

## 8. HAFTA: YAZILIM TASARIMI

### KULLANICI ARAYÜZÜ TASARIMI

Bilgisayar tabanlı sistemler insanların işlerini kolaylaştırmak üzere geliştirilirler. Ne kadar mükemmel tasarlanmış ve gerçekleştirilmiş olursa olsun, eğer bir sistem kullanıcısına zor anlar yaşıyorsa tam başarılı sayılamaz. Bu nedenle, sistemin insanlarla olan arayüzünün çok etkin ve kullanışlı olarak, verimliliği artıracak şekilde, kullanıcı dostu olarak tasarlanması gereklidir. Bilgisayar tabanlı sistem arayüzünün önemi çok geniş uygulama alanları düşünüldüğünde daha iyi ortaya çıkar. Bir programlanabilir fırın veya çamaşır makinesi, bir cep telefonu, bir tıbbi cihaz, bir uçuş kontrol altsistemi gibi gömülü sistemlerin arayüzleri, bir veritabanı işletim sisteminin arayüzü, hele hele günümüzün en yaygın bilgisayar uygulamalarından biri olan İnternet tabanlı programların arayüzleri İnsan mühendisliğinin ne kadar önemli olduğunu çok güzel ortaya koyan örnekler arasındadır. Bir sistemin arayüzünü öğrenmek ve etkin bir şekilde kullanmak ne kadar kolay olursa o sistemin yetenek ve işlevlerinden yarar sağlamak da o kadar artar.

### 1. İnsan-Bilgisayar Etkileşimi

**İnsan ile bilgisayar arasındaki etkileşim genellikle görsel, işitsel ve dokunmatik olarak gerçekleşir.** Ekrandaki bilgiler görsel iletişimi sağlarken, mikrofon ve hoparlör ile ses bilgileri alınır veya verilir. Bunlar yanında, girdi sağlamak için klavye, işaret aygıtı (mouse veya trackball), oyun çubuğu (joy-stick), çeşitli özel tuşlar ve giriş aygıtları kullanılır. Tüm bunlara *insan-makine arayüzü* (Human-Machine interface - HMI) adı verilmektedir (bazen Man-Machine Interface - MMI kısaltması da kullanılır). Etkileşimin daha çok bilgisayarla yapıldığı durumlarda bu isim *insan-bilgisayar arayüzü* (Human-Computer Interface - HCI) adını almaktadır. Bu şekilde, bir insan, bilgisayardan çeşitli bilgileri alabilir, kendi belleğinde saklayabilir, neden-sonuç ilişkilerini değerlendirebilir ve kendisi yeni bilgiler girebilir .

**Günümüzde, insan-bilgisayar arasındaki arayüzü daha da iyi duruma getirebilmek için bilimsel ve teknik çalışmalar sürdürülmektedir.** Örneğin, bir zamanlar siyah-beyaz olan ekranlar sonradan, 16 renk, 256 renk ve en sonunda da milyonlarca rengi destekleyerek gerçeğe en yakın görünüşü vermeye başlamışlardır. Aynı zamanda ekranların çözünürlüğü, canlılığı ve büyüklüğü artmış, insan gözünü

yormayacak (en az 72 Hz yatay tarama standardı gibi) ve sağığı tehdit etmeyecek (düşük radyasyon yayımı) hale gelmişlerdir.

Bilgisayar tarafından ekranda sergilenen bilgiler renk, şekil, büyüklük ve hareket bakımından göz ile algılanıp beyin tarafından değerlendirilerek insan tarafından algılanmaları sağlanmaktadır. Gittikçe daha rahat yani ergonomik yapıya ulaştırılan klavye ve işaretçi aygıtlarla bilgisayar ile iletişim kurmak, komutlar vermek, bilgileri seçmek daha da kolay olabilmektedir. Gelecekte sesli komutlarla da anlaşabilmek tabii ki mümkün olabilecektir. Tüm giriş/çıkış aygıtlarının iyi bir uyum içinde kullanılmasıyla insanlarla makine arasındaki büyük engeller yavaş yavaş aşılacaktır. Belki de gün gelecek düşüncelerimizle iletişim kuracağız ve bilgisayarları aklımızın bir uzantısı olarak kullanacağız.

Tüm bilgisayarlı sistem kullanıcılarının aynı anlayış derecesine sahip olduklarını kabul etmek yanlış olur. Bu düşünceyle, kullanıcıları üç sınıfa ayırmak doğru olur:

**Deneyimsiz:** Sistem hakkında hiç bilgisi olmayan ancak bir miktar bilgisayar kullanabilen kişilerdir.

**Az deneyimli:** Sistem hakkında bir miktar bilgisi olan ancak az miktarda kullanım pratiğine sahip kişilerdir.

**Çok deneyimli:** Hem sistemi hem de kullanımını iyi bilen, hatta daha hızlı kullanabilmek için kestirmeler arayan kişilerdir.

## 2. İnsan-Bilgisayar Arayüz Tasarımı

**Grafiksel sergileme ortaya çıkmadan önce bilgisayarlarla etkileşim yalnızca klavye ile komut satırından yapılmaktaydı.** Öğrenmesi oldukça zor ve hata yapma olasılığı çok yüksek olan komut satırı ile işlem yapmak artık geride kalmış sayılır. Ancak, Unix tabanlı işletim sistemleri gibi yaygın bazı sistemler hala komut satırı kullanmakta, büyük bir kullanıcı kitlesi de bu komutlarla daha etkin kullanım sağlayabilmektedir. Fakat uygulama yazılımları çoğunlukla grafiksel kullanıcı arayüzlerine sahip olacak şekilde geliştirilmektedir.

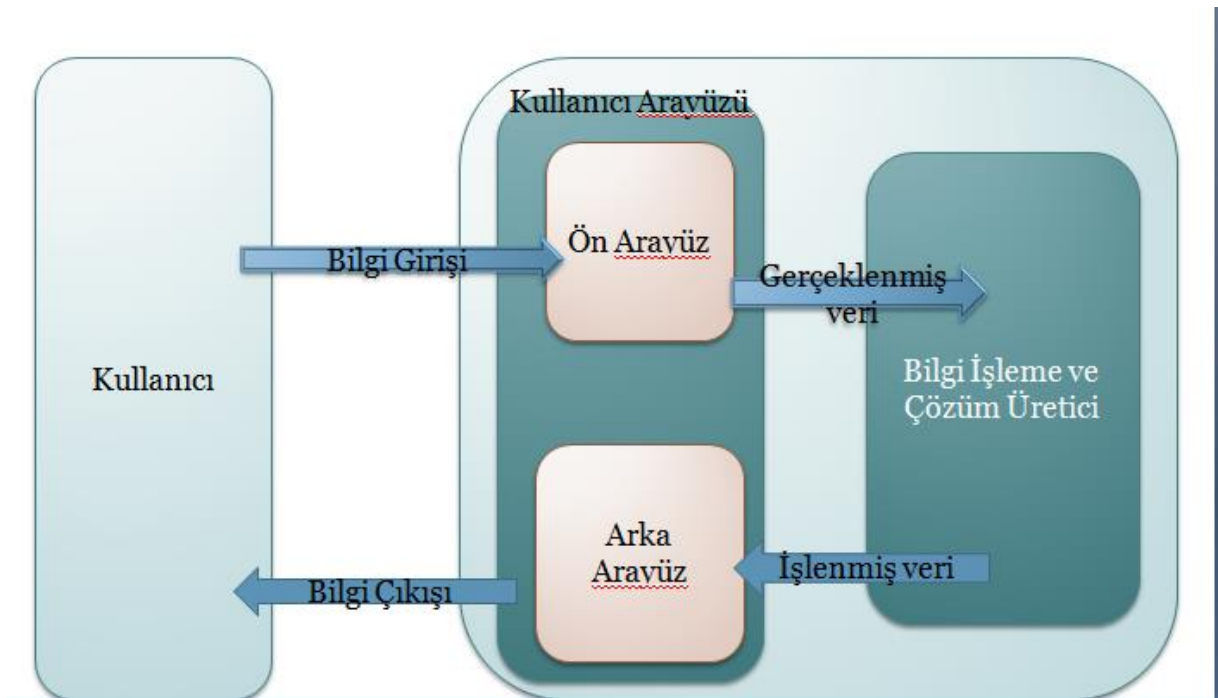
**Günümüz standartlarında bir sistem için kullanıcıyla doğrudan etkileşimi sağlayan en önemli yazılım ögesi grafik tabanlı pencerelerdir.** Bir pencere yönetim sistemi ile denetlenen çeşitli grafiksel birimler ile kullanıcının gözüne, anlayışına en uygun bilgi sergileme, en çabuk ve en doğru şekilde bilgi girme olanağı sağlanır. Standart bir pencere işleticisi çalıştırabilen, yeterli grafik özelliklere sahip bilgisayarlar üzerinde grafiksel arayüzler insanlara çok daha yakın olabilmektedir. Üzerinde resim veya simge bulunan küçük şekiller (ikonlar), renkli çubuklar, metin veya veri giriş alanları, bilgi sergileme alanları, seçme menüleri, tıklama veya seçme

tuşları gibi grafik öğeleri ile bir arayüz tasarımı artık hem bilim hem de sanat içerik hale gelmiştir.

**Günümüzdeki bilgisayar işletim sistemlerinden bir kısmı tamamen pencere sistemine dayalı olarak çalışmakta iken bir kısmı da ek bir pencere yönetim sisteminin (Linux ile X-Window Sistemi, vb.) çalıştırılmasını gerektirir.** Pencere yönetim sistemleri özel olarak geliştirilebileceği gibi yaygın standartlardan biri de seçilebilir. Ancak dikkat edilmesi gereken nokta, seçilen standarda uygun olarak yazılım geliştirme olanağı sağlayan yazılım geliştirme paketlerinin (Software Development Kit) kullanılmasıdır. Zira, kendi içlerinde standart olsalar da birbirleri arasında uyum olmayan pek çok pencere yönetim sistemi bulunmaktadır.

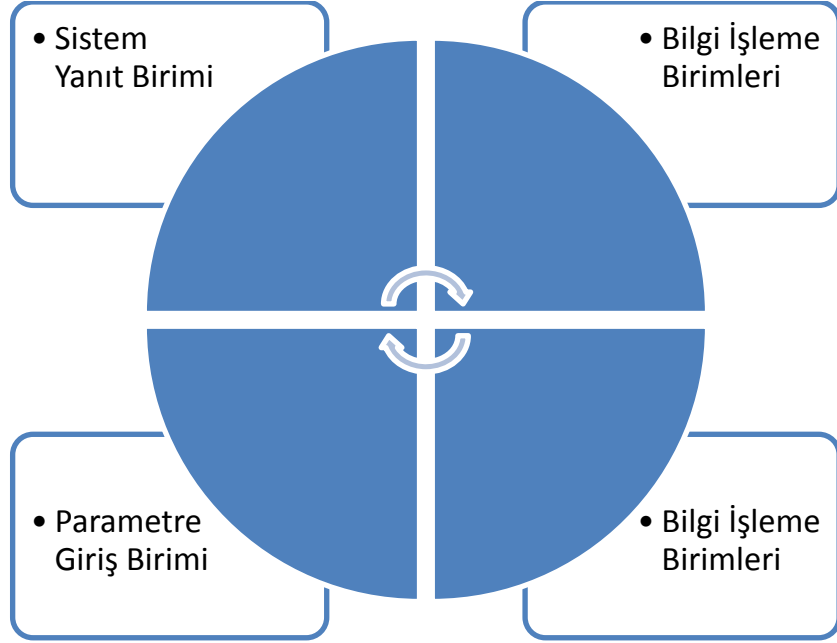
Herhangi bir etkileşimli yazılım, iki öğeden oluşur. Burada, **kullanıcı**, yazılıma girdi sağlayan ve çıktıları değerlendiren işletmendir. **Bilgi işleme ve çözüm üretici**, kendisine ulaştırılan gerçekleşmiş bilgilerin işlendiği ve eldeki problemin çözülmesi için gereken bilgilerin üretildiği öğedir.

**Kullanıcı arayüzü**, etkileşimli yazılım sistemlerinde, kullanıcı ile bilgi alışverişini sağlayan öğedir. Arayüz, kendi içinde, ön ve arka arayüz olmak üzere ikiye ayrılır. Ön arayüz, kullanıcının yazılıma bilgi girmesini sağlayan kısımdır. Örneğin, gün, ay, yıl olarak girilen tarih bilgisini sayısal hale getirerek çözüm üreticisine ulaştırır. Arka arayüz, çözüm üreticisi tarafından üretilen işlenmiş veriyi kullanıcının görmek ya da almak istediği biçime çeviren kısımdır. Sistemin ürettiği kayan nokta tipindeki zaman bilgisi arka arayüz tarafından saat, dakika ve saniye gösterimine çevrilerek sunulur.



## Arayüz Yazılım Mimarisi

Bir uygulama yazılımı tek bir yürütülebilir yazılım biriminden, yani programdan oluşabileceği gibi birden fazla birimden de oluşabilir. Pencere sistemi ile uyum sağlayan arayüz yazılımı ya ayrı bir birim halinde geliştirilir ya da aynı birimin içine gömülür. Tek birimden oluşan yazılımlarda, bilgi işleme birimleri ile arayüz birimleri aynı bağlam (context) içinde yer alırlar.



Birden fazla işlemcisi olan bilgisayarlarda veya tamamen dağıtık mimaride çalışabilmek üzere ayrık mimari kullanılır. Bu mimarinin yararı, bilgi işleme ile arayüz birimlerini birbirlerinden ayırarak hem geliştirme hem de yürütme bağımsızlığı sağlanır. Özellikle yüksek başarımlı uygulamalarda, bilgi işleme yeteneğinin arayüz ile kısıtlanmaması arzu edilir. Yeterli iletişim altyapısı bulunduğu takdirde, bilgi işleme ile arayüz yazılımları farklı işlemcilerde, hatta farklı özelliklere sahip bilgisayarlarda çalıştırılarak hem bağımsızlık, hem de verim artışı sağlanabilir.

### Arayüz Yazılım Birimleri

Arayüz donanımları cinsleri ne olursa olsun mutlaka ana sistemle tümleştirilmeleri gerekir. Tümleştirme işlemi için mutlaka birer arayüz donanımı ve onları etkinleştiren birer yazılım birimi kullanılmalıdır. Örneğin, ekranı uygun grafik özellikte sürmek için yeterli güce sahip bir grafik kartı kullanılmalıdır. Bu kartın da uygun bir sürücü yazılımı ile işletim sistemine tanıtılması gereklidir. Bir işaret aygıtı

yine özel bir yazılımla bilgisayarın belirli bir kapısını (port) kullanarak bir sürücü ile sisteme tanıtılır. Özel tuş takımları için bir mikrokontrolcü ile seri kapı üzerinden haberleşen bir aygıt sürücüsü kullanılır. Aygıt sürücüler, verdikleri arayüzleri kullanan arayüz yazılım birimleri aracılığıyla uygulama yazılımlarının rahatça giriş/çıkış yapabilmelerini sağlarlar.

Grafiksel arayüz yazılımlarının tasarımına mutlaka özel dikkat gösterilmeli, uygulama alanının gereksinimleri dikkate alınarak en hızlı şekilde kullanıcı ile iletişim kurulması sağlanmalıdır. Örneğin, bir hava trafik kontrol merkezinin bilgisayarlarında radar videosu, bölge haritası gösterilmeli, yerdeki ve havadaki uçakların sürekli yenilenen konumları, kimlik ve çağrı adları, rota ve süratleri sergilenebilmelidir.

Sistemin kullanıcı arayüz yazılımları grafik özellikleri daha uygun olan ayrı bilgisayarlar üzerinde çalıştırılabileceği gibi, sistem tasarımına ve yazılım mimarisine bağlı olarak, ana sistemle beraber, aynı bilgisayar üzerinde de çalışabilirler.

## Arayüz Tasarım İlkeleri

Kullanıcı arayüz tasarımı genellikle tasarımcının deneyimine ve uygulama alanını ne kadar tanıdığına bağlıdır. Yazılım geliştirmeye başlamadan önce uygulama alanının özellikleri dikkate alınarak arayüz standartlarının belirlenmesi büyük yarar sağlar. Bundan sonra da her bir arayüz yazılım birimini tasarlayan ve geliştirenler bu standartlara uyarak en etkin, en dostça arayüzü tasarlamaya ve gerçekleştirmeye çalışırlar.

1. Yüksek Nitelik
2. Kullanıcı Dostluğu
3. Güvenilirlik
4. Yardımlar
5. Hatalar ve Uyarılar
6. Yapısal Özellikler
7. Kullanıcı Arayüz Geliştirme Süreci

Tüm sistem yazılımı belirli bir disiplinin uygulandığı geliştirme süreci ile gerçekleştirilir. Kullanıcı arayüz yazılımı da ana sistem yazılımının temel öğelerinden biridir. Kendi içinde bir süreç uygulanarak geliştirilir. Bu süreç yine standart yazılım geliştirme sürecine benzer ve dört aşamadan oluşur.

## 1.Çözümleme

Bilgisayar tabanlı bir sistem genellikle elle yapılan işleri otomatik hale getirmek üzere tasarlanır ve geliştirilir. Otomasyonun ne derece olacağı, işletmenlerin hangi düzeyde devreye gireceği, sistemin girdi ve çıktıları sistem çözümlemesi sırasında belirlenir. Bu çözümleme sırasında, sistemin kullanıcı arayüz isterleri de belirlenir. Çözümleyici bu amaçla sistemin amaçlarını ve temel işlevlerini tanımlar. Her amaca karşılık düşen eylemleri ve bunların kullanıcı arayüzünden nasıl başlatılacaklarını belirtir. Eylemler sırasında sistemin içinde bulunması gereken durumları tanımlar. Kullanılacak denetim aygıtlarını ve düzenekleri, bu düzeneklerin sistem durumunu nasıl etkileyeceğini açıkça ortaya koyar. Kullanıcıya sistem durumu hakkındaki bilgilerin nasıl sunulacağını, uyarıların nasıl yapılacağını belirtir.

Bu çözümleme sonunda sistem belirtiminin arayüz tanımlaması ortaya çıkar. Arayüz şekillerinin karmaşıklığı aynı zamanda sistemin karmaşıklık derecesini de ortaya koyar. Etkileşim sırasında kullanılan eylemlerin, komutların ve sistem durumlarının sayısı ne kadar yüksekse kullanıcının hatırd tutması gereken bilgi miktarı da o kadar yüksek olacaktır. Bu nedenle sistemin genel tasarımı sırasında arayüzün en azından ön tasarımına da yer verilmelidir.

## 2.Tasarım

**Sistemin arayüz tasarımı sırasında kullanıcıya ne tür arayüzler sağlanacağı, giriş ve çıkışların hangi aygıtlarla, ne şekilde yapılacağı, hata ve uyarı iletilerinin nasıl verileceği belirlenir.** Özel bir donanım ile arayüz gerçekleştirilecekse, donanım tasarımı aşamasında gerekli çalışma yapılmalı, uygulama alanının özelliklerine göre en uygun giriş/çıkış aygıtları seçilmelidir. Yalnızca ekran, klavye ve işaretçi aygıtından oluşan standart giriş/çıkış ile denetlenen bir grafiksel kullanıcı arayüzü bulunacaksa bunun gerekli tasarımı yazılım geliştirme aşamalarında yapılmalıdır. Grafiksel kullanıcı arayüzünü gerçekleştirmeden önce bir ilkörneğini (prototip veya ön ürün) yapmak ve bunu kullanıcı ile gözden geçirmek her zaman büyük yarar sağlar.

Kullanıcıya dostça ve standart bir görünüm sağlamak için tüm uygulamaların aynı standart şablon üzerine oturtulması sağlanmalıdır. Birçok pencere yönetim sistemi ile çalışan uygulamalarda, menü çubuğu, araç çubuğu, gövde ve durum çubuğu standart olarak bulunmakta, bunlara ait ayarlar pencere sistemi tarafından yapılmaktadır.

## 3.Gerçekleştirim

Arayüz tasarımı çeşitli çizim araçlarıyla yapılabileceği gibi gerçekleştirimde kullanılacak geliştirme aracıyla da yapılabilir. Bu araçlar, önceden geliştirilmiş çeşitli grafiksel şablonlar, tuşlar, menüler ve daha pek çok grafik nesneler sağlayarak çok hızlı geliştirme olanağı sunmaktadırlar. Ne yazık ki, yaygın olarak kullanılan pek çok gereç belirli bir pencere yönetim sistemini desteklediğinden taşınabilirlik özelliği

son derece zayıftır. Bu araçlar aynı zamanda otomatik olarak kod üretirler. Ya aracın sağladığı arayüz üzerinden ya da üretilen kod içine elle yazarak yordamların gerçekleştirecekleri eylemler kodlanır. Bir arayüz yazılımının pencere kısmını geliştirmiş bir araç ile birkaç saatte tamamlayıp çalıştırmak mümkündür. Ancak bu arayüzü gerçek sistemle tümleştirmek, tuşlara basıldığında veya menülerden seçimler yapıldığında birtakım işlemler ve denetimler yaptırmak, hatâlara karşı dayanıklılık sağlamak için önemli miktarda kod yazılması gereklidir.

#### 4. Test

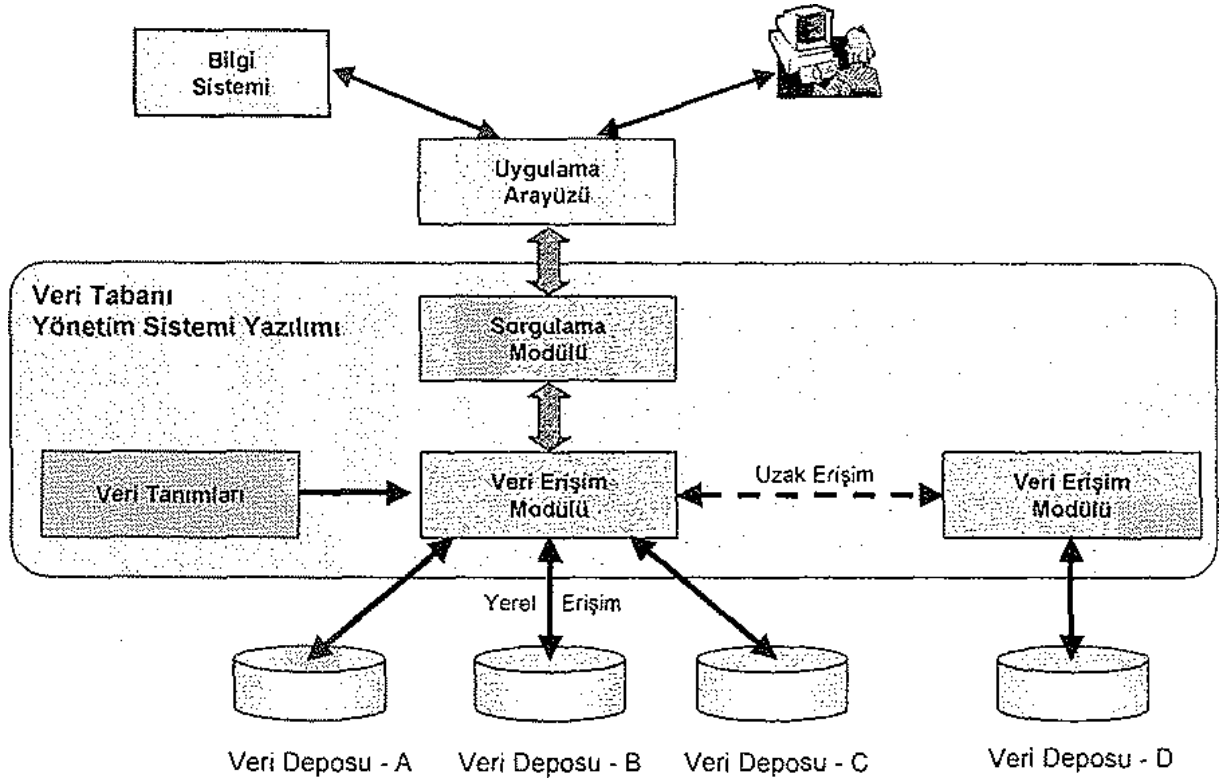
Arayüz testlerini ilkörnekleme arayüz testlerinden ayırmak gereklidir. Arayüz ile beraber yapılan işlevsel testler aslında tüm sistemin testi demektir. Zira, verilen komutlar, eylemler aslında sisteme bir giriştir, sergilenen çıktılar da sistemin yanıtı veya çıkışıdır. Bu giriş ve çıkışların doğruluğu kullanıcı için bir kabul testi sayılır.

İşlevsel testler yanında arayüzün gürbüzlüğü (robustness) mutlaka sınanmalıdır. Kullanıcının hiçbir zaman basmayacağı varsayılan tuş kombinasyonları hiç olmadık yerde ortaya çıkıp tüm yazılımı çökertebilir. Örneğin, klavye üzerine konmuş bir kitap uygunsuz bir tuş grubunu çalıştırıp arayüzün kilitlenmesine neden olabilir. Sistemi bilmeyen bir kullanıcı uygunsuz bir menü seçeneğini kullanarak sistemin durumunu değiştirebilir. İşte bu gibi durumların denenmesi, mantıklı senaryoların dışında rastgele testler yapılması, sistemi çökertmeye yönelik girişimlerde bulunulması aslında sistemin sağlamlığının sınanması için çok önemlidir.

## VERİTABANI UYGULAMALARI VE TASARIMI

Büyük miktardaki verileri belirli bir amaç için bir araya toplamak ve işlemek üzere veritabanı yönetim sistemleri kullanılmaktadır.

Tipik bir veritabanı yönetim sistemi yazılım mimarisi Şekilde gösterilmektedir:



Veritabanı kullanan bilgi sistemi ya da bir işletmen, uygulama arayüzü ile sisteme erişim sağlar. Erişim bir başka yazılım sisteminden yapılacaksa, bir dinamik kütüphane ya da bir İleti arayüzü kullanılabilir. Erişim İnsan tarafından yapılacaksa uygun bir kullanıcı arayüzü bulunur. Bu arayüz grafik yapıda olabileceği gibi basit bir komut satırı şeklinde de olabilir; ancak, özel komutlar veya amaca uygun menüler kullanılabilir.

Arayüz üzerinden asıl veritabanı yönetim sistemi yazılımına erişilir. Bu yazılımın **ilk düzey modülü** sorgulama görevini yerine getirir. Kullanıcı arayüzünden aldığı komutlara göre sistemin tanıyacağı standart bir sorgulama yaratır ve bunu veri erişim modülüne gönderir. Daha alt düzeyde bulunan **erişim modülü**, sorguyu açıp ne tür veri istendiğini ve bu verileri nerede bulabileceğini değerlendirir. Bunun için de genellikle ayrı bir tabloda veya bir veritabanı içinde bulunan veri tanımlarını kullanır. Buradan aldığı tanımlarla yerel ya da uzak veri tabanlarına erişerek fiziksel saklama yerlerinden verileri toplar, sonuçları **sorgulama modülüne** ulaştırır. Sorgulama modülü kullanıcının sorgusuna uygun olan verileri süzer ve uygulama arayüzüne verir. Arayüz, verilerin miktarına göre uygun bir yapılanmayı sağlayarak (ekranda kayıp giden bir defalık bilgi listesi halinde değil, sonuçlar üzerinde gezinti yapılabilecek şekilde) istenen verileri kullanıcı sisteme gönderir veya işletmene gösterir.

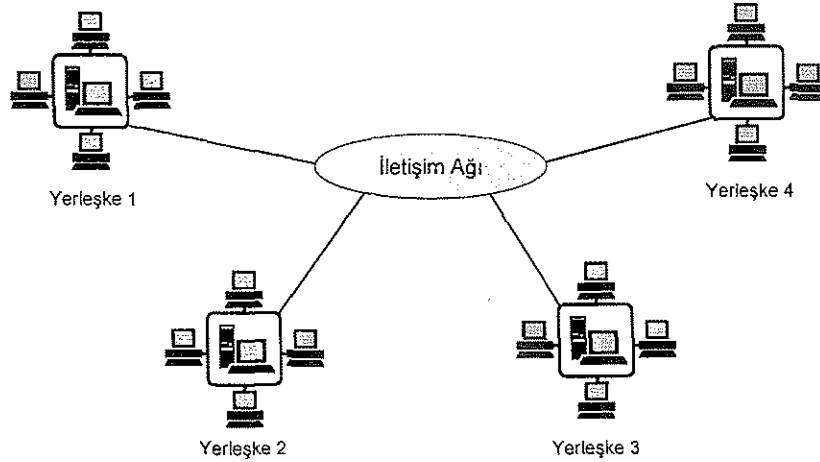
Veritabanı yönetim sistemleri için çeşitli programlama ve tanımlama dilleri geliştirilmiştir. Kullanıcı arayüzü, veri tanımlama, sorgulama ve veri saklama için farklı diller kullanılabilir. Yapılandırılmış Sorgu Dili (Structured Query Language - SQL) bunlardan biridir.



## Sınıflandırma

Veritabanı yönetim sistemleri kullanıcı sayılarına göre, **tek kullanıcılı veya çok kullanıcılı olarak ikiye ayrılabilir**. Ayrıca, bulundukları konumlara göre **de merkezi ve dağıtık ohtak ikiye ayrılabilirler**. Merkezi yapıda veri tabanları tek bir bilgisayar üzerinde iken, dağıtık yapıda veri tabanları fiziksel hatta coğrafi olarak ayrı bilgisayarlar üzerinde bulunabilirler.

Şekilde dağıtık bir yapı örneği göstermektedir:



Yerel alan veya geniş alan bilgisayar ağları üzerinde oluşturulan dağıtık veri tabanlarının en önemli özellikleri ise tüm dağıtık veri tabanları arasında sürekli güncelleme yaparak tutarlılığı sağlamalarıdır.

Veritabanı yönetim sistemleri genel amaçlı veya özel amaçlı olabilirler. Genel amaçlı olanlar çok çeşitli hizmetler verebilirken, özel amaçlı olanlar belirli bir başarıım düzeyini sağlamak ve belirli bir amaca hizmet etmek zorundadırlar.

Yapılarına ve kullandıkları veri modellerine göre veritabanı yönetim sistemleri çeşitli sınıflara ayrılabilirler:

- İlişkisel (relational)
- Ağ (network)
- Sıradüzensel (hierarchical)
- Nesne yönelimli (object-oriented)

## Sistem Yaşam Süreci

Veritabanı sistemleri genellikle daha büyük bilgi sistemlerinin parçası olarak yer alırlar. Bu sistemlerin geliştirilmesi de diğer sistemler gibi aynı süreci izlerler. Tipik bir sistemin yaşam çevrimi şu şekildedir:

### Sistem tanımı

veritabanı sisteminin kapsamı, kullanıcıları ve uygulamaları tanımlanır.

### Tasarım

Veritabanı sisteminin mantıksal ve fiziksel tasarımı yapılır.

### Gerçekleştirim

Sisteme ait kavramsal tanımlamalar yapılır, içsel ve dışsal veri tanımlamaları yapılır. Boş veritabanı dosyaları yaratılır. Yazılım modülleri geliştirilir.

### Veri yükleme

veritabanı sistemlerine büyük miktarda veriler ya elle girilir ya da daha önce var olan aynı ya da farklı biçimdeki veriler gerekli çevrimler yapılarak aktarılır.

### Test

Oluşturulan sistem kullanıcı isteklerine göre sınılanır, doğrulama ve geçerleme işlemleri yürütülür.

### Kullanım

Sistem kullanıma sunulur.

### İzleme

Sistemin büyüyen veri miktarları sürekli İzlenir, fiziksel sınırlar göz önünde bulundurularak düzenlemeler veya kaydırmalar yapılır, yedekler alınır.

### Bakım

Sistem üzerinde zaman içinde değişen istelere göre değişiklikler uygulanır.

## *Tasarım*

Bir veritabanı sisteminin tasarımı birçok sistemin yazılım tasarımından daha farklıdır. Tipik bir tasarım süreci aşağıdaki adımlardan oluşur:

### İsterlerin belirlenmesi

Hedef kullanıcı kitlesi ile uygulama alanı tanımlanır; işletim ortamı, veri ve bilgi işleme gereksinimleri belirlenir.

### Veritabanı tasarımı

Veritabanı isterlerine göre verilerin yapıları, anlamları, ilişkileri, bağımlılıkları ve uyulması gerekli kısıtlamalar üst düzeyde, kavramsal olarak ve belirli bir sisteme bağlı olmaksızın modellenir. Yukarıdan aşağıya veya aşağıdan yukarıya olacak şekilde veri modelleri, yani varlık ilişki diyagramları oluşturulur. Yukarıdan aşağıya modelleme yaklaşımında genelden ayrıntıya doğru, aşağıdan yukarı modelleme yaklaşımında önce ayrıntılı veri yapıları hazırlanır; sonra bunlar arasındaki ilişkiler tanımlanır. Her model daha sonra birleştirilir ve dengeli hale getirilir. Bundan sonra veritabanı üzerinde yapılacak ara-işlemler giriş/çıkış verileri, içerideki işlevsellik ve akış denetimi şeklinde tanımlanır.

## *Veritabanı yönetim sistemi seçimi*

Eğer veritabanı yönetim sistemi özel olarak geliştirilmeyecekse, hazır olarak kullanımı mümkün olan sistemler arasından teknik, ekonomik ya da yönetsel kararlara göre uygun bir veritabanı yönetim sistemi seçilir. Bu seçimde edinme, bakım, personel, eğitim, işletim, veritabanı yaratma ve dönüştürme maliyetleri, verilerin yapılması, personel deneyimi, üreticilerin teknik desteği, donanım ve yazılım gereksinimleri gibi konular dikkate alınmalıdır.

### Mantıksal tasarım

Daha önce oluşturulmuş veri modelleri, seçilen veritabanı yönetim sistemine göre kavramsal modellere dönüştürülür. Eğer veritabanı yönetim sistemi belirli bir tanımlama dili kullanıyorsa oluşturulan modeller bu dile çevrilir.

### Fiziksel tasarım

Veritabanının fiziksel olarak saklandığı dosyalar, dizinler hazır olarak kullanılan veritabanı yönetim sisteminin önerdiği şekilde ya da başka nedenlerle daha iyi başarımla elde etmek üzere tasarlanır. Burada hedeflenen en iyi erişim zamanı, en düşük saklama alanı ve en yüksek veri debisidir.

### *Gerçekleştirim*

Seçilen veritabanı yönetim sistemine özgü veri tanımlama dili ile yapılan tasarım gerçekleştirilir, derleme yapılır, veri dosyaları yaratılır, veri girişi hazır hale getirilir. Eğer önceki sistemde kullanılan veriler varsa bunların dönüşümü yapılır. Uygulama yazılımı ya da kullanıcı arayüzü ve ara-işlemler uygun bir yöntemle geliştirilir, veritabanı yönetim sistemi ile tümleştirilir. Gerekli sınamalar yapıldıktan sonra sistem kullanıma sunulur.

## **İNTERNET UYGULAMALARI TASARIMI**

İlk zamanlarda, yalnızca dosya paylaşmak şeklinde başlayan İnternet, gittikçe daha fazla metin bilgisi içeren siteler haline dönüşmüş, şimdi de grafik, resim, hareketli resim, ses ve video gibi çoklu ortamların hızlı bir şekilde aktarımları istenir hale gelmiştir. Elektronik posta artık yalnızca birkaç satırlık metin değil, ikili düzende dosyalar, resim ve hatta video da içermektedir. Birçok radyo ve televizyon kanalı İnternet üzerinden canlı yayın yapmaktadır.

Bilgi paylaşım ya da tanıtım amacıyla İnternet'te Web sitesi hazırlayan kişi, firma, kurum ya da kuruluşlar istekte bulunan bilgisayarlara kendi bilgilerini en iyi şekilde aktarabilmek için özel yazılımlar kullanırlar. İnternet'e erişen kullanıcı makinelerin standart bir tarayıcı (browser) çalıştırabilmek dışında, bu yazılımlara erişebilmek için özel bir yapıya sahip olmalarına gerek yoktur. İnternet uygulamaları için günümüzde en çok kullanılan araçlar JAVA programlama dili ve türevleri, Hyper-Text Mark-up Language (HTML), Extended Mark-up Language (XML), Hyper-Text Transfer Protocol (http) ve File Transfer Protocol (ftp) olmaktadır. Bu dil ve protokollerle geliştirilen yazılımların veya hazırlanan sayfaların gösterilebilmesi için kaynak kodun istekçi bilgisayara aktarılması gereklidir. Çok kullanılan küçük JAVA uygulamaları (applet) fazla yazılım mühendisliği gerektirmeyen, nispeten kolay sayılabilecek yazılımlardır. Fakat yine de belirli bir tasarımın önceden yapılması, kullanıcı arayüz standartlarına uyulması ve bakım kolaylığının gözetilmesi gereklidir. Son zamanlarda, İnternet üzerinde pek çok kullanıcı etkileşimli yazılım kullanılmaya başlanmıştır. Özellikle elektronik ortamda yapılan bankacılık, devlet

kurumları hizmetleri, veri erişimi ve dağıtımı gibi çeşitli sistemler gittikçe karmaşık hale gelmişlerdir. Bu sistemler büyüdükçe bir geliştirme sürecinin uygulanması gereksinimi de artmıştır.

Her türlü İnternet yazılımının tasarımı üç hedef gözetilerek yapılmalıdır: Amaç, arayüz ve güvenlik.

### 1-)Amaç

Bilgi paylaşım ya da tanıtım amaçlarıyla Web Sitesi hazırlayan kişi, firma, kurum ya da kuruluşlar istekçilere kendi bilgilerini en iyi şekilde aktarabilmeye çalışırlar. İnternet daha çok göze hitap ettiği için, ne yazık ki, altyapı sorunlarını dikkate almayan site sahipleri, sitelerini televizyon ya da gazete reklamlarındaki resimler ve kısa filmler topluluğu haline getirmektedirler. Oysa asıl gerekli olan en süratli şekilde bilgi paylaşımıdır. Bilgi eğer çoklu ortam (multi-media) şeklindeyse bunu belirterek seçimi siteye erişmek isteyen kişiye bırakmak en doğrusu olacaktır.

Bilgi toplamak üzere İnternet'te dolaşan bir bilgisayar kullanıcısı, mevcut altyapı nedeniyle, zamanının çoğunu sitelerdeki gereksiz resim ve reklamların tamamının görüntülenmesini beklemekle geçirmekte, yarar sağlayacağı bilgiye ulaşmak için gereksiz zaman harcamaktadır. Yapılan istatistiklere göre, İnternet'te rastgele dolaşan bir kişiye cazip bilgiyi verebilmek ve siteye girmesini sağlayabilmek için ortalama 30 saniye süre bulunmaktadır. Daha fazla bekleyen ve bulmak istediği bilgiye en fazla 3 sayfa değişiminde ulaşamayan kişi siteden ayrılmaktadır. O nedenle, özellikle ilk sayfanın amacı çok iyi belirlenmeli, bir tek sayfada çok miktarda özet bilgi vermek, alt sayfalara dallanmak üzere anlaşılır ve ayrıntılı menüler sunmak, ya da görsel reklam sunmak gibi hedefler ortaya konmalıdır.

### 2-)Arayüz

İnternet tabanlı yazılım geliştirirken düşünülmesi gereken en önemli unsur kullanım kolaylığı sağlayan grafiksel kullanıcı arayüzüdür. Web sayfası düzenlemek artık çocuk oyuncağı denecek kadar kolaydır. Ancak, kullanıcı arayüzü tasarlamak da bir bilim dalı haline geldiğinden bu işin belirli kurallara uygun olarak yapılmasında yarar vardır. Bu şekilde kullanıcıların dikkatinin istenen yerde toplanması, olabildiğince standart kullanım sağlayarak fazla bir öğrenme çabası gerektirmeden, açık ve anlaşılır arayüzlerle amaca en kısıda ulaşılması mümkün olur.

### 3-)Güvenlik

İnternet uygulamalarında bir diğer önemli konu da güvenlidir. İnternet'e erişim demek dünyadaki herkesin size erişebilmesi demektir. Bu kitle içinde kötü niyetli kişilerin de olduğu kabul edildiğinden, bankacılık işlemlerinde, mali işlerinde, sanal para kullanılan elektronik ticaret işlemlerinde, ticari ya da askeri gizlilik taşıyan bilgilerin aktarılması gerektiği durumlarda ya çok özel güvenlik yöntemleri (SSL, PGP, özel şifreleme gibi) kullanılmalı ya da bu tür işlemler hiç İnternet'e erişmeden

özel kriptolu hatlar kullanılarak yapılmalıdır. İnternet üzerinden yapılacak işlemler için özel kullanıcı adı ve şifresi kullanmak bazı durumlarda yeterli olabilir. Ancak, kritik işler için mutlaka ek güvenlik önlemleri alınmalıdır. Bunlar içinde özel bir yazılımı önceden kullanıcıya vermek, güvenilir iletişim ortamı oluşturmak, şifreleri İnternet üzerinden değil telefonla bildirmek, mektupla göndermek veya kişisel görüşme ile vermek gibi yöntemler sayılabilir. İnternet'e erişimde kendi bilgisayarımızın da güvenliğine dikkat etmek, istenmeyen programcıkların ve çerezlerin bilgisayarımıza girmesini engellemek gereklidir.

#### *4-)Tasarım Önerileri*

İnternet tabanlı yazılım geliştirirken dikkat edilmesi gerekli noktalar ve bazı öneriler aşağıda listelenmiştir:

- Tamamen pasif bir Web sitesi tasarlarken mutlaka basit ve amaca uygun olmasına dikkat edin. Sitenizi ziyaret eden kişi büyük bir olasılıkla belirli bir yarar bekleyen kişidir. Ona en çabuk şekilde istediği bilgiyi vermek sitenin amacı olmalıdır.
- Site girişlerine büyük resim veya uzun zaman alan animasyonlar koymayınız. Çok miktarda grafik ve hareketli resim bilgisinin istekçi bilgisayara aktarılması gerektiğini unutmayınız. Fazla beklenmesi kullanıcıda ters etki yaparak ayrılmasına da neden olabilir. Yükleme ve oynatması uzun zaman alacak tanıtıcı girişleri atlayabilmek için kullanıcıya olanak tanımak bir çözüm olabilir.
- Sayfalar arasında atlama yapmak için her bağ (link) ile açılan ve birbirinin tekrarı gibi olan ayrı sayfalar koymayınız; onun yerine paylaşılr sayfa düzenini seçip yalnızca yeni bilgilerin görüntülenmesini sağlayınız. Aynı yapıya sahip sayfaların yüklenmesi tampon kullanılmasına rağmen fazladan süre gerektirir.
- Kullanıcı arayüzü ve menü başlıkları içerik hakkında bilgi verici olmalıdır. Birbirine benzeyen başlıklar, kullanıcının kaybolmasına ve istemediği alt menüler- de bilgi aramasına neden olur.
- Sayfalara koyacağınız resimleri önce karşı tarafa süratle gidebilecek kadar düşük çözünürlük ve renk sayısı ile göndermek, daha sonra kullanıcı istediği takdirde büyük ve ayrıntılı resim aktarmak yöntemini kullanınız.
- Gereksiz yere artalan resimleri kullanmayınız. Herbir sayfa değişikliğinde bu resimlerin kullanıcı bilgisayara yüklenmesi gerektiğini unutmayınız.
- Sitenizi geliştirirken sürekli olarak sayfalardaki veri miktarını kontrol edip aktarma süresini ölçünüz. Siteyi tamamladıktan sonra da güncel olarak geçerli olan süratlerde İnternet bağlantısı sağlayarak süre testleri yapınız. Bugünkü teknoloji için, genel olarak 5-10 saniyelik sayfa yükleme süreleri normal olarak kabul edilebilir. Daha fazla olan durumlarda bu kadar verinin veya resimin gerçekten gerekli olup olmadığını tekrar düşününüz.

- İeriinde az bilgi bulunan bir tek sayfanın yklenmesi dahi nemli bir zaman alacaėı iin, kullanıcının yapmak istediėine ulařabilmesi iin en az sayıda iřlem gerektirecek bir dzen kurulmalıdır. rneėin, bir dosyayı indirmek isteyen kullanıcı bir baėı tıkladıėında indirme iřlemleri, dosya seme, onaylama gibi bir dizi sayfayı peř peře izlemek zorunda kalmamalıdır. Bir siteye bilgi almak iin giren kiři genellikle en fazla 3 baėı inceler, umduėunu bulamazsa siteden ayrılır.
- JAVA tabanlı "applet" tipi yazılım geliřtirirken de aynı řekilde, karřı tarafa gitmesi gerekli program kodunun en az tutulması iin gerekli nlemleri alıp eniyileme alıřmaları yapınız.
- Gvenlik gerektiren noktalarda gvenilirliėi mutlaka bilimsel olarak kanıtlanmış yntemler ya da zel geliřtirilmiş gvenlik paketleri kullanınız.
- Sayfalarda grsel kirliliėe izin vermeyiniz. Yalnızca amalanan bilgilerin bulunmasına dikkate ediniz. Siteyi ziyaret kiřilerin bir bakıřta fakat kaybolmadan İstedeiėi bilgiye ulařmalarını saėlayınız.
- Her site iin bir arama dzeneėi ve o sitede kullanılan tm baėların yer aldıėı bir dizin listesi veya geniř aplı bir men kullanınız.
- Sitenizde yer alan bilgileri gereken sıklıkta gncelleyiniz. Yeterince gncel olmayan siteler kullanıcıların ilgisini kaybederler.
- Etkileřimli yazılımlar geliřtirilirken dikkat edilmesi gereken en nemli noktalardan biri de yrtlmekte olan bir iřlemin İnternet baėlantısının kesilmesi durumunda oturumla beraber gvenli bir řekilde sonlandırılmasıdır.

## TASARIMDA GENEL KURALLAR

Yazılım tasarımı, her ne kadar bir sanat niteliėini de korumasına raėmen yine de belirli bir takım bilimsel kurallar ortaya konmuřtur. Modler tasarım, daha nceki tasarımın tekrar kullanımı, denetim ve iřlevin ayrılması, daha nce bahsettiėimiz eřitli tasarım yntemleriyle beraber kullanılabilir genel kurallardan bazılarıdır.

- 1-)Modler Tasarım
- 2-)Tekrar Kullanım
- 3-)Denetim ve İřlevin Ayrılması
- 4-) Genel Kurallar

Yazılım tasarımı sırasında genel olarak dikkat edilmesi gerekli bazı nemli noktaları řu řekilde zetleyebiliriz:

Tasarımın ncesinde belirli bir yntem seilmeli, tm tasarım ařaması boyunca bu yntemin uygulanmasına sadık kalınmalıdır.

İyi bir tasarım, yazılım bileřenleri arasındaki denetim ve etkileřimi sıradzensel ve anlaşılır bir yapıda sergilemeli, tasarımı belgeleri bu bilgileri anlaşılır bir řekilde iermelidir.

Tasarımın modüler olması için yazılım mantıksal olarak bileşenlere ayrılmalı ve her bileşenin yerine getirmesi gereken işlevler o kısımda ele alınmalıdır.

Tasarımda veri ve yordam gösterimleri birbirlerinden ayrılmalıdır.

Tasarımda paket, sınıf veya modüllerin arayüzleri, yordam ve fonksiyonların bedenleri belirlenmeli, bunlar belgelerde yer almalıdır.

Tasarım asla kaynak kod görüntüsü vermemelidir. Ancak, kodlayıcıya yeterli bilgiler ayrıntılı tasarımda yer almalıdır.

Tasarım, kodlayıcılar tarafından genel olarak bilinen yapılar hakkında gereksiz ayrıntılar içermemelidir. Örneğin, bir bağlı listeye eleman ekleme veya çıkarma için gerekli deyimlerin tasarımda yer almasına gerek yoktur.

## BELGELENDİRME

Yazılım tasarımı ister elle, isterse yardımcı araçlarla yapılsın, mutlaka belgelendirilmelidir. Özel bir aracın çıktısı da belge olarak kullanılabilir. Ancak, geliştirmede seçilmiş olan yöntemin ve belgelendirme standardının uygulanması gereklidir.

Yazılım tasarım aşamasında, her türlü tasarım bilgisinin yer aldığı bir Yazılım Tasarım Tanımlaması (Software Design Description) belgesi hazırlanır. Bazı standartlarda bu belge Yazılım Tasarım Belirtimi (Software Design Specification) adını da alabilir.

## KAYNAKLAR

1. Yazılım Mühendisliği; M. Erhan Sarıdoğan.