



T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMİ GÜVENLİĞİNDE
YAZMA RİTMİ TABANLI BİR ALGORİTMA

Levent YALÇIN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ
TEZ DANIŞMANI

Yrd.Doç.Dr. İbrahim SOĞUKPINAR

GEBZE

2002

121297

Levent YALÇIN'ın tez çalışması G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulunun 05.11.2021... tarih ve 2021/32... sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

ÜYE : Yrd.Doç.Dr. İbrahim SOĞUKPINAR ,GYTE

(Tez Danışmanı)

121297

ÜYE : Yrd.Doç.Dr. Mehmet GÖKTÜRK ,GYTE

ÜYE : Yrd.Doç.Dr. Hakan HOCAOĞLU ,GYTE

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 05.02.2022... tarih ve 2022/5... sayılı kararı.



ÖZET



Bilgi güvenliğinin saėlanmasındaki temel ama, sistemdeki bilgilere sadece yetkili kullanıcıların erişimini saėlamak ve yetkisi olmayan kullanıcıların sisteme girişini engelleyerek bilgilerin korunmasıdır. Bu amaca erişim için ise yetkili ve yetkisiz kişilerin ayırt edilmesi gerektiğı aşıkardır.

Bu tez çalışmasında yukarıdaki ama doğrultusunda kişilerin klavyeye yazma ritimlerinden ayırt edilebileceğı hipoteziyle, şifre ele geçirilse veya bilinse bile yetkisiz kişilerin sisteme girişini çok zorlaştıran bir güvenlik sistemi tasarlanmıştır.

Bu ama doğrultusunda geliştirilen programda biyometrik sistemlerde kullanılan temel sınıflama algoritmaları paralelinde ancak mevcut algoritmaların karışımından oluşan yeni bir algoritma uygulanmış ve çok tatmin edici sonuçlar elde edilmiştir. Bu çalışmayla ülkemizde üzerinde yeterince çalışma yapılmayan bu konuya farklı bir boyut getirilerek konu üzerinde çalışacak olan kişilere rehber olmaya çalışılmıştır.

Bu tez çalışmasında ayrıca diğer biyometrik sistemlere de değinilerek ve karşılaştırmalar yapılarak yazma ritmi yöntemiyle kullanıcı tanıma sistemlerinin daha iyi anlaşılmasına gayret edilmiştir.



SUMMARY

The primary goal in providing the security of the information systems is to protect the information by providing the authorized users to access to information in system and preventing unauthorized users not to access and log on systems. It is obvious that it is necessary to distinguish the authorized and unauthorized users to achieve this goal.

According to the above goal in this thesis, a security system that prevents the unauthorized users to access the system even if the password is known or stolen, was designed by the hipotez of typing rhytm of every person is different from the others.

In the implemented program to achive this goal, a new classification algorithm which is parellel to the basic classification algorithms in pattern recognition was implemented and achivied very impressive results. With this study it was tried to be guide to the others that will study about this subject which is not studied about very much in our country.

In this thesis study keystroke dynamics was tried to be understood better by mentioning and comparing the other biometric systems.

TEŞEKKÜR



Bu tezin hazırlanmasında bana her safhada yardımlarını hiçbir zaman esirgemeyen değerli hocam Yrd.Doç.Dr. İbrahim SOGUKPINAR başta olmak üzere kendilerine tez çalışmam süresince çok az zaman ayırabildiğim ancak buna rağmen her zaman desteklerini gördüğüm değerli Eşim Nurcan YALÇIN ve kızım Hüma YALÇIN ile burada adını sayamadağım daha birçok arkadaşına ve geçirdiği bir kaza sonucu şu anda aramızda olmayan Alper KÜÇÜKÖZ'e katkılarından dolayı teşekkürlerimi sunmayı bir borç bilirim.



İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
İÇİNDEKİLER DİZİNİ	vii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	xi
1. GİRİŞ	1
2. BİYOMETRİK GÜVENLİK SİSTEMLERİ	3
2.1 Parmak İzi Tanıma	5
2.1.1 Parmak İzi Okuyucuları	5
2.1.2 Parmak İzi Tanıma İşlemi	6
2.2 El Yapısını Tanıma	6
2.3 İris Paterni Tanıma	6
2.4 Retina Paterni Tanıma	7
2.5 Yüz Tanıma	8
2.6 Ses Tanıma	8
2.7 İmza Tanıma	9
2.8 Klavyeye Yazma Ritmi Tanıma	9
2.9 Biyometrik Sistemlerde Kullanılan Ana Sınıflama Algoritmaları ve Önceki Bazı Çalışmaların Matematiksel Modelleri	10
2.9.1 Minimum-Fark Sınıflandırması	11
2.9.2 Mahalanobis Sınıflandırması	12
2.9.3 K En Yakın Komşu (KNN) Sınıflandırması	15
2.9.4 Önceki Bazı Çalışmaların Matematiksel Modelleri	16
2.10. Klavyeye Yazma Ritmine Göre Kullanıcı Tanıma Sistemlerinde Bugüne Kadar Yapılan Çalışmalar	21
3. BİLİŞİM SİSTEMİ GÜVENLİĞİNDE YAZMA RİTMİ TABANLI SİSTEM TASARIMI	27
3.1 Biyolojik Yazma Ritmi Analiz Etme Yazılımı Algoritması ve Matematiksel Modeli	29

3.2 Biyolojik Yazma Ritmi Analiz Etme Yazılımı Tanıtımı	36
4. UYGULAMA SONUÇLARI VE YORUMLAMA	49
4.1. Diğer Yöntemler ile Karşılaştırma	53
5. SONUÇ	56
KAYNAKLAR	58
EKLER	61
ÖZGEÇMİŞ	133



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklama
FAR	False Acceptance Rate (Yanlış Kabul Edilme Oranı)
FRR	False Rejection Rate (Yanlış Reddedilme Oranı)
ERR	Equal Error Rate (Eşit Hata Oranı)
HMM	Hidden Markov Model
KNN	K Nearest Neighbour
PIN	Personal Identification Number
SSH	Secure Shell
VPN	Virtual Private Network

ŞEKİLLER DİZİNİ



<u>Sekil</u>	<u>Sayfa</u>
2.1	Klasik İz Tanıma Modeli 10
2.2	Fark Faktörleri 11
2.3	Minimum Fark Sınıflandırma Sistemi 12
2.4	X Değerlerine Ait Dağılım 13
2.5	Kovaryans Değerleri 13
2.6	Kovaryans Matrisi 14
2.7	Mahalanobis Sınıflandırması 14
2.8	K En Yakın Komşu Sınıflandırması Örnek Dağılımı 15
3.1	Çarpık Dağılım Tipleri 30
3.2	Basık Dağılım Tipleri 30
3.3	Öğrenme Safhasındaki Karakter Çiftleri Arası Basış Zamanları 32
3.4	Programın Ana Giriş Formu 36
3.5	Sistem Yöneticisi Tanıtımı Formu 37
3.6	Sistem Yöneticisi Tanıtıldıktan Sonraki Ana Formun Görünüşü 37
3.7	Sistem Yöneticisi Giriş Formu 38
3.8	Sistem Yöneticisine Sistem Ayarları Öncesi Verilen Mesaj 39
3.9	Sistem Ayarları Formu 39
3.10	Sistem Ayarlarının Değiştirilemeyeceğine İlişkin Mesaj 40
3.11	Eşit Hata Oranı 42
3.12	Kullanıcıların Kendilerini Sisteme Tanıtma Formu 43
3.13	Delete veya Backspace Tuşlarına Basıldığı Mesajı 44
3.14	Yeterli Sayıda Karakter Girilmediği Mesajı 45
3.15	Hatalı Veri Girişi Mesajı 45
3.16	Sisteme Kaydedilme Mesajı 46
3.17	Eski Kullanıcı Sisteme Giriş Formu 47
3.18	Sistemde Aynı İsimle Başka Bir Kullanıcı Olduğu Mesajı 48
3.19	Sistemde Böyle Bir Kullanıcı Olmadığı Mesajı 48



TABLÖLAR DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
1.1. Biometric Tabanlı Doğrulama Yöntemlerinin Karşılaştırması	3
1.2. Biyometrik Data Büyüklükleri (Birim Kayıt İçin)	3
3.1. Bir Kullanıcının Klavyeye Veri Girişi Sonundaki Ortaya Çıkan Zamanlayıcı (Timer) Değerleri	31
4.1. Kullanıcıların Bir Kullanıcı Adına Veri Girişi Denemeleri İstatistiği	51
4.2. Diğer Sınıflandırma Algoritmalarıyla Karşılaştırma	53





1. GİRİŞ

Bilgisayar sistemlerine erişimde kullanılan günümüzde üç temel doğrulama ve gerçekleştirme yöntemi vardır. Bu yöntemler sırasıyla sahiplik tabanlı (ownership-based), bilgi tabanlı (knowledge-based) ve biyometrik tabanlı (biometric-based) doğrulama yöntemleridir. Sahiplik tabanlı doğrulama yöntemine smart kart olarak adlandırılan üzerlerinde entegre taşıyan akıllı kart sistemleri v.b., bilgi tabanlı doğrulama yöntemlerine şifre veya PIN gibi bilgilerin bilinmesine dayalı doğrulama yöntemleri, biyometrik tabanlı doğrulama yöntemlerine ise Retina, İris, Parmak İzi, Avuç İçi tanıma gibi statik/fiziksel biyometrik sistemleri ile İmza, Ses, Yazma Ritmi Tanıma gibi dinamik/davranışsal biyometrik sistemleri örnek olarak verilebilir.

Sahiplik tabanlı doğrulama yöntemleri adından anlaşılacağı üzere sisteme erişim için kendilerine verilen ve korumaları fiziksel olarak kişilere bağlı olan doğrulama yöntemlerini, bilgi tabanlı doğrulama yöntemleri ise günümüzde en yaygın olarak kullanılan doğrulama sistemi olmakla birlikte aynı zamanda kullanıcı doğrulamada kullanılan çok zayıf bir mekanizmayı ifade eder. Çünkü herhangi bir şekilde şifre veya PIN bilgisini ele geçiren her kim olursa olsun sisteme giriş için herhangi bir engel kalmamaktadır. Biyometrik tabanlı doğrulama yöntemlerinde ise fiziksel veya davranışsal sistemler olurlarsa olsun sisteme erişim bilgisi herhangi bir şekilde ele geçirildiği takdirde bile sisteme giriş diğer sistemlere nazaran çok daha zordur. Çünkü sisteme giriş için gerekenler kişilerin kendine mahsus özellikleridir.

Bir kript/şifre sisteminin güvenliğini belirlemede üç ana faktör vardır. Bunlar şifreleme algoritmasının kompleksliği, şifre, PIN v.b. tanıma karakterinin uzunluğu ve tanıma karakterinin güvenli muhafazasıdır.

Şifreleme algoritmasının kompleksliği çok önemlidir çünkü şifrenin çözülme prosesinin kolaylığıyla direkt alakalıdır. Ancak birçok kişinin çözülmesi en kolay faktörün bu faktör olabileceğini düşünmesine rağmen günümüzde kript sistemleri son derece iyi bir şekilde oluşturulduğundan gerçekte saldırıya an az açık olan faktördür.

Tanıma karakteri/şifre uzunluğu ise şifreleme algoritmasına göre saldırıya daha açık bir faktördür. Çünkü örneğin 3 karakterlik bir şifrenin 10 karakterlik bir şifreye göre çözülme olasılığı, 10'un muhtemel permutasyon sayısı 3'e göre çok daha fazla

olduğundan, çok daha azdır. Bu nedenle bu tez çalışmasında da şifre uzunluğu en az 6 karakter olarak belirlenmiştir.

Tanıma karakteri/şifrenin güvenli bir şekilde saklanması veya korunması ise gerçekte saldırıya en açık olan faktördür. Çünkü tanıma karakteri/şifre kaybedilebilir veya çalınabilir. Diğer yandan gerçek hayatta şifreler genellikle çok kolay tahmin edilebilir kombinasyonlarda seçilir ve genellikle kişiler arasında paylaşılır ve zorunluluk olmadıkça değiştirilmek istenmezler. Diğer yandan şifrenin uzunluğu veya zor ezberde tutulmasından dolayı şifre kağıtta, masa üstünde klavye altında veya bilgisayar kasası/monitöre yapıştırılmış olarak bulunabilir veya diskette saklanabilir ve yetkisiz kullanıcıların erişimine çok açıktır.

Günümüzde şifrelemede kullanılan iki temel sistem vardır. Bunlar tek anahtar ve iki anahtar sistemleridir. Tek anahtar sistemlerinde alıcı ve verici tarafında verinin şifrelenmesinde veya deşifre edilmesinde tek anahtar kullanılır.

Günümüzde özellikle internet üzerinden yapılan elektronik ticarete yaygın olarak iki anahtar sistemleri kullanılmaktadır. İki anahtar sistemlerinde datanın şifrelenmesi için bir anahtar (public key), datanın deşifre edilmesi için ise ayrı bir anahtar (private key) kullanılmaktadır. İki anahtar sistemleri tek anahtar sistemlerine nazaran daha zor çözülebilir bir sistem olmakla birlikte alıcının sertifikasyonu ve sertifikasyon hiyerarşisi nedeniyle tek anahtara göre daha yavaş işlemektedir.

Biyometrik sistemler ise herkesin kendisine has bir özellik içermesi ve bu özelliklerin normal şartlar altında kaybolmaz ve çalınmaz olması özelliği ve gerektirdiği ekstra donanım maliyetleri düştüğü takdirde özellikle elektronik ticarete yakın gelecekte yaygın kullanılacak bir sistem olmaya aday görünmektedir.

2. BİYOMETRİK GÜVENLİK SİSTEMLERİ

Biyometrik sistemlerin yakın gelecekte yaygın olarak kullanılması beklenmekle birlikte mükemmel sistemler değildir, çünkü iki tip hatayı ihtiva ederler. Bunlar I. Tip hata yani gerçek veya yetkili kullanıcının sisteme kabul edilmeme olasılığı/oranı (False Rejection Rate (FRR)) ve II. Tip hata yani yetkisiz kullanıcının sisteme kabul edilme olasılığı/oranı (False Acceptance Rate (FAR))'dır.

Doğal olarak her iki hatanın da mümkün olabildiğince düşük olması arzulanan bir durumdur. Ancak yukarıda belirtilen iki hatanın birbiriyle ters orantılı olduğunu dolayısıyla kullanıcı doğrulama safhasında kullanılan eşik seviyesinin (threshold) değiştirilmesine göre bir hata artarken diğerinin azaldığını ifade etmek gerekir. Tablo 1.1'de biyometrik sistemlerdeki kullanıcı doğrulama metotlarının hatalar, maliyetler v.s. yönünden karşılaştırılmaları verilmiştir. [1]

Tablo 1.1. Biyometrik Tabanlı Doğrulama Yöntemlerinin Karşılaştırması

Doğrulama Yöntemi	FAR (%)	FRR (%)	Doğrulama Zamanı(Sn)	Maliyet (\$)	Güvenlik
Parmak İzi	0.001	0.5	4	4000	Yüksek
El Geometrisi	1	1 - 3	6	3500	Orta
Retina Tanıma	0.0001	5	2	6000	Yüksek
Ses Tanıma	0.1 - 2	0.25 - 5	3	1000 - 1500	Düşük
İmza Tanıma	3	0.7	5	700 - 1300	Düşük

Tablo 1.1'de dikkat edileceği üzere en çok tanıma oranına sahip dolayısıyla en az hataya sahip ancak buna karşılık en pahalı sistem retina tanıma sistemidir.

Biyometrik sistemlerin seçiminde dikkate alınması gereken diğer bir özelliğe ilgili biyometrik özelliğin veri tabanında kaplayacağı alanın ne olduğudur. Tablo 1.2'de her bir farklı kayıt için farklı biyometrik yöntemlerin kapladığı alanlar verilmektedir. [2]

Tablo 1.2. Biyometrik Data Büyüklükleri (Birim Kayıt İçin)

Biyometrik Yöntem	Data Büyüklükleri (Bytes)
Retina Tanıma	35
İris Tanıma	256
Parmak İzi Tanıma	512 - 1000
El Geometrisi Tanıma	9

Tablo 1.2’de dikkat edileceği üzere ise el geometrisi ve retina tanıma sistemleri diğer biyometrik yöntemlerden veri tabanında daha az yer kaplamaktadır. Verinin fazla yere ihtiyaç duyması ilave donanım ihtiyacı ve ağ trafiğinin gereksiz yere fazla yüklenmesi gibi dezavantajları da beraberinde getirecektir. Dolayısıyla biyometrik sistemleri kurmayı planlayanların dikkate alması gereken önemli bir özellik de veri büyüklüğü olmaktadır.

Biyometrik sistemlerin çalışma şekli temelde aynıdır. Çünkü tüm biyometrik sistemlerde öncelikle kullanıcının sisteme tanıtılması gerekir. Bunun için önce sistem tarafından kullanıcıyı tanımak ve onun karakteristik özelliklerini çıkarmak maksadıyla, kullanılan biyometrik yöntemle göre, kullanıcıdan belirli bir sayıda veri girişi yapılması istenir. Bu safhada yani kullanıcıları öğrenme safhasında sistem tarafından kullanıcıların biyometrik özellik vektörleri çıkartılır.

Bu safhadan sonra sisteme giriş yapmak isteyen veya sistemdeki kullanıcıların sistemde tanımlı olup olmadıklarına bakılır yani kimlik tespiti yapılır (Identification).

Daha sonra sistemde tanımlı ise sisteme giriş yapmak isteyen veya sistemdeki kullanıcının sistemde tanımlı gerçek kullanıcı olup olmadığının doğrulaması yapılır. Doğrulama/Gerçekleme (Verification) safhasında sisteme girişteki özellik vektörü sistemde tanımlı kullanıcının özellik vektörü ile karşılaştırılır.

Doğrulama/Gerçekleme safhası sonundaki karar verme/güncelleme safhasında ise kullanılan pattern/iz tanıma sınıflandırma algoritmasına göre iki vektör arasındaki fark belli bir eşik seviyesinin altında ise sistem girişine müsaade edilir değil ise müsaade verilmez. Ayrıca sisteme giriş yapabilen girişlere göre öğrenme safhasında tespit edilen özellik vektörleri güncellenir.

Bilgi güvenliğini sağlamada kullanılan birçok biyometrik güvenlik sistemi bulunmakla birlikte kararlılığı test ve kabul edilmiş ve ticari olarak uygulamaya geçirilmiş temel bazı biyometrik sistemleri, yazma ritmi tanıma sisteminin daha iyi anlaşılabilmesi ve karşılaştırmalar yapabilmek maksadıyla aşağıda kısaca açıklanmıştır. [3]



2.1 Parmak İzi Tanıma

Parmak izi tanıma biometrik tanıma tekniklerinden en eskisidir denebilir. Eski Çin’de bir dokümanın kim tarafından yazıldığını belirlemek maksadıyla kullanılmaya başlanmıştır.

Yakın geçmişte, 1960’lardan bu yana parmak izi tanıma teknolojisi hukuki olaylarda kullanılmaya başlanmış ve günümüze kadar geliştirilmiştir. Parmak izi tanıma teknolojisi üzerinde yapılan çalışmalar en iyi tanıma teknolojisini geliştirmek için büyük bir potansiyel oluşturmuştur. [3]

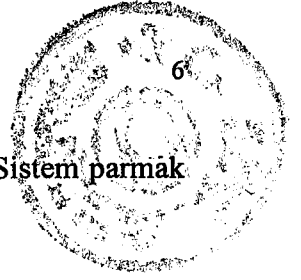
2.1.1 Parmak İzi Okuyucuları

Parmak izi tanıma teknolojisinde daha ileriye gidebilmek için parmak izinin dijital ortama geçirmek gerekmektedir. Önceleri kullanılan geleneksel yöntem; parmak izini mürekkep kullanmak suretiyle bir kağıt üzerine alıp, onu scanner ile dijital ortama geçirmektir. Öte yandan günümüzde modern parmak izi okuyucuları kullanılmaktadır. Bunlar mürekkep gerektirmez, modern parmak izi okuyucularından en sık kullanılanları şu şekilde sıralayabiliriz.

- (1) Optik
- (2) Termal
- (3) Silikon
- (4) Ultrasonik prensip ile çalışanlar.

Optik parmak izi okuyucuları en sık kullanılanlardır. Okuyucuya temas eden parmaktaki çizgileri yansıtan ışığın yansımadaki değişimleri algılama mantığı ile çalışır. Genel olarak 10 x 10 x 5 santimetre boyutlarında olup, daha küçültmenin imkanı bulunmamaktadır.

Silikon teknolojisi optik okuyucu tekniğinden daha eski bir teknolojidir. Parmağın bir nevi direnci olan elektronik kapasitesini algılaması üzerine çalışır.



Ultrasonik parmak izi tarayıcılar en yeni ve az yaygın olanıdır. Sistem parmak yüzeyinin tanınmasında ultrasonik teknolojisini kullanmaktadır.

2.1.2 Parmak İzi Tanıma İşlemi

Parmak izi tanıma teknolojisi “Önemsiz ayrıntılar” ve “Korelasyon” olmak üzere iki kategoride incelenir. Önemsiz ayrıntılar teknolojisi, öncelikle parmak üzerindeki önemsiz ayrıntıları bulur, daha sonra bunların parmaktaki birbiri ile olan ilişkisinin haritasını çıkartır. Bu yöntemde sık karşılaşılan problem, parmak izi kalitesinin düşük olduğu durumlarda önemsiz ayrıntıların açıklanamamasıdır.

Korelasyon metodu üzerine tasarlanmış teknik ise önemsiz ayrıntılar metodunda karşılaşılan bir takım güçlüklerin üstesinden gelir. İncelenen noktaların kesin yerlerinin belirlenmesi ve karşılaştırılması mantığına dayanır.

2.2 El Yapısını Tanıma

El yapısını tanıma tekniği; her insanın el yapısının farklı yapıda olduğu ve bunun belli bir yaşa kadar değişmeyeceği mantığı üzerine çalışır. El yapısı tanıma sistemi parmakların uzunluk ve genişlik ölçülerinin alınmasına dayalı bir mantıkla çalışır. Bu maksatla bir çok yöntem vardır. Bunlardan en yaygın olanı mekanik ve optik prensiple çalışandır.

El yapısını tanıma tekniği diğer biometrik sistemler ile karşılaştırıldığında çok fazla bilgi üretmez. Buna bağlı olarak bir eli bir diğerinden ayırt edemeyebilir. Sonuçlar göstermiştir ki, el yapısı tarama sistemleri sadece alçak güvenlik isteyen bölgeler için uygundur. [3]

2.3 İris Paterni Tanıma

İris, gözün içindeki çemberin göz bebeği haricinde kalan renkli kısmıdır. Bir insanın her iki gözünün bile farklı iris paterni mevcuttur.

İris paterni 10-40 santimetre uzaklıktan Siyah-Beyazkamera ile alınır. Kamera bir aynanın arkasında saklanır. Hareketsiz olan gözün görüntüsü alınır, bu esnada özel bir aydınlatmaya ihtiyaç duyulmaz.



İris taramanın herhangi bir zorluğu olmadığından bir çok kullanıcı tarafından kabul edilir. İris paterni bazı hastalıkların geçirilmesi dışında değişmez.

Siyah-Beyaz görüntüsü elde edildikten sonra, sistem yazılımı görüntüyü lokalize etmeye çalışır. Yazılım irisi kaplayan net eğrileri yaratır. Bu eğrilerin üzerindeki noktaların karaltılarından iris krateri ortaya çıkartılır. Karar verme aşamasında karşılaştırma yöntemi yapılır ki, bu oldukça hızlı gerçekleştirilir. Modern bilgisayarlar saniyede 4.000.000 iris kodunu karşılaştırabilecek kapasitededir. Buna bağlı olarak şunu söyleyebiliriz ki, iris tanıma teknolojisi biometrik sistemler içerisinde en hızlı olanıdır. Ayrıca güvenilirlik oranı oldukça yüksektir. İki iris kodunun aynı olma olasılığı 10^{-52} oranına eşittir. [3]

2.4 Retina Paterni Tanıma

Retina tanıma, gözdeki retinanın üzerinde bulunan kan damarlarının haritasının çıkartılması mantığına dayanır. Retina tarama teknolojisi iris taramadan daha eski bir teknolojidir.

Retina doğrudan görülemez ve kızıl ötesi bir ışık kaynağı retinayı aydınlatmak için gereklidir. Retinayı kaplayan kan damarlarınca kızıl ötesi enerji hızla tutulur.

Retinadaki kan damarlarının görüntüsü karakteristik noktaları çıkarılmak üzere analiz edilir. Retina irise oranla hastalıklara daha dirençli olduğundan karakteristikleri kolayca değişmez.

Retinal tarama kullanışlı olmadığı ve pahalı olduğu için çok yaygın değildir. Retina tarama tekniğinin yüksek güvenlik isteyen yerlerde kullanılması daha uygundur. [3]

2.5 Yüz Tanıma

Yüz tanıma biometrik tanıma tekniklerinden en doğalıdır. Birinin bir diğlerinden ayırt edilmesi her insanın görüntüsünün elde edilmesi ile mümkündür.

Herhangi bir kamera yüz görüntüsünün elde edilmesi maksadıyla kullanılabilir.

Yüz tanıma birinci basamak elde edilen yüz görüntüsünü yerleştirmektedir. Daha sonra yüz karakteristiğini çıkarmaktır. Yüz tanıma tekniğı iki farklı alanda geliştirilmektedir; yüz metriğı ve yüz uzvu şekli.

Yüz metriğı teknolojisi yüzdeki burun, göz, ağız gibi unsurların birbirine olan uzaklığının ölçülmesi mantığına dayanır. Diğ bir metot olan yüz uzvu son üç yılda geliştirilmiş olan bir yöntemdir. Bu yöntem daha önce belirlenmiş 150 yüz uzvu şeklinin yüze yerleştirilme açısının kategorize edilmesi mantığına dayanır. Bu yöntem polislerin robot resim çıkarmada kullandıkları yöntemle benzeşir.

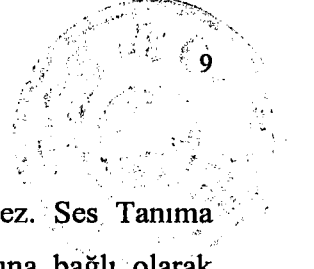
Bu yöntemde, benzer insanları veya ikizleri ayırt etmede zorluklar yaşanmaktadır. Ayrıca gözlük ve saçlarda yapılan belirgin değışiklikler sistemin yanılmasına neden olabilirler.

Eğer güvenlik temel kaygı ise bu yöntem yeterli sonucu veremeyebilir. [3]

2.6 Ses Tanıma

Ses Tanıma yöntemi daha önce kaydedilmiş ses bilgisi ile o an elde edilen ses bilgisinin karşılaştırılması mantığına dayanmaktadır. Ses Tanıma ve konuşma tanınmasının iki farklı görevi vardır. Konuşma tanınmasının amacı ne söylendiğini tanıma, Ses Tanımanın amacı ise kimin söylediğini bulmaktır.

Ses Tanıma söylenenin içindeki sesler veya telaffuz ile ilgilenmez, söylenenin içindeki insan sesi üzerine odaklanır. İnsan sesinin karakteristiğı ağız yapısı, geniz kavisi ve insan vücudunda ses çıkarma işlemine yardımcı olan diğ organlara bağılıdır.



Bu sistemin avantajı özel ve pahalı bir alt yapı gerektirmez. Ses Tanıma davranışsal karakteristikler üzerine oluşturulmuş bir sistemdir. Buna bağlı olarak fiziksel ortamdan negatif olarak etkilenmektedir. Örneğin networkdeki ve arka plandaki gürültülerden olumsuz yönde etkilenmektedir. [3]

2.7 İmza Tanıma

İmza hareketinden tanıma tekniği; imzaların karşılaştırılması yerine, imza atarken yapılan hareketlerin ölçülmesi mantığına dayanır. Hareket ölçümü basınç, yön, ivme, darbelerin uzunluğu ve darbelerin sayısının ölçülmesi ile elde edilir. Bu yöntemin en aşikar ve önemli avantajı, daha önce atılmış bir imzaya bakarak ölçüm prensiplerine baz teşkil eden bilgilerin elde edilememesidir.

İmzanın elde edilmesi maksadı ile kullanılan çeşitli araçlar mevcuttur. Bunlar geleneksel tabletler veya özel amaçlı tabletlerdir. Bunlarla kullanılan özel maksatlı kalemler imzanın üç boyutlu hareketinin elde edilmesinde kullanılır. Bu yöntemin iki dezavantajı mevcuttur, birincisi kullanıcı normal yaşantıda attığı imzanın aynısını özel aletlerle atamaz, ikinci ise imza atarken attığı imzayı göremez.

Bu sistem imzanın tanımlanmasında çıkan görüntü ile ilgilenmez, sadece imza atarken ortaya çıkan dinamik ile ilgilenir. Taklit edilme şansı vardır. Bu yüzden sadece gerçekleştirme maksadıyla kullanılması uygundur. Biometrik sistemler arasında kullanıcı yüzdesi % 2 dir. [3]

2.8 Klavyeye Yazma Ritmi Tanıma

Klavyeye yazma ritmi tanıma sistemleri kullanıcıları sistem girişi esnasında veya sistem girişi sonrası sürekli olarak tanıma amaçlı olmak üzere iki farklı şekilde tasarlanırlar. Ancak her iki durumda da kullanıcıları ayırt edebilen sistemler tasarlamak mümkündür.

Bu sistemler genel olarak kullanıcıların klavye tuşlarına basma süresi ve tuşdan parmağın kaldırılması ile tekrar tuşa basılması arasında geçen milisaniye cinsinden süreye göre kullanıcıları tanıma esasına dayanır. Kullanıcı ayırt etmede

kullanılabilecek ilave diğ er bir  zellik ise tuşlara basma basıncının  l  lmesidir. Kısaca klavyeye yazma ritmi tanıma sistemleri kullanıcıları ne yazdıklarına g re değıl nasıl yazdıklarına g re tanımaya  alışır.

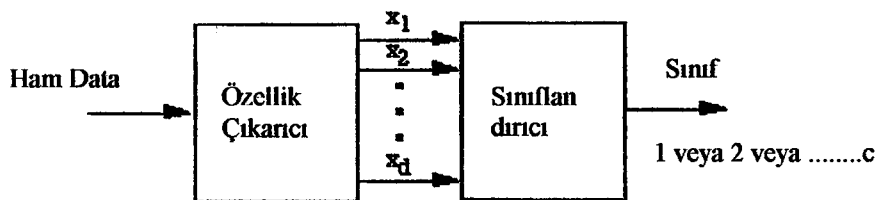
Bu sistemlerin en b y k avantajı zaten veri girişinde kullanılan klavye haricinde ilave bir donanım gerektirmemesi dolayısıyla d ş k maliyetle ekstra bir g venlik sağılamasıdır.  yle ki bu sistemleri kullanıcı başına 20–50 \$ gibi maliyetlerle temin etmek m mk n hale gelmiřtir. Bu y zden yakın gelecekte yaygın olarak kullanıcak bir sistem olmaya aday g r nmektedir.

Kullanıcı tanıma ve ayırt etme algoritma/y ntemleri konusunun daha iyi anlaşılabilmesi i in biyometrik sistemlerde kullanılan temel sınıflama algoritmalarının izahatını m teakip konu  zerinde bug ne kadar yapılan  alışmalar detaylı olarak kısım 2.10’da a ıklanmaktadır.

2.9 Biyometrik Sistemlerde Kullanılan Ana Sınıflama Algoritmaları ve  nceki Bazı  alışmaların Matematiksel Modelleri

Pattern veya iz tanıma sistemi olan biyometrik sistemler, genellikle elde edilmiř olan verinin bir sınıflama algoritmasından ge irilerek, verilerin sınıflandırmada kullanılacak referans iřaretlerle karřılařtırılması esasına dayanır.

Klasik sınıflandırma sistemlerinde tanımlanmak istenen veriler ham halden anlamlı bilgilere d n řt r l r. Bunu yapmak i in giriř verileri karakter ortaya  ıkarıcı olarak adlandırılabilcek bir algoritmaya sokulur, buradan elde edilen değıerler sınıflandırma algoritmasına girer ve sınıflandırmanın sonunda verinin hangi sınıfa ait olduğına karar verilir.



řekil 2.1. Klasik İz Tanıma Modeli

Burada alınan ham veriler önce sayısal X değerlerine dönüştürülür. Bu değerler X vektörünü oluştururlar. Sınıflandırıcı olarak adlandırılan algoritma ise bu değerleri alarak verileri daha önce tanımlanmış bir sınıfa eşitler. [4]

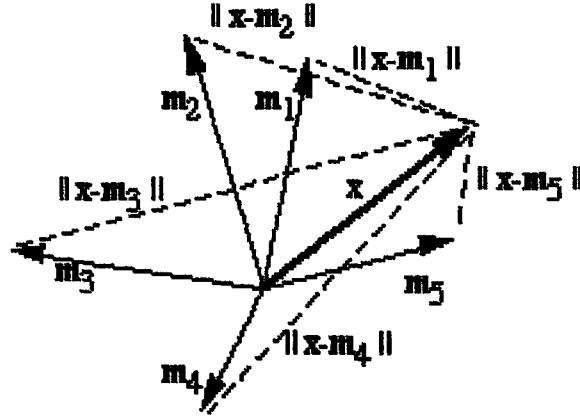
Biyometrik sistemlerinin sınıflandırılmasında kullanılan 3 önemli yöntem vardır. Bunlar

- Minimum-Fark Sınıflandırılması (Euclidian)
- Mahalanobis Sınıflandırılması (Bayes)
- En yakın K Komşu Sınıflandırması (KNN)'dir.

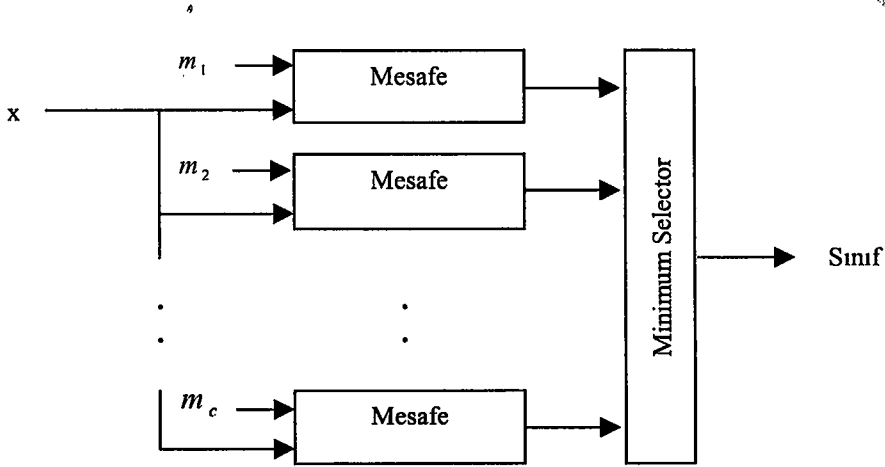
Şimdi bunları kısaca açıklayalım.

2.9.1 Minimum-Fark Sınıflandırması

Minimum-Fark sınıflandırmasında sınıflandırılacak vektör olarak kullanılan X vektörü ve gürültüden arındırılmış M şablon vektörleri vardır. Bu iki vektör arasındaki fark vektörleri $(x - m)$ bulunur ve fark vektörlerinden en küçük olanı seçilmek suretiyle sınıflandırma yapılır.



Şekil 2.2. Fark Vektörleri



Şekil 2.3. Minimum Fark Sınıflandırma Sistemi

2.9.2 Mahalanobis Sınıflandırması

Mahalanobis sınıflandırma yöntemi minimum fark sınıflandırma yöntemine göre daha gelişmiş ve güvenilir bir yöntemdir. Mahalanobis sınıflandırmasında metrik ve sınıflama işlemleri daha gelişmiştir.

Örneğin N boyutlu bir X iz vektörümüz olsun. Bu X vektörünün elemanları şöyle ifade edilebilir. X, X(1), X(2), X(3), X(4), ... ,X(n). Bu vektör elemanlarının aritmetik ortalaması alınır.

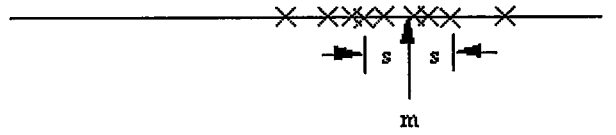
$$M = [X(1) + X(2) + X(3) + \dots + X(n)] / n \quad (2.1)$$

Burada M ortalama kütle diye de adlandırılabilir. Buradan izlere ait varyansı hesaplayacak olursak

$$V = [(X(1)-M)^2 + (X(2)-M)^2 + (X(3)-M)^2 + \dots + (X(n)-M)^2] / (n-1) \quad (2.2)$$

Görüldüğü üzere X ve M aynı boyutta, V ise farklı boyuttadır (X'in karesi). O halde V'nin karekökünü alacak olursak vektör değerlerine ait standart sapmayı bulabiliriz.

$$S = \text{sqrt} (V) \quad (2.3)$$



Şekil 2.4. X değerlerine ait dağılım

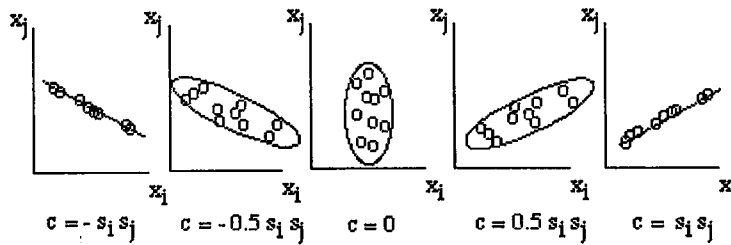
Bu dağılım eğer bir normal (Gaussian) dağılımsa m ortalamasının $+s$ ve $-s$ civarındaki salınımlarda, yani ortalamasının $+1$ ve -1 standart sapması içindeki izlerin %68.26'sı belirtilen aralıkta kalmaktadır. Aynı şekilde aralığı $+2s$ ve $-2s$ olarak ölçeklersek izlerin %95.4'ü belirtilen aralıkta yer alacaktır.

2.9.2.1 Kovaryans Metriği

Mahalobis sınıflandırmasında daha iyi sonuçlar elde etmek için Kovaryans metriği diye adlandırabileceğimiz bir yardımcı metrik kullanılır. Burada izler bir matris haline getirilir ve hesaplamalar bu yöntemle yapılır. Bu yöntem vektörler arasındaki farkların daha iyi ölçülmesini sağlar.

$$c(i,j) = \{ [x(1,i) - m(i)] [x(1,j) - m(j)] + \dots + [x(n,i) - m(i)] [x(n,j) - m(j)] \} / (n - 1) \quad (2.4)$$

Kovaryans aslında iki özelliğin birbirine göre değişimini ölçer. Buna göre Şekil 2.5.'de görüleceği gibi eğer iki özellik aynı yönde değişiyor yani doğru orantılı ise Kovaryans artı, tersi ise Kovaryans eksi değer alır. İki özellik birbirimden bağımsız ise de Kovaryans sıfır değerini alır.



Şekil 2.5. Kovaryans Değerleri

D boyutlu bir vektör olduğunu kabul edersek, bu vektöre ait Kovaryans matrisi Şekil 2.6.'daki gibi tanımlanır.

$$C = \begin{bmatrix} c(1,1) & c(1,2) & \dots & c(1,d) \\ c(2,1) & c(2,2) & \dots & c(2,d) \\ \vdots & \vdots & \ddots & \vdots \\ c(d,1) & c(d,2) & \dots & c(d,d) \end{bmatrix}$$

Şekil 2.6. Kovaryans Matrisi

$y = A x$ türünden ifadelerde A matrisi x vektörünü y vektörü türünden vektöre dönüştürür. Bu tür matrislere dönüşüm matrisleri denilir.

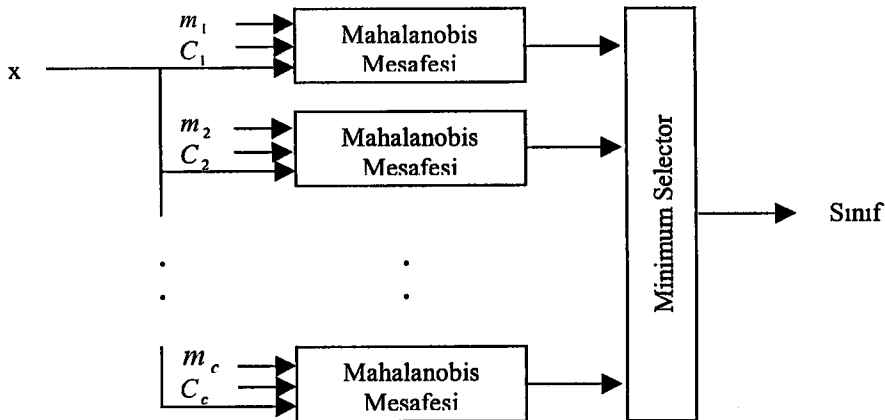
Vektör uzayı içerisinde bir x noktası ile bir y noktası arasındaki fark hesaplanmak istenirse bunun için aşağıdaki gibi bir bağıntı kullanılması gerekmektedir.

$$r^2 = \left[\frac{x - m}{s} \right]^2 = (x - m) \frac{1}{s^2} (x - m) \quad (2.5)$$

2.9.2.2 Mahalanobis Metriği

Mahalanobis metriğinde yöntem olarak Kovaryans matrisleri kullanılır. İki nokta arasındaki Mahalanobis farkı formül 2.6'deki gibi ifade edilebilir.

$$r^2 = (x - m_x)' C_x^{-1} (x - m_x) \quad (2.6)$$



Şekil 2.7. Mahalanobis Sınıflandırması

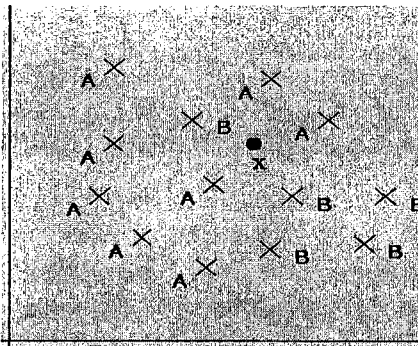
Burada m değerleri master (referans) değerleridir. X değerleri ise iz vektörüne ait değerlerdir. $C_1, C_2, C_3, C_4, \dots, C_c$ ise bu değerlere karşılık gelen Kovaryans matrisleri göstermektedir. Metrik bu ifadelerin algoritmada yerine koyulup hesaplanmasıyla elde edilebilir.

Mahalanobis sınıflandırmasının Minimum Fark Sınıflandırmasına göre avantajları şöyle sıralanabilir.

- Otomatik olarak koordinat eksenlerinin ölçeklemesini yani normalizasyonu yapar.
- Farklı özellikler arası korelasyonu düzeltir.
- Karar verme sınırlarının eğri olmasına imkan tanır.

2.9.3 K En Yakın Komşu (KNN) Sınıflandırması

Biometrik izleri sınıflandırmada kullanılabilecek yöntemlerden birisi de K en yakın komşu sınıflandırması olarak adlandırılabilen yöntemdir. Bu yöntem teorisi oldukça karmaşık sayılabilecek bir yöntemdir. Konunun pratikte anlaşılabilmesi açısından Şekil 2.8.'deki örnek değerlerin A ve B değerleri olduğunu varsayalım. Herhangi bir zamanda elde edilmiş olan X değeri, kendisine en yakın K adet komşu örnek değere bakılarak sınıflandırma yapılır. Şekil 2.8.'de örneğin $K = 1$ olarak belirlendiğinde sınıflandırma B 'dan yana, $K = 3$ olarak belirlendiğinde ise sınıflandırma A 'dan yana olabilir. Örnek değerler içerisinde kendisine en benzer olan değere eşitlenir. Böylece bir sınıflandırma yapılmış olur. Bu yöntem daha çok olasılığa dayanır. Bir izi tanımlamak için çevresindeki izler hesaplamaya konulur ve bir olasılık bulunmaya çalışılır.



Şekil 2.8. En yakın K Komşu Sınıflandırması Örnek Dağılımı

2.9.4 Önceki Bazı Çalışmaların Matematiksel Modelleri

Kaynaklarda belirtilen birtakım makalelerde herhangi bir matematiksel formül/modele göre anlatım yapılmadığından, birtakım makalelerde ise çok kapsamlı bir şekilde formül v.s. lerle anlatıldığından bu bölümde giriş bölümünde kısaca değinilen önceki çalışmaların referans makalelerde belirtildiği/anlaşılabildiği kadarıyla kısaca matematiksel modelleri anlatılmaya çalışılmıştır.

1989'da S.Bleha, C.Slivinsky ve B.Hussein tarafından Minimum-Fark (Euclidian) ve Bayes (Mahalanobis) sınıflandırması algoritmaları uygulanmış [2] ve doğrulama/gerçekleme safhasında (verification);

Normalize edilmiş Minimum-Fark sınıflandırmasında Test Vektörü ile Ortalama Vektörü arasındaki mesafe (2.7) ve (2.8) ile hesaplanarak gerçekleştirilmiştir.

$$D_i(X) = \frac{(X - m_i)^t (X - m_i)}{\|X\| \|m_i\|} < T_1 \quad (2.7)$$

Normalize edilmiş Bayes sınıflandırmasında ise

$$d_i(X) = \frac{(X - m_i)^t C_i^{-1} (X - m_i)}{\|X\| \|m_i\|} < T_2 \quad (2.8)$$

(2.7) ve (2.8) ifadelerinde;

X : Test vektörü

m_i : Ortalama Vektörü

C_i : Kovaryans Matrisi

T_1, T_2 : Eşik Değerleri
dir.

Bu çalışmada kullanıcı ortalama vektorlerinin tespiti için her kullanıcıya 30 giriş yaptırılmış ve Eşik Seviyesi olarak Minimum-Fark sınıflandırmasında birinci denemede 0.030, ikinci denemede 0.029, Bayes sınıflandırmasında ise sırasıyla 0.000030 ve 0.000029 değerleri kullanılmış ve test vektörü ile ortalama referans vektörü arasındaki mesafe yukarıda belirtilen eşik değerinden küçük ise test denemesinin gerçek veya yetkili kullanıcıya ait olduğuna karar verilmiştir. Dikkat edileceği üzere bu çalışmada eşik seviyeleri sabit tutulmuştur.

1997 ve 1999 yıllarında F.Monrose ve A.Rubin tarafından gerçekleştirilen çalışmalarda [6][7] sırasıyla Euclidean mesafesini ölçme, Ağırlıksız olasılık, Ağırlıklı olasılık ve Bayes sınıflama algoritmalarına göre denemeler yapılmıştır.

Euclidean mesafesini ölçme algoritmasında [7]

$$D(R, U) = \left[\sum_{i=1}^N (r_i - u_i)^2 \right]^{1/2} \quad (2.9)$$

Ağırlıksız Olasılık algoritmasında[7]

$$Score(R, U) = \sum_{i=1}^N S_{u_i} \quad (2.10)$$

$$S_{u_i} = \frac{1}{o_{u_i}} \left[\sum_{j=1}^{o_{u_i}} Prob\left(\frac{X_{ij}^{(u)} - \mu_{r_i}}{\sigma_{r_i}}\right) \right] \quad (2.11)$$

Ağırlıklı Olasılık algoritmasında[7]

$$Score(R, U) = \sum_{i=1}^N (S_{u_i} * weight_{u_i}) \quad (2.12)$$

$$weight_{u_i} = \begin{cases} 0 & \text{Ui veya Ri boşsa} \\ \frac{o_{u_i}}{\sum_{k=1}^N (o_{u_k})} & \text{Diğer} \end{cases} \quad (2.13)$$

Bayes sınıflandırması algoritmasında [7]

$$\Delta^{\alpha}(x, x') = \sum_{i=1}^n w_i \left(\frac{|x_i - x'_i|}{\sigma_i} \right)^{\alpha} \quad (2.14)$$

(2.9),(2.10),(2.11),(2.12),(2.13) ve (2.14) ifadelerinde

U : Test Vektörü

R : Referans (Ortalama) Vektörü

$X_{ij}^{(u)}$: U test vektörünün i'inci özelliğinin j'inci tekrarı

O_{u_i} : Tekrar Miktarı

x_i : Özellik Vektörü

w_i : Ağırlık Vektörü dür.

olup vektörlerin her bir elemanı ortalama, standart sapma, tekrar sayısı ve değerinden oluşmaktadır. Ağırlıklı olasılık ve Ağırlıksız olasılık algoritmalarında yukarıdaki formüllere göre hesaplamalar yapıldıktan sonra veritabanındaki en yakın komşu ile ilişkilendirilmektedir. Bayes sınıflandırmasında ise özellik vektörleri Faktör Analizi (FA) metodu ile hesaplanmaktadır. Tüm algoritmalarda ise dağılımların normal dağılıma benzediği hipoteziyle çalışmalar yapılmıştır.

2000 yılında Aykut GÜVEN tarafından yapılan çalışmada [10] sistemin girdi olarak alabildiği iki adet parametre vardır. Bunlar B(I) dizisi şeklinde ifade edilebilecek tuş basma zamanları ve G(I) şeklinde ifade edilebilecek tuş gecikme zamanlarıdır. Sistemin diğer aşamalarında ihtiyacı olan diğer parametreler ve fonksiyonlar bu iki diziden türetilirler.

Biyoritim sisteminin tasarımında matematiksel modelleme yapılırken aşağıdaki notasyonlar kullanılmıştır.

B(I) : Herhangi bir I . tuşa basma işlemi yapılırken geçen süre.

G(I) : I-1 .tuştan I . tuşa basılırken arada geçen gecikme süresi.

θ_b : Referans B vektörü ile ölçülen B vektörü arasındaki açının farkı.

θ_g : Referans G vektörü ile ölçülen G vektörü arasındaki açının farkı.

Γ_b : B vektörü için ağırlık çarpanı.

Γ_g : G vektörü için ağırlık çarpanı

ξ : Karar fonksiyonu

Ψ_b : B vektörü

Φ_g : G vektörü

Ω : ξ için ağırlık çarpanı

B_i 'nin N boyutlu bir uzayda birbirlerine dik ve ortogonal boyutlardan herhangi birisi olduğu kabulü yapılarak; Şifre Uzunluğu olan N ,N boyutlu bir uzaya karşılık geldiği yani şifrenin her bir karakteri N boyutlu uzayın bir boyutunu tasvir ettiği belirtilmektedir.

Sistemin kullanıcı girişinden türettiği parametrelerden B ve G dizileri tanımlanarak,

$$B = B(1)+B(2)+B(3)+B(4)+B(5)+.....+B(N) \quad (2.15)$$

$$G = G(1)+G(2)+G(3)+G(4)+G(5)+.....+G(N-1) \quad (2.16)$$

Yukarıda tanımlanan B dizisinin elemanlarından bir N boyutlu bir vektör türetilmiştir.

$$\Psi_b = \sum B(i) I_i \quad ; \quad N \text{ Boyutlu } \Psi_b \text{ vektörü} \quad (2.17)$$

$$\Phi_g = \sum G(j) I_j \quad ; \quad N-1 \text{ Boyutlu } \Phi_g \text{ vektörü} \quad (2.18)$$

Ψ_b, Φ_g vektörleri sistemden gerçek zamanlı olarak türetilmiş olan vektörlerdir. Aynı zamanda sistemin referans olarak kullandığı ve sistemin daha önce üretmiş olduğu iki adet benzer vektör şöyle ifade edilmiştir.

$$\Psi_{br} = \sum B_r(i) I_i \quad ; \quad N \text{ Boyutlu } \Psi_{br} \text{ vektörü (Referans Vektörü)} \quad (2.19)$$

$$\Phi_{gr} = \sum G_r(j) I_j \quad ; \quad N-1 \text{ Boyutlu } \Phi_{gr} \text{ vektörü (Referans Vektörü)} \quad (2.20)$$

A ve B iki benzer ve aynı boyutta vektör olmak üzere ; N boyutlu uzayda iki vektör arasındaki açı ise

$$\text{Cos } \theta = AB / [(\sum A^2)(\sum B^2)]^{1/2} \text{ şeklinde ifade edilmiş ve} \quad (2.21)$$

$$\text{Cos}\theta_b = \Psi_b \Psi_{br} / [(\sum \Psi_b^2)(\sum \Psi_{br}^2)]^{1/2} \quad (2.22)$$

$$\text{Cos}\theta_g = \Phi_g \Phi_{gr} / [(\sum \Phi_g^2)(\sum \Phi_{gr}^2)]^{1/2} \text{ bağıntıları türetilmiştir.} \quad (2.23)$$

Burada Ψ_b ve Ψ_{br} vektörleri birbirlerine ne kadar benzerlerse $\text{Cos}\theta_b$ 'nin değeri o kadar 1'e yaklaştığı ve $\text{Cos}\theta_b$ 'nin teorik (ideal) değerinin 1 olduğu belirtilmiştir.

Aynı durumun Φ_g ve Φ_{gr} vektörleri içinde geçerli olduğu ve

$\text{Cos}\theta_b$ değeri için ağırlık çarpanı Γ_b ; $\text{Cos}\theta_g$ değeri için ağırlık çarpanı Γ_g olarak tanımlanarak;

Γ_b ve Γ_g terimleri arasındaki ilişki (2.24) ve (2.25) 'deki gibi tanımlanmış

$$\Gamma_b + \Gamma_g = 1 \quad (2.24)$$

$$\Gamma_g = 1 - \Gamma_b \quad (2.25)$$

(2.24) ve (2.25) ifadelerinden ξ karar fonksiyonu türetilmiştir.

$$\xi = \Gamma_g \text{Cos}\theta_g + \Gamma_b \text{Cos}\theta_b \quad (2.26)$$

Ψ_b, Ψ_{br} ve Φ_g, Φ_{gr} vektörleri arasındaki açı farkının ağırlık fonksiyonlarıyla çarpımından ibaret olduğu belirtilerek,

(2.26)'daki ifadeden aşağıdaki bağıntı elde edilmiş ve

$$\xi = \Gamma_g (\text{Cos}\theta_g - \text{Cos}\theta_b) + \text{Cos}\theta_b \quad (2.27)$$

i değeri, sistemin herhangi bir zamandaki durumunu ifade etmek ve

ξ_{i-1} : Veritabanından alınan i. durumdan önceki girişe ait karar fonksiyonu olmak üzere;

Ω : Ağırlık çarpanı, $0 < \Omega < 1$

$$\xi_{i-1} - (\Omega \xi_{i-1}) < \xi_i < \xi_{i-1} + (\Omega \xi_{i-1}) \quad (2.28)$$

bağıntısı türetilmiştir.

Eğer elde edilen değer yukarıda ifade edildiği gibi bir aralık içerisindeyse biometrik sistem bu kullanıcıya giriş izni vermektedir.

2.10. Klavyeye Yazma Ritmine Göre Kullanıcı Tanıma Sistemlerinde Bugüne Kadar Yapılan Çalışmalar

Aslında konunun ilk ortaya çıkışı II. Dünya savaşına kadar uzanmaktadır. Nitekim II. Dünya savaşı sırasında Mors kodları ile telgraf yazan bir telgrafçıyı tanıma amaçlanmış ve gerçekten de telgraf yazan bir telgrafçının mors kodları ile yazma ritminin kendine özgü olduğu yani tek olduğu anlaşılmıştır.

Ancak klavyeye yazma ritmlerinin herkesin farklı olduğunu ispatlamaya yönelik ilk çalışmalar ise 70'li yılların sonunda başlamış ve 1980'de National Science Foundation sponsorluğunda Rand Corporation adına R.Stockton Gaines, William Lisowski, S.James Press, Norman Shapiro tarafından 7 sekreterden 1'er sayfalık 3 ayrı metni yazmaları istenmiş ve aynı metinlerin yine bu kişilerce 4 ay sonra yazılması istenmiştir. [5] Bu metinlerin 1.si, sıradan bir İngilizce metinden oluşurken, 2.si rastgele seçilen İngilizce kelimelerden, 3.si ise rastgele seçilen İngilizce metin kalıplarından oluşmaktaydı. Bu çalışmada sadece 10 kereden daha fazla tekrar eden sınırlı sayıda karakter çifti (diagraph) arası zamanlar incelendi ve 4 ay arayla yapılan diagraph zamanları ortalamalarının aynı ve varyanslarının aynı olduğu varsayımıyla bağımsız istatistik metodlarından olan T-Test istatistik metodu kullanıldı. Çalışmalar cesaret verici olmasına rağmen, çalışmaların sadece 7 sekreterle sınırlı kalması ve kullanıcıların kendisini sisteme tanıtmalarının çok uzun sürmesinden dolayı uygulanabilir bir sistem ortaya konulamadı.

1988'de Saleh Bleha tarafından konu ile ilgili doktora tezi yapıldı [6] ve 1989'da Saleh Bleha, Charles Slivinsky ve Bassam Hussein tarafından Saleh Bleha'nın doktora tezi paralelinde çalışmalar sürdürülerek konuya farklı bir boyut getirilmeye çalışıldı. [7] Bu çalışmada Minimum-Fark ve Bayes (Mahalanobis) Sınıflandırması algoritmaları test edildi ve şahısların isimleri ile belirli kalıptaki metinler (phrase) olmak üzere iki tip şifre kullanıldı. Kimlik tespitinde belirli kalıptaki metinler, doğrulama ve genel tanıma sisteminde ise şahısların isimleri kullanıldı. Kimlik tespiti safhasında 10 gönüllü kullanıcı test edildi ve % 1.2 yanlış karar verildi. Doğrulama sisteminde 26 gönüllü kullanıcı test edildi ve % 8.1 FAR ile % 2.8 FRR sonuçları elde edildi. Genel tanıma sisteminde ise 32 gönüllü ile % 3.1 FAR ve %0.5 FRR sonuçları elde edildi. Bu çalışmada kimlik tespiti safhasında şifre uzunluğu, şifre tipi (şahısların isimleri ile belirli kalıptaki metinler), referans izi oluşturabilmek için gerekli veri miktarı ve iz boyutlarının azaltılması gibi parametrelerin etkileri araştırılırken, doğrulama safhasında kullanıcı gerçekleştirme için gerekli eşik seviyelerinin etkisi her iki sınıflandırma algoritmasına göre de araştırıldı. Bu araştırmalarda şahısların isimlerinin belirli kalıptaki metinlerden daha iyi tanıma sağladığı, sisteme tanıtım için ne kadar çok veri girişi yapılırsa ve şifre uzunluğu ne kadar artarsa hata oranının buna göre azaldığı tespit edilmiştir. Örneğin özellikle kısa şifrelerde Bayes sınıflandırmasının Minimum-Fark sınıflandırmasına göre yaklaşık % 30 daha az hata oranına sahip olduğu gibi sonuçlar elde edilmiştir. Bu çalışmadan elde edilen diğer enteresan bir sonuç ise kullanıcıların aynı verileri iki kere girdiklerinde ortalamanın değilde küçük olanın hesaplamada kullanılmasının daha iyi sonuçlara götürdüğü tespit edilmiştir.(Shuffling Procedure)

1985 ile 1991 yılları arasında Glen Williams, John Leggett ve David Umphress tarafından 17 yüksek lisans sahibi programcıyla birlikte benzer çalışmalar yapıldı [8],[9],[10],[11] ve 1980'de yapılan çalışma [4] neticelerini değerlendiren bir yöntem geliştirildi ve % 5.5 FAR ile % 5 FRR neticeleri elde edildi. Bu çalışmalarda ortalama diagraph zamanının 0.5 standart sapması içinde kalan ve referans iz ile test izi arasında % 60'lık benzerlik olan kullanıcılara sisteme giriş izni verildi. Ancak daha önceki çalışmalarda olduğu gibi kullanıcıların kendini sisteme tanıtabilmesi için 1000 den fazla kelimenin yazılması gerekiyordu ve özellikle FRR çok yüksekti, dolayısıyla pratikte uygulanabilir bir sistem değildi. Yine bu çalışmalarda kullanılan bir diğer yöntem ise yetkisiz kullanıcıların saf dışı bırakılması için tüm sistemde tanımlı kullanıcılar için 500 ms tek bir alçak geçiren zaman filtresinin kullanılmasıdır

1980'li yılların ortalarında yapılan diğer çalışmalar ise J.Garcia ile J.Young ve R.Hammon tarafından yapılan çalışmalardır. [12], [13] Garcia'nın çalışması referans tuş geçiş süreleri vektörünün kovaryans matrisini kullanıcı izinin tutarlılığının bir ölçüsü olarak ele alır ve daha sonra referans ile test profilleri arasındaki benzerliği belirlemek üzere Mahalanobis uzaklığı fonksiyonunu kullanır. Young ve Hammon ise tuşa basma basınçlarını ve önceden belirlenen sayıda kelimeyi yazma zamanı v.b. gibi birkaç özellik vektörü arasındaki Euclidean mesafesini ölçme tekniğini kullanmış olmakla birlikte sırasıyla 1986 ve 1989 yıllarında patenti alınan iki çalışmada ticari amaçla geliştirildiğinden sistemlerin performansına ilişkin herhangi bir veri elde yoktur.

Klavyeye yazma ritmi tanıma sistemlerinde umut verici çalışmalardan biri 1990 yılında Rick Joyce ve Gupta Goyal tarafından yapılan çalışmadır. [14] Onların konuya yaklaşımı oldukça basit olmakla birlikte oldukça etkileyici sonuçlar elde edilmiştir. Bu çalışma, sistem girişi esnasında kullanıcıları tanıma amaçlı dizayn edilmiş bir sistemdir. Sistemin kullanıcıları tanınması için kullanıcılardan adı ,soyadı ile kullanıcı adı ve şifresini 8 defa sisteme girmesi istenir. Bu girişlerden ortalama referans vektörü çıkarılır. Daha sonra sistem girişi esnasında referans iz ile sistem girişi izleri karşılaştırılır ve fark belli bir eşik seviyesinin altındaysa sisteme giriş müsaadesi verilir, değilse verilmez. Bu çalışmada % 0.25 FRR ve % 16.36 FAR oranlarına erişilmiştir.

1989'da Saleh Bleha, Bassam Hussein ve Robert McLaren tarafından, [15] 1993'de Marcus Brown ve Samuel Joe Rogers tarafından [16] , 1996'da Thomas J. Alexandre [17] tarafından, 1998'de Sungzoon Cho, Chigeun Han, Dae Hee Han, Hyung_II Kim tarafından (1) ve 1999 yılında ise A.Sermet Anagün tarafından [18] yapay sinirsel ağ yaklaşımıyla kullanıcılar tanınmaya ve ayırt edilmeye çalışılmıştır. Küçük veri tabanlarında geri yayılma (back propagation) algoritması ile iyi sonuçlar elde edilmiş olmakla birlikte veri tabanına her yeni kullanıcı tanıtımında sistemin kullanıcı tanıma safhası tekrar çalışmak durumunda kaldığından özellikle erişim kontrolü amacıyla kullanılan sistemlerde bu durum oldukça zaman alıcı ve pahalı olabilmektedir. Sürekli olarak sistemin öğrenme problemini ortadan kaldırmak için veri tabanını küçük parçalara ayırma prosesi ne kadar uygulanırsa uygulansın kullanıcı değişim veya dönüşüm hızının fazla olduğu sistemlerde sistemin tekrar öğrenmeyi başarması oldukça fazla zaman almaktadır. Bu nedenle bu sistemler



bugüne kadar bu alanda gerçekten uygulanabilir bir sistem olarak tam anlamıyla ortaya konulamamışlardır.

1993 yılında Gentner tarafından yapılan çalışmada klavyeye yazma konusunda deneyimli, usta personelin ortalama tuş geçiş zamanının yaklaşık 96 ms iken klavyeye yazma konusunda tecrübesiz kişilerin ise ortalama tuş geçiş zamanının yaklaşık 825 ms olduğu tespit edilmiştir. [19] Bu nedenle Glen Williams, John Leggett ve David Umphress tarafından kullanılan 500 ms alçak geçiren filtrenin klavyeye yazma konusunda deneyimli, usta kişileri sisteme kabul ederken, klavyeye yazma konusunda tecrübesiz kişileri ise saf dışı bıraktığı ortaya konmuş ve alçak geçiren filtre yerine karakter çiftleri arası tuş geçiş zamanı değişkenlik ölçümlerinin, doğrulama safhasında kayde değer ilerlemeler sağlayabileceği belirtilmiştir.

1997 yılında Fabian Monroe ve Aviel Rubin tarafından yukarıda anlatılan klavye tuşları arası geçiş süresine ilaveten tuş basma zamanları da dikkate alınarak şahısların isimleri ile belirli kalıptaki metinler yerine serbest stilde yazılan metinlerin kullanımı ve çeşitli yaşta, kültür ve bilgi birikimine sahip değişik milliyetlerden insanlardan oluşan bir örnek kümesi teşkil edilerek çalışmalar daha da ilerilere götürülmeye çalışılmıştır. [20] Bu çalışmada kullanıcıları yazma hızlarına göre kümeleme mantığı uygulanmış ve Euclidean mesafesini ölçme, Ağırlıksız olasılık, Ağırlıklı olasılık ve Bayes sınıflama algoritmalarına göre denemeler yapılmış ve sırasıyla % 83.22, % 85.63, % 87.18 ve % 92.14 oranında kullanıcılar tanınabilmiştir. Ağırlıksız olasılık algoritmasında dağılımın normal dağılıma benzediği varsayımıyla ve en yakın komşu kuralının da birlikte işletildiği bir algoritma kullanılırken, ağırlıklı olasılık algoritmasında ise örneğin İngilizce’de th, er, re karakter çiftlerinin qu veya ts karakter çiftlerinden daha fazla dil içinde kullanılmasından dolayı karakter çiftlerinin ağırlıkları da dikkate alınarak hesaplamalar yapılmıştır. 1999 yılında Fabian Monroe ve Aviel Rubin tarafından konu ile ilgili tekrar bir makale yayınlanmış ancak bu 1997 yılında yapılan çalışmanın tekrarı olmuştur. [20]

1997 yılında yapılan bir diğer çalışma ise Dawn Song, Peter Venable, Adrian Perrig tarafından yapılan çalışmadır. [22] Bu çalışmanın diğer çalışmalardan olan en önemli farklarından birisi kullanıcıların sadece sistem girişi esnasında değil sürekli olarak yazma ritimlerinin incelenerek kullanıcıların ayırt edilmeye çalışılmasıdır. Bu çalışmada iki karakter arası gecikme zamanı histogramlarının normal dağılıma

benzediği tespit edilmiş ve iki karakter çifti arasındaki geçiş süresinin ortalama ve varyansını modelleyen Markov zincirine benzer bir yapıyla kullanıcıların ayırt edilmesine çalışılmıştır. Ancak söz konusu makalede algoritmanın ve deneysel çalışmaların hata oranı ve tanıma olasılıklarına ilişkin herhangi bir açıklama bulunmamaktadır.

1999'da Fabian Monroe, Michael K. Reiter, Susanne Wetzel tarafından polinomlar kullanılarak özellikle offline saldırılara karşı yazma ritminin tanınması zorlaştırılmaya çalışılmıştır. [22] Bu çalışmanın diğer çalışmalardan ayrılan yönü klavyeye yazma ritmi şifresinin aynı zamanda dosyaların şifrlenmesinde ve VPN'lere erişimde de kullanılabilecek olmasıdır. Ancak bu çalışmada da teste tabi tutulan kullanıcılar sınırlı kalmış ve deneysel çalışmalarda parametrelerin değişik değerleri için yeterince inceleme yapılamamış ve en fazla % 77.1 oranında tanıma oranına erişilebilmiştir.

2000 yılında Alen Peacock tarafından kullanıcıların kendilerini sisteme tanıtmaları için ad, soyad ve belli bir metin kalıbından oluşan bilgileri 20 defa girmesi istenmiş ve modifiye edilmiş k en yakın komşu kuralı (K Nearest Neighbor) ile kullanıcıların % 92'si başarı ile tanınabilmiştir. [24] Web tabanlı geliştirilen ve 11 kullanıcının test edildiği bu çalışmada 95 yetkisiz giriş denemesinin 4'ünde başarı sağlanmış ve % 4.2 FRR oranı elde edilmiştir. Başarılı olunan 4 denemede aynı kullanıcı tarafından gerçekleştirilmiştir.

2000 yılında yapılan bir çalışmada ise vektörler arası açının hesaplanmasına dayalı bir algoritma geliştirilmiş ve karar vermede tuşlara basma sürelerinin % 10, tuşlar arası geçiş sürelerinin ise % 90 oranında etkili olduğu varsayımıyla hesaplamalar yapılmıştır. [2] Tarafından yapılan çalışmalar ise özellikle klavyeye yazma konusunda deneyimli kişilerde tuşa basma sürelerinin tuşlar arası geçiş sürelerinden daha uzun ve dolayısıyla daha ağırlıklı olabildiğini göstermiştir. Ayrıca bu çalışmada kullanıcıların sistem tarafından tanınabilmesi için herhangi bir zaman limiti verilmeden programın kullanıcıyı tanıyana kadar veri girişi yapması gerektiği belirtilmiştir. Şifre seçiminde ise kullanıcılar 8 karakterlik şifre girmek zorunda bırakılarak deneysel çalışmalarda sadece yetkisiz kullanıcıların yetkili kullanıcılar yerine girme deneylerine yani FAR hata oranlarına yer verilmiş, FRR hata oranına ilişkin ise herhangi bir sonuç ortaya konulmamıştır. Ayrıca sistemin kullanıcıları tanımda kullandığı eşik seviyesi sabit tutulmuştur.

Son yıllarda kişilerin klavyeye yazma ritmi farklılığının sistem güvenliğini sağlamak yerine klavyeye yazma ritmlerinin tespit edilmesi suretiyle kullanıcı şifrelerinin hacker'lar tarafından daha kısa sürede deşifre edilebildiğine yönelik çalışmalar da yapılmaktadır. Konu ile ilgili olarak yapılan bir çalışmada [25] özellikle Unix işletim sisteminde sunumcu ile istemci arasında iletişim hattını şifrelemekte kullanılan ve oldukça kuvvetli kriptolama algoritmalarıyla donatılmış olan SSH' in (Secure Shell) gönderilen paketlerin 8 byte'lik paketler halinde sabit olarak gönderilmesi ve interaktif modda klavyeye her basışın, (Shift, Control v.b. tuşlar hariç) ardından ayrı bir paket olarak gönderilmesinden dolayı kullanıcı şifrelerinin normal bir Brute-Force saldırısıyla 7-8 karakterli bir şifreyi çözme süresine göre 50 kat daha kısa sürede çözülmesinin mümkün olduğu ve dolayısıyla SSH'in aslında sanıldığı kadar güvenli bir sistem olmadığı ortaya konmuştur. 142 karakter çiftinin denendiği ve HMM algoritmasının uygulandığı bu çalışmada kişilerin şifrelerini yazarken bastıkları karakter çiftlerini farklı/aynı ellerle, farklı/aynı parmakla yazıp yazmadığı ve şifre içinde rakam kullanılıp kullanılmadığına bakılarak çalışmalar yapılmış ve çok enteresan sonuçlar elde edilmiştir. Bu çalışmada aynı elle yazılan karakter çiftlerinin farklı ellerle yazılan aynı karakter çiftlerine göre daha fazla zaman aldığı, bunun büyük bir olasılıkla farklı ellerin paralel kullanılmasından kaynaklandığı değerlendirilmiştir. Örneğin bu çalışmada iki karakter çifti arasında geçen süre 175 ms'den fazla ise bu çiftin tek el kullanılarak yazıldığı veya 300 ms'den fazla bir süre ise bu karakter çiftinin aynı elle yazıldığı ve bir harf bir rakam çiftinden oluştuğu gibi enteresan sonuçlar ortaya çıkarılmıştır. Diğer bazı çalışmalarda ortaya çıkarıldığı gibi karakter çiftleri dağılımlarının normal dağılıma benzediği bu çalışmada da tespit edilmiştir.

Bu tez çalışmasındaki kaynaklarda kaynak olarak gösterilen ancak tezin yazımına kadar temin veya hakkında bilgi edilememiş olan makale ve dokümanlara, bundan sonra konu üzerinde çalışma yapacaklara yardımcı olmak maksadıyla kaynaklarda yer verilmiştir. [26], [27], [28]

3. BİLİŞİM SİSTEMİ GÜVENLİĞİNDE YAZMA RİTMİ TABANLI SİSTEM TASARIMI

Kullanıcıları klavyeye yazma ritimlerine göre tanıma amaçlı bu sistemde veri tabanı olarak önce Paradox daha sonra ise Oracle, arayüz tasarımı ise Delphi yazılımı kullanılmıştır.

Söz konusu sistemde kullanıcıların herhangi bir sisteme girişi anındaki (login) yazma ritmi tanınmaya çalışılarak sisteme giriş izni verilmekte veya verilmemektedir. Günümüzde tanıma sistemlerinin sisteme girişten sonra da sürekli olarak kullanıcı yazma ritimlerinin takip edilmesi suretiyle de çalışmalar yapılmakta olup tasarlanan sistemde ise sadece sisteme giriş esnasında kullanıcıyı ayırt etme ve tanıma amaçlanmıştır.

Yazma ritmi tanıma sistemi, biyometrik sistemler içinde ticari olarak kullanılmaya açılan son biyometrik sistemdir. Konu ile ilgili çalışmalar 70'li yılların sonunda başlamış ancak son birkaç yıl içinde ticari uygulamalara geçilmiştir. Bu yüzden hala yazma ritmi sisteminin yaygın olarak kullanıldığını söyleyebilmek mümkün değildir. Konu ile ilgili bilinen 2 ticari yazılımda da hali hazırda sisteme giriş esnasında kullanıcılar tanınmakta olup tanıma olayının sürekli yapılması konusunda firmalar/araştırma grupları tarafından da çalışmalar devam etmektedir.

Tasarlanan sistemi aşağıda belirtilen safhalara ayırmak mümkündür.

1. Öğrenme Safhası
2. Kimlik Tespiti Safhası
3. Doğrulama/Gerçekleme Safhası
4. Karar/Güncelleme Safhası

Öğrenme safhasında kastedilen giriş yapmak isteyen kullanıcıların kendini sistem tarafından belirlenen miktarda girişler yapmak suretiyle sisteme tanıtması ve sistemin kullanıcıları tanıması yani öğrenmesi olarak ifade edilebilir. Tasarlanan sistemde kullanıcıların kendini sisteme tanıtabilmesi için sistem yöneticisine 8 ila 24 arasında değerleri ayarlayabilmesi, dolayısıyla programın kullanıcıları tanıyabilmesi

için gerekli olan giriş sayısını tayin edebilme imkanı verilmiştir. Doğal olarak sistem yöneticisince yapılan ayara göre ne kadar çok veri girişi yapılıyorsa sistemin o kadar daha sağlıklı karar vereceği yani o kadar iyi öğreneceği söylenebilir. Ancak daha önce konu ile ilgili yapılan çalışmalarda ve bu tez çalışmasında 8 kere ad, soyad ve şifrenin girilmesinin dahi kullanıcıları ayırt edebilmeye imkan verdiği tespit edilmiş ve bu değer kullanıcıları çok sıkmadan sistemin kullanıcıları tanınması bakımından varsayılan olarak 8'e ayarlanmıştır. Zira ileride açıklandığı üzere bu safhada ortaya çıkabilecek zaafiyetler güncelleme safhasında anlatıldığı şekilde kullanıcıların belirli aralıklarla (ki bu çalışmada sistem yöneticisince seçilen kullanıcı tanıma için gerekli giriş sayısı kadar sisteme başarılı giriş yapıldığında) özellik vektörlerinin güncellenmesi suretiyle bertaraf edilmektedir.

Kimlik tespiti safhası sisteme girmek isteyen kullanıcının, kimlik bilgilerinin öğrenme safhasında sisteme tanıtılmış kullanıcılardan biri olup olmadığının tayin edilmesinden ibarettir. Ancak kimlik bilgileri doğru olan sisteme girmek isteyen kullanıcıların her zaman gerçek kullanıcı olduğunu söylemek mümkün değildir. Çünkü gerçek hayatta olduğu gibi tüm kimlik kartlarının gerçek olduğunu, sahte olmadığını söylemek mümkün değildir. Zira programda sisteme giriş yapan kullanıcının girdiği ad, soyad ve şifre bilgilerinin, öğrenme safhasında oluşturulan kullanıcılar tablosundaki kullanıcılardan herhangi birine uyup uymadığının kontrolü yapılır ve uyuşum varsa bir sonraki safhaya yani doğrulama ve gerçekleştirme safhasına geçilir, uyuşum yok ise böyle bir kullanıcının sistemde tanımlı olmadığı mesajıyla kullanıcı uyarılarak tekrar veri girişi yapması beklenir.

Doğrulama/gerçekleştirme safhasında ise yukarıda kısaca açıklandığı gibi sisteme girmek isteyen kullanıcının sistemde tanımlı olan kullanıcının kendisi olup olmadığının ayırt edilmesi gerçekleştirilmekte, bu yapılırken öğrenme safhasında sistem tarafından çıkarılan ilgili kişinin biometrik özellikleri ile sisteme giriş esnasındaki biometrik özelliklerinin karşılaştırılması yapılarak aradaki fark algoritmik olarak belirli bir sınırın altında ise sisteme girmek isteyen kullanıcı tanınmış olarak kabul edilerek sisteme giriş izni verilmekte, aksi halde sistem girişine müsaade edilmemektedir. Sınıflama/ayırt etme algoritmasının uygulandığı safha bu safha olup algoritma detayları 3.1.'de anlatılmaktadır.

Karar/Güncelleme safhasından kast edilen ise kullanıcıların zamanla yazma ritimlerinin değişebileceği gerçeğinden hareketle zaman içinde kullanıcıların yazma

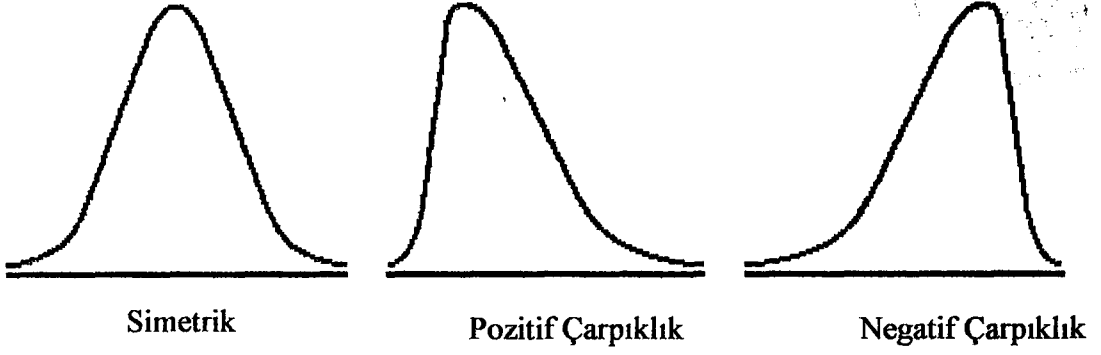
ritmi biyometrik özelliklerinin güncellenmesive nihayetinde giriş yapan kullanıcının gerçek kullanıcı olup olmadığına karar verilmesidir. Zira bu tez çalışmasında da kullanıcıların sisteme her öğrenme sayısı kadar başarılı giriş yaptığıında biyometrik özellikleri güncelleştirilerek bir nevi sürekli öğrenen bir sistem tasarlanmıştır.

3.1 Biyolojik Yazma Ritmi Analiz Etme Yazılımı Algoritması ve Matematiksel Modeli

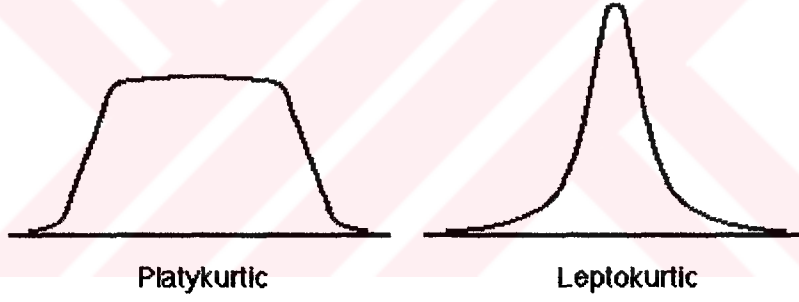
Programda kullanılan iki algoritmanın da temeli, normal dağılımda örneklerin aritmetiksel ortalama ile onun ± 1 standart sapma uzağı arasında olma olasılığının % 68.26 , ± 2 standart sapma uzağı arasında olma olasılığının % 95.4 , ± 3 standart sapma uzağı arasında olma olasılığının ise % 99.7 olması esasına dayanır.

Bilindiği üzere normal dağılımda eğri aritmetik ortalamaya göre simetriktir. Dolayısıyla eğrinin bir tarafında kalan alan tüm alanın yarısına eşit olup bu aynı zamanda % 50 olasılığa da eşittir. Dolayısıyla bir dağılımın normal dağılıma ne kadar benzediğinin göstergelerinden biri olan çarpıklık (skewness) kat sayısı normal dağılımda 0'a eşittir. Ancak pratikte de bu değer çoğunlukla tam 0 değil ancak 0'a yakın bir sayıdır. Ayrıca sola çarpık dağılımlarda da bu değer $-$, sağa çarpıklarda ise $+$ çıkar.

Simetrik dağılımların sayısı oldukça fazla olduğundan normal dağılımı diğerlerinden ayırmakta kullanılan diğer bir gösterge basıklık (kurtosis) katsayısıdır. Basıklık kat sayısı ise bir olasılık yoğunluk fonksiyonunun tepe noktasının yerini, dolayısıyla kuyruklarının kalınlığını yani ortalamanın her iki yanındaki dağılımların en büyük sapma değerini ölçer. Basıklık ve çarpıklık katsayılarının temel tipleri şekil 3.1 ve 3.2'da verilmiştir.



Şekil 3.1. Çarpık Dağılım Tipleri



Şekil 3.2. Basık Dağılım Tipleri

Çarpıklık katsayısı ve basıklık kat sayılarının hesabı (3.1) ve (3.2)'de gösterildiği gibidir.

$$\text{Çarpıklık Katsayısı} = \frac{\sum_{i=1}^n (x_i - \bar{x})^3 / n}{s^3} \quad (3.1)$$

$$\text{Basıklık Katsayısı} = \frac{\sum_{i=1}^n (x_i - \bar{x})^4 / n}{s^4} \quad (3.2)$$

(3.1) ve (3.2)'deki $\bar{x}$ örneklerin ortalamasını, s ise standart sapmasını göstermektedir. (3.1) ve (3.2)'deki ifadelerle dikkat edildiği takdirde varyans hesaplanırken ortalamadan sapmaların kareleri kullanılırken çarpıklık kat sayısında ise bu farkların küpü, basıklık kat sayısında ise bu farkların dördüncü kuvvetlerinin kullanıldığı kolayca fark edilebilir. Aslında aritmetik ortalama sıfıra göre 1. Moment, varyans aritmetik ortalamaya göre 2. Moment, çarpıklık kat sayısı aritmetik ortalamaya göre 3. Moment, basıklık katsayısı ise aritmetik ortalamaya göre 4. Momentten başka bir şey değildir.

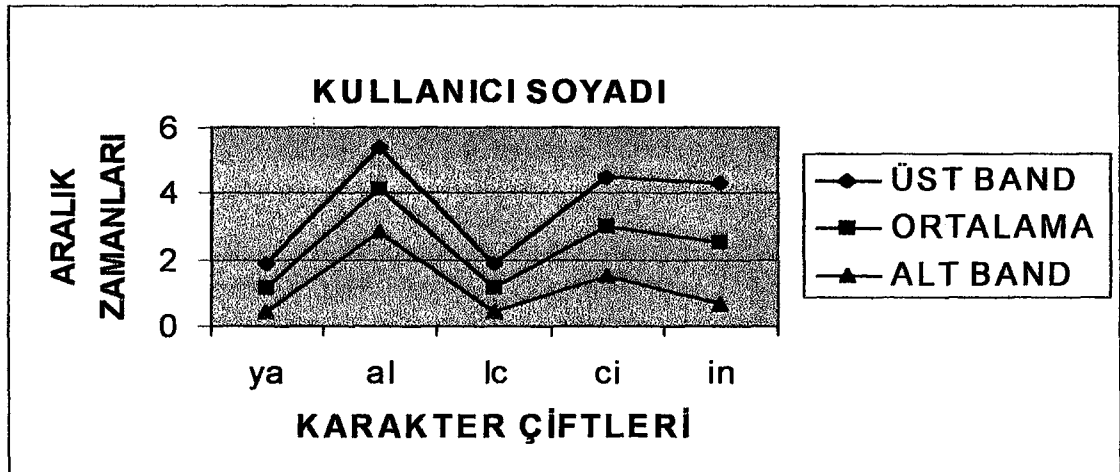
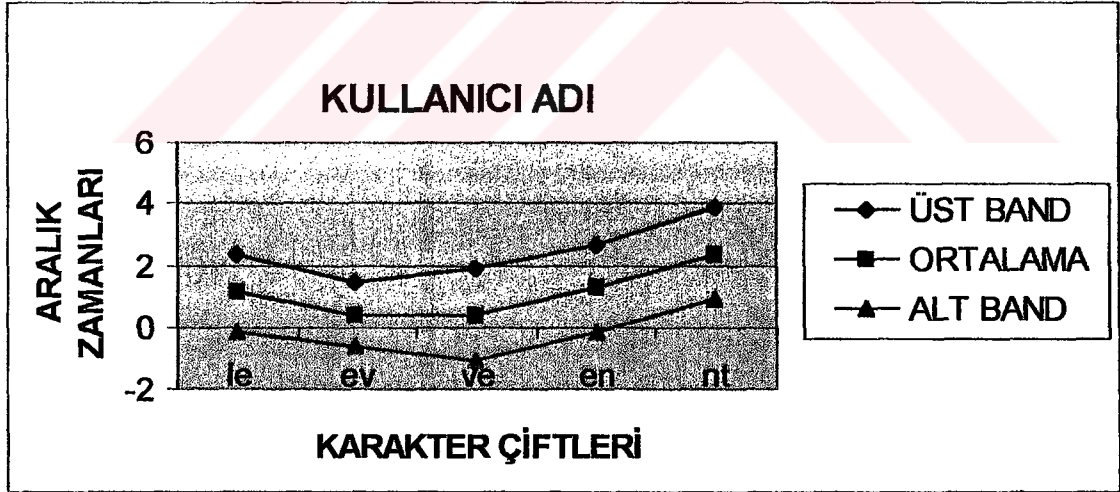
Kaldı ki bu tez çalışmasında karakterler arasında geçen zamanın normal dağılıma benzediği savından öte dağılımın basıklık ve çarpıklığı hesaba katılarak dağılımın aritmetik ortalaması ve standart sapması hesaplanmış ve ortalamadan 2 ila 3 standart sapmaya kadar deneysel çalışmalar yapılarak kullanıcı profilleri çıkarılmaya ve en uygun ve en iyi tanımayı sağlayacak eşik seviyesi (standart sapma katsayısı) tespit edilmeye çalışılmıştır.

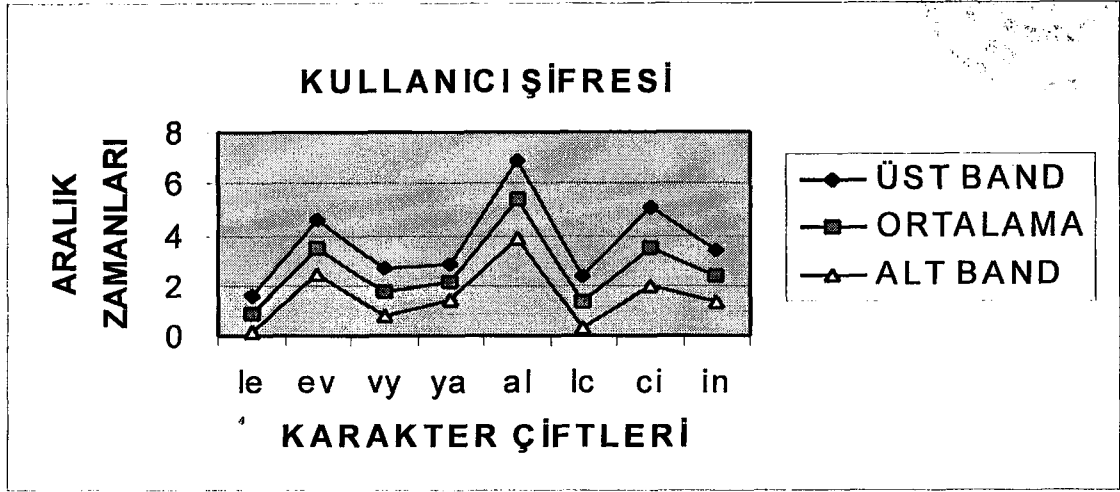
Birinci Algoritmanın dayandığı temel unsur şudur. Herkesin klavye üzerinde bazı karakter çiftlerine basmaları arasında geçen süre, özellikle ad ve soyadındaki bazı karakter çiftleri arası süre, hemen her sefer birbirine çok yakındır. Yani bazı karakter çiftlerine basış zamanlarının standart sapma değeri çok küçüktür. İşte algoritma özellikle bu karakter çiftlerini tam olarak taklit edemeyen yetkisiz kullanıcıların sistem girişine müsaade etmemektedir. Bunu bir örnekle açıklamaya çalışalım. Aşağıda adı 'levent' soyadı 'yalcin' ve şifresi 'levyalcin' ve sisteme kayıtlı olan kişinin karakter çiftlerine göre ad, soyad ve şifresi yazarkenki aralık değerleri ve buna ilişkin grafikler verilmiştir.

Tablo 3.1. Bir Kullanıcının Klavyeye Veri Girişi Sonundaki Ortaya Çıkan Zamanlayıcı (Timer) Değerleri

ADI	l	e	v	e	n	t
1. Veri Girişi	1.000	0.000	1.000	2.000	2.000	
2. Veri Girişi	2.000	1.000	0.000	1.000	3.000	
3. Veri Girişi	1.000	0.000	2.000	0.000	3.000	
4. Veri Girişi	1.000	1.000	0.000	1.000	3.000	
5. Veri Girişi	1.000	0.000	0.000	1.000	2.000	
6. Veri Girişi	2.000	0.000	0.000	2.000	2.000	
7. Veri Girişi	1.000	0.000	0.000	1.000	3.000	
8. Veri Girişi	0.000	1.000	0.000	2.000	1.000	
ORT	1.125	0.375	0.375	1.250	2.375	
ST SAPMA	0.640	0.517	0.744	0.707	0.744	

ORT + 2*ST	2.305	1.409	1.863	2.664	3.863	
ORT - 2*ST	-0.150	-0.650	-1.110	-0.160	0.887	
SOYADI	y	a	l	c	i	n
1. Veri Girişi	1.000	5.000	2.000	2.000	4.000	
2. Veri Girişi	2.000	4.000	1.000	2.000	2.000	
3. Veri Girişi	1.000	5.000	1.000	4.000	3.000	
4. Veri Girişi	1.000	4.000	1.000	3.000	3.000	
5. Veri Girişi	1.000	3.000	1.000	3.000	2.000	
6. Veri Girişi	1.000	4.000	1.000	3.000	1.000	
7. Veri Girişi	1.000	4.000	1.000	4.000	3.000	
8. Veri Girişi	1.000	4.000	1.000	3.000	2.000	
ORT	1.125	4.125	1.125	3.000	2.500	
ST SAPMA	0.353	0.640	0.353	0.755	0.925	
ORT + 2*ST	1.831	5.405	1.831	4.510	4.350	
ORT - 2*ST	0.419	2.845	0.419	1.490	0.650	
ŞİFRESİ	l	e	v	y	a	l
1. Veri Girişi	1.000	3.000	2.000	2.000	6.000	
2. Veri Girişi	1.000	4.000	1.000	2.000	5.000	
3. Veri Girişi	0.000	3.000	1.000	3.000	4.000	
4. Veri Girişi	1.000	4.000	2.000	2.000	6.000	
5. Veri Girişi	1.000	4.000	2.000	2.000	6.000	
6. Veri Girişi	1.000	3.000	2.000	2.000	5.000	
7. Veri Girişi	1.000	4.000	2.000	2.000	5.000	
8. Veri Girişi	1.000	3.000	2.000	2.000	6.000	
ORT	0.875	3.500	1.750	2.125	5.375	
ST SAPMA	0.353	0.534	0.462	0.353	0.744	
ORT + 2*ST	1.581	4.568	2.674	2.831	6.863	
ORT - 2*ST	0.169	2.432	0.826	1.419	3.887	





Şekil 3.3. Bir Kullanıcının Öğrenme Safhasındaki Karakter Çiftleri Arası Basış Zamanları

Bugüne kadar yapılan çalışmalarda ise iki karakter arasında geçen geçiş zamanının (keystroke latency) yeterli veri girişi olduğunda normal dağılıma çok yakın olduğu tespit edilmiştir. [22] [25] Kaldı ki yukarıdaki tabloda dikkat edilirse veri girişinin azlığına rağmen gerçekten de karakter çiftleri arasında geçen sürelerin normal dağılıma yakınsadığı kolaylıkla fark edilebilir. Ayrıca yukarıdaki tabloda koyu renkle gösterilen hücrelere dikkat edildiği takdirde ise bahse konu kişinin adında 1 adet, soyadı ve şifresinde ise 2'şer adet olmak üzere toplam 5 adet aralık zamanının ortalamasının ± 2 standart sapması dışında olduğunu görebiliriz. Diğer yandan bu 5 adet aralık zamanının 4 adetinin özellikle sapma miktarının çok az olduğu karakter çiftleri nedeniyle ortaya çıktığı rahatlıkla görülebilir. Sapma katsayısı 2.5 veya daha yukarıya çıkarıldığı takdirde ise bu miktarlar daha da azalacaktır.

Yetkisiz kullanıcıların yetkili kullanıcı yerine girebilmesi için tüm karakter çiftlerinde Şekil 3.3.'den de anlaşılacağı üzere yetkili kullanıcının bandı içinde kalacak şekilde yazma ritmini ayarlaması gerekmektedir. Yukarıdaki paragraflarda da izah edildiği gibi yetkisiz kullanıcıların özellikle bu 5 adet gri renkle belirtilen karakter çiftlerini taklit edebilmesi oldukça küçük bir olasılıktır.

Birinci algoritmanın matematiksel formülü ise aşağıdaki gibi ifade edilebilir.

n karakter sayısı ve t deneme girişi sayısı (öğrenme safhasındaki) olmak üzere

i = 1,2, ... , n

j = 1,2, ... , t

$$m_i = \frac{\sum_{j=1}^t R_{ij}}{t} \quad (3.3)$$

$$\sigma_i = \left(\frac{1}{t} \sum_{j=1}^t (R_{ij} - m_i)^2 \right)^{1/2} \quad (3.4)$$

$$m_i - SL * \sigma_i \leq X_i \leq m_i + SL * \sigma_i \quad (3.5)$$

(3.3), (3.4) ve (3.5) de

X_i : Test (Sistem Girişi) Vektörü

R_{ij} : Referans Vektörü

m_i : Ortalama Vektörü

σ_i : Standart Sapma

SL : Güvenlik Seviyesi dir.

Herbir karakter çifti (diagraph) için test girişlerindeki değerlerin formül (3.5)'deki aralık içinde olup olmadığına bakılarak sistem girişine müsaade edilir/edilmez.

Birinci ve ikinci algoritmaları, tasarlanan sistemin safhalarına göre izah etmeye çalışırsak **öğrenme safhasında** her iki algoritmada da kullanıcıların adlarını, soyadlarını ve şifrelerini yazarken girdiği karakterlere basma (tuş üzerinde kalma süresi) ve herhangi bir karakterden parmağın kaldırılmasından ikinci bir karaktere basana kadar geçen aralık zamanları ayrı tablolara kaydedilerek ortalama ve standart sapma bilgileri ile ikinci. algoritmada dağılımın basıklık ve çarpıklığı ile ilgili değerler hesaplanarak tablolara kaydedilmektedir. Daha basit bir şekilde ifade etmek gerekirse ortalama ve standart sapmalar hesaplanırken Tablo 3.1. deki değerlerden birinci algoritmada basma ve aralık zamanlarının düşey olarak, ikinci algoritmada ise yatay olarak ortalamaları ve standart sapmaları hesaplanmakta, diğer bir deyişle birinci algoritmada karakter çifti bazında deneme girişlerinin ortalamaları ve standart

sapmaları hesaplanırken, ikinci algoritmada herhangi bir denemeki aralık ve basma zamanlarının ortalamaları ile standart sapmaları ve daha sonra tüm denemelerin ortalamaları ile standart sapmaları hesaplanmaktadır. Ayrıca sisteme kendini tanıtan kişilerin ad,soyad ve şifre bilgileri kullanıcılar tablosuna kaydedilmektedir.

Kimlik tespiti safhasında her iki algoritmada da sisteme giriş yapmak isteyen kullanıcıların girmiş oldukları ad, soyad ve şifre bilgilerinin öğrenme safhasında sisteme kendini tanıtmış ve kullanıcı tablosuna atılan kişilerden birinin ad, soyad ve şifresine uyup uymadığının kontrolü yapılmakta ve uyuşum varsa doğrulama/gerçekleme safhasına geçilmekte, uyuşum yok ise sistemde böyle bir kullanıcının tanımlı olmadığı kullanıcıya bildirilerek kullanıcı ikaz edilmektedir.

Doğrulama/Gerçekleme safhasında ise kimlik tespiti safhasını geçen kullanıcıların, adına giriş yapılan kişinin gerçekten kendisi olup olmadığını ayırt etmek maksadıyla daha önce yapılan bir çok çalışmada uygulanan sabit eşik seviyesine göre karar vermek yerine ortaya çıkan standart sapmaya ve sistem yöneticisince değiştirilebilen değişken bir eşik seviyesine yani güvenlik seviyesine göre hesaplama yapılmaktadır. Sistem yöneticisinin bu eşik seviyesini tespitine yardımcı olmak üzere sistem girişlerinin logu tutulmakta, bu log tutulurken girişin yapıldığı tarih saat grubu, girişin başarılı olup olmadığı, başarılı ve başarısız giriş sayıları ve giriş yüzdeleri, giriş yapılırken ki sistem yöneticisince ayarlanmış olan sistem ayarları (güvenlik seviyesi, kullanıcı tanıma için gerekli giriş sayısı, hesap kapanma sayısı v.b.) ve testler sırasında FAR ve FRR hatalarının daha sağlıklı hesaplanabilmesi amacıyla sistem girişi yapan yetkili veya yetkisiz kişinin gerçek ad ve soyadı bilgileri tutulmaktadır.

Karar Verme/Güncelleme safhasında ise ortaya çıkan ve sistem yöneticisince belirtilen eşik seviyesine göre (ortalama +/- güvenlik seviyesi*standart sapma) giriş değerleri eşik seviyesini geçmiyorsa giriş yapan kişinin sistemde tanımlı kişi ile aynı kişi olduğuna karar verilerek sistem girişine müsaade edilmekte, geçiyorsa müsaade edilmemektedir. Sistem girişine müsaade edilen girişler ise başarılı giriş sayısı, öğrenme sayısı ve katlarına ulaştığında güncellenebilmektedir.

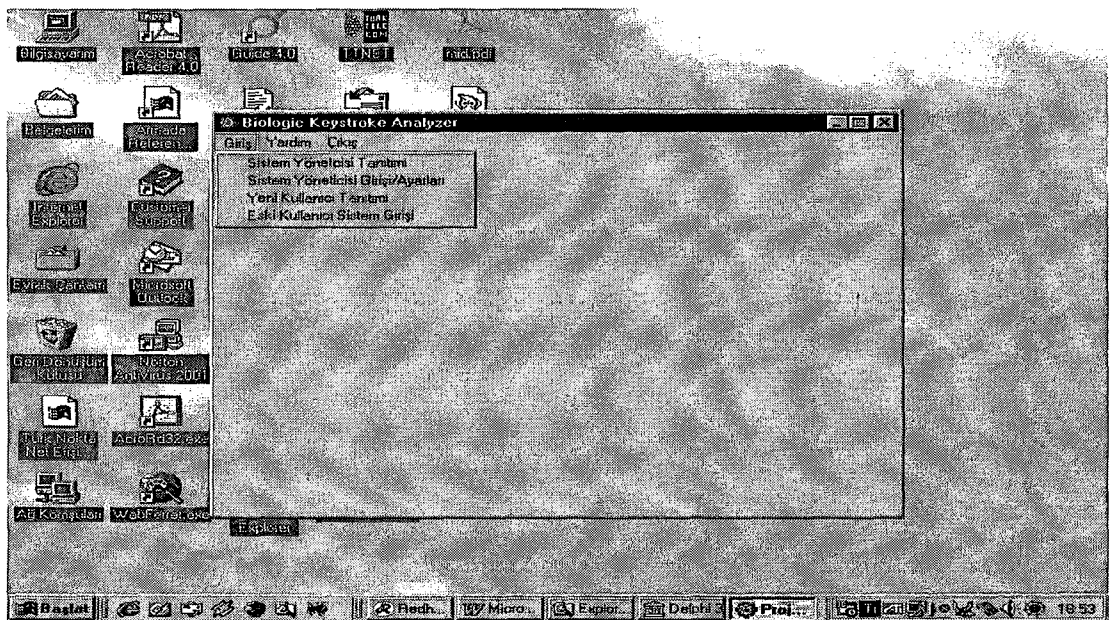
Sistem yöneticisinin sistem ayarlarından kullanıcı hesabı kitleme sayısını mümkün olduğunca düşük tutacağı varsayımıyla ise yetkisiz kullanıcının en çok 6 deneme civarında sisteme girebilmesi gerekmektedir

Burada daha önce değinildiği gibi deneysel sonuçlarla hesap kitleme sayısının seçilmesi oldukça büyük bir önem arz etmektedir. Çünkü ortalamadan ne kadar standart sapmanın hesaba katılacağı FRR hata oranını doğrudan etkilediğinden hesabı kitleme sayısının oldukça küçük seçilmesi durumunda gerçek kullanıcının da sisteme giriş yapmadan hesabının kilitlenmesi mümkün olacak ve uygulanması pek de mümkün olmayan bir sistem ortaya çıkacaktır.

3.2 Biyolojik Yazma Ritmi Analiz Etme Yazılımı Tanıtımı

Yazılımın temel unsurlarından birisi yazılımın yöneticisi ve kullanıcılarının ayırt edilmesidir. Bundaki amaç yazılımı yönetecek kişinin sistemin güvenliği ile ilgili bir takım ayarlarına müdahale edebilmesi ve ihtiyaç duyduğu ayarları yapabilmesidir.

Program ilk çalıştırıldığında aşağıda Şekil 3.4'deki gibi bir formla karşılaşılmakta ve yukarıda ki menüden giriş seçilerek sistem yöneticisi tanıtımı



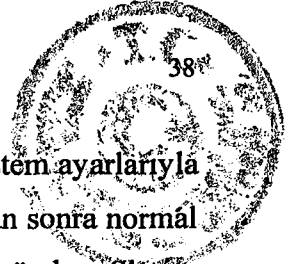
Şekil 3.4. Programın Ana Giriş Formu

37

anıtma imkanı
gibi bir form

Şekil 3.5. Sistem Yöneticisi Tanıtımı Formu

Şekil 3.6. Sistem Yöneticisi Tanıtıldıktan Sonraki Ana Formun Görünüşü



Sistem yöneticisi olarak kendini tanıtan bir kullanıcı ise artık sistem ayarlarıyla oynama yetkisine sahip olduğundan bir defa kendini sisteme tanıttıktan sonra normal olarak tasarladığı sistemin güvenlik ihtiyaçlarına göre giriş menüsünden Sistem Yöneticisi Girişi/Ayarlarına tıklamak suretiyle sistem ayarlarına erişmek üzere aşağıda Şekil 3.7’de belirtilen formdaki bilgileri doldurması beklenir.

Sistem Yöneticisi Girişi

1.Kısım: Lütfen Aşağıdaki Bilgileri Girip/Seçtikten Sonra 2. kısma geçiniz.

Kendi Adınız:

Kendi Soyadınız:

Güvenlik Seviyesi Seçini: ☐ 2 ☐ 2.25 ☐ 2.5 ☒ 2.75 ☐ 3

Başkasının Adına Giriş Yapacağınız Kişiyi Seçiniz:

Adı	Soyadı	Şifresi
Ievent	yalcin	yalcin

2.Kısım: Ad Soyad ve Şifreyi Girerek Deneme İşlemine Başlayınız.

Doğru Giriş Yüzdesi:	33,69
Doğru Giriş Yapılan Kayıt Sayısı:	31
Yanlış Giriş Yapılan Kayıt Sayısı:	61
Toplam Giriş Yapılan Kayıt Sayısı:	92

ADI

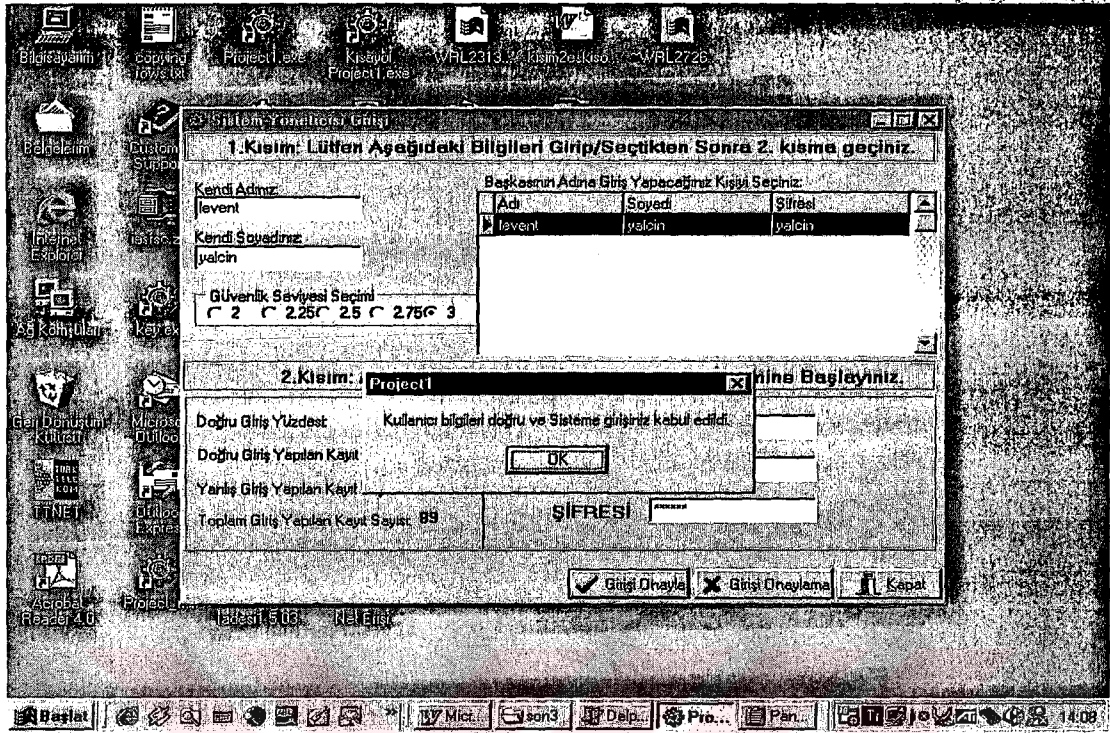
SOYADI

ŞİFRESİ

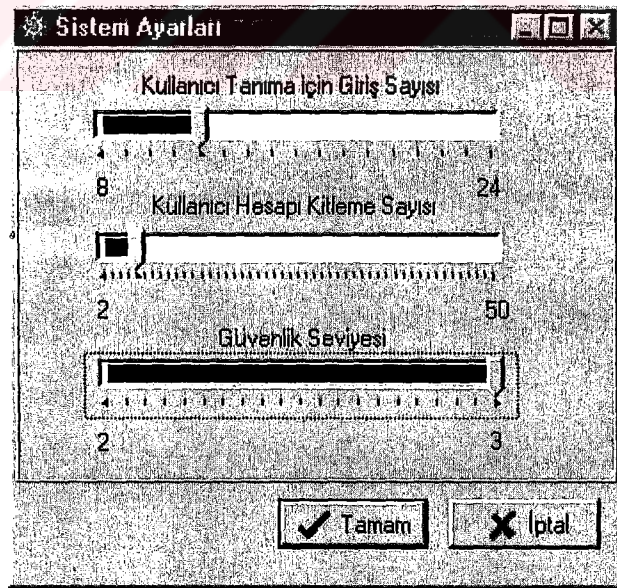
☒ Girişi Onayla ☒ Girişi Onaylama

Şekil 3.7. Sistem Yöneticisi Girişi Formu

Eğer yönetici bu menüye girmeyi istemez veya düşünmez ise sistem ayarları program tarafından varsayılan değerlerde kalacaktır. Ancak girmek isterse Şekil 3.7’de belirtilen formdaki bilgileri girerek sistem ayarlarına ulaşmak isteyen yöneticiye Şekil 3.8. deki mesajla Şekil 3.9.daki sistem ayarlarını değiştirebilme imkanı tanınır. Sistem Yöneticisinin diğer kullanıcılarda olduğu gibi önce kimlik tespiti ve ardından yönetici tanıma ve gerçekleştirme işlemi program tarafından yapılır.



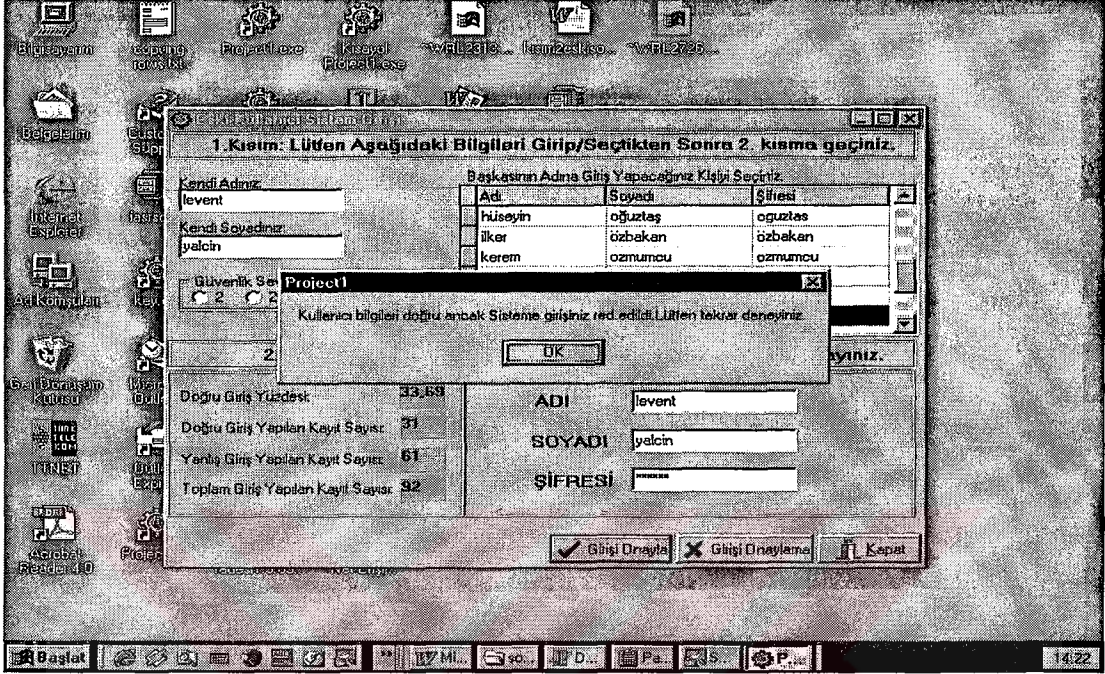
Şekil 3.8. Sistem Yöneticisine Sistem Ayarları Öncesi Verilen Mesaj



Şekil 3.9. Sistem Ayarları Formu

Sistem ayarları üzerinde değişiklik yapmak isteyen yetkisiz kullanıcının biometrik özellikleri sistemde tanımlı yöneticinin biometrik özellikleri ile uyuşmuyorsa daha doğrusu sistem yöneticisi biometrik özelliklerinin belirlenen

sınırları içinde değilse yetkisiz kullanıcının sistem yöneticisi ayarlarını değiştirmesine müsaade edilmez ve Şekil 3.10. deki mesajla kullanıcı uyarılır.



Şekil 3.10. Sistem Ayarlarının değiştirilemeyeceğine ilişkin mesaj

Şekil 3.8’de dikkat edileceği üzere sistem yöneticisine kullanıcıların sistem tarafından tanınabilmesi için gerekli giriş sayısı ile sisteme sistem yöneticisi tarafından girilen sayıda üst üste girememme durumunda kullanıcı hesabının kilitlenmesine ve güvenlik seviyesine ilişkin ayarını değiştirme yetkisi verilmektedir. Kullanıcı tanıma için Giriş Sayısı ayarına programda 8 ila 24 arasında değişen değer girme imkanı tanınmış olup tahmin edileceği gibi burada seçilen değerlerin yüksek olması sistemin kullanıcıları daha iyi tanıması anlamına gelmektedir. Çünkü ortalama, standart sapma, varyans v.b istatistik değerler örnek veri büyüklüğü ne kadar fazlaysa o kadar sağlıklı sonuçlara götürmektedir. Ancak bu sistemde daha önce yazma ritmi üzerine yapılan bir takım çalışmaların aksine bu sayı makul değerler içinde tutularak kullanıcıların sisteme kendisini birkaç dakika içinde tanıtabilme imkanı getirilmiş, böylelikle kullanıcıların sisteme göstereceği rezistans azaltılmak suretiyle sistemin uygulanabilirliği artırılmaya çalışılmıştır. Nitekim daha önce yapılan kimi çalışmalarda sistemin kullanıcıyı tanıması için oldukça uzun bir süre veri girişi yapması beklenmiş ve bu durum konu üzerinde ciddi bir handicap olarak ortaya çıkmıştır. Bu şekilde bir tanıma ancak kullanıcıların kendilerinin normal işlerini yaparken ki ritimlerinin farkında olmadan çözümlenmesi şeklinde

yapılabilir. Ancak bu konu üzerinde de kişilerin bilgileri dışında bir takım özelliklerinin tespit edilmesinin ahlaki ve kanuni açıdan bir takım sakıncaları olduğunu söyleyenler de oldukça fazladır. Bu nedenle bu konu, üzerinde ciddi bir şekilde durulması ve düşünülmesi gereken bir konudur. Konu üzerindeki çekinceler çözüldüğü takdirde örneğin kişilerin yazma ritimleri ofis içindeki belgelerini yazarken veya e-mail'lerini yazarken tespit edilebilir.

Ancak şu ana kadar yapılan çalışmalarda serbest şekilde yazılan metinlerden belli metin veya bilgilerin yazılması suretiyle alınan yazma ritmi bilgilerinin çözümlenmesinin çok daha zor olduğu yapılan deneysel çalışmalarda tespit edilmiştir.

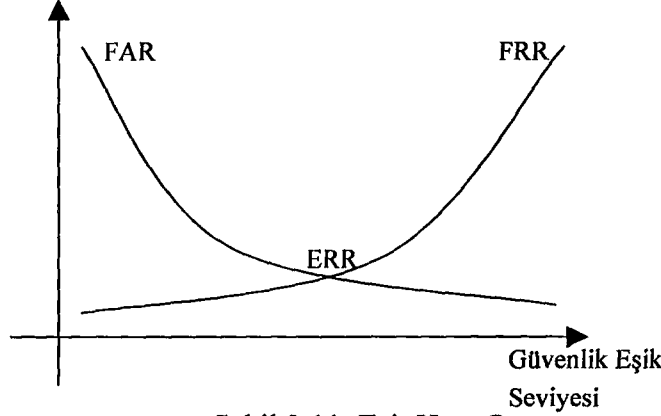
Yine yapılan bu çalışmalarda kullanıcıların özellikle ad ve soyadlarını kendi yazma ritimlerine en yakın şekilde yazdıkları metin kalıpları oldukları belirlenmiştir. Diğer yandan birçok kişinin herhangi bir sisteme girerken kullandığı şifrelerini, ezberlediğinden ve çok defa aynı metni girdiğinden dolayı tereddüt etmeden kendi yazma ritmine oldukça yakın bir şekilde yazacağı söylenebilir.

Bu nedenle bu tez çalışmasında da kullanıcıların ad, soyad ve şifre bilgilerinin sistem yöneticisi tarafından belirlenen miktarda girilmesi suretiyle kişilerin tanınması mümkündür. Öyle ki sistem birkaç dakika içinde kullanıcıyı tanıır hale gelmektedir.

Kullanıcı hesabı kitleme sayısına gelince, seçilen değer ne kadar küçükse yetkisiz kullanıcıların işinin o kadar zorlaşacağı yani sistemin oldukça güvenilir hale geleceği ancak bu durumda da gerçek kullanıcıların sisteme girmeden hesabının kilitlenmesinin mümkün olabileceği aşikardır.

Dolayısıyla her iki parametrenin de ve özellikle "Kullanıcı Hesabı Kitleme" sayısının sistem yöneticisi tarafından belirlenmesinin büyük bir önem arz ettiği görülmektedir. Sistem yöneticilerine değerleri belirli bir istatistiki bilgiye dayanarak verebilmesi açısından veri tabanında çeşitli düzenlemeler yapılmıştır. Örneğin sistem yöneticisine kullanıcı girişleri başarılı yada başarısız olsun kullanıcıların kaydını (logunu) ve hatta başarısız giriş yüzdesini veya sistem kilitlenme sayılarını görme ve tüm kullanıcılar için bu değerlerin ortalamalarını alma gibi imkanlar tanınarak bahse konu parametreleri eldeki verilere göre daha sağlıklı verebilme imkanı getirilmiştir. Öyle ki sistemin tasarımı yapılacak değişikliklerle tüm kullanıcılar için bu parametrelerin ayrı olarak belirlenmesi de mümkün

görülmektedir. Böylelikle kişilere özel bir şekilde FAR ve FRR hata oranlarını ayarlamak ve her iki hatanın Şekil 3.11’de belirtilen kesim noktaları olan ERR (Equal Error Rate) yani eşit hata oranlarını ayarlamak da söz konusu olabilecektir.



Şekil 3.11. Eşit Hata Oranı

Çünkü yalnızca bu noktada sistemin çok önemli iki hatası olan FAR ve FRR eşit olmakta, diğer tüm noktalarda bir hata büyürken diğeri küçülmektedir. Denilebilir ki iki sistemden ERR’si daha küçük olan diğerine nazaran daha iyi bir sistemdir. Ancak unutulmaması gereken bir diğer konuda seçilen veya uygulanan sistemin özelliğine göre FAR’ın büyük, FRR’nin ise çok küçük olması veya tam tersi FRR’nin büyük ama FAR’ın çok küçük olması arzulanan bir durum olabilir. Nitekim yazma ritmi tanıma sisteminde ve tüm biometrik sistemlerde önemli olan FAR’ın yeterince küçük tutabilmesidir. Dolayısıyla FRR çok büyük olmamak koşuluyla FAR’nin mümkün olduğunca küçük olmasını sağlayacak sistemlerin, biometrik sistemlerin genel amacı olduğu söylenebilir.

Tasarlanan sistemde sistem ayarları için varsayılan olarak kullanıcı tanımı için giriş sayısı 12, kullanıcı hesabı kitleme sayısı 6 ve güvenlik seviyesi 3 olarak belirlenmiş olup bundaki amaç sistem yöneticisinin biometrik özelliğinin mümkün olduğunca fazla veriye dayanarak tespit edilmeye çalışılmasıdır . Kullanıcı hesabı kitleme sayısının ise başlangıçta çok küçük tutulmayarak sistem yöneticisinin hesabı kitlenmeden sistem ayarlarını değiştirebilmesine imkan tanınmaya çalışılmıştır.

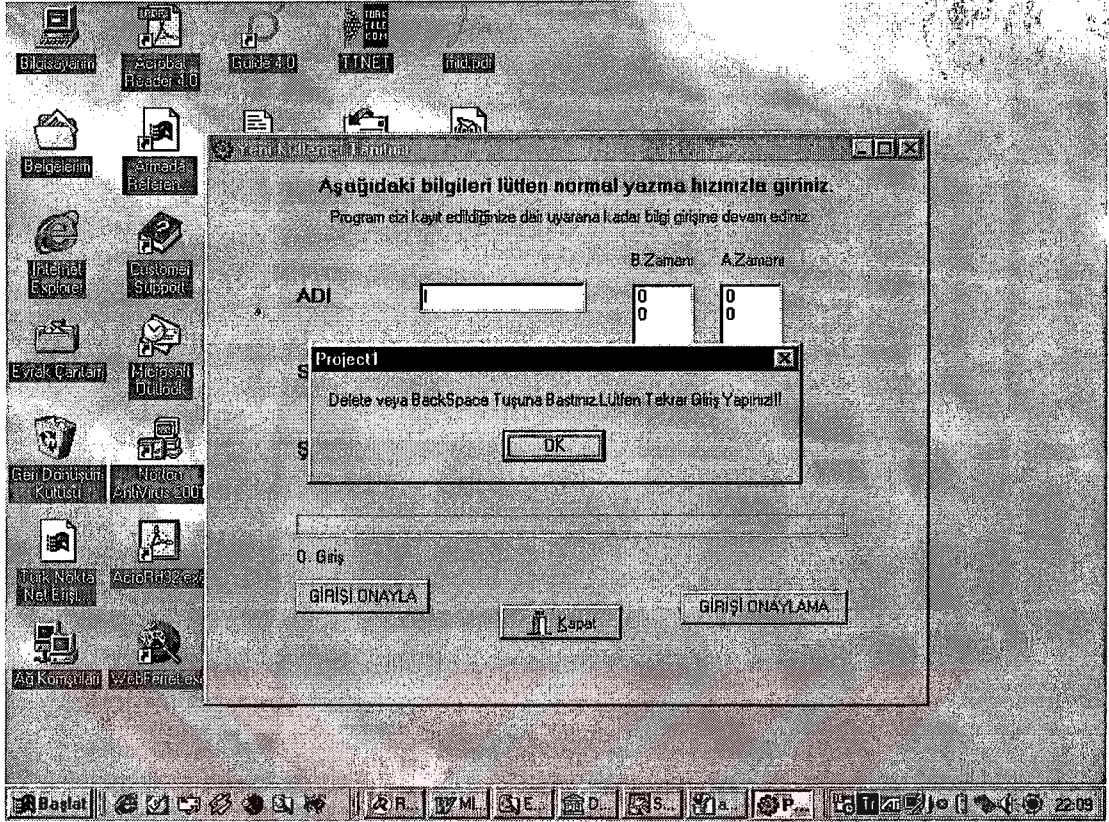
Sisteme sistem yöneticisinin tanıtılıp, sistem yöneticisi tarafından sistem ayarları yapıldıktan sonra sisteme dahil olacak kullanıcılar için Giriş menüsündeki

Yeni Kullanıcı Tanıtımına tıklanmak suretiyle yeni kullanıcıların sisteme tanıtılması mümkün olmaktadır. Yeni kullanıcıların sisteme kendini tanıttacağı form şekil 3.12’de gösterilmiştir.

Şekil 3.12. Kullanıcıların Kendilerini Sisteme Tanıtma Formu

Dikkat edileceği üzere bu formun sistem yöneticisi tanıtımı formundan bir farkı yoktur. Ancak gerçek uygulamada basma ve aralık zamanlarının görülebildiği seçme kutuları görülmemesi gerekirken, kullanıcıların ve özellikle sistem yöneticisinin tuşlar arası aralık zamanı ile tuşlara basma zamanlarını kendilerinin de görerek veri giriş değerlerini aynı zamanda kendilerinin de görerek girişi onaylayıp onaylamaması imkanı tanınarak kendilerini daha düzgün sisteme tanıtılma imkanı tanınmaya çalışılmıştır.

Kullanıcılar kendini sisteme tanıtırken ve sistem girişleri esnasında bir takım kontroller yapılmaktadır. Bunlardan biri backspace veya delete tuşlarına basılma kontrolüdür. Ad, soyad veya şifre girişi yapılırken bu tuşlardan birine basıldığında program yapılmakta olan veri girişini iptal ederek Şekil 3.13’deki mesajla kullanıcının aynı veriyi tekrar girmesini ister.

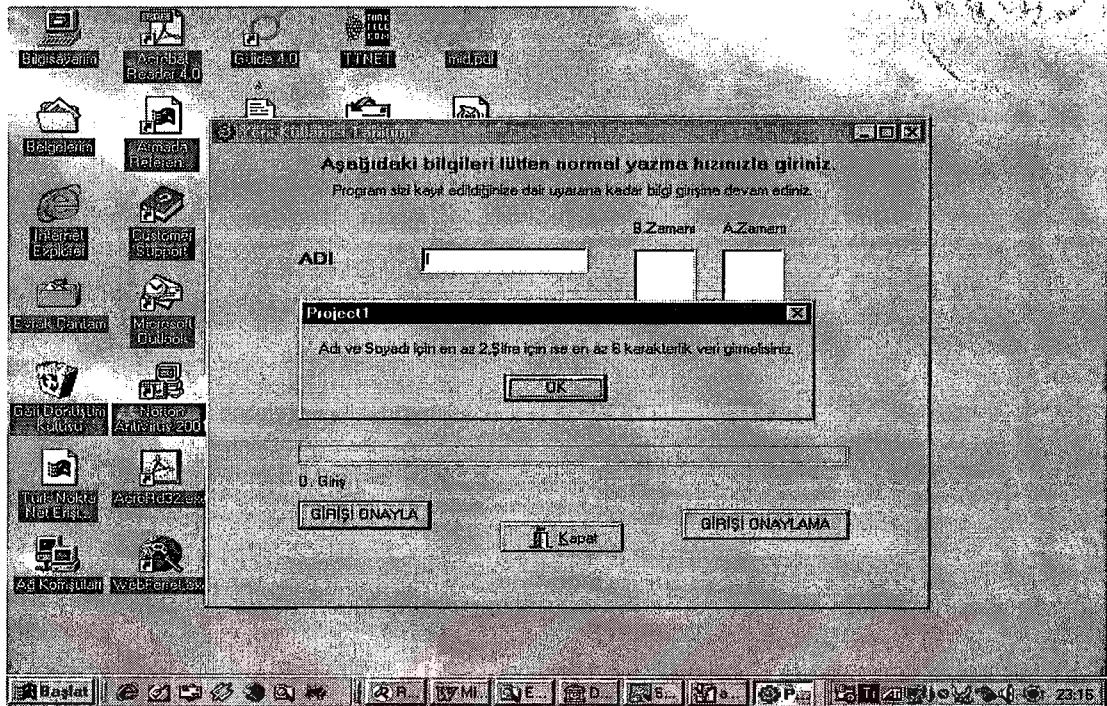


Şekil 3.13. Delete veya Backspace Tuşlarına Basıldığı Mesajı

Backspace ve delete tuşlarına basıldığının kontrol edilmesinde ki amaç bu tuşlara basıldığında kullanıcının ad, soyad veya şifreyi girerkenki normal yazma ritminin bozulmasıdır. Böylece mümkün olduğunca kullanıcıların gerçek yazma ritimleri yakalanmaya çalışılmaktadır.

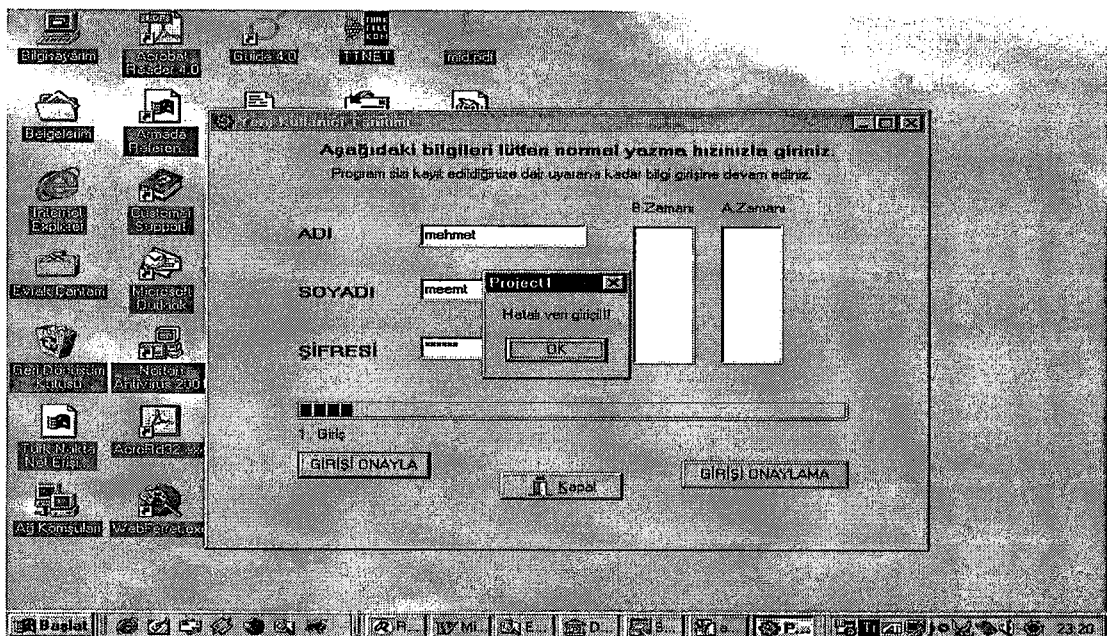
Veri girişindeki diğer kontrollerden biri de her veri alanına en az iki karakter ve şifreye en az 6 karakterli bir şifre girilmesinin kontrol edilmesidir.

Bu zorunlu kontrollerden biridir çünkü ad veya soyadına iki karakter girilmediği takdirde ad veya soyadının yazma karakteristiğini çıkarmak mümkün değildir. Şifrenin 6 karakterden az girilmemesinin sağlanması altında yatan birinci neden, ad veya soyad'ı kısa olan kişilerde o kişinin biometrik özelliklerinin yeterli seviyede çıkarılamama riskini engellemek, ikinci neden ise yetkisiz kullanıcıların yetkili kullanıcıların şifresini kolayca deşifre etmelerini engellemektir. İstenilen karakter adetleri girilmediği takdirde kullanıcı Şekil 3.14. deki gibi bir mesajla uyarılacaktır.



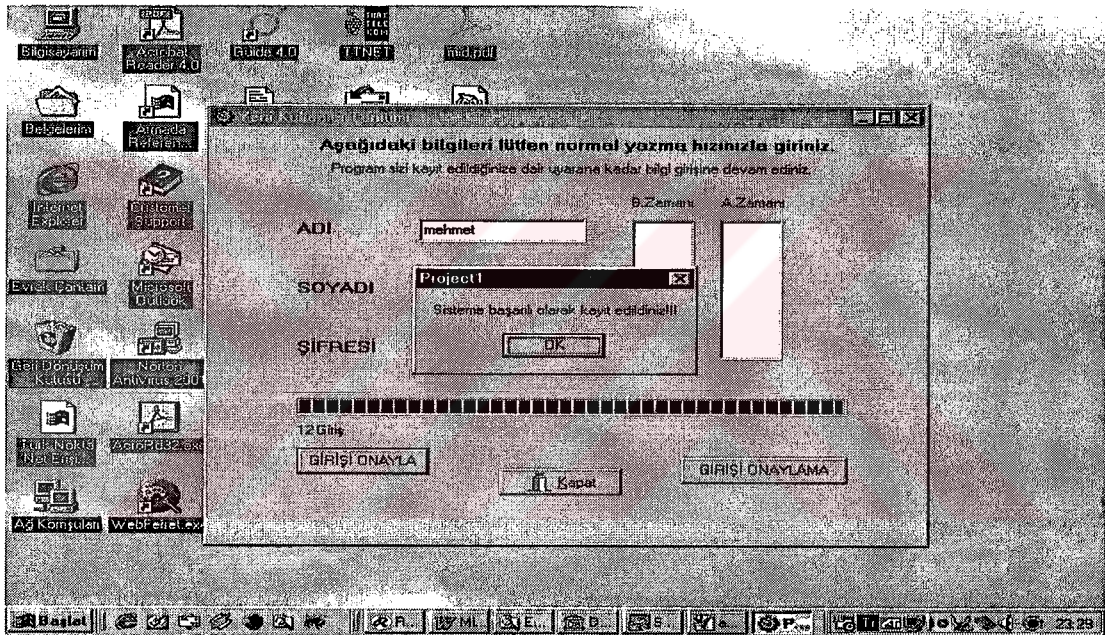
Şekil 3.14. Yeterli Sayıda Karakter Girilmediği Mesajı

Veri girişindeki bir diğer kontrol ise de birinci girişten sonraki girişlerde birinci girişteki ad, soyad veya şifreden herhangi birisi yanlış girilirse kullanıcının bu durumdan Şekil 3.15. deki mesajla haberdar edilmesidir. Böylelikle herhangi bir şekilde veri bütünlüğünün bozulmasına engel olunmaktadır.



Şekil 3.15. Hatalı Veri Girişi

Normal şartlarda veri girişinde herhangi bir problemi olmayan kullanıcı “Giriş Onayla” butonuna basmak suretiyle progress bar’ın bir ileri gitmesine neden olur. Progress bar en son sistem yöneticisi tarafından girilen “Kullanıcı Tanıma için Giriş Sayısı” değerine ulaştığında Şekil 3.16’de gösterilen mesajla kullanıcı sisteme tanıtılmış olur.



Artık bu kullanıcı yerine başka bir kullanıcının sistem üzerinde verilen yetkiler, haklar doğrultusunda sisteme erişme imkanı çok zorlaşmaktadır.

Sisteme kendini tanıtan kullanıcı ise bundan sonraki sistem girişlerinde ise giriş menüsünden Eski Kullanıcı Sistem Girişini seçmek suretiyle Şekil 3.17’de gösterilen forma ulaşır.

Adı	Soyadı	Şifresi
arif	aalcar	aalcar
bahadır	aytaç	757575
deniz	köse	hellgate
erol	yıldırım	yıldırım
ertan	güneş	egüneş
hüseyin	oğuztaş	oguztas

Doğru Giriş Yüzdesi:	0
Doğru Giriş Yapılan Kayıt Sayısı:	0
Yanlış Giriş Yapılan Kayıt Sayısı:	14
Toplam Giriş Yapılan Kayıt Sayısı:	14

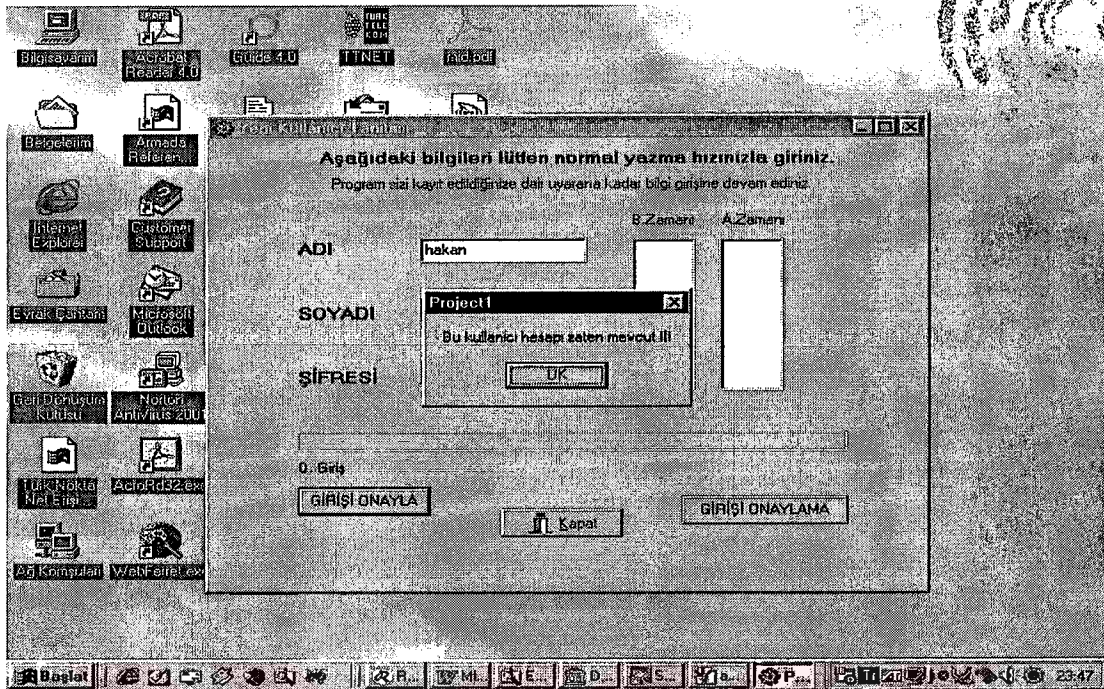
ADI	
SOYADI	
ŞİFRESİ	

☒ Girişi Onayla ☐ Girişi Onaylama

Şekil 3.17. Eski Kullanıcı Sistem Girişi Formu

Burada sistem yöneticisi girişinde olduğu gibi giriş esnasında oluşan kullanıcının biometrik özelliği sistemde tanımlı kullanıcının belli bir eşik seviyesi altındaysa sisteme giriş müsaadesi verilir, değilse verilmez. Biometrik özellikleri birbirini tutmuyan ancak kullanıcı bilgileri doğru olarak kimlik tespit etme safhasından geçen kullanıcıya Şekil 3.10’de görülen mesajla kullanıcı bilgilerinin doğru olduğu ancak sistem girişine müsaade edilmediği kullanıcıya bildirilir. Biometrik özellikleri tutan kullanıcıya ise Şekil 3.8. de görülen mesajla Kullanıcı Bilgilerinin doğru olduğu ve sistem girişine müsaade edildiği bildirilir.

Eğer kullanıcı daha önce var olan bir kullanıcı hesabıyla kendini sisteme tanıtmak isterse ise Şekil 3.18.’de görülen mesajla uyarılır.



Sistem girişlerinde ise sistemde tanımlı olmayan bir kullanıcı adına giriş yapılmak istendiğinde ise Şekil 3.19.'da ki mesajla kullanıcılar uyarılır.



4. UYGULAMA SONUÇLARI VE YORUMLAMA

Deneysel çalışmalarda biyometrik sistemlerin temel test ölçütlerinden olan FAR ve FRR, program algoritmasının değişik parametre değerleri için denenerek test edilmiştir. Biyometrik sistemlerde sistemlerin güvenliğini sağlamak birinci amaç olduğundan FRR makul seviyede olmak koşuluyla FAR'ın olabildiğince küçük olması arzulanan bir durumdur.

Yukarıda bahsedilen amaç doğrultusunda ve konu ile ilgili daha önce yapılan çalışmalarda ortaya çıkarılan karakter çiftleri arası dağılımın normal dağılıma benzediği ve normal dağılımda örneklerin % 68.26'sının ortalamanın ± 1 standart sapma, % 95.4 ' ünün ± 2 standart sapma ve , % 99.7'sinin ise ± 3 standart sapma içinde olduğu gerçeğinden hareketle ortalamanın 2 ila 3 standart sapma arasındaki FAR ve FRR değerleri hesaplanmıştır. Örneklem büyüklüğünün tespiti için bulunan dokümandaki [29] formüllere göre yapılan hesapta seçilen örnek büyüklüğünün tüm uzayı temsil edebilmesi için % 1 güven sınırına göre en az 2000 kişi üzerinde deneylerin yapılması gerektiği belirlenmiş ancak bu kadar çok kişiden veri toplama imkanı olmadığından tespit edilen rakamlara çok uzak olmakla birlikte tüm uzayı temsil edebileceği değerlendirilen 25 kişi sisteme tanıtılmıştır. Ancak sistem girişleri sırasında bu 25 kişinin haricinde 15 kişiye daha sisteme tanımlı diğer kişiler adına giriş denemeleri yaptırılmıştır.

Test parametrelerinin referans parametreye benzerliğinin ölçütü olarak kullanılan eşik seviyesi yani standart sapma katsayısı (k) ne kadar küçükse FAR'ın o kadar küçük yani yetkisiz kullanıcıların sisteme girme olasılığının o denli zor, standart sapma katsayısı ne kadar büyükse FRR'ın o kadar küçük yani yetkili kullanıcıların sisteme girememe olasılığının o denli küçük olduğunu söylemek mümkündür. Deneysel çalışmada kullanıcılara önce varsa kendi kullanıcı hesaplarına girmeleri, daha sonra ise sistemdeki diğer kullanıcıların ad, soyad ve şifreleri program aracılığıyla sunulurken diğer kullanıcıların hesaplarına girmeleri istenmiştir. Kullanıcıların seçiminde ofislerde sürekli yazı yazan kişiler değil bilgisayara uzak olmamakla birlikte bilgisayar klavyesini yoğun olarak kullanmayan kişilerinde seçilmesine gayret edilmiştir.

Genel olarak uygulanan iki algoritmanın deneysel çalışmaları neticesinde belirlenen farkları şunlardır ;

2. Algoritmada kullanıcılar kendi adına girmekte pek zorlanmazken 3 kullanıcının hesaplarına diğer bazı kullanıcılarca girilebilmiştir. Bu algoritmada kullanıcılar kendi hesaplarına standart sapma katsayısı yani güvenlik seviyesi 3 iken ortalama 1.4 girişte girebilmişler ve diğer kullanıcılar adına girebilen 5 kullanıcı ise sistemdeki bahse konu 3 kullanıcı adına ortalama % 15.6 oranında girebilmişlerdir. Bu algoritmada en iyi FAR oranları standart sapma katsayısı yani güvenlik seviyesi 1.5 iken elde edilmiş ancak bu durumda da gerçek kullanıcıların kendi adlarına girişleri oldukça zorlaşmıştır. Kullanıcıların kendi adına pek zorlanmadan girdikleri ve nispeten yetkisiz kullanıcıların sisteme girmesini engelleyen optimum güvenlik seviyesi 2 olarak tespit edilmiştir. Genel olarak bu algoritmadan arzulanan sonuçlar elde edilememiş olup bu durum algoritmanın bir nevi kişilerin ortalama klavyeye yazış hızlarına bakmasından kaynaklandığı değerlendirilmektedir. Bu algoritmaya göre en iyi FAR değeri güvenlik seviyesi 1.5 iken ortalama % 8.2 olarak tespit edilmiştir.

1. Algoritma da ise kullanıcılar güvenlik seviyesi 2.5'e kadar kendi adlarına çok zor girebilmişler, güvenlik seviyesi 2.5, 2.75 ve 3 iken kullanıcılar kendi adlarına ortalama sırasıyla 4.7, 2.5 ve 2.1 girişte girebilmişlerdir. Diğer kişiler adına girişlerde ise en iyi sınıflandırmaya güvenlik seviyesi 2.5 iken ulaşılabilmiş ancak bu seviyede kişilerin kendi adlarına girişleri (FAR) arzulanan seviyede olmadığı için optimum güvenlik seviyesi olarak 2.75 tespit edilmiştir. Optimum güvenlik seviyesi normalde sistem yöneticisince tespit edilmesi gereken bir değerdir ve sistem yöneticisince 0.05 oranında artırımlar mümkündür. Testler sırasında 0.25 aralıklarla güvenlik seviyeleri denenebildiğinden gerçek durumda optimum güvenlik seviyesi örneğin 2.6 olarak çıkabilir.

Test sonuçlarının daha rahat alınabilmesi bakımından testler/veriler önce Paradox for Windows veritabanında gerekli ayarlamalar yapılarak tek bir PC üzerindeki paylaşım açılan tabloların bulunduğu klasörün diğer kullanıcılara map edilmesi ile ağ sürücüsüne bağlanması suretiyle toplanmış ve 25 kullanıcının da öğrenme safhasındaki verilerinin toplandığı tablolar yeni hazırlanan setup seti içine

konulup kullanıcıların kendi makinalarına veri girişlerini kaydedeceği şekilde yeni setup oluşturulmuştur. Daha sonra her kullanıcının önce kendi adına belirli bir eşik seviyesinde daha sonra sistemde tanımlı diğer kişiler adına veri girişi yapması ve tüm çevrimi bitirdikten sonra bu denemeleri sırasıyla diğer eşik seviyelerinde de tekrarlaması istenmiştir. Böylece verilerin zaten derli toplu bir biçimde olması sağlanarak analizlerinin daha kolay ve kısa yapılması mümkün olmuştur.

Aşağıdaki tabloda Şekil 3.3'deki kullanıcı olan "Levent Yalçın" adına Güvenlik Seviyesi 2.75 iken testlere tabi tutulan 24 kişinin giriş denemesi istatistikleri gösterilmiştir.

Tablo 4.1. Kullanıcıların Bir Kullanıcı Adına Veri Girişi Denemeleri İstatistiği

Kullanıcı No :	Toplam Deneme Sayısı	Başarılı Olma Sayısı
1	100	0
2	100	0
3	100	0
4	112	0
5	92	0
6	100	0
7	100	0
8	100	0
9	104	0
10	106	0
11	100	0
12	100	0
13	100	0
14	96	0
15	100	0
16	100	0
17	100	0
18	100	0
19	102	0
20	105	0

Kullanıcı No :	Toplam Deneme Sayısı	Başarılı Olma Sayısı
21	100	0
22	100	0
23	94	0
24	100	0

Güvenlik Seviyesi 2.75 iken aynı kullanıcı ise kendi adına ortalama 3 girişte 1 defa girebilmiştir. Yukarıdaki tabloda dikkat edileceği üzere kullanıcılar oldukça yüksek sayılabilecek veri girişi yapmalarına rağmen kullanıcı “Levent Yalçın” adına hiç başarılı veri girişi yapamamışlardır. Elde edilen verilere göre bahse konu kullanıcı adına FRR % 0 tespit edilmiştir. Ancak sistemde tanımlı ve ad,soyad ve şifresi sırasıyla “Bahadır, Aytaç, 757575” olan kullanıcı adına ise toplam 6 kullanıcı başarılı olabilirken 1 kullanıcı bahse konu kişi adına ortalama 5 girişte bir başarılı giriş yapabilmektedir. Bu durumun ise bahse konu kullanıcının şifresinin çok kolay taklit edilebilir olması ve özellikle soyadının kısa olması ile adının klavyeye diziliş bakımından bir çok kişi tarafından rahat girilebilecek bir ad olmasından veya kullanıcının kendini sisteme tanıtmı esnasında dikkat dağılması benzeri bir nedenle bazı karakter çiftleri arasında gereksiz yere tereddüt ederek beklemesinden ve en önemlisi ise bahse konu kişinin oturmuş bir klavyeye yazma ritmi olmayan bir kullanıcı olmasından kaynaklandığı değerlendirilmektedir. Bunun haricinde ise diğer kullanıcılardan yalnızca birine daha 2 kişi tarafından ortalama % 1.5 oranında giriş yapılabilmektedir. Toplam 40 kişinin diğer kişiler adına yaptığı yaklaşık 4000 denemede sadece 25 denemede başarılı olunabilmiş ve genel olarak FAR oranı ise % 0.6 olarak ortaya çıkmıştır. Sadece ad,soyad ve şifresi sırasıyla “Bahadır, Aytaç, 757575” olan kullanıcı adına 20 defa girildiği düşünülüp bu kullanıcı test haricinde tutulduğu takdirde ise FAR oranı yaklaşık % 0.1’e inmektedir. Buna karşın genel olarak sistemin FRR oranı ise oldukça yüksek bir değer olarak % 60 olarak ortaya çıkmıştır. Diğer bir deyişle kullanıcılar kendi adlarına ortalama 2.5 girişte bir girebilmişlerdir. Ancak hesap kitlenme sayısının 5 veya 6’ya ayarlanması durumunda FRR’nin bu kadar büyük olmasının sisteme olumsuz bir etkisi olmayacaktır.

Yukarıda anlatılanlardan kullanıcılarca seçilen şifrelerin bir çok kişi tarafından rahatça girilebilecek bir şifre seçilmemesinin ve sisteme kullanıcıların kendini tanıtırken bazı karakter çiftlerini yazarken tereddüt ettiklerinde bu girişleri

onaylamamalarının ne kadar önemli olduğu açıkça ortaya çıkmaktadır. Ancak sistem yöneticilerince bu tür kişiler tespit edilerek bu kişilerin daha değişik bir şifre kullanmaları konusunda ikaz edilmesinin ve standart sapmaları fazla olan bu tür kişilerin standart sapma/ortalama ve standart sapma/değişim aralığı (max – min) oranları genel ortalamaya yaklaşıncaya kadar sistemin öğrenme safhasında daha çok veri girişi yaptırılmasının bahsedilen hataları bir miktar azaltacağı tahmin edilmektedir.

Ayrıca sistem yöneticisince ayarlanan güvenlik parametrelerinin (Öğrenme safhasındaki veri giriş sayısı, kullanıcı hesabı kitleme sayısı v.b.) tasarımı yapılacak değişikliklerle sistemdeki tüm kullanıcılar için tek değil, kullanıcıların herbirine göre ayrı olarak ayarlanabilmesinin sistemin genel performansını olumlu yönde etkileyeceği tahmin edilmektedir.

4.1. Diğer Yöntemler ile Karşılaştırma

Kaynakların (ref) veriliş sırasına göre bugüne kadar uygulanan çalışmalarla ilgili karşılaştırmalar elde edilebildiği kadarıyla aşağıdaki tabloda özetlenmiştir.

Tablo 4.2. Diğer Sınıflandırma Algoritmalarıyla Karşılaştırma

Ref No	Çalışmayı Yapanlar	Uygulanan Algoritma Özeti	Örnek Sayısı (Kişi)	FAR %	FRR %	Varsa Dezavantajları/ Sınırlamaları
1	Sungzoon Cho, Chigeun Han, Daa Hee Han, Hyung Il Kim	Çok Katmanlı Yapay Sinirsel Ağ	21	0	1	1. Sistemde çok kullanıcı tanımlı olduğunda her yeni kullanıcının sisteme tanıtılmasının çok zaman alması 2. Her Kullanıcı tanıtımında tüm sistemin tekrar öğrenme gereği
2	Aykut Güven	Vektörler arası açının hesaplanması	12	~1	-	1. Test edilen kişinin çok sınırlı olması 2. Şifrenin 8 karakter girilme zorunluluğu 3. Öğrenme safhasının çok

						zaman alabilmesi 4. Eşik Seviyesinin sabit olması 5. FRR'nin denenmemesi
4	R.Gaines,William Lisowski, S.James Press, Norman Shapiro	T-Test	7	?	?	1. Test edilen kişinin çok sınırlı olması 2. Konu ile ilgili ilk ciddi çalışma olması
5,6	Saleh Bleha, Charles Slivinsky, Bassam Hussein	Minumum Fark Sınıflandırması, Mahalanobis Sınıflandırması,	32	3.1	0.5	
7,8,9,10	Glen Williams, John Leggett ve David Umphress	Ortalama diagraph zamanının 0.5 standart sapması içinde kalan ve referans iz ile test izi arasında % 60'lık benzerlik olanların ayırt edilmesi	?	5.5	5	1.Öğrenme safhasının çok zaman alması 2. FRR'nin yüksek olması 3. Tuşlara basma ve tuşlar arası geçiş sürelerinin ayırt edilmemiş olması
11	J.Garcia	Mahalanobis Sınıflandırması	?	?	?	Patent alınmış bir çalışma olması nedeniyle herhangi bir neticesinin elde edilememiş olması
12	J.Young, R.Hammon	Minumum Fark Sınıflandırması	?	?	0	Patent alınmış bir çalışma olması nedeniyle detaylı bir neticenin elde edilememiş olması
13	Rick Joyce, Gupta Goyal	Ortalama referans vektörü ile test giriş vektörünün belli bir eşik seviyesine göre mukayesesi	?	16.36	0.25	FAR'ın yüksek olması
19,20	Fabian Monroe, Aviel Rubin	Minumum Fark Sınıflandırması	63	16.78	?	
19,20	Fabian Monroe, Aviel Rubin	Ağırlıksız Olasılık ile KNN	63	14.37	?	
19,20	Fabian Monroe, Aviel Rubin	Ağırlıklı Olasılık ile KNN	63	12.82	?	
19,20	Fabian Monroe, Aviel Rubin	Bayes Sınıflandırması	63	7.86	?	

23	Alen Peacock	Ağırlıklı K En Yakın Komşu Sınıflandırması	11	4.2	?	1. Test edilen kişinin çok sınırlı olması 2. Güncelleme yapılmaması
*	Levent YALÇIN, İbrahim SOĞUKPINAR	Ortalama diagraph zamanının 2.75 standart sapması içinde kalan test izi ile referans izinin bire bir incelenerek karar verilmesi	40	0.6	60	FRR'nin yüksek olması * Bu çalışma sonuçları

Bu tez çalışmasında uygulanan birinci algoritmanın yukarıdaki tabloda belirtilen ve bu tez çalışmasının yazımına kadar temin edilemeyen ancak diğer kaynaklardan elde edilen bilgilere göre John Leggett, Glen Williams, David Umphress ile Rick Joyce ve Gupta Goyal tarafından geliştirilen algoritmalara temelde benzediği ancak bir takım yöntem farklılıklarıyla implimente edildiği söylenebilir.

Geliştirilen algoritmanın K En Yakın Komşu Kuralı ve Yapay Sinirsel Ağ yaklaşımı algoritmalarıyla herhangi bir benzerliği bulunmayıp Minimum Fark sınıflandırmasında ortalama, Mahalanobis sınıflandırma algoritmasında da ortalama ve standart sapma bilgilerinin kullanılması nedeniyle benzerlik gösterdiği söylenebilir.

Geliştirilen algoritmanın bugüne kadar yapılan bazı çalışmalarda olmayan tarafları, herkesin kendine özgü standart sapmaları değerlendirildiğinden sistemdeki tüm kullanıcılar için aynı eşik seviyesinin değil kişilere özel eşik seviyesini kullanması, şifrenin 6 karakterin altına müsaade edilmemekte birlikte serbest uzunlukta seçilebilmesi, sistemin kullanıcıları tanımasının birkaç dakika gibi oldukça kısa zamanda gerçekleşebilmesi dolayısıyla uygulanabilir bir sistem olması olarak sıralanabilir.

5. SONUÇ

Biyometrik Güvenlik Sistemleri günümüzde sürekli gelişen, üzerinde yoğun çalışmaların devam ettiği ve yakın bir gelecekte ülkemizde dahi yaygın olarak kullanılacak sistemler olmaya aday görünmektedir.

Biyometrik Güvenlik Sistemlerinin en büyük avantajı her kişinin kendine özgü biyolojik özelliklerini kullanmasıdır. Böylelikle bu özelliklerin, dolayısıyla bir nevi şifrenin ele geçirilmesi, taklit edilmesi v.b. çok zorlaşmaktadır.

Biyometrik Güvenlik Sistemlerinin en büyük dezavantajı ise genellikle ilave bir donanıma ihtiyaç duyması ve bunun sonucu olarak şu an için yüksek maliyetleri gerektirmesi ile genel olarak belli bir hata oranını ihtiva etmeleridir. Ancak günümüzde konu üzerinde artan çalışmalar neticesinde yeni algoritmalarla veya mevcut algoritmaların geliştirilmesi suretiyle bu hata oranlarının sürekli olarak daha iyiye gittiğini söylemek mümkündür.

Bu tez çalışmasında ilave bir donanım gerektirmeyen ve dolayısıyla maliyeti diğer biyometrik güvenlik sistemlerine nazaran oldukça düşük olan klavyeye yazma ritmine göre kullanıcıların ayırt edilmesi ve tanınması üzerine çalışmalar yapılmış ve çok etkileyici sonuçlar elde edilmiştir. Elde edilen sonuçlar, ses, imza ve el geometrisi tanıma sistemlerinden daha iyi, parmak izi tanıma sistemleri hassasiyetine ise yakın olmakla birlikte herhangi bir ilave donanım gerektirmemesi bakımından bahse konu tanıma sistemlerinden daha çok uygulanabilir bir sistemdir.

Konu ile ilgili çalışmalar 20 yılı aşkındır sürdürülmekte olup günümüzde de halen hızla devam etmektedir. Bu çalışmada literatürdeki algoritmalar paralelinde mevcut algoritmaların hiçbirine tam olarak uymayan ancak mevcut algoritmalarından esinlenerek yeni bir yöntem ve algoritma geliştirilmiştir. Kişilerin kelimeleri yazarken genellikle yazacağı 3 veya 4 karakteri düşünerek yazdığı düşünülürse geliştirilen algoritma paralelinde sadece karakter çiftleri arası zamanların değil, aynı zamanda 3'lü veya 4'lü gruplar halindeki karakterler gruplarının da zaman analizlerinin yapılması konuya yeni bir boyut kazandırabilir.

Deneyisel çalışmaların daha büyük gruplara test ettirilerek aynı metin kalıplarının farklı kişilerce yazılmasının da analizleri yapıldığı takdirde daha da iyi sonuçlar elde edilebilecektir. Diğer tavsiye edilen bir konu da kişileri klavye önünde uzun süre testlere tabi tutmanın yerine, aralıklarla testlere tabi tutarak verilerin analiz edilmesidir. Zira bilgisayar önünde 5-10 dakika yazı yazdıktan sonra kişilerin normal ritminin bozulmaya başladığı söylenebilir.

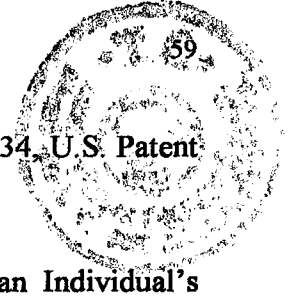
Sadece sistem girişi esnasında değil sisteme girişten sonra da kişilerin klavyeye basışlarının kaydedilerek mevcut Kişisel Bilgisayar/Terminal'den klavyeye basış ritimleri önemli ölçüde değiştiğinde sistem yöneticisinin muhtemel saldırı şeklinde otomatik ikazlanması, diğer araştırılması gereken bir konu olarak ortaya çıkmaktadır. Ancak daha öncede değinildiği gibi bu çok uzun araştırmaların yapılması gereken bir konu olup halihazırda firmalar ve araştırma grupları tarafından konu ile ilgili araştırmalar devam etmektedir ve bugüne kadar konu ile ilgili önemli bir gelişmenin sağlandığı literatürde yayınlanmamıştır.

Bundan sonra konu ile ilgili çalışma yapacak kişilerce çalışmanın daha da ilerilere götürülmesi açısından insanların yorgunluk, uykululuk, sinirlilik, dalgınlık v.b. hallerindeki yazma ritimleri ile öğrenme safhasındaki fiziksel koşullarla (Oturuş şekli, klavyenin yüksekliği ve uzaklığı, ellerinin klavyeye konuluş şekli v.b) sistem girişleri esnasındaki fiziksel koşulların kişilerin patern vektörlerini nasıl etkilediğinin incelenmesi, klavyeye yazan kişinin solak olup olmadığının tespiti v.b. çalışmalar yapılması konunun bugüne kadar üzerinde pek çalışma yapılmayan yönlerinin de parametre olarak hesaba katılmasında faydalı olacaktır ve uygulamalar web tabanlı veya işletim sistemleri ile entegre olacak şekilde tasarlanmalıdır.



KAYNAKLAR

- [1] Sungzoon Cho, Chigeun Han, Dae Hee Han, Hyung-II Kim, "Web Based Keystroke Dynamics Identity Verification using Neural Network", 1998
- [2] Aykut Güven, İbrahim Soğukpınar, "Yazma Ritmi Yöntemiyle Bilgisayar ve Ağ Güvenlik Sistemi Tasarımı", BAS2000, Ankara, Haziran 2000
- [3] Zdenek Riha, Vaclav Matyas, "Biometric Authentication Systems", Faculty of Informatics, Masarsky University, 1-46, Kasım 2000
- [4] Richard O.Duda, Peter E.Hart, "Pattern Classification and Scene Analysis", John Wiley & Sons, 1973
- [5] R.Gaines, W.Lisowski, S.Press and N.Shapiro, "Authentication by Keystroke Timing: Some Preliminary Results", Rand Report R-2526-NSF, Rand Corporation, 1-47, Mayıs 1980
- [6] Saleh Bleha, "Recognition Systems Based on Keystroke Dynamics", Ph.D. dissertation, Univ.Missouri-Colombia, Mayıs 1988
- [7] Saleh Bleha, Charles Slivinsky, Bassam Hussein, "Computer-Access Security Systems using Keystroke Dynamics", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, No.12, Aralık 1990
- [8] D.Umphress and G.Williams, "Identity Verification through Keyboard Characteristics", International Journal of Man-Machine Studies, 23(3):263-273, Eylül 1985
- [9] John Leggett and Glen Williams, "Verifying Identity Keystroke Characteristics", International Journal of Man-Machine Studies, 28(1):67-76, 1988
- [10] John Leggett, Glen Williams, David Umphress, "Verification of User Identity via Keystroke Characteristics", Human Factors in Management Information Systems, 1989
- [11] John Leggett, Glen Williams, Mark Usnick, M. Longnecker, "Dynamic Identity Verification via Keystroke Characteristics", International Journal of Man-Machine Studies, Vol 35, 859-870, 1991



- [12] J. Garcia, "Personal Identification Apparatus", Patent No. 4621334, U.S. Patent and Trademark Office, Washington D.C., 1986
- [13] J.Young, R.Hammon, "Method and Apparatus for Verifying an Individual's Identity", Patent No. 4805222, U.S. Patent and Trademark Office, Washington D.C., 1989
- [14] Rick Joyce and Goyal Gupta, "Identity Authorization Based on Keystroke Latencies", Communications of the ACM, 33(2):168-176, Şubat 1990
- [15] Bassam Hussein, R.Maclaren, Saleh Bleha, "An Application of Fuzzy Algorithms in a Computer Access Security System", Pattern Recognition Letters, 9, 39-43, 1989
- [16] Marcus Brown and Samuel Joe Rogers, "User Identification via Keystroke Characteristics of Typed Names Using Neural Networks", International Journal of Man-Machine Studies, Vol 39(6), 999 -1014, 1993
- [17] Thomas J.Alexandre, "Biometrics on Smartcards an Approach to Keyboard Behavioral Signature". In proceedings of the 2nd Smart Card Research & Advanced Applications Conference, 1996
- [18] A.Sermet Anagün, "Bilgi Güvenliğinin Sağlanmasında Kullanıcı Özelliklerine Dayalı bir Yapay Sinirsel Ağ Yaklaşımı", Endüstri Mühendisliği, Cilt 10, 1999
- [19] Gentner, "Keystroke timing in transcription typing", Cognitive Aspects of Skilled Typewriting, 95-120, 1993
- [20] Fabian Monroe, Aviel Rubin, "Authentication via Keystroke Dynamics", In proceedings of the 4th ACM Conference on Computer and Communications Security, 48-56, Nisan 1997
- [21] Fabian Monroe, Aviel D. Rubin, "Keystroke Dynamics as a Biometric for Authentication, Mart 1999
- [22] Dawn Song, Peter Venable, Adrian Perrig, "User Recognition by Keystroke Latency Pattern Analysis", Nisan 1997



- [23] Fabian Monrose, Michael K. Reiter, Susanne Wetzel, "Password Hardening Based On Keystroke Dynamics", In proceedings of the 6th ACM Conference on Computer and Communications Security, Kasım 1999
- [24] Alen Peacock, "Learning User Keystroke Latency Patterns (Preliminary Report)", 2000
- [25] Dawn Xiaodong Song, David Wagner, Xuqing Tian, "Timing Analysis of Keystrokes and Timing Attacks on SSH", 2001
- [26] D.Mahar, R.Naprer, M.Wagner, W.Laverty, R.Henderson, M.Hiron, "Optimizing Diagraph-Latencies Based on Biometric Typist Verification Systems Inter And Intra Typists Differences in Diqraph Latency Distribution", International Journal of Human Computer Studies, 93:579-552, 1995
- [27] J.A Robinson, V. M. Liang, J.A. Chambers, C.L.MacKenzie, "Computer User Verification Using Login String Keystroke Dynamics", IEEE Transactions on System Man and Cybernetics, 28(2), 1998
- [28] M.Obaidat, S. Sadoun, "Verification of Computer Users Using Keystroke Dynamics", IEEE Transactions on System, Man and Cybernetics, Vol 27 , No.2 , 261-269, 1997'
- [29] Orhan İdil, "İşletmeciler için Genel İstatistik", 237-251, 1989

EKLER

ALGORİTMA KAYNAK KODLARI

```
program Project1;
```

```
uses
```

```
  Forms,  
  Unit2 in 'Unit2.pas' {Form2},  
  Unit1 in 'Unit1.pas' {Form1},  
  Unit3 in 'Unit3.pas' {Form3},  
  Unit5 in 'Unit5.pas' {Form5},  
  Unit4 in 'Unit4.pas' {Form4};
```

```
{$R *.RES}
```

```
begin
```

```
  Application.Initialize;  
  Application.CreateForm(TForm1, Form1);  
  Application.CreateForm(TForm5, Form5);  
  Application.CreateForm(TForm4, Form4);  
  Application.Run;  
end.
```

UNIT1

```
unit Unit1;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
  Menus, Db, DBTables;
```

```
type
```

```
  TForm1 = class(TForm)  
    MainMenu1: TMainMenu;  
    AdminGirii1: TMenuItem;  
    File1: TMenuItem;  
    AdminTntm: TMenuItem;  
    SistemAyarlari: TMenuItem;  
    Yenikullanici: TMenuItem;  
    YardmKonular1: TMenuItem;  
    Haknda1: TMenuItem;  
    k1: TMenuItem;  
    Programdank1: TMenuItem;  
    EskiKullancSistemGirisi: TMenuItem;  
    Table1: TTable;  
    Table1NAME: TStringField;  
    Table1SNAME: TStringField;  
    Table1PASS: TStringField;  
    Table1LOCKED: TBooleanField;  
    Database1: TDatabase;  
    procedure AdminTntmClick(Sender: TObject);  
    procedure Programdank1Click(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    procedure SistemAyarlariClick(Sender: TObject);  
    procedure YenikullaniciClick(Sender: TObject);  
    procedure EskiKullancSistemGirisiClick(Sender: TObject);  
    procedure FormActivate(Sender: TObject);
```

```

    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    private
        { Private declarations }
    public
        { Public declarations }
        AdmEntry : boolean; //Admin Girisi
    end;
var
    Form1: TForm1;
implementation
uses Unit2, Unit3, Unit4, Unit5;
var
    Form2 : TForm2;
    Form3 : TForm3;
    Form4 : TForm4;
    Form5 : TForm5;
{$R *.DFM}

procedure TForm1.AdminTntmClick(Sender: TObject);
begin
    Form2 := TForm2.Create(nil);
    try
        Form2.ShowModal;
    finally
        Form2.Free;
        if Table1.RecordCount > 0 then AdminTntm.Enabled := False;
    end;
end;
procedure TForm1.Programdank1Click(Sender: TObject);
begin
    Close;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    Table1.Open;
    if Table1.RecordCount > 0 then AdminTntm.Enabled := False;
end;
procedure TForm1.SistemAyarlariClick(Sender: TObject);
begin
    Form3 := TForm3.Create(nil);
    try
        AdmEntry := True;
        Form3.ShowModal;
        if Form3.ModalResult = MrOK then
            begin
                try
                    Form5 := TForm5.Create(nil);
                    Form5.ShowModal;
                finally
                    Form5.Free;
                end;
            end;
    finally
        Form3.Free;
        if Table1.RecordCount > 0 then AdminTntm.Enabled := False;
    end;
end;
procedure TForm1.YenikullaniciClick(Sender: TObject);
begin
    Form4 := TForm4.Create(nil);
    try
        Form4.ShowModal;

```

```

finally
  Form4.Free;
  if Table1.RecordCount > 0 then AdminTntm.Enabled := False;
end;
end;
procedure TForm1.EskiKullancSistemGirisiClick(Sender: TObject);
begin
  Form3 := TForm3.Create(nil);
  try
    AdmEntry := False;
    Form3.caption := 'Eski Kullanıcı Sistem Girişi';
    Form3.Show;
  finally
    // Form3.Free;
    if Table1.RecordCount > 0 then AdminTntm.Enabled := False;
  end;
end;
procedure TForm1.FormActivate(Sender: TObject);
begin
  database1.Connected:=true;
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  database1.connected:=false;
end;
end.

```

UNIT2

```
unit Unit2;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Db, DBTables, Mask, DBCtrls, ExtCtrls, ComCtrls, Math, Buttons,
  Grids, DBGrids;
```

```
type
```

```

  TForm2 = class(TForm)
    DataSource1: TDataSource;
    Label1: TLabel;
    label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    Button2: TButton;
    ProgressBar1: TProgressBar;
    Label6: TLabel;
    Label7: TLabel;
    Timer1: TTimer;
    Timer2: TTimer;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    ListBox1: TListBox;
    ListBox2: TListBox;
    Label8: TLabel;
    Label9: TLabel;
    Table1: TTable;
    Table1NAME: TStringField;

```

Table1SNAME: TStringField;
Table1PASS: TStringField;
Table1LOCKED: TBooleanField;
Table9: TTable;
Table9CRN: TAutoIncField;
Table9NAME: TStringField;
Table9SNAME: TStringField;
Table9PASS: TStringField;
Table9ADIORT: TFloatField;
Table9ADISAP: TFloatField;
Table9ADISKEW: TFloatField;
Table9ADICURR: TFloatField;
Table9SOYORT: TFloatField;
Table9SOYSAP: TFloatField;
Table9SOYSKEW: TFloatField;
Table9SOYCURR: TFloatField;
Table9SIFORT: TFloatField;
Table9SIFSAP: TFloatField;
Table9SIFSKEW: TFloatField;
Table9SIFCURR: TFloatField;
Query2: TQuery;
Query2AVERAGEOFadiort: TFloatField;
Query2AVERAGEOFadisap: TFloatField;
Query2AVERAGEOFadiskew: TFloatField;
Query2AVERAGEOFadicurr: TFloatField;
Query2AVERAGEOFsoyort: TFloatField;
Query2AVERAGEOFsoysap: TFloatField;
Query2AVERAGEOFsoyskew: TFloatField;
Query2AVERAGEOFsoycurr: TFloatField;
Query2AVERAGEOFsifort: TFloatField;
Query2AVERAGEOFsifsap: TFloatField;
Query2AVERAGEOFsifskew: TFloatField;
Query2AVERAGEOFsifcurr: TFloatField;
Table10: TTable;
Table10CRN: TAutoIncField;
Table10NAME: TStringField;
Table10SNAME: TStringField;
Table10PASS: TStringField;
Table10ADIORTORT: TFloatField;
Table10ADISAPORT: TFloatField;
Table10ADISKEWORT: TFloatField;
Table10ADICURRORT: TFloatField;
Table10SOYORTORT: TFloatField;
Table10SOYSAPORT: TFloatField;
Table10SOYSKEWORT: TFloatField;
Table10SOYCURRORT: TFloatField;
Table10SIFORTORT: TFloatField;
Table10SIFSAPORT: TFloatField;
Table10SIFSKEWORT: TFloatField;
Table10SIFCURRORT: TFloatField;
Table10ADIORTSTSAPMA: TFloatField;
Table10SOYORTSTSAPMA: TFloatField;
Table10SIFORTSTSAPMA: TFloatField;
Table10ADIORTBBOTTOM: TFloatField;
Table10ADIORTBTOP: TFloatField;
Table10SOYORTBBOTTOM: TFloatField;
Table10SOYORTBTOP: TFloatField;
Table10SIFORTBBOTTOM: TFloatField;
Table10SIFORTBTOP: TFloatField;
Query6: TQuery;
Table12: TTable;
DataSource2: TDataSource;



Table9farkkare: TFloatField;
Table12NAME: TStringField;
Table12SNAME: TStringField;
Table12PASS: TStringField;
Table12ADIORTSTSAPMA: TFloatField;
Table12SOYADIORTSTSAPMA: TFloatField;
Table12SIFREORTSTSAPMA: TFloatField;
Query6toplamadi: TFloatField;
Query6toplamsoyadi: TFloatField;
Query6toplamsifre: TFloatField;
Table2: TTable;
Table2LockCounter: TFloatField;
Table2TrainingNo: TFloatField;
Table1ADMIN: TBooleanField;
Table12ADISKEWSTSAPMA: TFloatField;
Table12SOYADISKEWSTSAPMA: TFloatField;
Table12SIFRESKEWSTSAPMA: TFloatField;
Table12ADICURRSTSAPMA: TFloatField;
Table12SOYADICURRSTSAPMA: TFloatField;
Table12SIFRECURRSTSAPMA: TFloatField;
Table10ADIORTMIN: TFloatField;
Table10ADIORTMAX: TFloatField;
Table10SOYORTMIN: TFloatField;
Table10SOYORTMAX: TFloatField;
Table10SIFORTMIN: TFloatField;
Table10SIFORTMAX: TFloatField;
Table3: TTable;
Table3NAME: TStringField;
Table3SNAME: TStringField;
Table3PASS: TStringField;
Table3ADIBASORTSTSAPMA: TFloatField;
Table3SOYADIBASORTSTSAPMA: TFloatField;
Table3SIFREBASORTSTSAPMA: TFloatField;
Table3ADIBASSKEWSTSAPMA: TFloatField;
Table3SOYADIBASSKEWSTSAPMA: TFloatField;
Table3SIFREBASSKEWSTSAPMA: TFloatField;
Table3ADIBASCURRSTSAPMA: TFloatField;
Table3SOYADIBASCURRSTSAPMA: TFloatField;
Table3SIFREBASCURRSTSAPMA: TFloatField;
Table5: TTable;
Table5CRN: TAutoIncField;
Table5NAME: TStringField;
Table5SNAME: TStringField;
Table5PASS: TStringField;
Table5ADIBASORTORT: TFloatField;
Table5ADIBASSAPORT: TFloatField;
Table5ADIBASSKEWORT: TFloatField;
Table5ADIBASCURRORT: TFloatField;
Table5SOYBASORTORT: TFloatField;
Table5SOYBASSAPORT: TFloatField;
Table5SOYBASSKEWORT: TFloatField;
Table5SOYBASCURRORT: TFloatField;
Table5SIFBASORTORT: TFloatField;
Table5SIFBASSAPORT: TFloatField;
Table5SIFBASSKEWORT: TFloatField;
Table5SIFBASCURRORT: TFloatField;
Table5ADIBASORTSTSAPMA: TFloatField;
Table5SOYBASORTSTSAPMA: TFloatField;
Table5SIFBASORTSTSAPMA: TFloatField;
Table5ADIBASORTBBOTTOM: TFloatField;
Table5ADIBASORTBTOP: TFloatField;
Table5SOYBASORTBBOTTOM: TFloatField;



Table5SOYBASORTBTOP: TFloatField;
Table5SIFBASORTBBOTTOM: TFloatField;
Table5SIFBASORTBTOP: TFloatField;
Table5ADIBASORTMIN: TFloatField;
Table5ADIBASORTMAX: TFloatField;
Table5SOYBASORTMIN: TFloatField;
Table5SOYBASORTMAX: TFloatField;
Table5SIFBASORTMIN: TFloatField;
Table5SIFBASORTMAX: TFloatField;
Table4: TTable;
Table4CRN: TAutoIncField;
Table4NAME: TStringField;
Table4SNAME: TStringField;
Table4PASS: TStringField;
Table4ADIBASORT: TFloatField;
Table4ADIBASSAP: TFloatField;
Table4ADIBASSKEW: TFloatField;
Table4ADIBASCURR: TFloatField;
Table4SOYBASORT: TFloatField;
Table4SOYBASSAP: TFloatField;
Table4SOYBASSKEW: TFloatField;
Table4SOYBASCURR: TFloatField;
Table4SIFBASORT: TFloatField;
Table4SIFBASSAP: TFloatField;
Table4SIFBASSKEW: TFloatField;
Table4SIFBASCURR: TFloatField;
Query1: TQuery;
Query1AVERAGEOFadibasort: TFloatField;
Query1AVERAGEOFadibassap: TFloatField;
Query1AVERAGEOFadibasskew: TFloatField;
Query1AVERAGEOFadibascrr: TFloatField;
Query1AVERAGEOFsoybasort: TFloatField;
Query1AVERAGEOFsoybassap: TFloatField;
Query1AVERAGEOFsoybasskew: TFloatField;
Query1AVERAGEOFsoybascrr: TFloatField;
Query1AVERAGEOFsifbasort: TFloatField;
Query1AVERAGEOFsifbassap: TFloatField;
Query1AVERAGEOFsifbasskew: TFloatField;
Query1AVERAGEOFsifbascrr: TFloatField;
Query3: TQuery;
Query3toplamadibas: TFloatField;
Query3toplamsoyadibas: TFloatField;
Query3toplamsifrebas: TFloatField;
Button3: TButton;
Query4: TQuery;
Query5: TQuery;
tbl_nametime: TTable;
tbl_nametimeTNN: TAutoIncField;
tbl_nametimeNAME: TStringField;
tbl_nametimeSNAME: TStringField;
tbl_nametimePASS: TStringField;
tbl_nametimeKARA01: TFloatField;
tbl_nametimeKARA02: TFloatField;
tbl_nametimeKARA03: TFloatField;
tbl_nametimeKARA04: TFloatField;
tbl_nametimeKARA05: TFloatField;
tbl_nametimeKARA06: TFloatField;
tbl_nametimeKARA07: TFloatField;
tbl_nametimeKARA08: TFloatField;
tbl_nametimeKARA09: TFloatField;
tbl_nametimeKARA10: TFloatField;
tbl_nametimeKARA11: TFloatField;

tbl_nametimeKARA12: TFloatField;
tbl_nametimeKARA13: TFloatField;
tbl_nametimeKARA14: TFloatField;
tbl_nametimeKARA15: TFloatField;
tbl_snametime: TTable;
tbl_snametimeTSN: TAutoIncField;
tbl_snametimeNAME: TStringField;
tbl_snametimeSNAME: TStringField;
tbl_snametimePASS: TStringField;
tbl_snametimeKARA01: TFloatField;
tbl_snametimeKARA02: TFloatField;
tbl_snametimeKARA03: TFloatField;
tbl_snametimeKARA04: TFloatField;
tbl_snametimeKARA05: TFloatField;
tbl_snametimeKARA06: TFloatField;
tbl_snametimeKARA07: TFloatField;
tbl_snametimeKARA08: TFloatField;
tbl_snametimeKARA09: TFloatField;
tbl_snametimeKARA10: TFloatField;
tbl_snametimeKARA11: TFloatField;
tbl_snametimeKARA12: TFloatField;
tbl_snametimeKARA13: TFloatField;
tbl_snametimeKARA14: TFloatField;
tbl_snametimeKARA15: TFloatField;
tbl_passtime: TTable;
tbl_passtimeTPN: TAutoIncField;
tbl_passtimeNAME: TStringField;
tbl_passtimeSNAME: TStringField;
tbl_passtimePASS: TStringField;
tbl_passtimeKARA01: TFloatField;
tbl_passtimeKARA02: TFloatField;
tbl_passtimeKARA03: TFloatField;
tbl_passtimeKARA04: TFloatField;
tbl_passtimeKARA05: TFloatField;
tbl_passtimeKARA06: TFloatField;
tbl_passtimeKARA07: TFloatField;
tbl_passtimeKARA08: TFloatField;
tbl_passtimeKARA09: TFloatField;
tbl_passtimeKARA10: TFloatField;
tbl_passtimeKARA11: TFloatField;
tbl_passtimeKARA12: TFloatField;
tbl_passtimeKARA13: TFloatField;
tbl_passtimeKARA14: TFloatField;
tbl_passtimeKARA15: TFloatField;
tbl_aralikname: TTable;
tbl_araliknameSIRANO: TAutoIncField;
tbl_araliknameNAME: TStringField;
tbl_araliknameSNAME: TStringField;
tbl_araliknamePASS: TStringField;
tbl_araliknameORT: TFloatField;
tbl_araliknameSTSAPMA: TFloatField;
tbl_araliknameSKEW: TFloatField;
tbl_araliknameKURTOSIS: TFloatField;
tbl_araliksname: TTable;
tbl_araliksnameSIRANO: TAutoIncField;
tbl_araliksnameNAME: TStringField;
tbl_araliksnameSNAME: TStringField;
tbl_araliksnamePASS: TStringField;
tbl_araliksnameORT: TFloatField;
tbl_araliksnameSTSAPMA: TFloatField;
tbl_araliksnameSKEW: TFloatField;
tbl_araliksnameKURTOSIS: TFloatField;

```

tbl_aralikpass: TTable;
tbl_aralikpassSIRANO: TAutoIncField;
tbl_aralikpassNAME: TStringField;
tbl_aralikpassSNAME: TStringField;
tbl_aralikpassPASS: TStringField;
tbl_aralikpassORT: TFloatField;
tbl_aralikpassSTSAPMA: TFloatField;
tbl_aralikpassSKEW: TFloatField;
tbl_aralikpassKURTOSIS: TFloatField;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure GetTimeBas(Sender: TObject);
procedure GetTimeAra(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Edit1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Table9CalcFields(DataSet: TDataSet);
procedure Button3Click(Sender: TObject);
procedure TimeTblBosalt(sender:Tobject);
procedure TimeTotalTblDelete(sender:Tobject);
procedure Aralikmoments;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
// procedure tblnameyaz(sender:Tobject);

private
{ Private declarations }
procedure CheckType(Key : Word; Sender:TObject);
procedure FormClear;
procedure Clear;
procedure stsapmahesaplama;
public
{ Public declarations }
procedure momenthesaplama;
procedure ortalamahesaplama;
end;
var
  Form2: TForm2;
  KAra :integer=0;
  KDown :integer=0;
  i:integer=0;
  j:integer=0;
  tbl_ind:integer=3;
  tblname_son:integer=0;
  tblsname_son:integer=0;
  tblpasstime_son:integer=0;
  basma1:array[1..15] of double;
  aralik1:array[1..15] of double;
  basma2:array[1..15] of double;
  aralik2:array[1..15] of double;
  basma3:array[1..15] of double;
  aralik3:array[1..15] of double;
  count:integer=0;
  delbackmesaj : boolean;
implementation
uses Unit1,Unit3;
{$R *.DFM}

```

```

procedure TForm2.FormClear;
begin
    Edit1.Text := '';
    Edit2.Text := '';
    Edit3.Text := '';
    listBox1.Clear;
    listBox2.Clear;
end;

```

```

procedure TForm2.Clear;
begin
    ListBox1.Items.Clear;
    ListBox2.Items.Clear;
    Timer1.Enabled := False;
    Timer2.Enabled := False;
    KDwn := 0;
    KAra := 0;
    i := 0;
    j := 0;
    tbl_ind := 3;
end;

```

```

procedure TForm2.Button2Click(Sender: TObject);
begin

```

```

    formclear;
    clear;
    edit1.setfocus;
    TimeTblBosalt(sender);
end;

```

```

procedure TForm2.stsapmahesaplama;
begin

```

```

    Query6.Close;
    Query6.ParamByName('NAME').AsString := Table1NAME.AsString;
    Query6.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    Query6.ParamByName('PASS').AsString := Table1PASS.AsString;
    Query6.Prepare;
    Query6.Open;
    Table12.Open;
    Table12.Insert;
    Table12ADIORTSTSAPMA.asfloat := SQRT(Query6toplamadi.Value);
    Table12SOYADIORTSTSAPMA.asfloat := SQRT(Query6toplamsoyadi.Value);
    Table12SIFREORTSTSAPMA.asfloat := SQRT(Query6toplamsifre.Value);
    Table12.Post;
    Table12.Close;
    query3.Close;
    query3.ParamByName('NAME').AsString := Table1NAME.AsString;
    query3.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    query3.ParamByName('PASS').AsString := Table1PASS.AsString;
    query3.Prepare;
    query3.Open;
    table3.Open;
    table3.Insert;
    table3ADIBasORTSTSAPMA.asfloat := SQRT(query3toplamadibas.Value);
    table3SOYADIBasORTSTSAPMA.asfloat := SQRT(query3toplamsoyadibas.Value);
    table3SIFREbasORTSTSAPMA.asfloat := SQRT(query3toplamsifrebas.Value);
    table3.Post;
    table3.Close;
end;

```

```

procedure TForm2.FormCreate(Sender: TObject);
begin
    table1.open;
    table2.open;

```

```

table3.open;
table4.open;
table5.open;
table9.open;
table10.open;
table12.open;
TBL_nametime.open;
tbl_snametime.open;
tbl_passtime.open;
tbl_aralikname.open;
tbl_araliksname.open;
tbl_aralikpass.open;
Count:=0;
Progressbar1.max := Table2TrainingNo.AsInteger;
FormClear;
Clear;
end;

procedure TForm2.momenthesaplama;
var
  AdiM1, AdiM2, adiM3, adiM4, adiSkew, adiKurtosis: Extended;
  adiStsapma : Extended;
  AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew, AdibasKurtosis: Extended;
  adibasStsapma : Extended;
  soyadiM1, soyadiM2, soyadiM3, soyadiM4, soyadiSkew, soyadiKurtosis: Extended;
  soyadiStsapma : Extended;
  SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4, SoyAdibasSkew,
  SoyAdibasKurtosis: Extended;
  soyadibasStsapma : Extended;
  sifreM1, sifreM2, sifreM3, sifreM4, sifreSkew, sifreKurtosis: Extended;
  sifreStsapma : Extended;
  SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew, SifrebasKurtosis: Extended;
  sifrebasStsapma : Extended;
begin
  MomentSkewKurtosis(aralik1,AdiM1,AdiM2,AdiM3,AdiM4,AdiSkew,AdiKurtosis);
  AdiStSapma := Sqrt(AdiM2);
  MomentSkewKurtosis(basma1, AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew,
  AdibasKurtosis);
  AdibasStsapma := Sqrt(AdibasM2);
  MomentSkewKurtosis(aralik2,SoyAdiM1,SoyAdiM2,SoyAdiM3,SoyAdiM4,SoyAdiSkew,
  SoyAdiKurtosis);
  SoyAdiStSapma := Sqrt(SoyAdiM2);
  MomentSkewKurtosis(basma2, SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4,
  SoyAdibasSkew, SoyAdibasKurtosis);
  SoyAdibasStSapma := Sqrt(SoyAdibasM2);
  MomentSkewKurtosis(aralik3, SifreM1, SifreM2, SifreM3, SifreM4, SifreSkew, SifreKurtosis);
  SifreStSapma := Sqrt(SifreM2);
  MomentSkewKurtosis(basma3, SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew,
  SifrebasKurtosis);
  SifrebasStSapma := Sqrt(SifrebasM2);
  AdiM2:=0;AdiM3:=0;AdiM4:=0;AdiSkew:=0;AdiKurtosis:=0;
  AdibasM2:=0;AdibasM3:=0;AdibasM4:=0;AdibasSkew:=0;AdibasKurtosis:=0;
  SoyAdiM2:=0;SoyAdiM3:=0;SoyAdiM4:=0;SoyAdiSkew:=0;SoyAdiKurtosis:=0;
  SoyAdibasM2:=0; SoyAdibasM3:=0; SoyAdibasM4:=0; SoyAdibasSkew:=0;
  SoyAdibasKurtosis:=0;
  SifreM2:=0; SifreM3:=0; SifreM4:=0; SifreSkew:=0; SifreKurtosis:=0;
  SifrebasM2:=0; SifrebasM3:=0; SifrebasM4:=0; SifrebasSkew:=0; SifrebasKurtosis:=0; }
  try
    Table9.Open;
    Table9.Insert;
    Table9ADIORT.AsFloat := AdiM1;
    Table9ADISAP.AsFloat := AdiStSapma;

```

```

Table9ADISKEW.AsFloat := AdiSkew;
Table9ADICURR.AsFloat := AdiKurtosis;
Table9SOYORT.AsFloat := SoyAdiM1;
Table9SOYSAP.AsFloat := SoyAdiStSapma;
Table9SOYSkew.AsFloat := SoyAdiSkew;
Table9SOYcurr.AsFloat := SoyAdiKurtosis;
Table9SIFORT.AsFloat := SifreM1;
Table9SIFSAP.AsFloat := SifreStSapma;
Table9SIFSkew.AsFloat := SifreSkew;
Table9SIFcurr.AsFloat := SifreKurtosis;
Table9.Post;
Table4.Open;
Table4.Insert;
Table4ADIBasORT.AsFloat := AdibasM1;
Table4ADIBasSAP.AsFloat := AdibasStSapma;
Table4ADIBasSKEW.AsFloat := AdibasSkew;
Table4ADIBasCURR.AsFloat := AdibasKurtosis;
Table4SOYbasORT.AsFloat := SoyAdibasM1;
Table4SOYbasSAP.AsFloat := SoyAdibasStSapma;
Table4SOYbasSkew.AsFloat := SoyAdibasSkew;
Table4SOYbascurr.AsFloat := SoyAdibasKurtosis;
Table4SIFbasORT.AsFloat := SifrebasM1;
Table4SIFbasSAP.AsFloat := SifrebasStSapma;
Table4SIFbasSkew.AsFloat := SifrebasSkew;
Table4SIFbascurr.AsFloat := SifrebasKurtosis;
Table4.Post;
finally
Table9.Close;
Table4.Close;
end;
end;
procedure TForm2.ortalamahesaplama;
begin
try
Query2.Close;
Query2.Prepare;
Query2.ParamByName('NAME').AsString := Table1NAME.AsString;
Query2.ParamByName('SNAME').AsString := Table1SNAME.AsString;
Query2.ParamByName('PASS').AsString := Table1PASS.AsString;
Query2.Open;
Table10.Open;
Table10.Insert;
Table10ADIORTORT.AsFloat := Query2AVERAGEOFadiort.AsFloat;
Table10ADISAPORT.AsFloat := Query2AVERAGEOFadisap.AsFloat;
Table10ADISKEWORT.AsFloat := Query2AVERAGEOFadiskew.AsFloat;
Table10ADICURRORT.AsFloat := Query2AVERAGEOFadicurr.AsFloat;
Table10SOYORTORT.AsFloat := Query2AVERAGEOFsoyort.AsFloat;
Table10SOYSAPORT.AsFloat := Query2AVERAGEOFsoysap.AsFloat;
Table10SOYSkewORT.AsFloat := Query2AVERAGEOFsoyskew.AsFloat;
Table10SOYcurrORT.AsFloat := Query2AVERAGEOFsoycurr.AsFloat;
Table10SIFORTORT.AsFloat := Query2AVERAGEOFsifort.AsFloat;
Table10SIFSAPORT.AsFloat := Query2AVERAGEOFsifsap.AsFloat;
Table10SIFSkewORT.AsFloat := Query2AVERAGEOFsifskew.AsFloat;
Table10SIFcurrORT.AsFloat := Query2AVERAGEOFsifcurr.AsFloat;
Table10.Post;
query1.Close;
query1.Prepare;
query1.ParamByName('NAME').AsString := Table1NAME.AsString;
query1.ParamByName('SNAME').AsString := Table1SNAME.AsString;
query1.ParamByName('PASS').AsString := Table1PASS.AsString;
query1.Open;
table5.Open;

```

```

table5.Insert;
table5ADIBASORTORT.AsFloat := query1AVERAGEOFadibasort.AsFloat;
table5ADIBASSAPORT.AsFloat := query1AVERAGEOFadibassap.AsFloat;
table5ADIBASSKEWORT.AsFloat := query1AVERAGEOFadibasskew.AsFloat;
table5ADIBASCURORT.AsFloat := query1AVERAGEOFadibascurr.AsFloat;
table5SOYBASORTORT.AsFloat := query1AVERAGEOFsoybasort.AsFloat;
table5SOYBASSAPORT.AsFloat := query1AVERAGEOFsoybassap.AsFloat;
table5SOYBASSKEWORT.AsFloat := query1AVERAGEOFsoybasskew.AsFloat;
table5SOYBASCURORT.AsFloat := query1AVERAGEOFsoybascurr.AsFloat;
table5SIFBASORTORT.AsFloat := query1AVERAGEOFsifbasort.AsFloat;
table5SIFBASSAPORT.AsFloat := query1AVERAGEOFsifbassap.AsFloat;
table5SIFBASSKEWORT.AsFloat := query1AVERAGEOFsifbasskew.AsFloat;
table5SIFBASCURORT.AsFloat := query1AVERAGEOFsifbascurr.AsFloat;
table5.Post;
finally
    Table10.Close;
    Table5.Close;
end;
end;
procedure TForm2.Button1Click(Sender: TObject);
var
    Say,ind,i : Integer;
begin
    if Count < Table2TrainingNo.AsInteger -1
    then //1.then
        begin //1.begin
            with Table1 do
                begin //2.begin
                    if Count = 0
                    then //2.then
                        begin //3.begin
                            if(length(Edit1.Text)<2)or(length(Edit2.Text)<2)or(length(Edit3.Text)< 6)
                            then //3.then
                                begin //4.begin
                                    Showmessage('Adı ve Soyadı için en az 2,Şifre için ise en az 6 karakterlik veri
girmelisiniz.');
                                    TimeTblBosalt(sender);
                                    FormClear;
                                    Edit1.SetFocus;
                                    Exit;
                                end
                            else //1.else
                                begin
                                    try
                                        Insert;
                                        FieldByName('NAME').AsString := Edit1.Text;
                                        FieldByName('SNAME').AsString := Edit2.Text;
                                        FieldByName('PASS').AsString := Edit3.Text;
                                        FieldByName('ADMIN').AsBoolean := True;
                                        Post;
                                        momenthesaplama;
                                        Inc(Count,1);
                                        ProgressBar1.position:=ProgressBar1.position + 1;
                                        label6.caption := inttostr(count);
                                        FormClear;
                                        Clear;
                                        Edit1.SetFocus;
                                    except
                                        ShowMessage('Veri Tabanına kayıt esnasında hata olustu!!! Lütfen bilgilerinizi tekrar
girmeyi deneyiniz.');
                                        Timetotaltbldelete(sender);
                                        Clear;

```

```

FormClear;
For Say := 1 to 15 do begin
    basma1[say]:= 0; aralik1[say]:= 0;
    basma2[say]:= 0; aralik2[say]:= 0;
    basma3[say]:= 0; aralik3[say]:= 0;
end; //for
Exit;
end; //try
end; // 1.else end
end //3.begin
else //2.else
begin //5.begin
    if(Edit1.Text <> FieldByName('NAME').AsString)or
    (Edit2.Text <> FieldByName('SNAME').AsString)or
    (Edit3.Text <> FieldByName('PASS').AsString)
    then
        begin //6.begin
            showmessage('Ad, Soyad veya Şifrenizi Yanlış Girdiniz!!!');
            TimeTblBosalt(sender);
            Clear;
            FormClear;
            For Say := 1 to 15 do begin
                basma1[say]:= 0; aralik1[say]:= 0;
                basma2[say]:= 0; aralik2[say]:= 0;
                basma3[say]:= 0; aralik3[say]:= 0;
            end; //for
            Edit1.SetFocus;
            Exit;
        end //6.begin
    else
        begin
            momenthesaplama;
            Inc(Count,1);
            ProgressBar1.position:=ProgressBar1.position + 1;
            label6.caption := inttostr(Count);
            Clear;
            FormClear;
            For Say := 1 to 15 do begin
                basma1[say]:= 0; aralik1[say]:= 0;
                basma2[say]:= 0; aralik2[say]:= 0;
                basma3[say]:= 0; aralik3[say]:= 0;
            end; //for
            Edit1.SetFocus;
        end;
    end ; //2.else
end ;
end//1.begin
else //1.then
begin
    momenthesaplama;
    ortalamahesaplama;
    stsapmahesaplama;
    //nametime,snametime,passtime tablolarına isim,soyad,şifreleri yaz
    //nametime
    ind:=table2.FieldByName('trainingno').asinteger-1;
    tbl_nametime.Editkey;
    tbl_nametime.FieldByName('tnn').asinteger:=tblname_son;
    tbl_nametime.GotoKey;
    for i:=tblname_son downto tblname_son-ind do
    begin
        tbl_nametime.edit;
        tbl_nametime.fieldbyname('name').asString:=edit1.text;
    end;
end;

```

```

tbl_nametime.fieldbyname('sname').asstring:=edit2.text;
tbl_nametime.fieldbyname('pass').asstring:=edit3.text;
tbl_nametime.post;
tbl_nametime.prior;
end;
//snametime
tbl_snametime.Editkey;
tbl_snametime.FieldByName('tsn').asinteger:=tblsname_son;
tbl_snametime.GotoKey;
for i:=tblsname_son downto tblsname_son-ind do
begin
tbl_snametime.edit;
tbl_snametime.fieldbyname('name').asstring:=edit1.text;
tbl_snametime.fieldbyname('sname').asstring:=edit2.text;
tbl_snametime.fieldbyname('pass').asstring:=edit3.text;
tbl_snametime.post;
tbl_snametime.prior;
end;
//passtime
tbl_passtime.Editkey;
tbl_passtime.FieldByName('tpn').asinteger:=tblpasstime_son;
tbl_passtime.GotoKey;
for i:=tblpasstime_son downto tblpasstime_son-ind do
begin
tbl_passtime.edit;
tbl_passtime.fieldbyname('name').asstring:=edit1.text;
tbl_passtime.fieldbyname('sname').asstring:=edit2.text;
tbl_passtime.fieldbyname('pass').asstring:=edit3.text;
tbl_passtime.post;
tbl_passtime.prior;
end;
Aralikmoments;
ProgressBar1.position:=ProgressBar1.position + 1;
label6.caption := inttostr(ProgressBar1.position);
ShowMessage('Sisteme yönetici olarak başarıyla kayıt edildiniz!!!');
ModalResult := MrOk;
end;

end;

procedure TForm2.CheckType(Key: Word; Sender: TObject);
var
    Say : Integer;
begin
    if(key = VK_DELETE)or(key = VK_BACK)
    then
        begin
            Showmessage('Delete veya BackSpace Tuşuna Bastınız.Lütfen Tekrar Giriş Yapınız!!!');
            TEdit(Sender).Text := "";
            delbackmesaj := True;
            if TEdit(Sender).Name = 'Edit1' then
                begin
                    For Say := 1 to 15 do begin basma1[say]:= 0; aralik1[say]:= 0;end;
                end;
            if TEdit(Sender).Name = 'Edit2' then
                begin
                    For Say := 1 to 15 do begin basma2[say]:= 0; aralik2[say]:= 0;end;
                end;
            if TEdit(Sender).Name = 'Edit3' then
                begin
                    For Say := 1 to 15 do begin basma3[say]:= 0; aralik3[say]:= 0;end;
                end;
        end;
    end;
end;

```

```

        Clear;
    end
else
    timer1.enabled := true;
    timer2.enabled := false;
end;

```

```

procedure TForm2.Timer2Timer(Sender: TObject);
begin
    KAra := KAra + timer2.interval;
end;

```

```

procedure TForm2.Timer1Timer(Sender: TObject);
begin
    Kdwn := Kdwn + timer1.interval;
end;

```

```

procedure TForm2.GetTimeBas(Sender: TObject);
begin
    if not delbackmesaj
    then
        begin
            i := i + 1;
            if TEdit(Sender).Name = 'Edit1'
            then
                begin
                    basma1[i] := KDwn;
                    ListBox1.Items.Add(FloatToStr(basma1[i]));
                end;
            if TEdit(Sender).Name = 'Edit2'
            then
                begin
                    basma2[i] := KDwn;
                    ListBox1.Items.Add(FloatToStr(basma2[i]));
                end;
            if TEdit(Sender).Name = 'Edit3'
            then
                begin
                    basma3[i] := KDwn;
                    ListBox1.Items.Add(FloatToStr(basma3[i]));
                end;
            KDwn:=0;
            Timer2.Enabled:=True;
        end;
end;

```

```

procedure TForm2.GetTimeAra(Sender: TObject);
begin
    if TEdit(Sender).Name = 'Edit1' then if not (tbl_nametime.State in [dsinsert]) then
    Tbl_nametime.Insert;
    if TEdit(Sender).Name = 'Edit2' then if not (tbl_snametime.State in [dsinsert]) then
    Tbl_snametime.Insert;
    if TEdit(Sender).Name = 'Edit3' then if not (tbl_passtime.State in [dsinsert]) then
    Tbl_passtime.Insert;
    j:=j + 1;
    inc(tbl_ind);
    if TEdit(Sender).Name = 'Edit1'
    then
        begin
            aralik1[j] := KAra;
            ListBox2.Items.Add(FloatToStr(aralik1[j]));
            tbl_nametime.fields[tbl_ind].AsInteger:=KAra;
        end;
    if TEdit(Sender).Name = 'Edit2'

```

```

then
begin
    aralik2[j] := KAra;
    ListBox2.Items.Add(FloatToStr(aralik2[j]));
    tbl_snametime.fields[tbl_ind].AsInteger:=KAra;
end;
if TEdit(Sender).Name = 'Edit3'
then
begin
    aralik3[j] := KAra;
    ListBox2.Items.Add(FloatToStr(aralik3[j]));
    tbl_passtime.fields[tbl_ind].AsInteger:=KAra;
end;
KAra:=0;
Timer2.Enabled:=False;
end;
procedure TForm2.Edit1KeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN))then
    begin
        timer1.Enabled:=true;
        timer2.Enabled:=false;
        GetTimeAra(Sender);
        delbackmesaj := False;
    end;
    CheckType(Key,Sender);
end;
procedure TForm2.Edit1KeyUp(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN)or(Key =
VK_TAB)or(Key = VK_SHIFT))then
    begin
        Timer1.Enabled := False;
        GetTimeBas(Sender);
    end;
end;
Procedure TForm2.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if(Key = #13)then
    begin
        if TEdit(Sender).Name = 'Edit1' then begin
            ActiveControl := Edit2;
            tbl_nametime.post;
            tblname_son:=tbl_nametime.fieldbyname('tnn').asinteger;
        end;
        if TEdit(Sender).Name = 'Edit2' then begin
            ActiveControl := Edit3;
            tbl_snametime.post;
            tblsname_son:=tbl_snametime.fieldbyname('tsn').asinteger;
        end;
        if TEdit(Sender).Name = 'Edit3' then begin
            ActiveControl := Button1;
            tbl_passtime.post;
            tblpasstime_son:=tbl_passtime.fieldbyname('tpn').asinteger;
        end;
        Clear;
    end;
end;
procedure TForm2.Table9CalcFields(DataSet: TDataSet);
begin

```

```

{
  Table9.FieldByName('farkkare').AsFloat := sqr((Table10.FieldByName('adiortort').AsFloat -
Table9.FieldByName('adiort').AsFloat));
  aditoplam:= aditoplam + Table9.FieldByName('farkkare').AsFloat;
  aditoplamort := aditoplam/8;
  adistsapma := sqrt(aditoplamort);
}
end;
procedure TForm2.Button3Click(Sender: TObject);
begin
  if (count < Table2trainingno.asinteger-1) and (count > 0)
  then
    begin
      with query4 do
      begin
        close;
        ParamByName('NAME').AsString := Table1NAME.AsString;
        ParamByName('SNAME').AsString :=Table1SNAME.AsString;
        ParamByName('PASS').AsString := Table1PASS.AsString;
        execsql;
      end;
      with query5 do
      begin
        close;
        ParamByName('NAME').AsString := Table1NAME.AsString;
        ParamByName('SNAME').AsString :=Table1SNAME.AsString;
        ParamByName('PASS').AsString := Table1PASS.AsString;
        execsql;
      end;
      table1.delete;
      TimeTotalTblDelete(sender);

    end;
  close;
end;

```

```

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  table1.close;
  table2.close;
  table3.close;
  table4.close;
  table5.close;
  table9.close;
  table10.close;
  table12.close;
  TBL_nametime.close;
  tbl_snametime.close;
  tbl_passtime.close;
  tbl_aralikname.close;
  tbl_araliksname.close;
  tbl_aralikpass.close;
end;

```

```

procedure TForm2.TimeTblBosalt(sender:Tobject);
begin
  tbl_nametime.editkey;
  tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
  tbl_nametime.gotokey;
  tbl_snametime.editkey;
  tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
  tbl_snametime.gotokey;

```

```

tbl_passtime.editkey;
tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
tbl_passtime.gotokey;
tbl_nametime.Delete;
tbl_snametime.Delete;
tbl_passtime.Delete;
end;
procedure TForm2.TimeTotalTblDelete(sender:TObject);
var
    i:integer;
begin
    i:=1;
    tbl_nametime.editkey;
    tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
    tbl_nametime.gotokey;
    tbl_snametime.editkey;
    tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
    tbl_snametime.gotokey;
    tbl_passtime.editkey;
    tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.gotokey;
    // showmessage('count: '+inttostr(count));
    while i<=count do
    begin
        tbl_nametime.Delete;
        tbl_snametime.Delete;
        tbl_passtime.Delete;
        inc(i);
    end;
end;
procedure TForm2.Aralikmoments;
var
    AdiM1, AdiM2, adiM3, adiM4, adiSkew, adiKurtosis: Extended;
    adiStsapma : Extended;
    soyadiM1, soyadiM2, soyadiM3, soyadiM4, soyadiSkew, soyadiKurtosis: Extended;
    soyadiStsapma : Extended;
    sifreM1, sifreM2, sifreM3, sifreM4, sifreSkew, sifreKurtosis: Extended;
    sifreStsapma : Extended;
    dizi1:array of double;
    str_uz,indis,ind_a,ind_b,ind_c:integer;
    ad_str,soyad_str,pass_str:string;
begin
    indis:=4;
    //
    //tbl_name
    //
    setlength(dizi1,Table2TrainingNo.AsInteger);
    tbl_nametime.editkey;
    tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
    tbl_nametime.gotokey;
    //
    ad_str:=tbl_nametime.fieldbyname('name').asstring;
    soyad_str:=tbl_nametime.fieldbyname('sname').asstring;
    pass_str:=tbl_nametime.fieldbyname('pass').asstring;
    //
    str_uz:=length(tbl_nametime.FieldName('name').asstring);
    for ind_a:=1 to str_uz-1 do
    begin
        tbl_nametime.editkey;
        tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
        tbl_nametime.gotokey;

```

```

for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
begin
    dizi1[ind_b]:=tbl_nametime.Fields[indis+ind_a].AsFloat;
    tbl_nametime.Prior;
end;
//moments olustur
// MomentSkewKurtosis(dizi1,AdiM1,AdiM2,AdiM3,AdiM4,AdiSkew,AdiKurtosis);
// AdiStSapma := Sqrt(AdiM2);
adim1:=mean(dizi1);
Adistsapma:=stddev(dizi1);
try
    Tbl_aralikname.open;
    Tbl_aralikname.insert;
    Tbl_aralikname.FieldName('name').asString:=ad_str;
    Tbl_aralikname.FieldName('sname').asString:=soyad_str;
    Tbl_aralikname.FieldName('pass').asString:=pass_str;
    Tbl_aralikname.FieldName('ORT').asfloat:=AdiM1;
    Tbl_aralikname.FieldName('STSAPMA').asfloat:=AdiStSapma;
    Tbl_aralikname.Post;
finally
    Tbl_aralikname.close;
end;
end;
dizi1:=nil;
//
//tbl_sname
//
setlength(dizi1,Table2TrainingNo.AsInteger);
tbl_snametime.editkey;
tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
tbl_snametime.gotokey;
//
ad_str:=tbl_snametime.fieldbyname('name').asString;
soyad_str:=tbl_snametime.fieldbyname('sname').asString;
pass_str:=tbl_snametime.fieldbyname('pass').asString;
//
str_uz:=length(tbl_snametime.FieldName('sname').asString);
for ind_a:=1 to str_uz-1 do
begin
    tbl_snametime.editkey;
    tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
    tbl_snametime.gotokey;
    for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
    begin
        dizi1[ind_b]:=tbl_snametime.Fields[indis+ind_a].AsFloat;
        tbl_snametime.Prior;
    end;
    //moments olustur
//
// MomentSkewKurtosis(dizi1,soyadiM1,soyadiM2,soyadiM3,soyadiM4,soyadiSkew,soyadiKurtosis);
// soyadiStSapma := Sqrt(soyadiM2);
soyadim1:=mean(dizi1);
soyAdistsapma:=stddev(dizi1);
try
    Tbl_araliksname.open;
    Tbl_araliksname.insert;
    Tbl_araliksname.FieldName('name').asString:=ad_str;
    Tbl_araliksname.FieldName('sname').asString:=soyad_str;
    Tbl_araliksname.FieldName('pass').asString:=pass_str;
    Tbl_araliksname.FieldName('ORT').asfloat:=soyadiM1;
    Tbl_araliksname.FieldName('STSAPMA').asfloat:=soyadiStSapma;
    Tbl_araliksname.Post;

```

```

finally
    Tbl_araliksname.close;
end;
end;
dizi1:=nil;
//
//tbl_PASS
//
setlength(dizi1,Table2TrainingNo.AsInteger);
tbl_passtime.editkey;
tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
tbl_passtime.gotokey;
//
ad_str:=tbl_passtime.fieldbyname('name').asstring;
soyad_str:=tbl_passtime.fieldbyname('sname').asstring;
pass_str:=tbl_passtime.fieldbyname('pass').asstring;
//
str_uz:=length(tbl_passtime.FieldName('pass').asstring);
for ind_a:=1 to str_uz-1 do
begin
    tbl_passtime.editkey;
    tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.gotokey;
    for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
    begin
        dizi1[ind_b]:=tbl_passtime.Fields[indis+ind_a].AsFloat;
        tbl_passtime.Prior;
    end;
    //moments olustur
//    MomentSkewKurtosis(dizi1,sifreM1,sifreM2,sifreM3,sifreM4,sifreSkew,sifreKurtosis);
//    sifreStSapma := Sqrt(sifreM2);
sifrem1:=mean(dizi1);
sifrestsapma:=stddev(dizi1);
try
    Tbl_aralikpass.open;
    Tbl_aralikpass.insert;
    Tbl_aralikpass.FieldName('name').asstring:=ad_str;
    Tbl_aralikpass.FieldName('sname').asstring:=soyad_str;
    Tbl_aralikpass.FieldName('pass').asstring:=pass_str;
    Tbl_aralikpass.FieldName('ORT').asfloat:=sifreM1;
    Tbl_aralikpass.FieldName('STSAPMA').asfloat:=sifreStSapma;
//    dm.T_aralikpass.FieldName('SKEW').asfloat:=sifreSkew;
//    dm.T_aralikpass.FieldName('KURTOSIS').asfloat:=sifreKurtosis;
    Tbl_aralikpass.Post;
finally
    Tbl_aralikpass.close;
end;
end;
dizi1:=nil;
end;
end.

```

UNIT3

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,



Db, DBTables, StdCtrls, Buttons, ExtCtrls, Math, Grids, DBGrids, Mask, DBCtrls;

type

```
TForm3 = class(TForm)
  Table1: TTable;
  DataSource1: TDataSource;
  Table1NAME: TStringField;
  Table1SNAME: TStringField;
  Table1PASS: TStringField;
  Table1LOCKED: TBooleanField;
  Timer1: TTimer;
  Timer2: TTimer;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  Table10: TTable;
  Table11: TTable;
  Table11CRN: TAutoIncField;
  Table11NAME: TStringField;
  Table11SNAME: TStringField;
  Table11PASS: TStringField;
  Table11ADIORT: TFloatField;
  Table11ADISAP: TFloatField;
  Table11ADISKEW: TFloatField;
  Table11ADICURR: TFloatField;
  Table11SOYORT: TFloatField;
  Table11SOYSAP: TFloatField;
  Table11SOYSKEW: TFloatField;
  Table11SOYCURRE: TFloatField;
  Table11SIFORT: TFloatField;
  Table11SIFSAP: TFloatField;
  Table11SIFSKEW: TFloatField;
  Table11SIFCURRE: TFloatField;
  Table10CRN: TAutoIncField;
  Table10NAME: TStringField;
  Table10SNAME: TStringField;
  Table10PASS: TStringField;
  Table10ADIORTORT: TFloatField;
  Table10ADISAPORT: TFloatField;
  Table10ADISKEWORT: TFloatField;
  Table10ADICURRORT: TFloatField;
  Table10SOYORTORT: TFloatField;
  Table10SOYSAPORT: TFloatField;
  Table10SOYSKEWORT: TFloatField;
  Table10SOYCURREORT: TFloatField;
  Table10SIFORTORT: TFloatField;
  Table10SIFSAPORT: TFloatField;
  Table10SIFSKEWORT: TFloatField;
  Table10SIFCURREORT: TFloatField;
  Table10ADIORTSTSAPMA: TFloatField;
  Table10SOYORTSTSAPMA: TFloatField;
  Table10SIFORTSTSAPMA: TFloatField;
  Table10ADIORTBBOTTOM: TFloatField;
  Table10ADIORTBTOP: TFloatField;
  Table10SOYORTBBOTTOM: TFloatField;
  Table10SOYORTBTOP: TFloatField;
  Table10SIFORTBBOTTOM: TFloatField;
  Table10SIFORTBTOP: TFloatField;
  Table9: TTable;
  Table9CRN: TAutoIncField;
  Table9NAME: TStringField;
  Table9SNAME: TStringField;
```

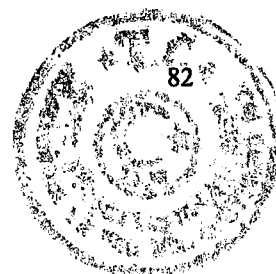


Table9PASS: TStringField;
Table9ADIORT: TFloatField;
Table9ADISAP: TFloatField;
Table9ADISKEW: TFloatField;
Table9ADICURR: TFloatField;
Table9SOYORT: TFloatField;
Table9SOYSAP: TFloatField;
Table9SOYSKEW: TFloatField;
Table9SOYCURR: TFloatField;
Table9SIFORT: TFloatField;
Table9SIFSAP: TFloatField;
Table9SIFSKEW: TFloatField;
Table9SIFCURR: TFloatField;
Table12: TTable;
Table12NAME: TStringField;
Table12SNAME: TStringField;
Table12PASS: TStringField;
Table12ADIORTSTSAPMA: TFloatField;
Table12SOYADIORTSTSAPMA: TFloatField;
Table12SIFREORTSTSAPMA: TFloatField;
Table2: TTable;
Table2LockCounter: TFloatField;
Table2TrainingNo: TFloatField;
Panel1: TPanel;
Label3: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Label4: TLabel;
Label5: TLabel;
Edit3: TEdit;
Table1ADMIN: TBooleanField;
Table10ADIORTMIN: TFloatField;
Table10ADIORTMAX: TFloatField;
Table10SOYORTMIN: TFloatField;
Table10SOYORTMAX: TFloatField;
Table10SIFORTMIN: TFloatField;
Table10SIFORTMAX: TFloatField;
Table11ADIORTSTSAPMA: TFloatField;
Table11SOYORTSTSAPMA: TFloatField;
Table11SIFORTSTSAPMA: TFloatField;
Table12ADISKEWSTSAPMA: TFloatField;
Table12SOYADISKEWSTSAPMA: TFloatField;
Table12SIFRESKEWSTSAPMA: TFloatField;
Table12ADICURRSTSAPMA: TFloatField;
Table12SOYADICURRSTSAPMA: TFloatField;
Table12SIFRECURRSTSAPMA: TFloatField;
Table11ENTRANCES: TBooleanField;
Table11SUCCESSENTRYCOUNT: TFloatField;
Table11LOGINDATE: TDateField;
Table11LOGINTIME: TTimeField;
Table3: TTable;
Table3CRN: TAutoIncField;
Table3NAME: TStringField;
Table3SNAME: TStringField;
Table3PASS: TStringField;
Table3ADIBASORTORT: TFloatField;
Table3ADIBASSAPORT: TFloatField;
Table3ADIBASSKEWORT: TFloatField;
Table3ADIBASCURRORT: TFloatField;
Table3SOYBASORTORT: TFloatField;
Table3SOYBASSAPORT: TFloatField;
Table3SOYBASSKEWORT: TFloatField;



Table3SOYBASCURRORT: TFloatField;
Table3SIFBASORTORT: TFloatField;
Table3SIFBASSAPORT: TFloatField;
Table3SIFBASSKEWORT: TFloatField;
Table3SIFBASCURRORT: TFloatField;
Table3ADIBASORTSTSAPMA: TFloatField;
Table3SOYBASORTSTSAPMA: TFloatField;
Table3SIFBASORTSTSAPMA: TFloatField;
Table3ADIBASORTBBOTTOM: TFloatField;
Table3ADIBASORTBTOP: TFloatField;
Table3SOYBASORTBBOTTOM: TFloatField;
Table3SOYBASORTBTOP: TFloatField;
Table3SIFBASORTBBOTTOM: TFloatField;
Table3SIFBASORTBTOP: TFloatField;
Table3ADIBASORTMIN: TFloatField;
Table3ADIBASORTMAX: TFloatField;
Table3SOYBASORTMIN: TFloatField;
Table3SOYBASORTMAX: TFloatField;
Table3SIFBASORTMIN: TFloatField;
Table3SIFBASORTMAX: TFloatField;
Table4: TTable;
Table4NAME: TStringField;
Table4SNAME: TStringField;
Table4PASS: TStringField;
Table4ADIBASORTSTSAPMA: TFloatField;
Table4SOYADIBASORTSTSAPMA: TFloatField;
Table4SIFREBASORTSTSAPMA: TFloatField;
Table4ADIBASSKEWSTSAPMA: TFloatField;
Table4SOYADIBASSKEWSTSAPMA: TFloatField;
Table4SIFREBASSKEWSTSAPMA: TFloatField;
Table4ADIBASCURRSTSAPMA: TFloatField;
Table4SOYADIBASCURRSTSAPMA: TFloatField;
Table4SIFREBASCURRSTSAPMA: TFloatField;
Table5: TTable;
Table5CRN: TAutoIncField;
Table5NAME: TStringField;
Table5SNAME: TStringField;
Table5PASS: TStringField;
Table5ADIBASORT: TFloatField;
Table5ADIBASSAP: TFloatField;
Table5ADIBASSKEW: TFloatField;
Table5ADIBASCURR: TFloatField;
Table5SOYBASORT: TFloatField;
Table5SOYBASSAP: TFloatField;
Table5SOYBASSKEW: TFloatField;
Table5SOYBASCURR: TFloatField;
Table5SIFBASORT: TFloatField;
Table5SIFBASSAP: TFloatField;
Table5SIFBASSKEW: TFloatField;
Table5SIFBASCURR: TFloatField;
Query1: TQuery;
Query1MAXOFsuccessentrycount: TFloatField;
Query2: TQuery;
Query2MAXOFfailentrycount: TFloatField;
Table2SecurityLevel: TFloatField;
Table11REALNAME: TStringField;
Table11REALSNAME: TStringField;
Table11SECURITYLEVEL: TFloatField;
Table11FAILENTRYCOUNT: TFloatField;
Table11SUCCESSENTRYPERCENT: TFloatField;
Query3: TQuery;
Table6: TTable;

Table6CRN: TAutoIncField;
Table6NAME: TStringField;
Table6SNAME: TStringField;
Table6PASS: TStringField;
Table6ADIORT: TFloatField;
Table6ADISAP: TFloatField;
Table6ADISKEW: TFloatField;
Table6ADICURR: TFloatField;
Table6SOYORT: TFloatField;
Table6SOYSAP: TFloatField;
Table6SOYSKEW: TFloatField;
Table6SOYCURRE: TFloatField;
Table6SIFORT: TFloatField;
Table6SIFSAP: TFloatField;
Table6SIFSKEW: TFloatField;
Table6SIFCURRE: TFloatField;
Query4: TQuery;
Table7: TTable;
Table7DELCOUNT: TFloatField;
Table11TRAININGNO: TFloatField;
Table6SUCCESSEENTRYCOUNT: TFloatField;
Table6TRAININGNO: TFloatField;
Query5: TQuery;
Query5AVERAGEOFadiort: TFloatField;
Query5AVERAGEOFadisap: TFloatField;
Query5AVERAGEOFadisaw: TFloatField;
Query5AVERAGEOFadicurr: TFloatField;
Query5AVERAGEOFsoyort: TFloatField;
Query5AVERAGEOFsoysap: TFloatField;
Query5AVERAGEOFsoysaw: TFloatField;
Query5AVERAGEOFsoycurre: TFloatField;
Query5AVERAGEOFsifort: TFloatField;
Query5AVERAGEOFsifsap: TFloatField;
Query5AVERAGEOFsifaw: TFloatField;
Query5AVERAGEOFsifcurre: TFloatField;
Table8: TTable;
Query6: TQuery;
Query7: TQuery;
Query8: TQuery;
Query6toplamadi: TFloatField;
Query6toplamsoyadi: TFloatField;
Query6toplamsifre: TFloatField;
Table13: TTable;
Table13CRN: TAutoIncField;
Table13NAME: TStringField;
Table13SNAME: TStringField;
Table13PASS: TStringField;
Table13ADIORTORT: TFloatField;
Table13ADISAPORT: TFloatField;
Table13ADISKEWORT: TFloatField;
Table13ADICURRORT: TFloatField;
Table13SOYORTORT: TFloatField;
Table13SOYSAPORT: TFloatField;
Table13SOYSKEWORT: TFloatField;
Table13SOYCURREORT: TFloatField;
Table13SIFORTORT: TFloatField;
Table13SIFSAPORT: TFloatField;
Table13SIFSKEWORT: TFloatField;
Table13SIFCURREORT: TFloatField;
Table13ADIORTSTSAPMA: TFloatField;
Table13SOYORTSTSAPMA: TFloatField;
Table13SIFORTSTSAPMA: TFloatField;

Table13ADIORTBBOTTOM: TFloatField;
Table13ADIORTBTOP: TFloatField;
Table13SOYORTBBOTTOM: TFloatField;
Table13SOYORTBTOP: TFloatField;
Table13SIFORTBBOTTOM: TFloatField;
Table13SIFORTBTOP: TFloatField;
Table13ADIORTMIN: TFloatField;
Table13ADIORTMAX: TFloatField;
Table13SOYORTMIN: TFloatField;
Table13SOYORTMAX: TFloatField;
Table13SIFORTMIN: TFloatField;
Table13SIFORTMAX: TFloatField;
Query9: TQuery;
Table14: TTable;
Table14CRN: TAutoIncField;
Table14NAME: TStringField;
Table14SNAME: TStringField;
Table14PASS: TStringField;
Table14ADIBASORT: TFloatField;
Table14ADIBASSAP: TFloatField;
Table14ADIBASSKEW: TFloatField;
Table14ADIBASCURR: TFloatField;
Table14SOYBASORT: TFloatField;
Table14SOYBASSAP: TFloatField;
Table14SOYBASSKEW: TFloatField;
Table14SOYBASCURR: TFloatField;
Table14SIFBASORT: TFloatField;
Table14SIFBASSAP: TFloatField;
Table14SIFBASSKEW: TFloatField;
Table14SIFBASCURR: TFloatField;
Table14ENTRANCES: TBooleanField;
Table14SUCCESSEENTRYCOUNT: TFloatField;
Table14LOGINDATE: TDateField;
Table14LOGINTIME: TTimeField;
Table14REALNAME: TStringField;
Table14REALSNAME: TStringField;
Table14SECURITYLEVEL: TFloatField;
Table14FAILENTRYCOUNT: TFloatField;
Table14SUCCESSEENTRYPERCENT: TFloatField;
Query10: TQuery;
Table15: TTable;
Table15CRN: TAutoIncField;
Table15NAME: TStringField;
Table15SNAME: TStringField;
Table15PASS: TStringField;
Table15ADIBASORTORT: TFloatField;
Table15ADIBASSAPORT: TFloatField;
Table15ADIBASSKEWORT: TFloatField;
Table15ADIBASCURRORT: TFloatField;
Table15SOYBASORTORT: TFloatField;
Table15SOYBASSAPORT: TFloatField;
Table15SOYBASSKEWORT: TFloatField;
Table15SOYBASCURRORT: TFloatField;
Table15SIFBASORTORT: TFloatField;
Table15SIFBASSAPORT: TFloatField;
Table15SIFBASSKEWORT: TFloatField;
Table15SIFBASCURRORT: TFloatField;
Table15ADIBASORTSTSAPMA: TFloatField;
Table15SOYBASORTSTSAPMA: TFloatField;
Table15SIFBASORTSTSAPMA: TFloatField;
Table15ADIBASORTBBOTTOM: TFloatField;
Table15ADIBASORTBTOP: TFloatField;

Table15SOYBASORTBBOTTOM: TFloatField;
Table15SOYBASORTBTOP: TFloatField;
Table15SIFBASORTBBOTTOM: TFloatField;
Table15SIFBASORTBTOP: TFloatField;
Table15ADIBASORTMIN: TFloatField;
Table15ADIBASORTMAX: TFloatField;
Table15SOYBASORTMIN: TFloatField;
Table15SOYBASORTMAX: TFloatField;
Table15SIFBASORTMIN: TFloatField;
Table15SIFBASORTMAX: TFloatField;
Query11: TQuery;
Query12: TQuery;
Query13: TQuery;
Table16: TTable;
Table16CRN: TAutoIncField;
Table16NAME: TStringField;
Table16SNAME: TStringField;
Table16PASS: TStringField;
Table16ADIBASORTORT: TFloatField;
Table16ADIBASSAPORT: TFloatField;
Table16ADIBASSKEWORT: TFloatField;
Table16ADIBASCURRORT: TFloatField;
Table16SOYBASORTORT: TFloatField;
Table16SOYBASSAPORT: TFloatField;
Table16SOYBASSKEWORT: TFloatField;
Table16SOYBASCURRORT: TFloatField;
Table16SIFBASORTORT: TFloatField;
Table16SIFBASSAPORT: TFloatField;
Table16SIFBASSKEWORT: TFloatField;
Table16SIFBASCURRORT: TFloatField;
Table16ADIBASORTSTSAPMA: TFloatField;
Table16SOYBASORTSTSAPMA: TFloatField;
Table16SIFBASORTSTSAPMA: TFloatField;
Table16ADIBASORTBBOTTOM: TFloatField;
Table16ADIBASORTBTOP: TFloatField;
Table16SOYBASORTBBOTTOM: TFloatField;
Table16SOYBASORTBTOP: TFloatField;
Table16SIFBASORTBBOTTOM: TFloatField;
Table16SIFBASORTBTOP: TFloatField;
Table16ADIBASORTMIN: TFloatField;
Table16ADIBASORTMAX: TFloatField;
Table16SOYBASORTMIN: TFloatField;
Table16SOYBASORTMAX: TFloatField;
Table16SIFBASORTMIN: TFloatField;
Table16SIFBASORTMAX: TFloatField;
Query13AVERAGEOfsifbascurr: TFloatField;
Query13AVERAGEOfadibasort: TFloatField;
Query13AVERAGEOfadibassap: TFloatField;
Query13AVERAGEOfadibasskew: TFloatField;
Query13AVERAGEOfadibascurr: TFloatField;
Query13AVERAGEOfsoybasort: TFloatField;
Query13AVERAGEOfsoybassap: TFloatField;
Query13AVERAGEOfsoybasskew: TFloatField;
Query13AVERAGEOfsoybascurr: TFloatField;
Query13AVERAGEOfsifbasort: TFloatField;
Query13AVERAGEOfsifbassap: TFloatField;
Query13AVERAGEOfsifbasskew: TFloatField;
Query14: TQuery;
Query15: TQuery;
Query15toplamadi: TFloatField;
Query15toplamsoyadi: TFloatField;
Query15toplamsifre: TFloatField;



Table17: TTable;
Table17NAME: TStringField;
Table17SNAME: TStringField;
Table17PASS: TStringField;
Table17ADIBASORTSTSAPMA: TFloatField;
Table17SOYADIBASORTSTSAPMA: TFloatField;
Table17SIFREBASORTSTSAPMA: TFloatField;
Table17ADIBASSKEWSTSAPMA: TFloatField;
Table17SOYADIBASSKEWSTSAPMA: TFloatField;
Table17SIFREBASSKEWSTSAPMA: TFloatField;
Table17ADIBASCURRSTSAPMA: TFloatField;
Table17SOYADIBASCURRSTSAPMA: TFloatField;
Table17SIFREBASCURRSTSAPMA: TFloatField;
Query16: TQuery;
DataSource2: TDataSource;
Query17: TQuery;
DataSource3: TDataSource;
Query17MAXOFsuccessentrycount: TFloatField;
Query17MAXOFfailentrycount: TFloatField;
RadioGroup1: TRadioGroup;
Label1: TLabel;
Label2: TLabel;
Label6: TLabel;
DBGrid1: TDBGrid;
Label7: TLabel;
Label8: TLabel;
BitBtn3: TBitBtn;
Bevel1: TBevel;
Panel2: TPanel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
DataSource4: TDataSource;
Table18: TTable;
Bevel2: TBevel;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
Label9: TLabel;
Edit4: TEdit;
Table19: TTable;
Table19CRN: TAutoIncField;
Table19NAME: TStringField;
Table19SNAME: TStringField;
Table19PASS: TStringField;
Table19ADIORTORT: TFloatField;
Table19ADISAPORT: TFloatField;
Table19ADISKEWORT: TFloatField;
Table19ADICURRORT: TFloatField;
Table19SOYORTORT: TFloatField;
Table19SOYSAPORT: TFloatField;
Table19SOYSKEWORT: TFloatField;
Table19SOYCURRORT: TFloatField;
Table19SIFORTORT: TFloatField;
Table19SIFSAPORT: TFloatField;
Table19SIFSKEWORT: TFloatField;
Table19SIFCURRORT: TFloatField;
Table19ADIORTSTSAPMA: TFloatField;
Table19SOYORTSTSAPMA: TFloatField;
Table19SIFORTSTSAPMA: TFloatField;



Table19ADIORTBBOTTOM: TFloatField;
Table19ADIORTBTOP: TFloatField;
Table19SOYORTBBOTTOM: TFloatField;
Table19SOYORTBTOP: TFloatField;
Table19SIFORTBBOTTOM: TFloatField;
Table19SIFORTBTOP: TFloatField;
Table19ADIORTMIN: TFloatField;
Table19ADIORTMAX: TFloatField;
Table19SOYORTMIN: TFloatField;
Table19SOYORTMAX: TFloatField;
Table19SIFORTMIN: TFloatField;
Table19SIFORTMAX: TFloatField;
Table20: TTable;
Table20NAME: TStringField;
Table20SNAME: TStringField;
Table20PASS: TStringField;
Table20ADIORTSTSAPMA: TFloatField;
Table20SOYADIORTSTSAPMA: TFloatField;
Table20SIFREORTSTSAPMA: TFloatField;
Table20ADISKEWSTSAPMA: TFloatField;
Table20SOYADISKEWSTSAPMA: TFloatField;
Table20SIFRESKEWSTSAPMA: TFloatField;
Table20ADICURRSTSAPMA: TFloatField;
Table20SOYADICURRSTSAPMA: TFloatField;
Table20SIFRECURRSTSAPMA: TFloatField;
Query18: TQuery;
Query19: TQuery;
Query18ort: TFloatField;
Query18stsapma: TFloatField;
Query19ort: TFloatField;
Query19stsapma: TFloatField;
Query20: TQuery;
Query20ort: TFloatField;
Query20stsapma: TFloatField;
Query18SIRANO: TIntegerField;
Query18NAME: TStringField;
Query18SNAME: TStringField;
Query18PASS: TStringField;
Query18SKEW: TFloatField;
Query18KURTOSIS: TFloatField;
Query19SIRANO: TIntegerField;
Query19NAME: TStringField;
Query19SNAME: TStringField;
Query19PASS: TStringField;
Query19SKEW: TFloatField;
Query19KURTOSIS: TFloatField;
Query20SIRANO: TIntegerField;
Query20NAME: TStringField;
Query20SNAME: TStringField;
Query20PASS: TStringField;
Query20SKEW: TFloatField;
Query20KURTOSIS: TFloatField;
Table2SecurityLevel2: TFloatField;
DataSource5: TDataSource;
DataSource6: TDataSource;
DataSource7: TDataSource;
tbl_tmprname: TTable;
tbl_tmprname: TTable;
tbl_tmppass: TTable;
q_updatekont: TQuery;
tbl_nametime: TTable;
tbl_snametime: TTable;

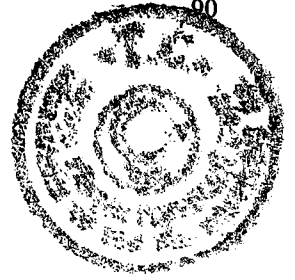


```
tbl_passtime: TTable;
tbl_aralikname: TTable;
tbl_aralikname: TTable;
tbl_aralikpass: TTable;
Button1: TButton;
Query21: TQuery;
procedure GetTimeBas(Sender: TObject);
procedure GetTimeAra(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Edit1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure RadioGroup1Click(Sender: TObject);
procedure DBEdit2Exit(Sender: TObject);
procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);
procedure DBGrid1CellClick(Column: TColumn);
procedure Table1AfterScroll(DataSet: TDataSet);
procedure Table11AfterPost(DataSet: TDataSet);
procedure Update_tbl;
procedure Updatetbl2;
private
  { Private declarations }
  procedure CheckType(Key : Word; Sender:TObject);
  procedure FormClear;
  procedure Clear;
  procedure EntryCalc;
  procedure EntryCalcBasma;
  Function CheckUser : boolean;
  Function GirisKont:boolean;

public
  { Public declarations }
  procedure momskewhesaplama;
  procedure momskewhesaplama2;

end;

var
  Form3: TForm3;
  KAra :integer = 0;
  KDwn :integer=0;
  i:integer = 0;
  j:integer = 0;
  tbl_ind:integer=3;
  tblname_son:integer=0;
  tblsname_son:integer=0;
  tblpasstime_son:integer=0;
  succount,delcount,fark,start : integer;
  basma1:array[0..15] of double;
  aralik1:array[0..15] of double;
  basma2:array[0..15] of double;
  aralik2:array[0..15] of double;
```



```

basma3:array[0..15] of double;
aralik3:array[0..15] of double;
counter:integer = 0;
SUCCESSCOUNT :integer = 0;
FAILcount:integer = 0;
delbackmesaj : boolean;
// adiBBottom , adiBTop : Extended;
AdiM1, AdiM2, adiM3, adiM4, adiSkew, adiKurtosis: Extended;
adiStsapma,adibasStsapma : Extended;
AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew, AdibasKurtosis: Extended;
soyadiM1, soyadiM2, soyadiM3, soyadiM4, soyadiSkew, soyadiKurtosis: Extended;
SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4, SoyAdibasSkew, SoyAdibasKurtosis:
Extended;
soyadiStsapma,soyadibasStsapma : Extended;
sifreM1, sifreM2, sifreM3, sifreM4, sifreSkew, sifreKurtosis: Extended;
SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew, SifrebasKurtosis: Extended;
sifreStsapma,sifrebasStsapma : Extended;
Say : Integer;
implementation
{$R *.DFM}
uses Unit1,Unit2;
procedure TForm3.FormClear;
begin
    Edit1.Text := '';
    Edit2.Text := '';
    Edit3.Text := '';
end;
procedure TForm3.Clear;
begin
    Timer1.Enabled := False;
    Timer2.Enabled := False;
    KDown := 0;
    KAra := 0;
    i:= 0;
    j:= 0;
    tbl_ind:=3;
end;
Function TForm3.CheckUser : boolean;
begin
    with Table1 do
    begin
        First;
        while not EOF do
        begin
            if(Edit1.Text = FieldByName('NAME').AsString)and
            (Edit2.Text = FieldByName('SNAME').AsString)and
            (Edit3.Text = FieldByName('PASS').AsString)
            then begin Result := True; Exit; end;
            Next;
        end;
    end;
    Result := False;
end;

procedure TForm3.CheckType(Key: Word; Sender: TObject);
var
    Say : Integer;
begin
    if(key = VK_DELETE)or(key = VK_BACK)then
    begin
        Showmessage('Delete veya BackSpace Tuşuna Bastınız.Lütfen Tekrar Giriş Yapınız!!!');
    end;
end;

```

```

TEdit(Sender).Text := "";
delbackmesaj := True;
if TEdit(Sender).Name = 'Edit1' then
begin
    For Say := 1 to 15 do begin basma1[say]:= 0; aralik1[say]:= 0;end;
end;
if TEdit(Sender).Name = 'Edit2' then
begin
    For Say := 1 to 15 do begin basma2[say]:= 0; aralik2[say]:= 0;end;
end;
if TEdit(Sender).Name = 'Edit3' then
begin
    For Say := 1 to 15 do begin basma3[say]:= 0; aralik3[say]:= 0;end;
end;
Clear;
end;
end;
procedure TForm3.EntryCalc;
var
    Present: TDateTime;
    Year, Month, Day, Hour, Min, Sec, MSec: Word;
begin
    try
        Present:= Now;
        DecodeDate(Present, Year, Month, Day);
        DecodeTime(Present, Hour, Min, Sec, MSec);
        Table11.Open;
        Table11.Insert;
        Table11ADIORT.AsFloat := AdiM1;
        Table11ADISAP.AsFloat := AdiStSapma;
        Table11ADISKEW.AsFloat := AdiSkew;
        Table11ADICURR.AsFloat := AdikURTOSIS;
        Table11SOYORT.AsFloat := SoyAdiM1;
        Table11SOYSAP.AsFloat := SoyAdiStSapma;
        Table11SOYSkew.AsFloat := SoyAdiSkew;
        Table11SOYcurr.AsFloat := SoyAdiKurtosis;
        Table11SIFORT.AsFloat := SifreM1;
        Table11SIFSAP.AsFloat := SifreStSapma;
        Table11SIFSkew.AsFloat := SifreSkew;
        Table11SIFcurr.AsFloat := SifreKurtosis;
        Table11LOGINDATE.AsDateTime := Present;
        Table11LOGINTIME.AsDateTime := Present;
        Table11.Post;
    finally
        Table11.Close;
    end;
end;

procedure TForm3.EntryCalcBasma;
var
    Present: TDateTime;
    Year, Month, Day, Hour, Min, Sec, MSec: Word;
begin
    try
        Present:= Now;
        DecodeDate(Present, Year, Month, Day);
        DecodeTime(Present, Hour, Min, Sec, MSec);
        table14.Open;
        table14.Insert;
        table14ADIBASORT.AsFloat := AdibasM1;
        table14ADIBASSAP.AsFloat := AdibasStSapma;
        table14ADIBASSKEW.AsFloat := AdibasSkew;
    end;
end;

```

```

table14ADIBASCURR.AsFloat := AdibasKURTOSIS;
table14SOYBASORT.AsFloat := SoyAdibasM1;
table14SOYBASSAP.AsFloat := SoyAdibasStSapma;
table14SOYBASSkew.AsFloat := SoyAdibasSkew;
table14SOYBAScurr.AsFloat := SoyAdibasKurtosis;
table14SIFBASORT.AsFloat := SifrebasM1;
table14SIFBASSAP.AsFloat := SifrebasStSapma;
table14SIFBASSkew.AsFloat := SifrebasSkew;
table14SIFBAScurr.AsFloat := SifrebasKurtosis;
table14LOGINDATE.AsDateTime := Present;
table14LOGINTIME.AsDateTime := Present;
Table14.Post;
finally
Table14.Close;
end;
end;
procedure TForm3.momskewhesaplama;
var
    tmp_aralik1:array of double;
    i,uz:integer;
begin
    MomentSkewKurtosis(aralik1, AdiM1, AdiM2, AdiM3, AdiM4, AdiSkew, AdiKurtosis);
    MomentSkewKurtosis(basma1, AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew,
AdibasKurtosis);
    MomentSkewKurtosis(aralik2, SoyAdiM1, SoyAdiM2, SoyAdiM3, SoyAdiM4, SoyAdiSkew,
SoyAdiKurtosis);
    MomentSkewKurtosis(basma2, SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4,
SoyAdibasSkew, SoyAdibasKurtosis);
    MomentSkewKurtosis(aralik3, SifreM1, SifreM2, SifreM3, SifreM4, SifreSkew, SifreKurtosis);
    MomentSkewKurtosis(basma3, SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew,
SifrebasKurtosis);

    end;
procedure TForm3.momskewhesaplama2;
var
    tmp_aralik1:array of double;
    i,uz:integer;
begin
    uz:=length(edit1.text)-1;
    setlength(tmp_aralik1,uz);
    for i:=1 to uz do tmp_aralik1[i-1]:=aralik1[i];
    MomentSkewKurtosis(tmp_aralik1, AdiM1, AdiM2, AdiM3, AdiM4, AdiSkew, AdiKurtosis);
    tmp_aralik1:=nil;
    uz:=length(edit2.text)-1;
    setlength(tmp_aralik1,uz);
    for i:=1 to uz do tmp_aralik1[i-1]:=aralik2[i];
    MomentSkewKurtosis(tmp_aralik1, SoyAdiM1, SoyAdiM2, SoyAdiM3, SoyAdiM4,
SoyAdiSkew, SoyAdiKurtosis);
    // soyadim1:=mean(tmp_aralik1);
    tmp_aralik1:=nil;
    //
    uz:=length(edit3.text)-1;
    setlength(tmp_aralik1,uz);
    for i:=1 to uz do tmp_aralik1[i-1]:=aralik3[i];
    MomentSkewKurtosis(tmp_aralik1, SifreM1, SifreM2, SifreM3, SifreM4, SifreSkew,
SifreKurtosis);
    // sifrem1:=mean(tmp_aralik1);
    tmp_aralik1:=nil;
end;
procedure TForm3.BitBtn1Click(Sender: TObject);
var
    Say,rec_i,true_cnt : Integer;

```

```

begin
with Table1 do
begin
if Form1.AdmEntry
then //Eyer Admin ise
begin
if(Edit1.Text <> FieldByName('NAME').AsString)or
(Edit2.Text <> FieldByName('SNAME').AsString)or
(Edit3.Text <> FieldByName('PASS').AsString)
then
begin
Showmessage('Sistem Yöneticisi Ayarlarını Değiştiremezsiniz!!!');
Formclear;
Clear;
For Say := 1 to 15 do begin
basma1[say]:= 0; aralik1[say]:= 0;
basma2[say]:= 0; aralik2[say]:= 0;
basma3[say]:= 0; aralik3[say]:= 0;
end; //for

Edit1.SetFocus;
Exit;
end
end//if
else
begin
if not CheckUser
then //kullanıcı yoksa
begin
ShowMessage('Bu kullanıcı hesabı mevcut degil!!!');
Clear;
FormClear;
For Say := 1 to 15 do
begin
basma1[say]:= 0; aralik1[say]:= 0;
basma2[say]:= 0; aralik2[say]:= 0;
basma3[say]:= 0; aralik3[say]:= 0;
end; //for
Edit1.setFocus;
Exit;
end; //else
end;
if Table1.LOCKED.Value = True
then
begin
Showmessage('Sistem tarafından hesabınız kapanmıştır');
Formclear;
Clear;
Edit1.SetFocus;
ModalResult := mrCancel;
Exit;
end;
if(length(Edit1.Text)<2)or(length(Edit2.Text)<2)or(length(Edit3.Text)< 6)
then
begin
Showmessage('Adı ve Soyadı için en az 2,Şifre için ise en az 6 karakterlik veri girmelisiniz.');
```

FormClear;

clear;

Edit1.SetFocus;

Exit;

end

else

```

begin
{
    momskewhesaplama;
//    showmessage('sec: '+floattostr(table2securitylevel.asfloat));
    if ((( (AdiM1 >= (Table10ADIORTORT.AsFloat -
table2securitylevel.asfloat*Table12ADIORTSTSAPMA.ASFLOAT)) AND
(AdiM1 <= (Table10ADIORTORT.AsFloat +
table2securitylevel.asfloat*Table12ADIORTSTSAPMA.ASFLOAT)))) AND
(((soyadiM1 >= (Table10SOYORTORT.AsFloat -
table2securitylevel.asfloat*Table12SOYADIORTSTSAPMA.ASFLOAT)) AND
(soyadiM1 <= (Table10SOYORTORT.AsFloat +
table2securitylevel.asfloat*Table12SOYADIORTSTSAPMA.ASFLOAT)))) AND
(((sifreM1 >= (Table10SIFORTORT.AsFloat -
table2securitylevel.asfloat*Table12SIFREORTSTSAPMA.ASFLOAT)) AND
(sifreM1 <= (Table10SIFORTORT.AsFloat +
table2securitylevel.asfloat*Table12SIFREORTSTSAPMA.ASFLOAT)))) AND
(((AdibasM1 >= (Table3ADIBASORTORT.AsFloat -
table2securitylevel.asfloat*Table4ADIBASORTSTSAPMA.ASFLOAT)) AND
(AdibasM1 <= (Table3ADIBASORTORT.AsFloat +
table2securitylevel.asfloat*Table4ADIBASORTSTSAPMA.ASFLOAT)))) AND
(((soyadibasM1 >= (Table3SOYBASORTORT.AsFloat -
table2securitylevel.asfloat*Table4SOYADIBASORTSTSAPMA.ASFLOAT)) AND
(soyadibasM1 <= (Table3SOYBASORTORT.AsFloat +
table2securitylevel.asfloat*Table4SOYADIBASORTSTSAPMA.ASFLOAT)))) AND
(((sifrebasM1 >= (Table3SIFBASORTORT.AsFloat -
table2securitylevel.asfloat*Table4SIFREBASORTSTSAPMA.ASFLOAT)) AND
(sifrebasM1 <= (Table3SIFBASORTORT.AsFloat +
table2securitylevel.asfloat*Table4SIFREBASORTSTSAPMA.ASFLOAT))))))
then
begin
}
    if GirisKont
    then
    begin
        Showmessage('Kullanıcı bilgileri doğru ve Sisteme girişiniz kabul edildi.');
```

EntryCalc;

EntryCalcBasma;

query1.Close;

query1.ParamByName('NAME').AsString := Table1NAME.AsString;

query1.ParamByName('SNAME').AsString := Table1SNAME.AsString;

query1.ParamByName('PASS').AsString := Table1PASS.AsString;

query1.Prepare;

query1.Open;

Table11.Open;

Table11.Last;

Table11.Edit;

Table11ENTRANCES.AsBoolean := True;

Table11SecurityLevel.asfloat := Table2SecurityLevel.asfloat;

Table11TRAININGNO.asinteger := Table2TRAININGNO.asinteger;

SUCCESSCOUNT := query1maxofsuccessentrycount.asinteger;

SUCCESSCOUNT := SUCCESSCOUNT + 1;

table11successentrycount.AsInteger := SUCCESSCOUNT;

query2.Close;

query2.ParamByName('NAME').AsString := Table1NAME.AsString;

query2.ParamByName('SNAME').AsString := Table1SNAME.AsString;

query2.ParamByName('PASS').AsString := Table1PASS.AsString;

query2.Prepare;

query2.Open;

Failcount := Query2MAXOFfailentrycount.asinteger;

table11FAILENTRYCOUNT.AsInteger := Failcount;

table11successentrypercent.asfloat := successcount/(successcount + failcount);

//realname,realname

table11.FieldByName('realname').asString:=table18.fieldbyname('adi').asString;

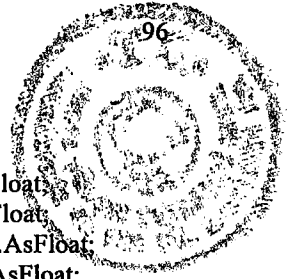
table11.FieldByName('realname').asString:=table18.fieldbyname('soyadi').asString;

```

Table11.post;
Table11.close;
table14.Open;
table14.Last;
table14.Edit;
table14ENTRANCES.AsBoolean := True;
table14SecurityLevel.asfloat := Table2SecurityLevel.asfloat;
// table14TRAININGNO.asinteger := Table2TRAININGNO.asinteger;
SUCCESSCOUNT := query1maxofsuccessentrycount.asinteger;
SUCCESSCOUNT := SUCCESSCOUNT + 1;
table14successentrycount.AsInteger := SUCCESSCOUNT;
table14FAILENTRYCOUNT.AsInteger := Failcount;
table14successentrypercent.asfloat := successcount/(successcount + failcount);
//realname,realsname
table14.FieldByName('realname').asString:=table18.fieldbyname('adi').asString;
table14.FieldByName('realsname').asString:=table18.fieldbyname('soyadi').asString;
table14.post;
table14.close;
Table11.open;
Table11.Last;
succount := table11successentrycount.asinteger;
Table11.close;
//UPDATE BOLUMU
update_tbl;

if ((succount mod (1*table2trainingno.asinteger)) = 0)
then
begin
//update table 2.nd algoritma
updateTbl2;
//
query3.ParamByName('NAME').AsString := Table1NAME.AsString;
query3.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query3.ParamByName('PASS').AsString := Table1PASS.AsString;
query3.execsql;
query10.ParamByName('NAME').AsString := Table1NAME.AsString;
query10.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query10.ParamByName('PASS').AsString := Table1PASS.AsString;
query10.execsql;
query9.ParamByName('NAME').AsString := Table1NAME.AsString;
query9.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query9.ParamByName('PASS').AsString := Table1PASS.AsString;
query9.execsql;
query11.ParamByName('NAME').AsString := Table1NAME.AsString;
query11.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query11.ParamByName('PASS').AsString := Table1PASS.AsString;
query11.execsql;
query7.ParamByName('NAME').AsString := Table1NAME.AsString;
query7.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query7.ParamByName('PASS').AsString := Table1PASS.AsString;
query7.execsql;
query12.ParamByName('NAME').AsString := Table1NAME.AsString;
query12.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query12.ParamByName('PASS').AsString := Table1PASS.AsString;
query12.execsql;
query5.Close;
query5.ParamByName('NAME').AsString := Table1NAME.AsString;
query5.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query5.ParamByName('PASS').AsString := Table1PASS.AsString;
query5.Prepare;
query5.Open;
query13.Close;

```



```
table19.Open;
table19.Insert;
table19ADIORTORT.AsFloat := query5AVERAGEOFadiort.AsFloat;
table19ADISAPORT.AsFloat := query5AVERAGEOFadisap.AsFloat;
table19ADISKEWORT.AsFloat := query5AVERAGEOFadiskew.AsFloat;
table19ADICURRTORT.AsFloat := query5AVERAGEOFadicurr.AsFloat;
table19SOYORTORT.AsFloat := query5AVERAGEOFsoyort.AsFloat;
table19SOYSAPORT.AsFloat := query5AVERAGEOFsoysap.AsFloat;
table19SOYSkewORT.AsFloat := query5AVERAGEOFsoyskew.AsFloat;
table19SOYcurrORT.AsFloat := query5AVERAGEOFsoycurr.AsFloat;
table19SIFORTORT.AsFloat := query5AVERAGEOFsifort.AsFloat;
table19SIFSAPORT.AsFloat := query5AVERAGEOFsifsap.AsFloat;
table19SIFSkewORT.AsFloat := query5AVERAGEOFsifskew.AsFloat;
table19SIFcurrORT.AsFloat := query5AVERAGEOFsifcurr.AsFloat;
table19.Post;
table19.close;
query13.ParamByName('NAME').AsString := Table1NAME.AsString;
query13.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query13.ParamByName('PASS').AsString := Table1PASS.AsString;
query13.Prepare;
query13.Open;
table16.Open;
table16.Insert;
table16ADIBASORTORT.AsFloat := query13AVERAGEOFadibasort.AsFloat;
table16ADIBASSAPORT.AsFloat := query13AVERAGEOFadibassap.AsFloat;
table16ADIBASSKEWORT.AsFloat :=
query13AVERAGEOFadibasskew.AsFloat;
table16ADIBASCURRTORT.AsFloat :=
query13AVERAGEOFadibascurr.AsFloat;
table16SOYBASORTORT.AsFloat := query13AVERAGEOFsoybasort.AsFloat;
table16SOYBASSAPORT.AsFloat := query13AVERAGEOFsoybassap.AsFloat;
table16SOYBASSkewORT.AsFloat :=
query13AVERAGEOFsoybasskew.AsFloat;
table16SOYBAScurrORT.AsFloat := query13AVERAGEOFsoybascurr.AsFloat;
table16SIFBASORTORT.AsFloat := query13AVERAGEOFsifbasort.AsFloat;
table16SIFBASSAPORT.AsFloat := query13AVERAGEOFsifbassap.AsFloat;
table16SIFBASSkewORT.AsFloat := query13AVERAGEOFsifbasskew.AsFloat;
table16SIFBAScurrORT.AsFloat := query13AVERAGEOFsifbascurr.AsFloat;
table16.Post;
table16.close;
query8.ParamByName('NAME').AsString := Table1NAME.AsString;
query8.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query8.ParamByName('PASS').AsString := Table1PASS.AsString;
query8.execsql;
query14.ParamByName('NAME').AsString := Table1NAME.AsString;
query14.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
query14.ParamByName('PASS').AsString := Table1PASS.AsString;
query14.execsql;
Query6.Close;
Query6.ParamByName('NAME').AsString := Table1NAME.AsString;
Query6.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
Query6.ParamByName('PASS').AsString := Table1PASS.AsString;
Query6.Prepare;
Query6.Open;
Table20.Open;
Table20.Insert;
Table20ADIORTSTSAPMA.asfloat := SQRT(Query6toplamadi.Value);
Table20SOYADIORTSTSAPMA.asfloat := SQRT(Query6toplamsoyadi.Value);
Table20SIFREORTSTSAPMA.asfloat := SQRT(Query6toplamsifre.Value);
Table20.Post;
Table20.Close;
Query15.Close;
```

```

Query15.ParamByName('NAME').AsString := Table1NAME.AsString;
Query15.ParamByName('SNAME').AsString := Table1SNAME.AsString;
Query15.ParamByName('PASS').AsString := Table1PASS.AsString;
Query15.Prepare;
Query15.Open;
Table17.Open;
Table17.Insert;
Table17ADIBASORTTSTSAPMA.asfloat := SQRT(Query15toplamadi.Value);
Table17SOYADIBASORTTSTSAPMA.asfloat :=
SQRT(Query15toplamsoyadi.Value);
Table17SIFREBASORTTSTSAPMA.asfloat :=
SQRT(Query15toplamsifre.Value);
Table17.Post;
Table17.Close;
query4.ParamByName('NAME').AsString := Table1NAME.AsString;
query4.ParamByName('SNAME').AsString := Table1SNAME.AsString;
query4.ParamByName('PASS').AsString := Table1PASS.AsString;
query4.execsql;
query16.ParamByName('NAME').AsString := Table1NAME.AsString;
query16.ParamByName('SNAME').AsString := Table1SNAME.AsString;
query16.ParamByName('PASS').AsString := Table1PASS.AsString;
query16.execsql;
end;

ModalResult := Mrok; // Admin ise Sistem Ayarları Formunu Aç
Counter := 0;
FormClear;
CLEAR;
Edit1.SetFocus;
For Say := 1 to 15 do
begin
    basma1[say]:= 0; aralik1[say]:= 0;
    basma2[say]:= 0; aralik2[say]:= 0;
    basma3[say]:= 0; aralik3[say]:= 0;
end; //for
end
// end
else
begin
    EntryCalc;
    EntryCalcBasma;
    query2.Close;
    query2.ParamByName('NAME').AsString := Table1NAME.AsString;
    query2.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    query2.ParamByName('PASS').AsString := Table1PASS.AsString;
    query2.Prepare;
    query2.Open;
    Table11.Open;
    Table11.Last;
    Table11.Edit;
    Table11ENTRANCES.AsBoolean := False;
    Table11SecurityLevel.asfloat := Table2SecurityLevel.asfloat;
    Table11TRAININGNO.asinteger := Table2TRAININGNO.asinteger;
    Failcount := Query2MAXOFFailentrycount.asinteger;
    FAILCOUNT := FAILCOUNT + 1;
    table11FAILENTRYCOUNT.AsInteger := FAILCOUNT;
    query1.Close;
    query1.ParamByName('NAME').AsString := Table1NAME.AsString;
    query1.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    query1.ParamByName('PASS').AsString := Table1PASS.AsString;
    query1.Prepare;
    query1.Open;

```

```

SUCCESSCOUNT := query1maxofsuccessentrycount.asinteger;
table11successentrycount.AsInteger := SUCCESSCOUNT;
table11successentrypercent.asfloat := successcount/(successcount + failcount);
//realname,realsname
table11.FieldName('realname').asString:=table18.fieldbyname('adi').asString;
table11.FieldName('realsname').asString:=table18.fieldbyname('soyadi').asString;
Table11.Post;
Table11.close;
Table14.Open;
Table14.Last;
Table14.Edit;
Table14ENTRANCES.AsBoolean := False;
Table14SecurityLevel.asfloat := Table2SecurityLevel.asfloat;
Failcount := Query2MAXOFFailentrycount.asinteger;
FAILCOUNT := FAILCOUNT + 1;
table14FAILENTRYCOUNT.AsInteger := FAILCOUNT;
SUCCESSCOUNT := query1maxofsuccessentrycount.asinteger;
table14successentrycount.AsInteger := SUCCESSCOUNT;
table14successentrypercent.asfloat := successcount/(successcount + failcount);
//realname,realsname
table14.FieldName('realname').asString:=table18.fieldbyname('adi').asString;
table14.FieldName('realsname').asString:=table18.fieldbyname('soyadi').asString;
Table14.Post;
Table14.close;
Showmessage('Kullanıcı bilgileri doğru ancak Sisteme girişiniz red edildi.Lütfen
tekrar deneyiniz.');
```

```

FormClear;
CLEAR;
Edit1.SetFocus;
Counter := Counter + 1;
if Counter = Table2LockCounter.AsInteger
then
begin
Table1.Edit;
Table1LOCKED.Value := True;
Table1.Post;
Showmessage('Kullanıcı hesabınız sistem tarafından kitlenmiştir');
ModalResult := mrCancel;
Exit;
end;
For Say := 1 to 15 do
begin
basma1[say]:= 0; aralik1[say]:= 0;
basma2[say]:= 0; aralik2[say]:= 0;
basma3[say]:= 0; aralik3[say]:= 0;
end; //for
end;
end; //else
end;
Clear;
Edit1.SetFocus;
end;

procedure TForm3.FormActivate(Sender: TObject);
var
deg:real;
begin
height:=412;
Edit1.SetFocus;
deg:=table2.FieldName('securitylevel2').asfloat;
if deg=2 then radiogroup1.ItemIndex:=0
else if deg=2.25 then radiogroup1.ItemIndex:=1

```

```

else if deg=2.5 then radiogroup1.ItemIndex:=2
else if deg=2.75 then radiogroup1.ItemIndex:=3
else if deg=3 then radiogroup1.ItemIndex:=4;
end;

```

```

procedure TForm3.Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN)or(Key =
VK_TAB)or(Key = VK_SHIFT))then
  begin
    timer1.Enabled:=true;
    timer2.Enabled:=false;
    GetTimeAra(Sender);
    delbackmesaj := False;
  end;
  CheckType(Key,Sender);
end;

```

```

procedure TForm3.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if(Key = #13)then
  begin
    if TEdit(Sender).Name = 'Edit1' then ActiveControl := Edit2;
    if TEdit(Sender).Name = 'Edit2' then ActiveControl := Edit3;
    if TEdit(Sender).Name = 'Edit3' then
    begin
      ActiveControl := bitbtn1;
    end;
    Clear;
  end;
end;

```

```

procedure TForm3.Edit1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN)or(Key =
VK_TAB)or(Key = VK_SHIFT))then
  begin
    Timer1.Enabled := False;
    GetTimeBas(Sender);
  end;
end;

```

```

procedure TForm3.Timer1Timer(Sender: TObject);
begin
  Kdwn:=Kdwn + timer1.interval;
end;

```

```

procedure TForm3.Timer2Timer(Sender: TObject);
begin
  KAra:=KAra + timer2.interval;
end;

```

```

procedure TForm3.GetTimeBas(Sender: TObject);
begin
  if not delbackmesaj then
  begin
    if TEdit(Sender).Name = 'Edit1' then basma1[i] := Kdwn;
    if TEdit(Sender).Name = 'Edit2' then basma2[i] := Kdwn;
    if TEdit(Sender).Name = 'Edit3' then basma3[i] := Kdwn;
    i := i + 1;
  end;

```

```

    KDwn:=0;
    Timer2.Enabled:=True;
end;
end;
procedure TForm3.GetTimeAra(Sender: TObject);
begin
if TEdit(Sender).Name = 'Edit1' then aralik1[j] := KAra;
if TEdit(Sender).Name = 'Edit2' then aralik2[j] := KAra;
if TEdit(Sender).Name = 'Edit3' then aralik3[j] := KAra;
j:= j + 1;
KAra:=0;
Timer2.Enabled:=False;
end;
procedure TForm3.FormCreate(Sender: TObject);
begin
    Timer1.Enabled := False;
    Timer2.Enabled := False;
    KDwn := 0;
    KAra := 0;
    i:= 0;
    j:= 0;
    Table1.Open;
    Table9.Open;
    Table10.Open;
    Table11.Open;
    Table12.Open;
    Table2.Open;
    Table3.Open;
    Table4.Open;
    Table5.Open;
    table18.Open ;
    table19.Open ;
    table20.Open ;
    tbl_tmppname.Open;
    tbl_tmppname.Open;
    tbl_tmppass.Open;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
    formclear;
    clear;
    edit1.setfocus
end;

procedure TForm3.FormShow(Sender: TObject);
begin
    if Form1.AdmEntry
    then
        begin
            Table1.Filter := 'Admin = True';
            Table1.Filtered := True;
        end
    else Table1.Filtered := False;
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
    form3.close;
end;

```

```

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Table1.close;
    Table9.close;
    Table10.close;
    Table11.close;
    Table12.close;
    Table2.close;
    Table3.close;
    Table4.close;
    Table5.close;
    table18.Close;
    tbl_tmpname.close;
    tbl_tmpsname.close;
    tbl_tmppass.close;
end;

```

```

procedure TForm3.Radiogroup1Click(Sender: TObject);
begin
    table2.Close;
    table2.Open;
    table2.edit;
    case radiogroup1.ItemIndex of
        0:table2.FieldByName('securitylevel2').Asfloat:=2;
        1:table2.FieldByName('securitylevel2').Asfloat:=2.25;
        2:table2.FieldByName('securitylevel2').Asfloat:=2.5;
        3:table2.FieldByName('securitylevel2').Asfloat:=2.75;
        4:table2.FieldByName('securitylevel2').Asfloat:=3;
    end;
    table2.Post;
    table2.Close;
    table2.open;
end;

```

```

procedure TForm3.DBEdit2Exit(Sender: TObject);
begin
    if not (datasource4.State in [dsedit]) then table18.edit;
    table18.post;
end;

```

```

procedure TForm3.DBEdit1KeyPress(Sender: TObject; var Key: Char);
begin
    if (key=#13) then begin
        if Tdbedit(sender).Name ='DBEdit1' then activecontrol:=dbedit2 ;
        if Tdbedit(sender).Name ='DBEdit2' then activecontrol:=edit1;
        if Tradiogroup(sender).Name ='radiogroup1' then activecontrol:=dbgrid1;
        if Tdbgrid(sender).Name ='dbgrid1' then activecontrol:=edit1;
    end;
end;

```

```

procedure TForm3.DBGrid1CellClick(Column: TColumn);
begin
    query17.Close;
    query17.ParamByName('NAME').AsString := Table1NAME.AsString;
    query17.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
    query17.ParamByName('PASS').AsString := Table1PASS.AsString;
    query17.Prepare;
    query17.open;
    edit7.text := inttostr(query17.maxofsuccessentrycount.asinteger);
    edit8.text := inttostr(query17.maxoffailentrycount.asinteger);

```

```

edit6.text := inttostr(query17maxoffailentrycount.asinteger +
query17maxofsuccessentrycount.asinteger);
if strtfloat(edit6.text) < 0 then
edit4.text := floattostr((100*query17maxofsuccessentrycount.asinteger)/strtfloat(edit6.text)) else
edit4.text := '0';
end;

```

```

procedure TForm3.Table1AfterScroll(DataSet: TDataSet);
begin
    query17.Close;
    query17.ParamByName('NAME').AsString := Table1NAME.AsString;
    query17.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    query17.ParamByName('PASS').AsString := Table1PASS.AsString;
    query17.Prepare;
    query17.open;
    edit7.text := inttostr(query17maxofsuccessentrycount.asinteger);
    edit8.text := inttostr(query17maxoffailentrycount.asinteger);
    edit6.text := inttostr(query17maxoffailentrycount.asinteger +
query17maxofsuccessentrycount.asinteger);
    if strtfloat(edit6.text) < 0 then
edit4.text := floattostr((100*query17maxofsuccessentrycount.asinteger)/strtfloat(edit6.text)) else
edit4.text := '0';
end;

```

```

procedure TForm3.Table11AfterPost(DataSet: TDataSet);
begin
    query17.Close;
    query17.ParamByName('NAME').AsString := Table1NAME.AsString;
    query17.ParamByName('SNAME').AsString := Table1SNAME.AsString;
    query17.ParamByName('PASS').AsString := Table1PASS.AsString;
    query17.Prepare;
    query17.open;
    edit7.text := inttostr(query17maxofsuccessentrycount.asinteger);
    edit8.text := inttostr(query17maxoffailentrycount.asinteger);
    edit6.text := inttostr(query17maxoffailentrycount.asinteger +
query17maxofsuccessentrycount.asinteger);
    if strtfloat(edit6.text) < 0 then
edit4.text := floattostr((100*query17maxofsuccessentrycount.asinteger)/strtfloat(edit6.text)) else
edit4.text := '0';
end;

```

Function TForm3.GirisKont:boolean;

var

```

    rec_i,true_count:integer;
    name_flg,sname_flg,pass_flg:boolean;
    sec_level,tmp_stsapma:real;

```

begin

```

    momskewhesaplama2;
    sec_level:=table2.fieldbyname('securityLevel2').asinteger;
    //aralikname tabledan aynı isim,soyisim şifreli olanları bul
    with query18 do
    begin
        close;
        ParamByName('NAME').AsString := edit1.text;
        ParamByName('SNAME').AsString := edit2.text;
        ParamByName('PASS').AsString := edit3.text;
        Prepare;
        open;
    end;
    tmp_stsapma:=0;
    //aralıkları karşılaştır.
    rec_i:=1;true_count:=0;

```

```

while (rec_i<=query18.recordcount) and (not query18.IsEmpty) do
begin
// showmessage('adim1: '+floattostr(aralik1[rec_i])+' ort:
'+floattostr(query18.FieldName('ort').asfloat)+' stsapma:
'+floattostr(query18.FieldName('stsapma').asfloat));
if ((aralik1[rec_i]>= (query18.FieldName('ort').asfloat-
sec_level*(query18.FieldName('stsapma').asfloat+tmp_stsapma))) and
(aralik1[rec_i]<=
(query18.FieldName('ort').asfloat+sec_level*(query18.FieldName('stsapma').asfloat+tmp_stsap
ma))))
then inc(true_count);
query18.Next;
inc(rec_i);
end;//while
if (true_count=query18.RecordCount)
then name_flg:=true
else name_flg:=false;

```

//araliksname tabledan aynı isim,soyisim şifreli olanları bul

```

with query19 do
begin
close;
ParamByName('NAME').AsString := edit1.text;
ParamByName('SNAME').AsString := edit2.text;
ParamByName('PASS').AsString := edit3.text;
Prepare;
open;
end;
tmp_stsapma:=0;
//aralıkları karşılaştır.
rec_i:=1;true_count:=0;
while (rec_i<=query19.recordcount) and (not query19.IsEmpty) do
begin
// showmessage('adim1: '+floattostr(aralik2[rec_i])+' ort:
'+floattostr(query19.FieldName('ort').asfloat)+' stsapma:
'+floattostr(query19.FieldName('stsapma').asfloat));
if ((aralik2[rec_i]>= (query19.FieldName('ort').asfloat-
sec_level*(query19.FieldName('stsapma').asfloat+tmp_stsapma))) and
(aralik2[rec_i]<=
(query19.FieldName('ort').asfloat+sec_level*(query19.FieldName('stsapma').asfloat+tmp_stsap
ma))))
then inc(true_count);
query19.Next;
inc(rec_i);
end;//while
if (true_count=query19.RecordCount)
then sname_flg:=true
else sname_flg:=false;

```

//aralikpass tabledan aynı isim,soyisim şifreli olanları bul

```

with query20 do
begin
close;
ParamByName('NAME').AsString := edit1.text;
ParamByName('SNAME').AsString := edit2.text;
ParamByName('PASS').AsString := edit3.text;
Prepare;
open;
end;
tmp_stsapma:=0;

```

```

//aralıkları karşılaştır.
rec_i:=1;true_count:=0;
while (rec_i<=query20.recordcount) and (not query20.IsEmpty) do
begin
  // showmessage('adim1: '+floattostr(aralik3[rec_i])+' ort:
  '+floattostr(query20.FieldName('ort').asfloat)+' stsapma:
  '+floattostr(query20.FieldName('stsapma').asfloat));
  if ((aralik3[rec_i]>= (query20.FieldName('ort').asfloat-
  sec_level*(query20.FieldName('stsapma').asfloat+tmp_stsapma))) and
  (aralik3[rec_i]<=
  (query20.FieldName('ort').asfloat+sec_level*(query20.FieldName('stsapma').asfloat+tmp_stsap
  ma))))
  then inc(true_count);
  query20.Next;
  inc(rec_i);
end;//while
if (true_count=query20.RecordCount)
then pass_flg:=true
else pass_flg:=false;

//result kontrol
if ((name_flg=true) and (sname_flg=true) and (pass_flg=true))
then GirişKont:=true
else GirişKont:=false;

end;

procedure TForm3.Update_tbl;
var
  i:integer;
begin
  if not (tbl_tmppname.State in [dsinsert]) then tbl_tmppname.insert;
  if not (tbl_tmppname.State in [dsinsert]) then tbl_tmppname.insert;
  if not (tbl_tmppass.State in [dsinsert]) then tbl_tmppass.insert;
  for i:=4 to 18 do begin
    tbl_tmppname.edit;tbl_tmppname.edit;tbl_tmppass.edit;
    tbl_tmppname.fieldbyname('name').asstring:=edit1.text;
    tbl_tmppname.fieldbyname('sname').asstring:=edit2.text;
    tbl_tmppname.fieldbyname('pass').asstring:=edit3.text;
    //
    tbl_tmppname.fieldbyname('name').asstring:=edit1.text;
    tbl_tmppname.fieldbyname('sname').asstring:=edit2.text;
    tbl_tmppname.fieldbyname('pass').asstring:=edit3.text;
    //
    tbl_tmppass.fieldbyname('name').asstring:=edit1.text;
    tbl_tmppass.fieldbyname('sname').asstring:=edit2.text;
    tbl_tmppass.fieldbyname('pass').asstring:=edit3.text;

    tbl_tmppname.fields[i].Asfloat:=aralik1[i-4];
    tbl_tmppname.fields[i].Asfloat:=aralik2[i-4];
    tbl_tmppass.fields[i].Asfloat:=aralik3[i-4];
    tbl_tmppname.post;
    tbl_tmppname.post;
    tbl_tmppass.post;
  end;
end;

procedure TForm3.UpdateTbl2;
var
  AdiM1: Extended;
  adiStsapma : Extended;
  soyadiM1: Extended;

```

```

soyadiStsapma : Extended;
sifreM1: Extended;
sifreStsapma : Extended;
dizi1:array of double;
str_uz,indis,ind_a,ind_b,ind_c:integer;
tbl_namei,tbl_snamei,tbl_passi:integer;
begin
    //true sayısı traninig no ya eşitmi kontrol
    //tbl_tmpname
    { with q_updatekont do
        begin
            close;
            clear;
            sql.add('select count(sname) as cnt from tmp_nametime where name=:par1 and sname=:par2 and
pass=:par3');
            params[0].asstring:=edit1.text;
            params[1].asstring:=edit2.text;
            params[2].asstring:=edit3.text;
            open;
        end;}
    // if (q_updatekont.FieldName('cnt').asinteger=table2trainingno.asinteger) then //training noya
    eşit
    // begin
    //
    // //Aralikmomentslerioluştur
    //
    //tbl_aralikname
    with q_updatekont do
        begin
            close;
            clear;
            sql.add('select * from tmp_nametime where name =:par1 and sname =:par2 and pass =:par3');
            parambyname('par1').asstring:=edit1.text;
            parambyname('par2').asstring:=edit2.text;
            parambyname('par3').asstring:=edit3.text;
            open;
        end;
        indis:=4;
        setlength(dizi1,Table2TrainingNo.AsInteger);
        str_uz:=length(q_updatekont.FieldName('name').asstring);
        tbl_aralikname.open;
        tbl_aralikname.Last;
        tbl_namei:=tbl_aralikname.fieldbyname('SIRANO').asinteger;
        tbl_aralikname.close;
        for ind_a:=1 to str_uz-1 do
            begin
                q_updatekont.First;
                for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
                    begin
                        dizi1[ind_b]:=q_updatekont.Fields[indis+ind_a].AsFloat;
                        q_updatekont.next;
                    end;
                adim1:=mean(dizi1);
                Adistsapma:=stddev(dizi1);
                try
                    Tbl_aralikname.open;
                    Tbl_aralikname.insert;
                    Tbl_aralikname.FieldName('name').asstring:=edit1.text;
                    Tbl_aralikname.FieldName('sname').asstring:=edit2.text;
                    Tbl_aralikname.FieldName('pass').asstring:=edit3.text;
                    Tbl_aralikname.FieldName('ORT').asfloat:=AdiM1;
                    Tbl_aralikname.FieldName('STSAPMA').asfloat:=AdiStSapma;

```

```

    Tbl_aralikname.Post;
finally
    Tbl_aralikname.close;
end;
end;
dizi1:=nil;
//
//tbl_araliksname
//
with q_updatekont do
begin
    close;
    SQL.clear;
    sql.add('select * from tmp__sname where name=:par1 and sname=:par2 and pass=:par3');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    open;
end;
indis:=4;
setlength(dizi1, Table2TrainingNo.AsInteger);
str_uz:=length(q_updatekont.FieldByName('sname').asstring);
tbl_araliksname.open;
tbl_araliksname.Last;
tbl_snamei:=tbl_araliksname.fieldbyname('SIRANO').asinteger;
tbl_araliksname.close;
for ind_a:=1 to str_uz-1 do
begin
    q_updatekont.First;
    for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
    begin
        dizi1[ind_b]:=q_updatekont.Fields[indis+ind_a].AsFloat;
        q_updatekont.next;
    end;
    soyadim1:=mean(dizi1);
    soyAdistsapma:=stddev(dizi1);
    try
        Tbl_araliksname.open;
        Tbl_araliksname.insert;
        Tbl_araliksname.FieldByName('name').asstring:=edit1.text;
        Tbl_araliksname.FieldByName('sname').asstring:=edit2.text;
        Tbl_araliksname.FieldByName('pass').asstring:=edit3.text;
        Tbl_araliksname.FieldByName('ORT').asfloat:=soyAdiM1;
        Tbl_araliksname.FieldByName('STSAPMA').asfloat:=soyAdiStSapma;
        Tbl_araliksname.Post;
    finally
        Tbl_araliksname.close;
    end;
end;
dizi1:=nil;
//
//tbl_aralikpass
//
with q_updatekont do
begin
    close;
    sql.clear;
    sql.add('select * from tmp_passtime where name=:par1 and sname=:par2 and pass=:par3');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    open;
end;

```

```

end;
indis:=4;
setlength(dizi1,Table2TrainingNo.AsInteger);
str_uz:=length(q_updatekont.FieldByName('pass').asString);
tbl_aralikpass.open;
tbl_aralikpass.Last;
tbl_passi:=tbl_aralikpass.fieldbyname('SIRANO').asinteger;
tbl_aralikpass.close;
for ind_a:=1 to str_uz-1 do
begin
  q_updatekont.First;
  for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
  begin
    dizi1[ind_b]:=q_updatekont.Fields[indis+ind_a].AsFloat;
    q_updatekont.next;
  end;
  sifrem1:=mean(dizi1);
  sifrestsapma:=stddev(dizi1);
  try
    Tbl_aralikpass.open;
    Tbl_aralikpass.insert;
    Tbl_aralikpass.FieldByName('name').asString:=edit1.text;
    Tbl_aralikpass.FieldByName('sname').asString:=edit2.text;
    Tbl_aralikpass.FieldByName('pass').asString:=edit3.text;
    Tbl_aralikpass.FieldByName('ORT').asfloat:=sifreM1;
    Tbl_aralikpass.FieldByName('STSAPMA').asfloat:=sifreStSapma;
    Tbl_aralikpass.Post;
  finally
    Tbl_aralikpass.close;
  end;
end;
dizi1:=nil;
//update okey
//tbl_araliknameden sil
//
tbl_namei:=176;tbl_snamei:=169;tbl_passi:=173;
with q_updatekont do
begin
  close;
  sql.clear;
  sql.add('delete from aralikname where name=:par1 and sname=:par2 and pass=:par3 and
SIRANO<=:par4');
  parambyname('par1').asString:=edit1.text;
  parambyname('par2').asString:=edit2.text;
  parambyname('par3').asString:=edit3.text;
  parambyname('par4').asinteger:=tbl_namei;
  execsql;
end;
//tbl_araliksnameden sil
with q_updatekont do
begin
  close;
  sql.clear;
  sql.add('delete from araliksname where name=:par1 and sname=:par2 and pass=:par3 and
SIRANO<=:par4');
  parambyname('par1').asString:=edit1.text;
  parambyname('par2').asString:=edit2.text;
  parambyname('par3').asString:=edit3.text;
  parambyname('par4').asinteger:=tbl_snamei;
  execsql;
end;
//tbl_aralikpass sil
with q_updatekont do

```

```

begin
    close;
    sql.clear;
    sql.add('delete from aralikpass where name=:par1 and sname=:par2 and pass=:par3 and
SIRANO<=:par4');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    parambyname('par4').asinteger:=tbl_passi;
    execsql;
end;
//tbl_tmpname den sil
with q_updatekont do
begin
    close;
    sql.clear;
    sql.add('delete from tmp_nametime where name=:par1 and sname=:par2 and pass=:par3');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    execsql;
end;
//tbl_tmpsname den sil
with q_updatekont do
begin
    close;
    sql.clear;
    sql.add('delete from tmp_snametime where name=:par1 and sname=:par2 and pass=:par3');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    execsql;
end;
//tbl_tmppass den sil
with q_updatekont do
begin
    close;
    sql.clear;
    sql.add('delete from tmp_passtime where name=:par1 and sname=:par2 and pass=:par3');
    parambyname('par1').asstring:=edit1.text;
    parambyname('par2').asstring:=edit2.text;
    parambyname('par3').asstring:=edit3.text;
    execsql;
end;
// end;
end;
end.

```

UNIT4

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, Db, DBTables, Buttons, StdCtrls, ComCtrls, Math;

type

TForm4 = class(TForm)
Label1: TLabel;

```

label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
ListBox1: TListBox;
ListBox2: TListBox;
Label9: TLabel;
Label8: TLabel;
ProgressBar1: TProgressBar;
Label7: TLabel;
Button1: TButton;
Button2: TButton;
Table1: TTable;
Table1NAME: TStringField;
Table1SNAME: TStringField;
Table1PASS: TStringField;
Table1LOCKED: TBooleanField;
DataSource1: TDataSource;
Query2: TQuery;
Query2AVERAGEOFadiort: TFloatField;
Query2AVERAGEOFadisap: TFloatField;
Query2AVERAGEOFadiskew: TFloatField;
Query2AVERAGEOFadicurr: TFloatField;
Query2AVERAGEOFsoyort: TFloatField;
Query2AVERAGEOFsoysap: TFloatField;
Query2AVERAGEOFsoyskew: TFloatField;
Query2AVERAGEOFsoycurr: TFloatField;
Query2AVERAGEOFsifort: TFloatField;
Query2AVERAGEOFsifsap: TFloatField;
Query2AVERAGEOFsifskew: TFloatField;
Query2AVERAGEOFsifcurr: TFloatField;
Table9: TTable;
Table9CRN: TAutoIncField;
Table9NAME: TStringField;
Table9SNAME: TStringField;
Table9PASS: TStringField;
Table9ADIORT: TFloatField;
Table9ADISAP: TFloatField;
Table9ADISKEW: TFloatField;
Table9ADICURR: TFloatField;
Table9SOYORT: TFloatField;
Table9SOYSAP: TFloatField;
Table9SOYSKEW: TFloatField;
Table9SOYCURR: TFloatField;
Table9SIFORT: TFloatField;
Table9SIFSAP: TFloatField;
Table9SIFSKEW: TFloatField;
Table9SIFCURR: TFloatField;
Table9farkkare: TFloatField;
Timer1: TTimer;
Timer2: TTimer;
DataSource2: TDataSource;
Table10: TTable;
Table10CRN: TAutoIncField;
Table10NAME: TStringField;
Table10SNAME: TStringField;
Table10PASS: TStringField;
Table10ADIORTORT: TFloatField;
Table10ADISAPORT: TFloatField;

```

Table10ADISKEWORT: TFloatField;
Table10ADICURRORT: TFloatField;
Table10SOYORTORT: TFloatField;
Table10SOYSAPORT: TFloatField;
Table10SOYSKEWORT: TFloatField;
Table10SOYCURRORT: TFloatField;
Table10SIFORTORT: TFloatField;
Table10SIFSAPORT: TFloatField;
Table10SIFSKEWORT: TFloatField;
Table10SIFCURRORT: TFloatField;
Table10ADIORTSTSAPMA: TFloatField;
Table10SOYORTSTSAPMA: TFloatField;
Table10SIFORTSTSAPMA: TFloatField;
Table10ADIORTBBOTTOM: TFloatField;
Table10ADIORTBTOP: TFloatField;
Table10SOYORTBBOTTOM: TFloatField;
Table10SOYORTBTOP: TFloatField;
Table10SIFORTBBOTTOM: TFloatField;
Table10SIFORTBTOP: TFloatField;
Table12: TTable;
Table12NAME: TStringField;
Table12SNAME: TStringField;
Table12PASS: TStringField;
Table12ADIORTSTSAPMA: TFloatField;
Table12SOYADIORTSTSAPMA: TFloatField;
Table12SIFREORTSTSAPMA: TFloatField;
Query6: TQuery;
Query6toplamadi: TFloatField;
Query6toplamsoyadi: TFloatField;
Query6toplamsifre: TFloatField;
Label6: TLabel;
Table2: TTable;
Table2LockCounter: TFloatField;
Table2TrainingNo: TFloatField;
Table12ADISKEWSTSAPMA: TFloatField;
Table12SOYADISKEWSTSAPMA: TFloatField;
Table12SIFRESKEWSTSAPMA: TFloatField;
Table12ADICURRSTSAPMA: TFloatField;
Table12SOYADICURRSTSAPMA: TFloatField;
Table12SIFRECURRSTSAPMA: TFloatField;
Table10ADIORTMIN: TFloatField;
Table10ADIORTMAX: TFloatField;
Table10SOYORTMIN: TFloatField;
Table10SOYORTMAX: TFloatField;
Table10SIFORTMIN: TFloatField;
Table10SIFORTMAX: TFloatField;
Table1ADMIN: TBooleanField;
Table3: TTable;
Table4: TTable;
Table4CRN: TAutoIncField;
Table4NAME: TStringField;
Table4SNAME: TStringField;
Table4PASS: TStringField;
Table4ADIBASORT: TFloatField;
Table4ADIBASSAP: TFloatField;
Table4ADIBASSKEW: TFloatField;
Table4ADIBASCURR: TFloatField;
Table4SOYBASORT: TFloatField;
Table4SOYBASSAP: TFloatField;
Table4SOYBASSKEW: TFloatField;
Table4SOYBASCURR: TFloatField;
Table4SIFBASORT: TFloatField;

Table4SIFBASSAP: TFloatField;
 Table4SIFBASSKEW: TFloatField;
 Table4SIFBASCURR: TFloatField;
 Query1: TQuery;
 Query1AVERAGEOFadibasort: TFloatField;
 Query1AVERAGEOFadibassap: TFloatField;
 Query1AVERAGEOFadibasskew: TFloatField;
 Query1AVERAGEOFadibascurr: TFloatField;
 Query1AVERAGEOFsoybasort: TFloatField;
 Query1AVERAGEOFsoybassap: TFloatField;
 Query1AVERAGEOFsoybasskew: TFloatField;
 Query1AVERAGEOFsoybascurr: TFloatField;
 Query1AVERAGEOFsifbasort: TFloatField;
 Query1AVERAGEOFsifbassap: TFloatField;
 Query1AVERAGEOFsifbasskew: TFloatField;
 Query1AVERAGEOFsifbascurr: TFloatField;
 Table5: TTable;
 Table5CRN: TAutoIncField;
 Table5NAME: TStringField;
 Table5SNAME: TStringField;
 Table5PASS: TStringField;
 Table5ADIBASORTORT: TFloatField;
 Table5ADIBASSAPORT: TFloatField;
 Table5ADIBASSKEWORT: TFloatField;
 Table5ADIBASCURRORT: TFloatField;
 Table5SOYBASORTORT: TFloatField;
 Table5SOYBASSAPORT: TFloatField;
 Table5SOYBASSKEWORT: TFloatField;
 Table5SOYBASCURRORT: TFloatField;
 Table5SIFBASORTORT: TFloatField;
 Table5SIFBASSAPORT: TFloatField;
 Table5SIFBASSKEWORT: TFloatField;
 Table5SIFBASCURRORT: TFloatField;
 Table5ADIBASORTSTSAPMA: TFloatField;
 Table5SOYBASORTSTSAPMA: TFloatField;
 Table5SIFBASORTSTSAPMA: TFloatField;
 Table5ADIBASORTBBOTTOM: TFloatField;
 Table5ADIBASORTBTOP: TFloatField;
 Table5SOYBASORTBBOTTOM: TFloatField;
 Table5SOYBASORTBTOP: TFloatField;
 Table5SIFBASORTBBOTTOM: TFloatField;
 Table5SIFBASORTBTOP: TFloatField;
 Table5ADIBASORTMIN: TFloatField;
 Table5ADIBASORTMAX: TFloatField;
 Table5SOYBASORTMIN: TFloatField;
 Table5SOYBASORTMAX: TFloatField;
 Table5SIFBASORTMIN: TFloatField;
 Table5SIFBASORTMAX: TFloatField;
 Table3NAME: TStringField;
 Table3SNAME: TStringField;
 Table3PASS: TStringField;
 Table3ADIBASORTSTSAPMA: TFloatField;
 Table3SOYADIBASORTSTSAPMA: TFloatField;
 Table3SIFREBASORTSTSAPMA: TFloatField;
 Table3ADIBASSKEWSTSAPMA: TFloatField;
 Table3SOYADIBASSKEWSTSAPMA: TFloatField;
 Table3SIFREBASSKEWSTSAPMA: TFloatField;
 Table3ADIBASCURRSTSAPMA: TFloatField;
 Table3SOYADIBASCURRSTSAPMA: TFloatField;
 Table3SIFREBASCURRSTSAPMA: TFloatField;
 Query3: TQuery;
 Query3toplamadibas: TFloatField;

Query3toplamsoyadibas: TFloatField;
Query3toplamsifrebas: TFloatField;
Query4: TQuery;
Query5: TQuery;
Button3: TButton;
tbl_nametime: TTable;
tbl_snametime: TTable;
tbl_passtime: TTable;
tbl_nametimeTNN: TAutoIncField;
tbl_nametimeNAME: TStringField;
tbl_nametimeSNAME: TStringField;
tbl_nametimePASS: TStringField;
tbl_nametimeKARA01: TFloatField;
tbl_nametimeKARA02: TFloatField;
tbl_nametimeKARA03: TFloatField;
tbl_nametimeKARA04: TFloatField;
tbl_nametimeKARA05: TFloatField;
tbl_nametimeKARA06: TFloatField;
tbl_nametimeKARA07: TFloatField;
tbl_nametimeKARA08: TFloatField;
tbl_nametimeKARA09: TFloatField;
tbl_nametimeKARA10: TFloatField;
tbl_nametimeKARA11: TFloatField;
tbl_nametimeKARA12: TFloatField;
tbl_nametimeKARA13: TFloatField;
tbl_nametimeKARA14: TFloatField;
tbl_nametimeKARA15: TFloatField;
tbl_snametimeTSN: TAutoIncField;
tbl_snametimeNAME: TStringField;
tbl_snametimeSNAME: TStringField;
tbl_snametimePASS: TStringField;
tbl_snametimeKARA01: TFloatField;
tbl_snametimeKARA02: TFloatField;
tbl_snametimeKARA03: TFloatField;
tbl_snametimeKARA04: TFloatField;
tbl_snametimeKARA05: TFloatField;
tbl_snametimeKARA06: TFloatField;
tbl_snametimeKARA07: TFloatField;
tbl_snametimeKARA08: TFloatField;
tbl_snametimeKARA09: TFloatField;
tbl_snametimeKARA10: TFloatField;
tbl_snametimeKARA11: TFloatField;
tbl_snametimeKARA12: TFloatField;
tbl_snametimeKARA13: TFloatField;
tbl_snametimeKARA14: TFloatField;
tbl_snametimeKARA15: TFloatField;
tbl_passtimeTPN: TAutoIncField;
tbl_passtimeNAME: TStringField;
tbl_passtimeSNAME: TStringField;
tbl_passtimePASS: TStringField;
tbl_passtimeKARA01: TFloatField;
tbl_passtimeKARA02: TFloatField;
tbl_passtimeKARA03: TFloatField;
tbl_passtimeKARA04: TFloatField;
tbl_passtimeKARA05: TFloatField;
tbl_passtimeKARA06: TFloatField;
tbl_passtimeKARA07: TFloatField;
tbl_passtimeKARA08: TFloatField;
tbl_passtimeKARA09: TFloatField;
tbl_passtimeKARA10: TFloatField;
tbl_passtimeKARA11: TFloatField;
tbl_passtimeKARA12: TFloatField;

tbl_passtimeKARA13: TFloatField;
tbl_passtimeKARA14: TFloatField;
tbl_passtimeKARA15: TFloatField;
tbl_aralikname: TTable;
tbl_araliknameSIRANO: TAutoIncField;
tbl_araliknameNAME: TStringField;
tbl_araliknameSNAME: TStringField;
tbl_araliknamePASS: TStringField;
tbl_araliknameORT: TFloatField;
tbl_araliknameSTSAPMA: TFloatField;
tbl_araliknameSKEW: TFloatField;
tbl_araliknameKURTOSIS: TFloatField;
tbl_araliksname: TTable;
tbl_araliksnameSIRANO: TAutoIncField;
tbl_araliksnameNAME: TStringField;
tbl_araliksnameSNAME: TStringField;
tbl_araliksnamePASS: TStringField;
tbl_araliksnameORT: TFloatField;
tbl_araliksnameSTSAPMA: TFloatField;
tbl_araliksnameSKEW: TFloatField;
tbl_araliksnameKURTOSIS: TFloatField;
tbl_aralikpass: TTable;
tbl_aralikpassSIRANO: TAutoIncField;
tbl_aralikpassNAME: TStringField;
tbl_aralikpassSNAME: TStringField;
tbl_aralikpassPASS: TStringField;
tbl_aralikpassORT: TFloatField;
tbl_aralikpassSTSAPMA: TFloatField;
tbl_aralikpassSKEW: TFloatField;
tbl_aralikpassKURTOSIS: TFloatField;
tbl_namecalc: TTable;
tbl_namecalcNCRN: TAutoIncField;
tbl_namecalcNAME: TStringField;
tbl_namecalcSNAME: TStringField;
tbl_namecalcPASS: TStringField;
tbl_namecalcNAMEARA1ORT: TFloatField;
tbl_namecalcNAMEARA2ORT: TFloatField;
tbl_namecalcNAMEARA3ORT: TFloatField;
tbl_namecalcNAMEARA4ORT: TFloatField;
tbl_namecalcNAMEARA5ORT: TFloatField;
tbl_namecalcNAMEARA6ORT: TFloatField;
tbl_namecalcNAMEARA7ORT: TFloatField;
tbl_namecalcNAMEARA8ORT: TFloatField;
tbl_namecalcNAMEARA9ORT: TFloatField;
tbl_namecalcNAMEARA10ORT: TFloatField;
tbl_namecalcNAMEARA11ORT: TFloatField;
tbl_namecalcNAMEARA12ORT: TFloatField;
tbl_namecalcNAMEARA13ORT: TFloatField;
tbl_namecalcNAMEARA14ORT: TFloatField;
tbl_namecalcNAMEARA15ORT: TFloatField;
tbl_namecalcNAMEARA1STSAPMA: TFloatField;
tbl_namecalcNAMEARA2STSAPMA: TFloatField;
tbl_namecalcNAMEARA3STSAPMA: TFloatField;
tbl_namecalcNAMEARA4STSAPMA: TFloatField;
tbl_namecalcNAMEARA5STSAPMA: TFloatField;
tbl_namecalcNAMEARA6STSAPMA: TFloatField;
tbl_namecalcNAMEARA7STSAPMA: TFloatField;
tbl_namecalcNAMEARA8STSAPMA: TFloatField;
tbl_namecalcNAMEARA9STSAPMA: TFloatField;
tbl_namecalcNAMEARA10STSAPMA: TFloatField;
tbl_namecalcNAMEARA11STSAPMA: TFloatField;
tbl_namecalcNAMEARA12STSAPMA: TFloatField;

tbl_namecalcNAMEARA13STSAPMA: TFloatField;
tbl_namecalcNAMEARA14STSAPMA: TFloatField;
tbl_namecalcNAMEARA15STSAPMA: TFloatField;
tbl_namecalcNAMEARA1SKEW: TFloatField;
tbl_namecalcNAMEARA2SKEW: TFloatField;
tbl_namecalcNAMEARA3SKEW: TFloatField;
tbl_namecalcNAMEARA4SKEW: TFloatField;
tbl_namecalcNAMEARA5SKEW: TFloatField;
tbl_namecalcNAMEARA6SKEW: TFloatField;
tbl_namecalcNAMEARA7SKEW: TFloatField;
tbl_namecalcNAMEARA8SKEW: TFloatField;
tbl_namecalcNAMEARA9SKEW: TFloatField;
tbl_namecalcNAMEARA10SKEW: TFloatField;
tbl_namecalcNAMEARA11SKEW: TFloatField;
tbl_namecalcNAMEARA12SKEW: TFloatField;
tbl_namecalcNAMEARA13SKEW: TFloatField;
tbl_namecalcNAMEARA14SKEW: TFloatField;
tbl_namecalcNAMEARA15SKEW: TFloatField;
tbl_namecalcNAMEARA1KURT: TFloatField;
tbl_namecalcNAMEARA2KURT: TFloatField;
tbl_namecalcNAMEARA3KURT: TFloatField;
tbl_namecalcNAMEARA4KURT: TFloatField;
tbl_namecalcNAMEARA5KURT: TFloatField;
tbl_namecalcNAMEARA6KURT: TFloatField;
tbl_namecalcNAMEARA7KURT: TFloatField;
tbl_namecalcNAMEARA8KURT: TFloatField;
tbl_namecalcNAMEARA9KURT: TFloatField;
tbl_namecalcNAMEARA10KURT: TFloatField;
tbl_namecalcNAMEARA11KURT: TFloatField;
tbl_namecalcNAMEARA12KURT: TFloatField;
tbl_namecalcNAMEARA13KURT: TFloatField;
tbl_namecalcNAMEARA14KURT: TFloatField;
tbl_namecalcNAMEARA15KURT: TFloatField;
tbl_snamecalc: TTable;
tbl_snamecalcSNCRN: TAutoIncField;
tbl_snamecalcNAME: TStringField;
tbl_snamecalcSNAME: TStringField;
tbl_snamecalcPASS: TStringField;
tbl_snamecalcSNAMEARA1ORT: TFloatField;
tbl_snamecalcSNAMEARA2ORT: TFloatField;
tbl_snamecalcSNAMEARA3ORT: TFloatField;
tbl_snamecalcSNAMEARA4ORT: TFloatField;
tbl_snamecalcSNAMEARA5ORT: TFloatField;
tbl_snamecalcSNAMEARA6ORT: TFloatField;
tbl_snamecalcSNAMEARA7ORT: TFloatField;
tbl_snamecalcSNAMEARA8ORT: TFloatField;
tbl_snamecalcSNAMEARA9ORT: TFloatField;
tbl_snamecalcSNAMEARA10ORT: TFloatField;
tbl_snamecalcSNAMEARA11ORT: TFloatField;
tbl_snamecalcSNAMEARA12ORT: TFloatField;
tbl_snamecalcSNAMEARA13ORT: TFloatField;
tbl_snamecalcSNAMEARA14ORT: TFloatField;
tbl_snamecalcSNAMEARA15ORT: TFloatField;
tbl_snamecalcSNAMEARA1STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA2STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA3STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA4STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA5STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA6STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA7STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA8STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA9STSAPMA: TFloatField;

tbl_snamecalcSNAMEARA10STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA11STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA12STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA13STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA14STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA15STSAPMA: TFloatField;
tbl_snamecalcSNAMEARA1SKEW: TFloatField;
tbl_snamecalcSNAMEARA2SKEW: TFloatField;
tbl_snamecalcSNAMEARA3SKEW: TFloatField;
tbl_snamecalcSNAMEARA4SKEW: TFloatField;
tbl_snamecalcSNAMEARA5SKEW: TFloatField;
tbl_snamecalcSNAMEARA6SKEW: TFloatField;
tbl_snamecalcSNAMEARA7SKEW: TFloatField;
tbl_snamecalcSNAMEARA8SKEW: TFloatField;
tbl_snamecalcSNAMEARA9SKEW: TFloatField;
tbl_snamecalcSNAMEARA10SKEW: TFloatField;
tbl_snamecalcSNAMEARA11SKEW: TFloatField;
tbl_snamecalcSNAMEARA12SKEW: TFloatField;
tbl_snamecalcSNAMEARA13SKEW: TFloatField;
tbl_snamecalcSNAMEARA14SKEW: TFloatField;
tbl_snamecalcSNAMEARA15SKEW: TFloatField;
tbl_snamecalcSNAMEARA1KURT: TFloatField;
tbl_snamecalcSNAMEARA2KURT: TFloatField;
tbl_snamecalcSNAMEARA3KURT: TFloatField;
tbl_snamecalcSNAMEARA4KURT: TFloatField;
tbl_snamecalcSNAMEARA5KURT: TFloatField;
tbl_snamecalcSNAMEARA6KURT: TFloatField;
tbl_snamecalcSNAMEARA7KURT: TFloatField;
tbl_snamecalcSNAMEARA8KURT: TFloatField;
tbl_snamecalcSNAMEARA9KURT: TFloatField;
tbl_snamecalcSNAMEARA10KURT: TFloatField;
tbl_snamecalcSNAMEARA11KURT: TFloatField;
tbl_snamecalcSNAMEARA12KURT: TFloatField;
tbl_snamecalcSNAMEARA13KURT: TFloatField;
tbl_snamecalcSNAMEARA14KURT: TFloatField;
tbl_snamecalcSNAMEARA15KURT: TFloatField;
tbl_passcalc: TTable;
tbl_passcalcPCRN: TAutoIncField;
tbl_passcalcNAME: TStringField;
tbl_passcalcSNAME: TStringField;
tbl_passcalcPASS: TStringField;
tbl_passcalcPARA1ORT: TFloatField;
tbl_passcalcPARA2ORT: TFloatField;
tbl_passcalcPARA3ORT: TFloatField;
tbl_passcalcPARA4ORT: TFloatField;
tbl_passcalcPARA5ORT: TFloatField;
tbl_passcalcPARA6ORT: TFloatField;
tbl_passcalcPARA7ORT: TFloatField;
tbl_passcalcPARA8ORT: TFloatField;
tbl_passcalcPARA9ORT: TFloatField;
tbl_passcalcPARA10ORT: TFloatField;
tbl_passcalcPARA11ORT: TFloatField;
tbl_passcalcPARA12ORT: TFloatField;
tbl_passcalcPARA13ORT: TFloatField;
tbl_passcalcPARA14ORT: TFloatField;
tbl_passcalcPARA15ORT: TFloatField;
tbl_passcalcPARA1STSAPMA: TFloatField;
tbl_passcalcPARA2STSAPMA: TFloatField;
tbl_passcalcPARA3STSAPMA: TFloatField;
tbl_passcalcPARA4STSAPMA: TFloatField;
tbl_passcalcPARA5STSAPMA: TFloatField;
tbl_passcalcPARA6STSAPMA: TFloatField;

```

tbl_passcalcPARA7STSAPMA: TFloatField;
tbl_passcalcPARA8STSAPMA: TFloatField;
tbl_passcalcPARA9STSAPMA: TFloatField;
tbl_passcalcPARA10STSAPMA: TFloatField;
tbl_passcalcPARA11STSAPMA: TFloatField;
tbl_passcalcPARA12STSAPMA: TFloatField;
tbl_passcalcPARA13STSAPMA: TFloatField;
tbl_passcalcPARA14STSAPMA: TFloatField;
tbl_passcalcPARA15STSAPMA: TFloatField;
tbl_passcalcPARA1SKEW: TFloatField;
tbl_passcalcPARA2SKEW: TFloatField;
tbl_passcalcPARA3SKEW: TFloatField;
tbl_passcalcPARA4SKEW: TFloatField;
tbl_passcalcPARA5SKEW: TFloatField;
tbl_passcalcPARA6SKEW: TFloatField;
tbl_passcalcPARA7SKEW: TFloatField;
tbl_passcalcPARA8SKEW: TFloatField;
tbl_passcalcPARA9SKEW: TFloatField;
tbl_passcalcPARA10SKEW: TFloatField;
tbl_passcalcPARA11SKEW: TFloatField;
tbl_passcalcPARA12SKEW: TFloatField;
tbl_passcalcPARA13SKEW: TFloatField;
tbl_passcalcPARA14SKEW: TFloatField;
tbl_passcalcPARA15SKEW: TFloatField;
tbl_passcalcPARA1KURT: TFloatField;
tbl_passcalcPARA2KURT: TFloatField;
tbl_passcalcPARA3KURT: TFloatField;
tbl_passcalcPARA4KURT: TFloatField;
tbl_passcalcPARA5KURT: TFloatField;
tbl_passcalcPARA6KURT: TFloatField;
tbl_passcalcPARA7KURT: TFloatField;
tbl_passcalcPARA8KURT: TFloatField;
tbl_passcalcPARA9KURT: TFloatField;
tbl_passcalcPARA10KURT: TFloatField;
tbl_passcalcPARA11KURT: TFloatField;
tbl_passcalcPARA12KURT: TFloatField;
tbl_passcalcPARA13KURT: TFloatField;
tbl_passcalcPARA14KURT: TFloatField;
tbl_passcalcPARA15KURT: TFloatField;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure GetTimeBas(Sender: TObject);
procedure GetTimeAra(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Edit1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Table9CalcFields(DataSet: TDataSet);
procedure Button3Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
  procedure CheckType(Key : Word; Sender:TObject);
  procedure FormClear;
  procedure Clear;
  function CheckUser : boolean;
  procedure stsapmahesaplama;
  procedure TimeTblRowDelete(sender:TObject);

```

```

procedure TimeTotalTblDelete(sender:TObject);
procedure Aralikmoments;

public
{ Public declarations }
procedure momenthesaplama;
procedure ortalamahesaplama;

end;

var
  Form4: TForm4;
  KAra :integer=0;
  KDwn :integer=0;
  i:integer=0;
  j:integer=0;
  tbl_ind:integer=3;
  tblname_son:integer=0;
  tblsname_son:integer=0;
  tblpasstime_son:integer=0;
  basma1:array[0..14] of double;
  aralik1:array[0..14] of double;
  basma2:array[0..14] of double;
  aralik2:array[0..14] of double;
  basma3:array[0..14] of double;
  aralik3:array[0..14] of double;
  count:integer=0;
  delbackmesaj : boolean;
//  adiBBottom , adiBTop, AdiM1 : Extended;
//  adiortstsap:double =0;

implementation

{$R *.DFM}

uses Unit1,Unit2,Unit3;

// var
//  aditoplam,aditoplamort,adistsapma:double;

procedure TForm4.FormClear;
begin
  Edit1.Text := '';
  Edit2.Text := '';
  Edit3.Text := '';
  listBox1.Clear;
  listBox2.Clear;
end;

procedure TForm4.Clear;
begin
  ListBox1.Items.Clear;
  ListBox2.Items.Clear;
  Timer1.Enabled := False;
  Timer2.Enabled := False;
  KDwn := 0;
  KAra := 0;
  i:= 0;
  j:= 0;

```



```
tbl_ind:=3;
end;
procedure TForm4.Button2Click(Sender: TObject);
begin
    formclear;
    clear;
    edit1.setfocus;
    Timetblrowdelete(sender);
    // ModalResult := mrCancel;
end;
```

```
Function TForm4.CheckUser : boolean;
begin
    with Table1 do
    begin
        First;
        while not EOF do
        begin
            if(Edit1.Text = FieldByName('NAME').AsString)and
            (Edit2.Text = FieldByName('SNAME').AsString)and
            (Edit3.Text = FieldByName('PASS').AsString)
            then
            begin
                Result := False;
                Exit;
            end;
            Next;
        end;
    end;
    Result := True;
end;
```

```
procedure TForm4.stsapmahesaplama;
begin
    Query6.Close;
    Query6.ParamByName('NAME').AsString := Table1NAME.AsString;
    Query6.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
    Query6.ParamByName('PASS').AsString := Table1PASS.AsString;
    Query6.Prepare;
    Query6.Open;
    Table12.Open;
    Table12.Insert;
    Table12ADIORTSTSAPMA.asfloat := SQRT(Query6toplamadi.Value);
    Table12SOYADIORTSTSAPMA.asfloat := SQRT(Query6toplamsoyadi.Value);
    Table12SIFREORTSTSAPMA.asfloat := SQRT(Query6toplamsifre.Value);
    Table12.Post;
    Table12.Close;
    query3.Close;
    query3.ParamByName('NAME').AsString := Table1NAME.AsString;
    query3.ParamByName('SNAME').AsString :=Table1SNAME.AsString;
    query3.ParamByName('PASS').AsString := Table1PASS.AsString;
    query3.Prepare;
    query3.Open;
    table3.Open;
    table3.Insert;
    table3ADIBasORTSTSAPMA.asfloat := SQRT(query3toplamadibas.Value);
    table3SOYADIBasORTSTSAPMA.asfloat := SQRT(query3toplamsoyadibas.Value);
    table3SIFREbasORTSTSAPMA.asfloat := SQRT(query3toplamsifrebas.Value);
    table3.Post;
    table3.Close;
end;
```

```

procedure TForm4.FormCreate(Sender: TObject);
begin
    table1.open;
    table2.open;
    table3.open;
    table4.open;
    table5.open;
    table9.open;
    table10.open;
    table12.open;
    TBL_nametime.open;
    tbl_snametime.open;
    tbl_passtime.open;
    tbl_aralikname.open;
    tbl_araliksname.open;
    tbl_aralikpass.open;
    TBL_namecalc.open;
    tbl_snamecalc.open;
    tbl_passcalc.open;
    Count:=0;
    Progressbar1.max := Table2TrainingNo.AsInteger;
    formclear;
    Clear;
end;

procedure TForm4.momenthesaplama;
var
    AdiM1, AdiM2, adiM3, adiM4, adiSkew, adiKurtosis: Extended;
    adiStsapma : Extended;
    AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew, AdibasKurtosis: Extended;
    adibasStsapma : Extended;
    soyadiM1, soyadiM2, soyadiM3, soyadiM4, soyadiSkew, soyadiKurtosis: Extended;
    soyadiStsapma : Extended;
    SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4, SoyAdibasSkew,
    SoyAdibasKurtosis: Extended;
    soyadibasStsapma : Extended;
    sifreM1, sifreM2, sifreM3, sifreM4, sifreSkew, sifreKurtosis: Extended;
    sifreStsapma : Extended;
    SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew, SifrebasKurtosis: Extended;
    sifrebasStsapma : Extended;
begin
    MomentSkewKurtosis(aralik1,AdiM1,AdiM2,AdiM3,AdiM4,AdiSkew,AdiKurtosis);
    AdiStSapma := Sqrt(AdiM2);
    MomentSkewKurtosis(basma1, AdibasM1, AdibasM2, AdibasM3, AdibasM4, AdibasSkew,
    AdibasKurtosis);
    AdibasStsapma := Sqrt(AdibasM2);
    MomentSkewKurtosis(aralik2,SoyAdiM1,SoyAdiM2,SoyAdiM3,SoyAdiM4,SoyAdiSkew,
    SoyAdiKurtosis);
    SoyAdiStSapma := Sqrt(SoyAdiM2);
    MomentSkewKurtosis(basma2, SoyAdibasM1, SoyAdibasM2, SoyAdibasM3, SoyAdibasM4,
    SoyAdibasSkew, SoyAdibasKurtosis);
    SoyAdibasStSapma := Sqrt(SoyAdibasM2);
    MomentSkewKurtosis(aralik3, SifreM1, SifreM2, SifreM3, SifreM4, SifreSkew, SifreKurtosis);
    SifreStSapma := Sqrt(SifreM2);
    MomentSkewKurtosis(basma3, SifrebasM1, SifrebasM2, SifrebasM3, SifrebasM4, SifrebasSkew,
    SifrebasKurtosis);
    SifrebasStSapma := Sqrt(SifrebasM2);
    try
        Table9.Open;
        Table9.Insert;
    
```

```

Table9ADIORT.AsFloat := AdiM1;
Table9ADISAP.AsFloat := AdiStSapma;
Table9ADISKEW.AsFloat := AdiSkew;
Table9ADICURR.AsFloat := AdiKurtosis;
Table9SOYORT.AsFloat := SoyAdiM1;
Table9SOYSAP.AsFloat := SoyAdiStSapma;
Table9SOYSKEW.AsFloat := SoyAdiSkew;
Table9SOYcurr.AsFloat := SoyAdiKurtosis;
Table9SIFORT.AsFloat := SifreM1;
Table9SIFSAP.AsFloat := SifreStSapma;
Table9SIFSkew.AsFloat := SifreSkew;
Table9SIFcurr.AsFloat := SifreKurtosis;
Table9.Post;
Table4.Open;
Table4.Insert;
Table4ADIBasORT.AsFloat := AdibasM1;
Table4ADIBasSAP.AsFloat := AdibasStSapma;
Table4ADIBasKEW.AsFloat := AdibasSkew;
Table4ADIBasCURR.AsFloat := AdibasKurtosis;
Table4SOYbasORT.AsFloat := SoyAdibasM1;
Table4SOYbasSAP.AsFloat := SoyAdibasStSapma;
Table4SOYbasSkew.AsFloat := SoyAdibasSkew;
Table4SOYbascurr.AsFloat := SoyAdibasKurtosis;
Table4SIFbasORT.AsFloat := SifrebasM1;
Table4SIFbasSAP.AsFloat := SifrebasStSapma;
Table4SIFbasSkew.AsFloat := SifrebasSkew;
Table4SIFbascurr.AsFloat := SifrebasKurtosis;
Table4.Post;
finally
Table9.Close;
Table4.Close;
end;
end;

procedure TForm4.ortalamahesaplama;
begin
try
Query2.Close;
Query2.Prepare;
Query2.ParamByName('NAME').AsString := Table1NAME.AsString;
Query2.ParamByName('SNAME').AsString := Table1SNAME.AsString;
Query2.ParamByName('PASS').AsString := Table1PASS.AsString;
Query2.Open;
Table10.Open;
Table10.Insert;
Table10ADIORTORT.AsFloat := Query2.AVERAGEOFadiort.AsFloat;
Table10ADISAPORT.AsFloat := Query2.AVERAGEOFadisap.AsFloat;
Table10ADISKEWORT.AsFloat := Query2.AVERAGEOFadiskew.AsFloat;
Table10ADICURRORT.AsFloat := Query2.AVERAGEOFadicurr.AsFloat;
Table10SOYORTORT.AsFloat := Query2.AVERAGEOFsoyort.AsFloat;
Table10SOYSAPORT.AsFloat := Query2.AVERAGEOFsoysap.AsFloat;
Table10SOYSKEWORT.AsFloat := Query2.AVERAGEOFsoyskew.AsFloat;
Table10SOYcurrORT.AsFloat := Query2.AVERAGEOFsoycurr.AsFloat;
Table10SIFORTORT.AsFloat := Query2.AVERAGEOFsifort.AsFloat;
Table10SIFSAPORT.AsFloat := Query2.AVERAGEOFsifsap.AsFloat;
Table10SIFSkewORT.AsFloat := Query2.AVERAGEOFsifskew.AsFloat;
Table10SIFcurrORT.AsFloat := Query2.AVERAGEOFsifcurr.AsFloat;
Table10.Post;
query1.Close;
query1.Prepare;
query1.ParamByName('NAME').AsString := Table1NAME.AsString;
query1.ParamByName('SNAME').AsString := Table1SNAME.AsString;

```

```

query1.ParamByName('PASS').AsString := Table1PASS.AsString;
query1.Open;
table5.Open;
table5.Insert;
table5ADIBASORTORT.AsFloat := query1AVERAGEOFadibasort.AsFloat;
table5ADIBASSAPORT.AsFloat := query1AVERAGEOFadibassap.AsFloat;
table5ADIBASSKEWORT.AsFloat := query1AVERAGEOFadibasskew.AsFloat;
table5ADIBASCURRTORT.AsFloat := query1AVERAGEOFadibascrr.AsFloat;
table5SOYBASORTORT.AsFloat := query1AVERAGEOFsoybasort.AsFloat;
table5SOYBASSAPORT.AsFloat := query1AVERAGEOFsoybassap.AsFloat;
table5SOYBASSkewORT.AsFloat := query1AVERAGEOFsoybasskew.AsFloat;
table5SOYBAScurrORT.AsFloat := query1AVERAGEOFsoybascrr.AsFloat;
table5SIFBASORTORT.AsFloat := query1AVERAGEOFsifbasort.AsFloat;
table5SIFBASSAPORT.AsFloat := query1AVERAGEOFsifbassap.AsFloat;
table5SIFBASSkewORT.AsFloat := query1AVERAGEOFsifbasskew.AsFloat;
table5SIFBAScurrORT.AsFloat := query1AVERAGEOFsifbascrr.AsFloat;
table5.Post;
finally
    Table10.Close;
    Table5.Close;
end;
end;

procedure TForm4.Button1Click(Sender: TObject);
var
    Say, ind, i : Integer;
begin
    if Count = 0
    then //ilk giris ise
    begin
        if not CheckUser
        then //kullanıcı tanımlıysa
        begin
            ShowMessage('Bu kullanıcı hesabı zaten mevcut !!!');
            Timetblrowdelete(sender);
            Clear;
            FormClear;
            For Say := 0 to 14 do
            begin
                basma1[say]:= 0; aralik1[say]:= 0;
                basma2[say]:= 0; aralik2[say]:= 0;
                basma3[say]:= 0; aralik3[say]:= 0;
            end; //for
            Edit1.setFocus;
            Exit;
        end;
    end;
    if Count < Table2TrainingNo.AsInteger -1
    then //1.then
    begin //1.begin
        with Table1 do
        begin //2.begin
            if Count = 0
            then //2.then
            begin //3.begin
                if (length(Edit1.Text)<2)or(length(Edit2.Text)<2)or(length(Edit3.Text)< 6)
                then //3.then
                begin //4.begin
                    Showmessage('Adı ve Soyadı için en az 2,Şifre için ise en az 6 karakterlik veri
girmelisiniz.');
```

```

        label1.Caption := IntToStr(count);
        FormClear;
        Clear;
        Edit1.SetFocus;
    except
        ShowMessage('Veri Tabanına kayıt esnasında hata olustu!!! Lütfen bilgilerinizi
ar girmeyi deneyiniz.');
```

```

else
begin
    momenthesaplama;
    Inc(Count,1);
    ProgressBar1.position:=ProgressBar1.position + 1;
    label6.caption := inttostr(Count);
    Clear;
    FormClear;
    For Say := 0 to 14 do
    begin
        basma1[say]:= 0; aralik1[say]:= 0;
        basma2[say]:= 0; aralik2[say]:= 0;
        basma3[say]:= 0; aralik3[say]:= 0;
    end;//for
    Edit1.SetFocus;
end;
end ; //2.else
end ;
end//1.then
else //1.then
begin
    momenthesaplama;
    ortalamahesaplama;
    stsapmahesaplama;
    // ShowMessage(FloatToStr(adiortstsap));
    // BitBtn2Click(nil);
    //nametime,snametime,passtime tablolarına isim,soyad,şifreleri yaz
    //nametime
    ind:=table2.FieldName('trainingno').asinteger-1;
    tbl_nametime.Editkey;
    tbl_nametime.FieldName('tnn').asinteger:=tblname_son;
    tbl_nametime.GotoKey;
    for i:=tblname_son downto tblname_son-ind do
    begin
        tbl_nametime.edit;
        tbl_nametime.fieldbyname('name').asstring:=edit1.text;
        tbl_nametime.fieldbyname('sname').asstring:=edit2.text;
        tbl_nametime.fieldbyname('pass').asstring:=edit3.text;
        tbl_nametime.post;
        tbl_nametime.prior;
    end;
    //snametime
    tbl_snametime.Editkey;
    tbl_snametime.FieldName('tsn').asinteger:=tblsname_son;
    tbl_snametime.GotoKey;
    for i:=tblsname_son downto tblsname_son-ind do
    begin
        tbl_snametime.edit;
        tbl_snametime.fieldbyname('name').asstring:=edit1.text;
        tbl_snametime.fieldbyname('sname').asstring:=edit2.text;
        tbl_snametime.fieldbyname('pass').asstring:=edit3.text;
        tbl_snametime.post;
        tbl_snametime.prior;
    end;
    //passtime
    tbl_passtime.Editkey;
    tbl_passtime.FieldName('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.GotoKey;
    for i:=tblpasstime_son downto tblpasstime_son-ind do
    begin
        tbl_passtime.edit;
        tbl_passtime.fieldbyname('name').asstring:=edit1.text;

```

```

tbl_passtime.fieldbyname('sname').asString:=edit2.text;
tbl_passtime.fieldbyname('pass').asString:=edit3.text;
tbl_passtime.post;
tbl_passtime.prior;
end;
Aralikmoments;
ProgressBar1.position:=ProgressBar1.position + 1;
label6.caption := inttostr(ProgressBar1.position);
ShowMessage('Sisteme başarılı olarak kayıt edildiniz!!!');
ModalResult := MrOk;
end;
end;

procedure TForm4.CheckType(Key: Word; Sender: TObject);
var
  Say : Integer;
begin
  if(key = VK_DELETE)or(key = VK_BACK)
  then
    begin
      Showmessage('Delete veya BackSpace Tuşuna Bastınız.Lütfen Tekrar Giriş Yapınız!!!');
      TEdit(Sender).Text := '';
      delbackmesaj := True;
      if TEdit(Sender).Name = 'Edit1'
      then
        begin
          For Say := 1 to 15 do
            begin
              basma1[say]:= 0;
              aralik1[say]:= 0;
            end;
          end;
        if TEdit(Sender).Name = 'Edit2'
        then
          begin
            For Say := 1 to 15 do
              begin
                basma2[say]:= 0;
                aralik2[say]:= 0;
              end;
            end;
          if TEdit(Sender).Name = 'Edit3'
          then
            begin
              For Say := 1 to 15 do
                begin
                  basma3[say]:= 0;
                  aralik3[say]:= 0;
                end;
              end;
            Clear;
          end
        else
          timer1.enabled := true;
          timer2.enabled := false;
        end;
      end;

procedure TForm4.Timer2Timer(Sender: TObject);
begin
  KAra:=KAra + timer2.interval;
end;

```

```

procedure TForm4.Timer1Timer(Sender: TObject);
begin
  Kdwn:=Kdwn + timer1.interval;
end;

```

```

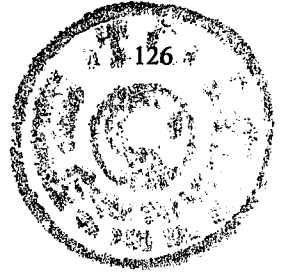
procedure TForm4.GetTimeBas(Sender: TObject);
begin
  if not delbackmesaj
  then
    begin
      i := i + 1;
      if TEdit(Sender).Name = 'Edit1'
      then
        begin
          basma1[i] := KDwn;
          ListBox1.Items.Add(FloatToStr(basma1[i]));
        end;
      if TEdit(Sender).Name = 'Edit2'
      then
        begin
          basma2[i] := KDwn;
          ListBox1.Items.Add(FloatToStr(basma2[i]));
        end;
      if TEdit(Sender).Name = 'Edit3'
      then
        begin
          basma3[i] := KDwn;
          ListBox1.Items.Add(FloatToStr(basma3[i]));
        end;
      // ListBox1.Items.Add(FloatToStr(KDwn));
      KDwn:=0;
      Timer2.Enabled:=True;
    end;
end;

```

```

procedure TForm4.GetTimeAra(Sender: TObject);
begin
  if TEdit(Sender).Name = 'Edit1'
  then
    if not ((tbl_nametime.State in [dsinsert]) or (tbl_namecalc.State in [dsinsert]))
    then
      begin
        Tbl_nametime.Insert;
        // Tbl_namecalc.Insert;
      end;
  if TEdit(Sender).Name = 'Edit2'
  then
    if not ((tbl_snametime.State in [dsinsert]) or (tbl_snamecalc.State in [dsinsert]))
    then
      begin
        Tbl_snametime.Insert;
        // Tbl_snamecalc.Insert;
      end;
  if TEdit(Sender).Name = 'Edit3'
  then
    if not ((tbl_passtime.State in [dsinsert]) or (tbl_passcalc.State in [dsinsert]))
    then
      begin
        Tbl_passtime.Insert;
        // Tbl_passcalc.Insert;
      end;

```



```
j:= j + 1;
inc(tbl_ind);
if TEdit(Sender).Name = 'Edit1'
then
begin
aralik1[j] := KAra;
ListBox2.Items.Add(FloatToStr(aralik1[j]));
tbl_nametime.fields[tbl_ind].AsInteger:=KAra;
// tbl_namecalc.fields[tbl_ind].AsInteger:=KAra;
end;
if TEdit(Sender).Name = 'Edit2'
then
begin
aralik2[j] := KAra;
ListBox2.Items.Add(FloatToStr(aralik2[j]));
tbl_snametime.fields[tbl_ind].AsInteger:=KAra;
// tbl_snamecalc.fields[tbl_ind].AsInteger:=KAra;
end;
if TEdit(Sender).Name = 'Edit3'
then
begin
aralik3[j] := KAra;
ListBox2.Items.Add(FloatToStr(aralik3[j]));
tbl_passtime.fields[tbl_ind].AsInteger:=KAra;
// tbl_passcalc.fields[tbl_ind].AsInteger:=KAra;
end;
// ListBox2.Items.Add(FloatToStr(KAra));
KAra:=0;
Timer2.Enabled:=False;
end;

procedure TForm4.Edit1KeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN))then
begin
timer1.Enabled:=true;
timer2.Enabled:=false;
GetTimeAra(Sender);
delbackmesaj := False;
end;
CheckType(Key,Sender);
end;

procedure TForm4.Edit1KeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if not((Key = VK_DELETE)or(Key = VK_BACK)or(Key = VK_RETURN)or(Key =
VK_TAB)or(Key = VK_SHIFT))then
begin
Timer1.Enabled := False;
GetTimeBas(Sender);
end;
end;

procedure TForm4.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if(Key = #13)
then
begin
if TEdit(Sender).Name = 'Edit1'
```

```

then
begin
    ActiveControl := Edit2;
    tbl_nametime.post;
    tblname_son:=tbl_nametime.fieldbyname('tnn').asinteger;
end;
if TEdit(Sender).Name = 'Edit2'
then
begin
    ActiveControl := Edit3;
    tbl_snametime.post;
    tblsname_son:=tbl_snametime.fieldbyname('tsn').asinteger;
end;
if TEdit(Sender).Name = 'Edit3'
then
begin
    ActiveControl := Button1;
    tbl_passtime.post;
    tblpasstime_son:=tbl_passtime.fieldbyname('tpn').asinteger;
end;
Clear;
end;
end;

procedure TForm4.Table9CalcFields(DataSet: TDataSet);
begin
{
    Table9.FieldByName('farkkare').Asfloat := sqrt((Table10.FieldByName('adiortort').Asfloat -
    Table9.FieldByName('adiort').Asfloat));
    aditoplam:= aditoplam + Table9.FieldByName('farkkare').Asfloat;
    aditoplamort := aditoplam/8;
    adistsapma := sqrt(aditoplamort);
}
end;

procedure TForm4.Button3Click(Sender: TObject);
begin
    if (count < Table2trainingno.asinteger-1) and (count > 0)
    then
    begin
        with query4 do
        begin
            close;
            ParamByName('NAME').AsString := Table1NAME.AsString;
            ParamByName('SNAME').AsString :=Table1SNAME.AsString;
            ParamByName('PASS').AsString := Table1PASS.AsString;
            execsql;
        end;
        with query5 do
        begin
            close;
            ParamByName('NAME').AsString := Table1NAME.AsString;
            ParamByName('SNAME').AsString :=Table1SNAME.AsString;
            ParamByName('PASS').AsString := Table1PASS.AsString;
            execsql;
        end;
        table1.delete;
        TimeTotalTblDelete(sender);
        close;
    end;
end;
close;

```

end;

```
procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);
begin
```

```
    table1.close;
    table2.close;
    table3.close;
    table4.close;
    table5.close;
    table9.close;
    table10.close;
    table12.close;
    TBL_nametime.close;
    tbl_snametime.close;
    tbl_passtime.close;
    tbl_aralikname.close;
    tbl_araliksname.close;
    tbl_aralikpass.close;
    TBL_namecalc.close;
    tbl_snamecalc.close;
    tbl_passcalc.close;
```

end;

```
procedure TForm4.TimeTblRowDelete(sender:Tobject);
begin
```

```
    tbl_nametime.editkey;
    tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
    tbl_nametime.gotokey;
    tbl_snametime.editkey;
    tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
    tbl_snametime.gotokey;
    tbl_passtime.editkey;
    tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.gotokey;
    tbl_nametime.Delete;
    tbl_snametime.Delete;
    tbl_passtime.Delete;
```

end;

```
procedure TForm4.TimeTotalTblDelete(sender:Tobject);
```

```
var
```

```
    i:integer;
```

```
begin
```

```
    i:=1;
    tbl_nametime.editkey;
    tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
    tbl_nametime.gotokey;
    tbl_snametime.editkey;
    tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
    tbl_snametime.gotokey;
    tbl_passtime.editkey;
    tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.gotokey;
```

```
// showmessage('count: '+inttostr(count));
```

```
while i<=count do
```

```
begin
```

```
    tbl_nametime.Delete;
    tbl_snametime.Delete;
    tbl_passtime.Delete;
    inc(i);
```

```

end;
//
tbl_name.IndexName:='ISMEGORE';tbl_sname.IndexName:='ISMEGORE';tbl_passtime.IndexName:
='ISMEGORE';
end;

```

```

procedure TForm4.Aralikmoments;
var

```

```

    AdiM1, AdiM2, adiM3, adiM4, adiSkew, adiKurtosis: Extended;
    adiStsapma : Extended;
    soyadiM1, soyadiM2, soyadiM3, soyadiM4, soyadiSkew, soyadiKurtosis: Extended;
    soyadiStsapma : Extended;
    sifreM1, sifreM2, sifreM3, sifreM4, sifreSkew, sifreKurtosis: Extended;
    sifreStsapma : Extended;
    dizi1:array of double;
    str_uz,indis,ind_a,ind_b,ind_c:integer;
    ad_str,soyad_str,pass_str:string;
begin

```

```

    indis:=4;
    //
    //tbl_name
    //
    setlength(dizi1,Table2TrainingNo.AsInteger);
    tbl_nametime.editkey;
    tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
    tbl_nametime.gotokey;
    //
    ad_str:=tbl_nametime.fieldbyname('name').asstring;
    soyad_str:=tbl_nametime.fieldbyname('sname').asstring;
    pass_str:=tbl_nametime.fieldbyname('pass').asstring;
    //
    str_uz:=length(tbl_nametime.FieldName('name').asstring);
    for ind_a:=1 to str_uz-1 do
    begin
        tbl_nametime.editkey;
        tbl_nametime.fieldbyname('tnn').asinteger:=tblname_son;
        tbl_nametime.gotokey;
        for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
        begin
            dizi1[ind_b]:=tbl_nametime.Fields[indis+ind_a].AsFloat;
            tbl_nametime.Prior;
        end;
        //moments olustur
    //    MomentSkewKurtosis(dizi1,AdiM1,AdiM2,AdiM3,AdiM4,AdiSkew,AdiKurtosis);
    //    AdiStSapma := Sqrt(AdiM2);
    adim1:=mean(dizi1);
    Adistsapma:=stddev(dizi1);
    try
        Tbl_aralikname.open;
        Tbl_aralikname.insert;
        Tbl_aralikname.FieldName('name').asstring:=ad_str;
        Tbl_aralikname.FieldName('sname').asstring:=soyad_str;
        Tbl_aralikname.FieldName('pass').asstring:=pass_str;
        Tbl_aralikname.FieldName('ORT').asfloat:=AdiM1;
        Tbl_aralikname.FieldName('STSAPMA').asfloat:=AdiStSapma;
    //    dm.T_aralikname.FieldName('SKEW').asfloat:=AdiSkew;
    //    dm.T_aralikname.FieldName('KURTOSIS').asfloat:=AdiKurtosis;
        Tbl_aralikname.Post;
    finally
        Tbl_aralikname.close;
    end;
end;

```

```

end;
end;
dizi1:=nil;
//
//tbl_sname
//
setlength(dizi1,Table2TrainingNo.AsInteger);
tbl_snametime.editkey;
tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
tbl_snametime.gotokey;
//
ad_str:=tbl_snametime.fieldbyname('name').asstring;
soyad_str:=tbl_snametime.fieldbyname('sname').asstring;
pass_str:=tbl_snametime.fieldbyname('pass').asstring;
//
str_uz:=length(tbl_snametime.FieldName('sname').asstring);
for ind_a:=1 to str_uz-1 do
begin
tbl_snametime.editkey;
tbl_snametime.fieldbyname('tsn').asinteger:=tblsname_son;
tbl_snametime.gotokey;
for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
begin
dizi1[ind_b]:=tbl_snametime.Fields[indis+ind_a].AsFloat;
tbl_snametime.Prior;
end;
//moments olustur
//
MomentSkewKurtosis(dizi1,soyadiM1,soyadiM2,soyadiM3,soyadiM4,soyadiSkew,soyadiKurtosis);
// soyadiStSapma := Sqrt(soyadiM2);
soyadim1:=mean(dizi1);
soyAdistsapma:=stddev(dizi1);
try
Tbl_aralikname.open;
Tbl_aralikname.insert;
Tbl_aralikname.FieldName('name').asstring:=ad_str;
Tbl_aralikname.FieldName('sname').asstring:=soyad_str;
Tbl_aralikname.FieldName('pass').asstring:=pass_str;
Tbl_aralikname.FieldName('ORT').asfloat:=soyadiM1;
Tbl_aralikname.FieldName('STSAPMA').asfloat:=soyadiStSapma;
// dm.T_aralikname.FieldName('SKEW').asfloat:=soyadiSkew;
// dm.T_aralikname.FieldName('KURTOSIS').asfloat:=soyadiKurtosis;
Tbl_aralikname.Post;
finally
Tbl_aralikname.close;
end;
end;
dizi1:=nil;
//
//tbl_PASS
//
setlength(dizi1,Table2TrainingNo.AsInteger);
tbl_passtime.editkey;
tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
tbl_passtime.gotokey;
//
ad_str:=tbl_passtime.fieldbyname('name').asstring;
soyad_str:=tbl_passtime.fieldbyname('sname').asstring;
pass_str:=tbl_passtime.fieldbyname('pass').asstring;
//
str_uz:=length(tbl_passtime.FieldName('pass').asstring);
for ind_a:=1 to str_uz-1 do

```

```

begin
    tbl_passtime.editkey;
    tbl_passtime.fieldbyname('tpn').asinteger:=tblpasstime_son;
    tbl_passtime.gotokey;
    for ind_b:=0 to Table2TrainingNo.AsInteger-1 do
    begin
        dizi1[ind_b]:=tbl_passtime.Fields[indis+ind_a].AsFloat;
        tbl_passtime.Prior;
    end;
    //moments olustur
    // MomentSkewKurtosis(dizi1,sifreM1,sifreM2,sifreM3,sifreM4,sifreSkew,sifreKurtosis);
    // sifreStSapma := Sqrt(sifreM2);
    sifrem1:=mean(dizi1);
    sifrestsapma:=stddev(dizi1);
    try
        Tbl_aralikpass.open;
        Tbl_aralikpass.insert;
        Tbl_aralikpass.FieldName('name').asstring:=ad_str;
        Tbl_aralikpass.FieldName('sname').asstring:=soyad_str;
        Tbl_aralikpass.FieldName('pass').asstring:=pass_str;
        Tbl_aralikpass.FieldName('ORT').asfloat:=sifreM1;
        Tbl_aralikpass.FieldName('STSAPMA').asfloat:=sifreStSapma;
    // dm.T_aralikpass.FieldName('SKEW').asfloat:=sifreSkew;
    // dm.T_aralikpass.FieldName('KURTOSIS').asfloat:=sifreKurtosis;
        Tbl_aralikpass.Post;
    finally
        Tbl_aralikpass.close;
    end;
    end;
    dizi1:=nil;
end;
end.

```

UNIT5

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Db, DBTables, StdCtrls, ComCtrls, Buttons, ExtCtrls;

type

```

TForm5 = class(TForm)
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    TrackBar1: TTrackBar;
    Label1: TLabel;
    Table1: TTable;
    TrackBar2: TTrackBar;
    Label2: TLabel;
    Table1LockCounter: TFloatField;
    Table1TrainingNo: TFloatField;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;

```

```

Label9: TLabel;
TrackBar3: TTrackBar;
Table1SecurityLevel: TFloatField;
procedure FormCreate(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure TrackBar3Change(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
private
    { Private declarations }

public
    { Public declarations }
end;

var
    Form5: TForm5;

implementation

{$R *.DFM}

procedure TForm5.FormCreate(Sender: TObject);
begin
    Table1.Open;
    if not Table1TrainingNo.IsNull then TrackBar1.Position := Table1TrainingNo.AsInteger;
    if not Table1LockCounter.IsNull then TrackBar2.Position := Table1LockCounter.AsInteger;
    if not Table1SecurityLevel.IsNull then TrackBar3.Position := Table1SecurityLevel.AsVariant*10;
end;

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
    Table1.Edit;
    Table1TrainingNo.AsInteger := TrackBar1.Position;
    Table1LockCounter.AsInteger := TrackBar2.Position;
    Table1SecurityLevel.Asfloat := (TrackBar3.Position/10);
    Table1.Post;
end;

procedure TForm5.TrackBar3Change(Sender: TObject);
begin
    trackbar3.Hint:='Deger: '+inttostr(trackbar3.position);
    trackbar3.SelEnd:=trackbar3.position;
end;

procedure TForm5.TrackBar2Change(Sender: TObject);
begin
    trackbar2.Hint:='Deger: '+inttostr(trackbar2.position);
    trackbar2.SelEnd:=trackbar2.position;
end;

procedure TForm5.TrackBar1Change(Sender: TObject);
begin
    trackbar1.Hint:='Deger: '+inttostr(trackbar1.position);
    trackbar1.SelEnd:=trackbar1.position;
end;
end.

```

ÖZGEÇMİŞ

Levent YALÇIN,1968 yılında Malatya’da doğmuştur. İlk ve Orta Öğretimini Ankara’da tamamladıktan sonra 1986 yılında Deniz Lisesin’den, müteakiben 1990 yılında Deniz Harp Okulun’dan mezun olarak Teğmen rütbesiyle Donanma’ya katılmıştır. 1996 yılı sonuna kadar Donanmanın muhtelif gemilerinde görev yaptıktan sonra Genel Kurmay Başkanlığın’ca ODTÜ’de verilen Bilgisayar Eğitimini müteakip Gölcük Tersanesi Komutanlığı Bilgi Sistemleri Şube Müdürlüğüne atanmış olup halen bu görevine devam etmektedir. 1998 yılında görevine devam ederken Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans yapmak üzere kaydolun Levent YALÇIN evli ve bir kız çocuk babası olup iyi derecede İngilizce bilmektedir.