



**LEARNING MANAGEMENT SYSTEM  
(UZAKTAN EĞİTİM SİSTEMİ)**

**HASAN UCUZAL  
FIRAT ŞAHİN**

**Mayıs – 2019**

# İÇİNDEKİLER

## **1.GİRİŞ**

- 1.1** Projenin Amacı
- 1.2** Projenin Kapsamı
- 1.3** Tanımlamalar ve Kısaltmalar

## **1. PROJE PLANI**

- 2.1** Giriş
- 2.2** Projenin Plan Kapsamı
- 2.3** Proje Zaman-İş Planı
- 2.4** Proje Ekip Yapısı
- 2.5** Önerilen Sistemin Teknik Tanımları
- 2.6** Kullanılan Özel Geliştirme Araçları ve Ortamları
- 2.7** Proje Standartları, Yöntem ve Metodolojiler
- 2.8** Kalite Sağlama Planı
- 2.9** Konfigürasyon Yönetim Planı
- 2.10** Kaynak Yönetim Planı
- 2.11** Eğitim Planı
- 2.12** Test Planı
- 2.13** Bakım Planı

## **3. SİSTEM ÇÖZÜMLEME**

### **3.1 Mevcut Sistem İncelemesi**

- 3.1.1 Örgüt Yapısı
- 3.1.2 İşlevsel Model
- 3.1.3 Veri Modeli
- 3.1.4 Varolan Yazılım/Donanım Kaynakları
- 3.1.5 Varolan Sistemin Değerlendirilmesi

### **3.2 Gereksenen Sistemin Mantıksal Modeli**

- 3.2.1 Giriş
- 3.2.2 İşlevsel Model
- 3.2.3 Genel Bakış

- 3.2.4 Bilgi Sistemleri/Nesneler
- 3.2.5 Veri Modeli
- 3.2.6 Veri Sözlüğü
- 3.2.7 İşlevlerin Sıradüzeni
- 3.2.8 Başarım Gerekleri

### **3.3 Arayüz (Modül) Gerekleri**

- 3.3.1 Yazılım Arayüzü
- 3.3.2 Kullanıcı Arayüzü
- 3.3.3 İletişim Arayüzü
- 3.3.4 Yönetim Arayüzü

### **3.4 Belgeleme Gerekleri**

- 3.4.1 Geliştirme Sürecinin Belgelenmesi
- 3.4.2 Eğitim Belgeleri
- 3.4.3 Kullanıcı El Kitapları

## **4. SİSTEM TASARIMI**

### **4.1 Genel Tasarım Bilgileri**

- 4.1.1 Genel Sistem Tanımı
- 4.1.2 Varsayımlar ve Kısıtlamalar
- 4.1.3 Sistem Mimarisi
- 4.1.4 Dış Arabirimler
  - 4.1.4.1 Kullanıcı Arabirimleri
  - 4.1.4.2 Veri Arabirimleri
  - 4.1.4.3 Diğer Sistemlerle Arabirimler
- 4.1.5 Veri Modeli
- 4.1.6 Testler
- 4.1.7 Performans

### **4.2 Veri Tasarımı**

- 4.2.1 Tablo tanımları
- 4.2.2 Tablo- İlişki Şemaları
- 4.2.3 Veri Tanımları
- 4.2.4 Değer Kümesi Tanımları

### **4.3 Süreç Tasarımı**

#### **4.3.1 Genel Tasarım**

#### **4.3.2 Modüller**

##### **4.3.2.1 Yönetici Modülü**

- 4.3.2.1.1 İşlev
- 4.3.2.1.2 Kullanıcı Arabirimi
- 4.3.2.1.3 Modül Tanımı
- 4.3.2.1.4 Modül iç Tasarımı

##### **4.3.2.2 Seçmen Modülü**

- 4.3.2.2.1 İşlev
- 4.3.2.2.2 Kullanıcı Arabirimi
- 4.3.2.2.3 Modül Tanımı
- 4.3.2.2.4 Modül iç Tasarımı

#### **4.3.3 Kullanıcı Profilleri**

#### **4.3.4 Entegrasyon ve Test Gereksinimleri**

### **4.4 Ortak Alt Sistemlerin Tasarımı**

- 4.4.1 Ortak Alt Sistemler
- 4.4.2 Modüller arası Ortak Veriler
- 4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri
- 4.4.4 Güvenlik Altsistemi
- 4.4.5 Veri Dağıtım Altsistemi
- 4.4.6 Yedekleme ve Arşivleme İşlemleri

## **5. SİSTEM GERÇEKLEŞTİRİMİ**

### **5.1. Giriş**

### **5.2. Yazılım Geliştirme Ortamları**

- 5.2.1 Programlama Dilleri
- 5.2.2 Veri Tabanı Yönetim Sistemleri
  - 5.2.2.1 VTYS Kullanımının Ek Yararları
  - 5.2.2.2 Veri Modelleri
  - 5.2.2.3 Şemalar
  - 5.2.2.4 VTYS Mimarisi

- 5.2.2.5 Veritabanı Dilleri ve Arabirimleri
- 5.2.2.6 Veri Tabanı Sistem Ortamı
- 5.2.2.7 VTYS'nin Sınıflandırılması
- 5.2.2.8 Hazır Program Kütüphane Dosyaları
- 5.2.2.9 CASE Araç ve Ortamları

### **5.3. Kodlama Stili**

- 5.3.1 Açıklama Satırları
- 5.3.2 Kod Biçimlemesi
- 5.3.3 Anlamlı İsimlendirme
- 5.3.4 Yapısal Programlama Yapıları

### **5.4. Program Karmaşıklığı**

- 5.4.1 Programın Çizge Biçimine Dönüştürülmesi
- 5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama

### **5.5. Olağan Dışı Durum Çözümleme**

- 5.5.1 Olağandışı Durum Tanımları
- 5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları

### **5.6. Kod Gözden Geçirme**

- 5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi
- 5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular
  - 5.6.2.1 Öbek Arayüzü
  - 5.6.2.2 Giriş Açıklamaları
  - 5.6.2.3 Veri Kullanımı
  - 5.6.2.4 Öbeğin Düzenlenişi
  - 5.6.2.5 Sunuş

## **6. DOĞRULAMA VE GEÇERLEME**

- 6.1. Giriş
- 6.2. Sınama Kavramları
- 6.3. Doğrulama ve Geçerleme Yaşam Döngüsü
- 6.4. Sınama Yöntemleri
  - 6.4.1 Beyaz Kutu Sınaması
  - 6.4.2 Temel Yollar Sınaması
- 6.5. Sınama ve Bütünleştirme Stratejileri

- 6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme
- 6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme
- 6.6. Sınama Planlaması
- 6.7. Sınama Belirtileri
- 6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri

## **7. BAKIM**

- 7.1 Giriş
- 7.2 Kurulum
- 7.3 Yerinde Destek Organizasyonu
- 7.4 Yazılım Bakımı
  - 7.4.1 Tanım
  - 7.4.2 Bakım Süreç Modeli

## **8. SONUÇ**

## **9. KAYNAKLAR**

## **1. Amaç**

Yaşam boyu öğrenme dayanağı ile günümüz mekan ve zaman kavramına çözüm üretmek için uzaktan eğitim yazılımlarına ihtiyaç duyulmaktadır. Bu bağlamda ihtiyaç duyulan bu yazılım mevcut örgün eğitim modellerine alternatif olarak sunulacaktır. Öğrencilerin birebir iletişim ve öğrenim ihtiyaçlarını karşılamaya yönelik canlı dersler, quiz, başarıyı ölçebilecek testler, doküman paylaşımları, sohbet ve paylaşım imkanı gibi öğrenme metodunu kullanarak e- öğrenme yöntemlerinin uygulanması amaçlanmaktadır.

### **1.2 Projenin Kapsam**

Öğrenme günümüz çağında bireylerin yaşam unsurlarını iyileştirmek için en önemli sayılabilecek etkenler biridir. Bu nedenle bireyler her zaman öğrenmeye ihtiyaç duymaktadır. Fakat günümüz koşullarında bireylerin yoğun hayat koşulları arasında örgün eğitim metodları ile öğrenme imkanları bulunmamaktadır. Uzaktan Eğitim platformları sayesinde bu imkânsızlıklar aşılmaktadır.

## 2.1 Giriş

Yapacağımız bu proje yaşam boyu öğrenimi amaçlayan tüm bireylere hitap eden bir sistemdir. Bu sistem web ve mobil platformlarda çalışacak olup, senkron veya asenkron bir şekilde bireyler tarafından kullanılabilir.

## 2.2 Projenin Plan Kapsamı

Uzaktan Eğitim, geleneksel öğrenme-öğretme yöntemlerinden farklı olarak eğitim çalışmalarını planlayanlar ve uygulayanlar ile öğrenenler arasındaki iletişim ve etkileşimi, zaman ve yer kısıtlaması olmaksızın gerçekleştirmeyi amaçlar. Bu nedenle uzaktan eğitim; kendine özgü ders tasarımı, iletişim yöntemleri ve örgütsel yapıyı kapsayacak bir kurumsal eğitim ortamına ihtiyaç duymaktadır. Bu aşamada, eğitim-öğretim faaliyetlerinin internet aracılığıyla gerçekleştirilmesini sağlayacak Eğitim Yönetim Sistemleri (EYS) oldukça önem kazanmaktadır.





### Uygulama'nın Maliyet Kestirim Analizi

Cocomo Maliyet Kestirimi modeline göre deneme için yapılan analizler ile elde edilen parametreler ile elde edilen maliyet kestirimi verileri aşağıdaki gibidir.

### Maliyet Kestirim Dokümanı

Ölçüm Parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	70	3	210
Kullanıcı Çıktı Sayısı	55	4	220
Kullanıcı Sorgu Sayısı	30	3	190
Kütük Sayısı	75	7	525
Dışsal Arayüz Sayısı	34	5	170
Ana İşlev Nokta Sayısı			1315

	0	1	2	3	4	5
Yedekleme & Kurtarma	*					
Veri İletişimi			*			
Dağıtık İşlem İşlevleri	*					
Kritik Performans	*					
Ağır Yüklü İşletim Ortamı	*					
Çevrim İçi Veri Girişi			*			
Çevrim İçi Kütük Güncelleme				*		
Girdiler,Çıktılar,Sorgular Karmaşık		*				
İçsel İşlemler Karmaşık		*				
Tasarlanacak Kod , yeniden kullanılabilir				*		
Dönüştürme ve kurulum	*					
Sistem birden çok yerde yerleşik farklı kurumlar	*					
Kolay kullanılabilir & Değiştirilebilir.	*					

<b>Maliyet Etmenleri</b>	<b>Çok Düşük</b>	<b>Düşük</b>	<b>Normal</b>	<b>Yüksek</b>	<b>Çok Yüksek</b>	<b>Oldukça Yüksek</b>
<b>Güvenilirlik</b>	0.75	0.88	<b>1.00</b>	1.15	1.40	-
<b>VT Büyüklüğü</b>	-	0.94	<b>1.00</b>	1.08	1.16	-
<b>Karmaşıklık</b>	0.70	0.85	<b>1.00</b>	1.15	1.30	1.65
<b>Zaman</b>	-	-	<b>1.00</b>	1.11	1.30	1.66
<b>Ana Bellek Kısıtı</b>	-	-	<b>1.00</b>	1.06	1.21	1.56
<b>Platform Değişim Olasılığı</b>	-	0.87	<b>1.00</b>	1.15	1.30	-
<b>İş-Geri Dönüş Zamanı</b>	-	0.87	<b>1.00</b>	1.07	1.15	-
<b>Analist Yeteneği</b>	1.46	1.19	1.00	<b>0.86</b>	0.71	-
<b>Uygulama Deneyimi</b>	1.29	1.13	1.00	<b>0.91</b>	0.82	-
<b>Programcı Yeteneği</b>	1.42	1.17	1.00	<b>0.86</b>	0.70	-
<b>Bilgisayar Platformu Deneyimi</b>	1.21	1.10	1.00	<b>0.90</b>	-	-
<b>Programlama Dil Deneyimi</b>	1.14	1.07	1.00	<b>0.95</b>	-	-
<b>Modern Programlama Teknikleri</b>	1.24	1.10	1.00	0.91	<b>0.82</b>	-
<b>Yazılım Geliştirme Araçları Kullanımı</b>	1.24	1.10	1.00	0.91	<b>0.83</b>	-
<b>Zaman Kısıtı</b>	1.23	1.08	1.00	1.04	<b>1.10</b>	-
<b>ORTALAMA : 1.02</b>						

**0:** Hiçbir etkisi yok

**1:** Çok az etkisi var

**2:** Etkisi var

**3:** Ortalama etkisi var

**4:** Önemli etkisi var

**5:** Mutlaka olmalı, kaçınılmaz

$$\dot{IN} = A\dot{IN} \times (0.65 \times 0.01 \times TKF) \rightarrow \dot{IN} = 1315 \times (0.65 \times 0.01 \times 60) = \underline{512,85}$$

$$\text{Satır sayısı} = \dot{IN} \times 30 = 512,85 \times 30 = \underline{15.385,5} \text{ SATIR YAKLAŞIK}$$

### COCOMO-Etkin Maliyet Modeli

Projemiz küçük bir ekip tarafından geliştirildiği için ayırık projeler arasına grubunda yer almaktadır.

$$\text{Aylık Kişi Başı İş Gücü} = E = a \cdot (KSS)^b$$

$$\text{Geliştirme Süresi (Ay)} = D = c \cdot (E)^d$$

$$\text{Eleman Sayısı} = E / D$$

(KSS = Kod Satır Sayısı manasına gelmektedir ve birimi bin satırdır. Projenin tahmini kaç bin satırdan oluşacağını belirtmemizi sağlar.)

### Ayrık Proje İçin Katsayılar İle

$$\text{İş Gücü } K = 2.4 \times S^{1.05} \rightarrow K = 2.4 \times 12^{1.05} \times 1.02 = \underline{33,26}$$

$$\text{Zaman } T = 2.5 \times K^{0.38} \rightarrow T = 2.5 \times 33,26^{0.38} = \underline{9 \text{ Ay}}$$

$$\text{N Ortalama Kişi Sayısı} = \text{İş Gücü} / \text{Zaman} = \underline{4 \text{ Kişi.}}$$

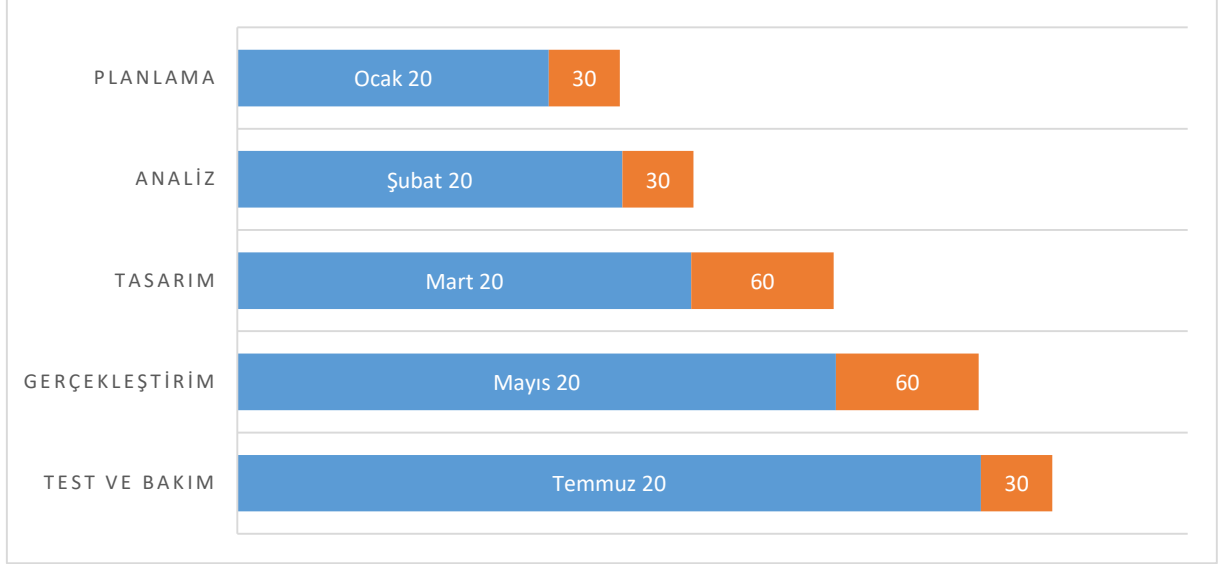
### Maliyet:

Tanımı/Adı	Miktar	Birimi	Birim Fiyatı
Domain	1	Adet	50
Sunucu Kiralama	1	Aylık	400
Proje Yöneticisi	1	Aylık	4000
Yazılım Ekip Lideri	1	Aylık	3500
Sistem Analisti	1	Aylık	3500
Test Mühendisi	1	Aylık	3000
<b>Toplam</b>			<b>8950,00 TL</b>

NOT = Tahmini fiyat listesi 2019 Mayıs ayı için hesaplanmıştır.

### 2.3 Proje Zaman İş Planı

Projenin öngörülen süresi yaklaşık 9 aydır.



### 2.4 Proje Ekip Yapısı

Görevi	Adı Soyadı
Proje Yöneticisi	xx
Yazılım Ekip Lideri	xx
Sistem Analisti	xx
Test Mühendisi	xx

GÖREV	AÇIKLAMA
Proje Yöneticisi	Projenin yönetilmesi Proje Ekip yapısının oluşturulması İş planlamasının yapılması
Yazılım Ekip Lideri	Yazılımın geliştirilme sürecinin takibi
Sistem Analisti	Sistemin Analizi Dökümantasyon - Raporlama
Test Mühendisi	Yazılan modüllerinin ve sistemin test edilmesi

## 2.6 Kullanılan Özel Geliştirme Araçları ve Ortam

Donanım :

Araç	Özellikleri	Amacı	Adet
Domain	Com uzantılı domain adı	Site adı oluşturma	1
Hosting	10 GB Alan	Web dosyaları ve veritabanı barındırma	1
Android Cihaz	Android 6.0 +	Android Uygulama Testi	1
iPhone Cihaz	IOS 9.0 +	IOS Uygulama Testi	1

#### Yazılım Geliştirme Araçları:

Araç	Kullanım Amacı	Lisans
PHP	Web Geliştirme	Open Source
MYSQL	Veritabanı	Open Source
Java Script	Web Geliştirme	Open Source
Apache	Web Sunucu	Open Source
Safari,Chrome,Opera,Firefox	Web Sitesinin Testi	Free

#### 2.6.1 Metodoloji

Uzaktan Eğitim yazılım sistemi özellik güdümlü yaklaşım ve Scrum Çevik Proje Geliştirme Metodolojisi ile geliştirilecektir. Bu metodolojiye göre proje geliştirme ekibi, geliştirilmesi istenen özellikleri belirleyerek başlar ve bu özelliklerin geliştirilmesi için gereken süreyi belirler. Sprint adı verilen her bir 1-4 haftalık sürede, bir özellik geliştirme hedeflenir. Günlük Scrum buluşmaları ile günlük ilerlemeler takip edilir, aylık scrum buluşmaları ile hedeflenen ilerleme kaydedilip kaydedilmediği kontrol edilir.

#### 2.6.2 Proje Geliştirme Standardı

Bu proje için Helezonik model kullanılacak, geliştirilmesi gereken özellikler belirlenecek, risk analizi yapılacak, özellikler geliştirilecek, onaya sunulacak ve bir önceki özelliklerle birleştirilecektir.

## 2.8 Kalite Saęlama Planı

- 1.**Ekonomi:** Akıllı deneme uygulaması ile dięer ürünler ile arasında rekabet farkı olacak, otomatik analiz yapan yazılım ile, reklam yapılabilir.
- 2.**Tamlık:** Tüm deęerler, doęru bir şekilde hesaplanmalı uygulama doęru bir şekilde çalışmalıdır.
- 3.**Yeniden Kullanılabilirlik:** Sistem, her koşulda tekrardan düzenlenip kullanılabilir. 4.**Etkinlik:** Kullanıcı sistemin her alanına hâkim olduęu için sistemi etkin bir biçimde kullanacak.
- 5.**Bütünlük:** Proje yöneticisi, sistemin tüm kısımlarına hakim olacak ve program bir bütün halinde çalışacaktır.
- 6.**Güvenilirlik:** Google altyapısı ile çalışıldığından altyapı konusunda, Google'ın güvenlik önlemleri referans gösterilecektir. Tasarım aşamasında, analiz ekibinden geliştirme ekibine kadar herkes güvenilirlikten sorumludur.
- 7.**Modülerlik:** Modüler yapısı ile , web kısmında farklı kullanıcı rollerine farklı yetkiler verilmiştir.
- 8.**Belgeleme:** Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olacak bu doküman oluşturulmuştur.
- 9.**Kullanılabilirlik:** Kullanılabilirlik olarak her seviyedeki insana hitap edeceğinden zor renkler karmaşık sistemlerden kaçınılmıştır.
- 10.**Temizlik:** Uygulama tasarlanırken gereksiz detaylardan kaçınılır.
- 11.**Deęiştirilebilirlik:** Otomasyonun veri tabanını erişme yetkisi olan , yazılım ekip lider, ve bakım , geliştirme ekibi yazılımı deęiştirebilir, güncelleyebilir.
- 12.**Esneklik:** Proje farklı platformlarda ve internet üzerinden çalışacağından gayet esnektir. 13.**Genellik:** Proje her sınav türü için kullanılabilirliğinden geneldir.
- 14.**Sınanabilirlik:** Uygulamanın beta sürümü ile sınanması gerçekleştirilecektir. 15.**Taşınabilirlik:** Sistem mobil cihazlar ve internet üzerinden kullanılacağından, herhangi bir özel cihaz gerektirmez ve istenilen cihazlarda taşınabilir ve kullanılabilir.
- 16.**Birlikte Çalışılabilirlik:** Bu projedeki en büyük sıkıntı olacak veri girişı şuanda var olan ve her bireyin bilgilerinin saklayan sistemle birleşik ve eş zamanlı çalışmakta.

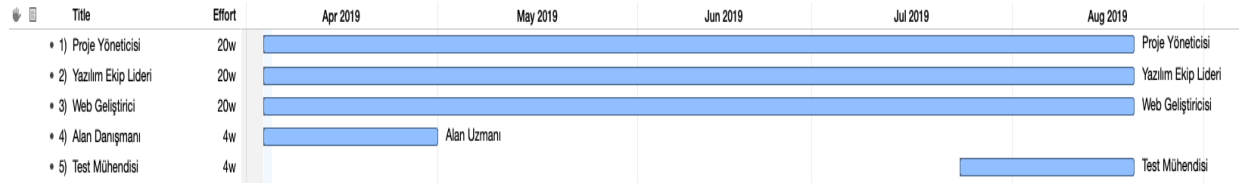
## 2.9 Konfigürasyon Yönetim Planı

Sistemin ileride kullanıcının yeni istemlerini karşılayamaması veya sistemin yapısındaki bazı bileşenlerin değişmesi sonucu güncelliğini kaybettiğinde sistemin kesintiye uğramaması için olası konfigürasyon planı hazırlandı.

- LMS sisteminin değişmesi,
- Herhangi bir sebepten dolayı ayrılan personel olması,
- Sistemde herhangi bir istenmeyen durum halinde,  
Durumları için konfigürasyon yönetim planı oluşturuldu.

## 2.11 Eğitim Planı

Yazılım tamamlandığında, kullanacak yayıncı şirkete yazılım ekip lideri tarafından otomasyonun kullanımı için eğitim verilir.





## **2.12 Test Planı**

### **Entegrasyon :**

Planlanan özelliklerin hepsi eklenip birleştirildiğinde entegrasyon testi yapılır. Entegrasyon testi bittiğinde uygulama beta sürümü ile kullanıcı testine açılır.

## **2.13 Bakım Planı**

Kullanıcılar otomasyon menüsünden olan bir hatayı bildirebilir. Uygulama içerisinde herhangi bir nedenden durdurulma hatası alınırsa loglar otomatik kaydedilir, ve geliştirme ekibine bildirilir. GitKraken yazılımı ile hata&problem takibi yapılır.

### **3.1.1 Mevcut Sistemlerin İncelenmesi**

Türkiye’de 2014 itibariyle toplam 82 tane üniversite uzaktan eğitim hizmeti vermektedir. Bunlar web tabanlı sistem olarak çoğunlukla ücretli sanal sınıf sistemi olan Adobe Connect yazılımını kullanmaktadırlar. Bunun dışında bazı üniversite ve eğitim kurumları açık kaynak kodlu öğretim yönetim sistemleri kullanmaktadırlar. Burada öğretim yönetim sistemleri seçilirken sistemin performansı, platform bağımsızlığı, esnekliği, görselliği ve son zamanlarda popüler hale gelen mobil uyumlu olup olmadığına bakılmaktadır. Ülkemizde ve dünyada en çok kullanılan açık kaynak kodlu uzaktan eğitim öğrenme yönetim sistemleri MOODLE, eFront, Chamilo, Akademik LMS, Blackboard, Canvas, Schology vb. dir. Özellikle mobil kullanıma geçmiş olması ve esnek, kolay kullanılabilir olması sebebiyle MOODLE üniversitelerde en çok kullanılan sistemdir.

## SCHOOLGY NEDİR?

Mevcut sistemin inceleme aşamasında ilk olarak Schoology platformu ele alınmıştır.

Schoology, yükseköğrenim kurumlarında kullanıcıların akademik içerik oluşturmaları için ve yöntemlerini paylaşımları için sanal bir ağ hizmeti aracılığı yapan bir sanal öğrenme ortamıdır. Schoology, sadece sanal öğrenme ortamı olmayıp Öğrenim Yönetim Sistemi ve ya Uzaktan Eğitim Sistemi (Learning Management System) olarak işlev sağlayan bulut tabanlı ve çevrimiçi sınıf yönetimine sahip bir e-öğrenme sistemidir.

## HİZMETLER

Schoology, çevrimiçi not defteri, katılım kayıtları dediğimiz yoklama, test, sınav, ve ödev gibi özelliklere sahiptir. Sınıf, grup ve okul arasındaki işbirliğini kolaylaştırmak amaçlı sosyal medya özelliğine sahip içerikleri mevcuttur. Ayrıca sistem e-doküman ve internet güvenliğini üst seviyeye çıkararak, güven ortamı içerisinde sınav sisteminin çalışmasını sağlamaktadır.

Sistem geliştirmeleri arasında metin mesaj bildirimi, Google Doküman entegrasyonu ve paylaşılan kaynak kütüphanesi, testler yer almaktadır.

## SİSTEM ARAYÜZLERİ VE İŞLEVLERİ

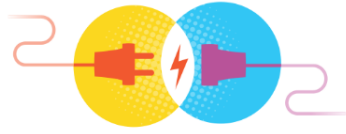


Görsel 1. Schoology, sisteminin hangi işlevlerde çalıştığı görüntülenmektedir.

### Schoology'nin Özelliklerini Keşfedin



Görsel 2. Schoology Sistem Özellikleri



### 200+ eğitim araçları ve platform entegrasyonları

Schoology, sınıfınızın ihtiyaç duyduğu her araca sahiptir ve 200'den fazla araç, öğrenci bilgi sistemleri (SIS) ve eğitim platformları ile önceden entegre olarak gelir.

[TÜM ENTEGRASYONLARA BAKIN](#)

Infinite Campus

Aeries®  
Student Information System

SKYWARD®

Google Drive

OneDrive

KHAN  
ACADEMY


nearpod

Brain  
POP

Görsel 3. Schoology Eğitim Araçları ve Platformlar



Görsel 4. Schoology Kullanıcıları




Sign in to Schoolology

All fields are required

Log in

Forgot your password  
SSO Login

or




Scan QR Code

Schoolology © 2019 · Privacy Policy · Terms of Use · Help Center

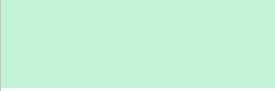
Görsel 5. Schoolology Öğrenci Giriş Paneli

Courses

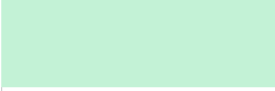
My Courses



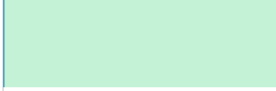
YMH 210 İngilizce İletişim Becerileri  
Gece A Şubesi  
Firat University




YMH 213 Mesleki İngilizce (Güz 2018 Gece...  
Gece A  
Firat University



YMH 214 Sayısal Analiz-Mazeret Sınavı-20...  
Gece A Şubesi  
Firat University



YMH 216 Yazılım Ekonomisi (2019 Bahar)  
Gece A Şubesi  
Firat University



Join a Course

Görsel 6. Schoolology Öğrenci Ders Paneli

## UDEMY NEDİR?

"academy of you" sloganlı online eğitim, öğrenim sitesidir. İnsanların kurs açmasını sağlar. Ücretli bir platformdur. Kurucuları ikisi Türk olmak üzere [Gagan Biyani](#), [Eren Bali](#), [Oktay Çağlar](#)'dır.

## HİZMETLER

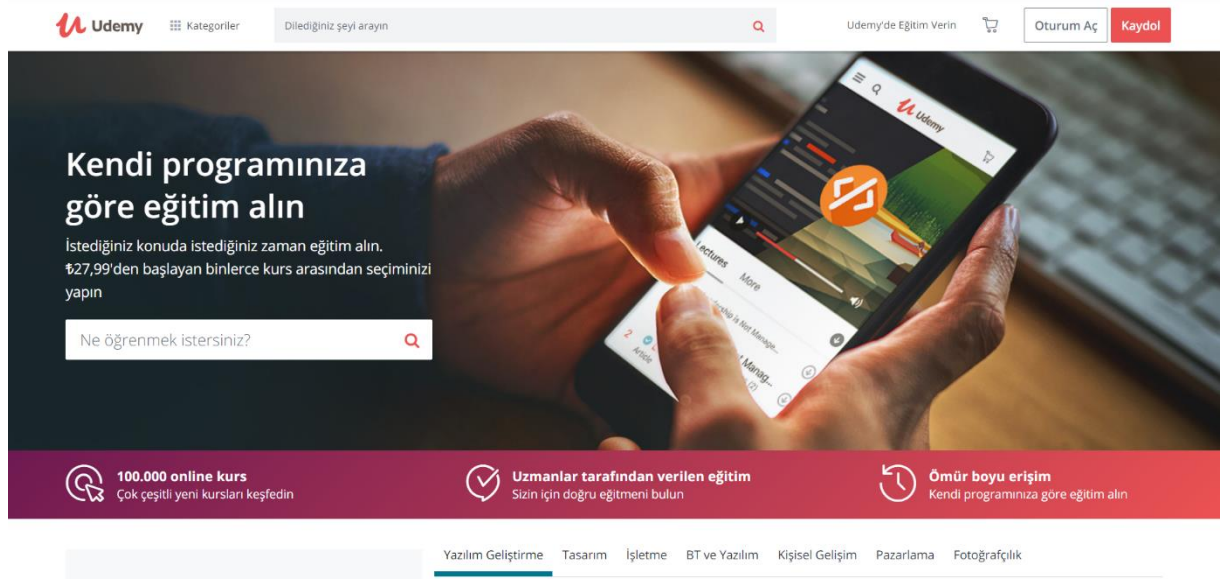
Udemy ile gerçekleştireceğimiz LMS sisteminin benzer özellikleri vardır. Bu özellikleri aşağıdaki şekilde sıralayabiliriz.

- Öğrenci olarak eğitmenlere soru sorabilmek.
- Öğrenci olarak eğitmenleri puanlayabilmek.
- Eğitmen olarak ise öğrencilere kurs verebilmek.
- Eğitmen olarak öğrencilerle doküman paylaşabilmek.
- Eğitmen olarak öğrencilerin kazanımlarını ölçmek için quiz hazırlamak.
- Eğitmen olarak öğrencilere ödev vermek.

Udemy ile LMS sisteminin bazı farklılıkları da vardır. Onları da aşağıdaki şekilde sıralayabiliriz.

- Udemy'de herhangi bir insandan kurs satın alabiliyoruz LMS sisteminde ise üniversite öğretim görevlileri tarafından ücretsiz eğitimlerden faydalanabiliyoruz.
- Udemy'de canlı ders içeriği yoktur fakat LMS sisteminde ise üniversite öğretim görevlileri tarafından canlı ders içeriklerinden faydalanabiliyoruz.
- Udemy'de eğitim sonucunda sadece sertifika alabiliriz fakat LMS'de girilen sınavlar sonucu üniversite ders notları etkilenmektedir.

## SİSTEMİN TEMEL ARAYÜZLERİ VE İŞLEVLERİ



Görsel 1. Ana Sayfa

Kaydolun ve Öğrenmeye Başlayın! X

Tam Ad

E-posta

Şifre

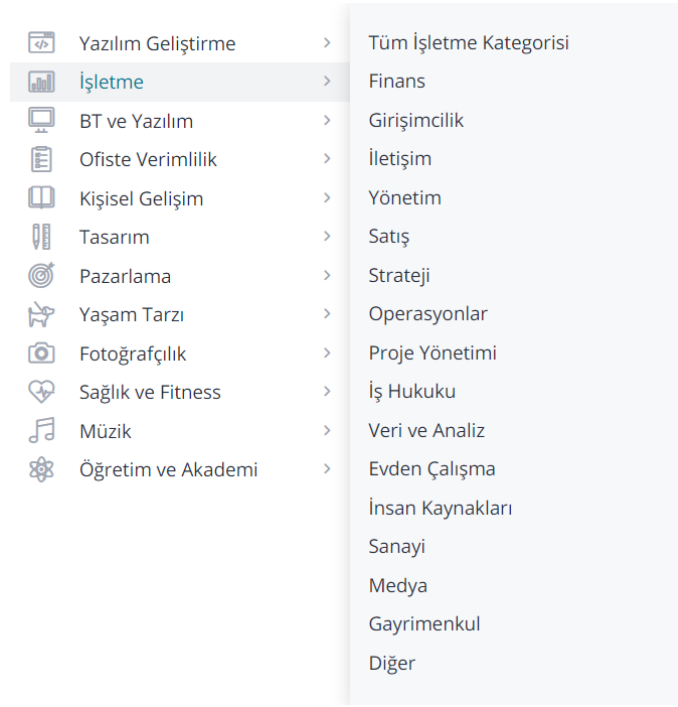
☐ Evet! Özel tekliflerin, kişisel önerilerin ve öğrenim ipuçlarının yer aldığı e-postaları alarak Udemy'den en iyi şekilde yararlanmak istiyorum!

Kaydol

Kaydolarak Kullanım Şartlarımızı ve Gizlilik Politikamızı kabul etmiş olursunuz.

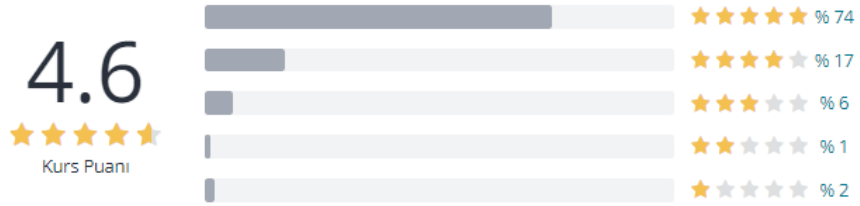
Zaten bir hesabınız var mı? [Oturum Aç](#)

Görsel 2. Kullanıcı Kayıt Ekranı

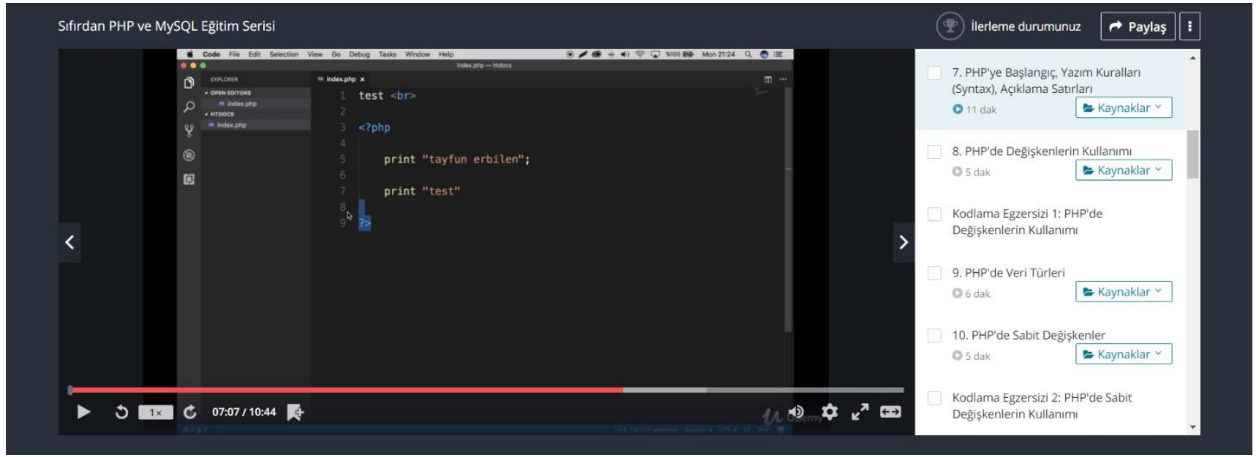


Görsel 3. Kurs Kategorileri

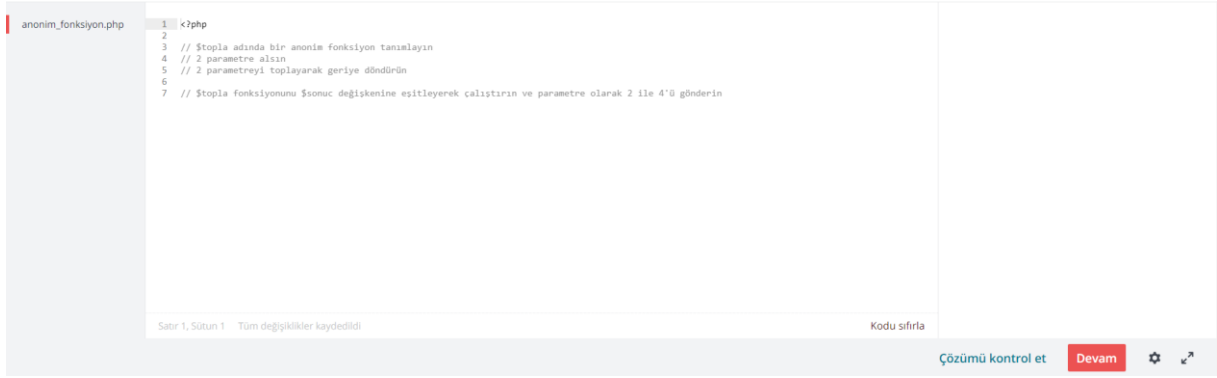
#### Öğrenci geri bildirimi



Görsel 4. Kurs Geri Bildirim Oranı



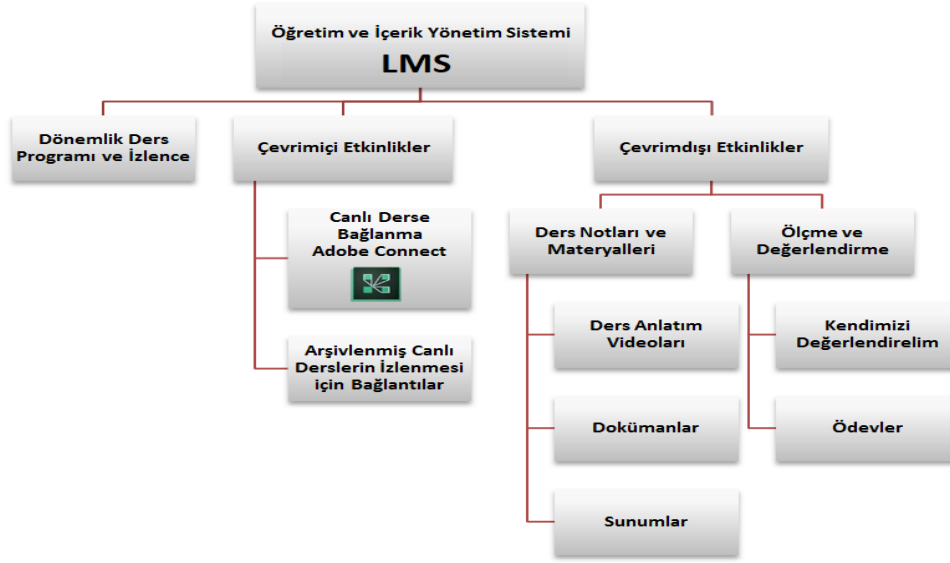
Görsel 5. Kurs Videosu



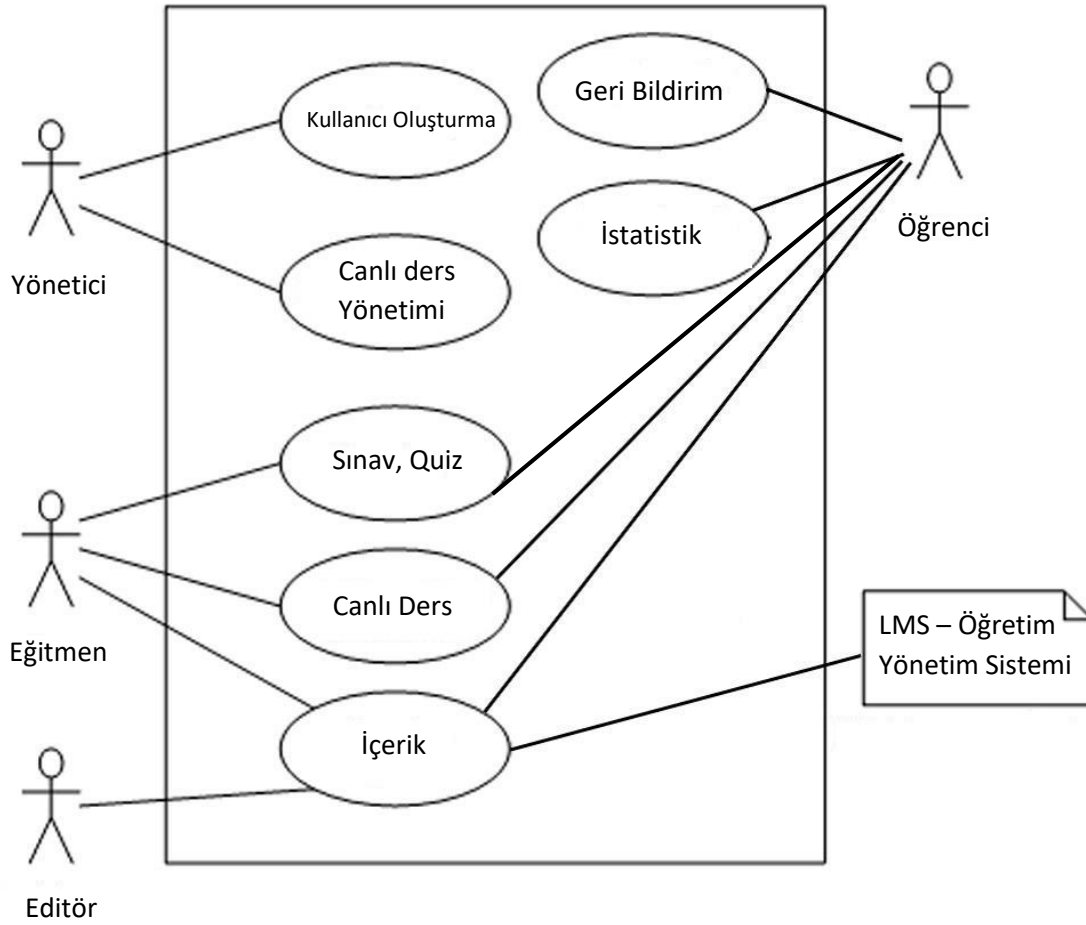
Görsel 6. Kurs Quiz



### 3.1.2 İşlevsel Model



### 3.2.2 İşlevsel Model – Kullanıcı Senaryosu



### 3.2.4 Bilgi Sistemleri/Nesneler

- 1. Online başvuru;** LMS'e ilk kez kayıt yaptıracak öğrencilerin sisteme girmeleri için oluşturulan kayıt formu.
- 2. Kullanıcı grupları;** Sistem kullanıcılarının öğrenci, öğretim elemanı, sistem yöneticisi, sekreter v.b. gruplara ayrılması.
- 3. Rol ve yetkiler;** Sistem içerisindeki modülleri yetkilendirmek için kullanılacak kullanıcı rol tanımları.
- 4. Birim, Bölüm, Program, Ders Tanımlama;** Sistemde Birim, Bölüm, Program, Ders tanımlarının yapılması ve bu tanımlarla öğrenci ve öğretim elemanlarının ilişkilendirilmesi.
- 5. Program Katalogu;** LMS'in ayrıntılı tanıtımlarının yapıldığı web sayfaları.
- 6. Haftalık Ders Programı;** Sisteme girilen saatlerin kullanıcılar tarafından erişilebiliyor olmasını ve Sohbet Saatinin haftalık ders programında görülebilir olmasını sağlamaktadır.
- 7. Şube İşlemleri;** Derslerin şube düzenlemesini, şubelere ders sorumlusu atama işlemlerini ve şubeye öğrenci ekleme ve şube bilgilerini düzenleme işlemlerinin yapılmasını sağlamaktadır.
- 8. Dönem İşlemleri;** Öğrencilerin bulunduğu eğitim yılı ve dönemin tanımlanması sağlanmaktadır.
- 9. Ders Kayıt ve Ders Onay İşlemleri;** Dersleri seçen ve seçmeyen öğrencilerin takip edilmesini, ders seçimini yapan öğrencilerin, ders onaylama işlemini yapmasını ve dönemine ait dersleri seçmesini sağlamaktadır.
- 10. Soru Tanımlama;** Deneme, Vize ve Final Sınav Sorularının sistemde tanımlı ve erişilebilir olmasını sağlamaktadır.
- 11. Sınav;** Ders sorumlularının ilgili dersin sınavını hazırlayabilmesini, Sistemde tanımlı soruları kullanarak sınav hazırlayabilmesini ve Öğrencinin ilgili dersine atanmış sınavları yanıtlayabilmesini sağlamaktadır.
- 12. Deneme Sınavı;** Öğrencinin deneme sınavını hazırlayıp kullanabilmesini sağlamaktadır.

**13. Ödevler;** Yeni ödevin sisteme eklenmesi ve ilgili öğrenciler tarafından görülüyor olmasını, öğrencilerin ise kendisine ait ödevleri görebiliyor olmasını, Öğretim üyelerinin yapmış oldukları ödevleri gönderebilmesi ve ödev veren kişiye ulaşmasını sağlamaktadır.

**14. Öğretim Elemanı Performans Takip;** Öğretim elemanını, vermiş olduğu dersin bilgilerine göre değerlendirebilmesini sağlamaktadır.

**15. Öğrenci Performans Takip;** Öğrencinin kendi notları ile birlikte performans notlarını listeleyebilmesini ve Öğrencinin ders sorumlusu hakkında değerlendirmelerini yapabilmesini sağlamaktadır.

**16. Öğrenci Takip Raporu;** Öğretim üyesinin şubelerdeki öğrencilerini ilgili derslerinde, performans notu ile değerlendirmelerini yapabilmesini sağlamaktadır.

**17. Kişisel Notlar;** Eklenen notun sistemde tanımlı ve ekleyen kullanıcı tarafından erişilebilir olmasını sağlamaktadır.

**18. Mesajlaşma;** Gönderilen mesajları ilgili kişinin görebilmesini ve Kullanıcının gelen mesajlarını görebilmesini sağlamaktadır.

**19. Dosyalar;** Öğretim görevlilerinin vermiş olduğu derslerle ilgili dokümanları sisteme ekleyebilmesini sağlamaktadır.

**20. Sohbet Saatleri;** Sisteme girilen saatlerin kullanıcılar tarafından erişilebiliyor olmasını ve Sohbet Saatinin haftalık ders programında görülebilir olmasını sağlamaktadır.

**21. Sanal Sınıf;** Oluşturulan toplantıların davet edilen kullanıcılar tarafından görülebilir olmasını, Silinen toplantıların davet edilen kullanıcılara iptalinin bildirilmesini ve Davetli olduğu toplantıları görüp, katılabiliyor olmasını sağlamaktadır.

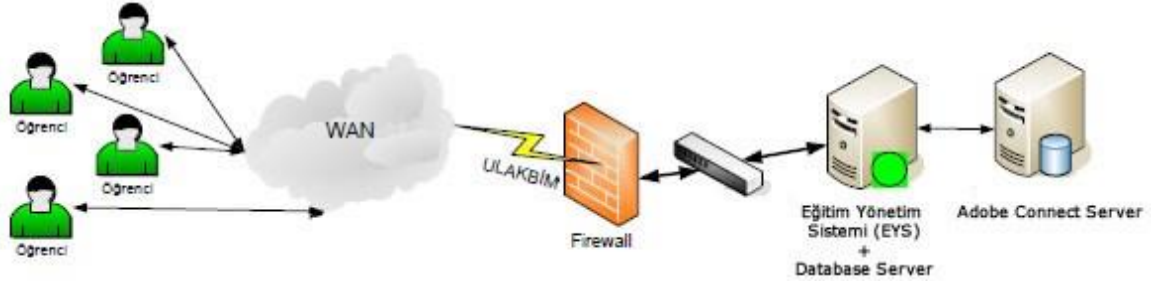
**22. Sıkça Sorulan Sorular;** Öğrencilerin sistemle ilgili sıkıntı yaşadıklarında sıkça sorulan sorular bölümünden problemlerine cevap bulacakları bir sistem bulunmaktadır.

**23. Görüş ve Öneriler;** Kullanıcılar tarafından gönderilen görüş ve önerileri onayladığında sistemde yayınlanmasını, Kullanıcıların, görüş veya önerilerini gönderebilmesini ve Onaylanan görüş ve önerileri görebilmesini sağlamaktadır.

**24. Duyurular Modülü;** Yeni oluşturulan duyurunun ve duyuru tiplerinin sisteme eklenmesini ve kullanıcılar tarafından görülüyor olmasını sağlamaktadır.



### 3.2.7 İşlevlerin Sıradüzeni

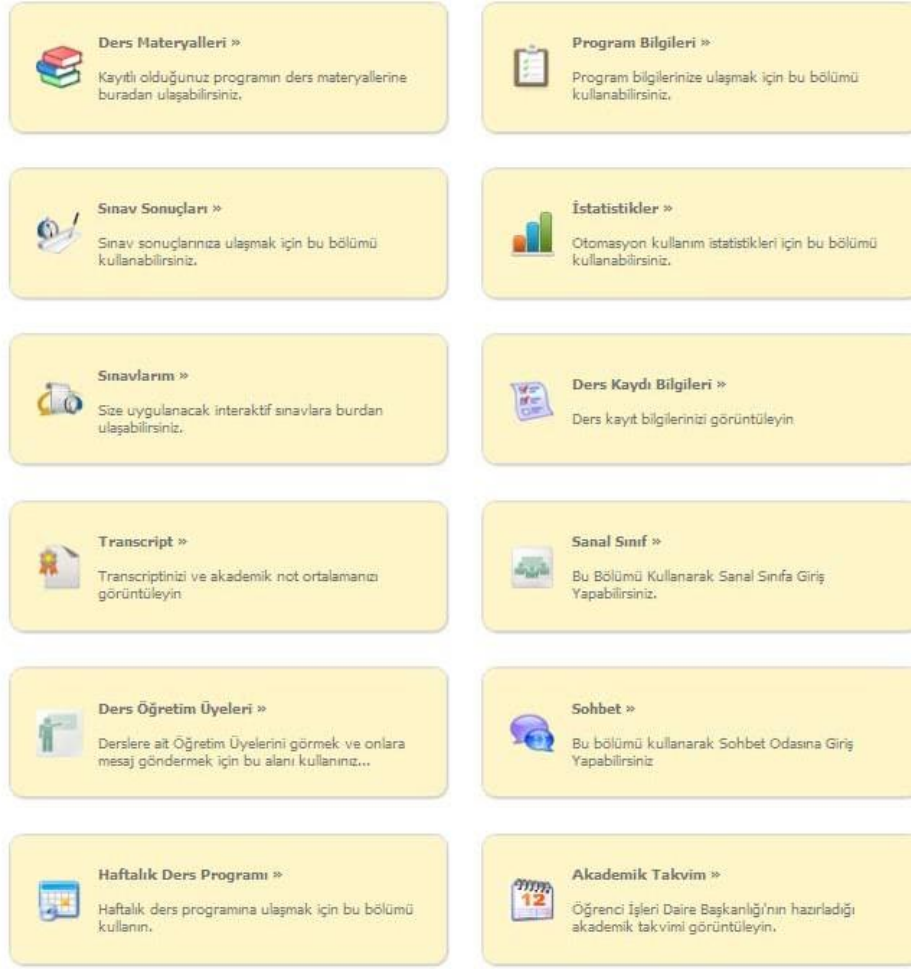


### 3.2.8 Başarım Gerekleri

Başarım gerekleri olarak;

- Katılımcıların ölçme ve değerlendirmelerini zahmetsizce yapması,
- Katılımcıların geçmiş ölçümlerinin görebilmesi ve takip edebilmesi,
- Katılımcıların gelişimini inceleyebilmesi ve analiz yapılabilmesi
- Öğrencilerin başarılarının dijital verilere dönüştürülerek, öğretmenleri ile paylaşabilmesi belirlenmiştir. Yukarıdaki gereklilikler bu ürün ile sağlanabilirse proje başarılı denilir.

### 3.2.1 Yazılım Arayüzü



### 3.2.2 Kullanıcı Arayüzü



Şekil-1: Kullanıcı Giriş Ekranı



Şekil-2: Kullanıcı İşlem Seçim Menusu

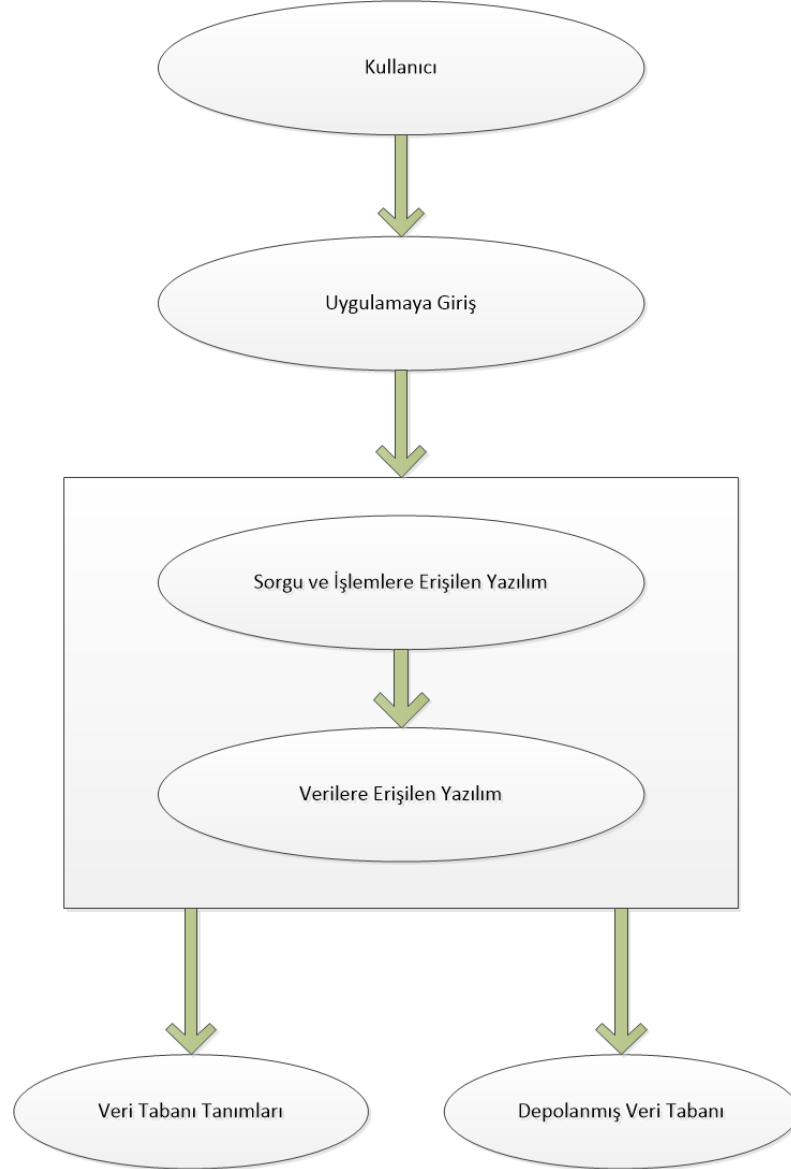
Ders materyalleri görüntüleniyor

Ders Materyalleri							
Türü	Konu	Yazar	Yayın No	Sayfa Sayısı	Kayıt Tarihi	İndir	İzle
	İstatistiksel Verilerin Tanımlanması	Dr. Öğr. Üyesi Dr. Öğr. Üyesi	978603480	0	2020-10-07		
	İstatistiksel Verilerin Tanımlanması	Dr. Öğr. Üyesi Dr. Öğr. Üyesi	975817073	0	2020-10-07		
	İstatistiksel Verilerin Tanımlanması	Dr. Öğr. Üyesi Dr. Öğr. Üyesi	2020-1	55	2020-10-14		

### 3.2.4 Yönetim Arayüzü

## 4. Tasarım

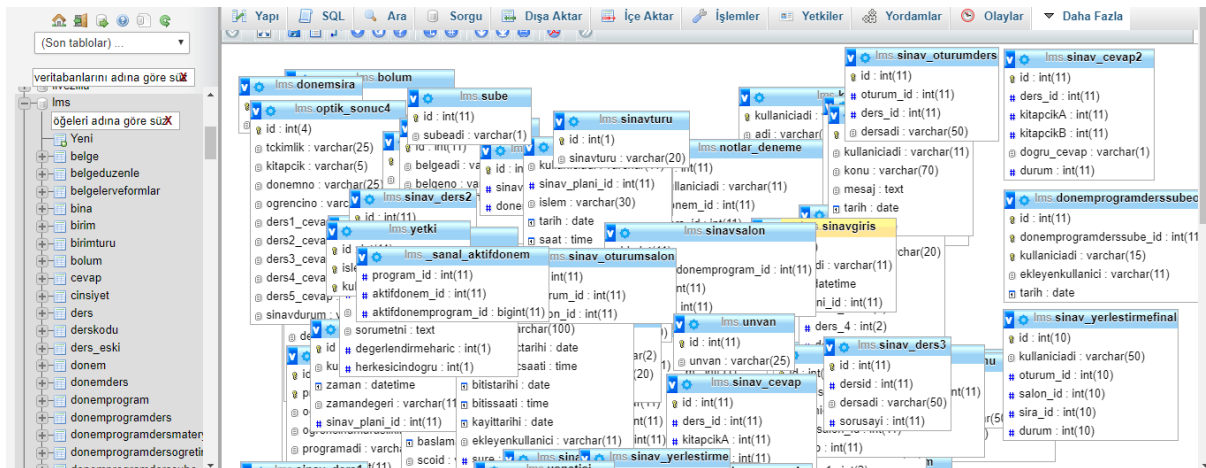
### 4.1.1. Sistem Mimarisi



Şekil 4.2 Veri Tabanı Sistemi

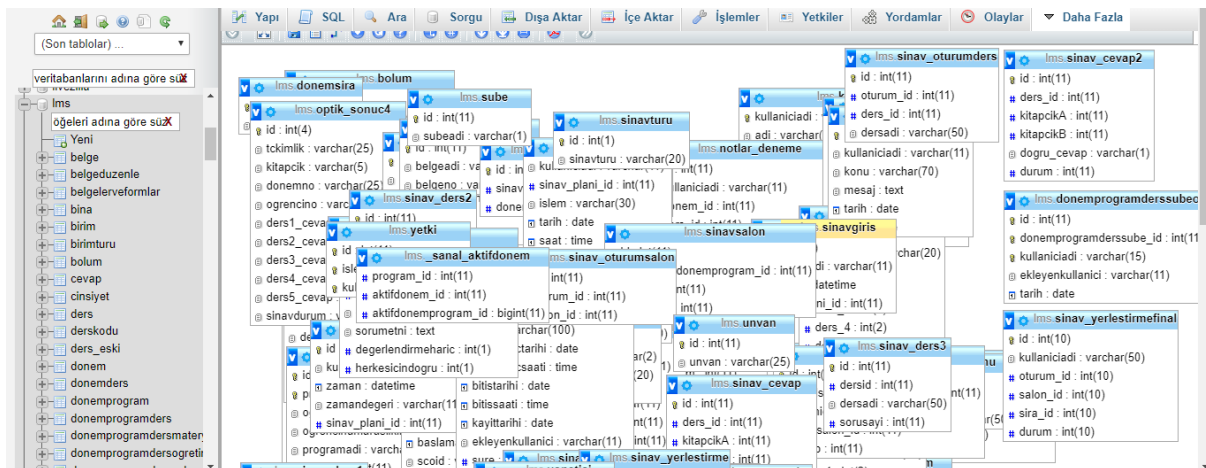


#### 4.1.2. Veri Modeli



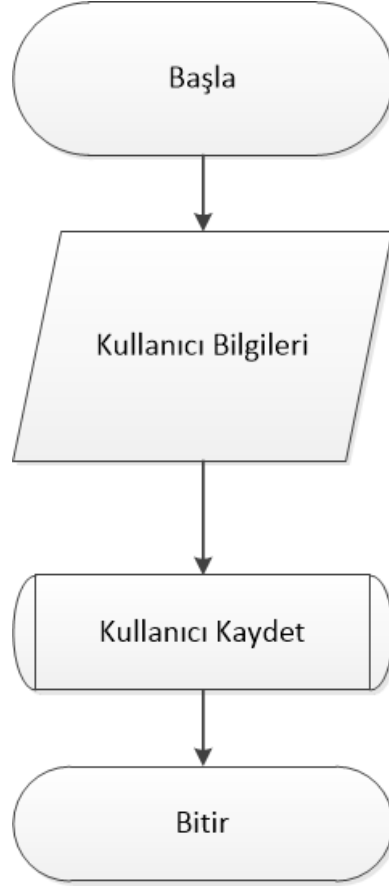
### Şekil 4.3 Veri Tabanı Diyagramları

### 4.3 Veri Tasarımı



#### Şekil 4.4 İlişki Şemaları

### 4.1.3. Genel Tasarım



Şekil 4.6 Yetkili Kullanıcı Tanımlama

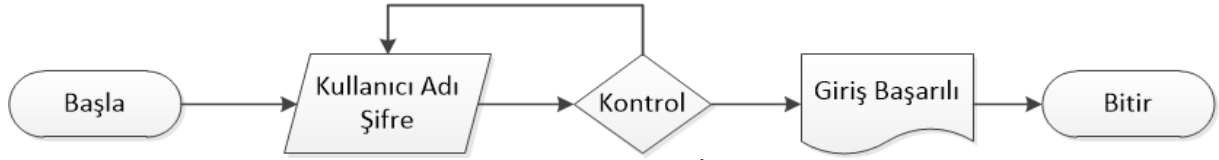
### 4.1.4. Modüller

#### 4.1.4.1. Giriş Modülü

Kullanıcıların sisteme giriş yapmasını sağlayan modüldür. Kullanıcı kodu ve paroladan oluşur.



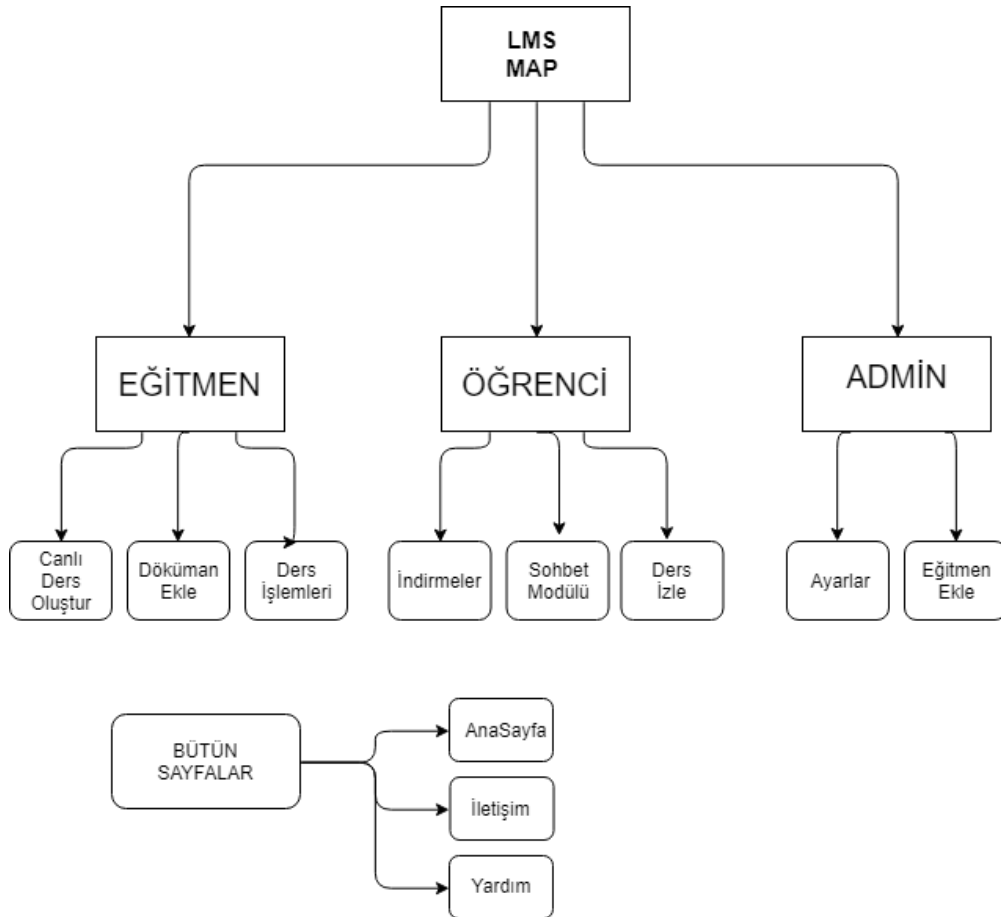
Şekil 4.7 Kullanıcı Giriş Ekranı



Şekil 4.8 Modül İç Tasarımı

## 5.1 GİRİŞ

Yazılım gerçekleştirim aşaması, tasarım aşamasına geçmeden önce olağan dışı durum çözümlerinin yapıldığı, yazılım bileşenlerinin belirlendiği, veri tabanı tablolarının oluşturulduğu ve program karmaşıklığı göz önüne alınarak kod gözden geçirme gibi aşamaları ele alan yazılım geliştirme aşamasıdır. Uzaktan Eğitim Sistemi için bunlar teker teker ele alınmıştır.



Uzaktan Eğitim Sistemi Genel Şablon Haritası

## 5.2 Yazılım Geliştirme Ortamları

Uzaktan Eğitim Sisteminin, kullanılmak istenilen ortamda çalıştırılması için Programlama Dilleri, Veri Tabanı Yönetim Sistemleri, Case Araçları, Sunucu gerekli olan geliştirme ortamlarıdır.

### 5.2.1 Programlama Dilleri

Uzaktan Eğitim Sistemi, bir web uygulaması olmasına esasen sistemin gerçekleşmesi için ve veri işleme yoğunluğuna sahip uygulama özelliği taşımasından görsel programlamaya ağırlık veren programlama dillerinin yanında online işlem yapma yetisine sahip web programlama dilleri göz önüne alınarak sistem gerçekleştirilmiştir.

Sistemin iç mantalitesi html ve css programlama dilleri ile oluşturulmuştur. Görsel sistemin tek başına yeterli olmamasından ve online çalışacak bir sistem olması neticesinde, html ve css programlama dillerinin yanında asp, java, php programlama dilleri dinamik ve tasarımsal bir yapı elde edilmiştir. Veri tabanı için mysql, sql, oracle gibi programlama dilleri kullanılmış olup görsel kalitesinin daha net bir şekilde ortada olması ve pencere-sekme hareketliliği için flash programları kullanılmıştır.



## 5.2.2 Veri Tabanı Yönetim Sistemleri

Uzaktan Eğitim Sistemi için birbiri arasında ilişki olan verilerin tutulduğu, mantıksal ve fiziksel olarak tanımların yer aldığı bilgi depoları, uzaktan eğitim sistemi için kullanım kolaylığı, geliştirme esnekliği ve çeşitli eğitim değişikliklerinin daha kolay yapılması öngörülerek kullanılmıştır.

### 5.2.2.1 Veri Tabanı Yönetim Sistemi Kullanmanın Yararları

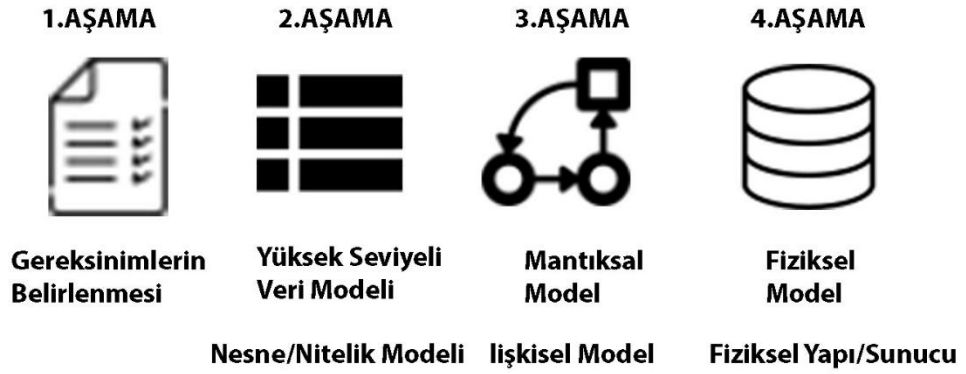
- Esneklik
- LMS hızlı çalışma potansiyeli
- Ortak verilerin admin ve eğitmen tarafından erişilebilme imkanı
- Kullanıcı/Öğrencilerin verilerine kolaylıkla ulaşma ve üzerinde değişiklik yapabilme
- LMS bakım kolaylığı
- Yeni özellik eklenirken kayıtlı verilerin korunması
- Her kullanıcı/eğitmen sadece yeterlilik seviyesindeki verilere ulaşma
- Güncel bilgilerin kullanıcılara aynı zamanda ve aynı doğrultuda oluşması

### 5.2.2.2 Veri Modelleri



Yüksek Seviyeli Veri Modellerinde, Uzaktan Eğitim Sistemi için ilişki tabloları, niteliklerin ve nesnelerin yer aldıkları veri modeli olarak oluşturulmuştur. Mantıksal Veri Modelinde ise

verinin fiziksel olarak bilgisayarda organizesi için oluşturulmuştur. Son olarak Fiziksel Veri Modeli veriye nasıl saklanacağı, saklanan veriye erişimini içeren tabloları oluşturmuştur.



Şema mimarisi, yapılan anket sonucunda belirlenen gereksinimlerden yola çıkılarak elde edilmiştir.

1. Gereksinimlerin Belirlenmesi
  - Veri Tipi
  - Veri Grupları
2. Yüksek Seviyeli Veri Modeli

Veriler arası bağlantının grafiksel olarak ilişkilendirilmesi işleminde kullanılmıştır.

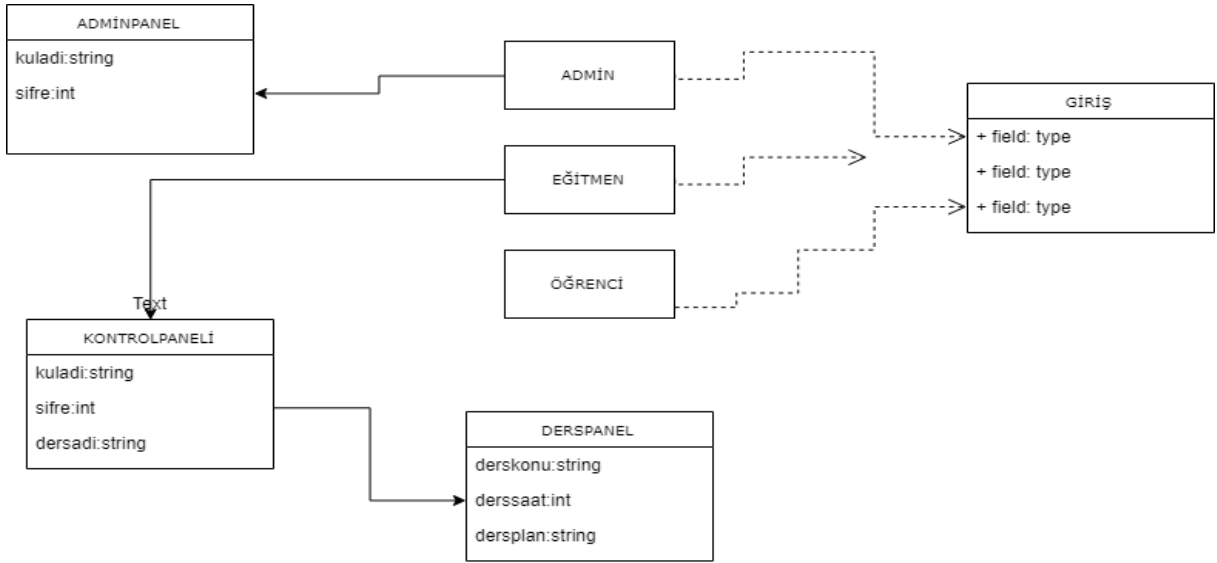
3. Mantıksal Model

Veri tabloları organizasyonunun ilişkileri dayanak alınarak oluşturulmuştur.

4. Fiziksel Model

Fiziksel olarak sistemin kurulup sunucu ile bağlanması işlemlerinde kullanılmıştır.

### 5.2.2.3 Şemalar



### 5.2.2.4 Vtys Mimarisi

Veri tabanı yönetim sistemi mimarisinin düzeyler halinde ele alacak olursak:

- İçsel Düzey: Uzaktan Eğitim Sistemi kullanıcılarının bir bilgiye ulaşmak istedikleri zaman bilgi için, sistemin veri tabanına ulaşma veya erişim yolunu temsil eder.
- Kavramsal Düzey: Uzaktan Eğitim Sistemin için veri tabanı yapısını şemalar yoluyla açıklayarak, arka planda bulunun fiziksel yapıyı kullanıcıdan gizler.
- Dışsal Düzey: Uzaktan Eğitim Sistemi için sistemin çalışma mantalitesinin anlamak yada öğrenmek isteyen kişilerin dış şemalar yoluyla kullanıcıya görsellik sağlar.

### 5.2.2.5 Veri Tabanı Dilleri Ve Arabirimleri

Uzaktan Eğitim Sistemi için veri tabanı programlama dili olarak SQL kullanılmıştır. Veri tabanı arabirimi olarak ise MySQL kullanılmıştır.

- Veri Tabanı Sistem Ortam: Veri tabanı sistem ortamı olarak SQL Server ve phpmyadmin kullanılmıştır.
- VTYS Sınıflandırması: Veri tabanı sınıflandırması gerçekleştirirken en çok kullanılan veri modeller ilişkisel, ağ, hiyerarşik modellerdir.

- Hazır Program Kütüphane Dosyaları: Uzaktan Eğitim Sistemi için çeşitli hazır kütüphaneler kullanılmıştır.
- Case Araç ve Ortamları: Case araçları olarak PhpStorm, Microsoft SQL Server başta olmak üzere farklı ortam araçları ve case araçları kullanıldı.

## **5.3 Kodlama Stili**

### **5.3.1 Açıklama Satırları**

Uzaktan Eğitim Sisteminin sistem kodlanma aşamasında bir yapıya başlamadan önce yapının işlevini belirtmek için açıklama satırları kullanılmıştır. Ayrıca bütün yapılandırıcılarda, yapılandırıcının özelliğini açıklamak için açıklama satırlarına dikkat edilmiştir.

### **5.3.2 Kod Biçimlesi**

Web tabanlı bir uygulama olması dayanağı ile web programlama dilleri hiyerarşisi ve söz dizimi kuralları gereği satır satır kodlanmış olup, iç içe bir hiyerarşiye sahiptir.

### **5.3.3 Anlamlı İsimlendirme**

Sistem kodlama aşamasında bütün veri tipleri, sınıf adları, fonksiyon adları ve yapılandırıcı adları ait olduğu yapıya örnek olarak adlandırılmıştır. Örneğin “Ders” adlı fonksiyonun altında “dersadı”, “derssaati” adlı bir veri tipi kullanılarak, veri tipleri ve sınıflar, yapılandırıcılar ve fonksiyonlar arasında anlamlı isimlendirmeye yol açarak kod okunurluğu sağlanmıştır.

### **5.3.4 Yapısal Programlama Yapıları**

- Ardışık işlem yapıları: Uzaktan Eğitim Sistemi için fonksiyon ve altyapılar değerlendirilerek kullanıldı.
- Koşullu İşlem Yapıları: Uzaktan Eğitim Sistemi için birçok yerde koşullu işlem kullanma mecburiyeti doğmuştur.



- Döngü Yapıları: Kod karmaşıklığının önüne geçmek için ve alt alta işlemleri tek bir işlem olarak gerçekleştirildi.

## **5.4 Program Karmaşıklığı**

Uzaktan Eğitim Sistemi için program karmaşıklığı hesaplama yöntemlerinden en eskisi olan McCabe karmaşıklık ölçütüdür. McCabe Karmaşıklık Ölçütü kullanma sebebimiz ise, program karmaşıklığı hesaplama konusunda geliştirilen diğer ölçütlerin temelini bu ölçüte dayanmasıdır. McCabe program karmaşıklık ölçütü iki bölüm olarak ele alınır. Bunlar:

## **5.5 Olağan Dışı Durum Çözümleme**

Uzaktan Eğitim Sistemi çalışır bir vaziyet aldıktan sonra sistem üzerinde oluşabilecek önceden öngörülme hata veya durumların çözümünü kolaylaştırmak için ve sistemin çalışma mantalitesini etkilememesi için ve sisteme olağan dışı durumlar esnasında ne yapması gerektiğini empoze ederek hatayı veya olağan dışı durum önlenmiştir.

### **5.5.1 Olağan Dışı Durum Tanımları**

Uzaktan Eğitim Sistemi için kod yapıları olarak olağan dışı hata yapıları için try-catch blokları tanımlanmıştır.

## **5.6 Kod Gözden Geçirme**

Kod okunabilirliğini arttırmak, programın kalitesini yükseltmek amaçlı kod gözden geçirme işlemi için program sınaama metodolojilerinden biri kullanılmıştır.

### **5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**

- Hata bulma
- Temel probleme yaklaşım
- Hata onarımı hedefi
- Kalite ekibinin hata telafisi
- Hata onarma

## **5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**

Programın her yordamının işlevsel özellik gösterdiğini anlamak için sorulan sorulardır.

### **5.6.2.1 Öbek Arayüzü**

- Kullanıcı Kayıt Yapabiliyor mu?
- Kullanıcı Yetkisi Belirlenebiliyor mu?

### **5.6.2.2 Giriş Açıklamaları**

- Eğitimci Ders Ekleyebiliyor mu?
- Yeni öğrenci Kayıt yapabiliyor mu?
- Eğitimci doküman paylaşabiliyor mu?
- Eğitimci canlı ders oluşturabiliyor mu?

## **7.BAKIM**

### **7.1 Giriş**

Proje son aşamaya geldiğinde bakım sürecinin başlaması gerekmektedir. Bakım süreci sürekli devam eden bir süreç olduğu için birden fazla bakım metodları kullanılabilir. Bakım bölümünde yapılan açıklamalar da IEEE 1219-1998 standardı baz alınmıştır.

### **7.2 Tanım**

#### **1-Düzeltilici bakım**

Bir yazılımın 100% sınanabilmesi teorik olarak mümkün olsa da pratikte pek mümkün değildir. Bu nedenle hata ile karşılaşılma olasılığı her zaman vardır. Zamanla ortaya çıkan hataların düzeltilmesi “düzeltilici bakım” olarak adlandırılır.

#### **2-Uyarlayıcı bakım**

Uygulamaya alınmış yazılımlar kurumların, şirketleri veya kişilerin günlük hayattaki işlerini bilgisayar ortamında takip etmesini sağlayan araçlardır. Gün geçtikçe iş süreçlerinde yeni gereksinimler veya var olan gereksinimlerin iptali söz konusu olabilmektedir. Örneğin

mevzuatların deęiřmesi, bir kurum yazılımında ilgili bölümün deęiřmesi anlamına gelmektedir. Bu tür uyarlamalar yazılımda “uyarlayıcı bakım” olarak adlandırılır.

### **3-En iyileyici bakım**

Uygulama yazılımının performansının zamanla arttırılması gerekebilir. Bu tür bakımlar “en iyileyici bakım” olarak adlandırılır.

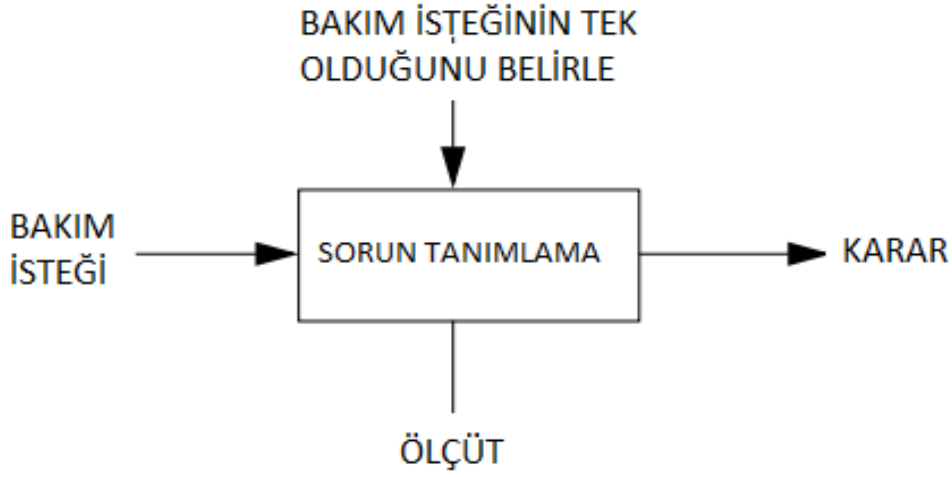
## **7.2 Yazılım Destek Organizasyonu**

Müşteriden gelen istekleri veya kullanıcadan gelen geri dönüşler toplanıp kritięi yapıldıktan sonra yazılım ekibi tarafında uygulama üzerinde guncellemeler yapılabilecek

## **7.3 Bakım Süreç Modeli**

Yazılım bakımı, uluslararası standart belirleme organizasyonu olan IEEE(i triple e) tarafından belirli kriterlere ve aşamalara göre gerçekleştirilmektedir. Bakıma ilişkin standart IEEE 1219-1998 baz alınarak gerçekleştirilmektedir. IEEE tarafından sunulan bakım süreci řu adımları içermektedir:

1. Sorun tanımlama süreci
2. Çözümleme süreci
3. Tasarım süreci
4. Gerçekleştirim süreci
5. Sistem test süreci
6. Kabul test süreci
7. Kurulum süreci



### 1-Sorun Tanımlama Süreci

**Girdi:** Sürecinin temel girdisi, “bakım isteği” şeklindedir. Örneğin;

- Sistemde beklenen ve yeni düzenlemelere ilişkin değişiklikler.
- Yeni fonksiyonel talepler.

**İşlem/Süreç:** Bakım isteği oluşturulduğunda yapılması gereken işlemlerdir.

1. Değişiklik isteğine bir tanım numarası atamak.
2. Değişiklik türünü belirlemek.
3. Değişiklik isteğinin kabul edilmesi ya da ayrıntılı incelenmesine karar verilmesi.
4. Değişiklik isteği ile ilgili *zaman/boyut/işgücü* kestirimi yapılması.
5. Değişiklik isteğinin önceliklendirilmesi.
6. Değişiklik isteğinin diğerleri ile birlikte zaman ve iş planına kaydedilmesi.

Bu adımların uygulanmasına değişiklik talebinde bulunan müşteri, kullanıcı temsilcileri, yazılım mühendisleri ve iş uzmanları ile birlikte çalışılarak karar verilir.

**Denetim:** Sorun tanımlama aşamasında ise değişiklik talebinin daha önceden yapıp yapılmadığı denetlenir ve tek olduğu belirlenir. Mükerrer iş yapmaktan kaçınmak adına daha önceki değişiklik talepleri taranır.

**Çıktı:** Doğrulanmış, geçerlenmiş ve karar verilmiş “bakım İsteğidir”. Bakım isteğinin ayrıntıları bir veritabanında saklanır ve isteğe ait bilgiler şu şekilde olmalıdır:

1. Sorun ya da yeni gereksinimin tanımı,
2. Sorun ya da gereksinimin değerlendirmesi,
3. Başlangıç önceliği,
4. Geçerleme verisi (Düzeltilici bakım için gereklidir),
5. Başlangıç kaynak gereksinimi,
6. Mevcut ve gelecekte kullanıcılar üzerindeki etkileri,
7. Yararlı ve aksak yönleri

**Ölçüt:** Sorun tanımlama sırasında kullanılacak ölçütler

- Bakım taleplerinde kabul edilmeyen madde sayısı,
- Gelen bakım istekleri sayısı,
- Sorunun aşılması için harcanan kaynak ve zaman biçimindedir.

Bu adımların tamamlanmasıyla birlikte çözüm süreci başlatılabilir.

## 2- Çözümleme Süreci

Çözümleme sürecinde, veri tabanında saklanmış ve geçerlenmiş bakım isteği girdi olarak alınır, projeye ilişkin bilgi ve belgeleri kullanarak söz konusu isteğin yerine getirilmesi için gerekli genel plan yapılır.

**Girdi:** Çözümleme sürecinin girdileri:

1. Geçerlenmiş bakım isteği,
2. Başlangıç kaynak gereksinimleri ve diğer veriler ve
3. Mevcut proje yada sistem bilgi ve belgeleri biçimindedir.

**İşlem/Süreç:** Çözümleme süreci temel olarak iki aşamadan oluşur. Bunlar olurluk aşaması ve ayrıntılı çözümleme aşamasıdır.

Olurluk çalışmasında, yapılan değişikliğin etkileri, güvenlik ve emniyet zorunlulukları, insan faktörleri, kısa ve uzun vadeli maliyetler ve yapılacak olan değişikliğin yararları değerlendirilir.

Ayrıntılı çözüm aşamasında, değişiklik isteği için ayrıntılı gereksinim tanımlaması yapılır. Bu çalışmada etkilenen yazılım öğeleri (*yazılım tanımları, yazılım gereksinimleri, tasarım, kod, vb*) belirlenir. Yazılım bileşenlerinin değişmesi gereken kısımları belirlenir. Bu aşamada en az üç düzeyli test stratejisi (*birim testleri, bütünleştirme testleri ve kabul kabul testleri*) oluşturulur. Bu aşamada kullanıcıya en az etki yapacak şekilde değişiklik gereksinimlerinin nasıl karşılanacağı bilgilerini içeren “**Başlangıç gerçekleştirim planı**” da hazırlanır.

**Denetim:** Çözümleme çalışmasının denetiminde gerçekleştirilen işlemler şu şekildedir:

1. Gerekli proje yada sistem bilgi/belgelerine erişimin sağlanması.
2. Önerilen değişikliklerin ve çözümleme çalışmasının teknik ve ekonomik olurluğunun gözden geçirilmesi,
3. Güvenlik ve emniyet konularının tanımlanması,
4. Önerilen değişikliğin, mevcut yazılımla bütünleştirilmesinin dikkate alınması,
5. Proje belgelerinin düzgün olarak günlendiğinin denetimi,
6. Çözümleme belgelerinin düzgün olarak hazırlanmasının sağlanması,
7. Sınama stratejilerinin uygun olarak belirlenmesi.

**Çıktı:** Çözümleme çalışmasının çıktıları:

- Değişiklik isteklerine ilişkin olurluk çalışması,
- Ayrıntılı çözümleme raporu,
- İzlenebilirlik listesini içeren günlenmiş gereksinim tanımları,
- Başlangıç değişiklik listesi,
- Sınama stratejisi,
- Gerçekleştirim planı şeklindedir.

**Ölçüt:** Çözümleme çalışmasında kullanılabilecek ölçütler:

1. Gereksinimlerdeki değişiklik sayısı,
2. Belgeleme hata oranı,
3. Her işlev alanı için gerekecek işgücü,
4. Toplam zaman biçimindedir.

### **3- Tasarım Süreci**

Tasarım aşamasında, değişiklikten etkilenebilecek tüm proje bilgi ve belgeleri üzerinde çalışma yapıp söz konusu bilgi ve belgeler değişiklikle ilgili olarak güncellenir.

**Girdi:** Tasarım çalışmasının girdileri:

1. Çözümleme çalışması çıktıları,
2. Ayrıntılı çözümleme,
3. Güncellenmiş gereksinim tanımları,
4. Başlangıç değişiklik listesi,
5. Sınama stratejisi,
6. Gerçekleştirim planı,
7. Sistem ve proje belgeleri ve
8. Var olan kaynak kodları ve veritabanları biçimindedir.

**İşlem/Süreç:** Tasarım için gerekli temel işlemler aşağıda belirtilmektedir.

1. Etkilenen yazılım modüllerinin tanımlanması,
2. Yazılım modül belgelerinin değiştirilmesi,

3. Yeni tasarım için, güvenlik ve emniyet konularını da içeren test senaryolarının hazırlanması,
4. İlişki testleri tanımlanması,
5. Kullanıcı belgelerinin güncelleme gereksinimlerinin tanımlanması,
6. Değişiklik listesinin güncellenmesi şeklindedir.

**Denetim:** Tasarımın belirlenen standartlara uygunluğunun denetlenmesi gerekmektedir.

**Çıktı:** Bakım tasarımı çalışmasının çıktıları:

1. Gözden geçirilmiş değişiklik listesi,
2. Güncellenmiş tasarım
3. Güncellenmiş test planları,
4. Güncellenmiş ayrıntılı çözümleme,
5. Güncellenmiş gereksinimler,
6. Gözden geçirilmiş gerçekleştirim planı,
7. Risk ve kısıtlar listesi biçimindedir.

**Ölçüt:** Tasarım çalışması için kullanılabilecek ölçütler aşağıda verilmektedir.

- Yazılım karmaşıklığı,
- Tasarım değişiklikleri
- Her işlev alanı için gerekecek işgücü,
- Toplam zaman,
- Sınama yönerge ve plan değişiklikleri,
- Önceliklendirmedeki hata oranları,
- Varolan kodda, eklenen, çıkarılan ve değiştirilen satır sayısı,
- Uygulama sayısı.



#### 4- Gerçekleştirim Süreci

Gerçekleştirim sürecinde, temel olarak tasarım çıktılarını ve kaynak kodları girdi olarak alınmakta ve değişiklik isteğini gerçekleştiren kod parçaları ile güncellenmiş yazılım kodları üretilmektedir. Güncellenmiş yazılıma ilişkin test bilgi ve belgelerinin ve eğitim belgelerinin üretimi de bu süreçte yapılmaktadır.

**Girdi:** Gerçekleştirim sürecinin girdileri:

1. Tasarım çalışması sonuçları,
2. Varolan kaynak kodlar, açıklamalar, belgeler ve
3. Proje ve sistem belgeleri biçimindedir.

**İşlem/Süreç:** Gerçekleştirim sürecinin dört ana işlemi vardır:

1. Kodlama ve birim testleri
2. Bütünleştirme
3. Risk çözümleme
4. Sınama hazırlığı gözden geçirme

Kodlama işleminde, değişiklik isteğini karşılayan yazılım kodları, varolan yazılıma eklenmektedir. İşlem sonucunda elde edilen yeni, değişmiş modüllere birim testleri uygulanmaktadır. Birim test işlemini, bütünleştirme testleri izlemekte, tüm sistem yeniden test edilmektedir. Uygulamadaki riskleri gidermek amacıyla, gerçekleştirim aşamasında sürekli risk çözümleme yapılmaktadır.

**Denetim:** Gerçekleştirim sürecinde oluşturulacak denetim yapısı, aşağıdaki özellikleri sağlamalıdır:

- Belirlenen standartlara uygun olarak kod ve yazılım gözden geçirmeleri yapılması,
- Birim ve bütünleştirme testleri ile ilgili bilgilerin derlenmesi ve kaydedilmesinin sağlanması,
- Test belgelerinin güncellenmesi ve oluşturulmasının sağlanması,
- Test hazırlıklarının gözden geçirilmesi sırasında risk çözümlemenin yapılması,
- Yeni yazılımın, yazılım ortam yönetimi altında kaydedilmesi ve denetlenmesinin sağlanması,

- Teknik ve eğitim belgelerinin güncellenmesi,

**Çıktı:** Gerçekleştirim süreci aşağıdaki çıktıları vermelidir:

1. Güncellenmiş Yazılım,
2. Güncellenmiş tasarım bilgi/belgeleri,
3. Güncellenmiş sınaama belgeleri,
4. Güncellenmiş kullanıcı belgeleri,
5. Güncellenmiş eğitim kılavuzları,
6. Riskler ve kullanıcılara etkileri,
7. Sınaama hazırlığı gözden geçirme raporu

**Ölçüt:** Gerçekleştirim çalışmasında kullanılabilecek ölçütler kodlama ile ilgili olması gerekmektedir. Bu nedenle teknik ölçütler kullanılmaktadır:

1. Değişiklik oranı
2. Hata oranı

## **5- Sistem Test Süreci**

Değişikliklerin mevcut yazılıma yansıtılmasından sonra elde edilen yeni yazılım sürümünün belirlenen standartlara uygun olarak tümüyle bütünleşik sistem üzerinde testlerin yapılması gerekmektedir. Sistem testlerinin, kullanıcı ve üretici ekiplerin tanıklığında bağımsız bir yapı tarafından gerçekleştirilmeleri önerilmektedir.

**Girdi:** Sistem test sürecinin girdileri:

1. Test hazırlık raporu
2. Belgeler
3. Sistem test planları,
4. Sistem testleri,
5. Sistem test yönergeleri,
6. Kullanıcı kılavuzları,
7. Tasarım
8. Güncellenmiş sistem biçimindedir.

**İşlem/Süreç:** Sistem testleri, tümüyle bütünlük bir sistem üzerinde yapılmalıdır. Bu aşamada, işlevsel sistem testi, arayüz testi, regresyon testi ve test hazırlık raporunun gözden geçirilmesi işlemleri yapılır.

**Denetim:** Sistem testleri, üretici ve kullanıcılardan bağımsız bir grup tarafından gerçekleştirilmelidir. Yazılım kodları ve her türlü bilgi belge, yazılım ortam yönetimi tarafından saklanır.

**Çıktı:** Bu aşamanın temel çıktıları:

1. Tümüyle test edilmiş bütünlük bir sistem,
2. Test raporu,
3. Gözden geçirilmiş test hazırlık raporu şeklindedir.

**Ölçüt:** Bu aşamada kullanılabilecek ölçütler, üretilen ve düzeltilen hata oranlarıdır.

## **6- Kabul Test Süreci**

Kabul test süreci, kullanıcılar ya da kullanıcı temsilcileri tarafından gerçekleştirilen bir süreçtir. Kullanıcıların, değişiklikleri içeren yeni yazılımı test etmeleri ve kabul etmeleri beklenmektedir.

**Girdi:** Kabul test sürecinin girdileri:

1. Gözden geçirilmiş test hazırlık raporu,
2. Tümüyle bütünlük sistem,
3. Kabul test planları,
4. Kabul testleri ve
5. Kabul test yönergeleri biçimindedir.

**İşlem/Süreç:** Kabul test işlemleri:

1. İşlevsel kabul testlerinin yapılması,
2. Birlikte çalışabilirlik testleri,
3. Regresyon testleri biçimindedir.

**Denetim:** Kabul testleri sırasında denetimi aşağıdaki işlemleri içermektedir.

1. Kabul testlerinin uygulanması,
2. Sınama sonuçlarının raporlanması,
3. İşlevsel denetim yapılması,
4. Yeni sistemin oluşturulması,
5. Kabul test belgelerinin yazılım konfigürasyonuna yerleştirilmesi.

**Çıktı:** Kabul testlerinin çıktıları, yeni sistem, işlevsel konfigürasyon denetim raporu ve kabul sınaması raporudur.

**Ölçüt:** Bu aşamada kullanılabilecek ölçütler: üretilen ve düzeltilen hata oranlarıdır.

## **6- Kurulum Süreci**

Kurulum süreci, geliştirilen ya da değiştirilmiş yeni yazılım sürümünün, uygulama sahasına aktarılma işlemlerini içerir.

**Girdi:** Bu sürecin temel girdisi, tümüyle sınanmış ve kabul edilmiş yeni yazılım sürümüdür.

**İşlem/Süreç:** Bu sürecin işlemleri:

- Fiziksel ortam denetiminin yapılması,
- Kullanıcıların bilgilendirilmesi,
- Mevcut sistemin yedeklerinin alınması,
- Kullanıcı tarafında kurulum ve eğitimlerin yapılması şeklindedir.

**Denetim:** Denetim işlemleri:

1. Fiziksel ortam denetiminin yapılması,
2. Sistem ile ilgili bilgi ve belgelerin kullanıcıya ulaştırılması,
3. Sürüm Tanımlama raporunun tamamlanması,
4. Yazılım konfigürasyon ortamına aktarımın sağlanması şeklindedir.

**Çıktı:** Bu sürecin temel çıktıları, fiziksel ortam denetim raporu ve Sürüm tanımlama raporudur.

**Ölçüt:** Bu süreçte kullanılabilecek ölçüt, belgeleme değişiklikleridir.

Tüm süreçlerin tamamlanmasıyla yazılım bakım döngüsü tamamlanmış ve müşteri istekleri eksiksiz yerine getirilmiş olmaktadır. Tekrar belirtmekte fayda var ki bu standartlar dizisi IEEE tarafından belirlenmiştir.

## **8. SONUÇ**

Sonuç olarak uzaktan eğitim sistemini kullandığımızda örgün eğitim sisteminin dezavantajlarının (maliyet, ulaşım güçlüğü, farklı bir ülkede, şehirde öğretim, bedensel engel sorunları) pek çoğu ortadan kalkacak.

Tüm kullanıcılar oturdukları yerden kullanacakları bu sistem sayesinde eskisi kadar yorulmayacak.

## **KAYNAKLAR:**

- 1-) M. Hakan Çetiner, Çiğdem Gencel, Y. Murat Erten -ODTÜ Enformatik Enstitüsü, Bilişim Sistemleri ABD
- 2-) ALKAN, C. (1998) “Eğitim Teknolojisi ve Uzaktan Eğitimin Kavramsal Boyutları” Uzaktan Eğitim Yaz 1998/Kış1999,
- 3-) DEMİRAY, U.(1999)”Kuruluşunun 5. Yılında Açıköğretim Lisesi İle İlgili Çalışmalar Kaynakçası 1992-1997 “Eğitim Teknolojileri Genel Müdürlüğü Yayınları, Ankara
- 4-) DEMİRAY, U.(1999) “Açıköğretim Fakültesi Mezunlarının Sektördeki Konumları” Uzaktan Eğitim, Yaz 1999, 3-22
- 5-) ÖZFIRAT, A., YÜRÜKER, S. (1999) Uzaktan Eğitim Materyallerinde Öğretim Tasarımı. Bilişim Teknolojileri Işığında Eğitim Konferansı ve Sergisi Bildirisi. 13-15 Mayıs. Ankara
- 6-) Songül Gültekin, Nurşen Gezen, Ayşe Fatma Tüney, Fatma Can, Kevser Gündoğar Uzaktan Eğitim nedir?
- 7-) Emin İbili, Fatih Bayram, Fidan Hakkari, Mahmut Kantar, Mevlüt Doğan SCORM Uyumlu Eğitim Yönetim Sisteminin Tasarlanması ve Üniversite Bazında Uygulanması Akademik Bilişim’09 - XI. Akademik Bilişim Konferansı Bildirileri 11-13 Şubat 2009 Harran Üniversitesi, Şanlıurfa