eklenmesi son derece kolay ve ucuza mal edilebilir bir işlemdir. Bu sayede, siteye giriş işlemlerinde verilerin sunucuya iletilmesi sırasında hattın üçüncü bir şahıs tarafından dinlenmesi suretiyle kullanıcı numarası ve şifre gibi bilgilerin ele geçirilebilmesi ihtimali minimize edilebilir.

Bir kişinin, kart sahibi tarafından tüm şifreler kendisine söylenmedikçe, sisteme erişimi son derece zordur. Bunun için öncelikle karttaki PIN kodunu okuması gereklidir ki, PIN kodu okuma işlemi bir gizli anahtar ile korunmaktadır. Dolayısıyla gizli anahtarın da bilinmesi gerekir. PIN kodunu bir şekilde ele geçirdiği varsayılan bir kişinin uygulamayı kullanabilmesi için veritabanında tutulan sekiz karakterlik ikincil şifreyi de bilmesi gerekir. Veritabanı kayıtlarından bir şekilde şifre alanını okuyabilen bir kişinin ele geçireceği değer şifrelenmiş bir değer olduğundan bir de şifreyi çözecek algortimayı bulmak gerekecektir. Hattı dinlemek suretiyle istemci-sunucu arasındaki iletim sırasında şifre ve kullanıcı bilgilerine ulaşmak ise SSL benzeri şifreli haberleşme yöntemleriyle iyice zorlaştırılmaktadır.

3.6. ITUCard.dll Sınıfları

3.6.1. Class: DB

Veritabanına bağlanma işlemini gerçekleştirir. Connection ve Command nesnelerini geri döndürür. Sınıflardaki SQL cümleleri ile veritabanına erişim işlemleri sırasında, geri dönen Comm nesnesi kullanılır. O nedenle, her bir görüntü-nesne yaratıldığında

mevcut veritabanı bağlantısına ait Command nesnesi, görüntü-nesnenin ilgili özelliğine atanmalıdır.

Özellikler

DB Sınıfı arayüzüne ait tüm özellikler aşağıda listelenmiştir:

Public Comm As ADODB.Command: Veri tabanına bağlantı gerçekleştirildikten sonra tüm sınıflar tarafından kullanılacak olan Command nesnesi.

Public Conn As ADODB.Connection: Veri tabanına bağlantı gerçekleştirildikten sonra tüm sınıflar tarafından kullanılacak olan Connection nesnesi.

Public ErrNum As Integer: Hata numarası.

Public ErrorMessage As String: Hata mesajı.

Metodlar

DB Sınıfı arayüzüne ait tüm metodlar aşağıda listelenmiştir.

Private Sub Class_Initialize()

Sınıfın bir nesnesi yaratıldığında veri tabanına bağlantı kurulur. Conn ve Comm nesnelerine ilgili atamalar yapılır. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özellikleri ile geri döndürülür.

Private Sub Class_Terminate()

Sınıfın bir nesnesi yok edildiğinde veritabanına kurulmuş olan bağlantı sonlandırılır ve nesneler yok edilir.

3.6.2. Class: Ders

TBL_DERS ve ilgili tablolardaki alanlara ve kayıtlara erişim metodlarını içeren sınıftır.

Özellikler

Ders Sınıfı arayüzüne ait tüm özellikler aşağıda listelenmiştir:

Private Param As New ITUCard.Parametre: Parametre Sınıfı metodlarına erişim amacıyla sadece sınıf içinde kullanılan nesnedir.

Public Comm As ADODB.Command: Bu özelliğe instance yaratıldığı anda mevcut veri tabanı bağlantısı ile ilişkili olan command nesnesi atanmalıdır; çünkü metodlar veri tabanı işlemleri için bu değişkeni kullanmaktadırlar.

Public ErrNum As Integer: Hata numarası.

Public ErrMessage As String: Hata mesajı.

Metodlar

Ders Sınıfı arayüzüne ait tüm metodlar aşağıda listelenmiştir.

Public Function BolumDersleriniOku(BolumNo As Integer) As ADODB.Recordset

BolumNo argümanı ile verilen bölümün tüm derslerini TBL_DERS tablosundan okuyarak bir resultset içinde Donem, DersTipNo ve Ad alanlarına göre sıralı olacak şekilde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function DersBilgileriniOku(DersNo As Integer) As ADODB.Recordset

DersNo argümanı ile verilen dersin tüm alanlarını TBL_DERS tablosundan okuyarak bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

**Public Function OgrDersBilgileriniOku(OgrenciNo As Long, DersNo As Integer)
As ADODB.Recordset**

OgrenciNo argümanı ile verilen öğrencinin DersNo ile verilen derse ilişkin tüm kayıtlarını TBL_OGRDERS tablosundan okuyarak, yıl ve yarıyıla göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function DersPrograminiOku(Yil As String, YariYil As String, DersNo As Integer) As ADODB.Recordset

DersNo argümanı ile verilen dersin Yil ve YariYil argümanları ile belirtilen dönemdeki haftalık ders programını TBL_DERSPRG tablosundan okuyarak bir resultset içinde Gun ve Seans alanlarına göre sıralı olacak şekilde geri döndürür. Herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function DersAdi(DersNo As Integer) As String

DersNo argümanı ile verilen dersin adını TBL_DERS tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Private Sub Class_Initialize()

Sınıfın bir nesnesi yaratıldığında public sınıf özelliklerinden olan Comm nesnesi yaratılır.

Private Sub Class_Terminate()

Sınıfın bir nesnesi yok edildiğinde public ve private Comm, ADORS, Param nesneleri yok edilir.

3.6.3. Class: Genel

Diğer sınıflar içinde yer almayan, projenin genelinde kullanılan metodları içeren sınıftır.

Özellikler

Genel Sınıfı arayüzüne ait tüm özellikler aşağıda listelenmiştir:

Public Comm As ADODB.Command: Bu özelliğe instance yaratıldığı anda mevcut veritabanı bağlantısı ile ilişkili olan command nesnesi atanmalıdır; çünkü metodlar veri tabanı işlemleri için bu değişkeni kullanmaktadırlar.

Public ErrNum As Integer: Hata numarası.

Public ErrMessage As String: Hata mesajı.

Metodlar

Genel Sınıfı arayüzüne ait tüm metodlar aşağıda listelenmiştir.

Public Function Aciklama(TabloAdi As String, KayitNo As Integer) As String

KayitNo argümanı ile verilen kaydın açıklamasını TabloAdi ile verilen tablodan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function GununTarihi() As String

Sistem tarihini yyyyymmdd formatında geri döndürür.

Public Function SimdikiSaat() As String

Sistem saatini hh:mm formatında geri döndürür.

Public Function SeansOku() As ADODB.Recordset

TBL_SEANS tablosundaki tüm kayıtları okuyarak bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function GunOku() As ADODB.Recordset

TBL_GUN tablosundaki tüm kayıtları okuyarak bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function BolumAdi(BolumNo As Integer) As String

BolumNo argümanı ile verilen bölümün adını TBL_BOLUM tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function FakulteAdi(FakulteNo As Integer) As String

FakulteNo argümanı ile verilen fakültenin adını TBL_FAKULTE tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function FakulteNo(BolumNo As Integer) As Integer

BolumNo argümanı ile verilen bölümün fakültesinin numarasını TBL_BOLUM tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırse hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function KurumAdi(KurumNo As Integer) As String

KurumNo argümanı ile verilen kurumun adını TBL_KURUM tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function KampusOku() As ADODB.Recordset

TBL_KAMPUS tablosundaki tüm kayıtları okuyarak Ad alanına göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function BinaOku() As ADODB.Recordset

TBL_BINA tablosundaki tüm kayıtları okuyarak BinaNo ve SinifTipNo alanlarına göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function SinifOku() As ADODB.Recordset

TBL_SINIF tablosundaki tüm kayıtları okuyarak Ad alanına göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function BinaNo(SinifNo As Integer) As Integer

SinifNo argümanı ile verilen sınıfın binasının numarasını TBL_SINIF tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function BinaAdi(BinaNo As Integer) As String

BinaNo argümanı ile verilen binanın adını TBL_BINA tablosundan okuyarak geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function SinifBilgileri(SinifNo As Long) As ADODB.Recordset

SinifNo argümanı ile verilen sınıfın tüm alanlarını TBL_SINIF tablosundan okuyarak bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Sub OdemeYap(KurumNo as Integer, KullaniciNo As Long, Tutar as Long, Oaciklama as String)

KurumNo ile verilen kuruma, KullaniciNo numaralı kullanıcının Tutar kadar ödeme yaptığına dair kayıt Oaciklama alanıyla birlikte TBL_ODEME tablosuna kaydedilir. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Function ymdTodmy(Tarih As String) As String

“yyyymmdd” formatındaki tarihi, “dd.mm.yyyy” formatına çevirip, geri döndürür.

Function dmyToymd(Tarih As String) As String

“dd.mm.yyyy” formatındaki tarihi, “yyyymmdd” formatına çevirip, geri döndürür.

Private Sub Class_Initialize()

Sınıfın bir nesnesi yaratıldığında public sınıf özelliklerinden olan Comm nesnesi yaratılır.

Private Sub Class_Terminate()

Sınıfın bir nesnesi yok edildiğinde public ve private Comm ve ADORS nesneleri yok edilir.

3.6.4. Class: Kullanici

TBL_KULLANICI ve ilgili diğer tablolardaki alanlara erişim metodlarını içeren sınıftır.

Özellikler

Kullanıcı Sınıfı arayüzüne ait tüm özellikler aşağıda listelenmiştir:

Private Param As New ITUCard.Parametre: Parametre Sınıfı metodlarına erişim amacıyla sadece sınıf içinde kullanılan nesnedir.

Dim AdoConn As ADODB.Connection: Bu özelliğe instance yaratıldığı anda mevcut veri tabanı bağlantısı ile ilişkili olan command nesnesi atanmalıdır. Metodların veri tabanı bağlantısına erişim için kullanımları özel bir değişkendir.

Public Comm As ADODB.Command: Bu özelliğe instance yaratıldığı anda mevcut veri tabanı bağlantısı ile ilişkili olan command nesnesi atanmalıdır; çünkü metodlar veri tabanı işlemleri için bu değişkeni kullanmaktadırlar.

Public ErrNum As Integer: Hata numarası.

Public ErrorMessage As String: Hata mesajı.

Metodlar

Kullanıcı Sınıfı arayüzüne ait tüm metodlar aşağıda listelenmiştir.

Public Sub SifreKontrol(KullaniciNo As Long, Sifre As String)

Kullanıcının kullanıcı numarası ve şifresinin doğruluğu TBL_KULLANICI tablosundaki ilgili alanlar okunarak kontrol edilir. TBL_KULLANICI tablosunda Sifre alanındaki

veri şifrelenmiş olarak tutulduğundan öncelikle şifresi çözülür. Verilen kullanıcı numarası tabloda bulunamadığında ve tablodaki Sifre alanının şifresinin çözülmüş şekli ile Sifre parametresi aynı olmadığında hata numarası ve mesajı özelliklerine gerekli atamalar yapılır.

Public Function KullaniciOku(KullaniciNo As Long) As ADODB.Recordset

KullaniciNo argümanı ile verilen sınıfın tüm alanlarını TBL_KULLANICI tablosundan okuyarak bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Sub YetkiVarMi(UnvanNo As Integer, islemNo As Integer)

UnvanNo argümanı ile verilen ünvanın islemNo argümanı ile verilen işlem için yetkisinin olup olmadığı TBL_YETKI tablosundan ilgili kayıt okunarak araştırılır. Eğer tabloda kayıt bulunuyorsa yetkinin olduğuna dair hata kodu 0 döner. Kayıt bulunmazsa veya okuma sırasında bir hata meydana gelirse hata numarası ve mesajı özelliklerine gerekli atamalar yapılır.

Public Function UnvanKisaltma(KullaniciNo As Long) As String

KullaniciNo argümanı ile verilen kullanıcının ünvan numarası TBL_KULLANICI okunarak elde edilir, daha sonra bu ünvan numarası ile TBL_UNVAN tablosunda sorgulama yapılır ve eğer bulunursa ünvanın kısaltması geri döndürülür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function AdSoyad(KullaniciNo As Long) As String

KullaniciNo argümanı ile verilen kullanıcının adını ve soyadını TBL_KULLANICI tablosundan okuyarak “Ad Soyad” formatında geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşrsa hata numarası ve mesajı özelliklerine sırasıyla kaydın

bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function DonemHesapla(KullaniciNo As Long) As Byte

TBL_KULLANICI tablosundan KullaniciNo ile verilen kullanıcının giriş yılı ve TBL_PARAM tablosundan mevcut yıl ve dönem okunur. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır. Okuma işlemleri sorunsuz bir şekilde tamamlanırsa eğer, kullanıcının kaçınıcı dönemde olduğu hesaplanır ve geri döndürülür. (Öğrencilerin tümünün okula ilk kayıtlarının Kış yarıyılında yapıldığı ve kayıt dondurma, vs işlemlerinin olmadığı varsayılmıştır.)

Public Sub DersEkle(KullaniciNo As Long, DersNo As Integer)

DersNo argümanı ile verilen dersi KullaniciNo ile verilen kullanıcı için TBL_OGRDERS tablosuna ekler. Bunun için ilk olarak, TBL_DERSPRG tablosundan dersin o dönemde verildiği günler ve seanslar okunur. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır. Eğer herhangi bir hata ile karşılaşılmazsa önceki sorgulama sonucunda bulunan her seans için, TBL_OGRDERS ve TBL_DERSPRG tabloları üzerinde sorgulama yapılarak, öğrencinin o gün ve seansta başka bir dersinin kayıtlı olup olmadığı araştırılır. Eğer bulunursa, hata numarası ve mesajı özelliklerine çakışma olduğuna dair gerekli atamalar yapılır. Çakışma yoksa ilgili ders o dönem için KullaniciNo numaralı kullanıcı ile ilişkilendirilecek şekilde TBL_OGRDERS tablosuna eklenir. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Sub DersSil(KullaniciNo As Long, DersNo As Integer)

KullaniciNo argümanı ile verilen kullanıcının daha önceden almış olduğu DersNo ile verilen dersine ilişkin kaydı TBL_OGRDERS tablosundan siler. Herhangi bir hata ile

karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function OgrenciDersPrograminiOku(KullaniciNo As Long) As ADODB.Recordset

KullaniciNo argümanı ile verilen kullanıcının o dönemdeki derslerine ait program TBL_OGRDERS ve TBL_DERSPRG tablolarından okunarak bir resultset içinde geri döndürülür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function OgrenciNotlariniOku(KullaniciNo As Long) As ADODB.Recordset

KullaniciNo argümanı ile verilen kullanıcının o dönemdeki ders kayıtlarını ve notlarını TBL_OGRDERS tablosundan okuyarak bir resultset içinde geri döndürür. Kayıt bulunamazsa veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kaydın bulunamadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function OdemeListesi(KullaniciNo As Long) As ADODB.Recordset

KullaniciNo argümanı ile verilen kullanıcı için TBL_ODEME tablosundaki o ana kadarki harcamaları gösteren tüm kayıtları okuyarak Tarih ve Saat alanlarına göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function RandevuListesi(KullaniciNo As Long) As ADODB.Recordset

KullaniciNo argümanı ile verilen kullanıcı için TBL_RANDEVU tablosundaki o ana kadarki harcamaları gösteren tüm kayıtları okuyarak Tarih ve Seans alanlarına göre sıralı olacak şekilde bir resultset içinde geri döndürür. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Sub RandevuEkle(KullaniciNo As Long, SinifNo As Integer, Tarih As String, Seans As Integer, Gun As Byte)

Tarih argümanı ile verilen randevu tarihinin geçmiş bir tarih olup olmadığı kontrol edilir. Eğer bugünden eski bir tarih ise hata numarası ve mesajı gerekli atamalar yapılır. Aksi takdirde, TBL_DERSPRG tablosundan o dönemde verilen Gun ve Seans'ta herhangi bir dersin olup olmadığı araştırılır. Kayıt bulunursa veya herhangi bir hata ile karşılaşılrsa hata numarası ve mesajı özelliklerine sırasıyla sınıfın dolu olduğuna veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır. Aksi halde, TBL_SINIF tablosundan SinifNo ile verilen sınıfın kapasitesi okunur ve TBL_RANDEVU tablosundaki Tarih ve Seans'ta alınmış randevuların sayısı ile karşılaştırılır. Kapasite dolmuşsa veya herhangi bir hata ile karşılaşılrsa hata numarası ve mesajı özelliklerine sırasıyla sınıfın dolu olduğuna veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır. Eğer herhangi bir hata meydana gelmezse, KullaniciNo ile verilen kullanıcı için ilgili kayıt diğer bilgilerle birlikte TBL_RANDEVU tablosuna eklenir. Herhangi bir hata ile karşılaşılrsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Sub RandevuSil(RandevuNo As Long)

RandevuNo argümanı ile verilen randevu kaydını TBL_RANDEVU tablosundan siler. Herhangi bir hata ile karşılaşılrsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Private Sub Class_Initialize()

Sınıfın bir nesnesi yaratıldığında public sınıf özelliklerinden olan Comm nesnesi yaratılır.

Private Sub Class_Terminate()

Sınıfın bir nesnesi yok edildiğinde public ve private Comm, ADORS ve Param nesneleri yok edilir.

Public Sub SifreKaydet(KullaniciNo As Long, Sifre As String)

KullaniciNo argümanı ile verilen kullanıcının şifresi değiştirilir. Şifre yazılmadan önce kullanıcı numarasının ilk dört ve son dört karakterleri key üretmek amacıyla kullanılarak Sifre bilgisi şifrelenir. Herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Public Function Encode(Sifre As String , KullaniciNo As Long) as String

Sifre argümanı ile verilen 8 karakterlik katar, KullaniciNo argümanının ilk dört ve son dört karakterleri key üretmek amacıyla kullanılarak simetrik bir algoritmayla şifrelenir. Şifrelenmiş 8 karakterlik veri geri döndürülür.

Public Function Decode(Sifre As String , KullaniciNo As Long) as String

Sifre argümanı ile verilen 8 karakterlik simetrik bir algoritmayla şifrelenmiş katar, KullaniciNo argümanının ilk dört ve son dört karakterleri key üretmek amacıyla kullanılarak şifreleme algoritmasının tersi olan algoritmayla çözülür. Şifresi çözülmüş 8 karakterlik veri geri döndürülür.

3.6.5. Class: Parametre

TBL_PARAMETRE tablosundaki alanlara ve kayıtlara erişim metodlarını içeren sınıftır.

Özellikler

Parametre Sınıfı arayüzüne ait tüm özellikler aşağıda listelenmiştir:

Public Comm As ADODB.Command: Bu özelliğe instance yaratıldığı anda mevcut veri tabanı bağlantısı ile ilişkili olan command nesnesi atanmalıdır; çünkü metodlar veri tabanı işlemleri için bu değişkeni kullanmaktadırlar.

Public ErrNum As Integer: Hata numarası.

Public ErrorMessage As String: Hata mesajı.

Metodlar

Parametre Sınıfı arayüzüne ait tüm metodlar aşağıda listelenmiştir.

Public Sub DersKayitZamaniMi()

Günün tarihi TBL_PARAM dosyasındaki kayıt başlangıç ve bitiş tarihleri olan DersKayitBasTar ve DersKayitBitTar arasında ise kayıt zamanı olduğuna dair hata kodu 0 döner. Aksi takdirde veya herhangi bir hata ile karşılaşırsa hata numarası ve mesajı özelliklerine sırasıyla kayıt zamanı olmadığına veya veritabanı işlemlerinde bir hata meydana geldiğine dair gerekli atamalar yapılır.

Function YilOku() As String

TBL_PARAM dosyasındaki o anki yılı gösteren Yil parametresi geri döndürülür.

Function YariYilOku() As String

TBL_PARAM dosyasındaki o anki yarıyı gösteren YariYil parametresi geri döndürülür.

Private Sub Class_Initialize()

Sınıfın bir nesnesi yaratıldığında public sınıf özelliklerinden olan Comm nesnesi yaratılır.

Private Sub Class_Terminate()

Sınıfın bir nesnesi yok edildiğinde public ve private Comm ve ADORS nesneleri yok edilir.

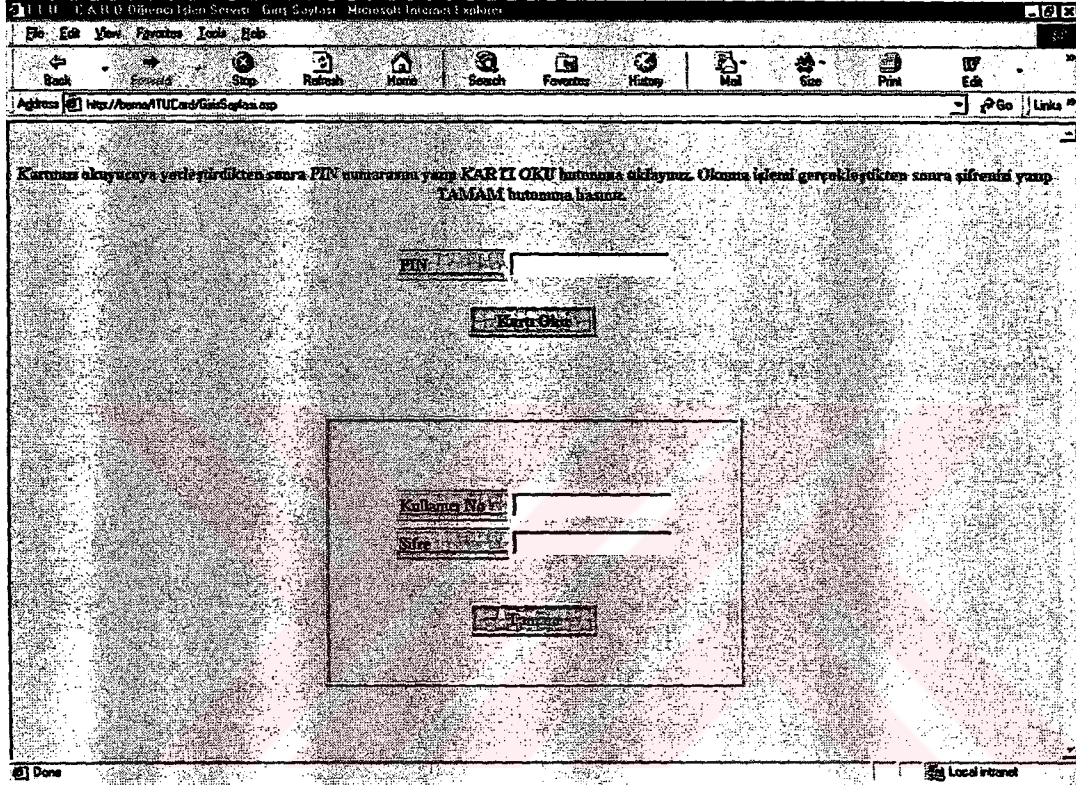
3.7. Web Uygulaması Sayfaları

İnternet tarayıcılardan çağırılarak kullanılacak kullanıcı arayüzüne ait sayfalar, görünümleri ve fonksiyonel açıklamalarıyla birlikte aşağıda verilmiştir.

3.7.1. Giriş Sayfası

Sayfanın görüntüsü Şekil 3.4'te verilmiştir.

Sayfa Adı: GirişSayfasi.asp



Şekil 3.4. Giriş Sayfası

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.1’de listelenmiştir.

Tablo 3.1 Giriş Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
Kartı Oku	Button	Üzerine tıklandığında karttaki tekil kullanıcı numarasının okunup ekrandaki Kullanıcı No alanına yazılmasını sağlar.
Kullanıcı No	Text Field	Maksimum 10 karakter
Şifre	Password	Maksimum 8 karakter
Tamam	Sbutton	AnaSayfa.asp’ye referans

İşlemler :

- Sitenin ilk sayfasıdır.
- Tüm alanlar boş olarak görüntülenir.
- Siteye giriş yapmadan herhangi bir işlem üzerinde önceden kalmış kullanıcı numarası ile işlem yapılabilmesini önlemek için KullanıcıNo session değişkeni sıfırlanır.
- Kartı Oku butonunun üzerine tıklandığında karttaki tekil kullanıcı numarasının okunup ekrandaki Kullanıcı No alanına yazılır.
- Okuma sırasında bir hata meydana gelmişse ilgili mesaj görüntülenir.
- Aksi takdirde, kullanıcıdan şifresinin istendiği mesaj görüntülenir.
- Tamam butonuna tıklandığında AnaSayfa.asp'ye yönlendirme yapılır.

Kullanılan Gemplus410.ocx Elemanları :

- KartOku() : Kartın numarasının okunup geri döndürülmesi işlemini yerine getirir.

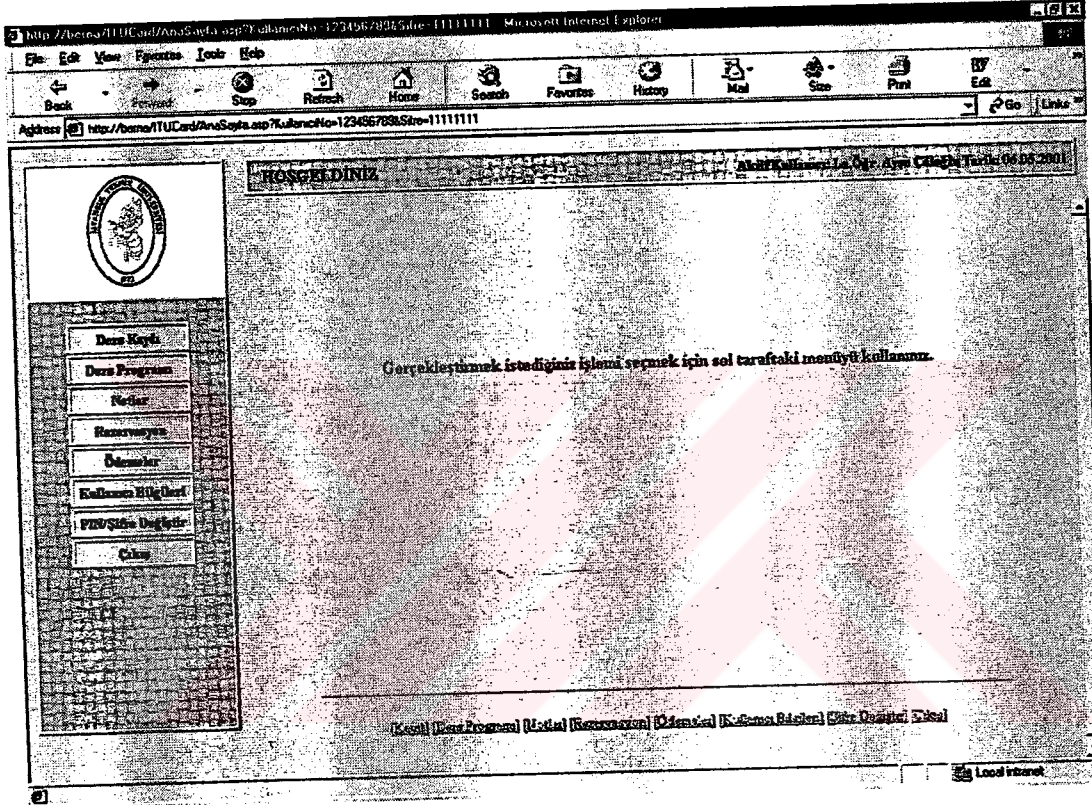
Mesajlar :

- Kartınızı okuyucuya yerleştirdikten sonra PIN numarasını yazıp KARTI OKU butonuna tıklayınız. Okuma işlemi gerçekleşikten sonra şifrenizi yazıp TAMAM butonuna basınız.
- Kullanıcı numarası sistemde kayıtlı değildir. Lütfen tekrar deneyiniz.
- Verdiğiniz şifre doğru değildir. Lütfen tekrar deneyiniz.

3.7.2. Ana Sayfa

Sayfanın görüntüsü Şekil 3.5'te verilmiştir.

Sayfa Adı: AnaSayfa.asp



Şekil 3.5. Ana Sayfa

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.2'de listelenmiştir.

Tablo 3.2. Ana Sayfa Bileşenleri

Etiketi	Tipi	Açıklama
Sol	Frame	Sol.asp : Menü'nün bulunduğu frame
Baslik	Frame	Baslik.asp : Sayfa başlıklarının bulunduğu frame
Sag	Frame	Sag.asp : Sayfaların yer aldığı frame

İşlemler :

- `GirisSayfasi.asp`'den gelen Kullanıcı No'sunun varlığı kontrol edilir. Kayıtlı değilse ilgili mesaj `GirisSayfasi.asp`'de görüntülenir.
- Kullanıcı No mevcut ise Şifre'nin doğruluğu kontrol edilir. Yanlış ise ilgili mesaj `GirisSayfasi.asp`'de görüntülenir.
- Şifre doğru ise ünvan kontrol edilir.
- Ünvan "Kurum" ünvanı değilse, kullanıcı bilgileri okunarak `KullaniciNo`, `AdSoyad`, `UnvanNo`, `BolumNo`, `ABDaliNo`, `Yil`, `YariYil` ve `Donem` session değişkenlerine ilgili değerler daha sonraki sayfalarda kullanılmak üzere atanır.
- Ünvan "Kurum" ünvanı ise, kurum numarası `KurumNo`, kurumun adı ise `AdSoyad` session değişkenlerine daha sonraki sayfalarda kullanılmak üzere atanır.
- `Sol`, `Baslik` ve `Sag` isimli frameler görüntülenerek sırasıyla `Sol.asp`, `Baslik.asp` ve `Sag.asp` yüklenir.

Kullanılan ITUCard.dll Elemanları :

- `Kullanici.SifreKontrol clng(KullaniciNo), cstr(Request("Sifre"))`
- `Kullanici.KullaniciOku(Request("KullaniciNo"))`
- `Kullanici.AdSoyad(clng(Session("KullaniciNo")))`
- `Parametre.YilOku()`
- `Parametre.YariYilOku()`
- `Kullanici.DonemHesapla(clng(Session("KullaniciNo")))`
- `Genel.KurumAdi(cint(RS("KurumNo")))`

Mesajlar :

- Yoktur.

3.7.3. Başlık Sayfası

Sayfa Adı: Baslik.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.3'te listelenmiştir.

Tablo 3.3. Başlık Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
<i>Sayfa Adı</i>	Div	-

İşlemler :

- Sol menüdeki linklere tıklandıkça Sag isimli framede açılan sayfanın başlığı yer alır.
- Sayfanın adını içeren başlık yazısı dinamik olarak değiştirilir.
- Aktif kullanıcının ünvanı ile birlikte adı, soyadı ve sistem tarihi bilgileri görüntülenir. Kullanıcı bilgilerinin alınması için session değişkenlerinden yararlanılır.

Kullanılan ITUCard.dll Elemanları :

- `Kullanici.UnvanKisaltma(cLng(Session("KullaniciNo"))Kullanici.KullaniciOku(Request("KullaniciNo"))`
- `Genel.ymdTodmy(Genel.GununTarihi())Parametre.YilOku())`

Mesajlar :

- Yoktur.

3.7.4. Menü Sayfası

Sayfa Adı: Sol.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.4'te listelenmiştir.

Tablo 3.4. Menü Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
Yeni Ödeme	Link	YeniOdeme.asp'ye referans
Ders Kaydı	Link	DersKaydi.asp'ye referans
Ders Programı	Link	DersProgrami.asp'ye referans
Notlar	Link	Notlar.asp'ye referans
Rezervasyon	Link	Rezervasyon.asp'ye referans
Ödemeler	Link	Odemeler.asp'ye referans
Kullanıcı Bilgileri	Link	KullaniciBilgileri.asp'ye referans
PIN/Şifre Değiştir	Link	SifreDegistir.asp'ye referans
Çıkış	Link	GirisSayfasi.asp'ye referans

İşlemler :

- Linklerde verilen tüm hedef sayfalar Sag frame içinde görüntülenerek, Baslik.asp'ye açılan sayfanın başlığı yazılır.
- Siteye giriş yapmış kullanıcının ünvanı bir kuruma ait ise, sadece Yeni Ödeme, PIN/Şifre Değiştir ve Çıkış linkleri görüntülenir. Aksi takdirde Yeni Ödeme dışındaki tüm linkler listelenir.

Kullanılan ITUCard.dll Elemanları :

- Yoktur.

Mesajlar :

- Yoktur.

3.7.5. Açılış Sayfası

Sayfa Adı: Sag.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.5'te listelenmiştir.

Tablo 3.5. Açılış Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
<i>Açılış Mesajı</i>	Label	-

İşlemler :

- Ana Sayfa açıldığında Sag frame içinde görüntülenen ilk sayfadır.
- Site kullanımına ilişkin açıklamalar veya başlangıç mesajları yer alır.

Kullanılan ITUCard.dll Elemanları :

- Yoktur.

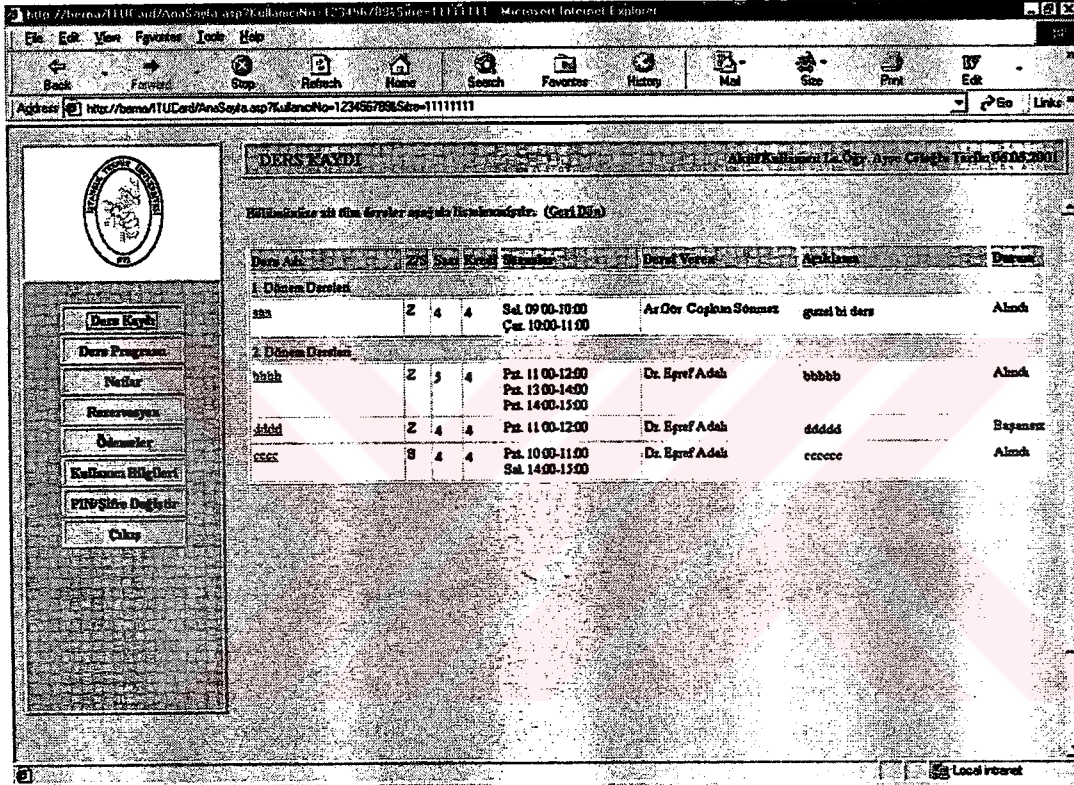
Mesajlar :

- Gerçekleştirmek istediğiniz işlemi seçmek için sol taraftaki menüyü kullanınız.
- Oturum kapanmıştır; yeniden giriş yapmanız gerekmektedir: [Giriş Sayfası](#)
- Bu sayfaya erişim yetkiniz yoktur.

3.7.6. Ders Kayıt Sayfası

Sayfanın görüntüsü Şekil 3.6’da verilmiştir.

Sayfa Adı: DersKaydi.asp



Şekil 3.6. Ders Kaydı Sayfası

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.6’da listelenmiştir.

Tablo 3.6. Ders Kayıt Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Bölüm Dersleri Listesi	Table	Aktif kullanıcının bölümündeki kendi dönemine kadar olan tüm dersler listelenir. Kolonlar: Ders Adı (DersBilgileri.asp’ye referans), Z/S, Saat, Kredi, Seanslar, Dersi Veren, Açıklama, Durum, Sil/Ekle (DersKaydiSil.asp / DersKaydiEkle.asp’ye referans)

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp’ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp’ye yönlendirme yapılır.
- Yetki varsa, kullanıcının bölümündeki kendi dönemine kadar olan tüm dersler okunarak bir tabloda listelenir. Tablodan okuma döneme göre sıralı olarak yapılır ve dersler dönem dönem gruplanarak görüntülenirler. Böylelikle kullanıcı hangi dersin hangi döneme ait olduğunu açıkça görebilir.
- Ders Adı kolonundan DersBilgileri.asp’ye referans verilir.
- Z/S kolonunda dersin zorunlu veya seçmeli ders olup olmadığı bilgisi yer alır.
- Saat alanı haftalık ders saatini ve Kredi kolonu dersin toplam kredisini gösterir.
- Seanslar kolonunda, dersin haftalık programdaki yeri okunduktan sonra hangi gün hangi seanslar için yer ayrılmışsa bir liste şeklinde görüntülenir.

- Dersi veren öğretim üyesinin adı ve soyadı okunarak yazılır.
- Dersin Açıklama ve Durum alanları ayrı birer kolonda görüntülenir. Durum aşağıdaki değerlerden birisini alabilir:
 - Alınmadı: Öğrencinin dersi daha önce hiç almadığı durumdur.
 - Alındı: Öğrenci dersi mevcut dönemde almıştır ve derse kayıtlı görünmektedir.
 - Başarılı: Öğrenci dersi daha önce almış ve başarılı bir şekilde tamamlamıştır.
 - Başarısız: Öğrenci dersi daha önce almış ancak başarılı olamamıştır.
 - Belirsiz: Durumun belirlenmesi sırasında bir hata meydana gelirse, yukarıdakilerden hiçbirisinin olmadığını vurgulamak için kullanılır.
- Eğer sistem tarihi ders kayıt başlangıç ve bitiş tarihleri arasında ise son kolondaki işlem linkleri görüntülenir. Aksi takdirde bu kolon görüntülenmez. Kolonda yer alacak işlemler için koşullar aşağıdaki gibidir:
 - Ders “Alınmadı” durumunda ise Ekle linki görüntülenir. DersKaydiEkle.asp’ye referans verilir.
 - Ders “Alındı” durumunda ise Sil linki görüntülenir. DersKaydiSil.asp’ye referans verilir.
 - Ders “Başarılı” durumunda ise hücre boş olarak görüntülenir.
 - Ders “Başarısız” durumunda ise Ekle linki görüntülenir. DersKaydiEkle.asp’ye referans verilir.
- Önceki kolonda ders durumu “Belirsiz” ise bu kolonda da sadece “Belirsiz” yazılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 2
- Ders.BolumDersleriniOku(cint(Session("BolumNo")))
- Parametre.DersKayitZamaniMi
- Ders.DersPrograminiOku(cstr(Session("Yil")),cstr(Session("YariYil")), cint(RS("DersNo")))
- Genel.Aciklama("TBL_GUN", cint(RSDersPrg("Gun")))
- Genel.Aciklama("TBL_SEANS", cint(RSDersPrg("Seans")))
- Kullanici.UnvanKisaltma(clng(RS("OgrGrvSicilNo")))
- Kullanici.AdSoyad(clng(RS("OgrGrvSicilNo")))
- Ders.OgrDersBilgileriniOku(clng(Session("KullaniciNo")), cint(RS("DersNo")))

Mesajlar :

- Bölümünüze ait tüm dersler aşağıda listelenmiştir. ([Geri Dön](#))
- Bölümünüze ait ders kaydı bulunamamıştır.
- Ders saati, aldığınız bir başka ders ile çakışmaktadır. Lütfen ders programınızı kontrol ediniz. ([Geri Dön](#))
- Ders kayıt zamanı değildir. Lütfen ders kayıt tarihlerini kontrol ediniz. ([Geri Dön](#))
- İlgili kayıt bulunamamıştır.
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. ([Geri Dön](#))

- Seçmiş olduğunuz ders, kaydınıza başarıyla eklenmiştir. ([Geri Dön](#))
- Seçmiş olduğunuz ders, başarıyla kaydınızdan çıkarılmıştır. ([Geri Dön](#))



3.7.7. Ders Bilgileri Detay Sayfası

Sayfanın görüntüsü Şekil 3.7’de verilmiştir.

http://www.ATUCad/ArasSayfa.asp?KullaniciNo=123456789&Ssn=111111111

DERS KAYDI

Sınıf: 2 Nolu Aras Ders Bilgileri (Görüşme)

Dersin Adı: ara

Dersin Kodu: 1. Dönem

Zorunlu / Seçmeli: Zorunlu

Dersin Durumu: 43 saat

Dersin İçeriği: Araştırma

Dersin Sorumlusu: Ar. Gör. Çayhan, Sönmez

Dersin Açıklaması: genel bi ders

Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	SÖZ	KÖZ	KÖZ	KÖZ	Dersin
2001	Yaz															Alınmış

Kayıt | Ders Programı | Notlar | Rezervasyon | Ödevler | Kullanıcı Bilgileri | PROĞRAm Değiştir | Çıkış

Şekil 3.7. Ders Bilgileri Detay Sayfası

Sayfa Adı: DersBilgileri.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.7’de listelenmiştir.

Tablo 3.7. Ders Bilgileri Detay Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Dersin Adı	Label	Dersin adı görüntülenir.
Dönemi	Label	Dersin dönemi görüntülenir.
Zorunlu / Seçmeli	Label	Dersin zorunlu ders mi seçmeli dersmi olduğu yazılır
Haftalık Ders Saati	Label	Dersin haftada kaç saat verildiği yazılır.
Toplam Kredi	Label	Dersin toplam kredisi görüntülenir.
Ders Hocası	Label	Dersi veren öğretim üyesinin adı ve soyadı ünvanı ile birlikte görüntülenir.
Seanslar	Label- Liste	Dersin haftalık ders programındaki yerine ilişkin gün ve seanslar bir liste halinde yazılır.
Açıklama	Label	Dersin açıklaması yazılır.
Öğrencinin Ders Geçmişi Listesi	Table	Aktif kullanıcının, dersi daha önce almışsa eğer, her alışına ait not çizelgesi listelenir. Kolonlar: Yıl, YY(YariYil), V1(Vize1), V2(Vize2), F(Final), Ö1(Ödev1), Ö2(Ödev2), Ö3(Ödev3), Ö4(Ödev4), Q1(Quiz1), Q2(Quiz2), SOrt(Sayısal Ortalama), Kort(Ortalamanın Harf Karşılığı), Ktlm(Dönem İçindeki Toplam Katılım Saati), Hak(Kaçıncı Kez Alındığı), Durum

İşlemler :

- Oturumun kapanıp kapanmadığı KullanıcıNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.

- Yetki varsa, derse ilişkin yukarıdaki tabloda yer açıklanmış olan tüm genel bilgiler okunarak görüntülenir.
- Ders genel bilgileri görüntülendikten sonra, aktif kullanıcının, eğer daha önce dersi almışsa, her alışına ait not çizelgesi okunarak bir tablo halinde listelenir.
- Çizelgedeki not bilgileri için eğer kayıtlı bir sayı bulunmuyorsa “-” görüntülenir.
- Dersin Durum alanlı ayrı birer kolonda görüntülenir. Durum aşağıdaki değerlerden birisini alabilir:
 - Alınmadı: Öğrencinin dersi daha önce hiç almadığı durumdur.
 - Alındı: Öğrenci dersi mevcut dönemde almıştır ve derse kayıtlı görünmektedir.
 - Başarılı: Öğrenci dersi daha önce almış ve başarılı bir şekilde tamamlamıştır.
 - Başarısız: Öğrenci dersi daha önce almış ancak başarılı olamamıştır.
 - Belirsiz: Durumun belirlenmesi sırasında bir hata meydana gelirse, yukarıdakilerden hiçbirisinin olmadığını vurgulamak için kullanılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Ders.DersBilgileriniOku(cint(Request("DersNo")))
- Kullanici.UnvanKisaltma(cLng(RS("OgrGrvSicilNo")))
- Kullanici.AdSoyad(cLng(RS("OgrGrvSicilNo")))
- Ders.DersPrograminiOku(cstr(Session("Yil")),cstr(Session("YariYil")), cint(RS("DersNo")))
- Genel.Aciklama("TBL_GUN", cint(RSDersPrg("Gun")))

- Genel.Aciklama("TBL_SEANS", cint(RSDersPrg("Seans")))
- Ders.OgrDersBilgileriniOku(clng(Session("KullaniciNo")), cint(Request("DersNo")))

Mesajlar :

- Seçtiğiniz ... Nolu derse ilişkin bilgiler aşağıdadır. ([Geri Dön](#))
- Bu dersi daha önce almamışsınız.
- İlgili kayıt bulunamamıştır. ([Geri Dön](#))
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. ([Geri Dön](#))

3.7.8. Ders Kaydı Ekleme Sayfası

Sayfa Adı: DersKaydiEkle.asp

Bileşenler:

- Yoktur.

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, önceki sayfadan gelen bilgiler aktif kullanıcının almış olduğu ders olarak kaydedilir.

- Herhangi bir hata ile karşılaşırsa eğer, ilgili hata mesajı ile birlikte DersKaydi.asp sayfasına yönlendirme yapılır.
- Aksi takdirde DersKaydi.asp sayfası, kaydetme işleminin başarılı bir şekilde sonuçlandığına dair mesajla birlikte çağırılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Kullanici.DersEkle cng(Session("KullaniciNo")), cint(Request("DersNo"))

Mesajlar :

- Yoktur.

3.7.9. Ders Kaydı Silme Sayfası

Sayfa Adı: DersKaydiSil.asp

Bileşenler:

- Yoktur.

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, önceki sayfadan gelen DersNo bilgisi ile verilen ders aktif kullanıcının programından silinir.

- Herhangi bir hata ile karşılaşırsa eğer, ilgili hata mesajı ile birlikte DersKaydi.asp sayfasına yönlendirme yapılır.
- Aksi takdirde DersKaydi.asp sayfası, silme işleminin başarılı bir şekilde sonuçlandığına dair mesajla birlikte çağırılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Kullanici.DersSil clng(Session("KullaniciNo")), cint(Request("DersNo"))

Mesajlar :

- Yoktur.

http://home111ulcard.ars-sayfa.org/KulancNo-123456789SLShp-11111111 Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Geo Print Edit

Address http://home111ulcard.ars-sayfa.org/KulancNo-123456789SLShp-11111111 Go Links

DERS PROGRAMI Araç Kulanc No: 123456789SLShp-11111111

2001 yılı TAE yılına ders programı: (Geri Dön)

Ders Zamanı	09:00-10:00	10:00-11:00	11:00-12:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00
Fis.		SLSh	SLSh	SLSh	SLSh			
Sal.	SLSh				SLSh			
Cum.		SLSh						
Cin.								
Fis.								

(Geri Dön) (Ders Programı) (Notlar) (Rezervasyon) (Öğrenciler) (Kulanc Bilgi) (PRVSLShp Değeri) (Çıkış)

Done Local intranet

Tablo 3.8. Haftalık Ders Programı Sayfası Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Haftalık Ders Programı Tablosu	Table	Aktif kullanıcının o dönemde almış olduğu derslere ilişkin haftalık ders programı her satırlarda günler, kolonlarda seanslar olacak şekilde bir tablo halinde listelenir. Satırlar: Pzt, Sal, Car, Per, Cum, Cts, Paz Kolonlar: 09:00-10:00, 10:00-11:00, 11:00-12:00, 13:00-14:00, 14:00-15:00, 15:00-16:00, 16:00-17:00, 17:00-18:00,

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, aktif kullanıcının o dönemde almış olduğu derslere ilişkin ders programı okunarak yedi satırlı, sekiz kolonlu tablodaki uygun hücreye ilgili dersin adı yazılır.
- Hücrelerdeki Ders Adı bilgisinden DersBilgileri.asp'ye referans verilir.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 2
- Genel.SeansOku()
- Genel.GunOku()

- Kullanici.OgrenciDersPrograminiOku(cLng(Session("KullaniciNo"))))

Mesajlar :

- .. yılı .. yarıyılı ders programınız: (Geri Dön)
- Seans kaydı bulunamamıştır. Lütfen öğrenci işlerine danışınız.
- Gün kaydı bulunamamıştır. Lütfen öğrenci işlerine danışınız.
- Bu dönemde aldığınız herhangi bir ders bulunmamaktadır.
- İlgili kayıt bulunamamıştır.
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. (Geri Dön)

3.7.11. Kullanıcı Bilgileri Detay Sayfası

Sayfanın görüntüsü Şekil 3.9'da verilmiştir.

KULLANICI BİLGİLERİ [Geri Dön]

Sayfa Açtıktan Çıktığınız kullanıcı ve Etilik bilgilerinize ulaşabilirsiniz. (Geri Dön)

Adı	Ca.Öğ.	Cinsiyet	1998
Soyadı		Etilik	Elektronik Elektronik
Ana Bilgi		Etilik	Elektronik Müh.
Adres	Fahri Bey 3. Kat Kat. K Blok D/5 Sahurcadı Etilik/İST		
E-Mail Adresi	ardalgha@gmail.com.tr	Telefon Numarası	2163301046
Çalışan	K	Devlet	TC
Etilik Adı	Mustafa Özköken	Ana Adı	Zehra
Doğum Tarihi	15.04.1975	Doğum Yeri	Ankara
Medeni Hali	B	Din	İslam
İl	İstanbul	İşlet	İstanbul
Mobil No	7077777777	Özellik Kart Numarası	Y04-12344
Çi No	24	Alan Sayı Numarası	12
Ölçü Numarası		Ekli Yönlendirme Numarası	

Şekil 3.9. Kullanıcı Bilgileri Sayfası

Sayfa Adı: KullaniciBilgileri.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.9'da listelenmiştir.

Tablo 3.9. Kullanıcı Bilgileri Detay Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Ünvan	Label	Kullanıcının ünvanı okunarak görüntülenir.
Giriş Yılı	Label	Kullanıcının giriş yılı görüntülenir
Enstitü	Label	Kullanıcının enstitüsünün adı okunarak görüntülenir
Fakülte	Label	Kullanıcının fakültesinin adı okunarak görüntülenir
Ana Bilim Dalı	Label	Kullanıcının ana bilim dalının adı okunarak görüntülenir
Bölüm	Label	Kullanıcının bölümünün adı okunarak görüntülenir
Adres	Label	Kullanıcının adresi görüntülenir.
E-Mail Adresi	Label	Kullanıcının e-mail adresi görüntülenir.
Telefon Numarası	Label	Kullanıcının telefon numarası görüntülenir.
Cinsiyet	Label	Kullanıcının cinsiyeti yazılır.
Uyruk	Label	Kullanıcının uyruğu görüntülenir.
Baba Adı	Label	Kullanıcının babasının adı görüntülenir.
Anne Adı	Label	Kullanıcının annesinin adı görüntülenir.
Doğum Tarihi	Label	Kullanıcının doğum tarihi “gg.aa.yyyy” formatında görüntülenir.
Doğum Yeri	Label	Kullanıcının doğum yeri görüntülenir.
Medeni Hali	Label	Kullanıcının medeni hali görüntülenir.
Dini	Label	Kullanıcının dini görüntülenir.
İli	Label	Kullanıcının nüfus kaydındaki ili görüntülenir.
İlçesi	Label	Kullanıcının nüfus kaydındaki ilçesi görüntülenir.
Mahallesi	Label	Kullanıcının nüfus kaydındaki mahallesi görüntülenir.
Nüfus Numarası	Seri	Label Kullanıcının nüfus kaydındaki seri numarası görüntülenir.

Cilt No	Label	Kullanıcının nüfus kaydındaki cilt numarası görüntülenir.
Aile Sıra Numarası	Label	Kullanıcının nüfus kaydındaki aile sıra numarası görüntülenir.
Oda Numarası	Label	Kullanıcının eğer varsa odasının numarası görüntülenir.
Dahili Telefon Numarası	Label	Kullanıcının eğer varsa dahili telefon numarası görüntülenir.

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, aktif kullanıcıya ilişkin yukarıdaki tabloda açıklanmış olan tüm genel bilgiler okunarak görüntülenir.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Kullanici.KullaniciOku(cLng(Session("KullaniciNo")))
- Kullanici.UnvanKisaltma(cLng(Session("KullaniciNo")))
- Genel.FakulteAdi(cint(Genel.FakulteNo(cint(RS("ABDaliNo")))))
- Genel.FakulteAdi(cint(Genel.FakulteNo(cint(RS("BolumNo")))))
- Genel.BolumAdi(cint(RS("ABDaliNo")))

- Genel.BolumAdi(cint(RS("BolumNo")))
- Genel.ymdTodmy(RS("DogumTarihi"))

Mesajlar :

- Sayın, kullanıcı ve özlük bilgileriniz aşağıdadır: (Geri Dön)
- İlgili kayıt bulunamamıştır. (Geri Dön)
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. (Geri Dön)



Tablo 3.10. Not Çizelgesi Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Not Çizelgesi Tablosu	Table	Aktif kullanıcının o dönemde almış olduğu derslere ilişkin not çizelgesi tablo halinde listelenir. Kolonlar: Dersin Adı (DersBilgileri.asp'ye referans), V1(Vize1), V2(Vize2), F(Final), Ö1(Ödev1), Ö2(Ödev2), Ö3(Ödev3), Ö4(Ödev4), Q1(Quiz1), Q2(Quiz2), SOrt(Sayısal Ortalama), Kort(Ortalamanın Harf Karşılığı), Ktlm(Dönem İçindeki Toplam Katılım Saati), Hak(Kaçıncı Kez Alındığı), Durum

İşlemler :

- Oturumun kapanıp kapanmadığı KullanıcıNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, aktif kullanıcının, o dönemde almış olduğu tüm derslere ait not çizelgesi okunarak bir tablo halinde listelenir.
- Hücrelerdeki Dersin Adı bilgisinden DersBilgileri.asp'ye referans verilir.
- Çizelgedeki not bilgileri için eğer kayıtlı bir sayı bulunmuyorsa “-” görüntülenir.
- Dersin Durum alanı ayrı birer kolonda görüntülenir. Durum aşağıdaki değerlerden birisini alabilir:
- Alındı: Öğrenci dersi mevcut dönemde almıştır ve derse kayıtlı görünmektedir.

- **Başarılı:** Öğrenci dersi daha önce almış ve başarılı bir şekilde tamamlamıştır.
- **Başarısız:** Öğrenci dersi daha önce almış ancak başarılı olamamıştır.
- **Belirsiz:** Durumun belirlenmesi sırasında bir hata meydana gelirse, yukarıdakilerden hiçbirisinin olmadığını vurgulamak için kullanılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 2
- Kullanici.OgrenciNotlariniOku(cLng(Session("KullaniciNo")))

Mesajlar :

- .. yılı ..yarıyılı notlarınız: (Geri Dön)
- Bu dönemde aldığınız herhangi bir ders bulunmamaktadır.
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. (Geri Dön)

3.7.13. Ödeme Özeti Sayfası

Sayfanın görüntüsü Şekil 3.11'de verilmiştir.

Tarih	Saat	Açıklama	Kurum Adı	Tutar
06.03.2001	13:16	aciklama	Elektronik-Elektronik Fak. Fotokopirisi	234 TL
20.03.2001	12:15	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
19.03.2001	12:20	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
18.03.2001	12:15	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
17.03.2001	12:20	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
16.03.2001	12:47	Defter Asıl 180 y.	Elektronik-Elektronik Fak. Fotokopirisi	1300000 TL
16.03.2001	12:15	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
15.03.2001	12:15	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
14.03.2001	12:20	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
13.03.2001	12:18	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
12.03.2001	12:20	Yemek	Ayazaga Kampüsü Yemekhanesi	200000 TL
02.03.2001	13:42	200 sayfa fotokopi + cilt	Elektronik-Elektronik Fak. Fotokopirisi	1800000 TL

Şekil 3.11. Ödeme Listesi Sayfası

Sayfa Adı: Odemeler.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.11'de listelenmiştir.

Tablo 3.11. Ödeme Özeti Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Ödeme Tablosu	Table	Aktif kullanıcının o zamana kadar yapmış olduğu tüm ödemeler tablo halinde listelenir. Kolonlar: Tarih, Saat, Açıklama, Kurum Adı, Tutar

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, aktif kullanıcının, o zamana kadar yapmış olduğu tüm harcamalar tabloda her sayfada 15 er kayıt olacak şekilde sayfalanarak listelenir.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 2
- Kullanici.OdemeListesi(cLng(Session("KullaniciNo")))

Mesajlar :

- Şimdiye kadar yaptığınız ödemeler aşağıda listelenmiştir : ([Geri Dön](#))
- Herhangi bir ödemeniz bulunmamaktadır.
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. ([Geri Dön](#))

3.7.14. Rezervasyon Yaptırma ve İzleme Sayfası

Sayfanın görüntüsü Şekil 3.12’de verilmiştir.

REZERVASYON

Rezervasyon yaptırarak bu kütüphaneye kitapları almanızı ve kütüphane ile ilgili soruları, sorular ve sorulara cevaplar almanızı sağlar. "REZERVE ET" butonuna tıklayarak. (Giriş)

Kısım No: Zaman:

Sınıf:

Tarih:

Mon	Tue	Wed	Thu	Fri	Sat	Sun
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Rezervasyon Detayları

Tarih	Saat	Bölüm/Ad	Sınıf
03.12.2001	10:00-11:00	Elektronik-Elektronik Fakültesi	Laboratuvar - Elektronik Elemanlar Lab
03.12.2001	13:00-14:00	Upak-Uzak Eğitim Fakültesi	Ofis - 109
04.07.2001	09:00-10:00	Mimarlık Fakültesi	Sınıf - D1002
04.07.2001	09:00-10:00	Mimarlık Fakültesi	Sınıf - D1002

Şekil 3.12. Rezervasyon Sayfası

Sayfa Adı: Rezervasyon.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.12’de listelenmiştir.

Tablo 3.12. Rezervasyon Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Mesaj alanı	Label	-
Kampüs	ComboBox	Kampüs isimleri listelenir.
Bina	ComboBox	İlgili bina isimleri listelenir.
Sınıf	ComboBox	İlgili sınıf isimleri listelenir.
Tarih	MonthView	Randevu tarihinin seçilmesi için kullanılır.
Seans	ComboBox	Tüm seanslar listelenir.
Rezerve Et	Sbutton	RezervasyonEkle.asp'ye yönlendirme yapılır.
Randevu Listesi	Table	Aktif kullanıcının o zamana dek almış olduğu tüm randevular listelenir. Kolonlar: Tarih, Seans, Bina Adı, Sınıf, Sil (RezervasyonSil.asp'ye referans)

İşlemler :

- Oturumun kapanıp kapanmadığı KullanıcıNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, tüm alanlar görüntülenir. Kampüs combobox'ında mevcut tüm kampüslerin isimleri alfabetik sıraya göre yer alır ve ilki seçilmiş durumdadır.
- Herhangi bir kampüs seçildiğinde Bina combobox'ı otomatik olarak seçilen kampüsteki binaların isimleriyle dolar.

- Herhangi bir bina seçildiğinde Sınıf combobox'ı otomatik olarak seçilen binadaki sınıfların isimleriyle dolar.
- Seans combobox'ında tüm seanslar listelenir.
- Rezerve Et butonuna tıklandığında RezervasyonEkle.asp'ye yönlendirme yapılır.
- Aynı sayfada aktif kullanıcının almış olduğu tüm randevular da okunarak bir tablo halinde listelenir.
- Tarih "gg.aa.yyyy" formatında yazılır.
- Seans görüntülenir.
- Bina adı okunarak görüntülenir.
- Sınıfın adı okunarak görüntülenir.
- Son kolonda, bugünden önceki günler için yapılmış rezervasyon varsa Sil linki görüntülenir ve RezervasyonSil.asp'ye referans verilir. Aksi takdirde, hücre boş olarak görüntülenir; çünkü rezervasyonun aynı gün içinde iptaline izin verilmez.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 2
- Genel.BinaOku()
- Genel.SinifOku()
- Genel.KampusOku()
- Genel.SeansOku()
- Kullanici.RandevuListesi(cLng(Session("KullaniciNo")))
- Genel.SinifBilgileri(cint(RS("SinifNo")))

- Genel.Aciklama("TBL_SINIFTIPI", cint(RSSinif("SinifTipNo")))
- Genel.GununTarihi()

Mesajlar :

- Rezervasyon yaptırmak için istediğiniz sınıfın binasını ve kampüsünü seçerek, tarih ve seans alanlarını doldurup "REZERVE ET" butonuna tıklayınız. ([Geri Dön](#))
- Herhangi bir rezervasyonunuz bulunmamaktadır.
- İlgili kayıt bulunamamıştır.
- Veri tabanı işlemlerinde bir hata meydana gelmiştir. Lütfen bir süre sonra tekrar deneyiniz. ([Geri Dön](#))
- Seçmiş olduğunuz sınıfta seçmiş olduğunuz seans için boş yer bulunmamaktadır. Lütfen tekrar giriş yapınız. ([Geri Dön](#))
- Geçmiş günlere ait rezervasyon yaptıramazsınız. Lütfen tekrar giriş yapınız. ([Geri Dön](#))
- Seçmiş olduğunuz tarih ve seansta ilgili sınıfta adınıza bir kişilik yer ayrılmıştır. ([Geri Dön](#))
- Seçmiş olduğunuz randevu, başarıyla iptal edilmiştir. ([Geri Dön](#))

3.7.15. Randevu Kaydı Ekleme Sayfası

Sayfa Adı: RezervasyonEkle.asp

Bileşenler:

- Yoktur.

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, önceki sayfadan gelen bilgiler aktif kullanıcı adına rezervasyon olarak kaydedilir.
- Herhangi bir hata ile karşılaşırsa eğer, ilgili hata mesajı ile birlikte Rezervasyon.asp sayfasına yönlendirme yapılır.
- Aksi takdirde Rezervasyon.asp sayfası, kaydetme işleminin başarılı bir şekilde sonuçlandığına dair mesajla birlikte çağırılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Kullanici.RandevuEkle clng(Session("KullaniciNo")), cint(Request("Sinif")), cstr(Yil & Ay & Gun), cbyte(Request("Seans")), cbyte(HaftaninGunu)

Mesajlar :

- Yoktur.

3.7.16. Rezervasyon Kaydı Silme Sayfası

Sayfa Adı: RezervasyonSil.asp

Bileşenler:

- Yoktur.

İşlemler :

- Oturumun kapanıp kapanmadığı KullaniciNo session değişkeninin değerinin 0 olup olmadığına bakılarak kontrol edilir. Eğer kapanmışsa uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Aksi takdirde, aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, önceki sayfadan gelen RezervasyonNo bilgisi ile verilen randevu kaydı tablodan silinir.
- Herhangi bir hata ile karşılaşırsa eğer, ilgili hata mesajı ile birlikte Rezervasyon.asp sayfasına yönlendirme yapılır.
- Aksi takdirde Rezervasyon.asp sayfası, silme işleminin başarılı bir şekilde sonuçlandığına dair mesajla birlikte çağırılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Kullanici.RandevuSil cng(Request("RezervasyonNo"))

Mesajlar :

- Yoktur.

3.7.17. Ödeme Yapma Sayfası

Sayfanın görüntüsü Şekil 3.13'te verilmiştir.

Şekil 3.13. Kurum – Ödeme Sayfası

Sayfa Adı: YeniOdeme.asp

Bileşenler: Sayfada kullanılan bileşenler Tablo 3.13'te listelenmiştir.

Tablo 3.13. Ödeme Yapma Sayfasının Bileşenleri

Etiketi	Tipi	Açıklama
Kartı Oku	Button	Üzerine tıklandığında karttaki tekil kullanıcı numarasının okunup ekrandaki Kullanıcı No alanına yazılmasını sağlar.
Kullanıcı No	Text Field	Maksimum 10 karakter
Şifre	Password	Maksimum 8 karakter
Tutar	Text Field	-
Açıklama	Text Field	-
Tamam	Sbutton	AnaSayfa.asp'ye referans

İşlemler :

- Sadece kurumlar siteye giriş yaptığında görüntülenebilen ödeme yapma sayfasıdır.
- Tüm alanlar boş olarak görüntülenir.
- Giriş yapmadan herhangi bir işlem üzerinde önceden kalmış kullanıcı numarası ile işlem yapılabilmesini önlemek için KullanıcıNo session değişkeni sıfırlanır.
- Kartı Oku butonunun üzerine tıklandığında karttaki tekil kullanıcı numarasının okunup ekrandaki Kullanıcı No alanına yazılır.
- Okuma sırasında bir hata meydana gelmişse ilgili mesaj görüntülenir.
- Aksi takdirde, kullanıcıdan şifresinin ve diğer ödeme alanlarının istendiği mesaj görüntülenir.
- Tamam butonuna tıklandığında öncelikle karttaki kredi alanından belirtilen miktarda eksiltme gerçekleştirilir arkasından OdemeYap.asp'ye yönlendirme yapılır.

Kullanılan Gemplus410.ocx Elemanları :

- KartOku() : Kartın numarasının okunup geri döndürülmesi işlemini yerine getirir.

Mesajlar :

- Kartınızı okuyucuya yerleştirdikten sonra PIN numarasını yazıp KARTI OKU butonuna tıklayınız. Okuma işlemi gerçekleştikten sonra şifrenizi yazıp, diğer alanları doldurarak TAMAM butonuna basınız.
- Kullanıcı numarası sistemde kayıtlı değildir. Lütfen tekrar deneyiniz.
- Verdiğiniz şifre doğru değildir. Lütfen tekrar deneyiniz.
- Yeni ödeme kaydı başarıyla oluşturulmuştur.

3.7.18. Ödeme Kaydı Ekleme Sayfası

Sayfa Adı: OdemeYap.asp

Bileşenler:

- Yoktur.

İşlemler :

- GirişSayfası.asp'den gelen ödemeyi yapacak olan kullanıcıya ait Kullanıcı No'sunun varlığı kontrol edilir. Kayıtlı değilse ilgili mesaj GirişSayfası.asp'de görüntülenir.
- Kullanıcı No mevcut ise Şifre'nin doğruluğu kontrol edilir. Yanlış ise ilgili mesaj GirişSayfası.asp'de görüntülenir.
- Aktif kullanıcının sayfaya erişim yetkisinin olup olmadığı kontrol edilir. Eğer yoksa sayfa görüntülenmeksizin yine uygun hata mesajı ile Sag.asp'ye yönlendirme yapılır.
- Yetki varsa, önceki sayfadan gelen bilgiler, aktif kullanıcı kurum olmak üzere, yeni bir ödeme kaydı olarak ilgili tabloya eklenir.

- Herhangi bir hata ile karşılaşırsa eğer, ilgili hata mesajı ile birlikte YeniOdeme.asp sayfasına yönlendirme yapılır.
- Aksi takdirde YeniOdeme.asp sayfası, kaydetme işleminin başarılı bir şekilde sonuçlandığına dair mesajla birlikte çağırılır.

Kullanılan ITUCard.dll Elemanları :

- Kullanici.SifreKontrol clng(KullaniciNo), cstr(Request("Sifre"))
- Kullanici.YetkiVarMi cbyte(Session("UnvanNo")), 3
- Genel.OdemeYap cint(Session("KurumNo")), clng(Request("KullaniciNo")), clng(Request("Tutar")), cstr(Request("Aciklama"))

Mesajlar :

- Yoktur.

4. SONUÇLAR ve TARTIŞMA

Tezin başlangıç aşamasında standart bir üniversite kartından beklenen belli başlı işlev örneklerini gerçekleştiren bir prototipin hazırlanması hedeflenmiştir. Bu amaç doğrultusunda İTÜKart sistemi planlanan şekilde geliştirilmiştir.

Proje kapsamında İTÜKart sistemi için veritabanı tasarlanmış ve MS SQL Server 7.0 üzerinde hayata geçirilmiştir. Programlarda kullanılan sabit değerler mümkün olduğunca tablolarda tutularak uygulamaya dinamik bir yapı kazandırılmıştır. Tasarım, performansın düşürülmemesi amacıyla sadece prototip işlevlerinin gereksinimlerine sadık kalınarak yapılmıştır. İleriki aşamalarda yeni özelliklerin eklenmesi sırasında veritabanı tasarımı da tekrar gözden geçirilmelidir.

Piyasadaki benzer ürünler incelenmiş, sunduğu yüksek performans/maliyet oranı, kolay ve etkin uygulama geliştirme imkanları (hazır kütüphane fonksiyonları, kart üzerindeki dosya yapısıyla ilgili işlemler için geliştirilmiş GemPilot Yönetim modülü) nedeniyle GemPlus firmasının GemClub ürününün kullanılmasına karar verilmiştir. Programlar ile kart okuyucu arasındaki iletişimin sağlanabilmesi için GemClub kütüphane fonksiyonlarının kullanıldığı arayüz ActiveX teknolojileriyle Visual Basic 6.0 kullanılarak yazılmıştır. ActiveX kontrol olarak hazırlanan genel amaçlı arayüz programı ileride benzer uygulamalarda da rahatlıkla kullanılabilir.

Sistemin çalışması için gerekli tanımlamaların yapıldığı İTÜKart Yönetim Otomasyonu Visual Basic 6.0 ile sunucu-istemci mimarisi temel alınarak hazırlanmıştır. İTÜKart sistemi için gereken ders, bölüm, fakülte, parametre, kullanıcı gibi genel tanımlamalar Yönetim Otomasyonu kullanılarak veritabanına yazılmaktadır. Bu programın çıktıları İTÜKart Kullanıcı Otomasyonu'na girdi oluşturmaktadır.[1].

İTÜKart Kullanıcı Otomasyonu web ortamında çalışmak üzere Microsoft ASP Teknolojileri kullanılarak gerçekleştirilmiştir. Böylelikle, dünyanın her yerinden haftanın 7 günü 24 saat erişim imkanı ile kullanıcıların son derece kolay, hızlı ve ucuz bir şekilde işlem yapabilmeleri mümkün olmaktadır. Modülerliğin ve yeniden kullanılabilirliğin sağlanması, veritabanı işlemlerinde güvenliğin artırılması amacı ile nesneye yönelik programlama teknikleri esas alınarak tüm veritabanı işlemlerinde kullanılmak üzere bir ActiveX DLL hazırlanmıştır.

Geliştirilen İTÜKart sistemi sayesinde daha önce de kısmen yapılabilen internet üzerinden ders seçimi, ders notlarının sorgulanması gibi işlemler geliştirilmiş ve daha güvenli bir şekilde gerçeklemleri sağlanmıştır. Bunun yanı sıra sisteme mevcut yapıda bulunmayan randevu, ödeme gibi yeni işlevler sisteme kazandırılmıştır. Ayrıca, daha önce sadece kimlik kartı olarak görev yapan akıllı kart, bu proje sayesinde diğer işlemler için de kullanılarak sistemin tüm olanaklarından tek bir kart ile faydalanılması sağlanmıştır.

Hazırlanan bu prototip esas alınarak daha yetkin bir üniversite kartı sistemi hazırlanabilir.

Veritabanında tutulan kullanıcı şifresi açık olarak değil, şifrelenmiş şekilde saklanmaktadır. İleride bu şifreleme algoritması geliştirilebilir ya da dünyada kabul görmüş bir algoritma (DES, DES3, vb) uyarlanabilir.

Güvenliğin artırılması amacı ile karttaki kayıt dosyaları farklı işlemler için (okuma, güncelleme, vb) farklı şifreler ile korunabilir.

Prototip'e Kütüphane Otomasyonu, Mediko-Sosyal Otomasyonu gibi yeni modüller eklenebilir, mevcut modüller genişletilebilir. Örneğin İTÜKart Kullanıcı Otomasyonu'nda öğrencinin sınav programını görebileceği bir sayfa hazırlanabilir.

Mevcut prototipte kart üzerinde sadece kullanıcı numarası ve kredi miktarı tutulmaktadır. Daha sonraki aşamalarda kullanıcıya ait diğer bilgiler de (kimlik bilgileri, kütüphaneden alınan kitaplara ilişkin takip bilgileri,vs) kart üzerinde depolanabilir.

Mevcut sistem bir prototip olduğundan bazı işlemler basitleştirilmiştir. Örneğin, öğrenciler için paket ders seçeneğinin bulunmadığı varsayılmıştır. Bu gibi varsayımlar ileriki versiyonlarda ortadan kaldırılarak program geliştirilebilir.

Üzerinde analizler yapılabilecek istatistiklerin alınabileceği Raporlama modülü eklenebilir.

Kartlara kredi yüklenmesi işlemini gerçekleştirebilecek bir ek uygulama hazırlanıp ilgili finans kuruluşunun (örneğin anlaşmalı banka) kullanımına sunulabilir.

Kuşkusuz benzer uygulamaların artmasıyla birlikte ülkemizde de akıllı kart kullanımı yakın bir gelecekte yaygınlaşacaktır. Hali hazırda devam etmekte olan MERNİS – Ulusal Kimlik Kartı, SSK – Bağkur projeleri tamamlandığında bu sistemlerin Emniyet Müdürlüğü ve Türk Telekom sistemleriyle tümleştirilmesi sonucunda Türkiye Cumhuriyeti vatandaşlarının tek bir kart ile kamu kurumlarına ait kaynakları kullanmaları, tüm resmi işlemleri gerçekleştirmeleri ve takip etmeleri sağlanacaktır.

KAYNAKLAR

- [1] ÇİNER, Ece, 2001. İTÜKart Yönetim Otomasyonu, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [2] ISO 7816-1, 1998. Design and Use of Identification Cards Having Integrated Circuit with Contacts - Physical Characteristics, *International Standards Organization*, Cenevre.
- [3] Card Europe UK (1994). *Smart Card Technology Background Paper*, <<http://www.cardeurope.demon.co.uk/rep1.htm>> adresli internet sitesi (27 Mayıs 1999)
- [4] Smart Card News: Introduction to Smart Cards (1998). *Smart Card Technology: Introduction to Smart Cards*, <<http://www.smartcard.co.uk>> adresli internet sitesi (06 Eylül 1998)
- [5] ISO 7816-4, 1995. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Inter-Industry Commands for Interchange, *International Standards Organization*, Cenevre.
- [6] ISO 7816-2, 1999. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Contact Locations and Minimum Size, *International Standards Organization*, Cenevre.
- [7] ISO 10536, 1987. Identification cards -- Contactless integrated circuit(s) cards, *International Standards Organization*, Cenevre.
- [8] Smart Card n d Security Overview (2000). *Smart Card and Security Overview*, <<http://www.smartcardbasic.com>> adresli internet sitesi (22 Ekim 2000)
- [9] ISO 7816-7, 1999. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Interindustry commands for Structured Card Query Language (SCQL), *International Standards Organization*, Cenevre.
- [10] Global Electronic Commerce (1998). *Smart Card with Electronic Commerce*, <<http://ubmail.ubalt.edu/~kkim/smartcard.htm>> adresli internet sitesi (24 Şubat 2000)
- [11] An Overview of Smart Card Security (1999). *An Overview of Smart Card Security*, <<http://www.hkstar.com/~alanchan/papers/smartcardsecurity>> adresli internet sitesi (22 Ekim 2000)

- [12] **Smart Card Applications** (1998). *Smart Card Applications*,
<<http://www.gemplus.com/app/index.htm>> adresli internet sitesi (06 Eylül 1998)
- [13] **Smart Card Technology** (1998). *Smart Card Technology*,
<<http://www.oberthurkirk.com/technology.htm>> adresli internet sitesi (06 Eylül 1998)
- [14] **Purdue University – Smart Card Page** (1996). *Smart Cards*,
<<http://www.tech.purdue.edu/it/resources/aidc/smc card.htm>> adresli internet sitesi (07 Mayıs 1999)
- [15] **ISO 7816-5**, 1994. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Application Identification, *International Standarts Organization*, Cenevre.
- [16] **ISO 7816-6**, 1996. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Inter-Industry Data Elements, *International Standarts Organization*, Cenevre.
- [17] **ISO 7816-3**, 1997. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Electrical Specifications and Communication Protocols, *International Standarts Organization*, Cenevre.
- [18] **ISO 7810 (ID-1)**, 1995. Identification cards - Physical characteristics, *International Standarts Organization*, Cenevre.
- [19] **Smart Card Standarts and electronic Purse** (1997). *Smart Card Standarts and ElectronicPurse*,
<<http://www.aston.ac.uk/smartcard/documentation/standarts1.hmt>> adresli internet sitesi (24 Şubat 2000)

EKLER

Ek. A GemClub Akıllı Kartı

GemClub, gelişmiş fonksiyonlara sahip kolaylıkla kullanılabilen mikroişlemcili bir karttır. Özel elektronik cüzdan, ölçümleme (kullanılan gaz veya elektriğin ölçülmesi), müşteri tanıma, vb gibi alanlardaki uygulamalarda kullanılmaktadır.

GemClub şunları içerir :

- 1 KB EEPROM içeren GemClub 1K kartları,
- EMV uyumlu GemClub-EMV kartları.

Veri Yapısı

GemClub, düz dosya yapısına sahiptir. Uygulama için gereken tüm dosyalar, bellekte aynı seviyede yer alır.

- Sistem dosyası: Kartın genel güvenlik yönetimi için kullanılır.
- Loyalty dosyaları: Puan ve kuralların saklanması için kullanılır.
- Kayıt Dosyaları: Veri depolama da kullanılır.
- Güvenlik dosyaları: Gizli kod (secret code) ve gizli anahtarların (secret key) depolanmasında kullanılır.

Komut Kümesi

GemClub işletim sistemi şu komutları içerir:

- Yönetim komutları: Nesne Yarat (Create Object), Nesne Yoket (Delete Object) ve Haberleşme Hızı Seç (Select Communication Speed)
- Uygulama Komutları: Ödül (Award), Geri Al (Redeem), Kuralı Kullan (Use Rule), Parametre Güncelle (Update Parameter) ve Parametre Oku (Read Parameter).

Veri Erişimi Yönetimi ve Güvenlik

Erişim koşulları veri dosyaları ve veri elemanlarını korumak için kullanılırlar. Bunlar, dosya için belirlenen koruma seviyesini de tanımlarlar. Genel olarak GemClub kartlarda yer alan dosyalar gizli kod veya gizli anahtar ile korunmaktadır.

Herbir veri dosyası veya veri elemanı için özel erişim koşulları belirlenebileceği gibi her komut veya komut grubu için de bu tanımlama yapılabilir.

Tek bir erişim koşulu bir byte üzerinde tanımlanır.

Gizli kodlar 8 byte uzunluğundadır ve 'Gizli Kod Dosyaları'nda saklanırlar. Her kod bir dosyada saklanır.

Başlıca erişim koşullarına (gizli kod, gizli anahtar) ek olarak PIN koruması da istenebilir. PIN, kart sahibine özel gizli bir koddur ve sistem dosyasından işaret edilen bir dosyada tutulur.

Gizli anahtarlara dayanan güvenli mesajlaşma mekanizması, kart ve terminal arasında gidip gelen verilerin doğruluğu ve tutarlılığını sağlamakta kullanılır. Üçlü DES (3DES - Triple DES) algoritmasını kullanır. Terminal ve kartın karşılıklı olarak birbirlerini doğrulamalarında da kullanılır.

Terminal, yolladığı komutla birlikte bunun güvenliğini sağlayacak bir de Mesaj Doğrulama Kodu (MAC_IN - Message Authentication Code_IN) gönderir. Kart, önce gelen MAC_IN'i yorumlar. Eğer doğru ise terminalin bu işlemi gerçekleştirmeye yetkisi vardır ve gelen veri tutarlıdır. Kart, gelen komut ile istenen işlevi yerine getirdikten sonra terminale MAC_OUT gönderir. Terminal MAC_OUT'a bakarak kartın doğruluğunu ve veri iletiminin tutarlılığını sınar.

Güvenlik için, gizli anahtar güncellenirken, yeni değer karta yollanmadan önce şifrelenir.

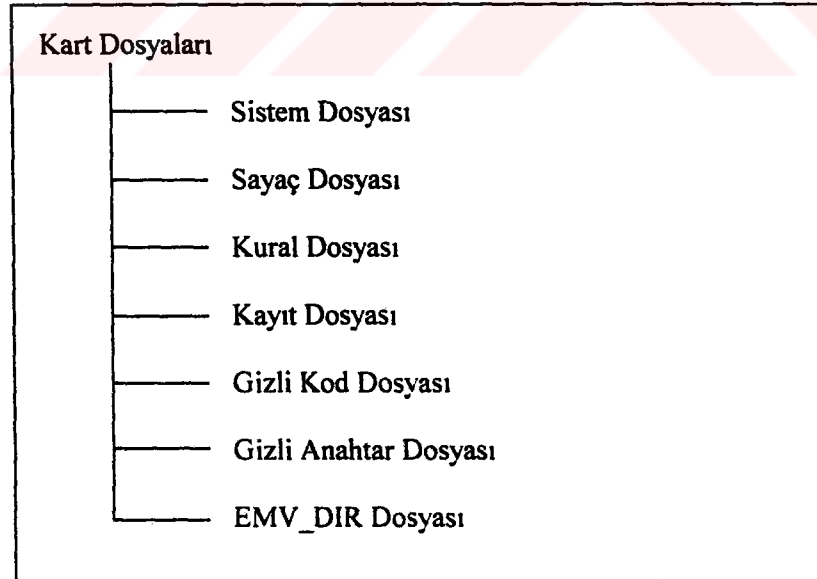
Uygulama komutundan sonra kartta işlemin sınaama değeri (checksum) oluşturulur. Burada gizli anahtarlar ve 3DES algoritması kullanılır. Oluşan sınaama değeri operatör tarafından işlemin doğrulanması için kullanılır.

Haberleşme

GemClub kartları ISO 7816-3'e uygun şekilde T=0 protokolüne göre veri gönderip alır. Ayrıca 9600 veya 115200 baud'luk yüksek hızlı iletişimi de destekler.

1. GemClub Dosyaları

Bu bölümde GemClub dosya yapısı ele alınacaktır. GemClub kartalarda düz dosya yapısı kullanılmaktadır. Bir başka deyişle, Şekil A.1'de de gösterilen tüm kart dosyaları aynı hiyerarşik seviyede yer almaktadır.



Şekil A.1 Dosya Yapısı

GemClub Dosya Organizasyonu

GemClub kartlarda aşağıdaki dosya tipleri desteklenmektedir:

- **Sistem Dosyası:** Her kartta bir tane sistem dosyası bulunmaktadır. Bu dosya kartın genel güvenlik yönetimi için kullanılmaktadır. Aynı zamanda PIN dosyasına işaretçi de burada saklanmaktadır.
- **İşlem Dosyası:** Sayaç dosyalarında puan, müşteri profili gibi bilgiler ile kural dosyalarında kuralların saklanması için kullanılırlar.
- **Güvenlik Dosyaları:** Gizli anahatar dosyaları ve gizli kod dosyaları.
- **Kayıt Dosyaları:** Veri depolanmasında kullanılırlar. ISO/IEC 7816-4 kayıt yönetimi standartlarına uygundur. Dosyaların adlandırılması ve dosyalara erişim bu standartlar çerçevesinde gerçekleştirilir.

Aşağıdaki dosya seçim yöntemleri kullanılmalıdır:

- **Dosya Tipi:** Veri dosyasının amacını belirler.
- **Kısa Dosya Belirteci (SFI – Short File Identifier):** 01h – 1Eh arasında değerler alan 5 bitlik bu sayı, dosya yaratıldığında oluşur. Sıfır değeri GemClub kartlarında geçerli değildir, reddedilecektir.

GemPlus Dosya Tipleri

GemClub işletim sisteminde Tablo A.1’de verilen dosya tipleri desteklenmektedir:

Tablo A.1 Dosya Tipleri

Dosya Tipi	Veri Nesnesi Tipi	Açıklama
Sistem Dosyası	01h	İşletim sistemi tarafından haricen yönetilen veri elemanları.
Kayıt Dosyası	02h	Terminal tarafından yönetilen kayıtlar.
Sayaç Dosyası	03h	Kart sahibinin hak ettiği puanlar.
Kural Dosyası	04h	Sayaçlar üzerinde yürütülebilecek makro-komutlar.
Gizli kod Dosyası	05h	Gizli kod.
Gizli Anahtar Dosyası	06h	Şifreleme algortimasında kullanılan gizli değer.

- Her kartta sadece bir sistem dosyası olabilir.
- Sistem dosyasının SFI değeri her zaman 01h’dir.

- Aynı tipteki iki dosya aynı SFI değerine sahip olamaz. GemClub kartlarında en fazla 16 tane kural dosyası yaratılabilir. Sistem dosyası ve kural dosyası dışındaki tiplerden en fazla 30 dosya yaratılabilir.

Kişiselleştirme Bilgileri

Kişiselleştirme bilgileri dosyalarda tutulur ve dosyanın kendi yönetimi için kullanılır. Şu bileşenleri içerebilir:

- Erişim koşulları
- PIN kodu ve EMV katalog dosyası referansları
- İşlem sınaması için anahtar değer
- Kural versiyonu
- İzin verilen maksimum deneme sayısı ve gizli kod ile gizli anahtarların doğrulanması için kalan deneme sayısını içeren onay sayaçları.

Parametre Güncelleme komutu ile, bu bilgiler dosya yaratıldıktan sonra güncellenmelidir. Komut kullanılırken, güncellenecek veri verilen bir etiket ile belirtilir. İşlemi basitleştirmek için, tüm kişiselleştirme bilgilerini aynı anda güncellemek için kullanılabilecek 'kişiselleştirme etiketi' bulunmaktadır.

Bu bölümdeki dosya tanımlarında, etiket değerlerinin yer aldığı tablolarda kişiselleştirme veri elemanları gri olarak gösterilmektedir.

Sistem Dosyası

Sistem dosyasında, kart işlem sayacı (CTC – Card Transaction Counter), işletim sistemi ve kişiselleştirme bilgilerinin de dahil olduğu 12 byte'lık bilgi bulunabilir.

GemClub kartlarda sistem dosyalarının önceden tanımlanmış olmasına rağmen, kartın işlemleri gerçekleştirebilmesi için sistem dosyasının özelliklerinin belirtilmesi gerekmektedir.

Sistem dosyası yaratıldığında kartta başka bir dosya bulunmamaktadır. Bu dosyalar daha sonraki aşamalarda oluşturulur.

Sistem dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı	01h
Dosya Tipi (Veri nesnesi tipi)	Sistem (01h)
Dosya Büyüklüğü	12 byte

Sistem dosyasının detayları Tablo A.2'de verilmiştir.

Tablo A.2 Sistem Dosyası Yapısı

Etiket	Uzunluk	Konu	Açıklama
21h	1 byte	Güncelleme / Yoketme için erişim koşulu	Sistem dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
22h	1 byte	Okuma için erişim koşulu	Sistem dosyasını okumak için gerekli yetkiyi belirler.
23h	1 byte	Yaratma için erişim koşulu	Dosya yaratmak için gerekli yetkiyi belirler.
24h	1 byte	PIN kodu dosyası referans bilgisi	Kart sahibini tanımak amacıyla kullanılan PIN olarak kullanılan gizli kod dosyası için işaretçi.
26h	1 byte	EMV_DIR dosyası referans bilgisi	EMV_DIR simülasyonu için kullanılan dosya için işaretçi.
27h	2 byte	Kart işlem sayacı	Doğrulama işlemlerinde güvenli mesajlaşma için kullanılır.

Etiket, 1 byte uzunluğunda sistemde tek (unique) olan bir sayıdır. Kullanılacak verinin belirlenmesinde kullanılır ancak kartta saklanmaz.

Kişiselleştirme etiketi 20h'dir. Bu değer kullanılarak 21h, 22h, 23h, 24h ve 26h etiketli alanların eş zamanlı olarak okunması veya güncellenmesi yapılabilir.

İşlem Dosyaları

İşlem dosyaları, sayaç dosyaları ve kural dosyalarına verilen genel isimdir.

Sayaç Dosyaları

Sayaç dosyaları, ödül ve/veya redeem uygulamalarında yer alan puanların saklanması için kullanılırlar. Sayaç dosyasına erişim kural dosyası üzerinden geçilerek gerçekleştirilir. Tek bir GemClub kartında 30 tane değişik sayaç dosyası yaratılabilir.

GemClub kartlarındaki sayaç dosyaları, belirtilen özelliklere bağlı olarak farklı yapılara sahip olabilirler. Tablo A.3'te verilen yapı alanı kullanılarak istenen tipte sayaç yaratılabilir.

Tablo A.3 Sayaç Dosyası Yapı Alanı

Bit 0	Toplam Bakiye
Bit 1	Ziyaret Sayacı
Bit 2	Geçerlilik Tarihi – ödül
Bit 3	Geçerlilik Tarihi – Redeem
Bit 4	Bu Sayaç İçin Belirtilen Kurallar
Bit 5	Başlık
Bit 6	(Gelecekteki Uygulamalar İçin)
Bit 7	(Gelecekteki Uygulamalar İçin)

Sayaç yapısına ekleme yapmak için ilgili bite ‘1’ değeri atanır. Bit ‘0’ değerinde ise ilgili veri nesnesi tipi sayaç yapısına dahil değildir.

Sayaç dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı 01h – 1Eh

Dosya Tipi (Veri nesnesi tipi) Sayaç (03h)

Dosya Büyüklüğü Belirtilen özelliklere göre 16-40 byte arasındır.

Sayaç dosyasında en azından Tablo A.4’te belirtilen detaylar yer almalıdır:

Tablo A.4 Sayaç Dosyası Yapısı

Ekler	Uzunluk	İçerik	Açıklama
61h	1 byte	Güncelleme / Yoketme için erişim koşulu	Sayaç dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
62h	1 byte	Okuma için erişim koşulu	Sayaç dosyasını okumak için gerekli yetkiyi belirler.
63h	1 byte	İşlem Sınama için anahtar referansı	İşlem sınama değeri hesaplamasında kullanılacak anahtar dosyasına ait SFI. İşlem sınamasının istendiğini belirtir.
64h	1 byte	Ödül için erişim koşulu	Ödül puanı verme yetkisini belirtir.
65h	1 byte	Geri alma için erişim koşulu	Puanları geri alma yetkisini belirtir.
70h	3 byte	Bakiye	Sayaçta o an bulunan puanları gösterir.

Kişiselleştirme etiketi olan 60h kullanılarak 61h, 62h, 63h, 64h ve 65h etiketi ile gösterilen dosya detayları eş zamanlı olarak okunup güncellenebilir.

GemClub kartlarındaki sayaçlara eklenebilecek seçimlik özellikler Tablo A.5'te verilmiştir. Bu özellikler sayaç dosya yapısına uygun bir şekilde belirlenmelidir.

Tablo A.5 Sayaç Dosyası Seçimlik Özellikler

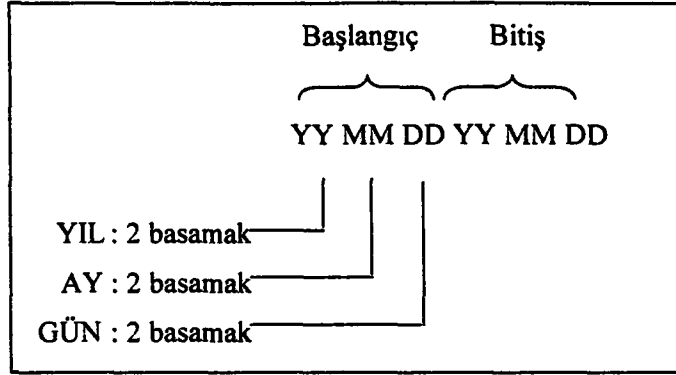
Etiket	Uzunluk	İçerik	Açıklama
71h	3 byte	Toplam Bakiye	Yaratılmasından bu yana bu sayaca verilen puanları gösterir.
72h	2byte	Ziyaret Sayacı	Yaratılmasından bu yana yürütülen ödül işlemleri sayısını verir.
73h	6byte	Ödül Geçerlilik Tarihi	Ödül işlemi için belirlenen geçerlilik tarihini verir. (Örneğin belirli bir tarihe kadar ödül veriliyorsa kullanılabilir)
74h	6 byte	Geri Alma Geçerlilik Tarihi	Geri alma işlemi için belirlenen geçerlilik tarihini verir. (Örneğin belirli bir tarihe kadar indirim uygulanıyorsa kullanılabilir)
75h	2byte	Bu Sayaç İçin İzin Verilen Kurallar	Bu sayaç üzerinde yürütülebilecek kuralları belirler. Bu alan boşsa ' <i>Kural Kullan</i> ' komutu ile sayaç detayları değiştirilemez. Aksi alde her bit bir kurala karşı gelir.
76h	8byte	Başlık	Sayacın adının ve sürümünün belirtildiği alfanümerik değer. Kart ömründe tekrar tekrar kullanılabilir.

Eğer sistem tarihi, belirlenmiş ödül/geri alma tarih aralığında değilse bu işlem reddedilir. Tarih bilgilerinin kodlanması aşağıda belirtilen kriter kontrol edilebilecek şekilde yapılmalıdır.

Başlangıç Tarihi <= Sistem Tarihi <= Bitiş Tarihi

İşletim sistemi, tarih formatları üzerinde hiçbir kontrol uygulamaz.

2000 yılı probleminin giderilmesi için GemClub kartlarında tarihler BCD formatında Şekil A.2'deki gibi tutulmaktadır.



Şekil A.2 GemClub Tarih Gösterimi

2000 yılının 1900 yılı olarak yorumlanmaması için terminal tarihi göndermeden önce Tablo A.6’da gösterildiği gibi değiştirir:

Tablo A.6 GemClub Tarih Dönüşümü

<u>Yıl</u>	<u>Dönüşüm Sonrası</u>
1997	‘00’
1998	‘01’
1999	‘02’
2000	‘03’
2001	‘04’

Kural Dosyaları

Kural dosyası, ‘*Kural Kullan*’ komutu sırasında yürütülen makrokomut kümesidir. Kurallar kullanılarak en fazla dört sayaç birleştirilebilir. Bir GemClub kartında maksimum 16 tane kural dosyası olabilir.

Kural dosyasının yapısı, dosya için belirlenen özelliklere bağlıdır. Kural dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 10h
Dosya Tipi (Veri nesnesi tipi)	Kural (04h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 12 + 8n byte.
(n : makrokomut sayısı)	

Kural dosyasında Tablo A.7’deki detaylar mutlaka yer almalıdır.

Tablo A.7 Kural Dosyası Yapısı

Etiket	Uzunluk	İçerik	Açıklama
81h	1 byte	Güncelleme / Yoketme için erişim koşulu	Kural dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
82h	1 byte	Okuma için erişim koşulu	Kural dosyasını okumak için gerekli yetkiyi belirler.
83h	1 byte	İşlem Sınama için anahtar referansı	İşlem sınama değeri hesaplamasında kullanılacak anahtar dosyasına ait SFI. İşlem sınamasının istendiğini belirtir.
84h	1 byte	Kural kullanma için erişim koşulu	'Kural Kullan' komutu ile bu kuralı çalıştırma yetkisini belirler.
85h	1 byte	Sürüm	Kuralın sürümünü gösterir.
90h	8 byte	Makrokomut 1	Belirlenen sayaç zerinde yürütülecek hareketi belirtir. Sayaç için yaratılan her kural için en az bir makro yazılmalıdır.

Kişiselleştirme etiketi 80h'dir. Bu etiket kullanılarak 81h, 82h, 83h, 84h ve 85h etiketli detaylar aynı anda güncellenebilir ya da okunabilir.

Yaratılan diğer makrokomutlar da Tablo A.8'de tarif edilen şekilde dosyaya eklenebilir.

Tablo A.8 Kural Dosyasına Başka Makrokomut Eklenmesi

Etiket	Uzunluk	İçerik	Açıklama
91h	1 byte	Makrokomut 2	Bu sayaç için kullanılacak komutu tanımlar.
92h	1 byte	Makrokomut 3	Bu sayaç için kullanılacak komutu tanımlar.
93h	1 byte	Makrokomut 4	Bu sayaç için kullanılacak komutu tanımlar.

Kural dosyasında en fazla dört makrokomut saklanabilir.

Belirli bir sayaç için tanımlanan işlemin (ödül veya indirim) mantıksal ifadesine 'Makrokomut' adı verilir. Makrokomutlar, Tablo A.9'da verilen bilgilerden oluşurlar.

Tablo A.9 Makrokomut Yapısı

Alan	Uzunluk	Açıklama
Fonksiyon	1 byte	Belirlenen sayaca uygulanacak işlem. 00h – Makrokomutu geç 01h – Komutun ödül işlemi için olduğunu belirtir. 02h – Komutun indirim işlemi için olduğunu belirtir. (Diğer değerler gelecekteki kullanımlar için ayrılmıştır.)
Sayaç Referansı	1 byte	Makrokomutun üzerinde yürütüleceği sayaç dosyasının SFI değeri.
Parametre A	3 byte	İlk dönüşüm parametresi. Parametre B’de belirtilen değere erişildiğinde sayaca eklenecek veya sayaçtan çıkarılacak puan sayısını verir.
Parametre B	3 byte	İkinci dönüşüm parametresi. Kart için belirtilen sayaça puan eklemekten ya da çıkarmadan önce erişmesi gereken değeri verir.

Makrokomutun yürütülebilmesi için sayaçtaki ‘izin verilen kurallar’ alanında bu kurala izin verilmiş olması gerekmektedir.

Güvenlik Dosyaları

Güvenlik dosyaları, gizli kod ve gizli anahtar dosyalarından oluşmaktadır.

Gizli Kod Dosyaları

Gizli kodlar dosya korumada kullanılırlar. Herbir gizli kod dosyasında sadece bir gizli kod bulunabilir. Bunların yanısıra kullanılabilecek olan PIN kodu da bir gizli koddur ve hangi dosyada bulunduğu sistem dosyasında belirtilmektedir. Bir dosyanın gizli kod veya PIN kodu ile korunabilmesi için o dosyanın erişim koşulunda söz konusu gizli kod veya PIN koduna yapılan bir referans yer almalıdır.

Gizli kod dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Gizli Kod (05h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 16 byte

Gizli kod dosyasında Tablo A.10’da verilen alanlar bulunmalıdır.

Tablo A.10 Gizli Kod Dosyasının Yapısı

Etiket	Uzunluk	İçerik	Açıklama
A1h	1 byte	Güncelleme / Yoketme için erişim koşulu	Gizli kod dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
A3h	1 byte	Maksimum deneme sayısı ve Kalan deneme sayısı	İzin verilen maksimum deneme sayısı ve kart reddedilip gizli kodun engellenmesine kadar kalan deneme hakkı sayısı.
B0h	8 byte	Gizli kod değeri	Gizli kod değeri.

Kişiselleştirme etiketi A0h'dir. Bu etiket kullanılarak A1h ve A3h etiketli detaylar aynı anda güncellenebilir.

Maksimum deneme sayısı olarak en fazla 15 verilebilir. Kalan deneme sayısı da en fazla 15 olabilir. Gizli kodun doğru olarak verilmesiyle birlikte kalan deneme sayısı, maksimum deneme sayısına eşitlenir.

Gizli Anahtar Dosyaları

GemClub'daki gizli anahtar değerleri güvenli mesajlaşmanın sağlanması ve işlem sinama değerinin hesaplanmasında kullanılırlar. Güvenli mesajlaşma amaçlı gizli anahtar kullanımı için ilgili SFI değerinin korunacak dosyanın erişim koşulunda belirtilmesi gerekmektedir. Erişim koşulunda da güvenli mesajlaşmanın kullanılacağı belirtilmelidir. Gizli anahtar işlem sinama değeri hesabında kullanmak için SFI, ilgili kural veya sayaç dosyasında belirtilmelidir. Herbir gizli anahtar dosyasında sadece bir anahtar değeri yer alabilir.

Gizli anahtar dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Gizli Anahtar (06h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 24 byte

Gizli anahtar dosyasında Tablo A.11'deki alanlar bulunmalıdır:

Tablo A.11 Gizli Anahtar Dosyasının Yapısı

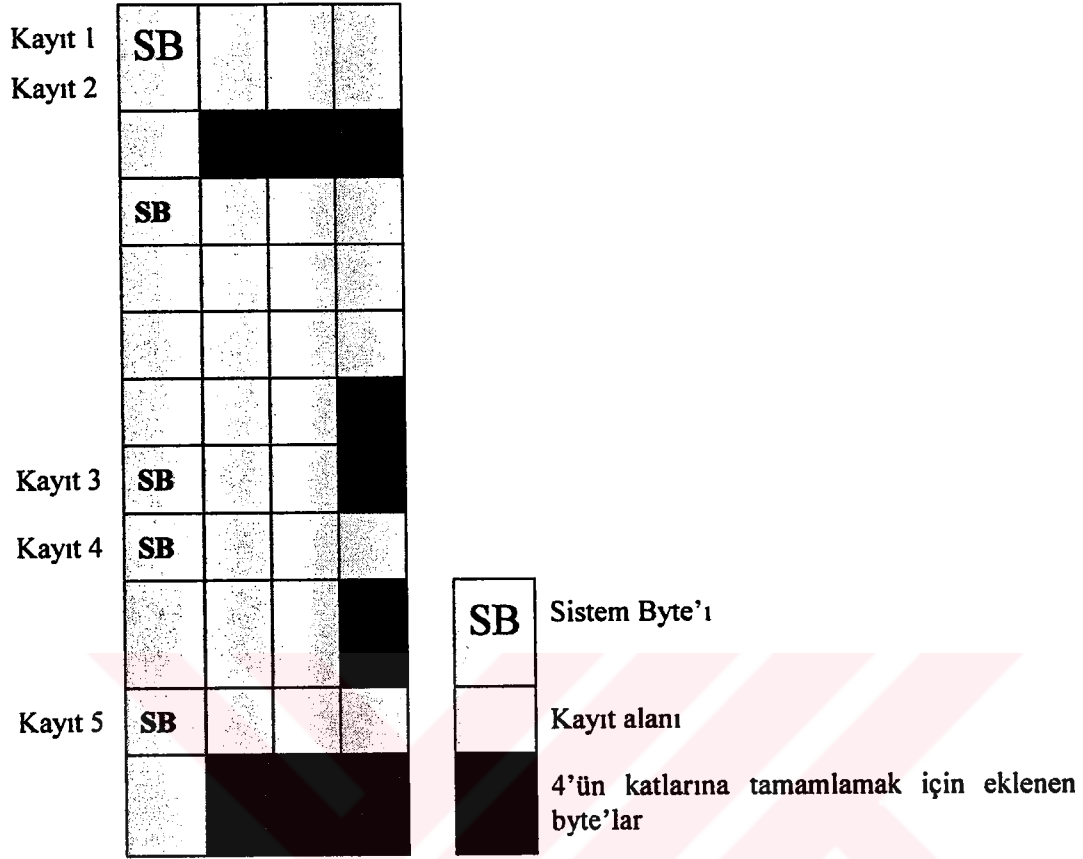
Etiket	Boyut (byte)	İşlev	Açıklama
C1h	1 byte	Güncelleme / Yoketme için erişim koşulu	Gizli anahtar dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
C3h	1 byte	Maksimum deneme sayısı ve Kalan deneme sayısı	İzin verilen maksimum deneme sayısı ve kart reddedilip gizli anahtarın engellenmesine kadar kalan deneme hakkı sayısı.
D0h	8 byte	Gizli anahtar değeri	3DES standardına uygun olarak,3DES anahtarı olarak kullanılacak değerin yarısı. Güncelleme işlemi sırasında gizli anahtarın iki yarısına ayrı ayrı erişilir.
D1h	8 byte	Gizli anahtar değeri	3DES standardına uygun olarak,3DES anahtarı olarak kullanılacak değerin yarısı. Güncelleme işlemi sırasında gizli anahtarın iki yarısına ayrı ayrı erişilir.

Kişiselleştirme etiketi C0h'dir. Bu etiket kullanılarak C1h ve C3h etiketli detaylar aynı anda güncellenebilir.

Maksimum deneme sayısı olarak en fazla 15 verilebilir. Kalan deneme sayısı da en fazla 15 olabilir. Gizli anahtarın doğru olarak verilmesiyle birlikte kalan deneme sayısı, maksimum deneme sayısına eşitlenir.

Kayıt Dosyaları

Kayıt dosyaları, GemClub kartlarında verilerin depolandığı yerlerdir. Kartlarda Şekil A.3'te de gösterildiği gibi değişken uzunluklu kayıtlar içeren doğrusal dosyalar kullanılmaktadır.



Şekil A.3 Değişken uzunluklu kayıtlar içeren doğrusal dosya

Dosyadaki ilk kayıt '1' olarak numaralandırılmıştır. Doğrusal değişken kayıt dosyasında en fazla 254 kayıt bulunabilir. Bir kayıt en fazla 255 byte uzunlukta olabilir. Güvenli mesajlaşma durumunda kaydın maksimum uzunluğu 24 byte ile kısıtlıdır.

Kayıt dosyasının yapısı belirlenen özelliklere göre değişir. Kayıt dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Kayıt (02h)
Dosya Büyüklüğü	(Değişken)

Kayıt dosyasında Tablo A.12'de verilen alanlar bulunmalıdır.

Tablo A.12 Kayıt Dosyasının Yapısı

Etiket	Uzunluk	İçerik	Açıklama
41h	1 byte	Güncelleme / Yoketme için erişim koşulu	Kayıt dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
42h	1 byte	Okuma için erişim koşulu	Kayıt dosyasını okumak için gerekli yetkiyi belirler.
	L ₁	Kayıt 1	Veri
	L ₂	Kayıt 2	Veri
	L _n	Kayıt N	Veri

Kişiselleştirme etiketi 40h'dir. Bu etiket sayesinde 41h ve 42h etiketlerine sahip alanlar aynı anda güncellenebilir, okunabilir.

Kayıt dosyasının uzunluğu, dosya tanımlayıcısı ve erişim koşulları uzunluğu ile kayıtların uzunluğunun toplamıdır. Kayıt dosyasında yer alan kaydın uzunluğu ise şöyle ifade edilebilir:

Kayıt uzunluğu = (1 byte - SB) + (veri için kullanılan byte'lar) + (4'e tamamlama için kullanılan byte'lar)

Bu hesaplama göre Şekil A.3'te gösterilen dosyanın uzunluğu =

8 byte (dosya tanımlayıcısı) +

8 byte (Kayıt 1 : 1 byte SB+4 byte kullanılan+3 byte tamamlama) +

16 byte (Kayıt 2 : 1 byte SB+14 byte kullanılan+1 byte tamamlama) +

4 byte (Kayıt 3 : 1 byte SB+2 byte kullanılan+1 byte tamamlama) +

8 byte (Kayıt 4 : 1 byte SB+6 byte kullanılan+1 byte tamamlama) +

8 byte (Kayıt 5 : 1 byte SB+4 byte kullanılan+3 byte tamamlama) = 52 byte'tır.

İsimle Referans Verme (EMV Simülasyonu)

EMV özellikleri sadece GemClub-EMV kartlarıyla kullanılabilir. EMV kart spesifikasyonlarında belirtilen '*Uygulama dosyasını ismiyle seç*' komutunun simülasyonu gerçekleştirilerek GemClub-EMV kartlarının bu standarda uyumu sağlanmıştır.

ISO/IEC 7816-4 standardı ile belirtilen isimler ile katalog seçimi de GemClub kartları tarafından desteklenmektedir. GemClub tarafında kabul edilen kayıt dosyası isimleri EMV_DIR dosyasında da kayıtlı olmalıdır.

Uygulama dosyasının adı, isimle seçim için sağdan kesilebilir.

Ödeme Sistemi Ortamı (PSE – Payment System Environment) de isimle seçim için sağdan kesilebilir.

Tarama algoritması karttaki EMV_DIR dosyasının içeriğini tarar. Algoritma, uyumlu ilk kaydı bulduğunda durur.

GemClub kartları, EMV standartlarına göre yapılandırılmış dosyaları da destekler. EMV_DIR uyumlu kayıt dosyasının yaratılmasına örnek aşağıda verilmiştir.

EMV_DIR dosyası, sistem dosyasında belirlenen ve EMV simülasyonunda görev alan bir kayıt dosyasıdır. Dosyadaki ilk kayıt EMV PSE yapısına (1PAY.SYS.DDF01) uygun olmalıdır. Etiket olarak 9Dh kullanılmalıdır.

EMV_DIR dosyasındaki her kayıt, EMV kart spesifikasyonları uyarınca basit bir etiket boyu değeri (TLV – Tag Length Value) kullanılarak yapılandırılmalıdır. EMV_DIR dosyasındaki zorunlu alanların başlangıcı Tablo A.13'te gösterilmiştir:

Tablo A.13 EMV_DIR Dosyasının Yapısı

	Etiket	Uzunluk	İçerik	Açıklama
Kayıt 1	70h	1 byte	Veri Taslağı	
= DDF	Değişken	1 byte	Uzunluk	Veri Taslağı
	61h	1 byte	Uygulama Taslağı	
	Değişken	1 byte	Uzunluk	Uygulama Taslağı
	9Dh	1 byte	Etiket	Katalog Tanım Dosyası Adı (DDF – Directory Definition File) Etiket = 9Dh
	5-16	1 byte	Uzunluk	Uygulama Tanımlayıcısı
	-	5-16 byte	Uygulama Tanımlayıcısı	Kayıt dosyası tarafından tanınan uygulama adı.
	50h	1 byte	Etiket	Uygulama Başlığı
	1-16	1 byte	Uzunluk	Uygulama Başlığı
	-	1-16 byte	Uygulama Başlığı	Alfanümerik değer.

Kayıt 2	70h	1 byte	Veri Taslağı	
= ADF	Değişken	1 byte	Uzunluk	Veri Taslağı
	61h	1 byte	Uygulama Taslağı	
	Değişken	1 byte	Uzunluk	Uygulama Taslağı
	4Fh	1 byte	Etiket	Uygulama Tanım Dosyası Adı (ADF – Application Definition File) Etiket = 4Fh
	5-16	1 byte	Uzunluk	Uygulama Tanımlayıcısı

-	5-16 byte	Uygulama Tanımlayıcısı	Kayıt dosyası tarafından tanınan uygulama adı.
50h	1 byte	Etiket	Uygulama Başlığı
1-16	1 byte	Uzunluk	Uygulama Başlığı
-	1-16 byte	Uygulama Başlığı	Alfanümerik değer.

Kayıt 3 = ADF	...	...	...	
------------------	-----	-----	-----	--

2. GemClub Başlangıç Durumu

Bu bölümde GemClub kartlardaki tüm bileşenlerin başlangıç durumları ele alınmıştır.

Koşullama İşlemi

İmalat sırasında, mikroişlemcili kartların koşulanmaları gerekmektedir. Bu aşamada karta aşağıda belirtilen temel bilgiler yazılır:

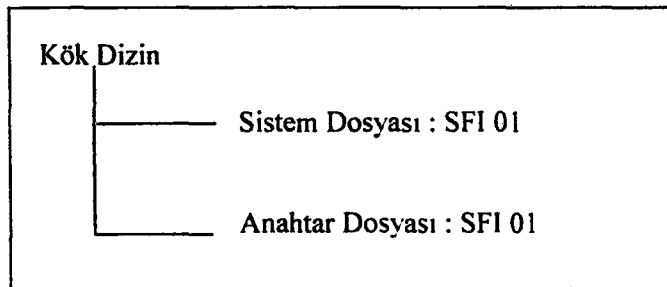
- Kart seri numarası ve Kart tedarikçisi referansı
- Sistem dosyası yapısı
- Koruma anahtarları
- Filtre yaratılması (GemClub-EMV kartları için)

İki çeşit kart koşullama işlemi bulunmaktadır:

- Müşteri kartı için: Kartlar farklı anahtarlar tarafından korunmaktadır.
- Örnek kartlar için: Geliştirme setleriyle birlikte gönderilen kartlar ortak bir anahtar ile korunmaktadır.

Başlangıçtaki Dosya Yapısı

Şekil A.4'te başlangıç druumunda GemClub kartı üzerindeki dosya yapısının nasıl olduğu gösterilmiştir.



Şekil A.4 Başlangıçtaki Dosya Yapısı

Sistem Dosyası

Karta ilişkin genel erişim koşulları sistem dosyasında tanımlanmaktadır. Dosya parametreleri Tablo A.14'te verilmiştir.

Tablo A.14 Sistem Dosyası Başlangıç Parametreleri

<u>Hareket Erişim Koşulları ve Parametreleri</u>	<u>Örnek Kart ve Müşteri Kartı Prosesleri</u>
Güncelleme / Yoketme	Anahtar 01
Okuma	Serbest
Yaratma	Anahtar 01
PIN Kodu Dosyası Referansı	00h
EMV_DIR Dosyası Referansı	00h
Kart İşlem Sayacı	0005h

Anahtar Dosyası (SFI 01)

Bu dosyada, kart uygulaması yaratıcısı sağlayıcısı tarafından kişileştirme işlemlerinde kullanılan 01 anahtarı bulunmaktadır. Değeri, kartın başlangıç durumuna bağlıdır:

Örnek proses tarafından koşullama yapılmış kartlarda sabit ASCII katarı 'TESTKEY1TESTKEY2'dir.

Müşteri kart prosesi tarafından ilklendirilen uygulama kartlarındaki anahtar, müşteri kartında yer alan müşteri anahtarı kullanılarak oluşturulan farklı bir değere sahiptir.

Dosya parametreleri Tablo A.15'te verilmiştir.

Tablo A.15 Anahtar Dosyası Parametreleri

<u>Belirtec</u>	<u>Uzunluk</u>
01h	24 byte

Güvenlik erişim koşulları ise Tablo A.16'da gösterildiği gibidir.

Tablo A.16 Erişim Koşulları

<u>İşlem</u>	<u>Erişim Koşulu</u>
Güncelleme / Yoketme	Anahtar 01 ile Güvenli Mesajlaşma

Sayaçlardaki maksimum deneme ve kalan deneme sayılarına ilişkin parametreler '3' değerine ayarlanmıştır.

Kart Seri Numarası

Kart seri numarası her bir GemClub kartı için tektir ve değiştirilemez. Şekil A.5'te gösterildiği gibi 8 byte'lık alana 32 bitlik iki sözcük olarak kodlanmıştır.

Bitler																																																																																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00																																																				
[1										10 bit																						]		[2										22 bit										]																													
[3										10 bit																						]		[4						6 bit						]		[5								8 bit								]		[6								8 bit								]	
1 : Batch numarası (10 bit)																																																																																			
2 : Seri Numarası (22 bit)																																																																																			
3 : Seri Numarası (10 bit)																																																																																			
4 : Hafta Numarası (06 bit)																																																																																			
5 : Seri Numarası (08 bit)																																																																																			
6 : Üretim Yılı (08 bit)																																																																																			

Şekil A.5 Seri Numarası

2, 3 ve 5. alanlar birleşerek seri numarasını oluştururlar. Seri numarası salt-okur bir parametredir.

Kart Tedarikçisi Referans Numarası

Kart tedarikçisi referans numarası üç byte ve sınav kontrolü byte'ından müteşekkil olmak üzere toplam dört byte'tan oluşmaktadır. Bu sayı her müşteri için Gemplus tarafından verilmektedir; daha sonra değiştirilemez. Örnek kartlarda kart tedarikçisi referans numarası 00 FF 00 10h olarak belirlenmiştir.

G/Ç Tampon Uzunluğu

Güvenli mesajlaşma kullanılmadığı zaman bir seferde 255 byte'lık bir kaydı güncellemek mümkündür; ancak terminal veya okuyucu kaynaklı kısıtlamalar sonunda bu rakam daha düşük olabilir. Güvenli mesajlaşma kullanılıyorsa en fazla 24 byte'lık veri söz konusudur.

EEPROM Büyüklüğü

Tablo A.17'de belirtilen bellek büyüklükleri koşullama sonrası değerlerdir.

Tablo A.17 EEPROM Büyüklüğü

Ürün	Sistem ve anahtar dosyaları oluşmadan EEPROM	Sistem ve anahtar dosyaları oluştuktan sonra kalan EEPROM
GemClub 1K	952 byte	916 byte
GemClub EMV	748 byte	712 byte

3. GemClub Güvenlik Mimarisi

GemClub kartlarda bulunan güvenlik mekanizmaları beş başlıkta toplanabilir:

- Güvenlik durumları
- Güvenlik özellikleri: Kartta bulunan dosyalar üzerinde yapılan hareketlerin yetki kontrolünü sağlayan güvenlik spesifikasyonları.
- Güvenlik mekanizmaları: Karttaki dosyaların güvenliğini sağlayan prosedürler. Bunlar, ayrıca kart ve terminal arasındaki haberleşmenin bozuk olmamasını garantilerler.
- İşlem sınamaları: Operatöre, işlemlerin doğruluğunu bildiren prosedürler.
- Erişim koşulu kontrol algoritması: Kullanıcı veya kart sağlayıcının, kart üzerinde işlem yapmaya yetkili olup olmadığını gösteren algoritma.

Güvenlik Durumları

Komut yürütülmesi sırasında, güvenli mesajlaşmanın başarıyla tamamlanması gibi durumları kaydetmek gerekebilir. Bu kayıtlara ‘Güvenlik Durumu’ adı verilir. İki çeşit güvenlik durumu vardır:

- Komuta özel güvenlik durumu: Anahtar doğrulamaları kaydederler. Ayrıca, komutun veri alanlarında bulunması durumunda, gizli kod doğrulamalarını da saklarlar. Bu durum kaydı, komutun yürütülmesi sırasında RAM’de tutulur. Bunların, genel güvenlik durumu üzerinde bir etkisi yoktur.
- Genel güvenlik durumu: Komuttan önce yürütülen ‘Gizli Kod Doğrula’ komutu sırasında, gizli kod verilirken kullanılır. PIN doğrulaması ve gizli kodun komut alanında iletilmediği ISO 7816-4 standartında tanımlanan komutlarda bu duruma ihtiyaç duyulmaktadır. Genel güvenlik durumunda, PIN kodu ve belirtilen son gizli kodun kaydı yer almaktadır. ‘Gizli Kod Doğrula’ komutu kullanılmadan verilen gizli kodlar bu durumda yer almaz. RAM’de tutulan genel güvenlik durumu kart oturumunun sonunda silinir.

Güvenlik Özellikleri

GemClub kullanıcılarının, karttaki dosyalar üzerinde herhangi bir işlem yapmadan önce yerine getirmeleri gereken, önceden tanımlanmış güvenlik isterlerine ‘Güvenlik Özellikleri’ adı verilir. Kart üzerinde dosya yarattıktan sonra bunlar için geçerli olacak erişim koşulları belirlenebilir. Erişim koşulları, belirli bir fonksiyon veya fonksiyon grubu için tanımlanır. İlgili veriyi korumada kullanılan gizli kod ve gizli anahtarları kullanır.

Erişim koşulu tanımlanırken Tablo A.18’deki fonksiyonlar kullanılabilir:

Tablo A.18 Eriřim kořulları iin kullanılan fonksiyonlar

Fonksiyon	Aıklama	Veri Nesnesi
Yaratma	Veri objesi yaratma.	Hepsi
Güncelleme/Yoketme	Veri elemanı güncelleme veya veri nesnesi yoketme. Kayıt güncelleme/ekleme.	Hepsi
Okuma	Veri elemanı/kaydı okuma.	Güvenlik dosyaları ıřında hepsi.
Ödöl	Ödöl komutu yürütme.	Sayalar
İndirim	İndirim komutu yürütme.	Sayalar.
Kural Kullan	'Kural Kullan' komutu yürütme.	Kurallar.

Güncelleme ve yoketme fonksiyonları aynı erişim kořulu ile korunur.

Eriřim kořulu, bir byte üzerinde Şekil A.6'da gösterildiđi gibi tanımlanır:

b7	b6	b5	B4	b3	b2	b1	b0
PIN	Koşul		Kısa Dosya Tanımlayıcısı - SFI				
B7 : 1 – PIN kodu zorunlu 2 – PIN kodu zorunlu değil. B6-b5 : 00 – Gizli kod koruması ya da güvenli mesajlaşma gerekli değil. 01 – Gizli kod koruması gerekli 10 – Güvenli mesajlaşma gerekli 11 – Erişim kilitlenir. B4-b0 : b6-b5'te belirtilen koşulda geçen gizli kod veya gizli anahtar dosyasının tanımlayıcısı.							

Şekil A.6 Eriřim kořulu yapısı

PIN olarak kullanılacak gizli kod dosyası sistem dosyasında belirtilmektedir.

GemClub'da erişim kořulları iin belirlenen kodlara ait kurallar Tablo A.19'da verilmiřtir.

Tablo A.19 Erişim koşullarında yer alan kodlar için kurallar

Gerekenler	Yapılacak
Gizli kod koruması ya da güvenli mesajlaşma gerekmiyor.	x00xxxxxb kullanılmalıdır.
Belirli bir fonksiyon için ‘Erişim Aç’	000xxxxxb kullanılmalıdır.
Belirli bir fonksiyonun kilitli olduğunu kontrol et	x11xxxxxb kullanılmalıdır.
Sadece PIN kod kontrolü	100xxxxxb kullanılmalıdır.
Sadece güvenli mesajlaşma	010xxxxxb kullanılmalıdır.
Sadece gizli kod	001xxxxxb kullanılmalıdır.
Güvenli mesajlaşma ve PIN kontrolü	110xxxxxb kullanılmalıdır.
Gizli kod ve PIN kontrolü	101xxxxxb kullanılmalıdır.

Gizli kodlar, her erişim koşulu grubu için dosyalara erişimi güvenli kılarlar. Sistem dosyasında PIN kodu referansı olarak 00h belirtilmişse erişim koşulu doğrulaması sırasında PIN kodu kontrolü es geçilir. Okuma işlemleri için güvenli mesajlaşma desteklenmemektedir. Eğer okumalar için güvenli okuma seçeneği belirtilirse bu ‘kilitli’ okuma olarak algılanır.

GemClub kartlarındaki her dosyanın erişim koşulları vardır. Her bir erişim koşulu dosyaya tanımlı komut/fonksiyon grubu için tanımlanır.

Bazı veri elemanlarına ilişkin okuma işlemi için işletim sisteminde bir erişim koşulu yer almaktadır.

Tablo A.20’de gösterilen veri elemanlarının okunması için işletim sisteminde dolaylı erişim koşulu bulunmaktadır.

Tablo A.20 Dolaylı Erişim Koşuluna Sahip Veri Elemanları

Veri Nesnesi	Veri Elemanı	Okuma İçin Erişim Koşulu
Gizli kod dosyası	Tüm veri elemanları	Kilitli
Gizli anahtar dosyası	Tüm veri elemanları	Kilitli

Güvenlik Mekanizmaları

Güvenlik mekanizmaları, karttaki dosyaların güvenliğini sağlamak için kullanılan yordamlardır. Bu mekanizmalar sayesinde kartın yetkisiz kullanımı da engellenmiş olur.

Güvenlik Mekanizmalarının Çalışma Şekli

Kart, üzerindeki ilgili gizli kod ile terminalden aldığı kodu karşılaştırır. İşletim sisteminin kullanacağı dahili gizli kod şunlardan biri olabilir:

- Komut içinde belirtilmiş ('*Gizli Kod Doğrula*' komutu ile)
- Kart tarafından dolaylı olarak bilinen ('*PIN Kodu Doğrula*' komutu ile)
- Komutta geçen veri nesneleri için belirtilen erişim koşulları içinde ('*Ödül*', '*Kuralı Kullan*' komutları ile)

Gizli Kod İle Varlık Doğrulama

GemClub'da her gizli kod için tutulan 'Gizli Kod Onay Sayacı' kullanılır. Bu sayaç sayesinde gizli kod üzerinde yapılabilecek mükerrer saldırılar önlenabilir. Sayaç şu elemanlardan oluşur:

- İzin verilen azami deneme sayısı: Dosyaya erişim sırasında ardışıl olarak gerçekleşecek azami yanlış deneme sayısını verir. Bu parametrenin değeri en fazla 15 olarak belirtilebilir.
- Kalan deneme sayısı: Dosya erişiminin engellenmesine kaç deneme kaldığını gösterir. Parametre değeri en fazla 15 olabilir. Başarılı bir denemeden sonra bu sayı, izin verilen maksimum deneme sayısı ile güncellenir. Kalan deneme sayısı değeri, her zaman izin verilen değere eşit ya da bu değerden daha küçüktür.

Gizli Kod Yönetim Kuralları

Gizli kod yönetiminde aşağıdaki kurallar geçerlidir:

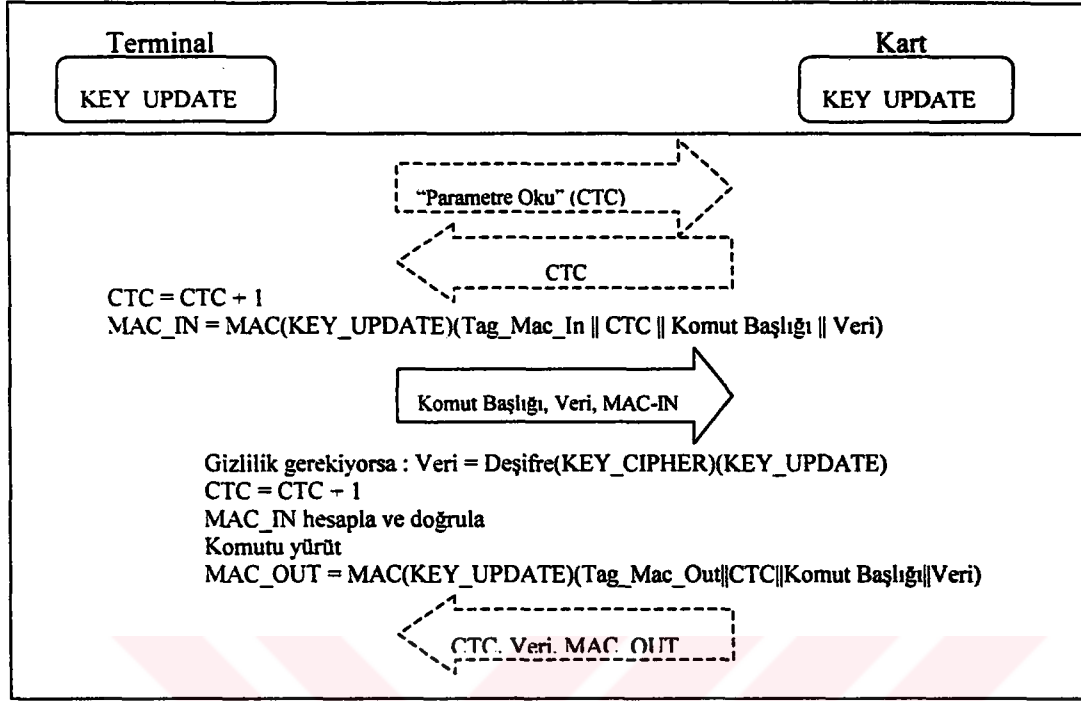
- Gizli kod bloke olmuşsa doğrulama yapılamaz, çünkü maksimum deneme sayısına ulaşılmıştır.
- Sunulan bir gizli kod ile karttaki gizli kod karşılaştırılmadan önce EEPROM'daki izin verilen deneme sayısı bir azaltılır.
- Yanlış gizli kod verilmesi nedeniyle gizli kodun bloke olması durumu dışında, yanlış bir gizli kod girişi genel güvenlik durumunu etkilemez. Bu duruma işletim sistemi genel güvenlik durumunda saklanan bilgileri iptal eder.
- Bloke olmuş bir gizli kodu etkinleştirmek için 'Kalan Deneme Sayısı' değerini güncellemek gerekir. Bunun için de güncelleme/yoketme erişim koşulunun sağlanması zorunludur. Bu da sadece gizli kod dosyası kendisi tarafından korunmuyorsa mümkündür.

Veri Doğrulama – Güvenli Mesajlaşma

Söz konusu mekanizma ile kart ile terminal arasındaki veri iletiminin doğruluğu ve tutarlılığı ve anahtar güncellemelerinin gizliliği sağlanmış olur.

Terminal, korunacak komut ile birlikte MAC_IN değerini de gönderir. Bu değeri alan kart da sınamaya işlemlerini gerçekleştirecek, böylece terminalin ilgili işlemi yapmaya yetkili olup olmadığını ve verinin tutarlı bir biçimde iletilip iletilmediğini öğrenebilecektir.

'*Parametre Güncelle*' komutunun doğrulanmasına ilişkin bir örnek Şekil A.7'de yer almaktadır.



Şekil A.7 Parametre Güncelleme için güvenli mesajlaşma

Birinci Adım:

Terminal, ilgili işlem için önceden belirlenmiş olan gizli anahtar değerini kullanarak MAC_IN değerini hesaplar. MAC_IN değerinde aşağıdaki detaylar bulunmalıdır:

- Tag_Mac_In
- CTC
- Komut başlığı
- Karta gönderilecek mesaj

Anahtar değerinin güncellenmesi için gizlilik gerekiyorsa terminal mesajı DES3 algoritmasını kullanarak şifreler.

İkinci Adım:

İkinci adımda, kart (gizlilik gerekiyorsa) gelen mesajı deşifre eder; hesaplamalardan sonra terminalden aldığı MAC değerini doğrular.

Komutun sonunda, kart ilgili işlem için önceden tanımlanmış, MAC_IN hesabındaki değer olan gizli anahtarı kullanarak MAC_OUT'u hesaplar. Bu MAC_OUT değeri içinde şu detay bilgileri yer alır:

- Tag_Mac_Out
- CTC
- Komut Başlığı
- Terminale gönderilecek mesaj

Doğrulama işlemi başarı ile tamamlanırsa, terminalin ilgili işlemi yürütmeye yetkili olduğu anlaşılır. Bu durumda güvenli mesajlaşmaya ilişkin onay sayacı sıfırlanır. Doğrulamanın başarısız olması durumunda kart kalan deneme sayısını bir azaltır ve komutu reddeder.

Üçüncü Adım:

Terminal MAC_OUT değerini doğrular.

Gizli Anahtar Onay Sayacı

‘Gizli anahtar onay sayacı’ her gizli anahtar dosyasında bulunur ve bu dosya üzerinde gerçekleşebilecek mükerrer saldırıları önleneyi amaçlar. Sayaç şu bileşenleri içerir:

- İzin verilen maksimum deneme sayısı: Dosyaya erişimde izin verilen maksimum ardışıl yanlış deneme sayısıdır. Bu değer en fazla 15 olabilir.
- Dosyaya erişimin bloke edilmesine kaç deneme kaldığını gösterir. Başarılı bir doğrulama işleminden sonra, izin verilen maksimum deneme sayısına eşitlenir.

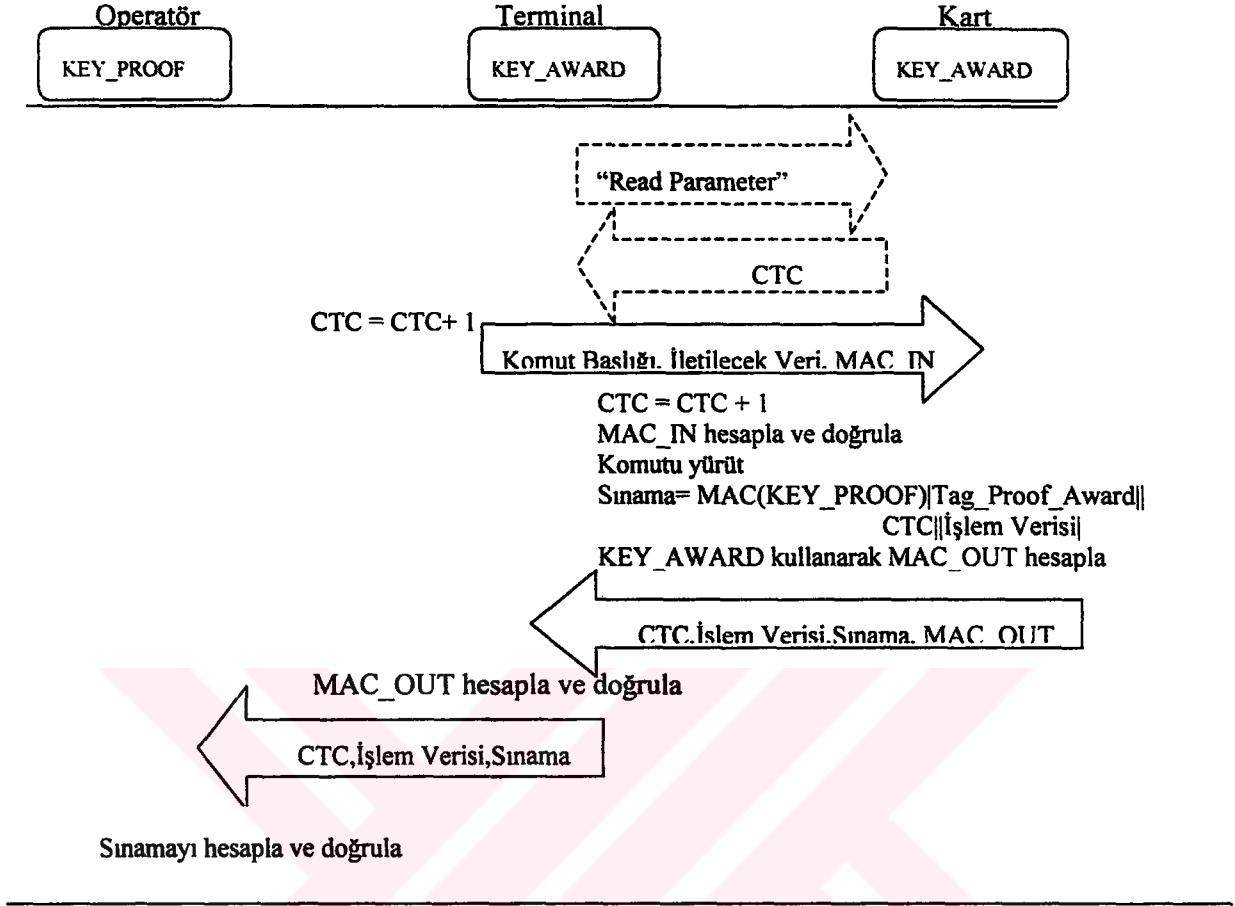
Kalan deneme sayısı her zaman izin verilen maksimum deneme sayısından küçüktür veya bu sayıya eşittir.

Gizli anahtar bloke olduğunda dosyaya erişim de engellenir. EEPROM’da tutulan sayaç değeri, terminalden gelen MAC_IN ve kartta hesaplanan MAC_IN değerlerinin karşılaştırılmasından önce bir azaltılır.

İşlem Sınama

GemClub kartları tarafından belirli işlemlerden sonra üretilen işlem sınama değeri, operatöre işlemin gerçekliğini gösterir. Örneğin, ‘Ödül’ işlemi sınamaya tabi tutulabilir.

Şekil A.8’deki örnekte kart, operatör tarafından sınanacak bir değer üretmektedir. Söz konusu işlem ‘Ödül’, ‘Geri Alma’ veya ‘Kural Kullan’ olabilir. Örnekte veri doğrulama adına güvenli mesajlaşma kullanılmaktadır. Ancak, veri doğrulama dışında da işlem sınama değeri üretilebilir.



Şekil A.8 – İşlem Sınama

Birinci Adım:

Terminal, ilgili işlem için önceden belirlenmiş olan gizli anahtar değerini (KEY_AWARD) kullanarak MAC_IN değerini hesaplar. MAC değerinde aşağıdaki alanlar bulunur:

- Tag_Mac_Award_In
- CTC
- Komut başlığı
- Karta iletilecek mesaj

İkinci Adım:

Kart, gelen MAC_IN değerini doğrular. İşlem başarı ile sonlanmışsa kartta önceden belirlenmiş gizli anahtar (KEY_PROOF) kullanılarak işlem sınama değeri hesaplanır. Bu sınama değerinde aşağıdaki detaylar yer alır:

- Tag_Proof_Award
- İşlem Verisi: Sayaç veya kural tanımlayıcısı, terminal verisi, mevcut veri, işlem miktarı, kural yürütme durumu.
- Kart işlem sayacı (Komutun başında artırılır.)

Komuttan sonra, kartta ilgili komut için önceden tanımlanmış olan gizli anahtar kullanılarak MAC_OUT değeri hesaplanır. MAC_OUT'ta aşağıdaki alanlar bulunur:

- Tag_Mac_Award_Out
- CTC
- Komut Başlığı
- Terminale gönderilen mesaj

Üçüncü Adım:

Terminal, gelen MAC_OUT değerini doğrular. İşlem başarılı ise, işlem verisi ve sınamaya değeri operatöre iletilir.

Dördüncü Adım:

Operatör sınamaya değerini doğrular, başarılı ise işlem için onay verir.

İşlem Sınama Kuralları

İşlem sınama için aşağıdaki kurallara uyulması tavsiye edilmektedir:

- Terminal İşlem sınama değeri hesaplayamaz.
- İşlem sınama için kullanılacak gizli anahtarlar başka bir işte kullanılmamalıdır.
- Sayaç veya kural dosyasında belirtilen anahtar referansı 00h ise işlem sınama gerekli değildir. (Kart sıfır geri döndürür.)

Erişim Koşul Kontrolü

GemClub kartlarında, kullanıcı veya firmanın kart üzerinde belirli işlem yapıp yapamamasını belirlemek için erişim koşulları kullanılır. Bunlar sistemde dolaylı olarak tanımlanabilceği gibi (işletim sistemindeki bazı işlemlerde erişim koşulu kontrolü otomatik yapılır) ilgili bir dosya yaratılırken de verilebilir.

'ISO Kayıt Yönetim Komutları' için:

- İlgili dosya için PIN kodu ya da gizli kod içeren bir erişim koşulu tanımlanmış ise bu kodun 'ISO Kayıt' komutundan önce karta verilmesi gerekmektedir.
- Kod doğru olarak verilmişse 'Kod Doğrula' komutu ile enel güvenlik durumu güncellenir.
- Erişim koşulu olarak güvenli mesajlaşma seçilmişse, Mesaj doğrulama kodu (MAC) da 'ISO Kayıt' komutu ile birlikte verilmelidir.

Diğer fonksiyonlar için:

- APDU komutunun yapısı dosyada belirtilen erişim koşuluna bağlı değildir. MAC ya da gizli kod belirtilmezse doğrulama verisi olarak sıfır gönderilir.
- Erişim koşulu olarak gizli kod gerekiyorsa, bu değer komutun veri alanında tutulabilir.
- Genel güvenlik durumu bu tip kod bildirimi için güncellenmez.

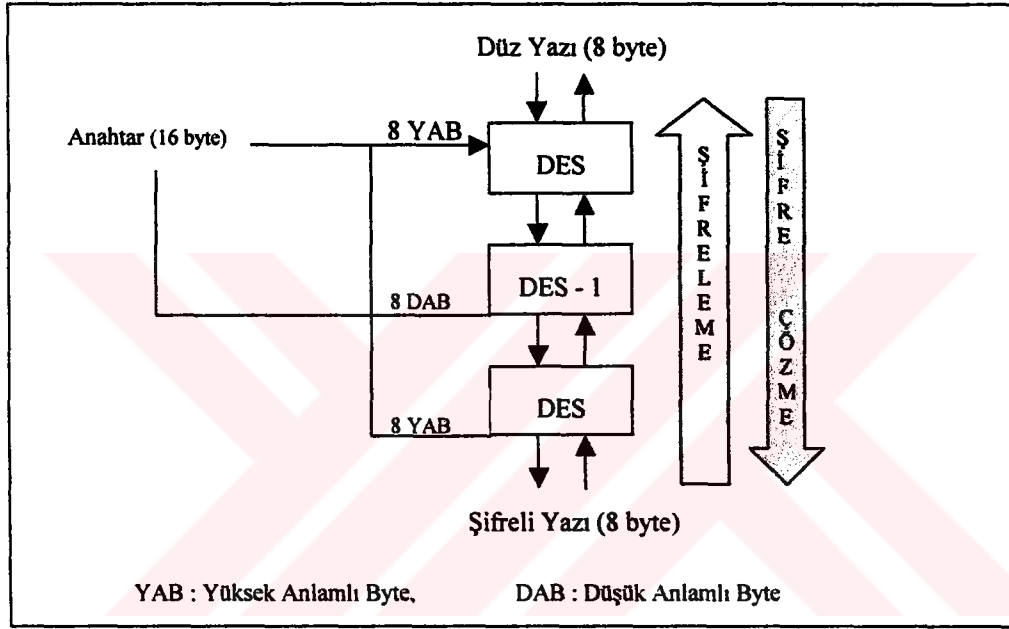
- Erişim koşulunda PIN kod gerekli gösteriliyorsa komuttan önce 'Kod Doğrula' komutu yürütülecektir. Bu işlem ile, PIN kodu için kartın genel güvenlik durumu güncellenecektir.

4. GemClub Şifreleme Yöntemleri

Bu bölümde, GemClub kartlarındaki güvenli mesajlaşma ve kullanılan komutlara ilişkin mesaj doğrulama kodları ele alınacaktır.

GemClub'da Güvenli Mesajlaşma – 3DES Algoritması

GemClub kartlarında güvenli mesajlaşma ve şifreleme için 3DES algoritması kullanılmaktadır. Çift uzunluklu anahtarlar kullanılarak yürütülebilen 3DES algoritması ile şifreleme veya şifre çözme Şekil A.9'da verilmiştir.



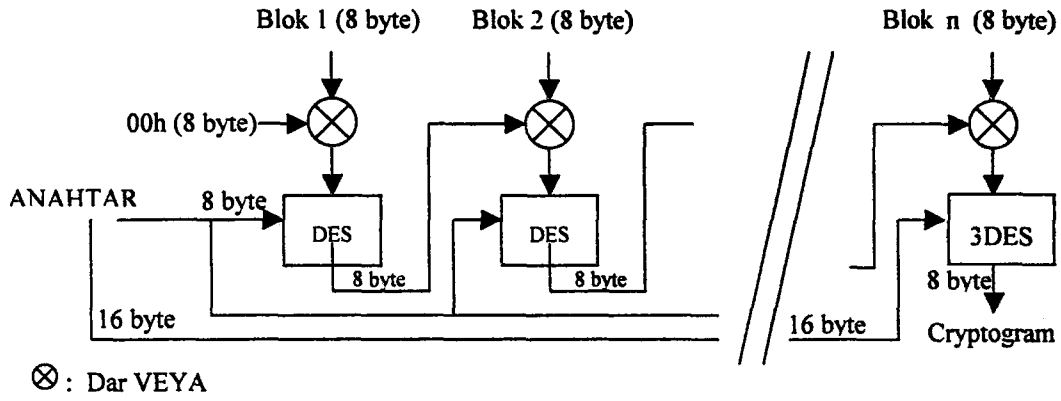
Şekil A.9 3DES Şifreleme / Şifre Çözme Algoritması

MAC Hesaplaması

GemClub kartta, aşağıdaki koşullara uyulması durumunda mesaj doğrulama kodu (MAC) CBC şifreleme algoritmasının son bloğunda hesaplanır.

- Ara bloklar üzerinde yüksek anlamlı 8 byte (pratikte, ilk yarım anahtar) kullanılarak DES uygulanır.
- Sonuncu blokta ise 16 byte'lık anahtar kullanılarak 3DES yürütülür.

MAC hesaplama algoritması Şekil A.10'da gösterilmektedir.



Şekil A.10 MAC Hesabı

GemClub'da Şifreleme ve Komutlar

GemClub kartlarda işlemler için birkaç MAC değeri kullanılır. Bu MAC veri alanlarının içeriği Tablo A.21 – Tablo A.24'te verilmiştir.

MAC hesaplamasında kullanılan her bir bloğun başında yer alan bir byte uzunluğundaki tekil sayıya 'Etiket' adı verilir.

Ödül İşlemleri

Tablo A.21 Ödül işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_AWARD – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Komut başlığı_{5 byte}) Blok 2 = (Etiket_{1 byte} Gelen Veri_{7 byte}) Blok 3 = (Etiket_{1 byte} Gelen Veri(devamı)_{7 byte}) Blok 4 = (Etiket_{1 byte} 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Sayaç Tanımlayıcısı_{1 byte} Terminal Verisi_{4 byte}) Blok 2 = (Etiket_{1 byte} Terminal Verisi (devamı)_{4 byte} Tarih₃)

	byte) Blok 3 = (Etiket ₁ byte İşlem Tutarı ₃ byte 80h 00 00 00)
--	--

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_AWARD – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₅ byte) Blok 2 = (Etiket ₁ byte Giden Veri ₇ byte) Blok 3 = (Etiket ₁ byte Giden Veri(devamı) ₆ byte 80h)

İndirim İşlemleri

Tablo A.22 İndirim işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_REDEEM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₅ byte) Blok 2 = (Etiket ₁ byte Gelen Veri ₇ byte) Blok 3 = (Etiket ₁ byte Gelen Veri(devamı) ₇ byte) Blok 4 = (Etiket ₁ byte 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Sayaç Tanımlayıcısı ₁ byte Terminal Verisi ₄ byte) Blok 2 = (Etiket ₁ byte Terminal Verisi (devamı) ₄ byte Tarih ₃ byte) Blok 3 = (Etiket ₁ byte İşlem Tutarı ₃ byte 80h 00 00 00)

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_REDEEM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket₁ byte CTC₂ byte Komut başlığı₁₅ byte) Blok 2 = (Etiket₁ byte Giden Veri₇ byte) Blok 3 = (Etiket₁ byte Giden Veri(devamı)₆ byte 80h)

Kural Kullanma İşlemleri

Tablo A.23 Kural Kullanma işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_USE_RULE – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket₁ byte CTC₂ byte Komut başlığı₁₅ byte) Blok 2 = (Etiket₁ byte Gelen Veri₇ byte) Blok 3 = (Etiket₁ byte Gelen Veri(devamı)₇ byte) Blok 4 = (Etiket₁ byte 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket₁ byte CTC₂ byte Sayaç Tanımlayıcısı₁ byte Terminal Verisi₄ byte) Blok 2 = (Etiket₁ byte Terminal Verisi (devamı)₄ byte Tarih₃ byte) Blok 3 = (Etiket₁ byte İşlem Tutarı₃ byte Kural Yürütme Durumu₁ byte 80h 00 00)

MAC_OUT

Ad	Kod
----	-----

KEY (16 byte)	KEY_ USE_RULE – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_ USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₁₅ byte) Blok 2 = (Etiket ₁ byte Giden Veri ₇ byte) Blok 3 = (Etiket ₁ byte Giden Veri(devamı) ₅ byte 80h 00)

Diğer Güvenli Mesajlaşma İşlemleri

GemClub'da MAC kullanan diğer işlemler şunlardır:

- Parametre Güncelle
- Nesne Yarat
- Nesne Yoket
- Kayıt Güncelle
- Kayıt Ekle

Tablo A.24 Diğer işlemlerde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_UPDATE_PARAM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_ UPDATE_PARAM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₁₅ byte) Blok 2 = (Etiket ₁ byte Gelen Veri _{değişken}) ... Blok n = (Etiket ₁ byte Gelen Veri(devamı) _{değişken} Doldurma)

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_UPDATE_PARAM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_ UPDATE_PARAM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₁₅ byte) Blok 2 = (Etiket ₁ byte Giden Veri ₂ byte 80h 00 00 00 00)

MAC Etiket Yönetim Kuralları

GemClub kartlarında Tablo A.25’de verilen önceden tanımlı etiketler kullanılmaktadır.

Tablo A.25 Önceden Tanımlanmış Etiket Değerleri

Komut	Etiket
TAG_MAC_IN_AWARD	11h
TAG_MAC_OUT_AWARD	12h
TAG_PROOF_AWARD	13h
TAG_MAC_IN_REDEEM	21h
TAG_MAC_OUT_REDEEM	22h
TAG_PROOF_REDEEM	23h
TAG_MAC_IN_USE_RULE	31h
TAG_MAC_OUT_USE_RULE	32h
TAG_PROOF_USE_RULE	33h
TAG_MAC_IN_UPDATE_PARAM	51h
TAG_MAC_OUT_UPDATE_PARAM	52h
TAG_MAC_IN_CREATE	61h
TAG_MAC_OUT_CREATE	62h
TAG_MAC_IN_DELETE	71h
TAG_MAC_OUT_DELETE	72h
TAG_MAC_IN_UPDATE_REC	81h
TAG_MAC_OUT_UPDATE_REC	82h
TAG_MAC_IN_APPEND_REC	91h
TAG_MAC_OUT_APPEND_REC	92h

Doldurma Kuralları

GemClub’da şifreleme yöntemi olarak 3DES algoritması kullanıldığından şifrelenecek veya şifresi çözülecek olan verilerin uzunluğu 8 byte’ın katı olmalıdır. Eğer gönderilen veya alınan verinin boyu bu şartı sağlamıyorsa kalan boşluklar ‘doldurulur’. Bu işlem aşağıda belirtilen kurallara göre gerçekleştirilir.

- Şifrelenecek veya şifresi çözülecek olan veri, 3DES algoritmasında belirtilen kurallara uymalıdır. Boyu 8 byte'ın katı olan bloklar halinde iletim yapılmalıdır.
- MAC hesaplaması sırasında, değeri 80h olan en az bir byte verinin sonuna eklenmelidir. Oluşan blok 8 byte'ın katı değilse sağdan sıfırlar ile doldurulmalıdır.

Örnekler:

- MAC uzunluğu 13 byte ise, sağdan 80 00 00h eklenerek toplam 16 byte'lık uzunluğa ulaşılır.
- MAC uzunluğu 15 byte ise, sağdan 80h eklenerek toplam 16 byte'lık uzunluğa ulaşılır.
- MAC uzunluğu 16 byte ise, sağdan 80 00 00 00 00 00 00h eklenerek toplam 24 byte'lık uzunluğa ulaşılır.

5. Diğer GemClub Özellikleri

Bu bölümde, GemClub kartlarındaki yedekleme yordamı ve nokta çevrim özellikleri ele alınacaktır.

GemClub Yedekleme Yordamı

Önemli verilerin iletim sırasında veya güç kesintisi sonrası değişmesi veya bozulmasını önlemek için 'Yedekleme Yordamı' özelliği kullanılır. Bu özellik sayesinde, hataya düşmüş bir işlemten sonra, işlem öncesi yedeklenen bilgilere geri dönülebilir. Bir komut bloke olduğunda ya da bir dosya yanlışlıkla bozulduğunda etkin hale geçen bu özellik, komut öncesi saklanan verilere geri dönülür.

Aşağıdaki komutların kullanımlarında yedekleme özelliği otomatik olarak etkinleşir:

- Nesne Yarat/Yoket
- Parametre Güncelle
- Ödül/İndirim
- Kural Kullan
- Sistem verilerinin güncellendiği tüm işlemler. (sınama değeri, dosya tanımlayıcısı, vb.)

Dosya güncellemelerinde, 'Yanıtı Al' komutu başarı ile yürütüldükten sonra dosya yedeklenmesi işlemi tamamlanmış olur.

Kural Mekanizması – Puan Dönüşümleri

GemClub kartlarındaki Kural Yordamı kullanılarak sayaçlara puan kazandırma veya sayaçlardan puan düşme işlemleri gerçekleştirilebilir. Belirli bir kuralın işlenmesi sırasında yürütülmek üzere en fazla dört makrokomut tanımlanabilir. Her bir kural farklı dört sayaç üzerinde tanımlanabilir.

Makrokomutlarda aşağıdaki bilgiler mutlaka bulunmalıdır:

- Kullanılacak sayacın tanımlayıcısı
- Yürütülecek komut tipi (Ödül veya İndirim)
- Parametre A
- Parametre B

'Puan Dönüşümlü Kural Kullan' komutu tanımlandığında, bu komutla ilgili bir 'Kural Kullan' komutu geldiğinde GemClub otomatik olarak, gelen verileri verilecek veya geri alınacak puanlara dönüştürür.

A ve B parametreleri, sayaç üzerinde tanımlanacak puan dönüşümünün belirtilmesinde kullanılırlar.

Parametre A : Sayaç değeri Parametre B'ye ulaştığında sayaçtan çıkarılacak veya sayaca eklenecek puan sayısını verir.

Parametre B : Sayaç değerinde değişiklik yapılması için ulaşılması gereken değeri belirtir. 'Kural Kullan' komutunda belirtilen işlem miktarı ile aynı birimden olmalıdır.

GemClub'da kullanılan puan dönüşüm fomülü şöyledir:

$$Puan = Parametre_A * \left\| \frac{Miktar}{Parametre_B} \right\| \quad (A.1)$$

Örneğin, 0Dh sayacına uçulan her kilometre için 50 puan verilmesi durumunda, A = 50 ve B = 1000'dir. Kartsahibi 4990 km uçmuş ise,

$$Puan = 50 * [4990 / 1000] = 200$$

puan kazanacaktır.

B parametresi sıfır olarak tanımlanırsa, miktar ne olursa olsun her işlemde A parametresi kadar puan verilir.

6. GemCLub Komut Formatı

Bu bölümde, GemClub kartlarında kullanılan Uygulama Protokol Veri Birimi (APDU - Application Protocol Data Unit) ele alınacaktır.

GemCLub'da, ISO 7816-4 standartında tanımlanan APDU formatında komut ve yanıtlar geçerlidir. Böylece, GemCLub komutlarının GemPlus GCR arayüz sürücü kütüphanesi ile uyumu sağlanmış olur. GemClub iletim (transport) katmanı, ISO 7816-3 standardında belirtilen T = 0 protokolü ile uyumludur. Burada APDU, iletim katmanı veri birimi 'ne (TPDU - Transport Protocol Data Unit) çevrilir. Okuyucu ile kart arasında TPDU bazında haberleşme yapılır.

GemClub kartlarda aşağıda belirtilen komutlar kabul edilir:

- Komut ya da yanıt yok: Sıfır uzunluklu bir T=0 ISO IN TPDU olarak gönderilir.
- Kısa Format: Komut yok, ama 1 ila 256 byte arası bir yanıt mevcut. Bu, T=0 ISO OUT TPDU olarak gönderilir.
- Kısa Format: 1 ila 255 byte arası uzunlukta komut verisi; yanıt yok. T=0 ISO OUT TPDU olarak gönderilir.
- Kısa Format: 1 ila 255 byte arası uzunlukta komut verisi; 1 ila 256 byte arası bir yanıt mevcut. Komut, T=0 ISO IN TPDU olarak gönderilir. Ardından T=0 ISO OUT TPDU olarak gönderilen bir 'Yanıt Al' komutu yollanmalıdır. Yanıt alma mekanizması ISO 7816-4 standardı ile uyumludur.

Ek. B Veritabanı Yapısı

Tablo Adı : TBL_BINA

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
BinaNo	int	4	NOT NULL
Ad	char	50	NULL
KampusNo	int	4	NULL

Tablo Adı : TBL_BOLUM

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
BolumNo	int	4	NOT NULL
Ad	char	50	NULL
FakulteNo	int	4	NULL
BaskanSicilNo	numeric	9	NULL
Bolum	bit	1	NOT NULL

Tablo Adı : TBL_DERS

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
DersNo	int	4	NOT NULL
Ad	char	100	NULL
BolumNo	int	4	NULL
DersTipNo	tinyint	1	NULL
OgrGrvSicilNo	numeric	9	NULL
Donem	tinyint	4	NOT NULL
Saat	tinyint	1	NULL
Kredi	tinyint	1	NULL
Aciklama	varchar	500	NULL
ArrSicilNo	numeric	9	NULL

Not: DersTipNo alanının son hanesi “0” ise seçmeli ders, “1” ise zorunlu ders anlamına gelmektedir.

Tablo Adı : TBL_DERSTIPI

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
TipNo	tinyint	1	NOT NULL
Aciklama	char	30	NULL

Tablo Adı : TBL_DERSPRG

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	numeric	9	NOT NULL
Yil	char	4	NULL
YariYil	char	1	NULL
DersNo	int	4	NOT NULL
Gun	tinyint	1	NULL

Seans	tinyint	1	NULL
BinaNo	int	4	NULL
SinifNo	int	4	NULL

Tablo Adı : TBL_FAKULTE

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
FakulteNo	int	4	NOT NULL
Ad	char	50	NULL
BinaNo	int	4	NULL
DekanSicilNo	numeric	9	NULL
Fakulte	bit	1	NOT NULL

Tablo Adı : TBL_GUN

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	tinyint	1	NOT NULL
Aciklama	char	10	NULL

Tablo Adı : TBL_HATA

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	int	4	NOT NULL
Aciklama	varchar	200	NULL

Tablo Adı : TBL_iSLEM

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	tinyint	1	NULL
Aciklama	varchar	100	NULL

Tablo Adı : TBL_KAMPUS

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KampusNo	int	4	NOT NULL
Ad	char	30	NULL
Semt	char	20	NULL
il	char	20	NULL

Tablo Adı : TBL_KULLANICI

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KullaniciNo	numeric	9	NOT NULL
UnvanNo	char	10	NULL
Ad	char	30	NULL
Soyad	char	30	NULL
GirisYili	char	4	NULL
ABDaliNo	int	4	NULL
BolumNo	int	4	NULL
Adres	varchar	300	NULL
TelefonNo	char	10	NULL

EMail	char	50	NULL
Resim	char	20	NULL
imza	char	20	NULL
Cinsiyet	char	1	NULL
Uyruk	char	3	NULL
NufusSeriNo	char	15	NULL
BabaAdi	char	30	NULL
AnneAdi	char	30	NULL
DogumTarihi	char	8	NULL
DogumYeri	char	20	NULL
MedeniHali	char	1	NULL
Dini	char	10	NULL
il	char	20	NULL
ilce	char	20	NULL
Mahalle	char	20	NULL
CiltNo	char	10	NULL
AileSiraNo	tinyint	1	NULL
SiraNo	tinyint	1	NULL
DahiliNo	char	4	NULL
OdaNo	int	4	NULL
Sifre	char	8	NULL
KurumNo	it	4	NULL

Not: KurumNo alanının 0 olması kullanıcının özel kişi, 0 dan farklı bir değer alması ise tüzel kişi olduğunu gösterir.

Tablo Adı : TBL_KURUM

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KurumNo	int	4	NOT NULL
KurumAdi	char	100	NULL
KampusNo	int	4	NULL

Tablo Adı : TBL_OGRDERS

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KayitNo	numeric	9	NOT NULL
OgrenciNo	numeric	9	NULL
Yil	char	4	NULL
YariYil	char	1	NULL
DersNo	int	4	NULL
Vize1	tinyint	1	NULL
Vize2	tinyint	1	NULL
Final	tinyint	1	NULL
Odev1	tinyint	1	NULL
Odev2	tinyint	1	NULL
Odev3	tinyint	1	NULL
Odev4	tinyint	1	NULL
Quiz1	tinyint	1	NULL
Quiz2	tinyint	1	NULL
SOrtalama	tinyint	1	NULL

KOrtalama	char	2	NULL
KatilimSaat	tinyint	1	NULL
Hak	tinyint	1	NULL
Durum	tinyint	1	NULL

Tablo Adı : TBL_ODEME

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	numeric	9	NOT NULL
OgrenciNo	numeric	9	NULL
KurumNo	int	4	NULL
Tarih	char	8	NULL
Saat	char	5	NULL
Tutar	numeric	9	NULL
Aciklama	varchar	250	NULL

Tablo Adı : TBL_ONKOSUL

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
DersNo	int	4	NOT NULL
KosulDersNo	int	4	NOT NULL

Tablo Adı : TBL_PARAM

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
DersKayitBasTar	char	8	NULL
DersKayitBitTar	char	8	NULL
BaslangicTarihi	char	8	NULL
BitisTarihi	char	8	NULL
FinalBasTar	char	8	NULL
FinalBitTar	char	8	NULL
YariYil	char	1	NULL
Yil	char	4	NULL

Tablo Adı : TBL_RANDEVU

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
RandevuNo	numeric	9	NOT NULL
OgrenciNo	numeric	9	NULL
SinifNo	int	4	NULL
Tarih	char	8	NULL
Seans	tinyint	1	NULL

Tablo Adı : TBL_SEANS

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
KayitNo	tinyint	1	NOT NULL
Aciklama	char	20	NULL

Tablo Adı : TBL_SINIF

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
SinifNo	int	4	NOT NULL
Ad	char	50	NULL
BinaNo	int	4	NULL
Kapasite	tinyint	1	NULL
SinifTipNo	tinyint	1	NULL

Tablo Adı : TBL_SINIFTIPI

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
TipNo	tinyint	1	NOT NULL
Aciklama	char	30	NULL

Tablo Adı : TBL_UNVAN

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
UnvanNo	tinyint	1	NOT NULL
Aciklama	char	50	NULL
Kisaltma	char	10	NULL

Tablo Adı : TBL_YETKI

<u>Alan Adı</u>	<u>Veri Tipi</u>	<u>Uzunluk</u>	<u>NULL Olabilir mi?</u>
UnvanNo	tinyint	1	NULL
islemNo	tinyint	1	NULL

Sabit Değerler:

Tablo Adı : TBL_GUN

<u>KayıtNo</u>	<u>Aciklama</u>
1	Pzt.
2	Sal.
3	Çar.
4	Per.
5	Cum.
6	Cts.
7	Paz.

Tablo Adı : TBL_SEANS

<u>KayıtNo</u>	<u>Aciklama</u>
1	09:00-09:50
2	10:00-10:50
3	11:00-11:50
4	12:00-12:50
5	13:00-13:50

6	14:00-14:50
7	15:00-15:50
8	16:00-16:50
9	17:00-17:50

Tablo Adı : TBL_UNVAN

<u>UnvanNo</u>	<u>Aciklama</u>	<u>Kisaltma</u>
0	Diğer	Diğer
1	Kurum	Kurum
2	Lisans	Ls.Ogr.
3	Yüksek Lisans	Ms.Ogr.
4	Doktora	Dr.Ogr.
5	Araştırma Görevlisi	Arş.Grv.
6	Doktor	Dr.
7	Yardımcı Doçent	Yrd.Doç.Dr.
8	Doçent	Doç.Dr.
9	Profesör	Prof.

Tablo Adı : TBL_SINIFTIPI

<u>KayıtNo</u>	<u>Aciklama</u>
0	Oda
1	Sınıf
2	Lab