

**SANAL MİKRO DENETLEYİCİ LABORATUARI İÇİN
SİSTEM YÖNETİCİSİ ARA YÜZÜ TASARIMI**

HASİBE HANDE ÇETİN

**YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANA BİLİM DALI
ISPARTA 2006**

T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SANAL MİKRO DENETLEYİCİ LABORATUARI İÇİN
SİSTEM YÖNETİCİSİ ARA YÜZÜ TASARIMI

HASİBE HANDE ÇETİN

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
ISPARTA 2006

Fen Bilimleri Enstitüsü Müdürlüğüne;

Bu çalışma jürimiz tarafından ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI' nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan :

Üye :

Üye :

ONAY

Bu tez/...../2006 tarihinde Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

...../...../2006

Prof. Dr. Fatma GÖKTEPE
S.D.Ü. Fen Bilimleri Enstitüsü Müdürü

İÇİNDEKİLER

İÇİNDEKİLER	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGELER DİZİNİ	v
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
1. GİRİŞ	1
2. KAYNAK BİLGİSİ	3
3. MATERYAL VE METOT	4
3.1. Internet Information Server (IIS)	4
3.2. .NET Framework	5
3.2.1. .NET Framework'ün Kurulumu	6
3.3. ASP.NET	7
3.3.1. Sanal Dizin (Virtual Directory)	11
3.3.2. Arka Planda Kod İçeren Sayfalar (Code Behind)	12
3.4. C#	13
3.5. Structure Query Language (SQL)	13
3.5.1. SQL Server'in Özellikleri	15
3.6. Control Area Network (CAN)	16
4. ARAŞTIRMA VE BULGULAR	17
4.1. Mühendislik Eğitimi İçin İnternete Dayalı, İnteraktif, Sanal Mikrodenetleyici Laboratuvarı (MicroLab)	17
4.1.1. Yazılım	17
4.1.2. Sistemin Çalışması	20
4.1.3. Donanım	25
4.1.4. Eksiklikleri	26
4.1.5. Yapılan Değişiklikler	27
4.2. Mikro Denetleyici Laboratuvarı İçin Sistem Yöneticisi Ara Yüzü (Yönetici Paneli)	30
4.2.1. Yönetici Panelinin Yapısı	30
4.2.2. Yönetici Paneli Tasarımı	42
5. SONUÇ	45
6. KAYNAKLAR	46
EKLER	49
EK – 1	50
Yönetici Paneli Program Kodları	50
ÖZGEÇMİŞ	76

ÖZET

SANAL MİKRO DENETLEYİCİ LABORATUARI İÇİN SİSTEM YÖNETİCİSİ ARA YÜZÜ TASARIMI

Hasibe Hande ÇETİN

Teknoloji gelişmesi ile birlikte teknolojinin birçok sektöre verdiği destek her geçen gün artmaktadır. Bu sektörlerden biri olan eğitimin içine hızla giren teknoloji eğitimi olumlu yönde etkilemektedir. Teknolojinin uygulamalı eğitimlerle birleşmesi sonucunda ise sanal laboratuvarlar oluşmuştur.

Bu kapsamda sanal laboratuvarlara web üzerinden erişim ve yönetim güçlü bir yazılım desteği gerektirmektedir. Teknik bilgiye sahip gerçek laboratuvarlarda çalışan birçok kullanıcının bir sanal laboratuvar oluşturmaları için gerekli yazılım alt yapısı zayıf kalmaktadır.

Sanal laboratuvarların fonksiyonelliğini artırmak, yönetimine hakim olmak ve bunları uygularken herhangi bir yazılım bilgisine ihtiyaç duymamak yönetici arayüz programlarının gerekliliğini ortaya çıkarmıştır.

Sanal laboratuvar olarak geliştirilen Mühendislik Eğitimi İçin İnternete Dayalı, İnteraktif, Sanal Mikrodenetleyici Laboratuvarının (MicroLab) ortak bir merkezden yönetimini sağlamak için kullanılabilecek bir yönetim paneli bulunmamaktadır. Mikro Denetleyici Laboratuvarı için Sistem Yöneticisi Ara Yüzü önceden yapılmış MicroLab programının yönetici ara yüzü eksiliğini gidermek amacı ile yapılmıştır. Gerçekleştirilen projede, uzaktan eğitim için internet üzerinden erişilebilen gerçek zamanlı ve interaktif mikro denetleyici laboratuvarı için sistem yöneticisi ara yüzü ASP.NET, SQL veritabanı dili kullanılarak tasarlanmış ve gerçekleştirilmiştir. MicroLab'daki bazı eksikliklerin tamamlanmasında C# dili kullanılmıştır. Bu yönetim (admin) paneli, MicroLab da kullanıcıların sisteme giriş çıkışlarını kontrol etmek, sistemde ne kadar kaldığını, hangi deneyleri uyguladığını tespit etmek, kullanıcıların bilgilerini güncellemek ve yeni deneyler ekleyerek sistemin daha verimli ve sürekli olmasını sağlamak için geliştirilmiştir. Bu çalışmada Yönetici arayüzü kullanımı ve temel olarak nasıl yapıldığı anlatılmıştır.

Anahtar Kelimeler: MicroLab, Yönetici Arayüzü, Sanal Laboratuvarlar

ABSTRACT

DESIGN OF SYSTEM ADMINISTRATOR INTERFACE FOR VIRTUAL MICROCONTROLLER LABORATORY**Hasibe Hande ÇETİN**

Support of technology sciences increase many different sectors day by day with development of technology. Education systems are one of the parts of these sectors which improved positively by the technology. With this development virtual laboratories are established with incorporation of practical education and technology.

So communication and management with web systems should have high software support. But persons who have technical knowledge in laboratory can not work in virtual laboratory because of not having enough software information about this. Because of these problems got necessary to user interface programmed.

To increase functions of virtual laboratory, to dominate the management and not requirement to any software knowledge when they practiced causes need of interface programmes'.

Internet-Based, Interactive, Virtual Microcontroller Laboratory Design For Engineering Education (MicroLab) which is improved by virtual laboratory has admin panel which can be managed centrally. System Administrator Interface for Microcontroller Laboratory was made because of solving problems of Administrator Interface. In this study System Administrator Interface for Microcontroller Laboratory was programmed to distance learning by using ASP.NET and SQL database language. At MicroLab to complete defectiveness, C# programming language was used for accomplishment some insufficiency. This admin panel improved for checking connections and disconnections from system, how long they stay there, to determine which experiment they used , to update user's datas of MicroLab users and to provide more effective and more constant system by add new experiments. In this study basic Administrator Interface usage and how basically it realized explained.

Key Words: MicroLab, Administrator Interface, Virtual Laboratory

TEŞEKKÜR

Danışmanım Sayın Doç. Dr. Akif KUTLU' ya bu çalışmanın gerçekleştirilmesindeki bilgi, deneyim ve katkılarından dolayı teşekkür ederim. Çalışmamın tasarlama safhasında fikirleriyle bana yardımcı olan sayın Öğrt.Gör. Ecir Uğur KÜÇÜKSİLLE'ye, bu tezin yapılmasına yardımları döküman ve emeği geçen tüm arkadaşlarıma teşekkür ederim. Çalışmanın uygulama safhasındaki yardımlarından dolayı arkadaşım Melike KARAYOL'a, maddi ve manevi desteklerini ve koşulsuz güvenlerini hiçbir zaman esirgemeyen değerli aileme ve eşim Hakan ÇETİN'e teşekkürü bir borç bilirim.

Hasibe Hande ÇETİN

SİMGELER DİZİNİ

SQL	Structured Query Language
ASP.NET	Active Server Pages.NET
JIT	Just-in-time
ASMX	Web Services
ASPX	Web Forms
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language

ŞEKİLLER DİZİNİ

Şekil 3.1 ISS Yönetim Ekranı	4
Şekil 3.2 ASP.NET'in .NET çatısı ile entegrasyonu	8
Şekil 4.1 Yazılım Mimarisi	18
Şekil 4.2 Deney İçin Kullanıcı Ara Yüzü (Kutlu,2004).....	21
Şekil 4.3 Giriş Sayfası.....	22
Şekil 4.4 Lobby Sayfası	23
Şekil 4.5 Kullanıcı Arayüzü	24
Şekil 4.6 8051 Tabanlı Mikrodenetleyici Laboratuvarı (Taşdelen, 2004).....	25
Şekil 4.7 Yönetici Paneli Giriş Paneli.....	30
Şekil 4.8 Kullanıcı İşlemleri Paneli	31
Şekil 4.9 Kullanıcı Ekleme Paneli	32
Şekil 4.10 Kullanıcı Düzenleme Paneli	32
Şekil 4.11 Kullanıcı Silme Paneli	33
Şekil 4.12 Operatör Atama Paneli.....	33
Şekil 4.13 Kayıt Sil Paneli	34
Şekil 4.14 Deneyler Paneli.....	34
Şekil 4.15 Deney Ekleme Paneli.....	35
Şekil 4.16 Deney Silme Paneli.....	36
Şekil 4.17 Deney Düzenleme	36
Şekil 4.18 Deney İzleme Paneli	37
Şekil 4.19 Deney Süresi/Guest Kontrolü Paneli	37
Şekil 4.20 Raporlar Paneli.....	38
Şekil 4.21 Aktif Kullanıcılar Paneli	39
Şekil 4.22 Kullanıcı Hareketleri Paneli.....	40
Şekil 4.23 Ayarlar Paneli	41
Şekil 4.24 Yönetici Adı Değiştirme Paneli	41
Şekil 4.25 Yönetici Şifresi Değiştirme Paneli	42
Şekil 4.26 Çıkış İşlemi	42
Şekil 4.27 Userlog tablosu	44
Şekil 4.28 KULLANICI Tablosu.....	44
Şekil 4.29 MicroLab veritabanı	44

ÇİZELGELER DİZİNİ

Çizelge 4.1 Client ve Server Komutları.....	18
Çizelge 4.2 Deney Modüllerinin tanımlayıcı haritası	20

1. GİRİŞ

İnternet, birçok alanda olduğu gibi eğitim sistemine de önemli ölçüde fayda sağlamaktadır. Eğitim ve teknoloji birleştiğinde, öğretim ve öğrenim fırsatları daha da genişler. Bununla birlikte, teknoloji alanındaki çok hızlı değişim oranı akademi enstitüleri için özellikle mühendislik disiplini için özel problemlere yol açar (Ertuğrul, 2005). Teknolojideki hızlı ilerlemelerden kaynaklanan yeni donanım ve materyal gereksinimi artmakta bu da kısıtlı imkânlarla sahip olan eğitim kurumları için problemlere yol açmaktadır.

1980'lerin başından beri grafiksel yaklaşıma sahip programlama yazılımları birçok başarılı mühendis, bilim adamı ve teknisyenin uygulamalarında çıkan sorunlara karşı çözüm yolları bulmakta yardımcı olmuştur. Endüstrileşme ve Laboratuvar araştırmalarının artmasıyla, laboratuvar aletlerinin takım fiyatları artmıştır. Uygun yazılımlar Data Acquisition Card (DAQ) ile yapılan işlemler veya bilgisayar parçalarındaki büyük fiyat indirimleriyle, geleneksel elektronik malzeme takımının değeri oldukça düşmüş ve yapay malzeme eşitliği ortaya çıkmıştır (Buckman, 1999). Böylece her eğitim ve mühendislik bilimleri ile uğraşan kişiler kendilerine birer sanal laboratuvar kurma ihtiyacı hissetmişlerdir. Bilgisayarlarında kullandıkları yazılımlar ile geliştirmeye harcadıkları zaman fark edilir derecede azalmış ve problemlere çözüm bulmak hızlanmıştır.

Teknik alanlarda eğitimin diğer bilimlerdeki eğitime göre farklılıkları bulunmaktadır. Belli bir alt yapı gerektiren, uygulamaya dayalı bir eğitim olması şarttır. Dolayısıyla eğitilecek kişinin bir ön eğitimden geçmesi gerekmektedir. Klasik teknik eğitimde birey okulda teorik derslerde yetiştirildikten sonra, atölye ve laboratuvar gibi uygulamalı dersler ve stajlar ile gerçek iş hayatına hazırlanmaktadır. Uygulamalı bir eğitim, ekonomik zorluklar getirmektedir. Eğitim kurumunda her uygulamayı yapabilmek büyük bir başarıdır. Teknik eğitime, uzaktan eğitim bu aşamaların uygulama olmayan kısımlarında eklenebilir. Teknik eğitimin ön hazırlığı uzaktan, ancak uygulama gerektiren bölümler klasik yöntemle atölye veya laboratuvarlarda gerçekleştirilmelidir. Bazı e-öğrenme modellerinde yapılması gereken deneyler

benzetilerek sanal ortamda uygulanmaktadır. Ancak teknik bir eleman olacak bireyin, gerçek araçlar ile deneyi yapıp sonucunu görmesi daha hazır hale gelmesini sağlayacaktır. Sanal Laboratuvarların kullanılmasıyla eski tip laboratuvarlarda yer alan bazı dezavantajlar avantajlara dönüşmektedir.

Web de ilk yapılan basit yazılımlarda sayfalar statik vaziyette idi. Yapılan bir sayfayı değiştirmek ya da yeni bir şeyler eklemek eziyet verici bir durumdu. İnternet'in kullanımı arttıkça sektörde bulunan işletmeler, fabrikalar, eğitim kurumları vs. satış ve yayın politikalarını değiştirerek 20. yy.'ın devrim niteliğinde olan internete taşınlar. Şirketlerin internet ortamına geçmesi ile eziyet haline gelmiş statik sayfa tasarımı yeni programların gelişmesi ile dinamik bir yapıya dönüştürüldü. Fabrikalarda, organizasyonu ve düzeni sağlayan yönetim kadrosu olduğu gibi yapılan web tabanlı programlarda da yazılımın tek bir merkezden yönetilmesi için bir yönetici (admin) paneline ihtiyaç duyuldu. Yazılımcıların programlara admin panelini eklemesi ile yazılımın sahibi olan kişi ya da kurumlar sayfalarını tek bir merkezden yönetme, sayfalarını değiştirme ve geliştirme olanaklarına kavuşmuşlardır. Böylelikle sayfaları değiştirme ve eklemede yazılımcıya duyulan ihtiyaç azalmıştır. Bunun tersini düşünecek olursak da yazılımcıların yükleri hafiflemiş ve sadece büyük problemler çıktığında onlara ihtiyaç duyulmuştur.

Sanal laboratuvar olarak geliştirilen Mühendislik Eğitimi İçin İnternete Dayalı, İnteraktif, Sanal Mikrodenetleyici Laboratuvarının (MicroLab) ortak bir merkezden yönetimini sağlamak için kullanılabilecek bir yönetim paneli bulunmamaktadır (Kutlu,2004). Gerçekleştirilen projede, uzaktan eğitim için internet üzerinden erişilebilen gerçek zamanlı ve interaktif mikro denetleyici laboratuvarı için sistem yöneticisi ara yüzü tasarlanmış ve gerçekleştirilmiştir. Bu yönetici (admin) paneli, MicroLab da kullanıcıların sisteme giriş çıkışlarını kontrol etmek, sistemde ne kadar kaldığını, hangi deneyleri uyguladığını tespit etmek ve yeni deneyler ekleyerek sistemin daha verimli ve sürekli olmasını sağlamak için geliştirilmiştir.

2. KAYNAK BİLGİSİ

Casini vd. (2004), yayınladıkları makalelerinde genel bir web server'a bağlı kontrol edilebilir deneylerin bulunduğu laboratuara uzaktan erişimini sağladıkları "Automatic Control Telelab – ACT" den bahsetmektedirler. Öğrenciler deneyler ile eş zamanlı çalışabilmektedir. Hazırlanan ara yüz ile öğrenciler kendilerine test edilebilen kontrol ara yüzü hazırlayabilmektedir.

Akın vd. (2003), hazırlamış olduğu bildirisinde, eğitim laboratuvarlarının yürütülmesinde geleneksel yöntemlerin değişik kısıtlamalarına göre bilgisayar temelli sanal laboratuvarların, geleneksel laboratuvarlara bir alternatif olarak büyük bir potansiyel kazandığından bahsetmiştir. Çalışmalarında değişik sanal laboratuvar gerçekleştirme modelleri ve bu laboratuvarın geleneksel laboratuvarlara göre üstün ve eksik yanları sunulmuştur. Matlab, LabVIEW, Mathematica ve VLEM gibi yazılımların sanal laboratuvar yazılımı olarak daha çok öne çıktığına değinilmiştir.

Mendi vd.(2004), yayınlamış olduğu makalesinde, Mühendislik ve Teknik Eğitim faaliyetinin yürütüldüğü kurumlarda öğretilen, Tasarı Geometri ders konularından, Görünürlük Prensiplerinin bilgisayar destekli öğretimini sağlamak için hazırlanan bir bilgisayar programından bahsetmiştir. AutoCAD ortamında, AutoLISP ve DCL programlama diliyle hazırlanan program, kullanıcıyla etkileşimli olarak çalışmaktadır. Program, geleneksel yöntemlerle yapılan öğretime göre derse ait uygulamaların canlandırılmasına yardımcı olarak öğrenmeyi daha kolay ve kalıcı yapmaktadır.

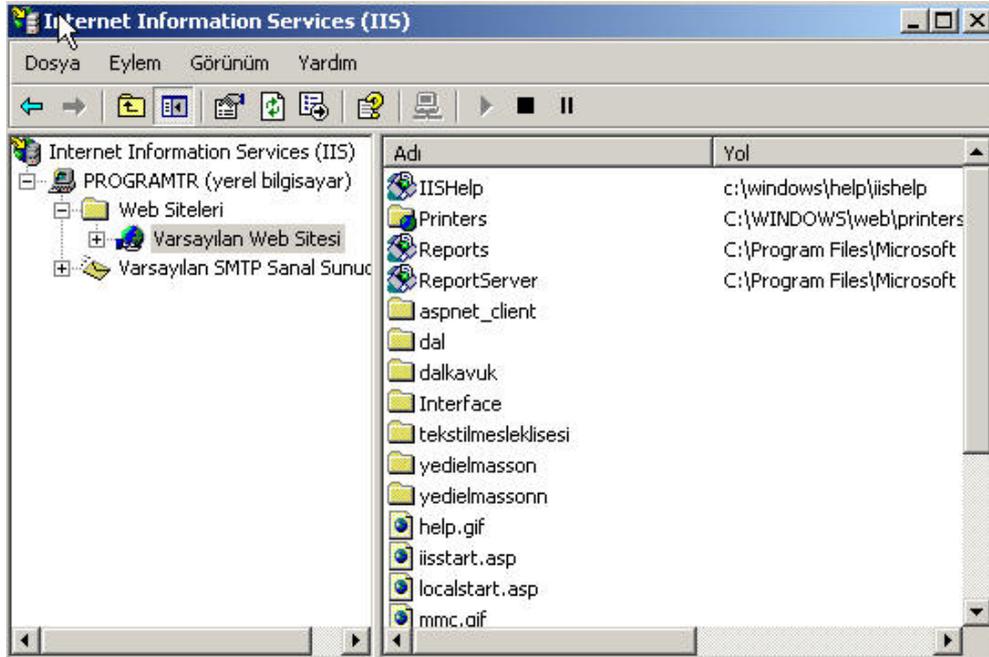
Ko vd. (2000, 2001a, 2001b), yayınlamış olduğu makalelerinde, üniversite öğrencilerinin eğitimi için frekans modülasyon deneyini, birleştirilmiş depo araçlarını, geniş ölçekli osilaskop deneylerini içeren yeni web tabanlı sanal laboratuvarından (Virtual Laboratory – VLAB) bahsedilmiştir. Laboratuvarın kullanıcıları doğrulamak ve veri trafiğini azaltan birleştirilmiş planlar ve sisteme giriş için kullanılan ortak web sunucusu anlatılmıştır. Kullanıcı ara yüzünü nasıl oluşturduklarına ve ne işe yaradığına değinilmiştir.

3. MATERYAL VE METOT

3.1. Internet Information Server (IIS)

.NET framework ile birlikte entegre çalışan Microsoft İnternet İnförmatıon Server (IIS), yönetıcı ve host web siteleri yaratmak için kullanılan bir web server yazılımıdır (Ryan D. ve Ryan T., 2002). ASP.NET uygulamalarını sunucu tarafında çalıştırdıktan sonra istemciye sunmada İnternet İnförmatıon Server 'a ihtiyaç duyulmaktadır. ASP'de kullanılan Personal Web Server (PWS) ASP.NET uygulamaları için kullanılamamaktadır (Çankaya, 2002).

IIS kurulumu gerçekleştirildikten sonra Control Panel ekranından Performance and Maintenance seçeneğini seçerek gelen ekrandan Administrative Tools sonrası IIS'i görmemiz mümkündür. IIS'e çift tıklayıp açtığımızda karşınıza Şekil 3.1.'deki bir ekran görüntüsü gelecektir (Çankaya, 2002).



Şekil 3.1 ISS Yönetim Ekranı

IIS'in yönetim ekranında üst kısımda yer alan Start/Stop ve Pause tuşlarını kullanarak servisi başlatma, durdurma veya duraklatma mümkündür (Çankaya, 2002).

3.2. .NET Framework

ASP.NET uygulamalarını veya sayfalarını geliştirebilmek için NotePad gibi basit bir text editöründen başka bir şeye ihtiyacınız yoktur. Ancak ASP.NET sayfalarını çalıştırabilmek veya test edebilmek için .NET framework'ü olan IIS'e (Internet Information Server) ihtiyaç vardır.

Bildiği gibi ASP ve ASP.NET, sunucuda çalışan uygulamalardır. Bir ASP.NET sayfasını (uygulamasını) geliştirdikten sonra, çalışıp çalışmadığını, çalışmıyorsa nerelerde hata yapıldığını mutlaka görmek istenilecektir.

Bunun için iki yöntem vardır; birincisi bilgisayarınızı bir ASP.NET sunucusuna çevirmek, ikincisi ise web'de ASP.NET sayfalarını host eden bir firmanın sunucusuna yükleyip buradan test etmektir. İkinci yöntem hem zaman hem de maliyet açısından çok verimli değildir. Eğer birinci yöntemi seçerseniz, yani bilgisayarınızı ASP.NET sunucusuna çevirip ASP.NET sayfalarınızı kendi bilgisayarınızda test etmek isterseniz, .NET Framework'ün yüklenmesi gerekmektedir.

.NET Framework, Microsoft işletim sistemini .NET uyumlu yapar, birtakım servis, program ve hizmetlerle birlikte ASP.NET sayfalarını yayımlamayı ve test etmeyi sağlar. Tabii ki bu Framework sadece test amaçlı değil, internet üzerinde profesyonel olarak sayfa barındıran (web hosting) firma, sunucularında ASP.NET hizmeti vermek isteniyorsa yine .NET Framework yüklenmelidir. Framework'ün "Redistributable" ve "SDK" isminde iki farklı dağıtım mevcuttur.

ASP.NET sayfalarınızı çalıştırmak için bunlardan sadece bir tanesini seçip yüklemeniz yeterlidir. Redistributable, ASP.NET uygulamalarınızı çalıştırmak için gerekli bütün bileşenlere sahip ve düşük dosya boyutundadır (21MB). "SDK" ise "Redistributable" ilave olarak, yardımcı her türlü dokümantasyon, örnek uygulamalar ve birtakım araçlar içermektedir. Ancak dosya boyutu oldukça fazladır

(106MB). Normalde SDK'nın yüklenmesi uygundur. Çünkü gerçekten güzel düzenlenmiş yardım dosyaları ve uygulamaları ile en azından referans kaynak olarak kullanılabilir.

Windows 95, Windows 98 veya Windows ME işletim sistemine sahip olan bilgisayarlara Framework yüklenememektedir. Bu işletim sistemlerinden birine sahip olan bilgisayarlar maalesef ASP.NET sayfalarını çalıştıramayacaklardır. Sayfalarını çalıştırmak istiyorlarsa yukarıda bahsedilen ikinci yöntemi kullanmak, yani dışarıdaki bir ASP.NET sunucusundan faydalanmak durumundadırlar.

ASP.NET ile uygulama geliştirmek istiyorsa, minimum Windows 2000 gibi bir işletim sistemi kullanılmalıdır. Uygulama geliştiren bir programcı bütün kontrolün kendinde olmasını ve geliştirme aşamasında uygulamalarını anında test etmek ister. Bunun için uzaktaki bir sunucudan yararlanmak çok geçerli bir çözüm değildir. İşin olması gerekeni, bütün geliştirme ortamını kendi bilgisayarında test etmelidir. Bu ve bunun gibi pek çok nedenden dolayı, eğer ASP.NET ile uygulamalar geliştirilecekse Windows 2000 veya XP'ye terfi edilmelidir.

3.2.1. .NET Framework'ün Kurulumu

.NET Framework ün bilgisayara kurulumunda. Kurulum işlemini başlatmak için Setup.exe programını çalıştırmak gerekir. Program başlangıçta kurulum için gereken dosyaları açacaktır. Daha sonra bir pencere çıkıp size programı yükleyeceğini bildirecektir, burada Next düğmesine basıldıktan sonra bir lisans anlaşması ekranı ile karşılaşılacaktır.

Burada "I accept the agreement" düğmesi işaretlenip tekrar Next düğmesine basılacaktır. Bundan sonraki çıkan ilk pencerede Framework' ü hangi opsiyonlarla yüklemek istediğiniz sorulacaktır.

Gerekli olan seçenekler işaretlendikten sonra tekrar Next düğmesine basılarak. Yazılımın hangi klasöre yükleneceği sorulacaktır. Framework'ün yüklenmek istendiği klasör, Browse düğmesiyle seçilir. Programın yükleneceği klasörü seçtikten sonra bir kez daha Next düğmesine basılarak, program kurulum işlemi başlatılır.

Bu süreç makinenin konfigürasyonuna göre zaman alıcı olabilir. Kurulum işlemi bittikten sonra "Installation of Microsoft .NET Framework SDK (English) i complete" diye bir mesaj ile karşılaşılabılır. Bu mesajdan anlaşıldığı gibi .NET Framework SDK programının kurulumu tamamlanmıştır (Demirkol, 2005).

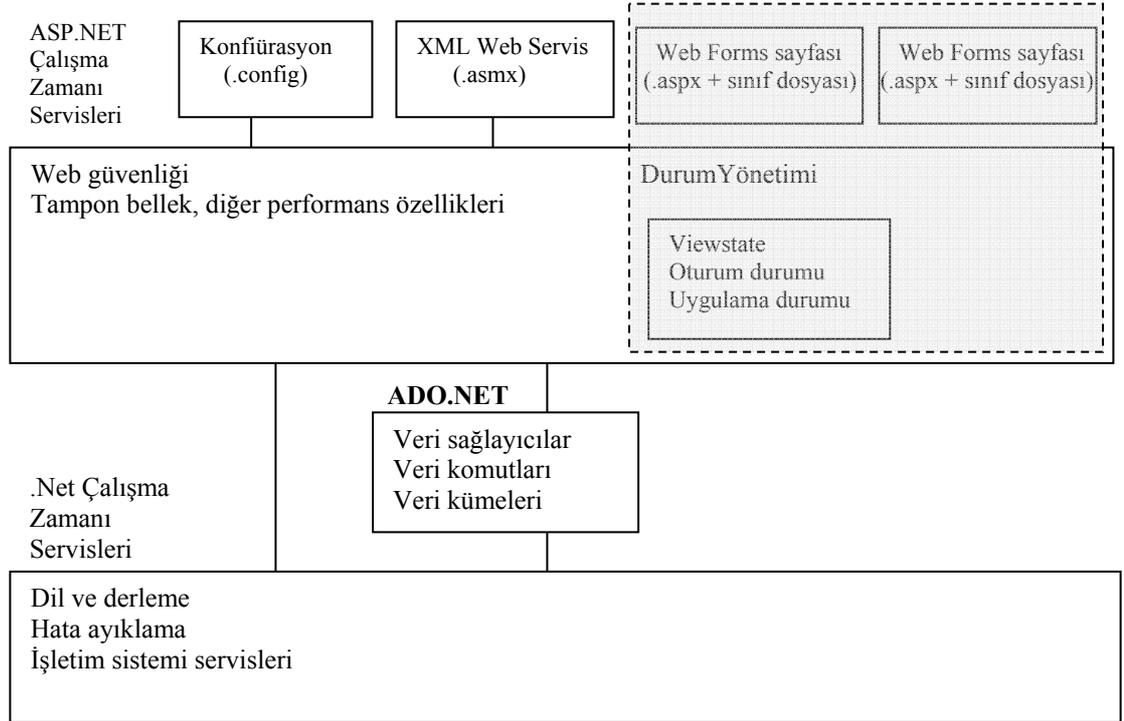
3.3. ASP.NET

ASP.NET yani Active Server Pages.NET, Microsoft'un .NET ile duyurmuş olduğu internet uygulamaları ve web servisleri için sunucu taraflı yazılım geliştirmeyi kolay ve genişleyebilir yapıda sağlayan teknolojidir (Çankaya, 2002).

ASP.NET, .NET çatısında belirli dosya tiplerine yönelik talepleri yöneten katmandır. Sözü edilen dosya tipleri arasında .aspx, .ascx ve .asmx uzantılı olanlarla birlikte başka dosyalar da yer almaktadır. ASP.NET motoru, dinamik içerik oluşturmak için güçlü bir nesne modeli sağlar ve .NET çatısına gevşek bir şekilde bağlıdır. Bu entegrasyon, .NET çatısının Windows dışındaki platformlara göç ettiği durumlarda uygulamanın değiştirilmesini kolaylaştırmaktadır.

ASP.NET, .NET çatısının bir parçasıdır ve .NET'in yüklendiği her sunucuda mevcuttur. Bir başka ifadeyle ASP, artık bir eklenti olmaktan çıkarılmış, meşrulaştırılmıştır. ASP.NET, önceden belirlenmiş özel görevleri yerine getiren sınıf ve nesnelere ortaya çıkararak bir şekilde uygulanmaktadır. "Klasik" ASP ile (yani .NET Öncesi ASP sürümleriyle) ASP.NET'in programlamasının biraz farklı yapıldığı görülecektir. Ama bir Web uygulaması inşa etmeyle ilgili temel kavramlar büyük ölçüde aynıdır.

ASP.NET uygulamaları, istemci taleplerine dinamik olarak yanıt veren bir ya da daha fazla sayıda Web sunucusu üzerinde barındırılan merkezi uygulamalardır. Yanıtlar dinamiktir, çünkü ASP.NET spesifik uzantıları olan (örneğin .aspx ve .ascx gibi) dosyalarla ilgili talepleri yakalayıp, bunları anında yanıt oluşturabilen tam zamanında (just-in-time - JIT) derlenmiş kod dosyalarına iletir. Şekil 3.2 ASP.NET'in .NET çatısının geri kalan kısmı ile nasıl bütünleştiğini göstermektedir.



Şekil 3.2 ASP.NET'in .NET çatısı ile entegrasyonu.

Şekil 3.2'ye bakarak, ASP.NET'in spesifik olarak konfigürasyon dosyaları (web.config ve machine.config), Web Services (ASMX) dosyaları ve Web Forms (ASPX) dosyaları ile ilgilendiğini çıkarabiliriz. Sunucu bu dosya tiplerinden hiç birini "sunmamaktadır". Onun yerine, uygun içerik türünü istemciye iletmektedir. Konfigürasyon dosya tiplerinde belli bir uygulama ya da uygulama parçası için başlangıç değerleri ve ayarları bulunur, machine.web adı verilen bir başka dosya da makine düzeyindeki başlangıç değer ve ayarlarını tutar. Sunucu, Web dosyaları ile ilgili talepleri göz ardı eder, çünkü bunları sunması bir güvenlik ihlali oluşturabilir.

Web Forms ve Web Services dosya tipleriyle ilgili istemci talepleri, sunucunun kodu yüklemesine, çözümlemesine, çalıştırmasına ve dinamik bir yanıt vermesine neden olur. Web Forms'da bu yanıt genellikle HTML ya da WML şeklindedir. Web Services'de ise sunucu tipik olarak bir Simple Object Access Protocol (SOAP) yanıtı oluşturur. SOAP yanıtları tanımları gereği "durum" içermediklerinden derhal çalıştırılabilirler. Ancak Web Forms, aksi belirtilmemişse belli bir durumdadır. Web Forms, durum bilgisini tutmak için kullanıcı arabirimi ve diğer kalıcı değerler arasında alışveriş yaparak her talep için sunucu ile istemci arasında bilgi değişimi sağlar. Şekil 2.2'de Sayfa Çatısı başlıklı ve kesik çizgiyle belirtilmiş dikdörtgen aradaki farkı göstermektedir. Bir Web Form'a yönelik bir talep, talepler arasında değer saklayabilmek için ViewState'i, oturum durumunu (session state) ya da uygulama durumunu (application state) kullanabilmektedir. ASP.NET'in durum bilgisi tutma özelliğini Web Service olarak kullanmak mümkündür (varsayılan durumda değil) ama performans açısından bu yola gitmekten kaçınmanız gereklidir.

Gerek Web Forms gerek Web Services talepleri, ASP.NET'in entegre güvenliğinden ve ADO.NET üzerinden veri erişimi sağlayabilmesinden yararlanarak, bir yanıt oluşturmak için sistem servislerini kullanan bir kodu çalıştırabilirler.

Statik ve dinamik talepler arasındaki ana fark, tipik bir Web talebinin statik bir dosyaya referansta bulunmasıdır. Sunucu dosyayı okur ve istenen dosyanın içeriğiyle yanıt verir. ASP.NET'te böyle bir sınırlama yoktur. Mevcut bir dosyayla yanıt verilmesi zorunlu değildir. Bir isteğe neyle isterseniz onla yanıt verilebilir. Yanıtlarınız arasında dinamik olarak oluşturulmuş HTML kodu, Extensible Markup Language (XML), grafik, ham metin ya da ikili veri de bulunabilir. Yani yanıt her şey olabilir. Bu yetenek kendi başına bir yenilik değildir. Tüm bu teknolojiler bir HTTP talebine dinamik olarak yanıt verebilmesine imkan sağlamaktadır. Bu kapsamda farklılıkları şunlardır :

- Klasik ASP'nin aksine ASP.NET, .NET dillerini kullanmaktadır. Dolayısıyla herhangi bir .NET assembly'si ya da sınıfının tüm gücüne tıpatıp C#'ta yazılmış

herhangi bir Windows programından olduğu gibi erişilebilir. ASP.NET bu anlamda daha önceki dönemlerin derlenmiş CGI programlarına benzemektedir. Ancak CGI'da her talep için programın ayrı bir kopyasının yüklenmesi ve çalıştırılması gerekir. ASP.NET kodu ise çok kanallı, JIT (tam zamanında) derlenmiş, talep üzerine yüklenen ASP.NET DLL assembly'lerinde bulunur. ASP.NET DLL'leri bir kez yüklenince bellekteki tek bir kopya üzerinden birden fazla talebe yanıt verebilirler.

- ASP.NET tüm .NET dillerini desteklemektedir (şu anda C#, C++, VB.NET ve JScript mevcuttur ama .NET üzerinde çalışmak üzere geliştirilmekte ya da entegre edilmekte olan 20'den fazla farklı dil söz konusudur). Dolayısıyla, zaman içinde arzu ettiğiniz tüm modern programlama dillerini kullanarak Web uygulamaları yazmak mümkün olacaktır. Buna ek olarak Intel ve Hewlett-Packard ile birlikte çalışan açık kaynak gruplarının .NET'i çeşitli Unix ve Linux versiyonları üzerinde destekleme çabaları da vardır. JavaServer Pages sadece Java'yı desteklemektedir. Ancak Java şu anda çok geniş bir tabandan destek bulduğu için bu pek bir kısıtlama sayılmamaktadır. Java Servlets, ASP.NET e daha çok benzemektedir. Ancak durum bakımı, Web Services ya da XML için fazla bir destek sağlamamaktadır. Ayrıca, hiçbir Java tasarım ortamı özellik ve kalite açılarından Visual Studio ile yarışacak durumda değildir.
- Klasik ASP çeşitli script dillerini desteklemekteydi. Ancak pratikte bunlar arasında VBScript ve JScript açık farkla başı çekmekteydi. Klasik ASP ile kullanabilecek script dilleri tipi olmayan, yorumlanan (yani derlenmeyen) ve pek fazla güçlü olmayan dillerdi. Yine de COM uyumlu herhangi bir dilde DLL'ler yazarak ASP'nin temel işlevselliğini genişletebilmekteydik. ASP.NET'in bir başka rakibi olan ColdFusion ise ColdFusion Markup Language (CFML) imini kullanmaktadır. Bu im güçlü ama sınırlı bir dizi yetenek sunmaktadır. Ancak CFML'in yeteneklerini C++ ya da Java'da kendi programlarınızı yazarak genişletme olanağı mevcuttur.
- Microsoft, geliştiricilerin klasik ASP üzerinde yaşadıkları milyonlarca saatlik

deneyimden yararlanmıştır. Bu nedenle ASP.NET, sadece hız ve güç bakımından dev ilerlemeler getirmekle kalmamaktadır. Aynı zamanda, sunucudan istemciye kesintisiz hata giderme, form verilerinin geçerlik denetiminin otomatik olarak yapılması ve Windows uygulamalarına çok benzeyen bir programlama modeli sunulması gibi geliştirme alanına ait önemli ilerlemeler getirmektedir (Jones, 2003).

3.3.1. Sanal Dizin (Virtual Directory)

ASP'den ASP.NET'e yeni geçildiğinde karşılaşılabilecek en büyük problem sanal dizin kavramıdır. İlk geçiş anında bir çok sorun ortaya çıkmaktadır.

ASP.NET'te tek tek sayfalarla çalışırken aslında bu dizinlere gerek yok ama uygulama bazında çalışacaksa o zaman uygulamanın sanal bir dizin içinde bulunması gereklidir. Özellikle web.config dosyası kullanacaksa bu dosyanın sanal dizinin kök (root) klasöründe olması gereklidir.

Bu zorunluluk sadece uygulamanın gerçekleştirildiği bilgisayarda ki çalışmalarda gerekli olacak. Zaten sunucuda uygulamanın kayıtlı duracağı alana atarken bu çalışmanın bulunduğu yer zaten sanal dizin olacaktır.

Sanal dizin aşağıda belirtildiği şekilde oluşturulur ;

- 1) Sanal dizin Internet **Informations Services (IIS) Yönetim Paneli**'nden oluşturulur. Bunun için IIS Yönetim Paneli, Denetim Masası'ndan, Yönetimsel Araçlar içindeki link'inden açılır.
- 2) Açılan yönetim panelinde, sol tarafta ağaç yapısında bulunan öğelerden **Varsayılan Web Sitesi**'ne gelene kadar artılara (+) tıklayarak alt öğeye geçilir. Son Varsayılan Web Sitesi'ne sağ tıklayarak **Yeni * Sanal Dizin...** öğesi seçilir.
- 3) Açılan pencerede dizin ismi ve yeri belirtilerek uygulamanın çalıştırılabilmesi için gerekli sanal dizin oluşturulmuş olur.

3.3.2. Arka Planda Kod İçeren Sayfalar (Code Behind)

Sunucu taraflı yazılım geliştirme teknolojilerinden olan ASP'nin sağladığı birçok kolaylıklarından biri dinamik içerik sunan sayfalar yapılabilmesidir. Teknolojinin işleyişi html sayfalarına dinamik olarak yüklenen kısımlarına asp program kodu eklemektir. Visual Interdev gibi görsel editörler kullanarak <% %> işaretleri arasındaki kısımların sunucu tarafında çalışan kodlar olduğunu anlaşılabilmekte, html kodlarından ayırt edip gerekli değişiklikler veya düzeltmeler yapılabilmekteydi. Bir sayfada yer alan sunucu taraflı kod sayısı oldukça büyümesi, aynı kod üzerinde çalışan yazılım geliştirici sayısı artması sonucu html kodları arasına serpiştirilmiş yüzlerce sunucu taraflı kod yığıntısı ile karşılaşılmaktaydı. .NET platformuna geçişte bu sorun giderildi. Artık ASP.NET ile içeriği program kodundan ayıştırmak mümkün olmaktadır. Bunun avantajları ise ;

- Uygulama geliştirme grubuna ait çalışanlar tamamen kendi geliştirdiği program kodları üzerinde çalışabilmekte, böylelikle ekipte bulunan hiç kimse diğerinin yaptığı çalışmadan dolayı engellenmemekte, örneğin tasarım ile uğraşan kişi bu işini sürdürürken aynı sayfada yer alan fonksiyonlar için bir yazılım geliştirici eş zamanlı çalışabilmektedir.
- İçeriğin program kodundan ayrışması ile uygulama geliştiriciler kendilerine en uygun geliştirme aracını ve programlama dilini kullanabilmektedir.
- Görsel içeriği hazırlayan ekip istediği tasarım programı veya editörü kullanarak sayfalar üzerinde değişiklikler yapabilmektedir.

Arka Planda Kod İçeren Sayfalar (Code Behind) programlama, tasarım ile kodlamayı birbirinden ayırmak ve böylelikle daha hızlı web uygulamaları geliştirmenin yanında kodumuzun bakımı ve yeni versiyonlarını üretmede de daha hızlı ve etkin çözümler üretebilmektir. Aslında Code-Behind programlama da iki farklı şekilde yapılabilir. Birincisinde program kodumuz (C# veya VB.NET) ayrı bir sayfada yer almaktadır. Bu kod sayfasına aspx sayfasından referans verilmekte ve kodumuz çalışma anında MSIL'e derlenmektedir. İkinci metotda ise, program kodu dll olarak derlenip aspx sayfamızda direk olarak derlenen dll'e referans verilmektedir.

ASP.NET ve Visual Studio.NET tarafından cs uzantılı dosyalar, ait olduđu aspx dosyasının ismi aspx + ve + cs şeklinde adlandırmaktadır. Örneğin aspx dosyasının ismi AspNetCodeBehind.aspx olsun bu durumda C# kod dosyası AspNetCodeBehind.aspx.cs şeklinde isim almaktadır (Nacarođlu, 2002).

3.4. C#

C#, C/C++ ve Java dilerinden türeyen, güçlü, basit, esnek, tip güvenli, modern ve Microsoft .NET platformu için sıfırdan geliştirilmiş tek programlama dilidir. C# orta düzeyli diller grubuna girer. Özellikle hem alt düzey hem üst düzey programlar yazılabilir (Algan, 2003).

C# .NET Microsoft tarafından geliştirilmiş, nesneye yönelik bir programlama dilidir. Yapı ve mantık olarak daha Önceki Java, Visual C++'a benzemekle birlikte .NET in getirdiđi bir çok yenilikler ve deđişimleri içermektedir. Her şeyden önce C# .NET Framework'ünü ve CLR yi tam olarak desteklemektedir (Demirkol, 2005).

3.5. Structure Query Language (SQL)

SQL, veritabanlarında çeşitli işlemler (sorgulama, veri girme, silme, güncelleme ya da tablo yaratma gibi) yapmak üzere tasarlanmış yapısal bir sorgulama dilidir (Taşdelen, 2003).

SQL bir veri tabanı ile iletişim kurmak için kullanılır. ANSI standardına göre ilişkisel veri tabanı yönetim sistemlerinin standart dilidir. SQL, bir bilgisayar veritabanı tarafından saklanan verileri düzenlemek, yönetmek ve elde etmek için kullanılan bir araçtır. Aslında SQL, ilişkisel veritabanı adı verilen özel tipte bir veritabanı ile çalışır. Bu veritabanı sisteminin bulunduğu bilgisayar sistemi bir işyerinde yer alıyorsa üretim, satışlar veya faturalar hakkındaki bilgileri saklayabilir. Kişisel bir bilgisayarda ise, yazılan çekler, telefon numaraları veya daha büyük bir bilgisayar sisteminden alınan veriler saklanabilir. Veritabanını kontrol eden bilgisayar programına veritabanı yönetim sistemi (database management system)

veya DBMS adı verilir (Groff ve Weinberg, 2003). DBMS, bir veriabanının kurulması, yaşatılması için gerekli bilgisayar yazılım sistemidir. Veritabanı yönetim sisteminde standart bir dil ile uygulamalara ve kullanıcılara hizmet verilir (Çubukçu, 2001).

Bir veritabanından veri alınması gerektiğinde, bu isteği belirtmek için SQL dili kullanılır. DBMS, SQL isteğini işler, istenen veriyi elde eder ve bunu geri döndürür. Bir veritabanından veri isteyip sonucu alma işlemine, veritabanı sorgusu adı verilir. Structured Query Language (Yapısal Sorgu Dili) adı da buradan gelir. SQL bir DBMS'in kullanıcılar için sunduğu tüm işlevleri kontrol etmek amacı ile kullanılır bu işlevler arasında aşağıdakiler işlevler bulunur.

- **Veri Tanımlama.** SQL, bir kullanıcının saklanan verilerin yapısını, düzenlenişini ve birbirleriyle olan ilişkilerini tanımlamasına izin verir.
- **Veri elde etme.** SQL, bir kullanıcının veya uygulama programının veri tabanından veri almasına ve kullanmasına izin verir.
- **Veri üzerinde değişiklik yapma.** SQL, bir kullanıcının veya uygulama programının, yeni veriler eklemek, eski verileri silmek ve daha önceden saklanmış olan verileri değiştirmek yoluyla veri tabanını güncellemesine izin verir.
- **Erişim Kontrolü.** SQL, bir kullanıcının verileri elde etme, değiştirme, silme ve verilere ekleme yapma yeteneğini kısıtlayarak, saklanan verileri yetkisiz erişimden korumak amacıyla kullanılabilir.
- **Veri paylaşımı.** SQL, eş zamanlı çalışan kullanıcıların veri paylaşımını ayarlayarak, birbirleri ile çalışmalarını sağlamak için kullanılabilir.
- **Veri bütünlüğü.** SQL, veritabanındaki bütünlük kısıtlamalarını tanımlayarak tutarsız güncellemelerden veya sistem hatalarından kaynaklanan bozulmalardan korur.

Tüm bu özelliklerden dolayı SQL, bir veritabanı yönetim sistemini kontrol etmek ve onunla etkileşime girmek için oldukça gelişmiş bir dildir (Groff ve Weinberg, 2003).

3.5.1. SQL Server'ın Özellikleri

SQL ne tek başına bir veritabanı yönetim istemi, ne de başlı başına bir üründür. Bunun yerine SQL, bir veritabanı yönetim sisteminin bütünleşik bir parçası, bir dil ve DBMS ile iletişim kurmak için bir araçtır. SQL farklı özelliklere sahiptir.

- SQL, etkileşimli bir sorgu dilidir. Kullanıcılar veri almak için etkileşimli bir SQL programına komutlar yazarlar ve bunu ekran üzerinde görüntülerler. Bu da, yoğun veritabanım sorguları için uygun ve kullanımı kolay bir araçtır.
- SQL bir veritabanım programlama dilidir. Programcılar, veritabanındaki verilere erişebilmek için uygulama programlarına SQL komutları gömerler. Hem kullanıcı tarafından yazılmış programlar, hem de küçük veritabanım uygulama programları (rapor yazıcılar ve veri giriş araçları gibi) veritabanına erişim için bu tekniği kullanır.
- SQL bir veritabanım yönetim dilidir. Bir küçük bilgisayar veya ana bilgisayar veritabanını yönetmekle sorumlu olan veritabanım yöneticisi, veritabanım yapısını belirlemek ve depolanmış verilere erişimi kontrol etmek için SQL kullanır.
- SQL bir istemci/sunucu dilidir. Kişisel bilgisayar programları, paylaşılan veri depolayan veritabanım sunucuları olan bir ağ üzerinden iletişim kurmak için SQL kullanır. Bu istemci/sunucu mimarisi, ticari tipteki uygulamalarda son derece popüler hale gelmiştir.
- SQL, bir Internet veri erişim dilidir. Kurumsal verilerle etkileşime giren Internet Web sunucuları ve Internet uygulama sunucularının tümü SQL'i, kurumsal veritabanlarına erişimde standart dil olarak kullanır.
- SQL dağınık bir veritabanım dilidir. Dağınık veritabanım yönetim sistemleri, birbirine bağlı birçok bilgisayar sistemi arasında verilerin dağıtılmasına yardımcı olarak SQL kullanır. Her bir sistem üzerindeki DBMS yazılımı SQL'i, veri erişimi için istekler göndererek diğer sistemlerle iletişim kurmak için kullanır.
- SQL bir veritabanım ağ geçidi [gateway) dilidir. Farklı DBMS ürünlerinin karışımından oluşan bir bilgisayar ağında SQL çoğunlukla bir DBMS

markasının başka bir markayla iletişim kurmasını sağlayan bir ağ geçidi olarak kullanılır (Groff ve Weinberg, 2003).

3.6. Control Area Network (CAN)

Kontrol Alan Ağı (CAN-Control Area Network) protokolü, endüstriyel ortamlarda kullanılmak üzere bir Alman firması olan Robert Bosch GmbH tarafından dizayn edilmiş olup, ilk olarak 1989 yılında kullanılmaya başlanmıştır (Lawrenz, 1995). Çok kısa bir süre içerisinde de sağlamış olduğu yararlar sebebiyle bir endüstri standardı haline gelmiştir. Bugün CAN protokolünü sağlayan Çip'ler pek çok dünya firması tarafından üretilmektedir. Siemens, Motorola, İntel ve Philips bu firmaların önde gelenlerindedir. 1997 yılı itibariyle 25 milyon adet CAN Çip'inin üretilerek satılmış olması; 1998 de ise rakamın 40 milyona yükselmesi bu alandaki ihtiyacın boyutunu ortaya koymaktadır.

CAN protokolünün en önemli özelliklerinden bazılarını şu şekilde belirtebiliriz:

- Tek bir merkezden tüm sistemin kolaylıkla kontrolü,
- Sistemin çalışmasını durdurmadan yeni kontrol noktaları eklenebilme olanağı,
- Tek bir kablo üzerinden gönderilen komutlar sayesinde karmaşık kablo yapısından kurtuluş,
- Güvenilir hata algılama,
- Öncelikli mesaj iletim olanağı,
- Mesaj transferi için farklı yapılarda kablo kullanılabilme özelliği (Putzmeister ve Rupp,1994).

CAN protokolü uygulamaları için paralel, bükümlü, koaksiyel veya fiber optik kabloların kullanımı mevcuttur (Kutlu vd., 1996).

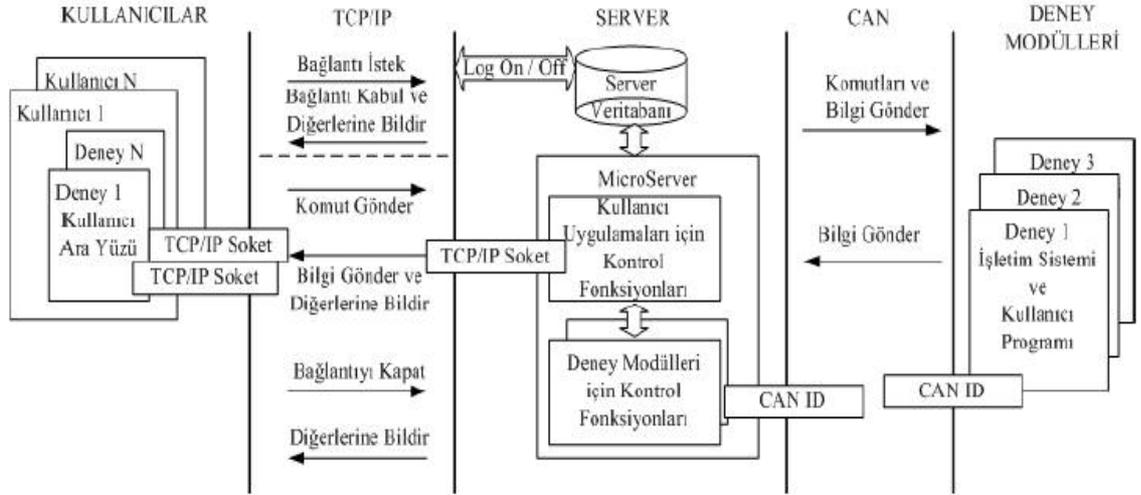
4. ARAŞTIRMA VE BULGULAR

4.1. Mühendislik Eğitimi İçin İnternete Dayalı, İnteraktif, Sanal Mikrodenetleyici Laboratuvarı (MicroLab)

MicroLab, elektrik - elektronik, elektronik - haberleşme ve bilgisayar mühendisliği ile Teknik Eğitim Fakülteleri' nin elektronik – bilgisayar eğitimi bölümünde okutulmakta olan mikrodenetleyici dersi için 8051 tabanlı internete dayalı laboratuvardır. Bu laboratuvar, deneylerin sadece sanal olarak gerçekleştirildiği bir laboratuvar ortamı değil deneylerin internet üzerinden gerçek elektronik devreler kullanılarak yapılabildiği ve sonuçların gerçek zamanlı olarak alınabildiği gerçek bir mikrodenetleyici laboratuvarıdır. Bu laboratuvarın diğer bir önemli özelliği ise aynı anda birden fazla öğrencinin laboratuvara bağlanarak “deney yöneticisi” olarak deney yapma veya “misafir öğrenci” olarak deneyleri izleme imkanı sağlanmasıdır. Bu mikrodenetleyici laboratuvarı sayesinde öğrenciler, laboratuvar çalışmalarını belirli zaman ve mekana bağlı kalmadan internet üzerinden istenilen yer ve zamanda gerçekleştirebilmektedirler (Kutlu vd.,2005).

4.1.1. Yazılım

MicroLab, C# programı kullanılarak dizayn edilmiş MicroClient ve MicroServer bölümlerinden oluşmaktadır (Kutlu, 2004). Kullanıcı adları ve şifreleri SQL veritabanı programı kullanılarak server programı üzerinde tutulmaktadır. Çalışmanın yazılım mimarisi Şekil 4.1 ' de gösterilmiştir (Taşdelen, 2004).



Şekil 4.1 Yazılım Mimarisi

Her bir deney modülü için server üzerinde çalışan kontrol fonksiyonları bulunmaktadır. Bu kontrol fonksiyonları kullanıcı TCP (Transmission Control Protocol) komutlarını CAN mesajlarına çevirmekle görevlidir. Deneylere bağlanabilmek için Client tarafından gönderilen kullanıcı TCP komutları CAN mesajlarına çevrilerek CAN hattına transfer edilir. Transfer fonksiyonları Kvaser firmasının CANlib SDK (Software Development Kit) C# kütüphaneleri kullanılarak gerçekleştirilmiştir. Client ve server için kullanılan TCP komutları Çizelge 4.1' de listelenmiştir (Taşdelen, 2004).

Çizelge 4.1 Client ve Server Komutları

TCP Komut Adı	Gönderen	İşlem
CONN	Client	Bağlantı İsteği
JOIN	Server	Bağlantı Kabul
LIST	Server	Client'a Kullanıcı Listesini Gönder
GONE	Client	Bağlantıyı Sonlandır
CHAT	Client	Metin Mesajı Gönder
CHAT	Server	Metnin Client'a Transfer et
PRIV	Client	Özel Metin Mesajı Gönder
PRIV	Server	Client'a Özel Metin Mesajı Transfer et

Çizelge 4.1 Client ve Server Komutları (Devam)

PASS	Client	Modül Kontrolüne Geç
PASS	Server	Modül Kontrolüne Geç
IPOINT	Client	Switch'lerin Durumunu Gönder
SWTC	Server	Client'lara Switch'lerin Durumunu Transfer et
EDS	Server	Client'lara Deney Modülünün Bilgisini Transfer et
START	Client	Deney Modülünü Başlat
RESET	Client	Deney Modülünü Resetle
PROG	Client	Deney Modülünü Programlamaya Başla
FILE	Client	Server'a Kullanıcı Programını Yükle
OK	Server	Transfer Tamamlandı Modül Programlandı

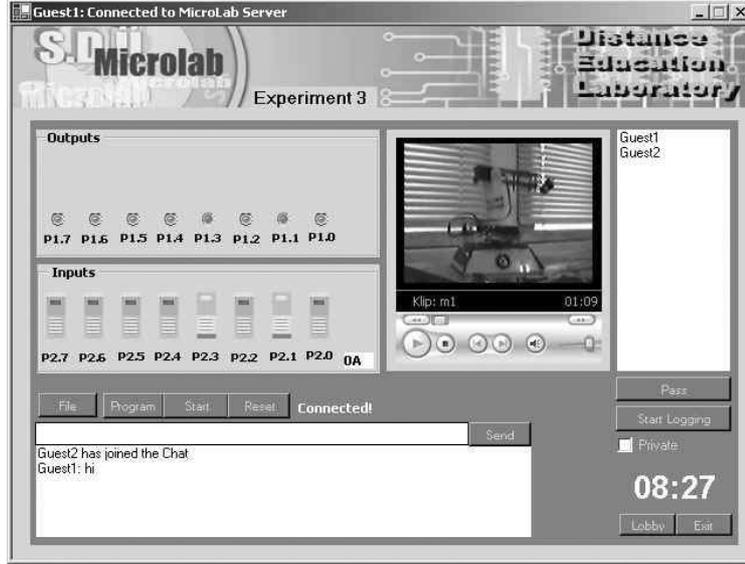
CAN tanımlayıcıları 11 bit uzunluğundaki standart ve 29 bit uzunluğundaki extended tanımlayıcılar olmak üzere iki çeşittir. Bu çalışmada 11 bit tanımlayıcı uzunluğu yeterli olduğu için standart CAN tanımlayıcıları kullanılmıştır. Çizelge 4.2' de deney modülleri için tanımlayıcı haritası görülmektedir. Bu çizelgeye göre, iki MSB (Most Significant Bit) biti sistemdeki sınıfların sayısını tanımlamak için yerleştirilmiştir. Sınıflar için 2 bit ayrıldığından dolayı 00, 01, 10 ve 11 olmak üzere sadece 4 farklı sınıf uygulanabilir. 11 tanımlayıcı bitin 5 tanesi deney modüllerini tanımlamak için yerleştirilmiştir. Bir sınıf için $2^5 = 32$ tane farklı deney modülü kullanılabilir. 4 LSB (Least Significant Bit) bitinin kullanımıyla her bir deney modülü için $2^4 = 16$ farklı komut uygulanabilir. Çizelge 4.2' de deney modülleri için hazırlanmış tanımlayıcı haritası görülmektedir.

Çizelge 4.2 Deney Modüllerinin tanımlayıcı haritası

Standard ID											Hex-Dec	Kullanımı
10	9	8	7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	0	0	0	0	000	ID_SELECT_NODE
0	0	0	0	0	0	0	0	0	0	1	001	ID_PROG_START
0	0	0	0	0	0	0	0	0	1	0	002	ID_PROG_DATA
0	0	0	0	0	0	0	0	0	1	1	003	ID_DISP_DATA
0	0	0	0	0	0	0	0	1	0	0	004	ID_WRITE_COMMAND
0	0	0	0	0	0	0	0	1	0	1	005	ID_READ_COMMAND
0	0	0	0	0	0	0	0	1	1	0	006	ID_ERROR
Sınıf			Deney moduüü				Komutlar					
0	0	0	0	0	0	0	1	1	1	1	01F	IDMASK_MODULE_1
0	0	0	0	0	0	1	0	0	0	0	010-16	BOOTLOADER_START(M1)_RX
0	0	0	0	0	0	1	0	0	0	1	011-17	MONITOR_P2(M1)_TX
0	0	0	0	0	0	1	0	0	1	0	012-18	WRITE_P1(M1)_RX
0	0	0	0	0	0	1	0	0	1	1	013-19	SET_PARAM(M1)_RX
0	0	0	0	0	0	1	0	1	0	0	014-20	RESET(M1)_RX
0	0	0	0	0	0	1	0	1	0	1	015	
...
0	0	0	0	0	0	1	1	1	1	1	01F	
0	0	0	0	0	0	1	0	1	1	1	02F	IDMASK_MODULE_2
0	0	0	0	0	1	0	0	0	0	0	020-32	BOOTLOADER_START(M2)_RX
0	0	0	0	0	1	0	0	0	0	1	021-33	MONITOR_P2(M2)_TX
0	0	0	0	0	1	0	0	0	1	0	022-34	WRITE_P1(M2)_RX
0	0	0	0	0	1	0	0	0	1	1	023-35	SET_PARAM(M2)_RX
0	0	0	0	0	1	0	0	1	0	0	024-36	RESET(M2)_RX
...
0	0	0	0	0	1	0	1	1	1	1	02F	

4.1.2. Sistemin Çalışması

Mikro denetleyici deney modülü birçok port kontrolünden oluşmaktadır. MicroLab da dört farklı deney modülü uygulanır. Bu yüzden kullanıcı ara yüzleri mevcut deney modülleri ile birlikte tasarlanmıştır. Örneğin robot kol deneyinin kullanıcı ara yüzü aşağıdaki Şekil 4.2 de gösterilmiştir (Kutlu, 2004).

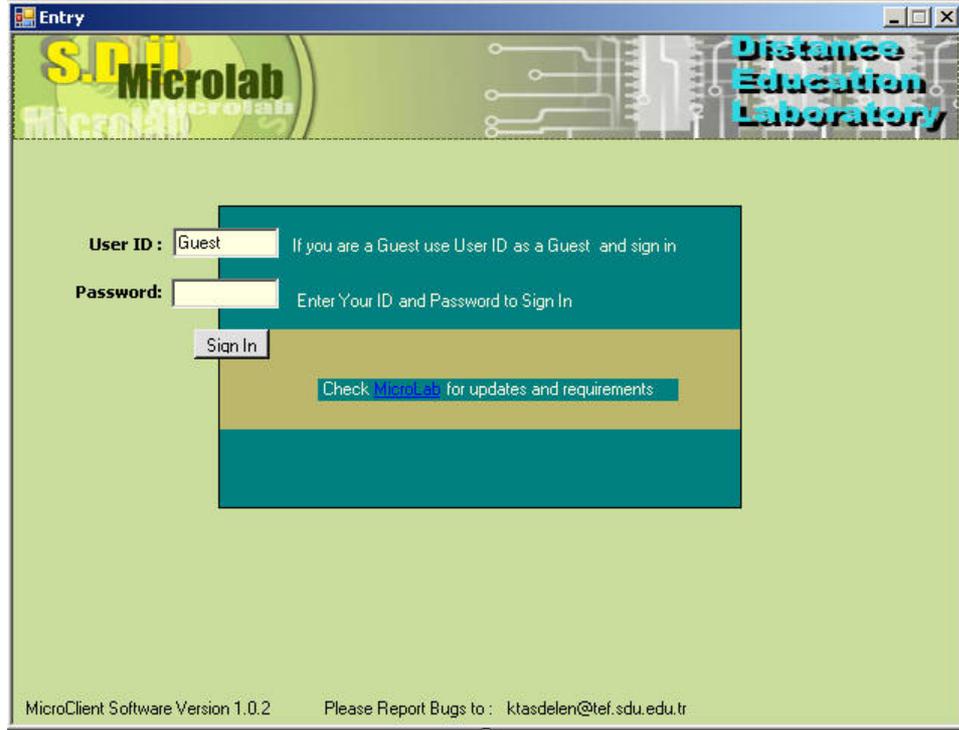


Şekil 4.2 Deney İçin Kullanıcı Ara Yüzü (Kutlu,2004)

MicroClient programı çalıştığı zaman kullanıcı öğrenciye kullanıcı adı ve şifre sorulur. Kullanıcı kayıtlı değilse oturuma guest (misafir) kullanıcı olarak giriş yapması mümkündür. Öğrenci oturuma guest olarak girdiği zaman diğer guest kullanıcılarından yeni guest kullanıcıasını ayırmak için ona ardışık guest numaralarından biri otomatik olarak verilecektir.

Kullanıcı adı ve şifre doğrulandığında öğrenci kullanılabilir deney modüllerinin listelendiği lobby diye adlandırılan ana giriş de bekler. "Lobby" sayfasında deney modülleri ile ilgili bilgilerin bulunduğu web bağlantıları bulunmaktadır. Bu bağlantılar sayesinde öğrenci istediği deney modülü ile ilgili bilgilere ulaşabilir (Kutlu, 2004).

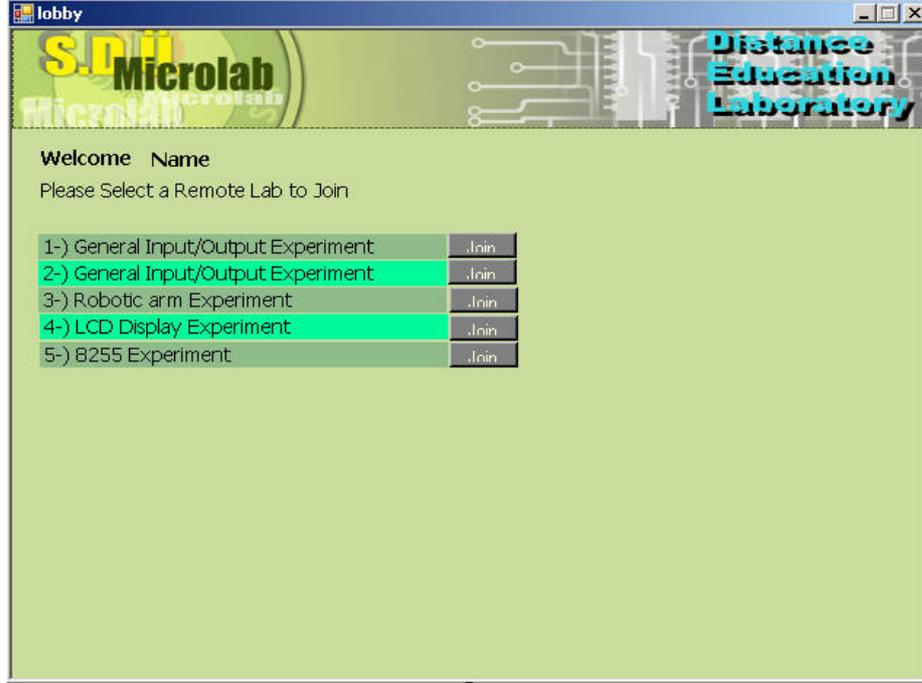
Öğrenci internet üzerinden deney sayfasına bağlandığında Şekil 4.3' da görüldüğü gibi bir giriş sayfası ile karşılaşacaktır.



Şekil 4.3 Giriş Sayfası

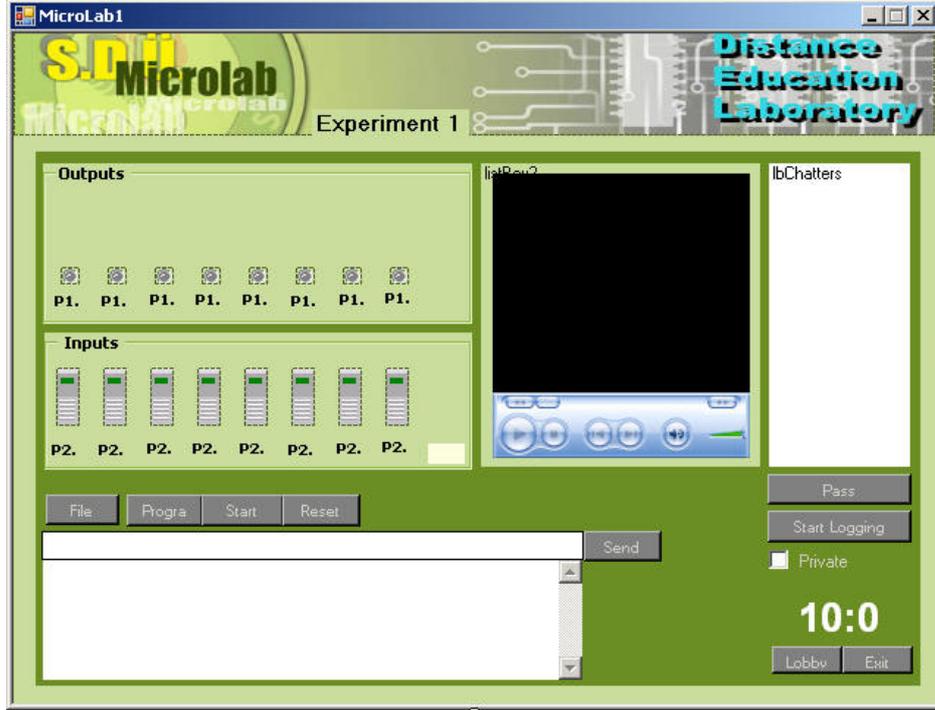
Bu sayfada öğrencilere, kullanıcı adları ve şifreleri sorulmaktadır. Kullanıcı adı ve şifresi olmayan öğrenciler deneye "Guest" kullanıcı adı ile herhangi bir şifreye ihtiyaç duymadan bağlanabilirler. Öğrenci deneye kendi kullanıcı adı ve şifresi ile girmişse "Online Kişiler" bölümünde kullanıcı adı yazacaktır. Eğer "Guest" kullanıcı adı ile bağlanmışsa kendisine veritabanından verilen "Guest" numarası ile diğer misafir kullanıcılardan ayrılacaktır.

Kullanıcı adı ve şifresinin sorulduğu giriş sayfasından giriş yapan öğrenci "Lobby" olarak adlandırılan deneylerin listesinin bulunduğu sayfaya geçecektir. Bu bağlantılar sayesinde Öğrenci istediği deney modülü ile ilgili bilgilere ulaşabilir. Şekil 4.4 de "Lobby" sayfası görülmektedir.



Şekil 4.4 Lobby Sayfası

Öğrenci "Lobby" sayfasından yapmak istediği deney modülünün linkine tıklayarak deneye bağlanıldığında, deney modülünün ara yüzü ekrana gelir. Deneye ilk bağlanan öğrenci o deney modülünün yöneticisidir ve her deney için sadece bir deney yöneticisi olabilir. Deney yöneticisi, deney modülünü programlayarak deneyi gerçekleştirirken diğer kullanıcılar deneyi gözlemci olarak izleyebilirler fakat deneye müdahale etme yetkileri yoktur. Fakat deney modülünün ara yüzünde bulunan "Chat" kısmını kullanarak birbirleriyle haberleşebilirler ve deney sonuçlarını tartışabilirler. Şekil 4.5 de I/O deney modülü için kullanıcı ara yüzü görülmektedir.



Şekil 4.5 Kullanıcı Arayüzü

Deney yöneticisine deneyi gerçekleştirme için verilen süre 10 dakikadır. Deney için kalan süre deney modülünün ara yüzünde görülebilmektedir. Bu süre laboratuvar yöneticisi tarafından ayarlanabilir. Bu süre zarfında deney yöneticisi deneyi bitirmek zorundadır. Deneyin bu süre içinde bitmemesi durumunda deney yöneticisinin oturumu kapatılacaktır. Bu durumda deney yöneticisi deneye ikinci sırada bağlanan öğrenci olacaktır. Eğer deney süresi bitmeden deney yöneticisi deneyi bitirirse veya deneyi yarım bırakmak isterse deneyi herhangi bir öğrenciye devredebilir. Deneyi devralan yeni yönetici deneye kaldığı yerden devam edebilir veya yeniden başlayabilir. Deney yöneticisi, kullanıcı ara yüzündeki düğmeleri kullanan tek kişidir, bu düğmeler diğer öğrencilerin kullanıcı ara yüzlerinde pasif durumdadır.

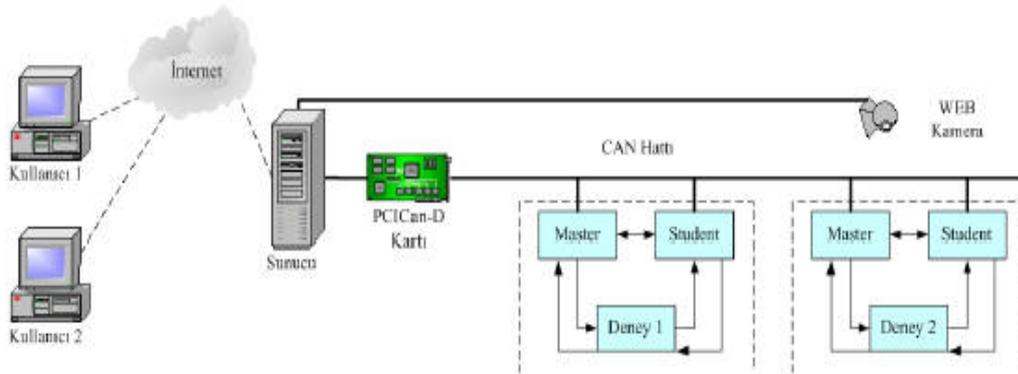
Deneyi bitiren veya yarım bırakan öğrenci, diğer bir deney modülüne bağlanabilir. Bunun için giriş sayfasından tekrar kullanıcı adı ve şifresiyle giriş yapmasına gerek yoktur. “Lobby” sayfasına geri dönülerek istenilen deneye bağlanılabilir.

Deney yöneticisinin bir deney modülünde çalışabilmesi için 8051 mikro denetleyicisi için yazılan hex dosyayı deney modülüne yüklemesi gerekmektedir. Dosyayı yüklemek için "File" düğmesi kullanılır. Yüklenecek dosyanın formatı sadece hex olmak zorundadır. Diğer dosya türleri "client" programı tarafından reddedilecektir. Öğrenci program düğmesine basarak hex dosyanın, "server" programı aracılığıyla deney modülüne yüklenmesini sağlar. Eğer deney yöneticisi deneyden çıkarsa veya deney yöneticiliğini başka bir öğrenciye devrederse deney modülüne yüklenen hex dosya server programı tarafından otomatik olarak silinecektir.

Deney modülünün kullanıcı ara yüzünde bulunan switch'ler ve ledler deneyleri görsel olarak gözlemlemek için kullanılmıştır. Switch'ler "on" ve "off" olmak üzere İki kademelidir ve giriş portlarını temsil etmektedir. Ledler ise çıkış portlarını göstermektedir. Kullanıcı ara yüzündeki ledlerin durumu, deney devresi üzerinde çalışan ledlerin durumu ile aynıdır yani deney sonuçları eş zamanlı olarak gözlenebilmektedir. LCD deney modülünde, robot kol deney modülünde ve 8255 deney modülünde deney sonuçlarını gözlemleyebilmek için kamera kullanılmıştır. Bu sayede deney sonuçlarının İnternet üzerinden canlı olarak izlenebilmesi sağlanmıştır (Taşdelen, 2004).

4.1.3. Donanım

MicroLabın donanım kısmı Şekil 4.6 da gösterildiği gibi CAN modülleri ve deney modülleri olmak üzere iki bölümden oluşmaktadır (Taşdelen, 2004).



Şekil 4.6 8051 Tabanlı Mikrodenetleyici Laboratuvarı (Taşdelen, 2004).

4.1.3.1. CAN Modülleri

CAN modüllerinin yapılması için temel devre elemanlarının yanı sıra MAX232, içinde mantıksal OR kapısı bulunan 7432 entegreleri, 8051 tabanlı CAN entegresi T89C51CC01 ve transceiver entegresi PCA82C251 kullanılmıştır.

Gerçekleştirilen CAN modüllerinin çalışmalarını kontrol etmek ve bu modüllerin seri porttan programlanmasını sağlamak için devreye RS232 portu eklenmiştir. RS232 portu sayesinde gerçekleştirilen CAN modüllerinin CAN hattına bağlanan CAN port ve transceiver kısımları hariç diğer bölümlerinin çalışıp çalışmadığı kontrol edilebilmektedir. Bu kontrol için bilgisayar ile CAN modülü seri port aracılığıyla birbirine bağlanır (Taşdelen, 2004).

4.1.3.2. Deney Modülleri

Tasarlanan 8051 tabanlı mikro denetleyici laboratuvarında Öğrencilerin uygulama yapabilmesi için 3 adet deney modülü hazırlanmıştır. Bu deney modüllerinden 2 tanesi ledlerle yapılan I/O modülleri, 1 tanesi LCD display deney modülüdür.

T89C51CC01 CAN entegresinin 4 adet portu vardır. CAN modülünü programlama ve çalışma moduna geçirmek için 3 numaralı port kullanıldığından deney modüllerini bağlamak için diğer 3 porttan herhangi bir tanesi kullanılabilir (Taşdelen, 2004).

4.1.4. Eksiklikleri

MicroLab'ın en büyük eksikliklerinden biri ortak bir merkezden yönetimini sağlayacak bir yönetim panelinin bulunmamasıdır.

MicroLabta kullanıcıların giriş çıkış kayıtlarını tutan bir sorgu yapılmamaktadır.

Hangi öğrencilerin, hangi deneylere, ne zaman bağlandıkları ne kadar süre ile bu deneyde kaldıklarının bilgisi tutulmamaktadır.

Deneylerin aktif ve pasifliğini belirleyici bir sorgu bulunmamaktadır.

MicroLab programında op (operatör) durumundaki kullanıcı dosya yüklerken diğer kullanıcılar dosya yüklendiğini görememektedirler. Bunu yalnız birbirlerine yazarak haber verebilmektedirler.

4.1.5. Yapılan Değişiklikler

Veritabanındaki “Prog” isimli tablonun “Aktif” alanı kontrol edilebilir duruma getirildi. Bu alan ile deneylerin aktiflik pasiflikleri belirlenmektedir.

Kullanıcılar ile ilgili giriş çıkış kayıtlarının tutulabilmesi için veritabanına “userLog” tablosu eklenmiştir. Bu tablo id, name, role, expid, connDate, connTime, disconnDate, disconnTime alanlarından oluşmaktadır.

Mevcut chatobject yazılımında bulunan deney modülleri yazılımları usercontrol1, usercontrol2, usercontrol3, usercontrol4 de bulunan “regexperiment()” fonksiyonuna kullanıcı giriş yaptığında “userlog” tablosuna aşağıdaki kodlar ile giriş kaydı atılmaktadır. Buradaki dikkat edilecek nokta guest’in op olma izninin olmadığı yani guest’ler deneyleri izleyici konumunda olduğu durumlarda giriş kaydı direk atılmakta, değilse “opcontrol() “ fonksiyonu çağrılmaktadır. Aşağıdaki kodlarda “cnDt” bağlantı tarihi (connectionDate), “cnTm “ bağlantı saati (connectionTime) bilgileri için kullanılmaktadır.

```
if (rakam == 0)
{
    butonkontrol(0);
    string cnDt, cnTm;
    cnDt = DateTime.Now.ToShortDateString();
    cnTm = DateTime.Now.ToShortTimeString();
    SqlCommand cm1 = new SqlCommand("INSERT INTO USERLOG (name,
role, expid, connDate, connTime, disconnDate, disconnTime) values
```

```

    (" + inputname + "', 'NO', '" + Convert.ToString(deneyno) + "', '"
+ cnDt + "', '" + cnTm + "')", a);
    cm1.ExecuteNonQuery();
}
else
opcontrol();

```

Yukarıda görüldüğü üzere “opcontrol()” fonksiyonu guest’in operatör olma izni olduğunda ya da kullanıcı adı guest’in haricinde bir isim olduğunda çalışmaktadır. “opcontrol” fonksiyonuna bu işlev için aşağıdaki kodlar eklenmiştir.

```

if(isaret.StartsWith("Guest") && kabul==0)
{
    label18.Text="Guests can't be an
operator";
}
else
{
string destclient = lbChatters.SelectedItem.ToString();
string command = "PASS|" + clientname + "|" + destclient;

Byte[] outbytes =
System.Text.Encoding.ASCII.GetBytes(command.ToCharArray());

    ns.Write(outbytes,0,outbytes.Length);
        butonkontrol(2);
        zaman=false;
        label19.Visible=false;

SqlDataAdapter adp = new SqlDataAdapter("SELECT TOP 1 * FROM USERLOG
WHERE name = '" + clientname + "' AND expid = '" +
Convert.ToString(deneyno) + "' ORDER BY ID DESC", a);

DataSet ds = new DataSet();
adp.Fill(ds, "Log");

string id = ds.Tables["Log"].Rows[0][0].ToString();
ds.Dispose();
SqlCommand cm = new SqlCommand("UPDATE USERLOG SET role = 'NO' WHERE
id = " + id + "'", a);

SqlCommand cm1 = new SqlCommand("UPDATE USERLOG SET role = 'OP'
WHERE id = " + id + "'", a);
    try
    {
        a.Open();
        cm.ExecuteNonQuery();
        cm1.ExecuteNonQuery();
    }
    catch (Exception hata)
    {
        Console.Write(hata.Message);
    }
}

```

```

    }
}

else
{
    label18.Text="You are allready operator";
}

else
{
    label18.Text="You are allready operator";
}

```

Çıkış yaptığı zamanki çıkış tarih ve zamanının tutulabilmesi için “Disconnect” fonksiyonuna aşağıdaki kodlar eklenmiştir.

```

string dnDt, dnTm;
dnDt = DateTime.Now.ToShortDateString();
dnTm = DateTime.Now.ToShortTimeString();

SqlConnection a = new SqlConnection();

a.ConnectionString = "workstation id=" + Entry.servername + ";packet
size=4096;user id=student;password=a;data source=" +
Entry.servername + ";persist security info=False;initial
catalog=microlab";

SqlDataAdapter adp = new SqlDataAdapter("SELECT TOP 1 * FROM USERLOG
WHERE name = '" + inputname + "' AND expid = '" +
Convert.ToString(deneyno) + "' ORDER BY ID DESC", a);

DataSet ds = new DataSet();

adp.Fill(ds, "Log");
string id = ds.Tables["Log"].Rows[0][0].ToString();
ds.Dispose();

SqlCommand cm1 = new SqlCommand("UPDATE USERLOG SET disconnDate = '"
+ dnDt + "', disconnTime = '" + dnTm + "' WHERE id = '" + id + "'",
a);

try
{
    a.Open();
    cm1.ExecuteNonQuery();
}

catch (Exception hata)
{
    Console.Write(hata.Message);
}

```

4.2. Mikro Denetleyici Laboratuvarı İçin Sistem Yöneticisi Ara Yüzü (Yönetici Paneli)

Yönetici paneli, MicroLab'ta bulunan MicroClient ve MicroServer bölümlerinin merkezi ayarlarının yapılması ve yönetimini gerçekleştirmek için kullanılmaktadır.

4.2.1. Yönetici Panelinin Yapısı

Admin paneline Şekil 4.7 deki kullanıcı ad ve şifre sorgu ekranı ile giriş yapılmaktadır. Kullanıcı adı metin kutusuna yöneticiye ait kullanıcı adı ve şifre metin kutusuna yöneticiye ait şifre girilerek, doğruluğu sağlandığında server'a bağlantı kurulmaktadır. Şayet kullanıcı adı ve şifresinden herhangi birinin yanlış olması durumunda sistem otomatik olarak yönetici paneli giriş sayfası (default.aspx) tekrar yüklemektedir.



Şekil 4.7 Yönetici Paneli Giriş Paneli

4.2.1.1. Kullanıcı İşlemleri

Kullanıcı adı ve şifresi doğrulanan yönetici, kullanıcı işlemleri paneline (user.aspx) yönlendirilir (Şekil 4.8). Kullanıcı işlem panelinde deney kullanıcıları ile ilgili kullanıcı ekleme, düzenleme, silme, operatör atama ve kayıt silme işlemleri yapılmaktadır.

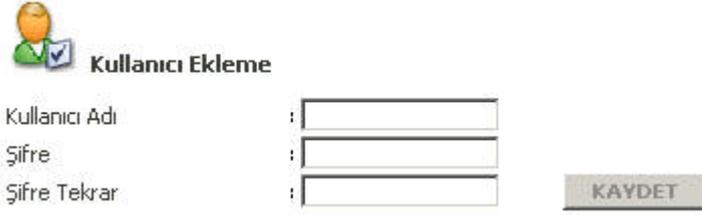


Şekil 4.8 Kullanıcı İşlemleri Paneli

- **Kullanıcı Ekleme**

Kullanıcı işlem panelinde kullanıcı ekleme linkine tıklandığında kullanıcı ekleme paneli (user_add.aspx) açılır (Şekil 4.9). Bu panelde MicroLab sistemini kullanacak yeni kişi eklemesi yapılmaktadır. Eklenecek kişinin kullanıcı adı, şifresi, girilen şifrenin tekrarı yazılarak "KAYDET" butonuna tıklanması ile kayıt işlemi gerçekleşmektedir. ASP.NET'in "required field validator" özelliği kullanılarak kullanıcı adının ve şifrenin boş geçilmesini önleyerek, bu alanların yanında ilgili kutucuğun boş geçilmemesi mesajı verilmektedir. Şayet şifreler uyuşmuyorsa "Şifreler uyuşmuyor. Lütfen kontrol ediniz." mesajı ile yöneticinin girişi tekrar

yapması istenir. Kaydetme işleminde herhangi bir problem çıktığında "Kaydedilemedi. Lütfen tekrar deneyiniz." mesajı ile durum haberi verilmektedir.



Şekil 4.9 Kullanıcı Ekleme Paneli

- **Kullanıcı Düzenleme**

Kullanıcı işlem panelinde kullanıcı düzenleme linkine tıklandığında kullanıcı düzenleme paneli (user_modify.aspx) açılır (Şekil 4.10). Bu panel, MicroLab sistemine önceden kayıtlı kullanıcıların kullanıcı adlarını ve şifrelerini değiştirmek veya düzenlemek için kullanılmaktadır. Kullanıcılar alanında açılabilir liste kutusunda kayıtlı kullanıcılardan düzeltme yapılacak kullanıcının seçimi gerçekleştirildikten sonra "DÜZENLE" butonuna tıklanarak kullanıcı adı veya şifresinin ya da her ikisinin birden düzenlemesi yapılabilmektedir.

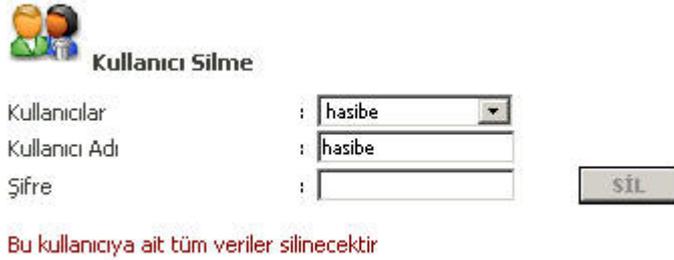


Şekil 4.10 Kullanıcı Düzenleme Paneli

- **Kullanıcı Silme**

Kullanıcı işlem panelinde kullanıcı silme linkine tıklandığında kullanıcı silme (user_delete.aspx) paneli açılır (Şekil 4.11). Bu panel, MicroLab sistemine önceden kayıtlı kullanıcıları silmek için kullanılmaktadır. Kullanıcılar alanında açılabilir liste kutusunda kayıtlı kullanıcılardan silinecek kullanıcının seçimi gerçekleştirildikten

sonra "Bu kullanıcıya ait tüm veriler silinecektir" mesajı ile yönetici uyarılır, yöneticinin "SİL" butonuna tıklaması ile silme işlemi gerçekleştirilir. Eğer kullanıcı aktif konumda ise "Bu kullanıcı şu anda sisteme bağlı durumda. Lütfen daha sonra tekrar deneyiniz." mesajı verilerek silme işlemi iptal edilmektedir. Silme işlemi esnasında herhangi bir problem çıktığında "İşlem başarısız. Lütfen tekrar deneyiniz." mesajı ile yönetici uyarılmaktadır.



Şekil 4.11 Kullanıcı Silme Paneli

- **Operatör Atama**

Kullanıcı işlem panelinde operatör atama linkine tıkladığında operatör atama paneli (user_op.aspx) açılır (Şekil 4.12). Bu panel, deneyi o an kullanan kullanıcılar arasındaki operatörlüğü değiştirmek için kullanılır. Deneyler alanında aktif deneylerin listesi açılabilir liste kutusunda seçilebilmektedir. Kullanıcılar alanında seçilen deneydeki aktif kullanıcıları göstermektedir. Seçimler yapıldıktan sonra "KAYDET" butonuna tıklanarak operatör aktarma işlemi gerçekleştirilmektedir. Kullanıcı zaten operatör ise "Bu kullanıcı zaten operatör." uyarı mesajı verilmektedir. Kullanıcı operatör değilse "Operatör atanmıştır." mesajı verilerek seçilen deneyin tablosunda op'un değerine NULL atanıp yeni operatör kullanıcının op'una true atanarak operatör atama işlemi gerçekleştirilmektedir.



Şekil 4.12 Operatör Atama Paneli

- **Kayıt Sil**

Kullanıcı işlem panelinde Kayıt Sil linkine tıkladığında Kayıt Silme paneli (record_delete.aspx) açılır (Şekil 4.13). Bu panel, kullanıcılar alanından seçilen kullanıcının şuna kadar kaydedilmiş kayıtlarını siler. Kayıt silme işleminden sonra bu kullanıcı için kayıt tutma devam edecektir. Kullanıcılar alanında açılan liste kutusunda mevcut “Tümü” seçeneği ile tüm kayıtlar silinmektedir.



Şekil 4.13 Kayıt Sil Paneli

4.2.1.2. Deneyler

Deneyler linkine tıklayarak yönetici, Deneyler paneline (experiment.aspx) yönlendirilir (Şekil 4.14). Deney panelinde deneylerle ilgili deney ekleme, deney silme, deney düzenleme, deney izleme ve deney süresi / guest kontrolü işlemleri yapılmaktadır.



Şekil 4.14 Deneyler Paneli

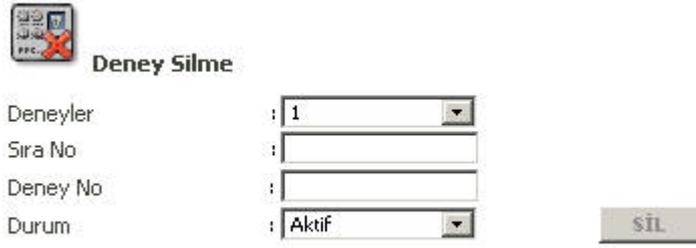
- **Deney Ekleme**

Deneyler panelinde Deney Ekleme linkine tıkladığında deney ekleme paneli (exp_add.aspx) açılır (Şekil 4.15). Bu paneldeki ekleme seçenekleri, mevcut veritabanındaki deney tablolarının (exp1,exp2) alanlarına göre yapılmaktadır. Deneyin sıra numarası, deney numarası ve deneyin çalışıp çalışmadığının bilindiği aktiflik/pasiflik durumu bilgilerinin girişi yapılarak deney modülü oluşturulmaktadır.

Şekil 4.15 Deney Ekleme Paneli

- **Deney Silme**

Deneyler panelinde Deney Silme linkine tıkladığında deney silme paneli (exp_delete.aspx) açılır (Şekil 4.16). Bu paneldeki silme işlemleri, deney ekleme panelindeki seçenekler yöneticinin karşısına sunulmaktadır. Yönetici silmek istediği deneyi seçip diğer özelliklerini de gördükten sonra seçilen deneyde aktif kullanıcı var mı kontrolü yapılır, aktif kullanıcılar için "Bu deneye girmiş kullanıcılar var. Silmeniz halinde bu bilgiler de kaybolacaktır." mesajı verilerek yönetici uyarılmaktadır. "SİL" butonuna tıkladığında deneyi kullanan kullanıcı varsa "Bu deneyi silemezsiniz. Şu an deneyi kullanan kullanıcılar var" mesajı verilerek silme işlemi engellenmektedir. Deneyi kullanan kullanıcı yoksa deney silinir. Bu aşamalarda çıkacak herhangi bir probleme karşı "İşlem başarısız. Lütfen tekrar deneyiniz." mesajı ile durum haberi verilmektedir.



Deney Silme

Deneyler : 1

Sıra No :

Deney No :

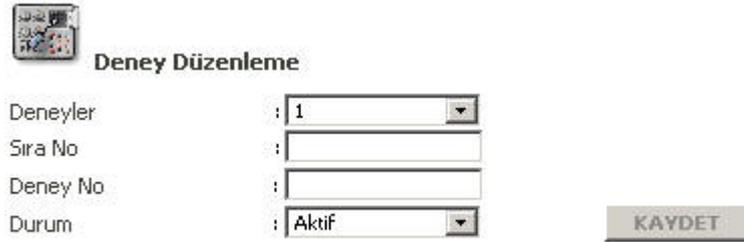
Durum : Aktif

SİL

Şekil 4.16 Deney Silme Paneli

- **Deney Düzenleme**

Deneyler panelinde Deney Düzenleme linkine tıklandığında deney düzenleme paneli (exp_modify.aspx) açılır (Şekil 4.17). Bu panelde deneyle ilgili deneyin sıra numarası, deney numarası, deneyin aktiflik/pasifliği ile ilgili düzenlemeler yapılmaktadır. Deney numarası düzenlemede önceden var olan bir deney numarası verildiyse yönetici "Bu numarayla başka deney kaydedilmiş. Lütfen değiştirip tekrar deneyiniz." mesajı ile uyarılmaktadır.



Deney Düzenleme

Deneyler : 1

Sıra No :

Deney No :

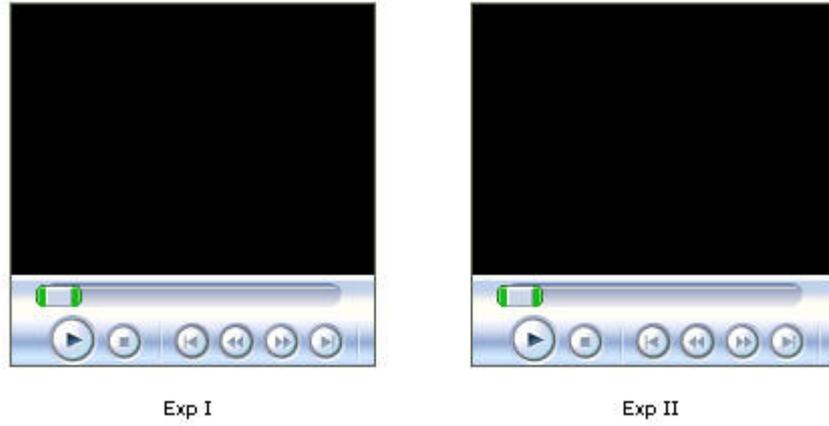
Durum : Aktif

KAYDET

Şekil 4.17 Deney Düzenleme

- **Deney İzleme**

Deneyler panelinde Deney İzleme linkine tıklandığında deney izleme paneli (exp_watch.aspx) açılır (Şekil 4.18). Bu panelde deney modüllerini izlemek için konulan web kameralarının görüntüleri alınmakta ve yöneticiye gösterilmektedir.



Şekil 4.18 Deney İzleme Paneli

- **Deney Süresi / Guest Kontrolü**

Deneyle panelinde Deney Süresi/Guest Kontrolü linkine tıkladığında Deney Süresi/Guest Kontrolü paneli (exp_time.aspx) açılır (Şekil 4.19). Bu panelde Guest kullanıcısının operatörlük izni ayarlanmakta ve kullanıcıların deneyleri çalışma süresinin değişimi buradan sağlanmaktadır.

Deney Süresi/Guest Kontrolü

Guest operatör olabilir mi? : Evet
 Hayır

Deney Süresi :

Şekil 4.19 Deney Süresi/Guest Kontrolü Paneli

4.2.1.3. Raporlar

Raporlar linkine tıklayan yönetici, raporlar paneline (report.aspx) yönlendirilir (Şekil 4.20). Raporlar panelinde aktif kullanıcılar ve kullanıcı hareketleri seçeneklerine ulaşılır.



Şekil 4.20 Raporlar Paneli

- **Aktif Kullanıcılar**

Raporlar panelinde Aktif Kullanıcılar linkine tıklandığında aktif kullanıcılar paneli (users_active.aspx) açılır (Şekil 4.21). Bu panelde deneylerde çalışan tüm aktif kullanıcıların operatörlük durumları, hangi deneylerde oldukları, bağlanma zamanları gösterilerek rapor edilmektedir. Aktif kullanıcıların durumları, deney modüllerine eklenen connect/disconnect alanlarının sorguları ile gerçekleşmektedir. Kullanıcıların disconnect alanları NULL bilgisini içermesi onun hala aktif olduğunu göstermektedir. Bu sorgu kullanılarak kullanıcıların deneyleri terk etmesi durumunda userLog tablosunda disconntime ve disconndate alanına çıkış tarih ve saat bilgisi atılarak kullanıcı işlemleri takip edilmektedir.

Kullanıcı Adı	Deney No	Kullanıcı Rolü	Bağlanma Tarihi	Bağlanma Zamanı
ecir	2	no	01.01.1900 00:00:00	16.06
ecir	1	op	01.01.1900 00:00:00	13.06
ecir	1	op	02.02.2006 00:00:00	14.00
ecir	2	op	01.01.1900 00:00:00	10.00
ecir	2	op	01.01.1900 00:00:00	15.30
hande	1	no	01.01.1900 00:00:00	13.08
hande	1	no	01.01.1900 00:00:00	15.01
hande	1	no	02.04.2006 00:00:00	19.30
hande	1	op	01.04.2006 00:00:00	19.30

Şekil 4.21 Aktif Kullanıcılar Paneli

- **Kullanıcı Hareketleri**

Raporlar panelinde Kullanıcı Hareketleri linkine tıklandığında kullanıcı hareketleri paneli (users_motion.aspx) açılır (Şekil 4.22). Bu panelde yönetici, kullanıcıların o ana kadar yaptıkları çalışmaları görmektedir. Yönetici tüm kullanıcılar ile ilgili bilgileri görmenin yanında tarih, aktif/pasif kullanıcılar, deney numaraları ve kullanıcı adına göre yapacağı sorgulamalar ile de kullanıcılar hakkındaki kayıtları görebilmektedir.

Tarih(1) ve Tarih(2) alanları birbirinden bağımsız çalışmaktadır. Tarih(1) alanına yazılan tarih bilgisi tarih aralığının ilk listelenmesi istenen tarihi olarak alınmakta, Tarih(2) alanına yazılacak tarih bilgisi tarih aralığının son listelenmesi istenen tarihi yazılarak sorgulanmaktadır.

Aktif/Pasif alanları listelenmesi istenen kayıtlar içinde seçime göre aktif kullanıcılar veya pasif kullanıcıların gösterilmesi için kullanılmaktadır.

Deney Numarası alanı ile listelenecek kayıtların istenen deney numarasına göre yapılması sağlanmaktadır.

Kullanıcı Adı alanı ile listelenecek kayıtların istenen kullanıcı adına göre yapılması sağlanmaktadır.

Tarih (1) Aktif/Pasif

Tarih (2) Deney Numarası

Kullanıcı Adı

Kullanıcı Adı	Deney No	Kullanıcı Rolü	Bağlanma Tarihi	Bağlanma Zamanı
ecir	2	no	01.01.1900 00:00:00	16.06
ecir	1	op	01.01.1900 00:00:00	13.06
ecir	1	op	02.02.2006 00:00:00	14.00
ecir	2	op	01.01.1900 00:00:00	10.00
ecir	2	op	01.01.1900 00:00:00	15.30
hande	1	no	01.01.1900 00:00:00	13.08
hande	1	no	01.01.1900 00:00:00	15.01
hande	1	no	02.04.2006 00:00:00	19.30
hande	2	no	05.03.2006 00:00:00	10.30

Şekil 4.22 Kullanıcı Hareketleri Paneli

4.2.1.4. Ayarlar

Ayarlar linkine tıklayarak yönetici, ayarlar paneline (settings.aspx) yönlendirilir(Şekil 4.23). Ayarlar paneli yönetici adı ve şifresinin düzenlemesi için kullanılmaktadır.



Şekil 4.23 Ayarlar Paneli

- **Yönetici Adı Değiştirme**

Ayarlar panelinde Yönetici Adı Değiştirme linkine tıklandığında yönetici adı değiştirme paneli (change_username.aspx) açılır (Şekil 4.24). Eski ad ve yeni ad alanları doldurularak kullanıcı adı değişikliği gerçekleştirilmektedir.

Şekil 4.24 Yönetici Adı Değiştirme Paneli

- **Yönetici Şifresi Değiştir**

Ayarlar panelinde Yönetici Şifresi Değiştir linkine tıklandığında yönetici şifresi değiştir paneli (change_pass.aspx) açılır (Şekil 4.25). Yeni şifre ve yeni şifre tekrar alanları doldurularak yönetici parolası değişikliği gerçekleştirilmektedir.

Yönetici Şifresi Değiştirme

Yeni Şifre :

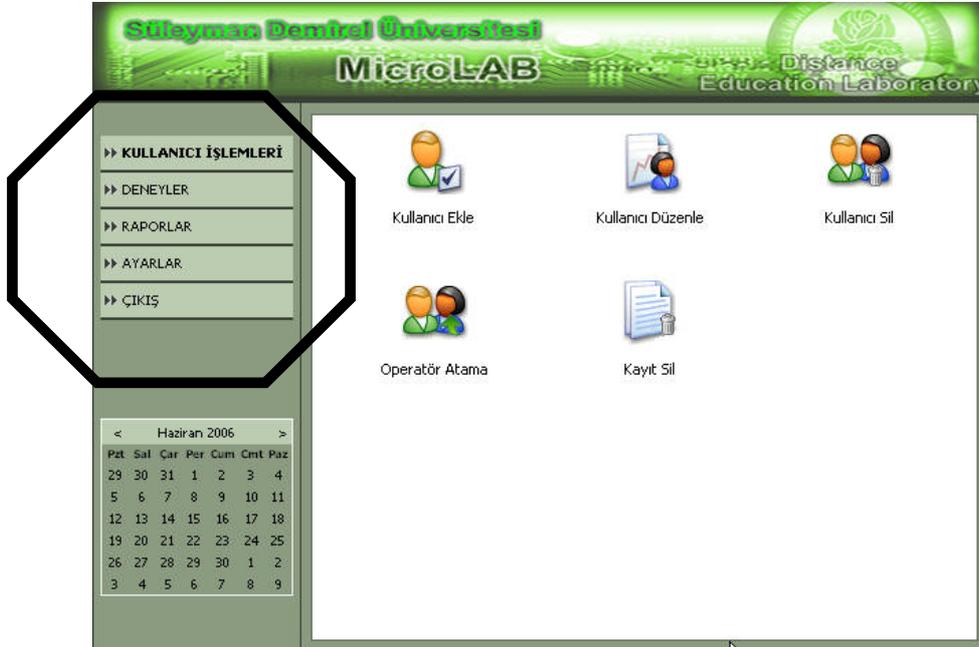
Yeni Şifre Tekrar :

KAYDET

Şekil 4.25 Yönetici Şifresi Değiştirme Paneli

4.2.1.5. Çıkış

Sistemden çıkmak için sol tarafta bulunan menü çubuğundaki Çıkış linki kullanılır (Şekil 4.26).



Şekil 4.26 Çıkış İşlemi

4.2.2. Yönetici Paneli Tasarımı

Yönetici paneli yazılım modülünü gerçekleştirebilmek için aşağıdaki program ihtiyaçları giderildi.

C#, .NET SDK, vb. yazılımlarınızı kurabilmeniz için işletim sisteminizin aşağıda belirtilenlerden birisi olmalıdır.

- Windows NT 4.0 (Service Pack 6)
- Windows 2000 (Service Pack 2) ve 2003 Ailesi
- Windows XP Professional

Bunun yanında .NET Framework'ün olanaklarından tam faydalanabilmek için aşağıda yazılan yazılım paketleri kuruldu. Sisteme kurulan .NET SDK'nın içinde bu yazılım paketleri mevcut olduğundan bu kurulum esnasında diğer gerekli yazılımlarda otomatik kurulmuş oldu.

- Microsoft Internet Explorer 5.01 veya daha üst versiyonu
- Veri kaynaklarına erişmek için Microsoft Data Access Component (MDAC) 2.6 veya daha üst versiyonu. .NET SDK'nın içinde MDAC 2.7 ulaşılmış olacaktır.
- ASP.NET uygulamaları için IIS 5.0 veya en son güvenlik yamaları ile daha üst versiyonları, servera da kurulması gerekli olan en az Framework 2.0 ve IIS yazılımlarıdır.

Bu yazılım paketlerinin yanında sisteminiz en az PII 400 veya üzeri işlemcinin bulunması, 192 MB belleğin bulunması ve 2 GB boş disk alanının bulunması etkin bir şekilde çalışabilmeniz için gerekmektedir.

Alt yapı hazırlandıktan sonra kayıtların tutulabilmesi için SQL Server 2005 veritabanı kullanılarak mevcut veritabanındaki tablolardaki eksiklikler tamamlandı. Yönetici panelinde kullanılacak Şekil 4.27 UserLog tablosu ve Şekil 4.28 KULLANICI tablosu mevcut veritabanına (Şekil 4.29) eklendi.

	Column Name	Data Type
🔑	id	bigint
	name	nvarchar(50)
	role	nvarchar(5)
	expid	nvarchar(2)
	connDate	datetime
	connTime	nvarchar(15)
	disconnDate	datetime
	disconnTime	nvarchar(15)

(a)

	Column Name	Data Type
🔑	ID	int
	KULID	nvarchar(50)
	KULSIFRE	nvarchar(50)

(b)

Şekil 4.27 (a) Userlog tablosu (b) KULLANICI Tablosu

Table - dbo.cfg		
	Column Name	Data Type
	allowguest	bit
	pertime	int

Table - dbo.guest		
	Column Name	Data Type
	Name	varchar(10)
	number	int

Table - dbo.exp1		
	Column Name	Data Type
🔑	username	varchar(10)
	expid	int
	sno	int
	op	bit

Table - dbo.Prog		
	Column Name	Data Type
	SiraNo	int
	DeneyNo	int
	Aktif	bit

Table - dbo.exp2		
	Column Name	Data Type
	username	varchar(10)
	expid	int
	sno	int
	op	bit

Table - dbo.users		
	Column Name	Data Type
🔑	id	int
	name	varchar(10)
	pass	varchar(6)
	guest	bit
	op	bit
	connect	bit

Şekil 4.29 MicroLab veritabanı

C# programlama dili kullanılarak mevcut programa bağlantı işlemleri için eklentiler yapıldı. ASP.NET programı ile sistem yöneticisi ara yüzü yazılımı codebehind yazılım tekniği kullanılarak gerçekleştirildi.

5. SONUÇ

Sanal laboratuvarların kurum maliyetinin düşük olması mühendislik eğitimi uygulamalarını uzaktan eğitime geçmesi sürecini başlatmıştır. Gerekli arayüz programlarının, sanal laboratuvarların kullanılabilirliğini artırması bu talepleri karşılamış, günümüzde de bu laboratuvarların kurulma hızları artmıştır. MicroLab sanal laboratuvarların mantığı ile yapılan, yalnız kamera sistemi ile gerçek laboratuvarları kullanıcıların hizmetine sunmaktadır.

Yapılan yüksek lisans tezinde, uzaktan eğitim için internet üzerinden erişilebilen gerçek zamanlı ve interaktif mikro denetleyici laboratuvarı için sistem yöneticisi arayüzü programı yapılmıştır. C# programlama dilinde yapılması planlanan yönetici arayüzü programı internet erişimi ile her yerden açılabilme fonksiyonelliğini göstermemiştir. Bu isteğe cevap verebilmesi için SQL veritabanı dili kullanılarak ASP.NET programlama dili ile yazılarak yönetici arayüzü gerçekleştirildi. Yönetici arayüzü ile MicroLab'ın bir merkezden, herhangi bir programlama bilgisine ihtiyaç duyulmadan yönetilebilme eksikliği kapatıldı.

Yönetici arayüzü ile MicroLab da kullanıcıların sisteme giriş çıkışları kontrol edilmekte, sistemde ne kadar kaldığı tutulmakta, hangi deneyleri uyguladığı tespit edilmekte ve oluşturulacak yeni deneylerin sisteme eklentisi bu arayüz ile yapılarak programcıya ihtiyacı ortadan kaldırmaktadır. Gerçekleştirilen arayüz ile MicroLab'ın daha verimli ve kullanışlı olmasını sağlanmıştır.

Bu şekilde sanal laboratuvarların arayüz programları ile kullanılabilirliklerin artırılması sanal laboratuvarların yaygınlaşmasını ortaya çıkaracaktır. Bu sayede teknik elemanların işlerini internetten halletmeleri ile en ufak bir deneme yapmak için bile laboratuvar ortamına taşınmamaları onlara zaman kazandırarak yeni fikirler üretmelerini hızlandıracaktır.

Kullanıcılar hakkında tutulan bilgilerin çeşitliğinin artırılması ile kullanıcı düzenleme panelinde gereken eklentiler yapılarak bu panel daha da zengin hale getirilebilir.

6. KAYNAKLAR

Akın, E., Karaköse, M., 2003. Elektrik ve Bilgisayar Mühendisliği Eğitiminde Sanal Laboratuvarların Kullanımı, Elektrik, Elektronik, Bilgisayar Mühendislikleri Eğitimi 1. Ulusal Sempozyumu, Ankara.

Algan, S., 2003. C#. Pusula Yayıncılık, 781s., İstanbul

Archer, T., 2004. C#'ı Kavramak. Arkadaş yayınları_Microsoft 404s., Ankara.

Bouce Buckman, A., 1999. Based Introductory Electrical Engineering Laboratory Course.

Casini, M., Prattichizzo, D., Vicino, A., 2004. The Automatic Control Telelab. A Web-based technology for Distance Learning, IEEE Control Systems Magazine, 24(3),36-44.

Çankaya, M.N., 2002. ASP.NET. Seçkin yayıncılık, 238s., Ankara.

Çubukçu, F., 2001. SQLServer 7.0. Alfa Yayıncılık, 771s., İstanbul

Demirkol, Z., 2005. C# ile ASP.NET. Pusula Yayıncılık, 546s., İstanbul.

Ertuğrul, N., 2005. Toward Virtual Laboratories A survey of LabVIEW based Teaching/Learning Tools and future trends .

Groff, J.R., Weinberg, P.N., 2003. Herkes için SQL. Alfa Yayıncılık, 889s., İstanbul.

Jones, A.R., 2003. Herkes İçin ASP.NET ve C# Web Programcılığı. Alfa Yayıncılık, 831s., İstanbul.

Ko, C. C., Chen, B. M., Hu, S. Y., Ramakrishnan, V., Cheng, C. D., Zhuang,, Y., Chen, J., 2001a. A web-based virtual laboratory on a frequency modulation experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 31-3, 295-303.

Ko, C. C., Chen, B. M. , Chen, J., Zhuang, Y., Tan, K. C., 2001b. Development of a Web-based laboratory for control experiments on a coupled tank apparatus. IEEE Transactions on Education, 44-1, 76-86,

Ko, C. C., Chen, B. M., Chen, S. H., Ramakrishnan, V., Chen, R., Hu, S. Y., Zhuang, Y., 2000. A large scale web-based virtual oscilloscope laboratory experiment. IEE Engineering Science and Education Journal, 9-2, 69-76.

Kutlu, A., 2004. Microlab: A Web based Multi - user remote microcontroller laboratory for engineering education, Int. J. Engng. Ed., 20, 5, 879-885.

Kutlu, A., Ekiz, H., Powner, E.T., 1996. Wireless Control Area Network, Networking Aspects of Radio Communication Systems, IEE Colloquium, 3/1-3/4, London, UK.

Kutlu, A., Taşdelen, K., Küçüksille, E. U., 2005. 8051 Mikrodenetleyicisi İçin İnternet Tabanlı Laboratuar Tasarımı. 4 th International Advanced Technologies Symposium, Konya.

Lawrenz, W.,1995. World-wide Status of CAN- Present and Future, ICC'95 2. International CAN conference proceedings, 0-12 0-25.

Mendi, F., Karabıyık, Ö., Toktaş, İ., 2004. Tasarı Geometride Görünürlük Prensiplerinin Bilgisayar Destekli Öğretimi. Makine Teknolojileri Elektronik Dergisi, 1-3, 15-21.

Putzmeister, O.W., Rupp, K.D., 1994. Implementation of CAN system in truck-based Aircraft washing system”, 1. International CAN Conference, Germany.

Ryan, D., Ryan, T., 2002. ASP.NET: Your Visual Blueprint for Creating Web Applications on the .NET Framework. Hungry Minds, 318s., New York

Taşdelen, A., 2003. C# ile Veritabanı Programlama ve ADO.NET. Pusula Yayıncılık, 433s., İstanbul.

Taşdelen, K., 2004. Mühendislik Eğitimi İçin İnternete Dayalı, İnteraktif, Sanal Mikrodenetleyici Laboratuar. S.D.Ü. Fen Bilimleri Enstitüsü, Y.Lisans Tezi,83s, Isparta.

EKLER

EK – 1

Yönetici Paneli Program Kodları

Default.aspx.cs

```

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

    }

    protected void girisBtn_Click(object sender,
    EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data
        Source=BEGONVIL;Initial Catalog=microlab;Integrated
        Security=True");

        DataSet ds = new DataSet("KULLANICI");

        SqlDataAdapter adp;

        string strText = kad.Text.Replace("'", "*");
        string strText2 = pas.Text.Replace("'", "*");

        adp = new SqlDataAdapter("SELECT * FROM KULLANICI
        WHERE KULID = '" + strText + "' AND KULSIFRE = '" +
        strText2 + "'", con);
        adp.Fill(ds, "KULLANICI");

        DataTable myTbl = new DataTable();
        myTbl = ds.Tables[0];

        if (myTbl.Rows.Count != 0)
        {
            DataRow rw;
            rw = ds.Tables[0].Rows[0];
            Session["UserID"] = rw["ID"];
            Response.Redirect("Entry.aspx");
        }
        else
            Response.Redirect("default.aspx");
        con.Close();
    }
}

```

Exp_add.aspx.cs

```

public partial class exp_add : System.Web.UI.Page

```

```

{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        try
        {
            DataSet ds = new DataSet("DENEY");

            adp = new SqlDataAdapter("SELECT * FROM Prog
Where DeneyNo = '" + dNo.Text.ToString().Replace("'",
''') + "'", con);
            adp.Fill(ds, "DENEY");

            if (ds.Tables[0].Rows.Count > 0)
            {
                uyari.Text = "Bu numarayla başka deney
kaydedilmiş. Lütfen
                    değiştirip tekrar deneyiniz.";
                return;
            }
            else
            {
                string com = "INSERT INTO Prog
(SiraNo,DeneyNo, Aktif) VALUES ('" +
sNo.Text.ToString().Replace("'", ''') + "'," +
dNo.Text.ToString().Replace("'", ''') + "'," +
                    "'" + drmList.SelectedItem.Value
+ "')";

                try
                {
                    con.Open();
                    SqlCommand cmd = new SqlCommand(com,
con);

                    cmd.ExecuteNonQuery();

                    drmList.Enabled = false;
                    sNo.Enabled = false;
                    dNo.Enabled = false;
                    uyari.Text = "Deney bilgisi
kaydedilmiştir.";
                }
            }
        }
    }
}

```

```

        }
        catch
        {
            uyarı.Text = "Kaydedilemedi. Lütfen tekrar
deneyiniz.";
        }
    }
    catch
    {
        uyarı.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
    }
}
}

```

exp_delete.aspx.cs

```

public partial class exp_delete : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("DENEY");

        adp = new SqlDataAdapter("SELECT * FROM Prog
ORDER BY DeneyNo", con);
        adp.Fill(ds, "DENEY");

        denList.DataSource =
ds.Tables["DENEY"].DefaultView;
        denList.DataTextField = "DeneyNo";
        denList.DataValueField = "DeneyNo";

        if (!IsPostBack)
        {
            denList.DataBind();
        }
    }
    protected void denList_SelectedIndexChanged(object
sender, EventArgs e)
    {
        DataSet ds = new DataSet("DENEY");
    }
}

```

```

        adp = new SqlDataAdapter("SELECT * FROM Prog
Where DeneNo = '" + denList.SelectedItem.Text + "'",
con);
        adp.Fill(ds, "DENEY");

        sNo.Text =
ds.Tables[0].Rows[0]["SiraNo"].ToString();
        dNo.Text =
ds.Tables[0].Rows[0]["DeneNo"].ToString();
        if (Convert.ToBoolean(drmList.Items[0].Value) ==
Convert.ToBoolean(ds.Tables[0].Rows[0]["Aktif"].ToString(
)))
            drmList.SelectedIndex = 0;
        else
            drmList.SelectedIndex = 1;

        DataSet ds2 = new DataSet("LOG");

        string cm = "SELECT * FROM userLog Where expid =
'" + denList.SelectedItem.Value + "'";
        adp = new SqlDataAdapter(cm, con);
        adp.Fill(ds2, "LOG");

        if (ds2.Tables[0].Rows.Count > 0)
        {
            uyari.Text = "Bu deneye girmiş kullanıcılar
var. Silmeniz halinde bu bilgiler de kaybolacaktır.";
        }
    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        try
        {
            DataSet ds = new DataSet("DENEY");

            string cm = "SELECT * FROM userLog Where
expid = '" + denList.SelectedItem.Value + "' AND connDate
= getDate() AND " +
                "disconnDate IS NULL";
            adp = new SqlDataAdapter(cm, con);
            adp.Fill(ds, "DENEY");

            if (ds.Tables[0].Rows.Count > 0)
            {
                uyari.Text = "Bu deneyi silemezsiniz. Şu an
deneyi
                    kullanan kullanıcılar var";
                return;
            }
        }
    }
}

```

```

    }
    else
    {
        string com = "DELETE FROM Prog WHERE
DeneyNo = '" + denList.SelectedItem.Value + "'";
        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(com,
con);
            cmd.ExecuteNonQuery();

            sNo.Text = "";
            dNo.Text = "";
            uyarı.Text = "Deney bilgisi
silinmiştir.";
            denList.DataSource = null;

            DataSet ds2 = new DataSet("DENEY");

            adp = new SqlDataAdapter("SELECT *
FROM Prog ORDER BY SiraNo", con);
            adp.Fill(ds2, "DENEY");

            denList.DataSource =
ds2.Tables["DENEY"].DefaultView;
            denList.DataTextField = "DeneyNo";
            denList.DataValueField = "DeneyNo";

            denList.DataBind();
        }
        catch
        {
            uyarı.Text = "İşlem başarısız. Lütfen
tekrar deneyiniz.";
        }
    }
}
catch
{
    uyarı.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
}
}
}
}

```

exp_modify.aspx.cs

```
public partial class exp_modify : System.Web.UI.Page
```

```

{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("DENEY");

        adp = new SqlDataAdapter("SELECT * FROM Prog
ORDER BY DeneyNo", con);
        adp.Fill(ds, "DENEY");

        denList.DataSource =
ds.Tables["DENEY"].DefaultView;
        denList.DataTextField = "DeneyNo";
        denList.DataValueField = "DeneyNo";

        if (!IsPostBack)
        {
            denList.DataBind();
        }
    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        try
        {
            DataSet ds = new DataSet("DENEY");

            adp = new SqlDataAdapter("SELECT * FROM Prog
Where DeneyNo = '" + dNo.Text.ToString().Replace("'",
"''") + "'", con);
            adp.Fill(ds, "DENEY");

            if (ds.Tables[0].Rows.Count > 0)
            {
                uyari.Text = "Bu numarayla başka deney
kaydedilmiş. Lütfen değiştirip tekrar deneyiniz.";
                return;
            }
            else
            {
                string com = "UPDATE Prog SET SiraNo = '"
+ sNo.Text.ToString().Replace("'", "''") + "', DeneyNo =
'" + dNo.Text.ToString().Replace("'", "''") + "', Aktif =
'" + drmList.SelectedItem.Value + "' WHERE DeneyNo = '" +
denList.SelectedItem.Value + "'";

```

```

        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(com,
con); //Veritabanına kaydet
            cmd.ExecuteNonQuery();

            uyari.Text = "Deney bilgisi
düzenlenmiştir.";
            denList.DataSource = null;

            DataSet ds2 = new DataSet("DENEY");

            adp = new SqlDataAdapter("SELECT *
FROM Prog ORDER BY DeneyNo", con);
            adp.Fill(ds2, "DENEY");

            denList.DataSource =
ds2.Tables["DENEY"].DefaultView;
            denList.DataTextField = "DeneyNo";
            denList.DataValueField = "DeneyNo";

            denList.DataBind();
        }
        catch
        {
            uyari.Text = "İşlem başarısız. Lütfen
tekrar deneyiniz.";
        }
    }
    catch
    {
        uyari.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
    }
}
protected void denList_SelectedIndexChanged(object
sender, EventArgs e)
{
    DataSet ds = new DataSet("DENEY");

    adp = new SqlDataAdapter("SELECT * FROM Prog
Where DeneyNo = '" + denList.SelectedItem.Text + "'",
con);
    adp.Fill(ds, "DENEY");

    sNo.Text =
ds.Tables[0].Rows[0]["SiraNo"].ToString();
}

```

```

        dNo.Text =
ds.Tables[0].Rows[0]["DeneyNo"].ToString();
        if (Convert.ToBoolean(drmList.Items[0].Value) ==
Convert.ToBoolean(ds.Tables[0].Rows[0]["Aktif"].ToString(
)))
            drmList.SelectedIndex = 0;
        else
            drmList.SelectedIndex = 1;
    }
}

```

exp_time.aspx.cs

```

public partial class exp_time : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("EXP");

        adp = new SqlDataAdapter("SELECT * FROM cfg",
con);
        adp.Fill(ds, "EXP");

        if (ds.Tables[0].Rows.Count > 0)
        {
            dSr.Text =
ds.Tables[0].Rows[0]["pertime"].ToString();
            ek.Text = "V";
        }
    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        string com;
        if (ek.Text.Length > 0) //daha önce kayıt
eklenmiş, update yapılacak
        {
            com = "UPDATE cfg SET pertime = '" +
dSr.Text.ToString().Replace("'", "'') + "', allowguest =
'" + gOp.SelectedValue + "'";
            try
            {
                con.Open();

```

```

        SqlCommand cmd = new SqlCommand(com,
con);        //Veritabanına kaydet
            cmd.ExecuteNonQuery();

            uyarı.Text = "Program bilgisi
düzenlenmiştir.";
        }
        catch
        {
            uyarı.Text = "İşlem başarısız. Lütfen
tekrar deneyiniz.";
        }
    }
    else        //ilk kayıt
eklenecek
    {
        com = "INSERT INTO cfg (allowguest, pertime)
VALUES ('" + gOp.SelectedValue + "',' +
dSr.Text.ToString().Replace("'", "'') + "')";
        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(com, con);
            cmd.ExecuteNonQuery();

            uyarı.Text = "Program bilgisi
kaydedilmiştir.";
        }
        catch
        {
            uyarı.Text = "Kaydedilemedi. Lütfen tekrar
deneyiniz.";
        }
    }
}
}
}

```

exp_watch.aspx

```

<div align="center">&nbsp;  <object class=frame
id="Object1" visible=false width="180" height="180"

classid="CLSID:22d6f312-b0f6-11d0-94ab-0080c74c7e95"

codebase="http://activex.microsoft.com/activex/controls/m
player/en/nsmp2inf.cab#Version=5,1,52,701"

standby="Loading Microsoft Windows Media Player
components..."

```

```

type="application/x-oleobject">

<param name="FileName" value="http://195.214.174.30:80">

<embed type="application/x-mplayer2"

pluginspage="http://www.microsoft.com/Windows/MediaPlayer
/download/default.asp"

src="http://195.214.174.30:80" name="NSPlay"
showcontrols="1"

showdisplay="0" showstatusbar="0" width="200"
height="200">

</embed>

</object>

```

record_delete.aspx.cs

```

public partial class record_delete : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("KAYIT");

        adp = new SqlDataAdapter("SELECT DISTINCT name
FROM UserLog ORDER BY name", con);
        adp.Fill(ds, "KAYIT");

        kulList.DataSource =
ds.Tables["KAYIT"].DefaultView;
        kulList.DataTextField = "name";
        kulList.DataValueField = "name";

        if (!IsPostBack)
        {
            kulList.DataBind();
        }
    }
}

```

```

protected void kaydet_Click(object sender, EventArgs
e)
{
    string com = "DELETE FROM userLog WHERE name = '"
+ kullList.SelectedItem.Value + "'";
    try
    {
        con.Open();
        SqlCommand cmd = new SqlCommand(com, con);
//Veritabanına kaydet
        cmd.ExecuteNonQuery();

        uyarı.Text = "Kullanıcı kayıtları
silinmiştir.";
        kullList.DataSource = null;

        DataSet ds = new DataSet("KAYIT");

        adp = new SqlDataAdapter("SELECT DISTINCT
name FROM UserLog ORDER BY name", con);
        adp.Fill(ds, "KAYIT");

        kullList.DataSource =
ds.Tables["KAYIT"].DefaultView;
        kullList.DataTextField = "name";
        kullList.DataValueField = "name";

        kullList.DataBind();
    }
    catch
    {
        uyarı.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
    }
}
}

```

user_add.aspx.cs

```

public partial class user_add : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");

    protected void Page_Load(object sender, EventArgs e)
    {

```

```

    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        if (kulPass1.Text == kulPass2.Text)
//şifreler uyusuyor mu
        {
            string pass = kulPass1.Text.Replace("'",
"*"); //şifrede ' varsa değiştir
            string com = "INSERT INTO users (name, pass)
VALUES ('" + kulAd.Text.ToString().Replace("'", "'') +
"', " +
                "'" + pass + "')";
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand(com,
con); //Veritabanına kaydet
                cmd.ExecuteNonQuery();

                kulAd.Enabled = false;
                kulPass1.Enabled = false;
                kulPass2.Enabled = false;
                uyari.Text = "Kullanıcı bilgisi
kaydedilmiştir.";
            }
            catch
            {
                uyari.Text = "Kaydedilemedi. Lütfen
tekrar deneyiniz.";
            }
        }
        else
        {
            uyari.Text = "Şifreler uyuşmuyor. Lütfen
kontrol ediniz.";
        }
    }
}

```

user_delete.aspx.cs

```

public partial class user_delete : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

```

```

protected void Page_Load(object sender, EventArgs e)
{
    DataSet ds = new DataSet("KULLANICI");

    adp = new SqlDataAdapter("SELECT name, id FROM
users ORDER BY name", con);
    adp.Fill(ds, "KULLANICI");

    kulList.DataSource =
ds.Tables["KULLANICI"].DefaultView;
    kulList.DataTextField = "name";
    kulList.DataValueField = "id";

    if (!IsPostBack)
    {
        kulList.DataBind();
    }
}
protected void kulList_SelectedIndexChanged(object
sender, EventArgs e)
{
    kulAd.Text = kulList.SelectedItem.Text;
    kulPass1.Text = "*****";
    uyari.Text = "Bu kullanıcıya ait tüm veriler
silinecektir";
}
protected void kaydet_Click(object sender, EventArgs
e)
{
    try
    {
        DataSet ds2 = new DataSet("DENEY");

        string cm = "SELECT * FROM users Where id =
'" + kulList.SelectedItem.Value + "' AND connect = 1";
        adp = new SqlDataAdapter(cm, con);
        adp.Fill(ds2, "DENEY");

        if (ds2.Tables[0].Rows.Count > 0)
        {
            uyari.Text = "Bu kullanıcı şu anda sisteme
bağlı durumda. Lütfen daha sonra tekrar deneyiniz.";
            return;
        }
        else
        {
            string com = "DELETE FROM users WHERE id
= '" + kulList.SelectedItem.Value + "'";
            try

```

```

        {
            con.Open();
            SqlCommand cmd = new SqlCommand(com,
con);          //Veritabanına kaydet
                cmd.ExecuteNonQuery();

                kulAd.Text = "";
                kulPass1.Text = "";
                uyari.Text = "Kullanıcı bilgisi
silinmiştir.";

                kulList.DataSource = null;

                DataSet ds = new
DataSet("KULLANICI");

                adp = new SqlDataAdapter("SELECT *
FROM users ORDER BY name", con);
                adp.Fill(ds, "KULLANICI");

                kulList.DataSource =
ds.Tables["KULLANICI"].DefaultView;
                kulList.DataTextField = "name";
                kulList.DataValueField = "id";

                kulList.DataBind();
            }
            catch
            {
                uyari.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
            }
        }
    }
    catch
    {
        uyari.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
    }
}
}
}

```

user_motion.aspx.cs

```

public partial class users_motion : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp, adp2;

```

```

protected void Page_Load(object sender, EventArgs e)
{
    trh1.Attributes.Add("onkeypress", "return
CheckDate(event,'" + trh1.ClientID + "')");
    //Add Event OnBlur to be sure that Typed numbers
could be a Date 8 Characters and adds the /
    //to get a Date 01/01/2006
    trh1.Attributes.Add("onblur", "return
ChangeDate(event,'" + trh1.ClientID + "')");
    trh2.Attributes.Add("onkeypress", "return
CheckDate(event,'" + trh2.ClientID + "')");
    //Add Event OnBlur to be sure that Typed numbers
could be a Date 8 Characters and adds the /
    //to get a Date 01/01/2006
    trh2.Attributes.Add("onblur", "return
ChangeDate(event,'" + trh2.ClientID + "')");

    DataSet ds = new DataSet("DENEY");

    adp = new SqlDataAdapter("SELECT * FROM Prog
ORDER BY DeneNo", con);
    adp.Fill(ds, "DENEY");

    denList.DataSource =
ds.Tables["DENEY"].DefaultView;
    denList.DataTextField = "DeneNo";
    denList.DataValueField = "DeneNo";

    DataSet ds2 = new DataSet("KULLANICI");

    adp2 = new SqlDataAdapter("SELECT name, id FROM
users ORDER BY name", con);
    adp2.Fill(ds2, "KULLANICI");

    kulList.DataSource =
ds2.Tables["KULLANICI"].DefaultView;
    kulList.DataTextField = "name";
    kulList.DataValueField = "id";

    if (!IsPostBack)
    {
        Bind_DataGrid(0);
        kulList.DataBind();
        kulList.Items.Add("-----");
        kulList.SelectedIndex = kulList.Items.Count-
1;

        denList.DataBind();
        denList.Items.Add("-----");
    }
}

```

```

        denList.SelectedIndex = 2;
    }
}
protected void snList_PageIndexChanging(object
sender, GridViewPageEventArgs e)
{
    snList.PageIndex = e.NewPageIndex;
    if ((trh1.Text.Length > 0) ||
(aktList.SelectedIndex > 0) || (denList.SelectedIndex !=
2) || (kulList.SelectedValue != "-----"))
        Bind_DataGrid(1);
    else
        Bind_DataGrid(0);
}
public void Bind_DataGrid(int cont)
{
    if (cont == 0)
    {
        string com = "SELECT name, role, expid,
connDate, connTime FROM dbo.userLog " +
        "GROUP BY name, role, expid, connDate,
connTime";

        DataSet ds = new DataSet("ACT");

        adp = new SqlDataAdapter(com, con);
        adp.Fill(ds, "ACT");

        snList.DataSource = ds.Tables["ACT"];

        try
        {
            snList.DataBind();
        }
        catch
        {
            snList.PageIndex = 0;
            Bind_DataGrid(0);
        }
    }
    else
    {
        int i = 0;
        string com = "SELECT name, role, expid,
connDate, connTime FROM dbo.userLog WHERE ";
        if (trh1.Text.Length > 0)
        {
            if (trh2.Text.Length <= 0)

```

```

        trh2.Text =
DateTime.Today.ToShortDateString();
        com = com + "connDate >= '" + trh1.Text +
"' AND connDate <= '" + trh2.Text + "' ";
        i++;
    }
    if (aktList.SelectedIndex > 0)
    {
        if (i > 0)
            com = com + " AND ";
        if (aktList.SelectedIndex == 1)
            com = com + " disconnDate IS NOT NULL
";

        if (aktList.SelectedIndex == 2)
            com = com + " disconnDate IS NULL ";
        i++;
    }
    if (denList.SelectedIndex != 2)
    {
        if (i > 0)
            com = com + " AND ";
        com = com + "expid = '" +
denList.SelectedValue + "' ";
        i++;
    }
    if (kulList.SelectedValue.Length > 0)
    {
        if (i > 0)
            com = com + " AND ";
        com = com + "name = '" +
kulList.SelectedItem.Text + "' ";
    }
    com = com + "GROUP BY name, role, expid,
connDate, connTime";

    DataSet ds = new DataSet("ACT");

    adp = new SqlDataAdapter(com, con);
    adp.Fill(ds, "ACT");

    snList.DataSource = ds.Tables["ACT"];

    try
    {
        snList.DataBind();
    }
    catch
    {
        snList.PageIndex = 0;
    }

```

```

        Bind_DataGrid(1);
    }
}
protected void kaydet_Click(object sender, EventArgs
e)
{
    if ((trh1.Text.Length == 0) &&
(aktList.SelectedIndex == 0) && (denList.SelectedIndex ==
2) && (kulList.SelectedValue == "-----"))
    {
        uyari.Visible = true;
        uyari.Text = "Lütfem arama kriteri giriniz!";
        return;
    }
    else
    {
        uyari.Visible = false;
        Bind_DataGrid(1);
    }
}
}

```

user_active.aspx.cs

```

public partial class active_users : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Bind_DataGrid();
        }
    }
    protected void snList_PageIndexChanging(object
sender, GridViewPageEventArgs e)
    {
        snList.PageIndex = e.NewPageIndex;
        Bind_DataGrid();
    }
}

```

```

public void Bind_DataGrid()
{
    string com = "SELECT dbo.users.id,
dbo.users.name, dbo.users.pass, dbo.users.guest,
dbo.users.op, dbo.users.connect, AA.name AS Expr1, " +
        "AA.role, AA.expid, AA.connDate, AA.connTime
FROM        dbo.users INNER JOIN " +
        "(SELECT        name, role, expid, connDate,
connTime FROM        dbo.userLog " +
        "GROUP BY name, role, expid, connDate,
connTime) AS AA ON AA.name = dbo.users.name WHERE
(dbo.users.connect = 1)";

    DataSet ds = new DataSet("ACT");

    adp = new SqlDataAdapter(com, con);
    adp.Fill(ds, "ACT");

    snList.DataSource = ds.Tables["ACT"];

    try
    {
        snList.DataBind();
    }
    catch
    {
        snList.PageIndex = 0;
        Bind_DataGrid();
    }
}
}

```

user_op.aspx.cs

```

public partial class user_op : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("DENEY");

        adp = new SqlDataAdapter("SELECT 'Exp' +
CAST(DeneyNo AS varchar(10)) As Deney, DeneyNo FROM Prog
ORDER BY DeneyNo", con);
        adp.Fill(ds, "DENEY");
    }
}

```

```

        denList.DataSource =
ds.Tables["DENEY"].DefaultView;
        denList.DataTextField = "Deney";
        denList.DataValueField = "DeneyNo";

        if (!IsPostBack)
        {
            denList.DataBind();
        }
    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {
        DataSet ds2 = new DataSet("OP");

        adp = new SqlDataAdapter("SELECT * FROM exp" +
denList.SelectedValue + " WHERE username = '" +
kulList.SelectedValue + "' AND op = 1", con);
        adp.Fill(ds2, "OP");

        if (ds2.Tables[0].Rows.Count > 0)
        {
            uyari.Text = "Bu kullanıcı zaten operatör.";
        }
        else
        {
            string com = "UPDATE exp" +
denList.SelectedValue + " SET op = 1 WHERE username = '"
+ kulList.SelectedValue + "'";
            con.Open();
            SqlCommand cmd = new SqlCommand(com, con);
//Veritabanına kaydet
            cmd.ExecuteNonQuery();

            com = "UPDATE exp" + denList.SelectedValue +
" SET op = 1 WHERE op = 1 AND expid = '" +
denList.SelectedValue + "'";
            SqlCommand cmd1 = new SqlCommand(com, con);
//Veritabanına kaydet
            cmd1.ExecuteNonQuery();

            uyari.Text = "Operatör atanmıştır.";
        }
    }
    protected void denList_SelectedIndexChanged(object
sender, EventArgs e)

```

```

    {
        DataSet ds2 = new DataSet("KUL");

        adp = new SqlDataAdapter("SELECT DISTINCT
username FROM exp" + denList.SelectedValue + " WHERE
expid = '" + denList.SelectedValue + "' ORDER BY
username", con);
        adp.Fill(ds2, "KUL");

        kulList.DataSource =
ds2.Tables["KUL"].DefaultView;
        kulList.DataTextField = "username";
        kulList.DataValueField = "username";

        kulList.DataBind();
    }
}

```

user_modify.aspx.cs

```

public partial class user_modify : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {
        DataSet ds = new DataSet("KULLANICI");

        adp = new SqlDataAdapter("SELECT * FROM users
ORDER BY name", con);
        adp.Fill(ds, "KULLANICI");

        kulList.DataSource =
ds.Tables["KULLANICI"].DefaultView;
        kulList.DataTextField = "name";
        kulList.DataValueField = "id";

        if (!IsPostBack)
        {
            kulList.DataBind();
        }
    }
    protected void kaydet_Click(object sender, EventArgs
e)
    {

```

```

        if (kulPass1.Text == kulPass2.Text)
//şifreler uyusuyor mu
        {
            string pass = kulPass1.Text.Replace("'",
"*"); //şifrede ' varsa değiştir
            string com = "UPDATE users SET name = '" +
kulAd.Text.ToString().Replace("'", "'') + "', pass = '"
+ pass + "' WHERE id = '" + kulList.SelectedItem.Value +
"'";
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand(com,
con); //Veritabanına kaydet
                cmd.ExecuteNonQuery();

                kulAd.Text = "";
                kulPass1.Text = "";
                kulPass2.Text = "";
                uyari.Text = "Kullanıcı bilgisi
düzenlenmiştir.";
                kulList.DataSource = null;

                DataSet ds = new DataSet("KULLANICI");

                adp = new SqlDataAdapter("SELECT * FROM
users ORDER BY name", con);
                adp.Fill(ds, "KULLANICI");

                kulList.DataSource =
ds.Tables["KULLANICI"].DefaultView;
                kulList.DataTextField = "name";
                kulList.DataValueField = "id";

                kulList.DataBind();
            }
            catch
            {
                uyari.Text = "İşlem başarısız. Lütfen
tekrar deneyiniz.";
            }
        }
        else
        {
            uyari.Text = "Şifreler uyuşmuyor. Lütfen
kontrol ediniz.";
        }
    }
}

```

```

        protected void kulList_SelectedIndexChanged(object
sender, EventArgs e)
        {
            kulAd.Text = kulList.SelectedItem.Text;
        }
    }
}

```

change_pass.aspx.cs

```

public partial class change_pass : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");

    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void kaydet_Click(object sender, EventArgs
e)
    {
        if (eSfr.Text == ySfr.Text)
        {
            string pass = eSfr.Text.Replace("'", "*");
            string com = "UPDATE KULLANICI SET KULSIFRE = '" +
pass + "'";
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand(com,
con);

                cmd.ExecuteNonQuery();

                eSfr.Enabled = false;
                ySfr.Enabled = false;
                uyari.Text = "Kullanıcı bilgisi
düzenlenmiştir.";
            }
            catch
            {
                uyari.Text = "İşlem başarısız. Lütfen tekrar
deneyiniz.";
            }
        }
        else
        {

```

```

        uyari.Text = "Şifreler uyuşmuyor. Lütfen kontrol
ediniz.";
    }
}
}

```

change_urname.aspx.cs

```

public partial class urname_change : System.Web.UI.Page
{

    SqlConnection con = new SqlConnection("Data
Source=BEGONVIL;Initial Catalog=microlab;Integrated
Security=True");
    SqlDataAdapter adp;

    protected void Page_Load(object sender, EventArgs e)
    {

        DataSet ds = new DataSet("DENEY");

        adp = new SqlDataAdapter("SELECT * FROM
KULLANICI", con);
        adp.Fill(ds, "DENEY");

        eAd.Text =
        ds.Tables[0].Rows[0]["KULID"].ToString();

    }

    protected void kaydet_Click(object sender, EventArgs e)
    {
        string com = "UPDATE KULLANICI SET KULID = '" +
yAd.Text.ToString().Replace("'", "'') + "'";

        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(com, con);
            cmd.ExecuteNonQuery();

            uyari.Text = "Kullanıcı bilgisi
düzenlenmiştir.";

```

```
        eAd.Enabled = false;
        yAd.Enabled = false;

    }
    catch
    {

        uyari.Text = "İşlem başarısız. Lütfen tekrar
        deneyiniz.";

    }
}
```

ÖZGEÇMİŞ

Adı Soyadı : Hasibe Hande ÇETİN

Doğum Yeri : Isparta

Doğum Yılı : 1981

Medeni Hali : Evli

Eğitim Ve Akademik Durumu:

Lise : 1995 – 1999 Isparta Gürkan Süper Lisesi

Lisans : 1999 – 2003 Süleyman Demirel Üniversitesi Üniversitesi
Teknik Eğitim Fakültesi
Elektronik ve Bilgisayar Eğitimi Bölümü
Bilgisayar Sistemleri Öğretmenliği Programı

Yabancı Dil : İngilizce

İş Denevimi

2003 - 2004 Süleyman Demirel Üniversitesi
Keçiborlu Meslek Yüksekokulu – Öğretim Görevlisi
(Sözleşmeli)

2004 - Milli Eğitim Bakanlığı Denizli Çardak Çok Programlı Lisesi
Bilgisayar Öğretmeni