

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**GÖRME ENGELLİLER İÇİN ZEKİ BİR ÖĞRETİM
SİSTEMİ TASARIMI**

(Yüksek Lisans Tezi)

Tezi Hazırlayan: **Ozan ALPTEKİN**

İstanbul, 2011

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**GÖRME ENGELLİLER İÇİN ZEKİ BİR ÖĞRETİM
SİSTEMİ TASARIMI**

(Yüksek Lisans Tezi)

Tezi Hazırlayan:

Ozan ALPTEKİN

Öğrenci No:

070820001

Danışman:

Prof. Dr. Mehmet Yahya KARSLIGİL

İstanbul, 2011

YEMİN METNİ

Yüksek Lisans tezi olarak sunduđum ‘‘Görme engelliler için zeki bir öğretim sistemi tasarımı’’ başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 04/04/2011

Aday: Ozan ALPTEKİN

T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

GÖRME ENGELLİLER İÇİN ZEKİ BİR ÖĞRETİM TASARIMI

Tezi Hazırlayan: Ozan ALPTEKİN

Özet

Bilgi, kişisel ve toplumsal gelişim açısından günümüzün en önemli araçlarından biridir. Bilgi edinebilmek için her yaş grubundan insan, gerek örgün gerekse yaygın eğitim-öğretim faaliyetlerinin içerisinde bulunmakta ve kişisel gelişimlerini sağlamaktadır. Örgün eğitim faaliyetleri, dershaneler, mesleki kurslar, çevrimiçi dersler, uzaktan öğretimler ve bunun gibi çalışmalarla insanlar gereksinim sahibi oldukları bilgilere kavuşabilmektedir. Ülkemizde de örgün eğitim yaşının altına indirilmesi, bilgiye verilen önemi göstermektedir. Ancak her insan aynı şartlara sahip olamamakta, çeşitli engellerden dolayı bilgiye ulaşmakta zorluk yaşamaktadır. Özellikle görme engelliler, görsel materyalleri kullanamamakta, onlar için özel üretilen materyallerle bilgiye dokunarak veya duyarak sahip olmaktadır. İşte bu durumdan hareketle bu tez çalışması yapılmıştır.

Tezde internet tabanlı bir öğretim sistemi tasarlanmış, bu yazılımı görme engellerin ve görme zorluluğu çekenlerinde herhangi bir fazladan donanım ihtiyacı duymadan kullanmaları hedeflenmiştir. Sistem, görme engelliler için ara yüzü bir fare ile kontrol edilebilir ve ses ile yönlendirilebilir şekilde tasarlanmıştır. Görme zorluluğu çekenler için ise renk ve boyutlar değiştirilebilir hale getirilmiştir. Sistem, kullanıcılara bir test uygulayıp, sonuca göre öğrenme stilleri tespit edip, kullanıcıları uygun derslere yönlendirmektedir. Değerlendirme aşamasında, soru uzmanı modülü ile soru hazırlamakta, ön test, konu testi ve son test ile öğretimin her aşamasını kontrol etmektedir. Ayrıca sistem yapay zekâ teknikleri içermektedir.

Anahtar Kelimeler: Uzman Sistemler, Zeki Öğretim Sistemi, Yapay Zekâ, Bulanık Mantık, Uzaktan Eğitim

DESIGNING AN INTELLIGENT TEACHING SYSTEM FOR VISUALLY IMPAIRED PEOPLE

Abstract

Presented by: Ozan ALPTEKİN

Information is one of the most important tools of our age for personal improvement and development of the society. People from different age groups are attending either formal education or non-formal education possibilities for acquiring information and ensuring personal improvement. People can reach information that they need through formal education possibilities, private teaching institutions, vocational courses, online lessons, distant education, and similar possibilities. Today, reaching the information is quite easy, especially thanks to internet. However, not every person has the same opportunities, and some people face difficulties in reaching the information due to various disabilities. Especially, visually impaired people are not able to use visual materials, and they can only get information by touching and feeling with specially produced materials. And this is the starting point of my thesis study.

My thesis includes a web-based instruction system which is aimed to be used by visually impaired people, including dim-sighted people, without need to use any additional equipment. The system has been designed for visually impaired people with an interface which ensures control by using a mouse and giving voice commands. For dim-sighted people, on the other hand, colours and dimensions are designed in a changeable manner. The system subjects user to a test, and determines learning styles according to the result of test, and directs users to proper lessons. In evaluation stage, it prepares questions through question master module, and it controls each stage of the education through pre-test, subject test, and final test. Moreover, the system includes artificial intelligence techniques.

Keywords: Expert Systems, Intelligent Teaching System, Artificial Intelligent, Fuzzy Logic, Distance Education

İÇİNDEKİLER

TABLOLAR LİSTESİ.....	vi
ŞEKİLLER TABLOSU	vii
KISALTMALAR	ix
1. GİRİŞ	1
1.1 Giriş.....	1
2 GENEL BİLGİLER	3
2.1 Görme Engelliler Açısından Bilgisayar	3
2.2 Bilgisayar ile ilgili Genel Bilgiler	3
2.2.1 İnternet	3
2.2.2 İnternet'in Tarihçesi	4
2.2.3 İnternet'in Çalışma Prensibi.....	5
2.2.4 HTML	6
2.2.5 İstemci Taraflı Programlama.....	7
2.2.6 Sunucu Taraflı (Server-Side) Programlama.....	7
2.2.7 ASP .NET.....	8
2.2.8 .NET Framework.....	9
2.2.9 Common Language Runtime(CLR).....	9
2.2.10 C# Programlama Dili	9
2.2.11 SQL Server 2008.....	10
2.3 Öğrenme Öğretme Yaklaşımları	10
2.3.1 İnternet'in Eğitimdeki Rolü	10
2.3.2 Bilgisayar Destekli Öğretim(BDÖ).....	10
2.3.3 Uzaktan Eğitim.....	11
2.3.4 İnternet Tabanlı Öğretim.....	12
2.4 Zeki Öğretim Sistemleri	12
2.4.1 Zeki Öğretim Sisteminin Tanımı	13

2.4.2 Zeki Öğretim Sistemlerini Oluşturan Modüller	13
2.5 Yapay Zekâ	14
2.5.1 Yapay Sinir Ağları	15
2.5.2 Bulanık Mantık	16
2.5.3 Uzman Sistemler	25
3. GELİŞTİRİLEN SİSTEM.....	25
3.1 Sistem Analizi	25
3.1.1 Problemin Tanımı	25
3.1.2 Teknik İnceleme.....	25
3.2 Geliştirilen Sistem Hakkında Genel Bilgi.....	26
3.3 Veritabanı.....	28
3.3.1 E-R Diyagramı	29
3.4 Sistemin Elemanları	41
3.4.1 Yönetici	41
3.4.2 Öğretmen.....	41
3.4.3 Öğrenci.....	41
4. TASARLANAN UYGULAMANIN İNCELENMESİ	42
4.1 Yönetici Modülü	42
4.1.1 Öğretmen Hesapları	43
4.1.2 Öğrenci Onay İşlemleri.....	43
4.2 Öğretmen Modülü	49
4.2.1 Eğitim Programı	50
4.2.2 Kurs	59
4.2.3 Raporlar.....	60
4.3 Öğrenci Modülü	60
4.3.1 Eğitime Başla	61
4.3.2 Profil Ayarları	61

4.3.3 Öğrenme Stilleri	61
4.3.4 Ders Geçmişi	61
4.4 Öğrenci Modülünün İşleyişi.....	61
4.4.1 Görme Zorluğu Yaşayan Ve Görme Engeli Olmayan Kullanıcılar İçin İşleyiş	61
4.4.2 Görme Engelliler İçin Sistemin İşleyişi	62
4.5 Dersi Bitirme Mantığı	62
5. SONUÇ VE DEĞERLENDİRME	64
KAYNAKLAR	67
EK SİSTEMDEKİ BAZI ÖNEMLİ KODLAR	69

TABLÖLÄR LİSTESİ

Tablo.1. E-R Diyagramı.....	29
Tablo.2. Bulanık mantık karar tablosu.....	47

ŞEKİLLER TABLOSU

Şekil.1. İnternet'in çalışma prensibi	6
Şekil.2. Örnek bir yapay sinir ağı	16
Şekil.3. Klasik küme örneği.....	17
Şekil.4. Örnek bir bulanık mantık kümesi	19
Şekil.5. Üçgen üyelik fonksiyonu	20
Şekil.6. Yamuk üyelik fonksiyonu grafiği	21
Şekil.7. Gaussian üyelik fonksiyonu grafiği	22
Şekil.8. Bulanık sistemler akış diyagramı	23
Şekil.9. Veritabanı görünümü	28
Şekil.10. E-R Diyagramı.....	29
Şekil.11. aspnet_Applications tablosunun sütun ve veri tipleri.....	30
Şekil.12. aspnet_Membership tablosunun sütun ve veri tipleri	30
Şekil.13. aspnet_Paths tablosunun sütun ve veri tipleri.....	31
Şekil.14. aspnet_PersonalizationAllUsers tablosunun sütun ve veri tipleri	31
Şekil.15. aspnet_PersonalizationPerUsers tablosunun sütun ve veri tipleri	32
Şekil.16. aspnet_Profile tablosunun sütun ve veri tipleri	32
Şekil.17. aspnet_Roles tablosunun sütun ve veri tipleri	32
Şekil.18. aspnet_SchemaVersion tablosunun sütun ve veri tipleri	33
Şekil.19. aspnet_User tablosunun sütun ve veri tipleri.....	33
Şekil.20. aspnet_UsersInRoles tablosunun sütun ve veri tipleri.....	33
Şekil.21. aspnet_WebEvent_Events tablosunun sütun ve veri tipleri	34
Şekil.22. BRANS tablosunun sütun ve veri tipleri	34
Şekil.23. ENGELSİZ_SORU tablosunun sütun ve veri tipleri.....	35
Şekil.24. GORME_ENGELLI_SORU tablosunun sütun ve veri tipleri.....	35
Şekil.25. DERS tablosunun sütun ve veri tipleri	36
Şekil.26. KONU tablosunun sütun ve veri tipleri	36
Şekil.27. KONUYA_KURAL_EKLE tablosunun sütun ve veri tipleri	36
Şekil.28. KURALLAR tablosunun sütun ve veri tipleri.....	37
Şekil.29. KURS tablosunun sütun ve veri tipleri.....	37
Şekil.30. OGRENCI_OZELLIK tablosunun sütun ve veri tipleri.....	38
Şekil.31. OGRETMEN_OZELLIK tablosunun sütun ve veri tipleri	38
Şekil.32. SINIF tablosunun sütun ve veri tipleri.....	38
Şekil.33. BRANS tablosunun sütun ve veri tipleri	39

Şekil.34. SINIF_OGRETMEN tablosunun sütun ve veri tipleri	39
Şekil.35. SORU tablosunun sütun ve veri tipleri.....	40
Şekil.36. UNITE tablosunun sütun ve veri tipleri.....	40
Şekil.37. OGRENME_STILI tablosunun sütun ve veri tipleri	40
Şekil.38. Yönetici modülü anasayfası.....	42
Şekil.39. Yönetici modülünün bölümleri.....	42
Şekil.40. Öğrenci üyelik derecesi bulma grafiği.....	45
Şekil.41. Normalizasyon algoritması.....	45
Şekil.42. Bulanıklaştırma algoritması.....	46
Şekil.43. Öğretmen modülü anasayfa	49
Şekil.44. Öğretmen modülünün bölümleri	50
Şekil.45. 7,8 ve 9 basamaklı sayıları yazıya çeviren algoritma	52
Şekil.46. 7,8 ve 9 basamaklı sayıları yazıya çeviren kural için örnek soru	53
Şekil.47. 7,8 ve 9 basamaklı yazıyla yazılmış sayıların rakamla yazılmış sayıya çeviren kural için örnek soru.....	53
Şekil.48. Belirli sayıdaki rakamı en büyük en küçük sayıyı oluşturacak şekilde sıralayan kurala örnek bir soru.....	53
Şekil.49. Belirli sayıdaki rakamı en büyük en küçük sayıyı oluşturacak şekilde sıralama kuralı algoritması.....	55
Şekil.50. Sayıları bölüklerine ayırma kuralı algoritması	55
Şekil.51. Sayıları bölüklerine ayırma kuralı için örnek bir soru.....	56
Şekil.52. Örüntü oluşturma kuralı için örnek bir soru	56
Şekil.53. Örüntü oluşturma kuralı algoritması.....	57
Şekil.54. Ortalama buldurma kuralı algoritması.....	58
Şekil.55. Ortalama buldurma kuralı için örnek bir soru	58
Şekil.56. Toplama kuralı için örnek bir soru	59
Şekil.57. Öğrenci modülü anasayfası.....	60
Şekil.58. Öğrenci modülünün bölümleri.....	60
Şekil.59. Ders bitirme başarı hesabı	63
Şekil.60. Ders bitirme çıkışı.....	64

KISALTMALAR

BDÖ	: Bilgisayar Destekli Öğretim
ZÖS	: Zeki Öğretim Sistemi
HTML	: Hiperteks İşaretleme Dili
IIS	: Internet Information Service(İnternet Bilgi Sunucusu)
ASP	: Active Server Pages
PHP	:PHP: Hipertext Preprocessor
HTTP	: Hypertext Transfer Protocol
JSP	: Javaserver Pages
SQL	: Structured Query Language
AJAX	: Asenkron Java ve XML
MEB	: Milli Eğitim Bakanlığı
WWW	:World Wide Web
TCP/IP	: Transfer Control Protocol/Internet Protocol
ARPA	: Advanced Research Project Agency
MIT	: Massacusses Information Technology
BPS	:Bit per second
FTP	:File Transfer Protocol
NSFNET	:National Science Foundation Network
MSIL	:Microsoft Intermediate Language
ADO	:Activex Data Objects

1. GİRİŞ

1.1 Giriş

İnsanođlu doğumundan ölümüne kadar bir öğrenme süreci içerisinde bulunmaktadır. Bazen sadece meraktan öğrenirken, bazen ihtiyaçlarını karşılayabilmek için bazen de tamamen zorunluluklardan dolayı öğrenme gereksinimini gidermektedir. Daha dünyaya geldiđi ilk andan itibaren hayatta kalmayı öğrenmek zorundadır. İlk başlarda en temel ihtiyaçlarını gidermeyi öğrenirken, zamanla “anne”, “baba” gibi hitapları söylemeyi öğrenerek sosyalleşme çabası içersine girmektedir. Daha sonraki süreçte dünyaya olan merakı öğrenme güdülerini arttırmaktadır. Bu şekilde ömrünün sonuna kadar sürekli öğrenme öğretme sürecine etkin olarak katılmaktadır.

Örgün eğitim yaygınlaşmadan önce insanlar, yaşamı, herhangi bir disipline bađlı olmayan yollardan öğrenmek zorunda kalmışlardır. Bilgileri bazen anne, baba veya toplumdaki sosyal öğrenme yöntemiyle kazanırken, bazen sadece gözlemlerle, bazen de yaparak ve yaşarak öğrenmeye çalışmışlardır. Özellikle sosyal öğrenme, öğrenme sürecinde önemli bir yer tutmakta iken toplumdaki aktarılan bilgilerin belirli bir disipline göre aktarılması, doğruluğunun ve bilimselliğinin kesin olmaması, tabuların oluşmasına sebep olması, insanların yaşamı tanıması ve anlamasına yetmemesi gibi sebeplerden zamanla sadece sosyal öğrenme yetersiz kalmaya başlamıştır. İnsanların yaşamdan beklentilerinin artması, yaşam hakkında bildiklerinin yaşamı anlamaya yetmemesi, doğruluđu kanıtlanmış daha güvenilir bilgileri belirli bir disiplin içerisinde edinme çabası, beraberinde örgün eğitimi getirmiştir. Her ne kadar sosyal öğrenme hala çocukların hayatında önemli bir yer tutsa da örgün eğitim çocukları yaşama hazırlayan en önemli basamaktır. Bu basamak başlangıcı ve sonu çok iyi hesaplanması gereken bir süreçtir.

Örgün eğitime, zihinsel ve fiziksel gelişim göz önünde tutularak, olabildiğince erken başlamak çocuğun sonraki yıllardaki gelişimini oldukça etkilemektedir. Nitekim şu anda Türkiye’de okula başlama yaşı yedi iken, Milli Eğitim Bakanlığı (MEB) stratejik amaçları arasına, 2014 yılına kadar eğitime başlama yaşını altıya indirmeyi almıştır. (Milli Eğitim Bakanlığı[MEB],2009)

Teknolojik gelişmelerle paralel olarak bilgi teknolojilerinin eğitim-öğretimde kullanılmasında artmaktadır. Özellikle bilgisayar kimi zaman öğretilmek istenileni desteklemek amacıyla kullanılırken, kimi zaman da öğretme ortamını sunmaktadır. Bilgisayarların sunmuş olduğu çoklu ve zengin medya desteği, karşılıklı etkileşim fırsatları ve bireyselleştirilmiş öğretime olanak tanınması gibi özellikler sayesinde, eğitimde önemli bir araç haline almıştır. Aktarılabilecek bilginin bir sunum veya video ile desteklenmesi, herhangi bir konunun masaüstü programcılığı sayesinde sanal sınıflarda işlenmesi ya da internet ortamı kullanılarak uzaktan eğitim yapılması olağan haline gelmiştir. Hatta birçok üniversite, lisans ve yüksek lisans programlarını uzaktan eğitimi temel alacak hale getirmişleridir.

(Sönmez, 2007)' e göre uzaktan eğitim "Öğreten ve öğrenenin aynı ortamda bulunmadan, öğrenenin eğitim hizmetlerinden yararlanması için öğretim süreçlerini uzaktan ve bir merkezden düzenlenmesi, sunması, değerlendirip geliştirmesine" denilmektedir. Zamandan ve mekândan bağımsızlık, bireyselleştirilebilme şansı, çoklu medya desteği gibi sebeplerden günümüzde örgün ve yaygın eğitim kurumları tarafından çok kullanılan uzaktan eğitim, her geçen yıl daha da önem kazanmaktadır. Özellikle çalışma şartları, mesafe uzaklığı, engellilik ve bunun gibi sebeplerden sınıf ortamına katılamayacak olan öğrenciler için önemli bir şanstır.

Bilgi teknolojilerindeki gelişmelerle birlikte, günümüzde, oldukça iyi tasarlanmış materyaller bulunmaktadır. Ancak şu da bir gerçektir ki; yüz yüze öğretim yani öğrenci ile öğreticinin aynı ortamda bulunduğu ve karşılıklı etkileşime girdiği ortamın aynısını dijital ortamda yaratabilmek oldukça zordur. İşte bu noktadan hareketle bilgisayar destekli öğretimin(BDÖ) bir ileri aşaması olan zeki öğretim sistemi(ZÖS), öğretimi, klasik BDÖ' in tek düzelikinden kurtarıp, yüz yüze öğretime olabildiğince yaklaştırmayı hedeflemiştir. (Nwana, 1990) göre ZÖS tanımı neyi öğreteceğini, kime öğreteceğini ve nasıl öğreteceğini bilen, yapay zekâ ortak oluşumunda yer alan tekniklerden yararlanılarak hazırlanmış bilgisayar programları denilebilmektedir.

2 GENEL BİLGİLER

2.1 Görme Engelliler Açısından Bilgisayar

Bilgisayar, günümüzün en önemli bilgiye ulaşma, iletişim ve eğlence ortamlarından biri olmasına rağmen çoğunluğu görsel öğelerden oluştuğu için görme engelliler ve görme zorluğu yaşayanlar açısından zorlukları da barındırır. Görme engellilerin, monitörü ve bilgisayardaki işlemlerin yapmasında en büyük yardımcılardan biri olan fareyi hemen hemen hiç kullanmadıkları düşünülünce, bilgisayarı kontrol olanaklarının ne kadar kısıtlı olduğu anlaşılmaktadır. Bu sorunların çözümü bilgisayar bilimleri açısından çok zor olmamakla birlikte belirli bir çaba göstermek gerekmektedir. Kısmen gören görme engelliler için sorun kontrastları, resimleri ve yazı puntolarını uygun seçmekle çözülebilirken, hiç görmeyenler için sesli olan çözümler dışında başka seçenek bulunmamaktadır. Tezin konusunda yer alan görme engelliler kavramı bu iki durumu da kapsamaktadır.

Kısmen gören ve özellikle hiç görmeyenler için sesli çözümler en çok kullanılan çözümlerdir. Görme engellilerin en çok kullandıkları yardımcı programların başında bulunduğu sayfayı olduğu gibi okuyan programlar gelmektedir. Bu programlar herhangi bir kopyalama işlemine gerek duymadan, bilgisayarda o andan bulunan sayfayı baştan sona okumaktadır. Diğer bir seçenek kopyalanan metinleri kendi yazılımı üzerinde okuyan yardımcı programlardır. Öncelikle okunmak istenen yer kopyalanıp yardımcı programın içerisine yapıştırılmaktadır. Okuma işlemini yapıştırmadan sonra gerçekleşmektedir.

Diğer bir seçenek son yıllarda büyük gelişme yaşanan, ancak hala çok fazla yazılımlarda kullanılmayan, kullanıcıdan sesli komutlar almayı sağlayan yazılımlardır. Kullanıcıya, sesle yönlendirme şansı veren bu yazılımlar, muhtemelen ileriki yıllarda daha etkin olarak kullanılacaktır.

2.2 Bilgisayar ile ilgili Genel Bilgiler

2.2.1 İnternet

Bilgisayarların tüm dünyada yayılmaya başlaması ile birlikte daha önceleri kara kalem yapılan birçok çalışma artık dijital ortamda yapılmaya başlamıştır. Dijital ortamdaki dokümantasyon işlemlerinin, sonradan kullanabilme, düzeltilebilme ve çoğaltılabilme gibi birçok faydası bulunmaktadır. Özellikle belgelerin kolaylıkla

çoğaltılabilme özelliği bir dosyanın başkalarıyla da paylaşılabilme düşüncesini akıllara getirdi. Bu da internetinin doğması ve gelişmesinde önemli rol oynamıştır. Ancak günümüzde sadece dosya paylaşımı değil, aynı zamanda iletişim, eğlence, bilgi paylaşımı gibi birçok amaçla kullanılmaktadır.

Internet, dünya üzerinde birçok ağın birbiriyle ortak bir protokol çerçevesinde iletişim kurmasını ve birbirinin kaynaklarını paylaşmasını sağlayan kaynak kümesidir. TCP/IP denen bir protokol vasıtasıyla dünyadaki milyonlarca bilgisayar ve mobil cihazın birbirine bağlanmasıyla oluşmuştur. Internet herhangi bir kurum ya da kişiye ait değildir. Ancak bu her istenilenin sınırsız olarak yapılacağı anlamına gelmemektedir. Her ülkenin kendine özgü kural ve düzenlemeleri vardır.

2.2.2 İnternet'in Tarihçesi

Günümüzün önemli iletişim araçlarından biri olan internet, müzik, video, oyun gibi eğlence amaçlı, sosyal ağlar, anında mesajlaşma, elektronik posta, video konferans gibi iletişim amaçlı, elektronik ticaret siteleri, çevrimiçi bankacılık işlemleri gibi ticari amaçlı, kişisel web siteleri ve şirket web siteleri gibi reklam, tanıtım amaçlı ve uzaktan eğitim, internet tabanlı eğitim gibi eğitim-öğretim amaçlı olarak kullanılabilir.

Türkiye’de milyonlarca, dünyada ise milyarlarca kişinin etkin biçimde kullandığı internet MEB (2003)’e göre “İnternet, ilk olarak 1969 yılında Amerika Birleşik Devletleri’ inde, ABD Savunma Bakanlığına bağlı olarak çalışan Advanced Research Projects Agency (ARPA) adlı kurum tarafından, herhangi bir nükleer savaş durumunda askeri birimler arasında hiç kesintiye uğramayacak bir iletişim ağı kurmak amacıyla ARPANET adıyla bir bilgisayar ağı olarak kuruldu”. 1965 yılında ARPA, Massacusses Information Technology (MIT) ve System Development şirketindeki üç bilgisayar arasında hızı 1.200bps olan bir ağ kuruldu. Bu deney ARPANET denen ve internet’in atası olarak kabul edilen ağın ilk adımlarını oluşturmaktadır.

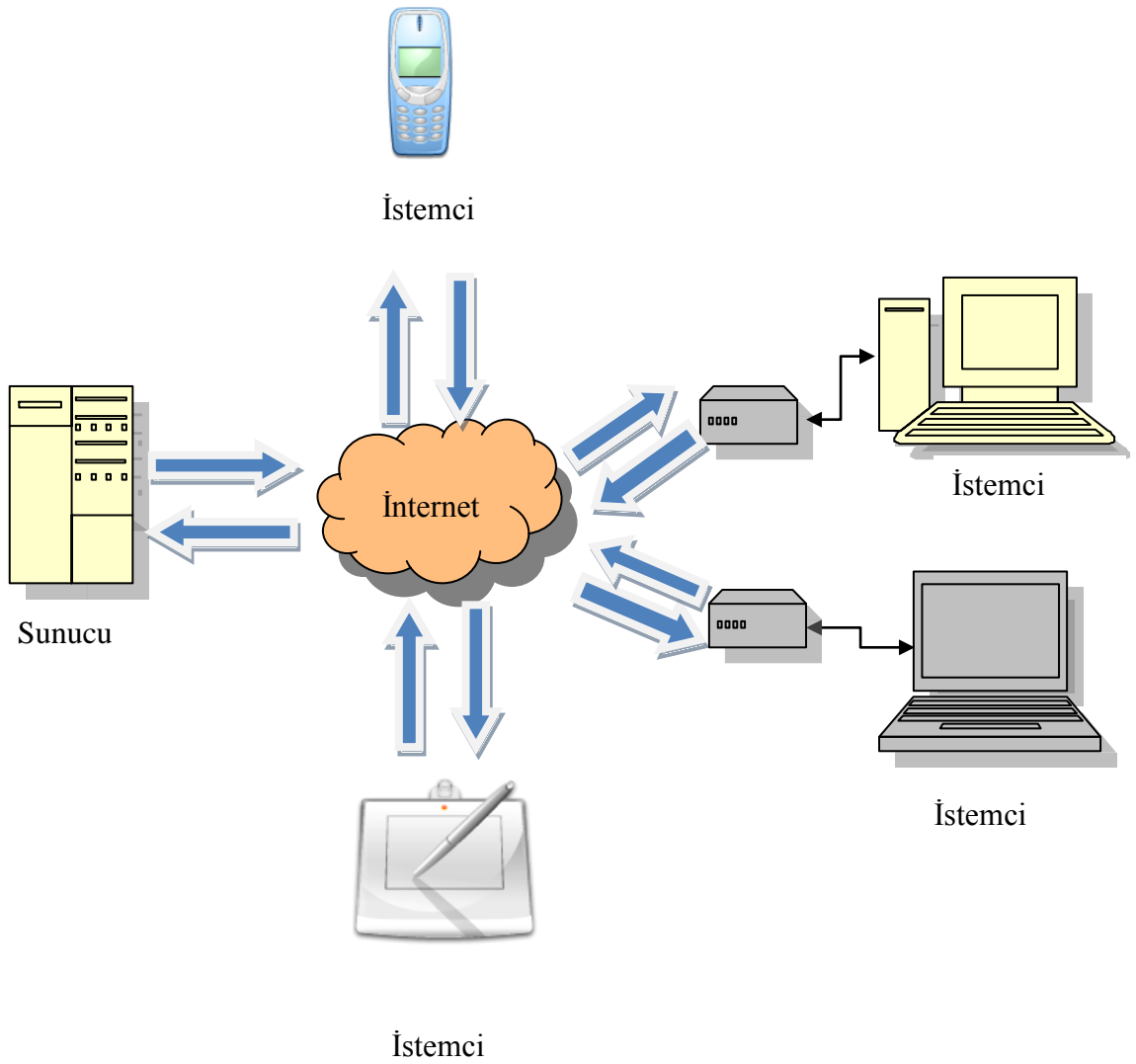
1970’li yıllarda ARPANET’ e birçok kurum katılmıştır. 1971 yılında ilk kez @ işaretli adresleme ortaya çıkmıştır. 1973 yılında File Transfer Protocol (FTP) protokolü geliştirilmiştir. 1974 yılında bugünkü İnternetin alt yapısını oluşturan TCP/IP protokolünün atası olan TCP protokolü geliştirilmiştir.

1980'li yıllarda çok önemli adım atılmıştır. NSFNET kurularak, ağ, araştırma kuruluşları ve üniversiteler açılmıştır. 1990'lı yılların başlarında ise bildiğimiz anlamda internet kurulmuştur ve ticari olarak kullanılmaya başlamıştır.

Ülkemiz ise internet ile ilk kez 1993 yılında tanışmıştır. İlk yıllarda Türkçe İnternet sitesi bulmak oldukça zor olmuştur. Ancak yıllar içerisinde bilgisayar donanım fiyatların düşmesi, bilgisayar ve bilgisayar kullanıcılarının sayısı artması ve internet'in cazibesini artmasıyla ülkemizde İnternet giderek yaygınlaşmıştır.

2.2.3 İnternet'in Çalışma Prensipleri

İnternet, sunucu istemci modeline göre çalışmaktadır. Bu model basit anlamda istekte bulunan bir cihaz (bilgisayar, cep telefonu, cep bilgisayarı gibi), ki buna istemci denir ve bu istemciye cevap veren bir sunucu arasında belli protokollere dayanan bilgi alışverişidir. Bu alışverişin gerçekleşebilmesi için mutlaka cihaz tarafından istekte bulunulmalıdır. Yani televizyonlarda ve radyolarda olduğu gibi yayınlama işlemi yoktur. İstemci ile sunucu bilgi alışverişini sağlayabilmek için önce istemci bir istekte bulunmalı, sunucunun onaylaması durumunda bir takım protokoller tamamlanmalıdır. Ancak bundan sonra sunucunun isteği cevaplanmaktadır.



Şekil.1. İnternet'in çalışma prensibi

Sunucu, istemciden gelen isteği değerlendirmekte ve sonra istemciye Hypertext Markup Language (HTML) kodları içeren bir dosya gönderir. İstemci bu dosyayı alınca internet tarayıcısı yardımıyla yorumlar ve doğru biçimde görüntüler.

2.2.4 HTML

Açılımı Hypertext Markup Language olan HTML, bir programlama dili olarak kabul edilmemekle birlikte, temel hedefi web sayfaları yaratmak olan bir işaretleme dilidir. C veya pascal gibi derlenip çalıştırılmamakta, sadece sunucudan gönderilen bir dosya, istemci tarafındaki internet tarayıcı programlar tarafından okunup, yorumlanmaktadır. Html, bilgisayar ve bilgisayar ağları arasında

dönüştürme problemlerini çözerek çeşitli dokümanları genel olarak erişilebilir hale getirmektedir.

Html dili ile hazırlanmış dosyalar sunucu ve istemci arasında Hypertext Transfer Protocol (http) protokolü ile taşınmaktadır. Bu protokol aslında bir sunucu protokolüdür.

2.2.5 İstemci Tarafı Programlama

Html işaretleme dili durağan bir dildir. İstemci ile etkileşime geçememektedir. İstemci tarafı programla, işte bunu ortadan kaldıran, istemci ile etkileşime geçebilen, sunucuya gitmeye gerek bırakmadan birçok işlemi istemci tarafında gerçekleştirmemizi sağlayan script'lerdir. Script, web sayfaları ile ilgilenen programlar anlamına gelmektedir. Bu programlar web sayfası daha esnek ve kullanışlı hale getirmektedir. Sunucu ile iletişime girmeden, kullanıcının istediği işlemleri istemci tarafında gerçekleştirebilmekte ve kullanıcıyı sunucuya gidip gelme süresince geçen bekleme zamanından kurtarabilmektedirler.

İstemci tarafında bu programları çalıştırma görevi internet tarayıcıya aittir. İnternet tarayıcı bu programları eğer güvenlik ayarları uygun ayarlanmışsa çalıştırmaktadır.

Bu programlar Javascript, VBScript ve JScript dilleriyle oluşturulur. Bunların içerisinde en çok kullanılanı Javascript dilidir. Javascript dili Java ile karıştırılmaması gerekmektedir. Java dili nesne tabanlı ve platform bağımsız bir programlama diliyken Javascript konusu tamamen web olan bir script dilidir. C ailesi sözdiziminden gelmektedir. HTML kodları içine gömülerek yazılır. Tarayıcı tarafından satır satır yorumlanarak çalıştırılmaktadır.

2.2.6 Sunucu Tarafı (Server-Side) Programlama

Sunucu tarafında yazılan ve çalıştırılan programlardır. İstemciden gelen istek doğrultusunda sunucu, kendi üzerinde yazılı olan programı çalıştırmakta ve sonuçları da bir html kodu olarak istemciye göndermektedir. İstemci bu programların içyapısını göremez ancak sunucunun verdiği yetki doğrultusunda istenilen parametrelerle bu programları çalıştırabilir ve sonuçlarını kendi bilgisayarında izleyebilmektedir. Her ne kadar istemci sunucu iletişim sırasında biraz vakit

kaybedilse de birçok geniş kapsamlı projelerin temelini sunucu taraflı programlama oluşturmaktadır.

En çok kullanılan sunucu taraflı programlama dilleri PHP: Hipertext Preprocessor (PHP), Active Server Page, (ASP) , ve Java Server Page (JSP)' dir. PHP açık kaynak kodlu bir web programlama dilidir. Apache ismi verilen web sunucusu üzerinde çalışmaktadır. JSP, java programlama dili ailesine bağlıdır ve java server üzerinden çalışmaktadır.

ASP, Microsoft ailesinin sunucu taraflı programlama için çözüm üreten yazılımıdır. Sözdizimi Basic dili temel alınarak oluşturulmuştur. Bir Microsoft işletim sistemi üzerine kurulan Internet Information Service (IIS) sunucusu üzerinden çalışmaktadır. ASP.net ise ASP devamı niteliğinde ancak ASP' den oldukça farklı bir web programlama dilidir.

2.2.7 ASP .NET

Asp.net, Microsoft ailesinin .net platformunun bir parçasıdır. Sunucu taraflı programlama dillerinden birdir ve sunucu taraflı programlamaya yeni bir yorum getirmektedir. Diğer sunucu taraflı programlama dillerine göre farklılığı olay yönlendirmeli modeli benimsemiş olmasıdır. Olay yönlendirme, herhangi bir rutinin veya programın çalıştırması için gerekli olan tetikleme işlemi şeklinde ifade edilmektedir. Bu işlem bir fare tıklaması, bir butona basılması, bir yazı yazılması gibi eylemler olabilmektedir (Demirkol, 2009). İstek-cevap modelinde en büyük problemlerden birisi bağlantısızlıktır yani sunucu isteği cevapladıktan sonra istemci tarafında olan olaylar hakkında bir bilgi sahibi olamamaktadır. İşte bu problemi aşabilmek için olay yönlendirme modeli Microsoft tarafından benimsenmiştir. Kısaca şu şekilde çalışmaktadır; istemciye gönderilen web sayfasının içerisine istemci tarafında çalışan programlarda gömülmektedir. Herhangi bir olay tetiklendiğinde istemci tarafında çalışan program sunucu ile iletişime geçerken yazılan kodlara göre işlemleri gerçekleştirir.

Asp.net programla dili ile yazılmış olan bir web sitesinin çalışabilmesi için sitenin yüklendiği sunucuda IIS sunucusunun bulunması gerekmektedir. IIS Microsoft ailesinin web sunucu programıdır. Ayrıca IIS dışında .NET Framework teknolojisinin kurulu olması gerekmektedir.

2.2.8 .NET Framework

Framework, .NET teknolojisi altında programlama yapılabilmesi için gerekli olan çatıyı sunan bir teknolojidir. Framework ayrıca kurulması gereken bir program olmakla birlikte, .NET programcılığı için mecbur olan bir programdır. .NET Framework yanında birçok yenilik getirmiştir. Getirdiği en büyük yeniliklerden biri VB, C#, C++, Jscript, Perl ve hatta Java ile bile program yazabilmeye olanak sağlamıştır. Böylece daha önceden programcılık deneyimi olanlar hangi dile ve sözdizimine alışkınsa .NET programcılığında da bu dili kullanmaya devam edebilmektedirler.

Ayrıca Framework ADO veritabanı yönetimini de geliştirip yeni bir yönetim anlayışı getirmiştir ki buna da ADO.NET adı verilmiştir.

2.2.9 Common Language Runtime(CLR)

CLR .NET teknolojisinin en yenilikçi özelliklerinden birini oluşturmaktadır. CLR .NET teknolojisi altında herhangi bir programlama dili ile yazılan kodları MSIL adı verilen orta seviye bir dile dönüştürmektedir. Geleneksel programlama dilleri yazılan kodları makine koduna çevirir ki buna derleme denilmektedir (Demirkol, 2009). Ancak CLR bu düzeni değiştirmektedir. VB.NET ve C# ile yazılan kodlar CLR tarafından MSIL denilen bir orta düzey dile çevrilmektedir. Böylece program kodları hangi dille yazılırsa yazılsın MSIL ile her platformda çalışabilir. Sonuç olarak CLR platformdan ve programlama dilinden bağımsız bir programlama ortamı sunmaktadır.

2.2.10 C# Programlama Dili

C#, C++ dilinin yeni bir sürümüdür. C++ bütün özelliklerini korumakla birlikte .NET platformu için bir çok yeni özellik eklenmiştir. Ancak sonuç olarak MSIL diline çevrildiği için performans açısından VB.NET bir farkı yoktur sadece C++ programlama diline alışmış programcıların .NET platformunda da alıştıkları sözdizimine devam etmeleri sağlanmıştır.

2.2.11 SQL Server 2008

Veriler bir programın temel yapı taşını oluşturmaktadır. Verilerin program tarafından doğru kaydedilmesi önemli olmakla beraber düzenli tutulması ve gerektiğinde hızlı ulaşılabilmesi de gerekmektedir.

Birbiriyle ilişkili verilerin sistematik olarak oluşturdukları yapılara veritabanı denmektedir (Kaya, Y., Ramazan, T.,2007). Günümüzde hemen hemen her uygulamada kullanılan veritabanları verilerimizi daha düzenli saklamamızı sağlar. Verilerdeki herhangi bir karışıklık veya hata çok önemli mağduriyetler doğurabilmektedir. E-ticaret siteleri, bankacılık işlemleri gibi hata kabul etmeyen işlemlerde en ufak hatalar hem programcıya, hem kullanıcıya hem de müşteriye telafisi olmayan sonuçlar doğurabilmektedir. Bundan dolayıdır ki veritabanını en iyi şekilde kurup, düzenleyip, yönetebilmek için sistemler oluşturulmuştur. Veritabanı tabloları yaratmak, bu tablolara girişi sağlamak, bu tabloları yönetmek gibi işlemleri yapan yazılımlara Veritabanı Yönetim Sistemleri denir. En çok tercih edilen veritabanı yönetim sistemleri Oracle İlişkisel Veritabanı Yönetimi, Microsoft Sql Server, Microsoft Access, DB2' dir (Kaya, Y., Ramazan T.,2007).

Bu sistemi geliştirirken Microsoft SQL Server kullanılmıştır. ASP.NET ile bağlantı kolaylığından dolayı bu veritabanı yönetim sistemi seçilmiştir. Microsoft Sql Server 2008 yazılımının ücretsiz sürümü olan Express Edition kullanılmıştır.

2.3 Öğrenme Öğretme Yaklaşımları

2.3.1 İnternet'in Eğitimdeki Rolü

İnternet'in getirmiş olduğu zamandan ve mekândan bağımsızlık eğitim öğretim alanında da birçok yeniliğin de ortaya çıkmasına yardımcı olmuştur. Önceleri öğretim denildiğinde öğrenci ve öğreticinin aynı ortamda olduğu yüz yüze iletişim akla gelirken, bugün uzaktan eğitim, internet tabanlı eğitim ve her ne kadar güvenilirliği tartışılrsa da forumlar öğretim işleminde önemli bir yer tutmaktadır.

2.3.2 Bilgisayar Destekli Öğretim(BDÖ)

Bilgisayarın ve internetin yaygınlaşması ile birlikte öğretmenin rolü de yavaş yavaş değişmeye başlamıştır. Önceleri bilginin tek kaynağı olan öğretmen, modern eğitim anlayışına göre bilgiye giden yolda öğrenciye yardımcı olan bir rehberdir.

Bilgiyi doğrudan vermektense bilginin nerde olduğunu ve nasıl ulaşılacağını öğretmek günümüz öğretmenin felsefesi haline gelmiştir.

Önceleri öğretmenlerin bir konuyu öğretmek için kullandıkları materyal sayısı çok sınırlı iken günümüzde bilgisayarların sunduğu çoklu medya desteği ile öğretim işlemi oldukça ilgi çekici hâle gelmiştir. Daha önceleri tek duyu organı ile derse katılan ve derste pasif olan öğrenciler, bilgisayarla desteklenen derslerde çoklu duyu organıyla katılarak öğrenmelerini kalıcı hale getirmişlerdir.

BDÖ' i şu şekilde tanımlayabiliriz; öğrencilerin programlı öğrenme materyalleri ile bilgisayar kullanarak etkileşimde bulunduğu; diğer bir deyişle, bilgisayar programları aracılığıyla öğrenmeyi gerçekleştirdiği, öğrenmelerini izleyip kendi kendilerini değerlendirebildiği bir öğretim biçimidir (Senemoğlu, 1997)

Bilgisayar destekli öğretim adından da anlaşılacağı gibi öğretim süresince öğretmenin yerine geçen değil, öğretmene destek veren materyallerdir. Konu öğretilirken destekleyici olduğu gibi, öğretim sonrası pekiştirici amaçlı veya değerlendirme amaçlı olarak da kullanılabilir.

BDÖ kullanabilmek için bazı şartların oluşması gerekmektedir. Bilgisayar, projeksiyon, mikrofon gibi donanımlar ve gerekli yazılımlar belli bir maliyet getirmektedir. Ayrıca BDÖ kullanacak öğretmen de belli bir eğitimden geçirilmek zorundadır. Öğrencilerin de sisteme alışması belirli bir süre almaktadır.

Sonuç olarak; zaman ve maliyet olarak belirli bir kaynak gerektirse de BDÖ birden çok duyuya hitap etmesi, kişiselleştirilebilir öğrenmeye uygun olması, öğrencileri derste aktif hale getirmesi, tekrar kullanılabilir ve güncellenebilir olması açısından günümüzde eğitimde önemli araçlarından biridir.

2.3.3 Uzaktan Eğitim

Uzaktan eğitim son yıllarda oldukça gözde olan bir eğitim öğretim yaklaşımıdır. Tanım olarak uzaktan eğitim “Öğreten ve öğrenin aynı ortamda bulunmadan öğrenenin eğitim hizmetinden yararlanması için öğretim süreçlerini uzaktan ve bir merkezden düzenlenmesi, sunması, değerlendirip geliştirmesine uzaktan eğitim denilebilir” (Sönmez, 2008).

Uzaktan eğitim her yaş grubu, her düzeyde öğrenci için kullanılabilir. Genellikle eğitimine devam edememiş veya zamandan ve mekândan bağımsız şekilde eğitim almak isteyen kişiler için ideal bir ortam oluşturmaktadır.

Uzaktan eğitimde öğrenci ve öğretici aynı ortamda bulunmazlar. Bazı uygulamalarda belirli aralıklarla bir araya gelebilmekle birlikte genellikle aynı ortamda bir araya gelmemektedir.

Öğrenci ile öğretici görüşmeleri gerektiğinde e-posta veya web sayfaları üzerinden görüşebilecekleri gibi anında mesajlaşma ve görüntülü görüşme gibi yöntemlerle de görüşebilmektedirler.

2.3.4 İnternet Tabanlı Öğretim

Öğrenciye, internete bağlı bir cihaz (bilgisayar, cep telefonu, el bilgisayarları vs.) üzerinden zamandan ve mekândan bağımsız olarak kendi hızı ve öğrenme kapasitesine göre öğrenme şansı veren öğretim programlarıdır. Bir nevi bireyselleştirilmiş öğretimdir. Program, öğrenciye isteği zaman isteği yerden öğretime katılabilme fırsatı vermektedir. Böylece sınıf ortamına katılmayan öğrenciler için önemli bir şans oluşturur. Ayrıca her öğrencinin öğrenme hızı aynı olmadığından yani bireysel farklarından dolayı öğrenci kendi hızında öğrenebilmektedir.

Öğretim sürecine herhangi bir öğretici katılabileceği gibi, öğretici programın kendisi de olabilmektedir. Sonuç olarak; bilgisayarın sunduğu geniş medya desteği ve internet hızlarının artmasıyla birlikte internet tabanlı öğretime destek giderek artmaktadır.

2.4 Zeki Öğretim Sistemleri

BDÖ sunduğu birçok avantaja rağmen bir takım dezavantajlara da sahiptir. Örneğin her öğrenciye aynı içeriği sunması bireysel farklılıklara sahip öğrenciler için doğru bir yaklaşım değildir. Öğrenme stillerine göre kimi öğrenciler görsel öğelerle daha iyi öğrenirken kimi öğrenciler duyarak, kimi öğrenciler ise dokunarak hareket ederek daha iyi öğrenebilmektedir. Bunu ayırt etmek çok zor olmamakla birlikte her biri için ayrı materyaller üretilmesi gerekmektedir. Kendi öğrenme stiline göre hazırlanmış materyalle çalışan öğrenci daha rahat öğrenebilmektedir.

Yine BDÖ' de değerlendirme de öğrencilere sabit dönütler verilmektedir. “Doğru” ve “Yanlış” gibi verilen dönütler sadece sorunun yanıtını verirken, sınıf ortamındaki gibi öğrenciye “Toplama işleminde elde hatası yaptınız. Lütfen elde işlemlerini tekrar ediniz” gibi verilebilecek öğütler, öğrenciyi hatasını düzeltme yolunda yardımcı olabilmektedir ancak BDÖ de sorulara sabit dönüt verildiğinden, hatalar görülememektedir. Ayrıca BDÖ, öğrencinin öğretim süresince olan gelişimini dikkat almamaktadır ancak öğrenci öğretim süresince değişimler geçirebilir. Sistemin bunu göz ardı etmesi öğretim sürecinde istenmeyen bir durumdur. Sınıf ortamında öğretmen öğrencilerinin gelişimini takip edip, buna göre stratejiler geliştirmektedir. Bunun BDÖ ortamında yapılamaması sürecin esnekliğine zarar vermektedir.

Bu düşüncelerden hareketle, BDÖ' i tek düzelikten kurtarıp, sisteme zeki etmenleri de dâhil ederek, sınıf ortamına olabildiğince yakın bir ortam elde etmek için zeki öğretim sistemleri geliştirilmektedir. Zeki öğretim sistemi, içerisine yapay zekâ yöntemleri dâhil edilmiş, BDÖ olarak düşünülebilmektedir.

2.4.1 Zeki Öğretim Sisteminin Tanımı

Zeki öğretim sistemi şu şekilde tanımlanabilir: “Zeki öğretim sistemleri, bilgisayar destekli olarak yapılan bir eğitim sürecini, öğrencinin daha iyi öğrenebilmesi, öğrencinin öğrenim tercihlerine göre eğitimine devam ettirebilmesi, sistemin öğrenciyi öğrenim süreci zarfında daha iyi izleyebilmesi ve gerektiğinde müdahale edebilmesi için yapay zekâ tekniklerini kullanarak zeki hale getirme sürecidir.” (Özbek, 2007).

Zeki öğretim sisteminin bir başka tanımı ise bilgisayar destekli öğretim sürecini gerçek sınıf ortamına olabildiğince yaklaştırabilmek için sisteme yapay zekâ yöntemlerini ekleyip, süreci olabildiğince esnek ve kişiselleştirilebilir hale getirmektir.

2.4.2 Zeki Öğretim Sistemlerini Oluşturan Modüller

Zeki öğretim modeli 4 modülden oluşmaktadır;

a-)Uzman Modülü

b-) Öğretmen Modülü

c-) Öğrenci Modülü

d-) Kullanıcı Ara yüzü Modülü

2.5 Yapay Zekâ

Yapay zekâ konusu son yüz yıl içerisinde insanların üzerine çok düşündükleri bir konudur. İnsan gibi düşünen robotlar, kendi kendine karar veren yazılımlar gibi birçok öge üzerine çok konuşulmuş ve tartışılmıştır. Yapay zekânın tanımına gelmeden, birbiriyle çok karıştırılan iki kelimenin tanımlanması çok yararlı olacaktır. Bu kelimeler akıl (us) ve zekâdır. Bu iki kelime günlük hayatta bazen birbirinin yerine kullanılabilir ancak anlam olarak birbirlerinden farklıdır.

Elmas (2007) akıllı şöyle tanımlamaktadır:

Düşünme, anlama, kavrama, idrak etme, karar verme ve önlem alma yetenekleridir.

Akılın bir başka tanımı ise muhakeme ve bilgi elde etme gücüdür (Elmas, 2007). Akıl genetik yollardan gelmekte olan hayatın sonuna kadar gelişebilen bir yetenektir. Akıl, makine, bilgisayar, yazılım veya başka bir yolla taklit edilemez (Elmas, 2007).

Zekâ ile akıl arasında bazı farklılıklar vardır. Binet'e göre zekâ; insanın sahip olduğu dikkat, bellek, yargılama, akıl yürütme, soyutlama vs. gibi yetiler topluluğudur (Nabiyev, 2005).

Zeka ölçülebilen bir yetenektir. "Zeka terimi yıllarca halk arasında ve literatürde farklı anlamlarda kullanıldıktan sonra, ancak yirminci yüzyılın başlarında bugünkü özel anlamı ile kullanılan ilk zeka testini hazırlayan Binet olmuştur" (Özgüven, 2003). Bu yüzden ancak Binetten sonra zekâ testleri bugünkü anlamına uygun sonuçlar vermiştir.

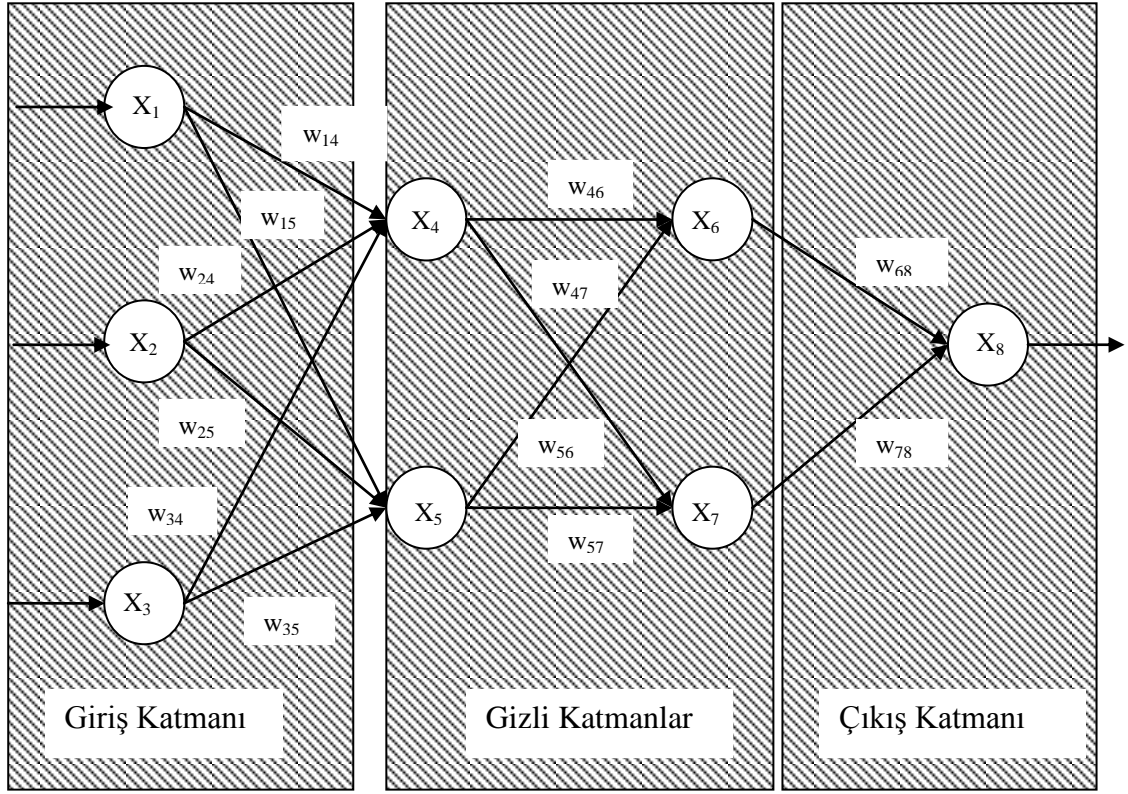
Bu tanımlardan hareketle akıl ile zekâ arasında yazılımcılar açısından en büyük fark şudur; akıllı herhangi bir donanım veya yazılım tarafından taklit edilemezken, zekâ taklit edilebilmektedir. İşte bu özelliğinden dolayı insanlar yıllardır donanım ve yazılımlarla insan zekâsını taklit etme çabası içerisinde olmuşlardır. İnsan zekâsının karmaşıklığı ve bulanıklığı bu işi oldukça zorlaştırmaktadır.

Yazılımlardan farklı olarak insan doğasında birçok durum ve karar keskin çizgilerle birbirinden ayrılmaktadır. Örnek vermek gerekirse geliştirilen ve daha sonra ayrıntılarıyla anlatılacak olan uygulamada, yazılım, öğrencinin öğrenme stilini görsel, işitsel ve kinetikselsel olarak ayırmaktadır. Sorulara verilen yanıtlara göre puanlar toplanır, en fazla hangi öğrenme stili fazla puan aldıysa o stile sahip olduğuna karar verilmektedir. Ancak gerçek dünyada bu tam olarak doğru değildir. Bir öğrenci birden fazla öğrenme stiline sahip olabilir ve diğerleri görmezden gelinemez. İşte bu tip problemlerin çözüm çalışmaları ile birlikte “Yapay Zekâ” tanımı ortaya çıkmıştır. “Yapay Zekâ” kelimesinin ilk olarak 1956 yılında John McCarty tarafından kullanılmıştır (Nabiyev, 2005).

Son yıllarda yapay zekâ konusu üzerine çok konuşulan, çalışılan ve makaleler yazılan bir konudur. Yapay zekâ insan zekâsının yazılım ve tümleşik yongalarla taklit edilmesidir (Elmas, 2007). Bilgisayarların çok karmaşık sayısal problemleri saliseler mertebesinde çözmesine karşın, anlama, birikim ve deneyim edinme gibi işlemlerde yetersiz olduğu bilinmektedir. İnsan zekâsı gibi muazzam bir mekanizmanın taklidinin oldukça zor olduğu hakikattir ancak Uzman Sistemler, Yapay Sinir Ağları, Bulanık Mantık, Genetik Algoritmalar gibi yapay zekâyâ ait kuramlarla çok büyük gelişmeler gösterilmektedir.

2.5.1 Yapay Sinir Ağları

İnsan beyni muazzam bir organdır. Nabiyev (2005)’e göre “Oldukça zor matematiksel hesaplama gerektiren uygulamalarda, bilgisayarların insanlardan her zaman üstün ve hata olasılığı çok düşük olmasına karşın, gerçek zamanlı doğal dil ve görüntü işleme gibi uygulamalarda, insanın bilgisayarlara karşı ezici bir üstünlüğü vardır. Bu üstünlüğü en büyük sebeplerinden biri sinir hücreleri (nöron) adı verilen yapılardır.”



Şekil.2. Örnek bir yapay sinir ağı

2.5.2 Bulanık Mantık

2.5.2.1 Klasik Küme Teorisi

Küme belirli ortak özellikleri olan elemanlar topluluğudur. Kümeler A,B,C ve D gibi harflerle ifade edilmektedir. Kümelere ait olan ve onu oluşturan nesnelere küme elemanı denir. Klasik küme teorisine göre küme elemanları ile kümeler arasındaki ilişki her zaman net ve kesin olmaktadır. Herhangi bir nesne herhangi bir kümenin ya elemanıdır ya da değildir. Herhangi bir belirsizlik söz konusu olmamaktadır. Kısacası klasik küme teorisine göre dereceli veya kısmi üyelik söz konusu değildir (Elmas,2007).

Klasik kümelerin gösterilmesi için değişik yollar bulunmaktadır. Bunlardan birisi liste yöntemidir. Liste yöntemi kümenin adı ile başlayıp, küme parantezleri

arasında yazılan elemanların birbirine eşitlenmesi ile oluşmaktadır. $k_1, k_2, k_3, \dots, k_n$ elemanlarından oluşan A kümesi şu şekilde gösterilir;

$$A = \{k_1, k_2, k_3, \dots, k_n\}$$

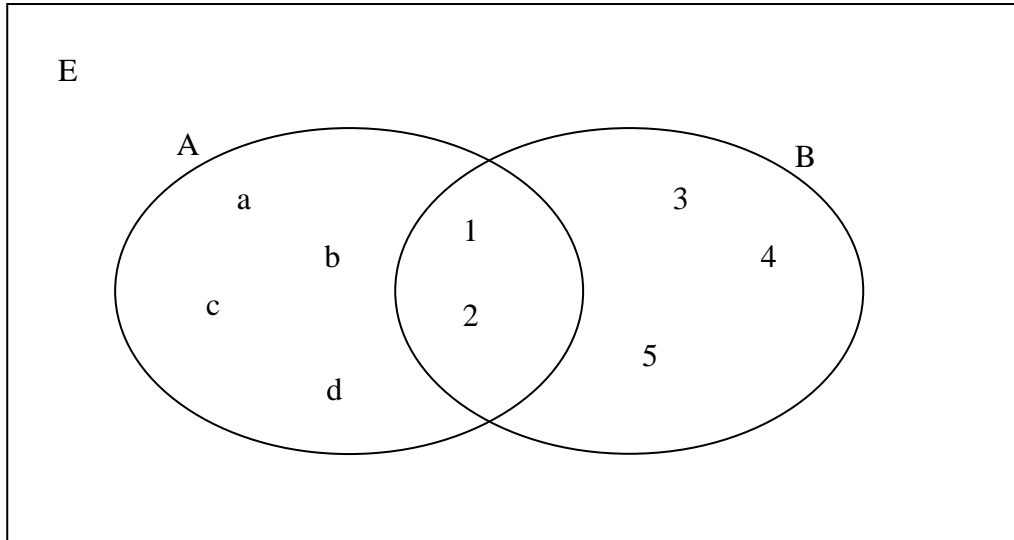
Diğer bir gösterim yolu ise kural yöntemi veya içlemsel yöntem denilen bir yöntemdir (Baykal, Beyan, 2004). Bu yöntemde liste yöntemine benzemekle birlikte elemanlar tek tek değil, belirli bir kurala göre gruplandırılmasıdır. B kümesini $m_1, m_2, m_3, \dots, m_n$ koşullarına göre bir küme oluşturuyorsa;

$$B = \{x \mid x \text{ } m_1, m_2, m_3, \dots, m_n \text{ koşullarını gerçekleştirenler}\}$$

Klasik kümelerin gösterim şekillerinden birisi de Venn şemasıdır. Venn şeması kümelerin geometrik şekillerle gösterilmesine verilen addır. Kümenin venn şemasını ile gösterilmesinde dikkat edilecek unsurlardan bir tanesi kümeyi oluşturan geometrik şekillerin kapalı olmasıdır. Aşağıda örnek bir Venn şeması gösterilmektedir.

$$A = \{a, b, c, d, 1, 2\}$$

$$B = \{x \mid x \in Z \text{ ve } 0 < x < 6\}$$



Şekil.3. Klasik küme örneği

Şekildeki A ve B harfleri ile gösterilen yuvarlaklar kümeleri temsil etmektedir. E harfi ile gösterilen dikdörtgen kümeye ise “Evrensel Küme” denilmektedir. Evren küme, olayın veya olaylar dizisinin olabilecek tüm sonuçlarının topladığı kümeye verilen isimdir (Baykal, Beyan, 2004).

Bir eleman bir küme için ya elemandır ya değildir. Buna karar veren fonksiyona üyelik fonksiyonu denilmektedir. Üyelik fonksiyonun sonucu elemanın kümeye ait olup olmamasına göre ya 0 değeri alır ya da 1 değeri alır. Bir nesnenin kümeye üyelik derecesini $\mu_A(x)$ olarak tanımlanabilmektedir.

Küme işlemlerinin bazı önemli konuları şunlardır;

a-)Kesişim: İki veya daha fazla kümenin mantıksal “ve” işlemine tabi tutulmasıdır. Şu şekilde tanımlanabilmektedir;

$$A \cap B = \{ x \mid x \in A \cap x \in B \}$$

b-)Birleşim: İki veya daha fazla kümenin mantıksal “veya” işlemine tabi tutulmasıdır. Şu şekilde tanımlanabilmektedir;

$$A \cup B = \{ x \mid x \in A \cup x \in B \}$$

c-)Fark İşlemi: İki kümenin karşılaştırmasında bir kümede olup diğer kümede olmayan elemanlardır.

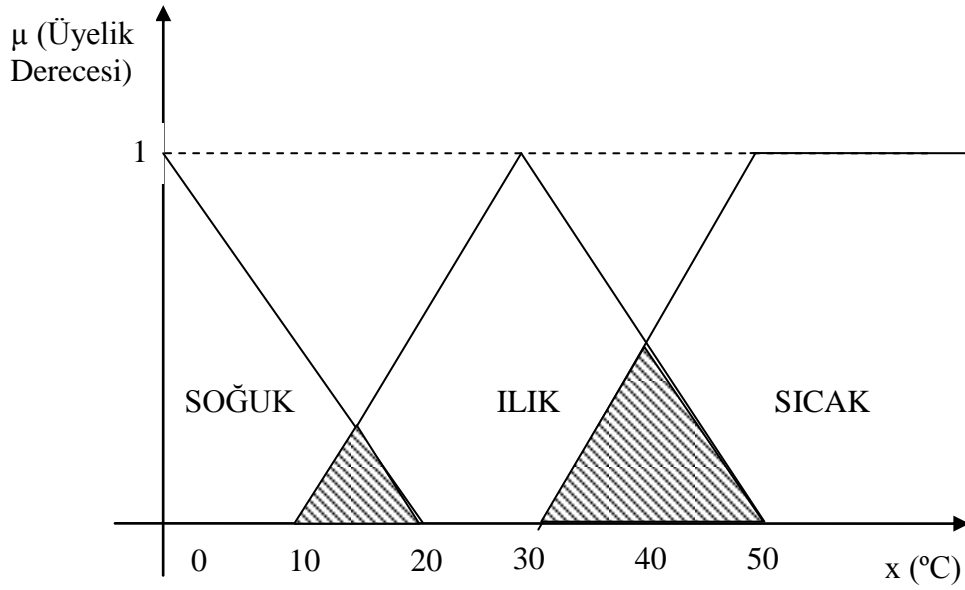
$$C \setminus D = \{ x \mid x \in C \text{ } x \notin D \}$$

d-)Tümleme: Herhangi bir F kümesinde olmayıp F kümesini kapsayan E evrensel kümesinde olan elemanların oluşturduğu kümeye verilen addır.

2.5.2.2 Bulanık Kümeler

Klasik küme teorisi bir küme ile bir nesnenin ilişkisini üyesi olma veya olmama diğer bir deyişle 0 ve 1 olarak tanımlamaktadır. Ancak gerçek dünyada klasik kümelerde olduğu kadar net ayrımlar bulunmamaktadır. İşte bu noktadan hareketle 1960’lı yılların ortalarında Lotfi A. Zadeh karmaşık sistemlerin tasarımında klasik küme teorisinin yetersiz kalmasında dolayı, bulanık küme kavramını ortaya atmıştır (Baykal, Beyan, 2004).

Örnek vermek gerekirse“Sıcak” ve “Soğuk” kelimelerini ele alalım. Belirli sıcaklıkta bir su, içmek için ılık, yüzmek için sıcak veya yemek yapmak için soğuk olabilmektedir. Yine aynı şekilde belirli bir sıcaklıktaki hava, yüzmek için soğuk, spor yapmak için sıcak, seyahat için ılık ya da gömlek giymek için serin olabilmektedir. Bu sıcaklık değerini klasik kümeleme yöntemi ile kümelemek istersek bu değer ya sıcak kümesine ya da soğuk kümesine girmektedir. Ancak gerçek hayatta ara değerlerde vardır ve bunlar göz ardı edilememektedir. Bu düşünceden hareketle bulanık mantık, belli bir değer için keskin bir ayrıma gitmeden, değerini, her kümeyle belli oranda yaklaşabileceğini kabullenen bir kuramdır.



Şekil.4. Örnek bir bulanık mantık kümesi

Bulanık mantık kümelerinde her üye, verilen her küme için bir üyelik derecesine sahip olmaktadır. O derecelere göre her kümeyle ne kadar ait olduğu hesaplanabilmektedir. Örnek vermek gerekirse bulanık mantık kümelemesine göre 30°'lik bir su belli bir oranda sıcak belli bir oranda ise soğuktur.

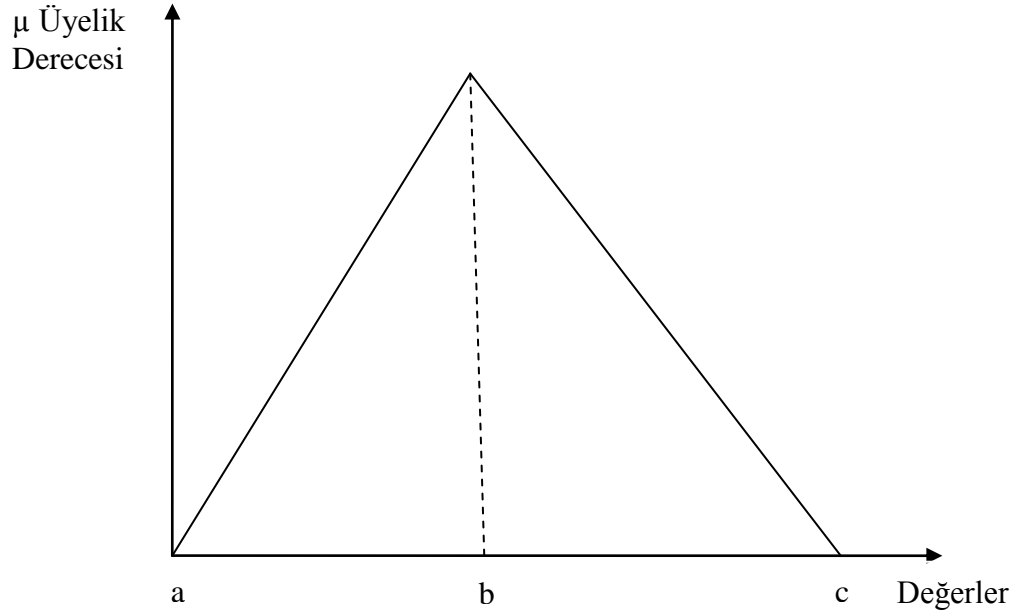
Bulanık küme teorisine göre bir nesne ile küme arasında sadece 0 ve 1 ilişkisi bulunmaktadır. Küme ile eleman arasındaki ilişkiye göre 0 ile 1 arasında herhangi bir değer alabilmektedir. Bir elemanın aldığı bu değer küme üyelik derecesi denilmektedir. 0 değeri kümeyle ait olmamayı 1 ise kümeyle kesin olarak ait olmayı

göstermektedir. Üyelik derecesi 0'dan 1'e doğru yükseldikçe eleman kümeye olan bağlantısı daha fazla artmaktadır. Bir üyelik derecesi fonksiyonu şu şekildedir;

$$\mu_A : E \longrightarrow [0,1]$$

2.5.2.3 Üyelik Fonksiyonları

a-) Üçgen Üyelik Fonksiyonu

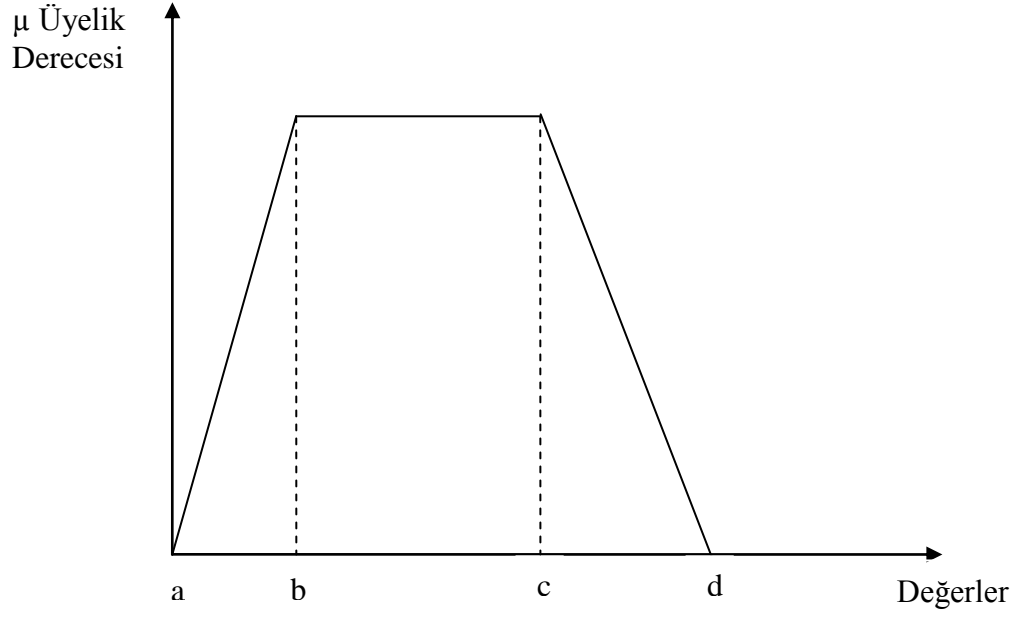


Şekil.5. Üçgen üyelik fonksiyonu

$$\mu(x,a,b,c)= \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x > c \text{ veya } x < a \end{cases}$$

Bir üçgen üyelik fonksiyonunda x değerine sahip bir elemanın üyelik derecesi yukarıdaki formülle bulunabilmektedir.

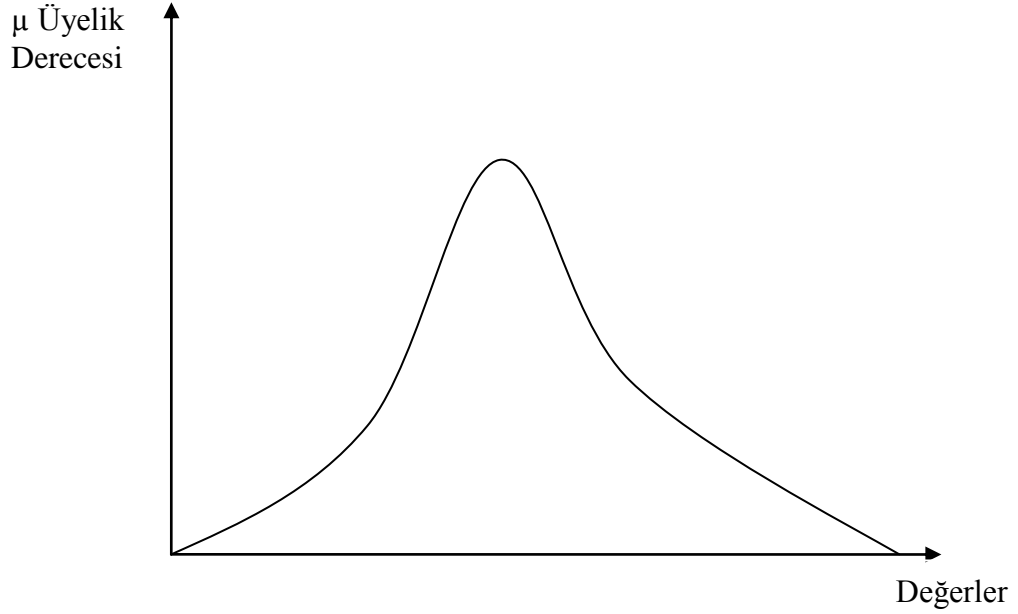
b-)Yamuk Üyelik Fonksiyonu



Şekil.6. Yamuk üyelik fonksiyonu grafiği

$$\mu(x,a,b,c,d)= \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x > d \text{ veya } x < a \end{cases}$$

c-)Gaussian Üyelik Fonksiyonu



Şekil.7. Gaussian üyelik fonksiyonu grafiği

$$\mu A(x; m, \sigma) = \frac{\exp(-(x - m)^2)}{2\sigma^2}$$

m: fonksiyon merkezi

σ : genişlik

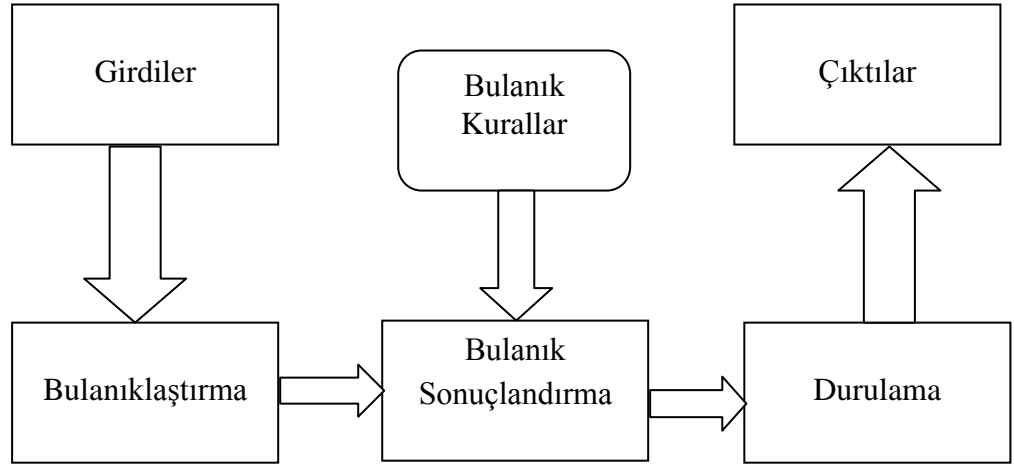
2.5.2.4 Bulanık Sistemler

Bulanık sistemler insanın düşünme tarzını temel alan sistemlerdir. Bulanık küme teorisini kullanarak oluşturulan bu sistemler temel anlamda üç aşamada oluşturulmaktadır;

a-)Bulanıklaştırma

b-)Bulanık sonuçlandırma

c-)Durulaştırma (Nabiyev, 2005)



Şekil.8. Bulanık sistemler akış diyagramı

a-)Bulanıklaştırma

Bulanıklaştırma işlemi girdilerin bulanık küme elemanı haline dönüştürülmesi işlemidir. Girdilerin bulanık küme elemanına çevirebilmek için uygun bir fonksiyon seçilmelidir. Bu fonksiyona göre veriler normalleştirme işlemine tabi tutulmaktadır. Normalleştirme sonucunda fonksiyona uygun olarak bulanık kümelere göre küme üyelik değeri hesaplanmaktadır.

b-)Kurallar

Kurallar bulanık sonuçlandırmanın en temel öğelerinden biridir. Bulanık kurallar, bulanık olarak gelen sonuçların karşılaştırıldığı kurallar dizisidir. Sözel olarak şu şekilde ifade edilmektedir;

“Eğer x doğru ise z’ de doğrudur”

Bulanık kurallar tam, tutarlı ve kesin olması gerekmektedir(Baykal, Beyan,2004). Her türlü durumun mutlaka düşünüldüğü bu kurallar dizisi, çelişkili sonuçlar içermemektedir. Bulanık kurallar “ve”, “veya” ve “değil” gibi mantıksal işlemcilerle desteklenmektedir. Örneğin;

“Eğer x a ve y b doğru ise, o halde z c’ dir”

c-)Bulanık Sonuçlandırma

Bulanık kurallara uygunluğunun incelendiği ve sonuçlandırıldığı bölümdür. Bulanık küme değeri olarak gelen değerleri gerekli kuralları uygulayarak duruluma işlemine uygun hale getirmektedir.

d-)Durulama

Durulama işlemi bulanıklaştırmanın tersine bulanık değerlerin kesin değerlere çevrilme işlemidir. Birçok uygulama kesin değerlere ihtiyaç duymaktadır. Bulanık olan verileri kesin değerlere çevirme işlemine durulama işlemi denir. Birçok yöntem olmasına rağmen en çok kullanılan yöntemler şunlardır; Maksimum üyelik ilkesi (max-membership principle), en büyüklerin ortası yöntemi (Mean max membership), ağırlık merkezi yöntemi (COA-center of area), ağırlık ortalama yöntemi (Weighted average method), açılı ortay yöntemidir.

Maksimum Üyelik İlkesi:

Üyelik derecesi en büyük olan üye kesin değer olarak kabul edilir.

En Büyüklerin Ortası Yöntemi:

Üyelik derecesi en büyük olan üyelerin ortalamasının alınarak bilgilerin kesinleştirilmesine dayanan durulama ilkesidir.

$$z_0 = \sum_{i=1}^k (z_i/k)$$

Ağırlık Merkezi (Sentroid) Yöntemi:

Bulanık A kümesinin olasılık dağılımının çekim noktasını üretir (Baykal, Beyan, 2004).

$$z_0 = \frac{\sum_{i=1}^k \mu_A(z_i) \cdot z_i}{\sum_{i=1}^k \mu_c(z_i)}$$

n: Çıktının niceleme sayısı

z:A bulanık kümesinin çıkış boyutu

Ortalama Merkezi Yöntemi

$M = \{i | \mu_i = EB\{\mu_1, \dots, \mu_n\}\}$, $x = x_h$ olmak üzere;

$$\sum_{i=EK_j(j \in M)}^h \mu(x_i) = \sum_{i=h}^{\mu} \mu(x_i)$$

2.5.3 Uzman Sistemler

Nabiyev (2004)'e göre "Uzman Sistemler(US), belirli konuda uzman olan bir ve birçok insanın yapabildiği muhakeme ve karar verme işlemlerini modelleyen bir yazılım sistemidir". Kısaca belirli bir ve birden fazla konuda uzmanlaşmış yazılımlardır. 1960'lı yıllarda J. Lederberg tarafından ortaya atılmıştır. Belirli becerilere sahip olan bu yazılımlar, günümüzde tıp alanından askeri alana kadar birçok alanda kullanılmaktadırlar.

3. GELİŞTİRİLEN SİSTEM

3.1 Sistem Analizi

3.1.1 Problemin Tanımı

Bilgisayar destekli öğretimin basmakalıplığından kurtulup, yapay zekâ yöntem ve tekniklerini içeren bir öğretim sistemi tasarlayıp, bu sistemi hiç görmeyenlerin ve az görenlerin de kullanabileceği internet tabanlı bir uygulamayla sunmaktır.

3.1.2 Teknik İnceleme

Sistemin olabildiğince fazla kullanıcıya ulaşabilmesi için masaüstü ortamından çok internet ortamından yararlanılmalıdır. Bu yüzden yaygın olan iki platformdan birisinin seçilmesi gereklidir. Bu platformlardan birisi JAVA platformu diğeri ise Microsoft .NET platformudur. Java Platformu JVM teknolojisi sayesinde herhangi bir platformdan bağımsız, programlama dili olarak, açık kodlu, nesneye yönelik, güvenilir ve sağlam, yazım ortamı olarak ücretsiz araçlar sunan bir teknolojidir. Microsoft .NET ortamı ise Visual Basic, C, Java gibi bir çok dili bünyesinde çalıştırabilen tamamen internet ortamına uyumlu bir ortamdır. Ayrıca

AJAX ve Silverlight gibi yeni teknolojileri de barındırmaktadır. 2008 sürümüyle birlikte Express Edition isimiyle ücretsiz yazılım araçları da sunmaktadır.

İki platformunda birbirlerine göre avantajları ve dezavantajları vardır. Bunlar göz önüne alınarak proje için platform olarak .NET platformu, uygulamaya geliştirme yazılımı olarak Microsoft Visual Web Developer, programlama dili olarak C#, veritabanı yönetim sistemi olarak SQL Server seçilmiştir.

3.2 Geliştirilen Sistem Hakkında Genel Bilgi

Bu sistem görme engelliler, az görenler ve engeli olmayan kullanıcılar için tasarlanmıştır. Zeki öğretim sistemi prensiplerini örnek alarak içerisine zeki etmenler eklenerek tasarlanan sistem; “kime”, “neyi”, “nasıl” öğreteceğini bilen bir sistem olarak tasarlanmıştır.

“Kime” ile kastedilen, programın, görme engelli, az gören veya engelsiz olma durumuna göre kullanıcıyı tanımasıdır. Kullanıcı durumuna göre kendi profilini belirleyebilmektedir. Görme engelliler kullanıcıların program ile etkileşimi ses ve fare ile olmaktadır. Sadece giriş login işleminde klavye kullanması gereken görme engelli kullanıcılar, programı üç tuşlu herhangi bir fare ile yönetebilmektedirler. Program her aşamasında görme engelli kullanıcıyı ses ile uyarmaktadır.

Az gören veya görme sorunları yaşayan kullanıcılar için program, arka plan rengi, yazı büyüklüğü, tipi ve rengi gibi görsel öğeleri değiştirmeye izin vermektedir. Özellikle küçük karakterleri okumakta zorlanan veya hiç okuyamayan, renklerle ilgili görme sıkıntısı yaşayan kullanıcılar profil ayarlarını istedikleri gibi ayarlayarak sistemi en verimli şekilde kullanabilmektedirler.

Görme sıkıntısı yaşamayan diğer kullanıcılar ise standart olarak diğer yazılımları kullandıkları gibi bu yazılımı da kullanabilmektedirler.

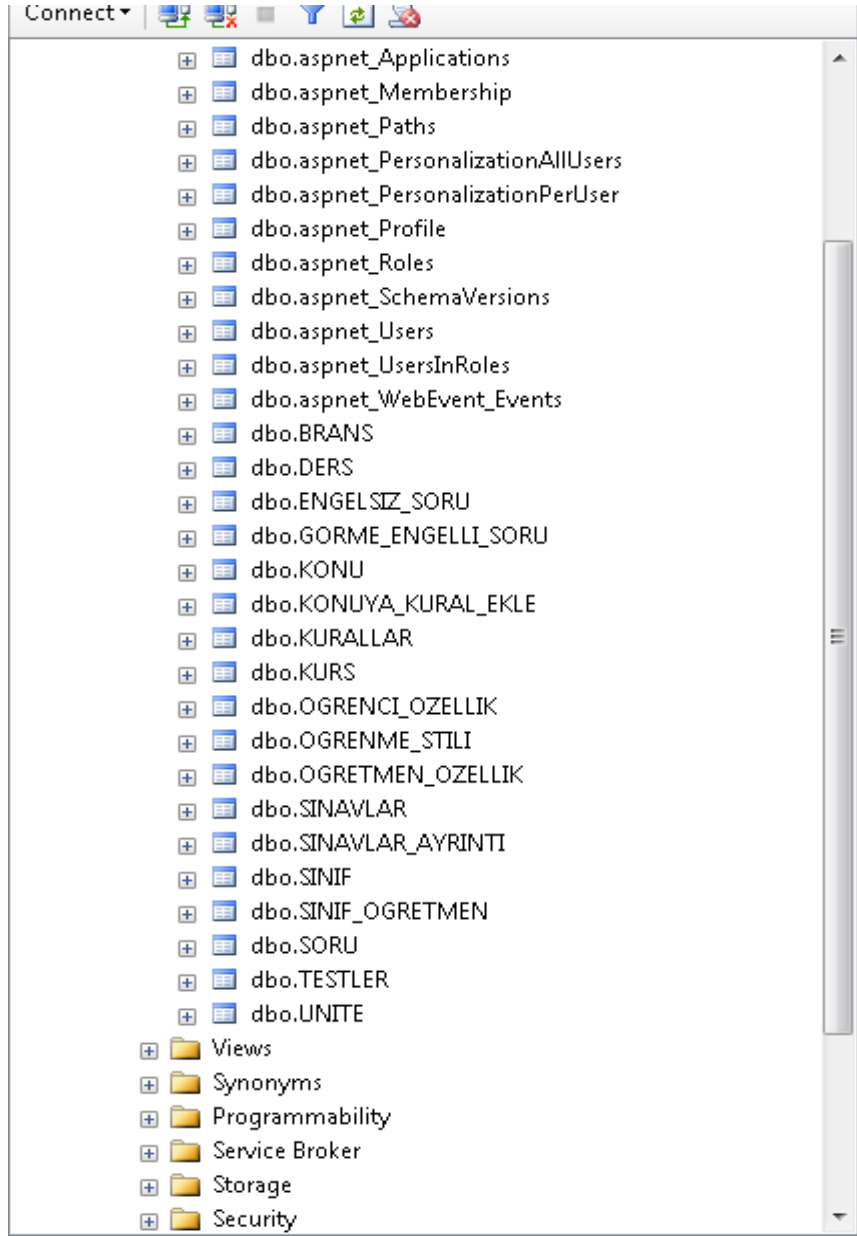
Kullanıcı sisteme ilk giriş sırasında sadece bir kere için kullanıcı tipini ayarlayabilmektedir. Daha sonra sistem kullanıcı tipine göre kendini otomatik olarak ayarlamaktadır.

“Neyi” ile kastedilen, sistemin kullanıcılara hangi konuları öğreteceğini seçmesidir. Kullanıcı her ders için önce “ön test” adı verilen bir teste girer. Derse ait her konudan öğrencilere sorular sorulur ve bu sınavın sonucuna göre öğrenci bazı

derslerden muaf tutulabilir. Buna sistem kendisi karar verir. Eđer öğrenci bir konu ile ilgi sorulardan %80 ve üzeri başarı elde ederse, o dersten muaf tutulur. Kullanıcı muaf tutulduđu dersi görmek zorunda değildir ancak “son test” denilen ve dersin bitirilmesi için başarılması gereken testte yine bu konu ile ilgili sorular karşısına çıkmaktadır.

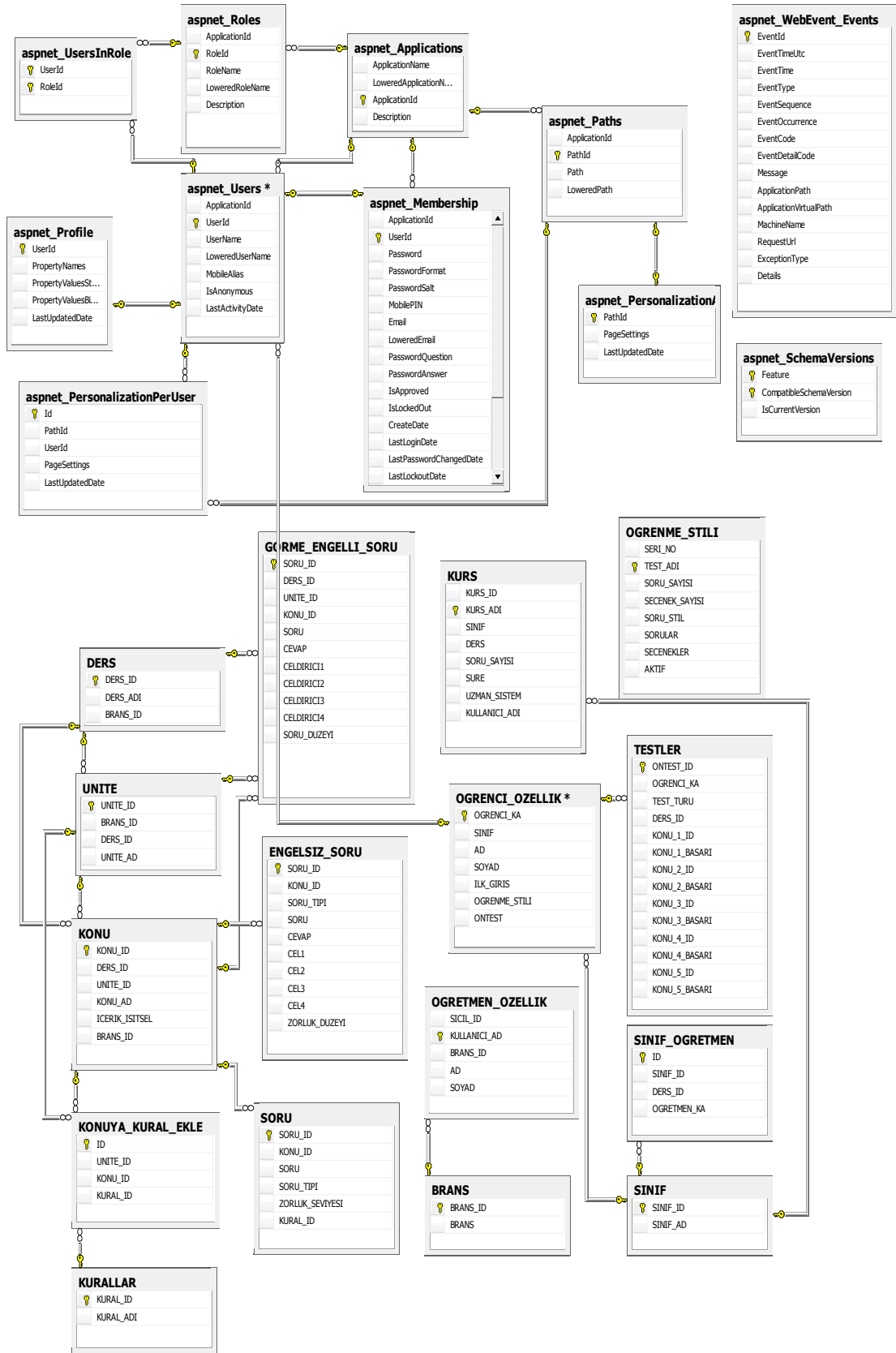
“Nasıl” ile kastedilen öğrencinin hangi öğrenme stiline sahip olduđu sistem tarafından öğrenilip, o stile uygun olarak geliştirilen öğretim programının sunulması işlemdir. Sistem öğrenciyi sahip olduđu öğrenme stiline en uygun bölüme yönlendirmektedir. Yönetici tarafından sisteme girilen bir test veya envantere göre kullanıcı bir teste tabi tutulur. Sistem bulanık mantık prensiplerine göre kullanıcının hangi öğrenme stiline sahip olduğuna karar vermektedir. Sistem yalnızca görme engelliler için bir istisnaya sahiptir. Bunun sebebi ise görme engellilerin sadece duyarak öğrenebilmesidir.

3.3 Veritabanı



Şekil.9. Veritabanı görünümü

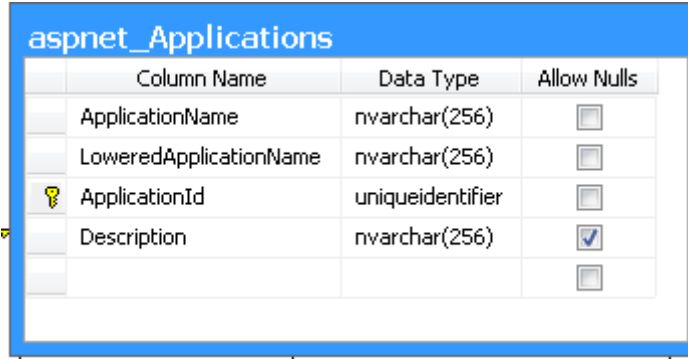
3.3.1 E-R Diyagramı



Şekil.10. E-R Diyagramı

Veritabanı olarak Microsoft'un veritabanı yönetim sistemi olan Microsoft Sql Server 2008 programı kullanıldı. Bu programın ücretsiz sürümü olan Express Edition kullanıldı.

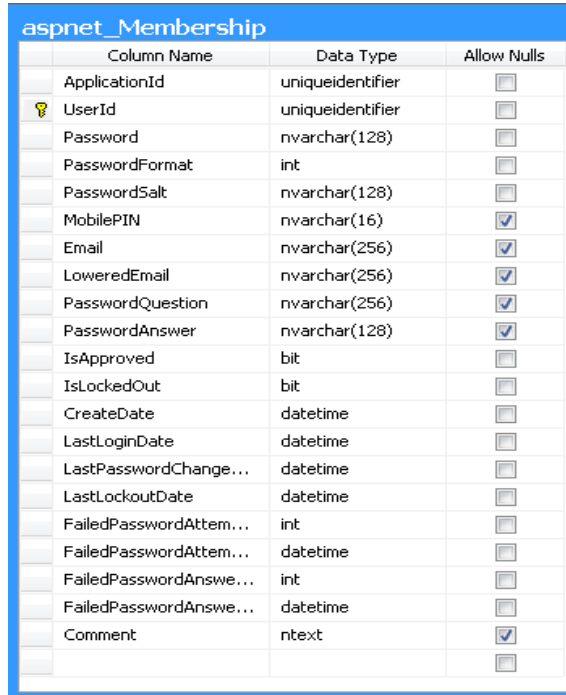
aspnet_Applications: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.



Column Name	Data Type	Allow Nulls
ApplicationName	nvarchar(256)	<input type="checkbox"/>
LoweredApplicationName	nvarchar(256)	<input type="checkbox"/>
ApplicationId	uniqueidentifier	<input type="checkbox"/>
Description	nvarchar(256)	<input checked="" type="checkbox"/>

Şekil.11. aspnet_Applications tablosunun sütun ve veri tipleri

aspnet_Membership: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir. Sisteme kayıt kullanıcıları kullanıcı adı, şifre, eposta adresi, sisteme giriş çıkış saatleri gibi kullanıcı hesapları ile ilgili bilgilerin tutulduğu tablodur.



Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
Password	nvarchar(128)	<input type="checkbox"/>
PasswordFormat	int	<input type="checkbox"/>
PasswordSalt	nvarchar(128)	<input type="checkbox"/>
MobilePIN	nvarchar(16)	<input checked="" type="checkbox"/>
Email	nvarchar(256)	<input checked="" type="checkbox"/>
LoweredEmail	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordQuestion	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordAnswer	nvarchar(128)	<input checked="" type="checkbox"/>
IsApproved	bit	<input type="checkbox"/>
IsLockedOut	bit	<input type="checkbox"/>
CreateDate	datetime	<input type="checkbox"/>
LastLoginDate	datetime	<input type="checkbox"/>
LastPasswordChangeDate	datetime	<input type="checkbox"/>
LastLockoutDate	datetime	<input type="checkbox"/>
FailedPasswordAttemptCount	int	<input type="checkbox"/>
FailedPasswordAttemptDate	datetime	<input type="checkbox"/>
FailedPasswordAnswerCount	int	<input type="checkbox"/>
FailedPasswordAnswerDate	datetime	<input type="checkbox"/>
Comment	ntext	<input checked="" type="checkbox"/>

Şekil.12. aspnet_Membership tablosunun sütun ve veri tipleri

asp_Paths: : Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
PathId	uniqueidentifier	<input type="checkbox"/>
Path	nvarchar(256)	<input type="checkbox"/>
LoweredPath	nvarchar(256)	<input type="checkbox"/>

Şekil.13. aspnet_Paths tablosunun sütun ve veri tipleri

aspnet_PersonalizationAllUsers: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.

Column Name	Data Type	Allow Nulls
PathId	uniqueidentifier	<input type="checkbox"/>
PageSettings	image	<input type="checkbox"/>
LastUpdatedDate	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

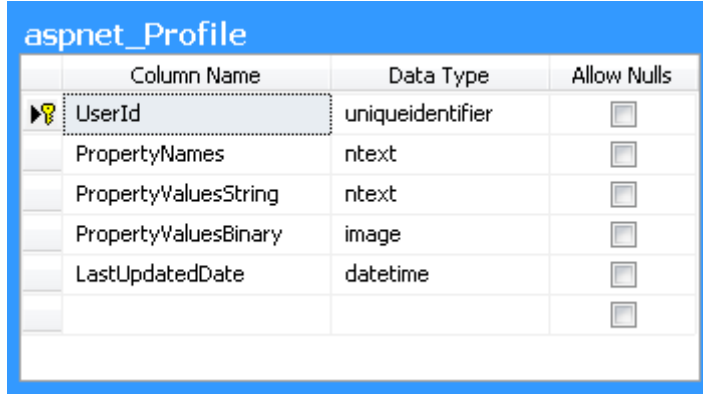
Şekil.14. aspnet_PersonalizationAllUsers tablosunun sütun ve veri tipleri

aspnet_PersonalizationPerUsers: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.

Column Name	Data Type	Allow Nulls
Id	uniqueidentifier	<input type="checkbox"/>
PathId	uniqueidentifier	<input checked="" type="checkbox"/>
UserId	uniqueidentifier	<input checked="" type="checkbox"/>
PageSettings	image	<input type="checkbox"/>
LastUpdatedDate	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

Şekil.15 aspnet_PersonalizationPerUsers tablosunun sütun ve veri tipleri

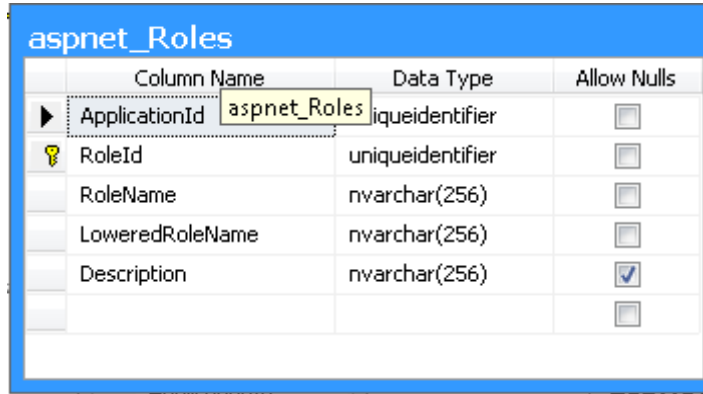
aspnet_Profile: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir. Kullanıcılara ait profillerin tutulduğu tablodur.



Column Name	Data Type	Allow Nulls
UserId	uniqueidentifier	<input type="checkbox"/>
PropertyNames	ntext	<input type="checkbox"/>
PropertyValuesString	ntext	<input type="checkbox"/>
PropertyValuesBinary	image	<input type="checkbox"/>
LastUpdatedDate	datetime	<input type="checkbox"/>

Şekil.16. aspnet_Profile tablosunun sütun ve veri tipleri

aspnet_Roles: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir. Sistemde bulunan rollerin kayıtlı olduğu tablodur. Sistemde üç adet rol vardır. Bunlar: öğrenci, öğretmen ve yöneticidir.



Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
RoleId	uniqueidentifier	<input type="checkbox"/>
RoleName	nvarchar(256)	<input type="checkbox"/>
LoweredRoleName	nvarchar(256)	<input type="checkbox"/>
Description	nvarchar(256)	<input checked="" type="checkbox"/>

Şekil.17. aspnet_Roles tablosunun sütun ve veri tipleri

aspnet_SchemaVersions: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.

aspnet_SchemaVersions			
	Column Name	Data Type	Allow Nulls
▶	Feature	nvarchar(128)	<input type="checkbox"/>
▶	CompatibleSchemaVer...	nvarchar(128)	<input type="checkbox"/>
	IsCurrentVersion	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil.18. aspnet_SchemaVersion tablosunun sütun ve veri tipleri

aspnet_Users: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir. Kullanıcılara ait profillerin tutulduğu tablodur. Sisteme kayıtlı kullanıcıların listelendiği tablodur.

aspnet_Users			
	Column Name	Data Type	Allow Nulls
▶	ApplicationId	uniqueidentifier	<input type="checkbox"/>
▶	UserId	uniqueidentifier	<input type="checkbox"/>
	UserName	nvarchar(256)	<input type="checkbox"/>
	LoweredUserName	nvarchar(256)	<input type="checkbox"/>
	MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
	IsAnonymous	bit	<input type="checkbox"/>
	LastActivityDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil.19. aspnet_User tablosunun sütun ve veri tipleri

aspnet_UserInRoles: Microsoft .Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir. Sisteme kayıt kullanıcıların hangi rolde olduklarının tutulduğu tablodur.

aspnet_UsersInRoles			
	Column Name	Data Type	Allow Nulls
▶	UserId	uniqueidentifier	<input type="checkbox"/>
▶	RoleId	uniqueidentifier	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil.20. aspnet_UsersInRoles tablosunun sütun ve veri tipleri

aspnet_WebEvent_Events: Microsoft.Net Framework 2.0 ile birlikte gelen asp.net' in kendisi için kullandığı tablolardan biridir.

Column Name	Data Type	Allow Nulls
EventId	char(32)	<input type="checkbox"/>
EventTimeUtc	datetime	<input type="checkbox"/>
EventType	nvarchar(256)	<input type="checkbox"/>
EventSequence	decimal(19, 0)	<input type="checkbox"/>
EventOccurrence	decimal(19, 0)	<input type="checkbox"/>
EventCode	int	<input type="checkbox"/>
EventDetailCode	int	<input type="checkbox"/>
Message	nvarchar(1024)	<input checked="" type="checkbox"/>
ApplicationPath	nvarchar(256)	<input checked="" type="checkbox"/>
ApplicationVirtualPath	nvarchar(256)	<input checked="" type="checkbox"/>
MachineName	nvarchar(256)	<input type="checkbox"/>
RequestUrl	nvarchar(1024)	<input checked="" type="checkbox"/>
ExceptionType	nvarchar(256)	<input checked="" type="checkbox"/>
Details	ntext	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil.21. aspnet_ WebEvent_Events tablosunun sütun ve veri tipleri

BRANS: Sistemde kayıtlı olan branşların tutulduğu tablodur. BRANS_ID ve BRANS sütunlarından oluşmuştur.

Column Name	Data Type	Allow Nulls
BRANS_ID	int	<input type="checkbox"/>
BRANS	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil.22. BRANS tablosunun sütun ve veri tipleri

ENGELSİZ_SORU: Görme engeli bulunmayan kullanıcıların soru havuzunun tutulduğu tablodur. SORU_ID, KONU_ID, SORU_TIPI, SORU, CEVAP, CEL1, CEL2, CEL3, CEL4 ve ZORLUK_DUZEYI sütunlarından oluşmuştur.

ENGELSİZ_SORU			
	Column Name	Data Type	Allow Nulls
🔑	SORU_ID	ENGELSİZ_SORU	<input type="checkbox"/>
	KONU_ID	int	<input checked="" type="checkbox"/>
	SORU_TIPI	int	<input checked="" type="checkbox"/>
	SORU	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CEVAP	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CEL1	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CEL2	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CEL3	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CEL4	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ZORLUK_DUZEYI	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.23. ENGELSİZ_SORU tablosunun sütun ve veri tipleri

GORME_ENGELLI_SORU: Görme engeli bulunmayan kullanıcıların soru havuzunun tutulduğu tablodur. SORU_ID, DERS_ID, UNITE_ID, KONU_ID, KONU_ID, SORU_TIPI, SORU, CEVAP, CEL1, CEL2, CEL3, CEL4 ve ZORLUK_DUZEYI sütunlarından oluşmuştur.

GORME_ENGELLI_SORU			
	Column Name	Data Type	Allow Nulls
🔑	SORU_ID	int	<input type="checkbox"/>
	DERS_ID	int	<input checked="" type="checkbox"/>
	UNITE_ID	int	<input checked="" type="checkbox"/>
	KONU_ID	int	<input checked="" type="checkbox"/>
	SORU	nvarchar(50)	<input checked="" type="checkbox"/>
	CEVAP	nvarchar(50)	<input checked="" type="checkbox"/>
	CELDIRICI1	nvarchar(50)	<input checked="" type="checkbox"/>
	CELDIRICI2	nvarchar(50)	<input checked="" type="checkbox"/>
	CELDIRICI3	nvarchar(50)	<input checked="" type="checkbox"/>
	CELDIRICI4	nvarchar(50)	<input checked="" type="checkbox"/>
	SORU_DUZEYI	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.24. GORME_ENGELLI_SORU tablosunun sütun ve veri tipleri

DERS: Sisteme öğretmen rolündeki kullanıcılar tarafından kaydedilen derslerin kaydedildiği tablodur. DERS_ID, DERS_ADI ve KULLANICI_AD sütunlarından oluşmuştur.

DERS			
	Column Name	Data Type	Allow Nulls
🔑	DERS_ID	int	<input type="checkbox"/>
	DERS_ADI	nvarchar(50)	<input checked="" type="checkbox"/>
	BRANS_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.25. DERS tablosunun sütun ve veri tipleri

KONU: Sisteme öğretmen rolündeki kullanıcılar tarafından kaydedilen konunun kaydedildiği tablodur. KONU_ID, DERS_ID, UNITE_ID, KONU_AD ve ICERIK sütunlarından oluşmuştur.

KONU			
	Column Name	Data Type	Allow Nulls
🔑	KONU_ID	int	<input type="checkbox"/>
	DERS_ID	int	<input checked="" type="checkbox"/>
	UNITE_ID	int	<input checked="" type="checkbox"/>
	KONU_AD	nvarchar(50)	<input checked="" type="checkbox"/>
	ICERIK_ISITSEL	nvarchar(50)	<input checked="" type="checkbox"/>
	BRANS_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

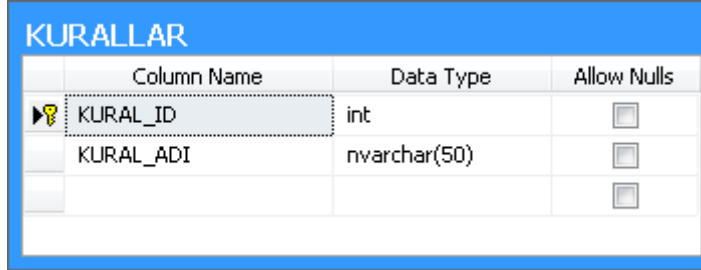
Şekil.26. KONU tablosunun sütun ve veri tipleri

KONUYA_KURAL_EKLE: Sisteme kayıtlı konular ile matematiksel kuralları ilişkilendiren tablodur. ID, KONU_ID, KURAL_ID ve UNITE_ID sütunlarından oluşur.

KONUYA_KURAL_EKLE			
	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	UNITE_ID	int	<input type="checkbox"/>
	KONU_ID	int	<input type="checkbox"/>
	KURAL_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

Şekil.27. KONUYA_KURAL_EKLE tablosunun sütun ve veri tipleri

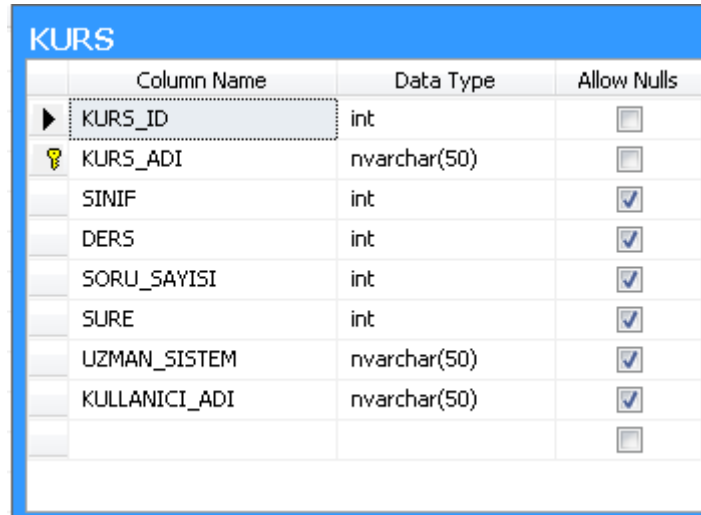
KURALLAR: Sisteme kayıtlı olan soruların oluşturulması ve çözülmesinde kullanılan kuralların tutulduğu bölümdür. KURAL_ID ve KURAL adı sütunlarından oluşmuştur.



Column Name	Data Type	Allow Nulls
KURAL_ID	int	<input type="checkbox"/>
KURAL_ADI	nvarchar(50)	<input type="checkbox"/>

Şekil.28. KURALLAR tablosunun sütun ve veri tipleri

KURS: Her sınıfa hangi eğitimin açıldığını, ön test soru sayısının ve sınav sürelerinin kaydedildiği tablodur. KURS_ID, KURS_ADI, SINIF, DERS, SORU_SAYISI ve SURE sütunlarından oluşmuştur.



Column Name	Data Type	Allow Nulls
KURS_ID	int	<input type="checkbox"/>
KURS_ADI	nvarchar(50)	<input type="checkbox"/>
SINIF	int	<input checked="" type="checkbox"/>
DERS	int	<input checked="" type="checkbox"/>
SORU_SAYISI	int	<input checked="" type="checkbox"/>
SURE	int	<input checked="" type="checkbox"/>
UZMAN_SISTEM	nvarchar(50)	<input checked="" type="checkbox"/>
KULLANICI_ADI	nvarchar(50)	<input checked="" type="checkbox"/>

Şekil.29. KURS tablosunun sütun ve veri tipleri

OGRENCI_OZELLIK: Sisteme kayıtlı öğrencilerin adı, soyadı, ön test olup olmadığı, öğrenme stili, sisteme kaçınıcı girişi gibi bilgilerinin kaydedildiği tablodur. OGRENCI_KA, SINIF, AD, SOYAD, ILK_GIRIS, ONTEST ve OGRENME_STILI sütunlarından oluşmaktadır.

OGRENCI_OZELLIK			
	Column Name	Data Type	Allow Nulls
▶	OGRENCI_KA	nvarchar(50)	<input type="checkbox"/>
	SINIF	int	<input checked="" type="checkbox"/>
	AD	nvarchar(50)	<input checked="" type="checkbox"/>
	SOYAD	nvarchar(50)	<input checked="" type="checkbox"/>
	ILK_GIRIS	bit	<input checked="" type="checkbox"/>
	OGRENME_STILI	nchar(2)	<input checked="" type="checkbox"/>
	ONTEST	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.30. OGRENCI_OZELLIK tablosunun sütun ve veri tipleri

OGRETMEN_OZELLIK: Sisteme kayıtlı öğretmenlerin sicil numaraları, branşı, adı ve soyadı gibi özelliklerinin tutulduğu tablodur. SICIL_ID, KULLANICI_AD, BRANS_ID, AD ve SOYAD sütunlarından oluşmaktadır.

OGRETMEN_OZELLIK			
	Column Name	Data Type	Allow Nulls
▶	SICIL_ID	int	<input type="checkbox"/>
⚠	KULLANICI_AD	nvarchar(50)	<input type="checkbox"/>
	BRANS_ID	int	<input checked="" type="checkbox"/>
	AD	nchar(40)	<input checked="" type="checkbox"/>
	SOYAD	nchar(40)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

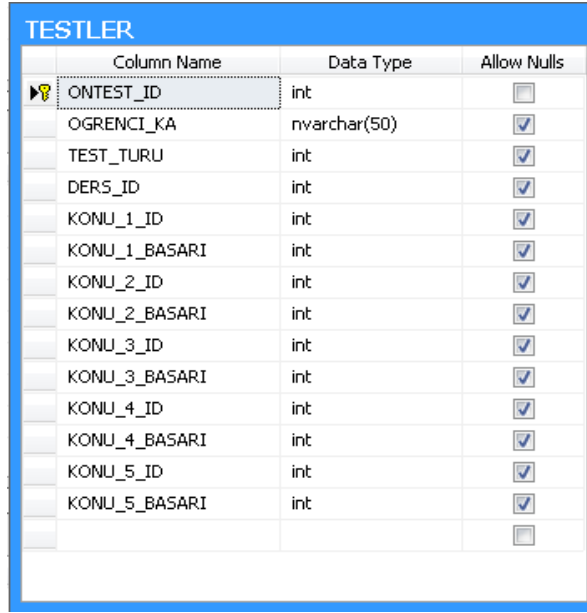
Şekil.31. OGRETMEN_OZELLIK tablosunun sütun ve veri tipleri

SINIF: Sisteme kayıtlı sınıf bilgilerinin tutulduğu tablodur. SINIF_ID ve SINIF_AD sütunlarından oluşmaktadır.

SINIF			
	Column Name	Data Type	Allow Nulls
⚠	SINIF_ID	int	<input type="checkbox"/>
	SINIF_AD	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.32. SINIF tablosunun sütun ve veri tipleri

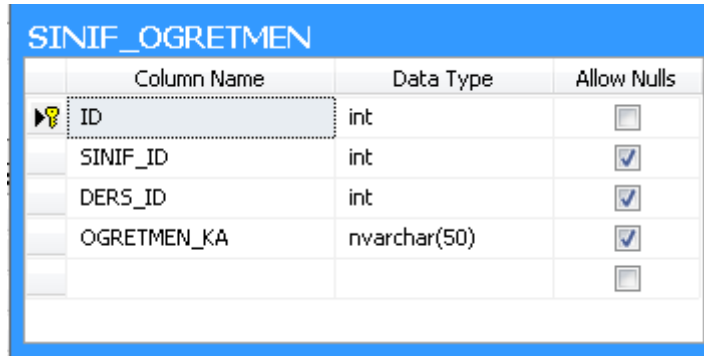
TESTLER: Sisteme kayıtlı öğrencilerin girdikleri sınavların bilgilerinin tutulduğu tablodur.



Column Name	Data Type	Allow Nulls
ONTEST_ID	int	<input type="checkbox"/>
OGRENCI_KA	nvarchar(50)	<input checked="" type="checkbox"/>
TEST_TURU	int	<input checked="" type="checkbox"/>
DERS_ID	int	<input checked="" type="checkbox"/>
KONU_1_ID	int	<input checked="" type="checkbox"/>
KONU_1_BASARI	int	<input checked="" type="checkbox"/>
KONU_2_ID	int	<input checked="" type="checkbox"/>
KONU_2_BASARI	int	<input checked="" type="checkbox"/>
KONU_3_ID	int	<input checked="" type="checkbox"/>
KONU_3_BASARI	int	<input checked="" type="checkbox"/>
KONU_4_ID	int	<input checked="" type="checkbox"/>
KONU_4_BASARI	int	<input checked="" type="checkbox"/>
KONU_5_ID	int	<input checked="" type="checkbox"/>
KONU_5_BASARI	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil.33. BRANS tablosunun sütun ve veri tipleri

SINIF_OGRETMEN: Sisteme kayıtlı öğretmenlerle sınıfları ilişkilendiren tablodur. SINIF_ID, DERS_ID ve OGRETMEN_KA sütunlarından oluşmaktadır.



Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
SINIF_ID	int	<input checked="" type="checkbox"/>
DERS_ID	int	<input checked="" type="checkbox"/>
OGRETMEN_KA	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil.34. SINIF_OGRETMEN tablosunun sütun ve veri tipleri

SORU: Sisteme kayıt öğretmenler tarafından ders, ünite, konu, kural seçilerek oluşturulan soru kalıplarının oluşturulduğu ve kaydedildiği tablodur. SORU_ID, DERS_ID, UNITE_ID, KONU_ID, SORU, KURAL_ID, SORU_TIPI, ZORLUK_SEVIYESI ve KULLANICI_AD sütunlarında oluşmuştur.

SORU			
	Column Name	Data Type	Allow Nulls
🔑	SORU_ID	int	<input type="checkbox"/>
	KONU_ID	int	<input checked="" type="checkbox"/>
	SORU	nvarchar(MAX)	<input checked="" type="checkbox"/>
	SORU_TIPI	int	<input checked="" type="checkbox"/>
	ZORLUK_SEVIYESI	int	<input checked="" type="checkbox"/>
	KURAL_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.35. SORU tablosunun sütun ve veri tipleri

UNITE: Sistemdeki derslere bağlı ünitelerin kaydedildiği tablolardır. UNITE_ID, DERS_ID VE UNITE_AD sütunlarından oluşmuştur.

UNITE			
	Column Name	Data Type	Allow Nulls
🔑	UNITE_ID	int	<input type="checkbox"/>
	BRANS_ID	nchar(10)	<input checked="" type="checkbox"/>
	DERS_ID	int	<input checked="" type="checkbox"/>
	UNITE_AD	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.36. UNITE tablosunun sütun ve veri tipleri

OGRENME_STILI: Öğrenme stili testlerinin tutulduğu tablodur.

OGRENME_STILI			
	Column Name	Data Type	Allow Nulls
▶	SERI_NO	int	<input type="checkbox"/>
🔑	TEST_ADI	nvarchar(50)	<input type="checkbox"/>
	SORU_SAYISI	int	<input checked="" type="checkbox"/>
	SECENEK_SAYISI	int	<input checked="" type="checkbox"/>
	SORU_STIL	nvarchar(MAX)	<input checked="" type="checkbox"/>
	SORULAR	nvarchar(MAX)	<input checked="" type="checkbox"/>
	SECENEKLER	nvarchar(MAX)	<input checked="" type="checkbox"/>
	AKTIF	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Şekil.37. OGRENME_STILI tablosunun sütun ve veri tipleri

3.4 Sistemin Elemanları

Sistemde üç tane rol vardır. Bunlar;

3.4.1 Yönetici

Yönetici sisteme öğrenci ve öğretmenleri ekleyen kullanıcıdır. Öğretmen rolündeki kullanıcılarının sisteme eklenmesi, silinmesi gibi işlemleri gerçekleştirmektedir. Ayrıca öğrenci rolündeki kullanıcıların kayıtlarını onaylama işlemini gerçekleştirir.

3.4.2 Öğretmen

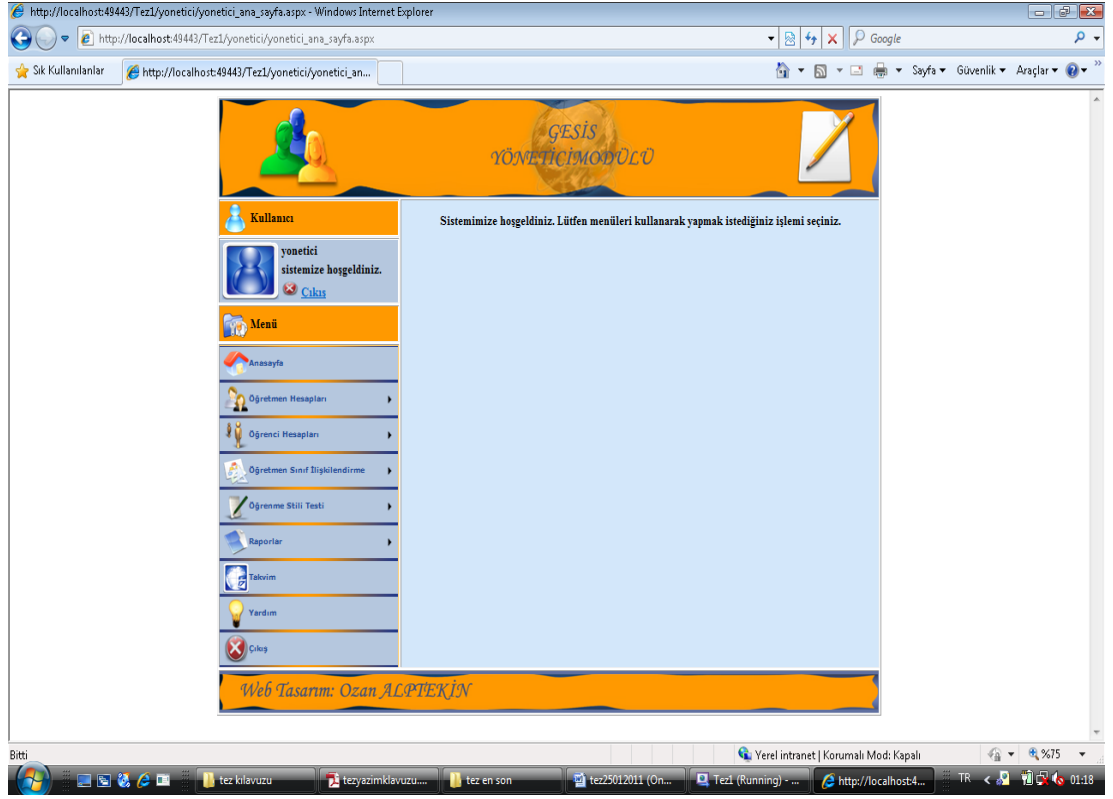
Sistemdeki en kilit role sahiptir. Derslerin planlanması, derslere içeriklerin eklenmesi, soru havuzunun ve kuralların oluşturulması gibi işlemleri gerçekleştirir. Öğrencilerin yaptıkları işlere ait raporları ve istatistikleri takip eder.

3.4.3 Öğrenci

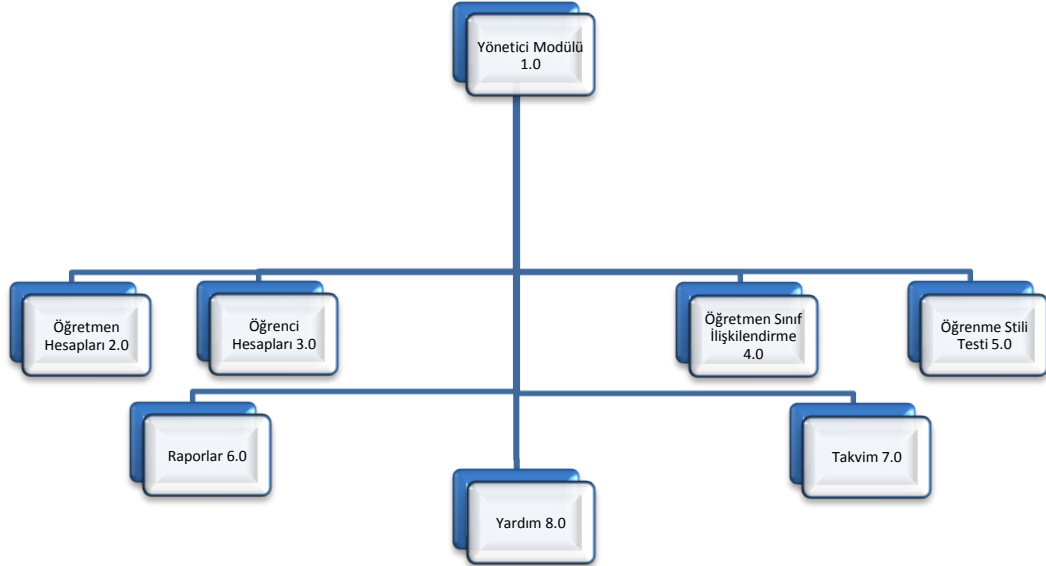
Sistemin hedefindeki role sahip kullanıcılardır. Normal görenler, görme zorluğu çekenler, az görenler ve hiç görmeyen kullanıcılar hedeflenmiştir. Sistemin belirlediği derslere girerler, yapılan ölçme işlemlerine göre sistem tarafından yönlendirilirler.

4. TASARLANAN UYGULAMANIN İNCELENMESİ

4.1 Yönetici Modülü



Şekil.38. Yönetici modülü anasayfası



Şekil.39. Yönetici modülünün bölümleri

Yönetici modülü öğrenci ve öğretmen kullanıcı hesaplarının kontrolünü sağlayan, öğretmen sınıf ilişkilendirmesini yapan ve öğrenme stil testlerini kaydeden modüldür. Öğretmen kullanıcı hesabı oluşturma, öğrenci kullanıcı hesapları yönetimi, raporlama işlemleri ile uğraşmaktadır.

4.1.1 Öğretmen Hesapları

Öğretmenler sisteme kullanıcı adları ve şifreleri ile girmektedir. Öğretmenlerin kullanıcı hesaplarının yönetildiği menü bu menüdür.. Öğretmenlerin sistemi kullanabilmeleri için yönetici tarafından hesap açılması ve hesapların yönetilmesi işlemleri yapılmaktadır. Bu menüde öğretmen hesapları oluşturulur, silinir, aranır ve düzeltilir.

4.1.2 Öğrenci Onay İşlemleri

Öğrenciler sisteme kullanıcı adı ve şifre ile girmektedir. Öğrenciler kullanıcı adlarını, şifrelerini ve kişisel bilgilerini kendileri oluşturmaktadır. Ancak sisteme giriş yapabilmeleri için hesaplarının yönetici tarafından onaylanması gerekmektedir. Yönetici, bu menüde kullanıcı hesaplarını ya onaylamaktadır ya da onaylanmadan silmektedirler.

4.1.3 Öğretmen Sınıf İlişkilendirme

Yönetici, bu menüde sistemde kayıtlı olan öğretmenleri öğretim dallarına göre sınıflara atanmaktadır. Eğer bir sınıfa bir ders için öğretmen ataması yapılmazsa o sınıf için o ders açılmayacaktır.

4.1.4 Öğrenme Stilleri

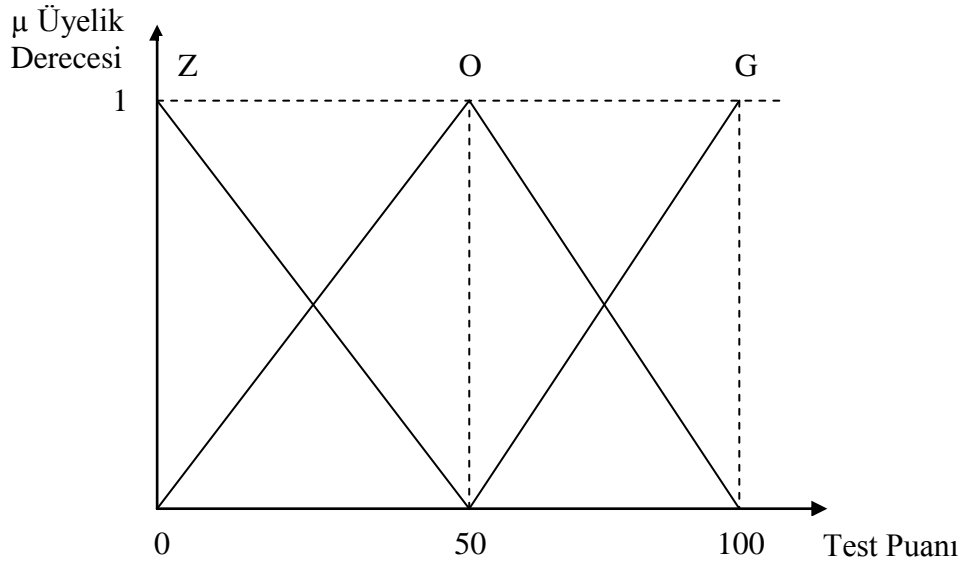
Sistemin en kritik menülerinden biridir. Bu menüde, öğrencilerin öğrenme stilini bulabilmek için, test girişi yapılmaktadır. Testin tipi, soru sayısı, seçenek sayısı tamamen yönetici tarafından Fleming öğrenme stiline uygun olarak oluşturulmaktadır. Birçok test girişi yapılabilmektedir ancak sadece bir tanesi aktif olarak kullanılabilir.

4.1.4.1 Öğrenim Stili Testi Genel Bilgi

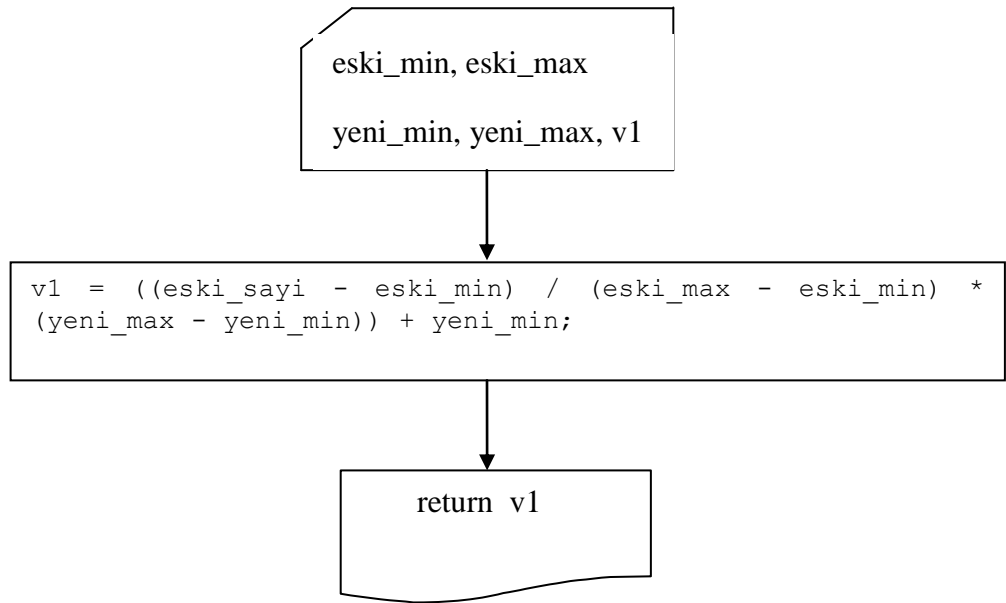
Öğrenmeyi olumlu ya da olumsuz olarak etkileyen etmenler vardır. Bunlar öğrenenle, öğrenme yönetimiyle, öğrenilecek malzemenin türü ve öğrenme ortamıyla ilgili etmenlerdir (Yeşilyaprak, B., 2003). Öğrenenle ilgili önemli etmenlerin biri olan “bireysel farklılıklar” öğretimin gerçekleştirilmesinde göz ardı edilemeyecek unsurlardandır. Her öğrencinin, diğerlerinden ayrılan bireysel özelliklere sahip olduğu kabul edilip, öğretim sistemi ona göre inşa edilirse, öğretim sürecinin daha sağlıklı yürür.

Sistem derslere başlamadan önce kullanıcıyı öğrenme stili testine zorlamaktadır yani öğrenme stili testi yapılmadan derslere başlamasına izin vermemektedir. Öğrenme stili, yönetici tarafından girilen bir envanterdir. Çok çeşitli öğrenme stilleri yaklaşımları olmasına rağmen sistem için seçilen öğrenme stili “Fleming’ in Öğrenme Stilleri”dir. Fleming’ e göre bir kişinin öğrenme stili görsel, işitsel, dokunsal ve kinetiksel olabilmektedir. Eğer ikiden çok öğrenme stiline sahipse, kişi, çok stilli öğrenmeye sahip kabul edilir. Buradan hareketle sistem yedi farklı öğrenme grubuna ayrılmaktadır.

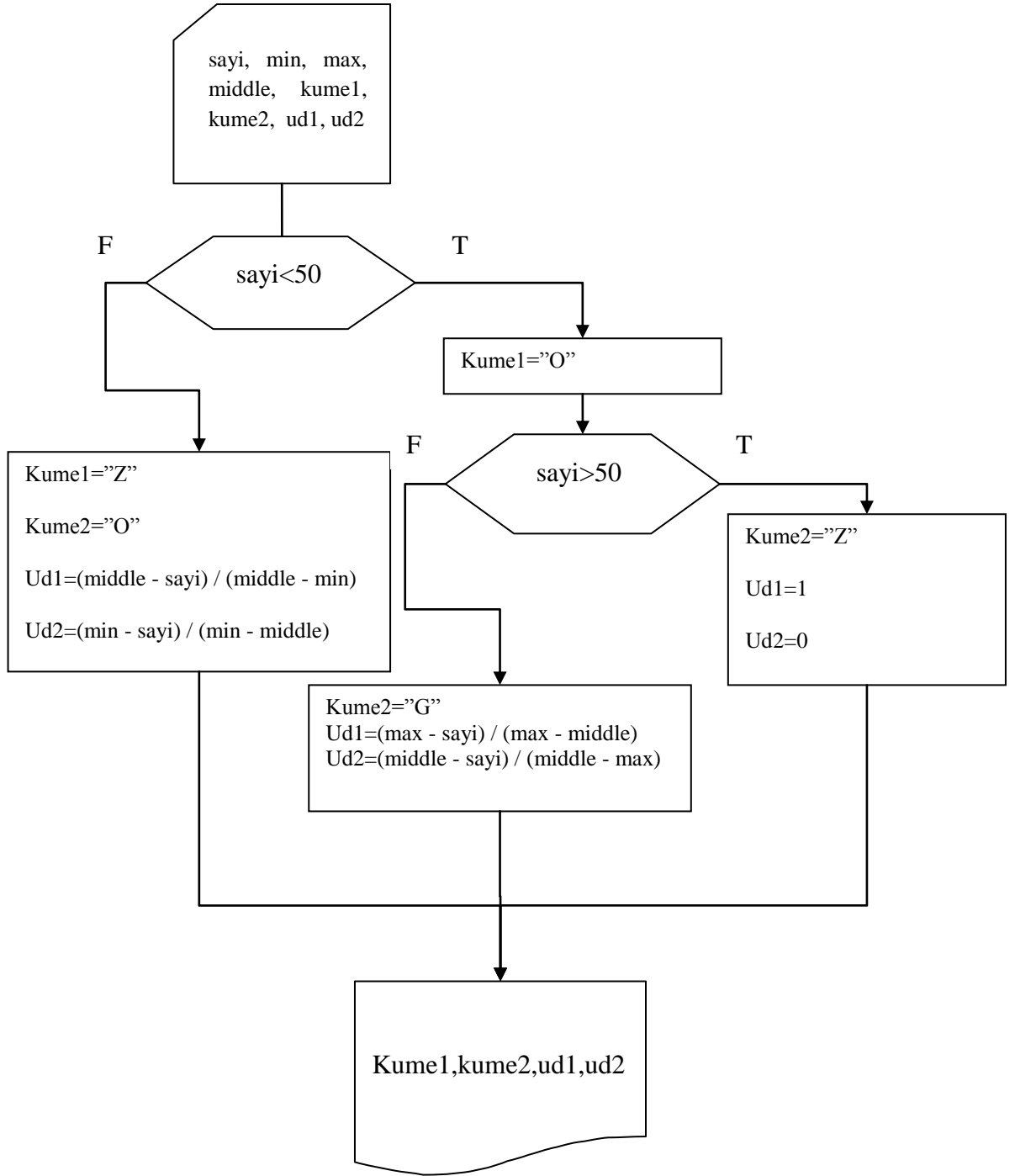
- a) HC: Hiçbir öğrenme stiline sahip olmayan veya ağırlıklı olarak görsel, işitsel ve dokunsal öğrenme stillerine sahip kullanıcıların öğrenme grubudur.
- b) G: Ağırlıklı olarak görsel öğrenme stiline sahip kullanıcıların öğrenme grubudur.
- c) I: Ağırlıklı olarak işitsel öğrenme stiline sahip kullanıcıların öğrenme grubudur.
- d) D: Ağırlıklı olarak dokunsal öğrenme stiline sahip kullanıcıların öğrenme grubudur.
- e) GI: Ağırlıklı olarak görsel ve işitsel öğrenme stiline sahip kullanıcıların öğrenme grubudur
- f) GD: Ağırlıklı olarak görsel ve dokunsal öğrenme stiline sahip kullanıcıların öğrenme grubudur.
- g) ID: Ağırlıklı olarak işitsel ve dokunsal öğrenme stiline sahip kullanıcıların öğrenme grubudur.



Şekil.40. Öğrenci üyelik derecesi bulma grafiği



Şekil.41. Normalizasyon algoritması



Şekil.42. Bulanıklaştırma algoritması

Teste tabi tutulan kullanıcıların aldıkları puanlar öğrenme stilleri bazında Z(Zayıf), O(Orta) ve G(Güçlü) olarak gruplandırılır.

G(3⁰)	I(3¹)	D(3²)	X(ÇIKIŞ)
0	0	0	0
0	0	1	9
0	0	2	18
0	1	0	3
0	1	1	12
0	1	2	21
0	2	0	6
0	2	1	15
0	2	2	24
1	0	0	1
1	0	1	10
1	0	2	19
1	1	0	4
1	1	1	13
1	1	2	22
1	2	0	7
1	2	1	16
1	2	2	25
2	0	0	2
2	0	1	11
2	0	2	20
2	1	0	5
2	1	1	14
2	1	2	23
2	2	0	8
2	2	1	17
2	2	2	26

Tablo.2. Bulanık mantık karar tablosu

0:ZAYIF, 1:ORTA, 2:GÜÇLÜ

1. Kural: Eğer $x \bmod 13=0$ ise öğrenme stili HC'dir.

(G=I=D)

2. Kural: Eğer $x=1$ veya $x=2$ veya $x=5$ veya $x=11$ veya $x=14$ ise öğrenme stili G'dir.

(G>I VE G>D)

3. Kural: Eğer $x=3$ veya $x=6$ veya $x=7$ veya $x=15$ veya $x=16$ ise öğrenme stili I'dir.

(I>G VE I>D)

4. Kural: Eğer $x=9$ veya $x=18$ veya $x=19$ veya $x=21$ veya $x=22$ ise öğrenme stili D'dir.

(D>G VE D>I)

5. Kural: Eğer $x=4$ veya $x=8$ veya $x=17$ ise öğrenme stili GI'dir.

(G=I>D)

6. Kural: Eğer $x=12$ veya $x=24$ veya $x=25$ ise öğrenme stili ID'dir.

(D=I>G)

7. Kural: Eğer $x=10$ veya $x=20$ veya $x=23$ ise öğrenme stili GD'dir.

(G=D>I)

Sonuç olarak yukarıdaki tablo ve kurallara göre hesaplanan puanlar maximum üyelik derecesi prensibine göre en yüksek üyelik puan çarpımına sahip kural kabul edilmekte ve bu kuralın sonucu olan grup öğrenme stili olarak kullanıcının profiline eklenmektedir.

4.1.5 Raporlar

Öğretmen ve öğrencilerle ilgili raporların verildiği menüdür.

4.1.6 Takvim

Yöneticilerin kullanabileceği kullanışlı bir takvim içeren menüdür.

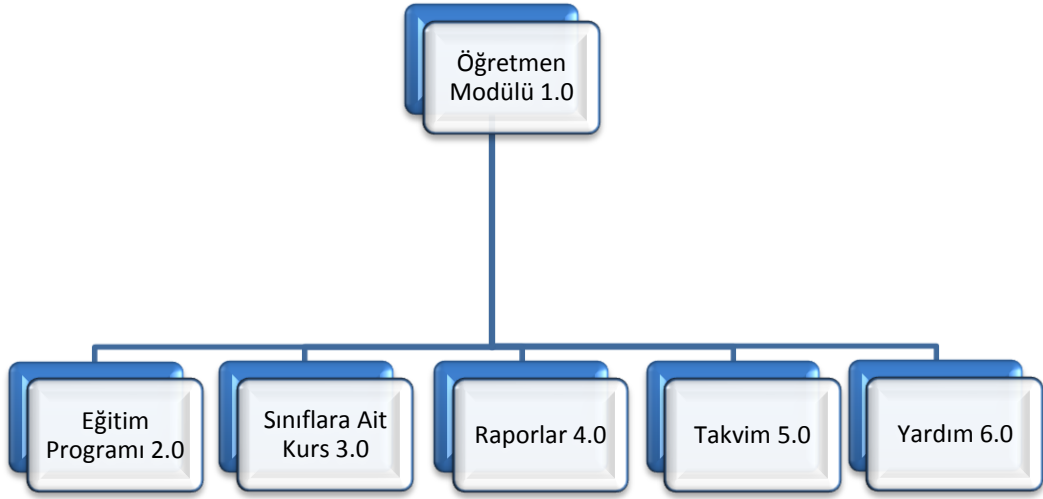
4.1.7 Yardım

Yönetici modülünün kullanımı ile ilgili bilgilerin yer aldığı menüdür. Bu menüde, yöneticinin sistemi kullanırken karşılaşılabileceği zorlukları aşabilmesi için kısa ve yönlendirici bilgiler içermektedir.

4.2 Öğretmen Modülü



Şekil.43. Öğretmen modülü anasayfa



Şekil.44. Öğretmen modülünün bölümleri

Yönetici tarafından öğretmen rolü ile eklenmiş kullanıcıların dersleri, konuları, soru havuzu, soru kalıpları girmesini ve kurs oluşturmasını sağlayan bölümdür.

4.2.1 Eğitim Programı

Eğitim programında yer alan dersler ve konuların görüldüğü ve soruların oluşturulduğu bölümdür.

4.2.1.1 Engelsizler İçin Soru Oluşturma Modülü

Bu modül “ön test”, “konu testi” ve “son test”e 25’er soruluk test hazırlayan uzman bir sistemdir. Üç değişik şekilde soru hazırlayabilmektedir; klasik beş seçenekli çoktan seçmeli soru, doğru-yanlış sorusu ve boşluk doldurma sorusu. Öğretmen tarafından hazırlanan sorular bu modülle öğrenciye sunulmaktadır.

Sistem üç değişik şekilde soru hazırlamaktadır.

a-)Soru Havuzu: Öğretmen tarafından hazırlanan kalıp sorular soru modülü tarafından okunup öğrenciye sunulmaktadır.

b-)Soru Uzmanı: Sayı limitleri öğretmen rolündeki kullanıcı tarafından hazırlanan bu uzman sistem, otomatik olarak soru ve çeldiricileri hazırlamaktadır.

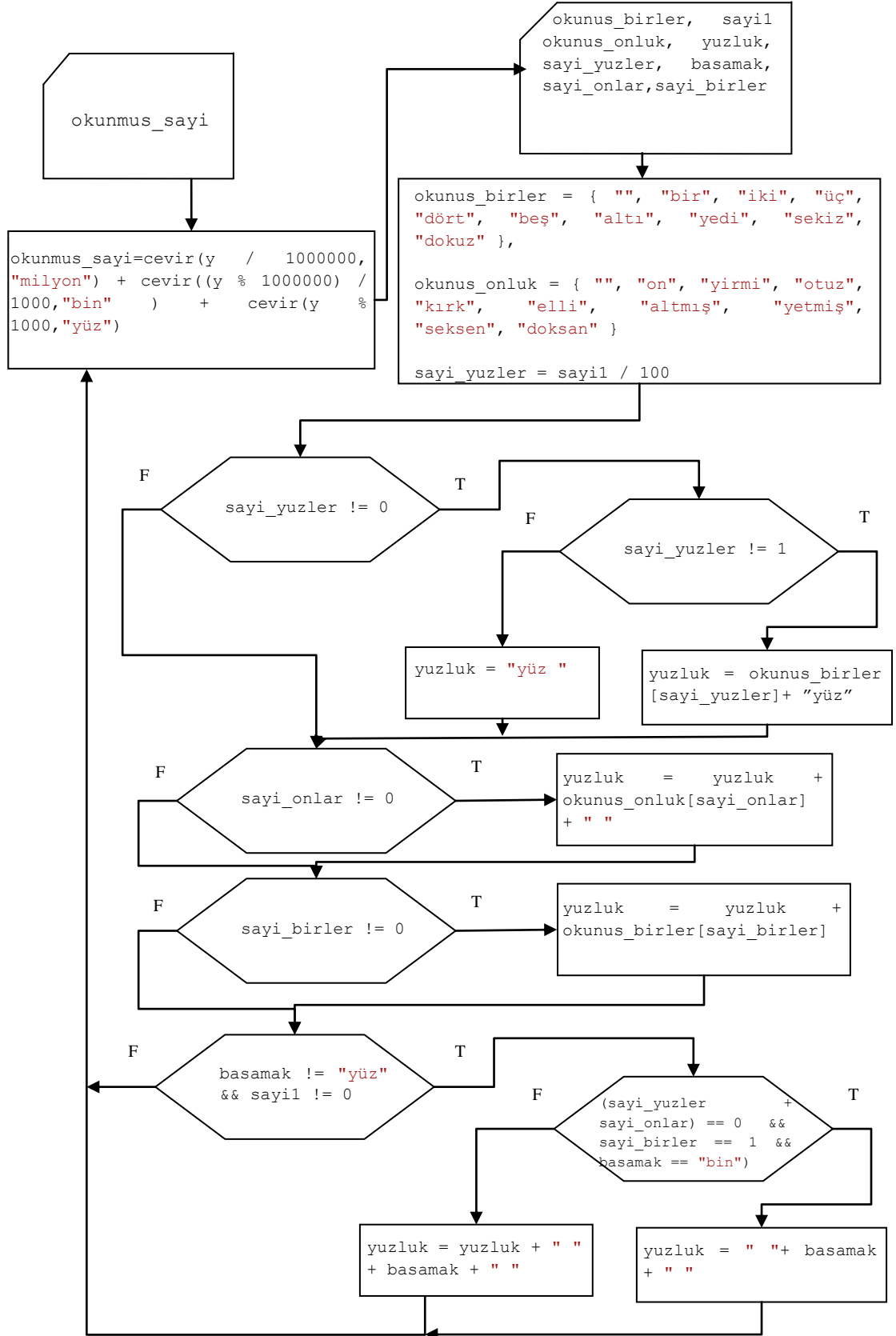
Sorular belirlenen kurallara uygun şekilde ve belirlenen limitler dâhilinde uzman sistem tarafından hazırlanmaktadır. Çeldiriciler ise rastgele değil doğru yanıtı uygun olarak oluşturulmaktadır.

c-)Karma: Belirli bir kısmı soru havuzundan çekilen, belirli bir kısmı da soru uzmanı tarafından hazırlanan testlerdir.

4.2.1.2 Soru Uzmanı Modülü

Bu modül öğretmenin belirlediği sınırlar ve kalıplar dahilinde sistem tarafından soru üreten bir modüldür. Sistem içerisindeki yedi kurala göre soruyu ve çeldiricileri kendisi üretir.

4.2.1.2.1 7,8 ve 9 Basamaklı Sayıların Yazıya Çevrilmesi



Şekil.45. 7,8 ve 9 basamaklı sayıları yazıya çeviren algoritma

Bu kurala göre sistem 7,8 ve 9 basamaklı sayıları yazıya çevirmektedir.

2. soru Aşağıdakilerden hangisi 9349003 sayısının okunusudur?

- doksan üç bin dört yüz doksan milyon otuz
- dokuz milyon üç yüz kırk dokuz bin üç
- doksan üç milyon dört yüz doksan bin otuz
- dokuz milyon üç yüz kırk dokuz milyon üç
- dokuz üç yüz kırk dokuz üç

Şekil.46. 7,8 ve 9 basamaklı sayıları yazıya çeviren kural için örnek soru

4.2.1.2.2 7,8 ve 9 Basamaklı Sayıların Sayıya Çevrilmesi

Bu kurala göre sistem 7,8 ve 9 basamaklı yazıyla yazılmış sayıların, rakamla yazılmış sayıya çevirmektedir.

5. soru Aliye, öğretmeni 4834144 şeklinde bir sayı okuyor. Ali bu sayıyı defterine dört milyon sekiz yüz otuz dört bin yüz kırk dört şeklinde yazıyor. Ahmet sayıyı doğru şekilde yazmış mıdır?

- Doğru
- Yanlış

Şekil.47. 7,8 ve 9 basamaklı yazıyla yazılmış sayıların rakamla yazılmış sayıya çeviren kural için örnek soru

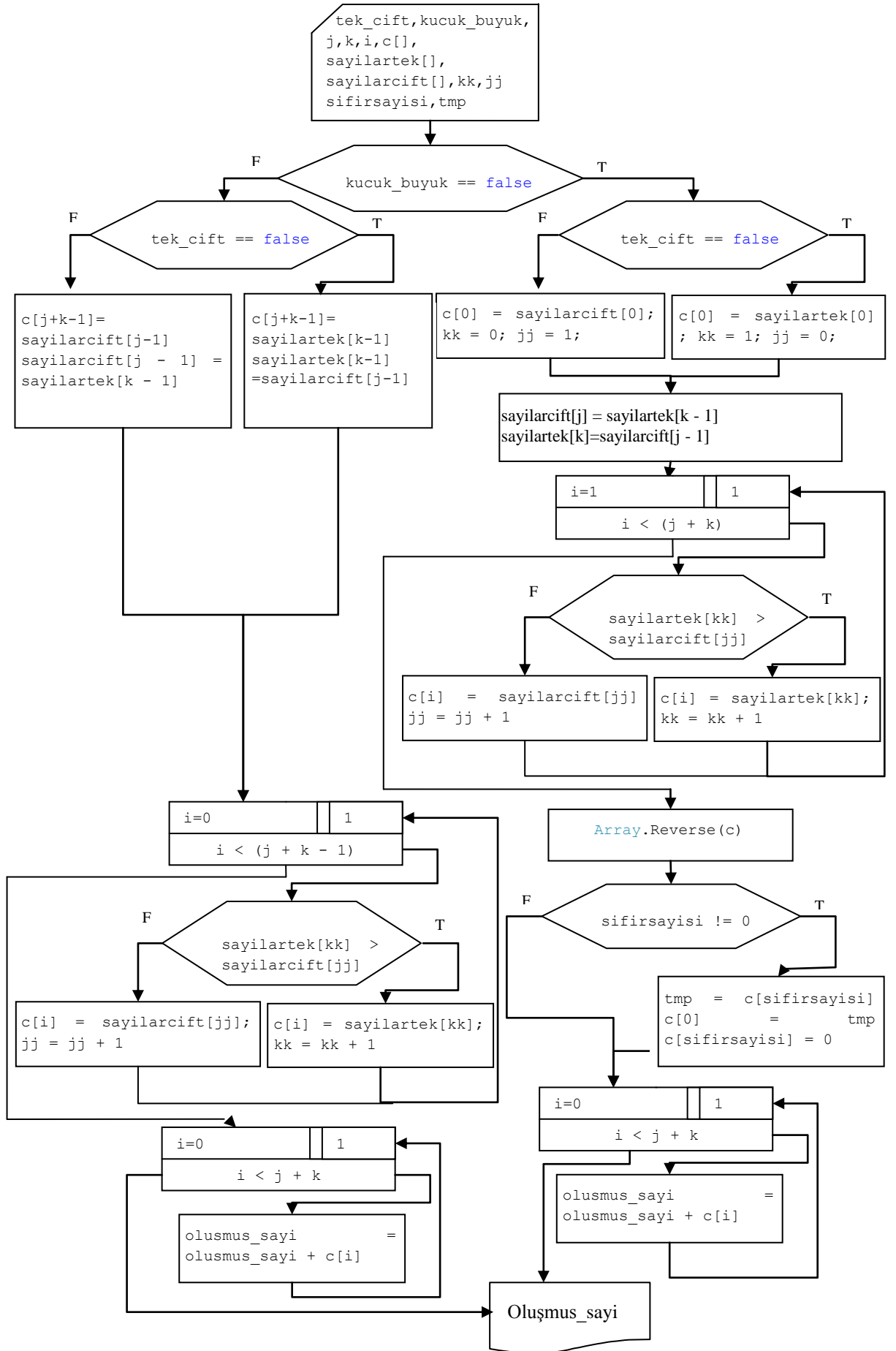
4.2.1.2.3 Belirli Sayıdaki Rakamı En Büyük En Küçük Sayıyı Oluşturacak Şekilde Sıralama Kuralı

Bu kurala göre sistem verilen belirli sayıda rakamı en büyük veya en küçük olacak şekilde sayıya çevirmektedir.

7. soru Aşağıdakilerden hangisi 2,7,9,5,8,3,6,1 sayılarından oluşan en büyük çift sayıdır?

- 38765912
- 98765312
- 98165372
- 98265317
- 98761352

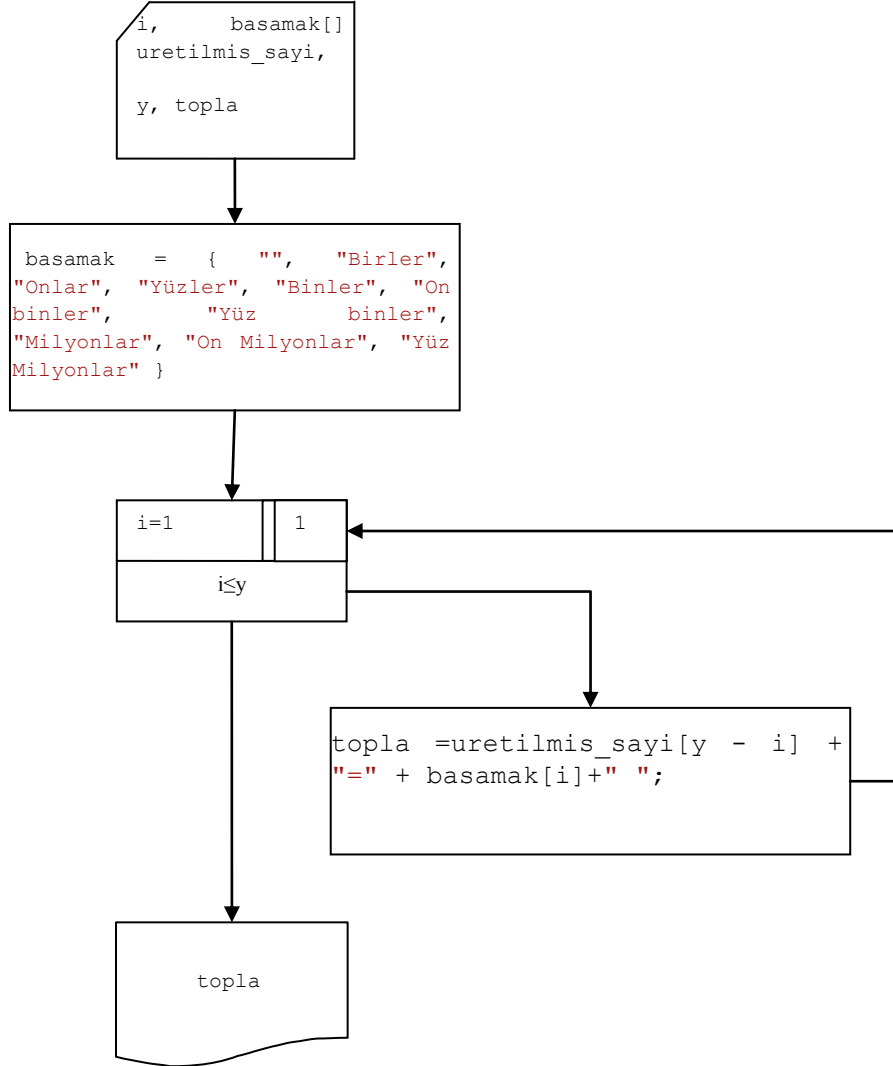
Şekil.48. Belirli sayıdaki rakamı en büyük en küçük sayıyı oluşturacak şekilde sıralayan kurala örnek bir soru



Şekil.49. Belirli sayıdaki rakamı en büyük en küçük sayıyı oluşturacak şekilde sıralama kuralı algoritması

4.2.1.2.4 Sayıları Bölüklerine Ayırma Kuralı

Milyonlar mertebesindeki sayıları bölüklerine ayıran kuraldır.



Şekil.50. Sayıları bölüklerine ayırma kuralı algoritması

- 6. soru** 53467425 sayısının bölüklerine ayrılmış hali 5=Birler 2=Onlar 4=Yüzler 7=Binler 6=On binler 4=Yüz binler 3=Milyonlar 5=On Milyonlar dir.
- Doğru
 Yanlış

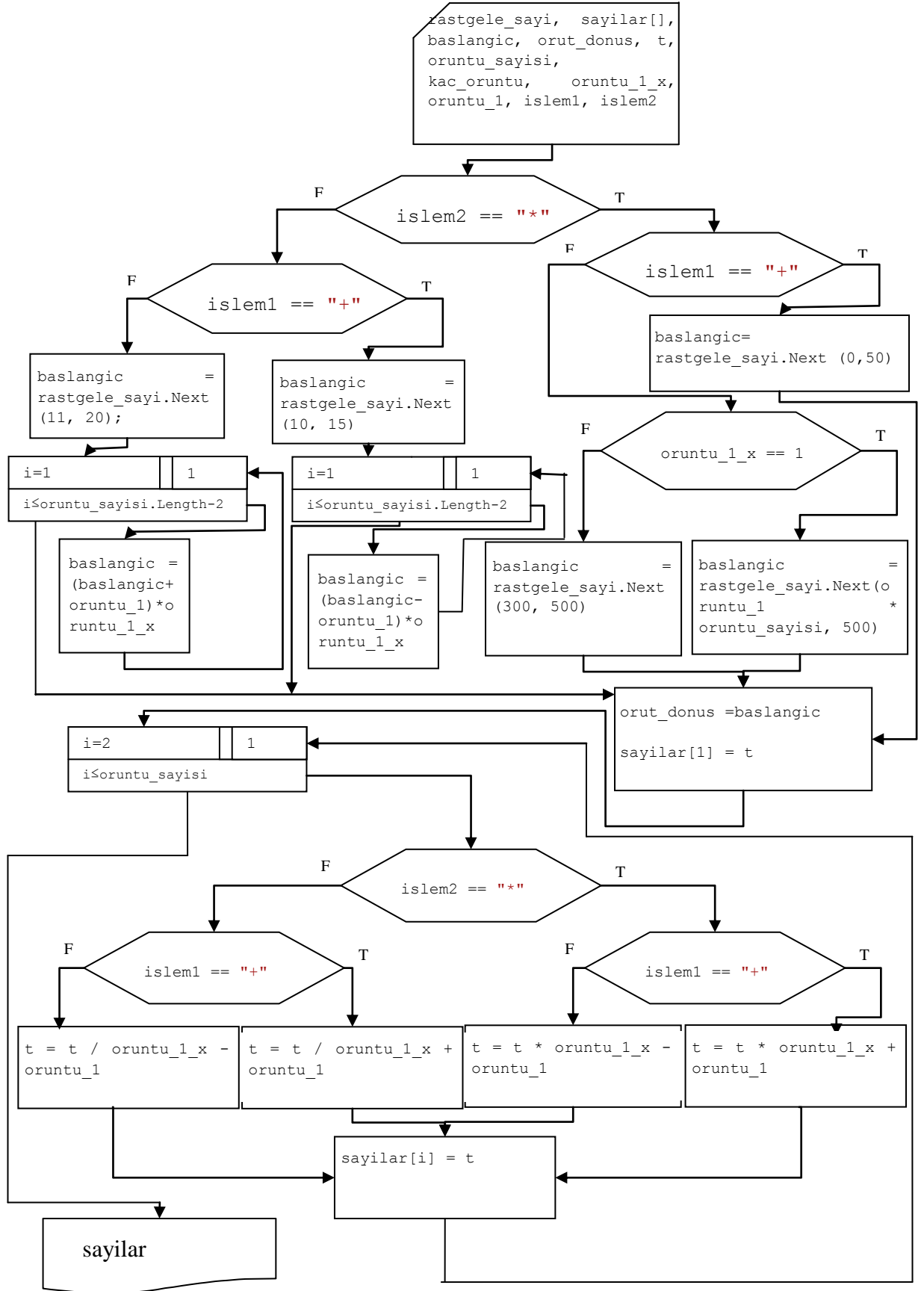
Şekil.51. Sayıları bölüklerine ayırma kuralı için örnek bir soru

4.2.1.2.5 Örüntü Oluşturma Kuralı

Sistemin belirlediği sınırlara göre örüntü oluşturan kuraldır.

- 13. soru** ? 41 ? 33 29 örüntüsündeki 1. ve 3. sıralardaki 45,37 dir
- Doğru
 Yanlış
- 14. soru** ? 41 35 ? 23 17 örüntüsünün 1. ve 4. sıralardaki sayılar dir

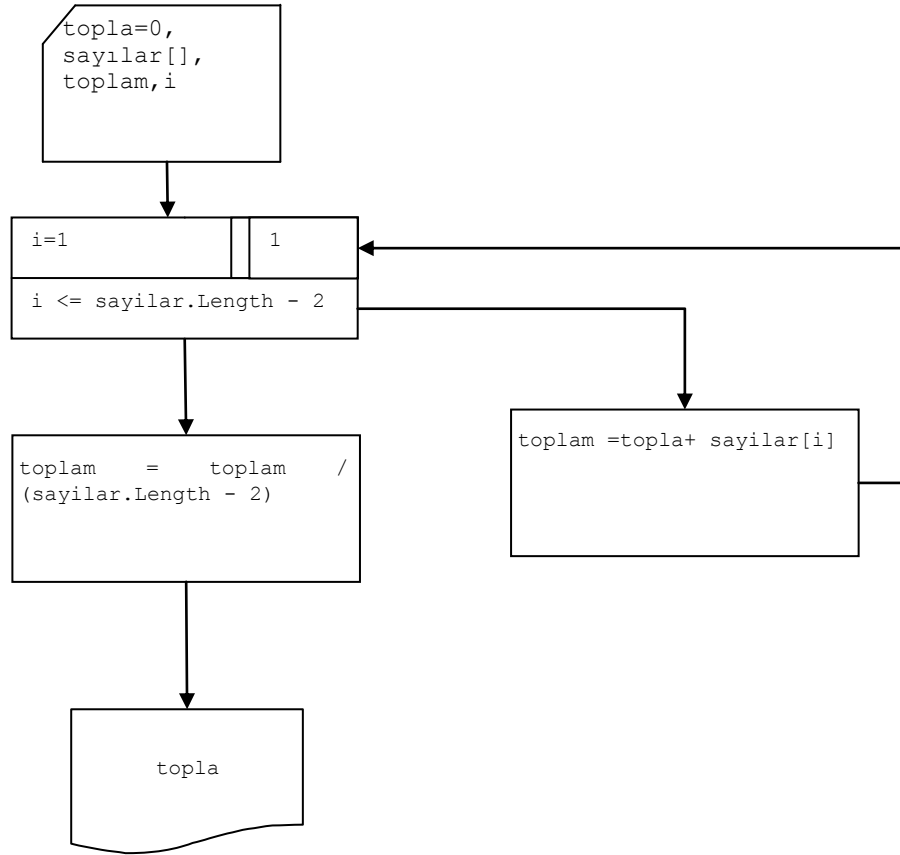
Şekil.52. Örüntü oluşturma kuralı için örnek bir soru



Şekil.53. Örüntü oluşturma kuralı algoritması

4.2.1.2.6 Ortalama Buldurma Kuralı

Sistem tarafından oluşturulan sayıların ortalamasını bulduran kuraldır.



Şekil.54. Ortalama buldurma kuralı algoritması

Bir baba ve 3 çocuğunun yaşlarının ortalaması bulunması
16. isteniyor. Küçük çocuk 3 yaşında, ortanca çocuk 5 yaşında, en büyük çocuk ise 12 yaşında. Baba 44 yaşında olduğuna göre bu ailenin yaş ortalaması nedir?

- 11
- 16
- 12
- 15
- 17

Şekil.55. Ortalama buldurma kuralı için örnek bir soru

4.2.1.2.7 Toplama Kuralı

Sistem tarafından oluşturulan sayıları toplayan kuraldır.

21. soru **84058+11410=95475 işlemi doğrudur?**

Doğru

Yanlış

22. soru **20+344=..... işleminin sonucu nedir?**

23. soru **37+313=..... işleminin sonucu nedir?**

24. soru **400+4670=? işleminin sonucu kaçtır?**

5069

5063

5070

5071

5077

25. soru **13696+30466=..... işleminin sonucu nedir?**

Şekil.56. Toplama kuralı için örnek bir soru

4.2.1.3 Görme Engelliler İçin Soru Modülü

Görme engelliler için sorular ve cevapları ses dosyası olarak öğretmenler tarafından yüklenmektedir. Öğretmenler “.wav” veya “.mp3” biçimde kaydettikleri soru, cevap ve çeldiricileri bu menüyü kullanarak sisteme yüklemektedirler. Başka biçimler sistem tarafından kabul edilmemektedir. Bu yüklenen sorularla bir soru havuzu oluşturulmaktadır. Bu havuzdan ön test, konu testi ve son test için sorular çekilip, öğrencilerin değerlendirilmesi için kullanılmaktadır.

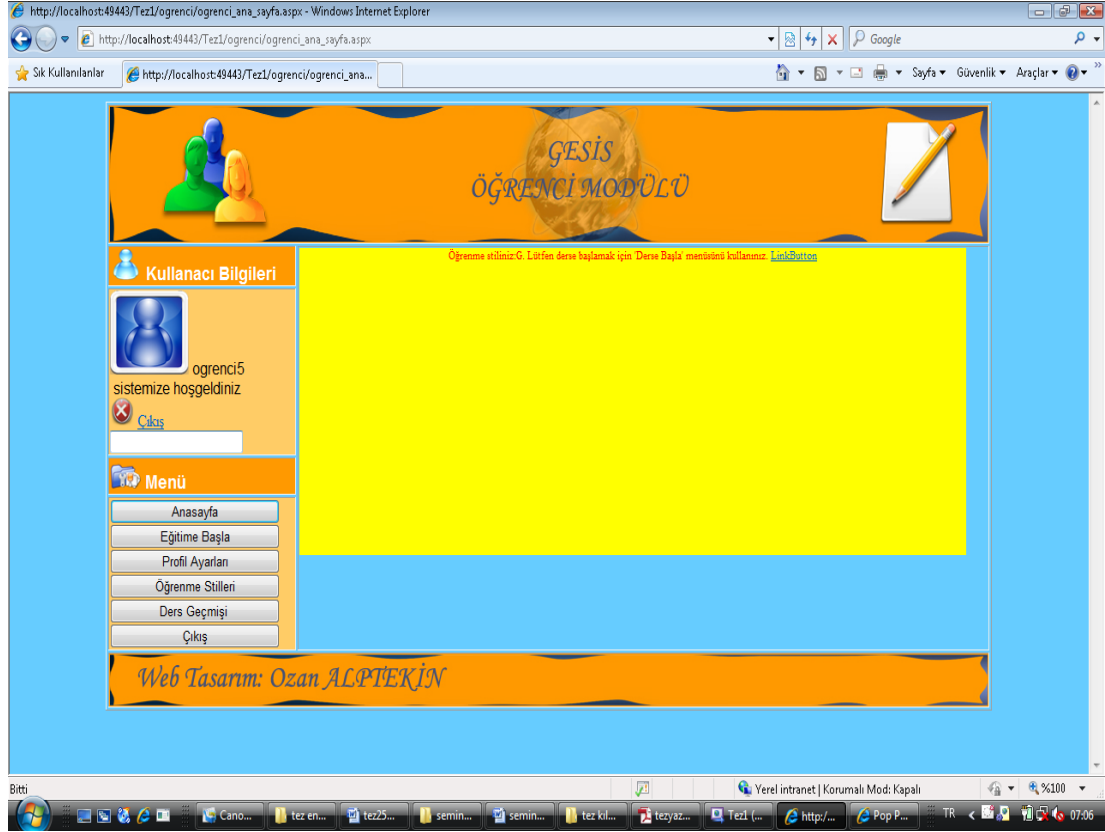
4.2.2 Kurs

Yönetici tarafından sınıflara atanan öğretmenler kendi öğretim dalları için kurs oluşturmak zorundadır. Kurslar sınıf bazında oluşturulmaktadır yani o sınıfa ait olan öğrenciler bir alan dersini alabilmeleri için öğretmen tarafından kurs oluşturulmaktadır. Bu bölümde kurslar yönetilmektedir.

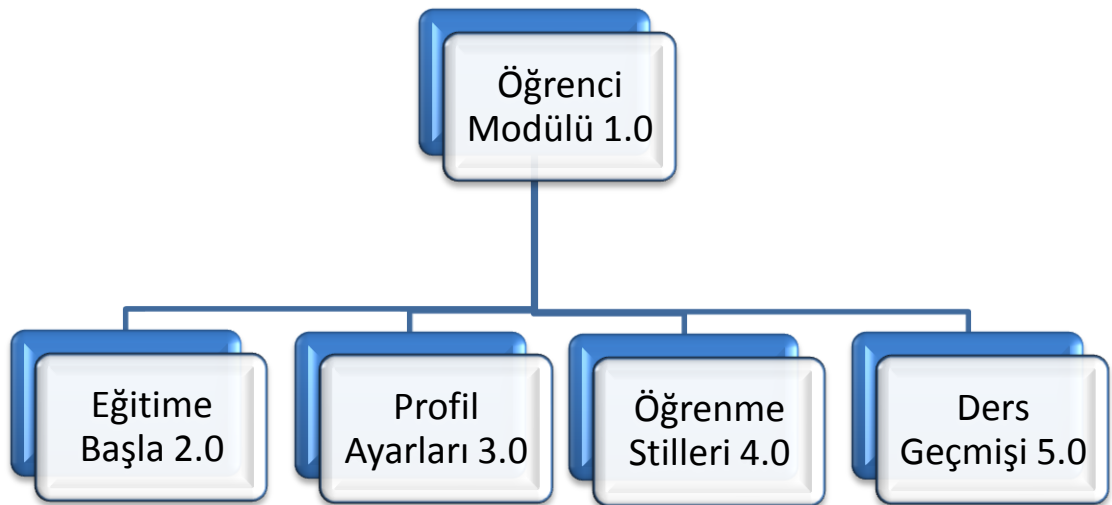
4.2.3 Raporlar

Öğrenciler ve sınıflar ile ilgili raporların kontrol edildiği menüdür. Ayrıca kursları bitiren öğrencilere sertifikalar bu bölümde oluşturulmaktadır.

4.3 Öğrenci Modülü



Şekil.57. Öğrenci modülü anasayfası



Şekil.58. Öğrenci modülünün bölümleri

Öğrencilerin kendi kullanıcı adı ve şifreleri ile girip eğitim aldıkları bölümdür. Öğrenciler sadece birkaç tıklamayla eğitime başlayabilmektedir. Hangi konuların nasıl gösterileceği, hangi soruların sorulacağı, ne zaman diğer üniteye geçebileceği gibi konulara sistemin kendisi karar vermektedir. Sistemin içinde var olan zeki etmenler vasıtasıyla sistem gerekli bütün soruları kendisi üretmektedir.

4.3.1 Eğitime Başla

Öğretmenler tarafından öğrencinin ait olduğu sınıfa açılan derslere ulaşılan menüdür. Bütün eğitim öğretim faaliyetleri bu menüdedir.

4.3.2 Profil Ayarları

Yazı rengi, yazı boyutu, yazı tipi ve arka plan rengi gibi sistemi kişiselleştirebilen ayarlar bu menüden yapılmaktadır. Özellikle görme zorluğu yaşayanlar için önemli bir menüdür. Yazıların boyutlarını ve renklerini bu menüden ayarlayarak sistemi daha verimli kullanabilmektedirler.

4.3.3 Öğrenme Stilleri

Sisteme giren öğrencinin ilk olarak yapması gereken öğrenme stilleri testi bu menüdedir. Bu test yapılmadan yani öğrencinin öğrenme stiline karar verilmeden öğretim faaliyetlerine geçilmemektedir.

4.3.4 Ders Geçmişi

Öğrencinin daha önceden gördüğü dersler ve bu derslerdeki başarı oranlarının verildiği menüdür.

4.4 Öğrenci Modülünün İşleyişi

4.4.1 Görme Zorluğu Yaşayan Ve Görme Engeli Olmayan Kullanıcılar İçin İşleyiş

Öğrenci ilk olarak sisteme kullanıcı adı ve şifresi ile giriş yapmaktadır. Eğer sisteme ilk kez giriyorsa sistem öncelikle görme durumu sayfasına yönlendirmektedir. Burada öğrenci görme durumuna göre “Görme Engelli Değil”, “Görme Zorluğu Var” ve “Görme Engelli” seçeneklerinden birini seçmek zorundadır. İlk iki seçenektan birini seçerse sistem öğrenciyi profil ayarlarına yönlendirmektedir. Öğrenci sistemin yazı tipi, boyutu, rengi ve arka plan rengini en

iyi görme durumuna göre bu sayfadan ayarlayabilmektedir. Bu ayarlar bitince sistem öğrenciyi “Öğrenci Ana Menü”ye yönlendirir. İlk girişte öğrenim stilleri menüsü dışındaki tüm menüler pasif durumdadır. Sistem öğrenciyi öğrenim stili yapamaya zorlamaktadır. Öğrenim stili belirlenmeyen öğrenciler diğer menülere ulaşamamaktadır. Öğrenme stili belirlenen öğrenci öğretmeni tarafından açılmış derslere eğitime başla menüsünü kullanarak başlayabilmektedir. Burada kendi öğrenme stiline göre tasarlanmış dersleri görebilmektedir. Ancak öncelikle her ders için ön test yapmalı ve buna göre ders programı belirlenmelidir. Sonuçta tüm dersleri görüp toplamda 7 sınavı başarıyla bitiren öğrenci dersi tamamlamış sayılır ve sertifika almaya hak kazanır.

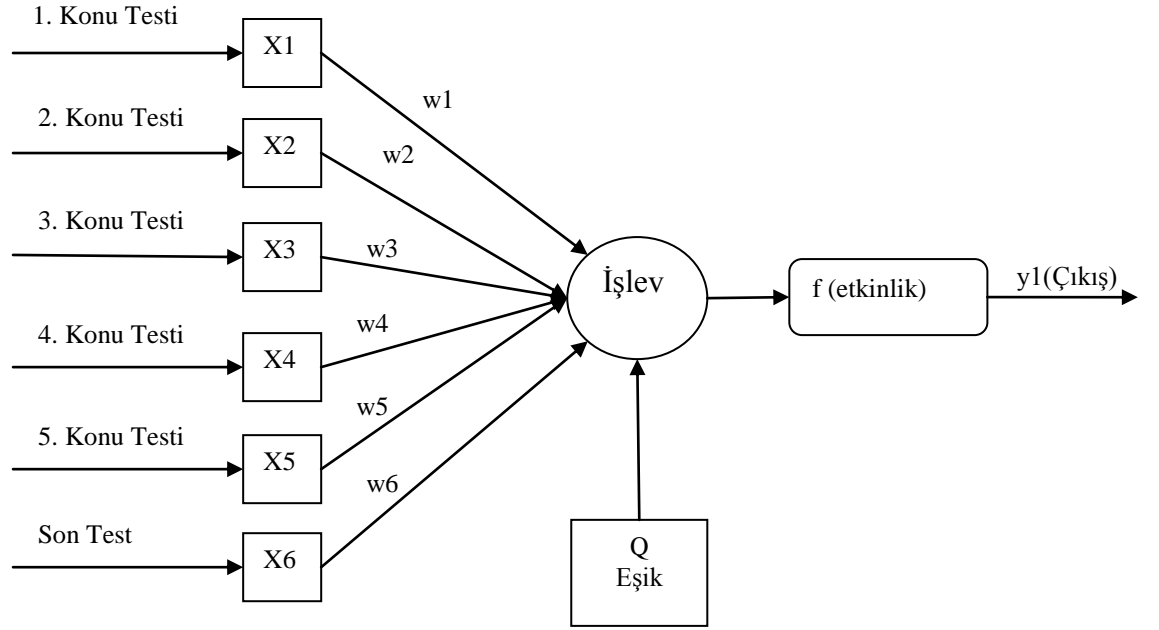
4.4.2 Görme Engelliler İçin Sistemin İşleyişi

Görme engelliler sisteme bilgi girişini fare ve klavyeden yaparken sistemden çıkışı sadece ses yoluyla almaktadır. Klavye sadece kullanıcı girişinde kullanılmaktadır. Bunun dışında sistemin tamamı üç tuşlu fare ile kontrol edilmektedir. Menüler arası geçiş sağ tuşla, o sayfadaki seçenekler arası dolaşım orta tuşla ve onaylama işlemi de sol tuşla yapılmaktadır. Görme engelli kullanıcıların diğer kullanıcılardan farkı öğrenme stilleri her zaman “İşitsel” olarak kabul edilir yani öğrenme testine girmezler, profil ayarları menüsü yoktur, girdikleri sayfalar sesle kendini tanıtır ve değerlendirme testleri sesli soru havuzdan çekilen sorularla yapılmaktadır.

4.5 Dersi Bitirme Mantığı

Programa göre her ders beş konudan oluşmaktadır. Bir ders için öncelikle ön test yapılmaktadır. Ön test’ in amacı, öğrencilerin, ders içindeki konuların bilgi bazında hazır bulunuşluk düzeylerini ölçmektir. Her konudan 5’er soru olmak üzere toplam 25 soru ile yapılan sınavdan %80 başarı sağlanan her konudan muaf olunur. Muaf olunan dersler ders programından çıkarılmaktadır. %80 başarı sağlanamayan dersler, önce konu olarak gösterilir. Daha sonra ise dersin bitimde öğrenci konu testine tabi tutulur. Eğer öğrenci, konu testinden %50 ve üzeri başarı elde ederse o konuyu tamamlamış kabul edilir. Eğer %50’den daha az bir başarı gösterilirse, konu tekrar edilmek zorundadır. En son olarak tüm konular bittiğinde son test adı verilen bir test uygulanır. Bu test’in sonucu, diğer konu testleri ile birleştirilip bir değer

bulunur. Eğer bu değer %65 ve üzeri ise ders bitirilmiş sayılır.



Şekil.59. Ders bitirme başarı hesabı

Girişler:

x1, x2, x3, x4, x5, x6

Ağırlıklar:

w1, w2, w3, w4, w5, w6

İşlev:

$$v1 = \sum_{i=1}^6 (w_i * x_i)$$

Eşik:

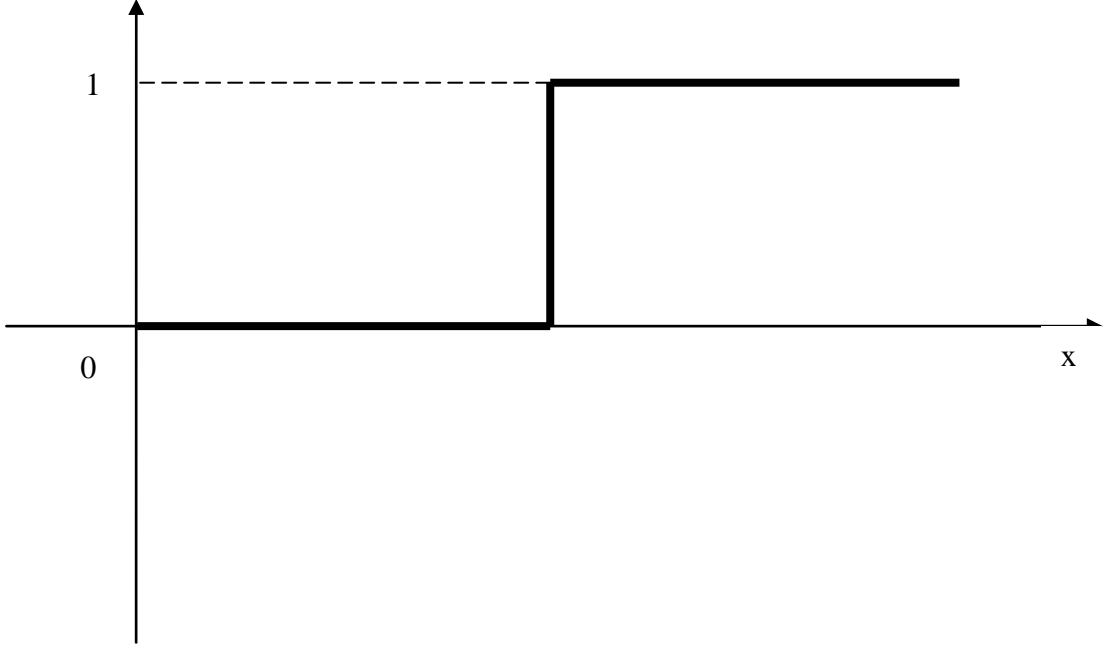
Q

Etkinlik İşlevi:

EĞER $v1 \geq Q$ İSE ÇIKIŞ 1

EĞER $v2 < Q$ İSE ÇIKIŞ 0

Çıkış:



Şekil.60. Ders bitirme çıkışı

5. SONUÇ VE DEĞERLENDİRME

Bu tez çalışması, hedeflenen üç kullanıcı tipi için de kolaylıkla kullanılabilir şekilde tasarlanmıştır. Özellikle görme engelli kullanıcılar için ara yüz oldukça sade ve hatasız tasarlanmıştır. Hiç görmeyenler için tasarlanan ara yüzün sadece fare ile kontrol edilmesi, öğrenciler için kolaylık oluşturmaktadır. Özel amaçlar için üretilen fareleri göz ardı edersek, standart farelerin sadece üç komut tuşunun olması sistemi tasarlayan kişinin seçeneklerini kısıtlasa da görme engelli bir kullanıcı için sistemi kolay kontrol edebilme açısından oldukça yararlı olmaktadır. Haliyle fare sistem açısından hayati bir donanım olmaktadır. Görme engelliler, bilgisayar kullanımında fareyi genellikle kullanmadıkları için fare kullanımında bir takım zorluk yaşayabilir ve bu sistemin zayıf yönlerinden biri olarak gözükebilir. Ancak fare kullanımının sistemde bir yere odaklanmak için değil sadece komut vermek için olduğu gözden kaçırılmamalıdır.

Sistem Őu an iin masaüstü programcılığı veya internet tabanlı olarak sunulan öğretim sistemlerinin eksik kaldığı bir noktayı tamamlamaktadır. Bu nokta, kişiye özel öğretim sistemi sunabilmedir. Eğitim bilimlerinde öğretim stilleri olarak geçen kişinin öğrenme tipinin belirlenmesi olayı öğrenmenin kişiselleştirilebilmesi ve öğrenmenin kalıcılığı açısından oldukça önemlidir. Sistem bu özelliğı belirlemekle kalmayıp kişiye özel bir öğretim sunmaktadır. Bu durum sistemi kullanan için kolaylık getirirse de, sistemi tasarlayan için fazladan yük getirmektedir çünkü her grup için fazladan bir öğretim sistemi tasarlanmaktadır. Sistem Őu an için öğretim stillerine göre yedi grup belirlese de bu sayı kolaylıkla arttırılabilmektedir. Bundan sonraki aşamalarda bu sayının arttırılması sistemin bireyselleştirilmiş öğretim güvenilirliğini arttıracağı göz ardı edilmemelidir.

Sistem üç tip test sistemine sahip olmasına rağmen görme engelliler sadece soru havuzu sisteminden yararlanabilmektedir. Bunun sebebi görme engellilerin sistemden alabileceğı tek dönütün ses olmasıdır. Soru havuzunda soruyu ekleyen öğretmen soruları ses dosyası olarak eklemektedir yani materyal hazırlama görevi öğretmendedir. Ancak soru uzmanı modülü, soruları kendisi otomatik olarak ve görsel olarak hazırlandığından görme engellilere hitap etmemektedir. Görsel materyali sese çevirme işlevi ise ya sistem dışında bir programa ya da sistem içerisinde bir modüle yüklenmelidir. Sistem dışında da bir program kullanmak sistemin kendi kendine yetebilme prensibine ters düşmektedir. Sistem içerisinde böyle bir modül yaratmak ise başlı başına bir proje olduğundan, ilerleyen aşamalarda sisteme eklenmesi düşünülmektedir.

Sistem, son aşamada kullanıcıları ait olduğu sınıfın ortalamasına göre “Geçti” veya “Kaldı” olarak değerlendirmektedir. Genel olarak, sonucunda sertifika vermeyi hedefleyen öğretimlerde bu değerlendirme yeterlidir. Ancak sistem, kişisel başarıların karşılaştırıldığı bir öğretim için kullanılacaksa, bağıl değerlendirmeye bağılı harfle notlandırma sistemine çevrilmesi değerlendirme kıstasları açısından daha doğrudur. Bağılı değerlendirme, sınıf içi dinamikleri göz önünde tutarak kişisel başarıların karşılaştırılması açısından sisteme daha uygundur.

Tasarlanan sistemi kullanmak ve geliştirmek isteyenler, öğrenim stilleri ve öğretimin bitirilmesi aşamaları geliştirilebilirler. Sistemde öğrenim stillerinin kuralları sabittir ancak kurallar dinamik olarak belirlenecek şekilde tasarlanabilir.

Böylece, sistem, bulanık mantık kurallarını öğrencilerin durumuna göre değiştirebilir. Ayrıca öğretim bitirilirken kullanılan algoritma, sistemin, öğrencileri değerlendirirken daha gerçekçi karar verebilmesi için yapay sinir ağı tekniği kullanılarak yapılabilir. Sistem gelişimlere açık ve bu beklentileri karşılayacak bir alt yapı kurma amacı ile tasarlanmıştır.

Sonuç olarak sistem özellikle görme engelliler ve görme zorluğu yaşayanlar için iyi bir alternatif olmaktadır. Ağırlığını görselliğin oluşturduğu bilişim teknolojilerinde, sesli kitap ve sayfa okuyucu programlara bağlı olan bu kullanıcılar, sistemi standart bir fare ile kullanabilmekte ve başka bir programa ihtiyaç duymamaktadır. Sistem, öğretim sistemini hazırlayan ve materyal geliştiren öğretmenler için belirli bir yük getirirse de, bu öğrenme amaçlı sistemi kullanan kullanıcıya yansımamaktadır. Sistemin, istenilen hedeflere ulaşmakta yeterli olduğu görülmekle birlikte dâhili sayfa okuyucu ve büyüteç gibi araçlarda eklendiğinde daha kullanışlı ve verimli olacağı düşünülmektedir.

KAYNAKLAR

- Baykal, N., Beyan T.(2004). *Bulanık Mantık İlke ve Temelleri*, Ankara: Bıçaklar Yayınevi
- Baykal, N., Beyan T.(2004). *Bulanık Mantık Uzman Sistemler ve Denetleyiciler*, Ankara: Bıçaklar Yayınevi
- Elmas, Ç.(2007). *Yapay Zekâ Uygulamaları* (1. Baskı). İstanbul: Seçkin Yayıncılık.
- Gözütok, F. D. *Öğretim İlke ve Yöntemleri* (2. Baskı) Ankara: Ekinos Kitapevi
- Kalıpsız, O., Buharalı, A. ve Biricik, G. (2008). *Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme* (2. Baskı). İstanbul: Papatya Yayıncılık.
- Kaya, Y., Tekin, R.(2007). *Veritabanı ve Uygulamaları* (1. Baskı). İstanbul : Papatya Yayıncılık Eğitim
- Milli Eğitim Bakanlığı. (2003). *Eğitim Teknolojileri Bilgisayar* (1. Baskı). Ankara: MEB Eğitim teknolojileri Genel Müdürlüğü
- Milli Eğitim Bakanlığı. 2009. Milli Eğitim Bakanlığı 2010-2014 Stratejik Planı.
- Nabiyev, V. V.(2005). *Yapay Zekâ Problemler-Yöntemler-Algoritmalar* (2. Baskı). Ankara: Seçkin Yayıncılık.
- Nwana, S. H.(1990). *Intelligent Tutoring Systems: an Overview*. Kluwer Academic Publishers.
- Öğün, V., Pektaş, Y. ve Serfiçeli, Z. (2009). *İlköğretim Matematik Öğretmen Kılavuz Kitabı*(1. Baskı). Ankara: Öğün Yayınları.
- Özbek, M. (2007). *Etmen Tabanlı Zeki Öğretim Sistemi Geliştirme*, Yayınlanmış Yüksek Lisans Tezi
- Özgüven, İ. E.(2003). *Psikolojik Testler* (1. Baskı). Ankara: Pdrem Yayınları.
- Senemoğlu, N.(1997), *Gelişim Öğrenme ve Öğretim Kuramdan Uygulamaya*, Ankara: Özsen Matbaası
- Sönmez, V. (2008). *Öğretim İlke ve Yöntemleri* (2. Baskı). Ankara: Anı Yayıncılık
- Sönmez, V.(2007). *Program Geliştirmede Öğretmen Elkitabı* (13. Baskı) Ankara: Anı Yayıncılık

Yeşilyaprak, B.(ed.)(2003). *Gelişim ve Öğrenme Psikolojisi* (5. Baskı). Ankara: Pegem A Yayıncılık.

EK SİSTEMDEKİ BAZI ÖNEMLİ KODLAR

1. Normalizasyon İşlemi

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

public class max_min_normalizasyon1
{
    public double max_min_normalizasyon(double eski_sayi,double
yeni_min,double yeni_max, double eski_min,double eski_max)
    {
        double v1;
        v1 = ((eski_sayi - eski_min) / (eski_max - eski_min) * (yeni_max - yeni_min))
+ yeni_min;
        return Math.Round(v1);
    }
}
```

2. Bulanıklaştırma İşlemi

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

public class bulanik_kume1
{
    public struct bulanik_kume
    {
        public int[] kume;
        public double[] kume_uyelik_degeri;
        public bulanik_kume(int kume_sayi, int kume_uyelik_sayi)
        {
            kume = new int[3];
            kume_uyelik_degeri=new double[3];
        }
    }

    public bulanik_kume bulaniklastir(int i,double sayi,int min, int middle, int max )
    {
        bulanik_kume donus=new bulanik_kume(3,3);
        if (sayi < 50)
        {
```

```

        donus.kume[1] = 0;
        donus.kume[2] = 1;
        donus.kume_uyelik_degeri[1] = (middle - sayi) / (middle - min);
        donus.kume_uyelik_degeri[2] = (min - sayi) / (min - middle);
    }
    else if (sayi > 50)
    {
        donus.kume[1] = 1;
        donus.kume[2] = 2;
        donus.kume_uyelik_degeri[1] = (max - sayi) / (max - middle);
        donus.kume_uyelik_degeri[2] = (middle - sayi) / (middle - max);
    }
    else { donus.kume[1] = 1; donus.kume_uyelik_degeri[1] = 1; donus.kume[2] =
0; donus.kume_uyelik_degeri[2] = 0; }
    return donus;
}
}

```

3. Sonuçlandırma ve Durulama

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for kural_tabani
/// </summary>
public class kural_tabani1
{
    public string kural_tabani(bulanik_kume1.bulanik_kume[] stiller)
    {
        double enb = 0, tmp = 0;
        int t=1, tmp1;
        string kume=null, kume1=null;
        for (int i = 1; i <= 2; i++)
        {
            for (int j = 1; j <= 2; j++)
            {
                for (int k = 1; k <= 2; k++)
                {
                    tmp1=stiller[1].kume[i] *1+ stiller[1].kume[j] *3+ stiller[1].kume[k]*9;
                    if (tmp1 % 13 == 0) kume = "HC";
                    else if(tmp1==1||tmp1==2||tmp1==5||tmp1==11||tmp1==14) kume="G";
                    else if (tmp1 == 3 || tmp1 == 6 || tmp1 == 7 || tmp1 == 15 || tmp1 == 16)
kume = "I";
                    else if (tmp1 == 9 || tmp1 == 18 || tmp1 == 19 || tmp1 == 21 || tmp1 ==
22) kume = "D";

```

```

        else if (tmp1 == 4 || tmp1 == 8 || tmp1 == 17) kume = "GI";
        else if (tmp1 == 12 || tmp1 == 24 || tmp1 == 25 ) kume = "ID";
        else if (tmp1 == 10 || tmp1 == 20 || tmp1 == 23) kume = "GD";
        tmp = stiller[1].kume_uyelik_degeri[i] *
stiller[2].kume_uyelik_degeri[j] * stiller[3].kume_uyelik_degeri[k];

        if (tmp > enb) { enb = tmp; kume1 = kume; t++; }
    }
}
return kume1;
}
}

```

4. Aritmetik Ortalama Buldurma İşlemi

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for aritmektiksel_ortalama
/// </summary>
public class aritmektiksel_ortalama
{
    public string aritmektiksel_ortalama_bul(string[] sayilar)
    {
        int toplam = 0;
        for (int i = 1; i <= sayilar.Length - 2; i++)
            toplam += Convert.ToInt32(sayilar[i]);
        toplam = toplam / (sayilar.Length - 2);
        return Convert.ToString(toplam);
    }

    public int[] karakterleri_ver(string aranan, string cumle)
    {
        int i = 0, k = 0;
        int[] a = new int[10];
        do
        {
            i++;
            a[i] = cumle.IndexOf(aranan, k);
            k = a[i] + 1;
        }
        while (a[i] != -1);
        Array.Resize(ref a, i);
        return a;
    }
}

```

```

public int[,] aradegerler(string cumle, int[] baslangic, int[] son)
{
    int[,] sonuc = new int[baslangic.Length, 4];
    string[] tmp1 = new string[2];
    int j = 0;
    string tmp;
    for (int i = 1; i <= baslangic.Length - 1; i++)
    {
        tmp = cumle.Substring(baslangic[i] + 1, son[i] - baslangic[i] - 1);
        tmp1 = tmp.Split('-');
        foreach (string k in tmp1)
        {
            j++;
            sonuc[i, j] = Convert.ToInt32(k);
            if (j == 2) j = 0;
        }
    }
    return sonuc;
}

public string yerine_yerlestir(int[] baslangic, int[] son, string cumle, string[]
sayilar)
{
    string tmp = cumle;
    int t = 0;
    for (int i = 1; i <= baslangic.Length - 1; i++)
    {
        tmp = tmp.Remove(baslangic[i] - t, 9);
        tmp = tmp.Insert(baslangic[i] - t, sayilar[i]);
        t = t + 9 - sayilar[i].Length;
    }
    return tmp;
}

public string[] celdirici(int dogru_cevap)
{
    Random rastgele_sayi = new Random();
    int cel_sayi = 0; int i = 1;
    string[] celdiriciler = new string[5];
    string[] islem = { "+", "-" };
    if (dogru_cevap <= 10)
    {
        while (i <= 4)
        {
            cel_sayi = rastgele_sayi.Next(1, 10);
            if (cel_sayi != dogru_cevap && kontrol(cel_sayi, i, celdiriciler) == 1)
            {
                celdiriciler[i] = Convert.ToString(cel_sayi);
                i++;
            }
        }
    }
}

```

```

    }
}
else if (dogru_cevap > 10)
{
    while (i <= 4)
    {
        cel_sayi = rastgele_sayi.Next(1, 8);
        if (rastgele_sayi.Next(0, 2) == 0) cel_sayi = dogru_cevap - cel_sayi;
        else cel_sayi = dogru_cevap + cel_sayi;
        if (cel_sayi != dogru_cevap && kontrol(cel_sayi, i, celdiriciler) == 1)
        {
            celdiriciler[i] = Convert.ToString(cel_sayi);
            i++;
        }
    }
}

}
return celdiriciler;
}
int kontrol(int deger, int i, string[] celdiriciler)
{
    int t = 1;
    for (int j = 1; j < i; j++)
    {
        if (celdiriciler[j] == Convert.ToString(deger)) t = 0;
    }
    return t;
}
}
}

```

5. Basamaklara Ayırma

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for basamaklara_ayir
/// </summary>
public class basamaklara_ayir
{
    string topla;
    public string basamaklara_ayir1(int uretilmis_sayi, int y)
    {
        string[] basamak = { "", "Birler", "Onlar", "Yüzler", "Binler", "Onbinler",
"Yüzbinler", "Milyonlar", "On Milyonlar", "Yüz Milyonlar" };
    }
}

```

```

    for (int i = 1; i <= y; i++)
    {
        toplu = toplu + Convert.ToString(uretilmis_sayi)[y - i] + "=" + basamak[i];
    }
    return toplu;
}
public string[] celdirici(string uretilmis_cevap)
{
    Random rastgele=new Random();
    string[] celdiriciler=new string[5];
    string[] basamak = { "", "Birlere", "Onlar", "Yüzler", "Binler", "Onbinler",
"Yüzbinler", "Milyonlar", "On Milyonlar", "Yüz Milyonlar" };
    int i=1;
    string sec1, sec2, tmp;
    while (i <= 4)
    {
        sec1 = basamak[rastgele.Next(1, 10)];
        sec2 = basamak[rastgele.Next(1, 10)];
        if (sec1 != sec2)
        {
            tmp=uretilmis_cevap.Replace(sec1, "x");
            tmp = tmp.Replace(sec2, sec1);
            tmp = tmp.Replace("x", sec2);
            if (tmp != uretilmis_cevap)
            {
                int x = 1;
                for (int j = 1; j <= i; j++)
                {
                    if (celdiriciler[j] == tmp) x = 0;
                }
                if (x == 1) { celdiriciler[i] = tmp; i++; }
            }
        }
    }
    return celdiriciler;
}
}
}

```

6. En Büyüğü En Küçüğü Buldurma

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for en_buyuk_en_kucuk

```

```

/// </summary>
public class en_buyuk_en_kucuk
{
    int kk = 0, jj = 0, tmp = 0; string olusmus_sayi;

    public string sayi_olustur(Boolean tek_cift, Boolean kucuk_buyuk, int j, int k,
int[] sayilartek, int[] sayilarcift, int sifirsayisi)
    {
        int[] c = new int[j + k];
        // Küçük sayı (tek veya çift) oluşturur
        if (kucuk_buyuk == false)
        {
            if (tek_cift == false) { c[0] = sayilartek[0]; kk = 1; jj = 0; }
            else { c[0] = sayilarcift[0]; kk = 0; jj = 1; }
            sayilarcift[j] = sayilartek[k - 1];
            sayilartek[k] = sayilarcift[j - 1];

            for (int i = 1; i < (j + k); i++)
            {
                if (sayilartek[kk] > sayilarcift[jj]) { c[i] = sayilartek[kk]; kk = kk + 1; }
                else { c[i] = sayilarcift[jj]; jj = jj + 1; }
            }
            Array.Reverse(c);
            if (sifirsayisi != 0) { tmp = c[sifirsayisi]; c[0] = tmp; c[sifirsayisi] = 0; }
            for (int i = 0; i < j + k; i++)
            {
                olusmus_sayi = olusmus_sayi + c[i];
            }
        }
        // Büyük sayı (tek veya çift) oluşturur
        else
        {
            if (tek_cift == false) { c[j + k - 1] = sayilartek[k - 1]; sayilartek[k - 1] =
sayilarcift[j - 1]; }
            else { c[j + k - 1] = sayilarcift[j - 1]; sayilarcift[j - 1] = sayilartek[k - 1]; }

            for (int i = 0; i < (j + k - 1); i++)
            {
                if (sayilartek[kk] > sayilarcift[jj]) { c[i] = sayilartek[kk]; kk = kk + 1; }
                else { c[i] = sayilarcift[jj]; jj = jj + 1; }
            }

            for (int i = 0; i < j + k; i++)
            {
                olusmus_sayi = olusmus_sayi + c[i];
            }
        }
        return olusmus_sayi;
    }
}

```



```

public string[] celdirici(string dogru_cevap)
{
    Random random_sayi=new Random();
    string[] celdiriciler = new string[5];
    int i = 1,a,b;
    char tmp,tmp1;
    string celdir;
    while (i <= 4)
    {
        celdir = dogru_cevap;
        a = random_sayi.Next(0, dogru_cevap.Length);
        do
        {
            b = random_sayi.Next(0, dogru_cevap.Length);
        }
        while (a == b);
        tmp = celdir[a];tmp1=celdir[b];
        celdir = celdir.Replace(celdir[a], 'A');
        celdir = celdir.Replace(celdir[b], 'B');

        celdir=celdir.Replace('A',tmp1);
        celdir=celdir.Replace('B', tmp);
        if (celdir != dogru_cevap && Array.BinarySearch(celdiriciler, celdir) < 0) {
celdiriciler[i] = celdir; i++; }
    }
    return celdiriciler;
}
}

```

7. Örüntü Oluşturma

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for oruntu_olustur
/// </summary>
public class orut_olustur
{
    Random random_sayi = new Random();

    public int[] orut(int oruntu_sayisi, int kac_oruntu, int oruntu_1_x, int oruntu_1,
string islem1, string islem2)
    {
        Random rastgele_sayi=new Random();
        int[] sayilar = new int[oruntu_sayisi + 1];
        int baslangic;
        string orut_donus=null;
        if (islem2 == "")

```

```

{
    if (islem1 == "+")
        baslangic = rastgele_sayi.Next(0,50);

    else
    {
        if (oruntu_1_x == 1)
            baslangic = rastgele_sayi.Next(oruntu_1 * oruntu_sayisi, 500);
        else
            baslangic = rastgele_sayi.Next(300, 500);
    }
}
else
{
    if (islem1 == "+")
    {
        baslangic = rastgele_sayi.Next(10, 15);
        for (int i = 1; i <= oruntu_sayisi; i++)
            baslangic = (baslangic-oruntu_1)*oruntu_1_x;
    }
    else
    {
        baslangic = rastgele_sayi.Next(11, 20);
        for (int i = 1; i <= oruntu_sayisi; i++)
            baslangic = (baslangic + oruntu_1) * oruntu_1_x;
    }
}
urut_donus =Convert.ToString(baslangic);
int t=baslangic;
sayilar[1] = t;
for (int i = 2; i <= oruntu_sayisi; i++)
{
    if (islem2 == "*")
    {
        if (islem1 == "+")
            t = t * oruntu_1_x + oruntu_1;
        else
            t = t * oruntu_1_x - oruntu_1;
    }
    else
    {
        if (islem1 == "+")
            t = t / oruntu_1_x + oruntu_1;
        else
            t = t / oruntu_1_x - oruntu_1;
    }
    sayilar[i] = t;
}
return sayilar;
}

```

```

public Uzman_Sistem.oruntu_struct soru_hazirla(string cumle, int[] sayilar)
{
    Uzman_Sistem.oruntu_struct temp=new Uzman_Sistem.oruntu_struct(3);
    temp.cikan_sayilar[1] = random_sayi.Next(1, sayilar.Length-1);
    temp.cikan_sayilar[2] = -1;
    do
    {
        temp.cikan_sayilar[2] = random_sayi.Next(1, sayilar.Length -1);
    }
    while (temp.cikan_sayilar[1] == temp.cikan_sayilar[2]);
    for (int i = 1; i < sayilar.Length; i++)
    {
        if (i != temp.cikan_sayilar[1] && i != temp.cikan_sayilar[2])
            temp.soru += sayilar[i] + " ";
        else
            temp.soru += "? " + " ";
    }
    temp.soru = cumle.Replace("<1>", temp.soru);
    temp.soru = temp.soru.Replace("<2>", temp.cikan_sayilar[1] + ". ve " +
temp.cikan_sayilar[2] + ". sıralardaki");
    return temp;
}
public string[] celdiriciler(string dogru_yanit)
{
    string[] celdiri=new string[5];
    int i=1;
    int islem_random = 0, sayi_random = 0;
    string[] sayilar=dogru_yanit.Split(',');
    while (i <= 4)
    {
        islem_random = random_sayi.Next(0, 2);
        sayi_random = random_sayi.Next(0, 2);
        if (islem_random == 0)
            celdiri[i]
=Convert.ToString(random_sayi.Next(Convert.ToInt32(sayilar[sayi_random])+1,Co
nvert.ToInt32(sayilar[sayi_random])+10));
        else
            celdiri[i] =Convert.ToString(
random_sayi.Next(Convert.ToInt32(sayilar[sayi_random])-
10,Convert.ToInt32(sayilar[sayi_random])));

        if (Convert.ToInt32(celdiri[i]) > 0)
        {
            celdiri[i] = celdiri[i] + "," + sayilar[random_sayi.Next(0, 2)];
            if (celdiri[i] != dogru_yanit) i++;
        }
    }

    return celdiri;
}

```

```
}
```

8. Sayıyı Harfe Çevirme

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

/// <summary>
/// Summary description for sayi_harfe_cevirme
/// </summary>
public class sayi_harfe_cevirme
{
    public string sayioku(int y)
    {
        string okunmus_sayi;
        okunmus_sayi=ozan(y / 1000000) + " milyon " + ozan((y % 1000000) / 1000) +
" bin " + ozan(y % 1000);
        return okunmus_sayi;
    }
    string ozan(int sayi1)
    {
        string[] okunus_birler = { "", "bir", "iki", "üç", "dört", "beş", "altı", "yedi",
"sekiz", "dokuz" };
        string[] okunus_onluk = { "", "on", "yirmi", "otuz", "kırk", "elli", "altmış",
"yetmiş", "seksen", "doksan" };
        string yuzluk = null;
        int sayi_yuzler = sayi1 / 100;
        int sayi_onlar = (sayi1 % 100) / 10;
        int sayi_birler = (sayi1 % 100) % 10;
        if (sayi_yuzler != 0) { if (sayi_yuzler != 1) yuzluk = okunus_birler[sayi_yuzler]
+ " yüz "; else yuzluk = "yüz "; }
        if (sayi_onlar != 0) yuzluk = yuzluk + okunus_onluk[sayi_onlar] + " ";
        if (sayi_birler != 0) yuzluk = yuzluk + okunus_birler[sayi_birler];
        return (yuzluk);
    }

    string tmp;
    Random rastgele = new Random();
    public string[] celdirici(int sayi, string okunusu)
    {
        for (int i = 1; i <= 10; i++)
        {
        }
    }

    string[] celdiriciler = new string[5];
    //Birinci tür celdirici
```

```

    if (Convert.ToString(sayi).Length == 7) { celdiriciler[1] =
sayioku(Convert.ToInt32(Convert.ToString(sayi).Insert(7, "0"))); }
    else if (Convert.ToString(sayi).Length == 9) { celdiriciler[1] =
sayioku(Convert.ToInt32(Convert.ToString(sayi).Remove(6, 1))); }
    else { celdiriciler[1] =
sayioku(Convert.ToInt32(Convert.ToString(sayi).Remove(6, 1))); }
    //İkinci Tür çeldirci
    if (rastgele.Next(0, 2) == 0)
    {
        celdiriciler[2] = okunusu.Replace("milyon", "milyon1");
        celdiriciler[2] = celdiriciler[2].Replace("bin", "milyon");
        celdiriciler[2] = celdiriciler[2].Replace("milyon1", "bin");
    }
    else
    {
        celdiriciler[2] = celdiriciler[1].Replace("milyon", "milyon1");
        celdiriciler[2] = celdiriciler[2].Replace("bin", "milyon");
        celdiriciler[2] = celdiriciler[2].Replace("milyon1", "bin");
    }
    //Üçüncü tür çeldirci
    if (rastgele.Next(0, 2)==0)
    {
        if (rastgele.Next(0, 2) == 0)
            celdiriciler[3] = okunusu.Replace("milyon", "bin");
        else
            celdiriciler[3] = okunusu.Replace("bin", "milyon");
    }
    else
    {
        if (rastgele.Next(0, 2) == 0)
            celdiriciler[3] = celdiriciler[1].Replace("milyon", "bin");
        else
            celdiriciler[3] = celdiriciler[1].Replace("bin", "milyon");
    }
    //Dördüncü tür çeldirci
    if (rastgele.Next(0, 2) == 0)
    {
        tmp = okunusu;
        tmp = tmp.Replace("milyon", ""); tmp = tmp.Replace("bin", "");
        celdiriciler[4] = tmp;
    }
    else
    {
        tmp = okunusu;
        tmp = tmp.Replace("milyon", ""); tmp = tmp.Replace("bin", "");
        celdiriciler[4] = tmp;
    }
}

return celdiriciler;

```

```

    }
    public string[] celdirici2(int sayi)
    {
        int tmp;
        string [] celdiriciler = new string[10];
        //Birinci çeldiriciler
        if (Convert.ToString(sayi).Length == 7) { celdiriciler[1]
=Convert.ToString(Convert.ToInt32(Convert.ToString(sayi).Insert(7, "0"))-
rastgele.Next(1,10)); }
        else if (Convert.ToString(sayi).Length == 9) {tmp=rastgele.Next(1,3);
celdiriciler[1] =Convert.ToString(
Convert.ToInt32(Convert.ToString(sayi).Remove((5+tmp),(8-
(5+tmp))))+rastgele.Next(1,10)); }
        else { celdiriciler[1]
=Convert.ToString(Convert.ToInt32(Convert.ToString(sayi).Remove(6,
1))+rastgele.Next(1,100)); }
        //İkinci Çeldiriciler
        if (rastgele.Next(1,3)==1)
            celdiriciler[2] = Convert.ToString(sayi / 10);
        else
            celdiriciler[2] = Convert.ToString(sayi / 100);
        //Üçüncü Çeldirici
        if (rastgele.Next(1, 3) == 1)
            celdiriciler[3] = Convert.ToString(sayi * 10);
        else
            celdiriciler[3] = Convert.ToString(sayi * 100);
        //Dördüncü Çeldirici
        celdiriciler[4] = Convert.ToString(sayi+rastgele.Next(11,20));
        return celdiriciler;
    }
}

```

9. Toplama

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for toplam
/// </summary>
public class toplam
{
    public string toplami_bul(string[] sayilar)
    {
        int toplam = 0;
        for (int i = 1; i <= sayilar.Length - 2; i++)
            toplam += Convert.ToInt32(sayilar[i]);

        return Convert.ToString(toplam);
    }
}

```

```

}
public int[] karakterleri_ver(string aranan, string cumle)
{
    int i = 0, k = 0;
    int[] a = new int[10];
    do
    {
        i++;
        a[i] = cumle.IndexOf(aranan, k);
        k = a[i] + 1;

    }
    while (a[i] != -1);
    Array.Resize(ref a, i);
    return a;
}

public int[,] aradegerler(string cumle, int[] baslangic, int[] son)
{
    int[,] sonuc = new int[baslangic.Length, 4];
    string[] tmp1 = new string[2];
    int j = 0;
    string tmp;
    for (int i = 1; i <= baslangic.Length - 1; i++)
    {
        tmp = cumle.Substring(baslangic[i] + 1, son[i] - baslangic[i] - 1);
        tmp1 = tmp.Split('-');
        foreach (string k in tmp1)
        {
            j++;
            sonuc[i, j] = Convert.ToInt32(k);
            if (j == 2) j = 0;
        }
    }
    return sonuc;
}

public string yerine_yerlestir(int[] baslangic, int[] son, string cumle, string[]
sayilar)
{
    string tmp = cumle;
    int t = 0;
    for (int i = 1; i <= baslangic.Length - 1; i++)
    {
        tmp = tmp.Remove(baslangic[i] - t, 13);
        tmp = tmp.Insert(baslangic[i] - t, sayilar[i]);
        t = t + 13 - sayilar[i].Length;
    }
    return tmp;
}

```

```

public string[] celdirici(int dogru_cevap)
{
    Random rastgele_sayi = new Random();
    int cel_sayi = 0; int i = 1;
    string[] celdiriciler = new string[5];
    string[] islem = { "+", "-" };
    if (dogru_cevap <= 10)
    {
        while (i <= 4)
        {
            cel_sayi = rastgele_sayi.Next(1, 10);
            if (cel_sayi != dogru_cevap && kontrol(cel_sayi, i, celdiriciler) == 1)
            {
                celdiriciler[i] = Convert.ToString(cel_sayi);
                i++;
            }
        }
    }
    else if (dogru_cevap > 10)
    {
        while (i <= 4)
        {
            cel_sayi = rastgele_sayi.Next(1, 8);
            if (rastgele_sayi.Next(0, 2) == 0) cel_sayi = dogru_cevap - cel_sayi;
            else cel_sayi = dogru_cevap + cel_sayi;
            if (cel_sayi != dogru_cevap && kontrol(cel_sayi, i, celdiriciler) == 1)
            {
                celdiriciler[i] = Convert.ToString(cel_sayi);
                i++;
            }
        }
    }
    return celdiriciler;
}
int kontrol(int deger, int i, string[] celdiriciler)
{
    int t = 1;
    for (int j = 1; j < i; j++)
    {
        if (celdiriciler[j] == Convert.ToString(deger)) t = 0;
    }
    return t;
}
}

```


10. Öğrenme Stili Testi

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

public partial class ogrenci_ogrenme_stili : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Button2.Enabled = false; Button3.Enabled = false; Button5.Enabled = false;
        Button6.Enabled = false; Button7.Enabled = false;
        gorme_engeli.Value = Convert.ToString(Profile.gorme_engelli);
        ((TextBox)Master.FindControl("textbox")).Focus();
        bilgileri_getir();
        for (int i = 0; i < soru_sayisi; i++)
            tabloyu_olustur(i);
        ((Panel)Master.FindControl("Panel1")).Height =
        Convert.ToInt32(Table1.Height.Value);
        Response.Write(((Panel)Master.FindControl("Panel1")).Height+"<br>");
        Response.Write(Convert.ToInt32(Table1.Height.Value));
    }

    protected void tabloyu_olustur(int ii)
    {
        TableRow satir = new TableRow();
        TableCell hucre = new TableCell();
        TableCell hucre1 = new TableCell();
        TableCell hucre2 = new TableCell();

        hucre.Text = (ii + 1) + ". soru :";
        satir.Cells.Add(hucre);

        hucre1.Text = sorular[ii];
        satir.Cells.Add(hucre1);

        RadioButtonList rbl = new RadioButtonList();
        ListItem ogrenme_stili1 = new ListItem();
        ListItem ogrenme_stili2 = new ListItem();
        ListItem ogrenme_stili3 = new ListItem();
        ListItem ogrenme_stili4 = new ListItem();
        ListItem ogrenme_stili5 = new ListItem();
```

```

rbl.RepeatDirection = 0;
ogrenme_stili1.Text = secenekler[0];
ogrenme_stili2.Text = secenekler[1];
ogrenme_stili1.Selected = true;

ogrenme_stili1.Value = "0";
ogrenme_stili2.Value = "1";
rbl.Items.Add(ogrenme_stili1);
rbl.Items.Add(ogrenme_stili2);

if (secenek_sayisi == 3)
{
    ogrenme_stili3.Text = secenekler[2];
    ogrenme_stili3.Value = "2";
    rbl.Items.Add(ogrenme_stili3);
}
else if (secenek_sayisi == 5)
{
    ogrenme_stili3.Text = secenekler[2];
    ogrenme_stili4.Text = secenekler[3];
    ogrenme_stili5.Text = secenekler[4];
    ogrenme_stili3.Value = "2";
    ogrenme_stili4.Value = "3";
    ogrenme_stili5.Value = "4";
    rbl.Items.Add(ogrenme_stili3);
    rbl.Items.Add(ogrenme_stili4);
    rbl.Items.Add(ogrenme_stili5);
}
hucre2.Controls.Add(rbl);
satir.Cells.Add(hucre2);

Table1.BorderWidth = 1;
Table1.GridLines = System.Web.UI.WebControls.GridLines.Both;
Table1.Rows.Add(satir);
}
int soru_sayisi;
int secenek_sayisi;
string[] sorunun_stili = new string[41];
string[] sorular = new string[41];
string[] secenekler = new string[5];
protected void bilgileri_getir()
{
    SqlConnection sql_baglanti;
    sql_baglanti = new
SqlConnection("server=OZAN\\SQLEXPRESS;database=TEZ;uid=sa;
password=sa");
    try
    {
        char[] bol = new char[1] { 'é' };
        sql_baglanti.Open();

```

```

        SqlCommand komut = new SqlCommand("SELECT * FROM
OGRENME_STILI WHERE AKTIF='True'", sql_baglanti);
        SqlDataReader bilgiler = komut.ExecuteReader();
        bilgiler.Read();
        soru_sayisi = Convert.ToInt32(bilgiler["SORU_SAYISI"]);
        secenek_sayisi = Convert.ToInt32(bilgiler["SECENEK_SAYISI"]);
        secenekler = bilgiler["SECENEKLER"].ToString().Split(bol,
StringSplitOptions.None);
        sorunun_stili = bilgiler["SORU_STIL"].ToString().Split(bol,
StringSplitOptions.None);
        sorular = bilgiler["SORULAR"].ToString().Split(bol,
StringSplitOptions.None);
    }
    catch (Exception hata)
    {
        bilgi.Text = hata.Message.ToString();
    }
    finally
    {
        if (Convert.ToBoolean(sql_baglanti.State) == true) sql_baglanti.Close();
    }
}
protected void kaydet(string ogrenme_stili)
{
    MembershipUser kullanıcı; kullanıcı = Membership.GetUser();
    SqlConnection sql_baglanti;
    sql_baglanti = new
SqlConnection("server=OZAN\\SQLEXPRESS;database=TEZ;uid=sa;
password=sa");
    sql_baglanti.Open();
    SqlCommand komut = new SqlCommand("UPDATE OGRENCI_OZELLIK
SET OGRENME_STILI="" + ogrenme_stili + "" WHERE OGRENCI_KA="" +
kullanıcı.UserName + """, sql_baglanti);
    komut.ExecuteNonQuery();
    Profile.ogrenme_stili = ogrenme_stili;
    Panel2.Enabled = false;
    Button8.Enabled = false;
    Button4.Enabled = false;
    bilgi.Text = "Kayıt Tamamlandı. Öğrenme Stiliniz="" + ogrenme_stili + "" .
Lütfen Anasayfaya geri dönün...";

}
protected void Button8_Click(object sender, EventArgs e)
{
    int[] say = new int[5];
    for (int i = 0; i < soru_sayisi; i++)
    {

```

```

        RadioButtonList secenek =
(RadioButtonList)this.FindControl("ctl00$ContentPlaceHolder1$ctl" +
Convert.ToInt32(i / 10) + i % 10);
        say[Convert.ToInt32(sorunun_stili[i])] +=
Convert.ToInt32(secenek.SelectedValue);
    }
    if (say.Max() != 0)
        bulanik(say);
    else
        bilgi.Text = "Lütfen seçim yapınız";
}

protected void bulanik(int[] say)
{
    double temp_sayi;
    double[] deneme = new double[30];
    bulanik_kume1.bulanik_kume[] temp_bulanik_sayi = new
bulanik_kume1.bulanik_kume[4];

    max_min_normalizasyon1 mmn = new max_min_normalizasyon1();
    bulanik_kume1 bk = new bulanik_kume1();
    kural_tabani1 kt = new kural_tabani1();
    for (int i = 1; i <= 3; i++)
    {
        temp_sayi = mmn.max_min_normalizasyon(say[i], 0, 100, 0, ((soru_sayisi /
3) * (secenek_sayisi - 1)));
        temp_bulanik_sayi[i] = bk.bulaniklastir(i,temp_sayi, 0, 50, 100);

        Response.Write(say[i] + "---->" + temp_sayi + "----->" +
temp_bulanik_sayi[i].kume[1] + "----->" +
temp_bulanik_sayi[i].kume_uyelik_degeri[1] + temp_bulanik_sayi[i].kume[2] + "----
->" + temp_bulanik_sayi[i].kume_uyelik_degeri[2] + "<br>");
    }
    Response.Write(kt.kural_tabani(temp_bulanik_sayi));
    kaydet(kt.kural_tabani(temp_bulanik_sayi));
}
}
}

```

ÖZGEÇMİŞ

Adı Soyadı	:Ozan ALPTEKİN
Doğum Tarihi	:05.07.1982
Doğum Yeri	:İzmir
Yabancı Dili	:İngilizce
E-posta	:ozanalptekin@yahoo.com
İlköğretim	:İzmir Bornova Muzaffer Taşdemir İlköğretim Okulu
Lise	:İzmir Konak Çınarlı Endüstri Meslek Lisesi Bilgisayar Donanım Bölümü
Üniversite	:Ege Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi
Yüksek Lisans	:Beykent Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bölümü
İş Deneyimi	:İzmir Üniarsiade Oyunları Sistem Destek Uzmanı Gaziosmanpaşa Yıldıztabya İlköğretim Okulu Bilişim Teknolojileri Öğretmeni