FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği

YMT114-Yazılım Mühendisliğinin Temelleri Dersi

Proje Uygulamaları ve Dökümantasyonu

Tez Analizi

Proje Çalışma Grubu

CAN BALAMAN

Proje Yürütücüleri

CAN BALAMAN

ŞUBAT -2021

İçindekiler

İçindekiler Tablosu

1. Giriş

1.1 Projenin Amacı

Yazılan tezlerin belli kurallar çerçevesinde uygun formatta olup olmadığını öğrenmeyi amaçlar

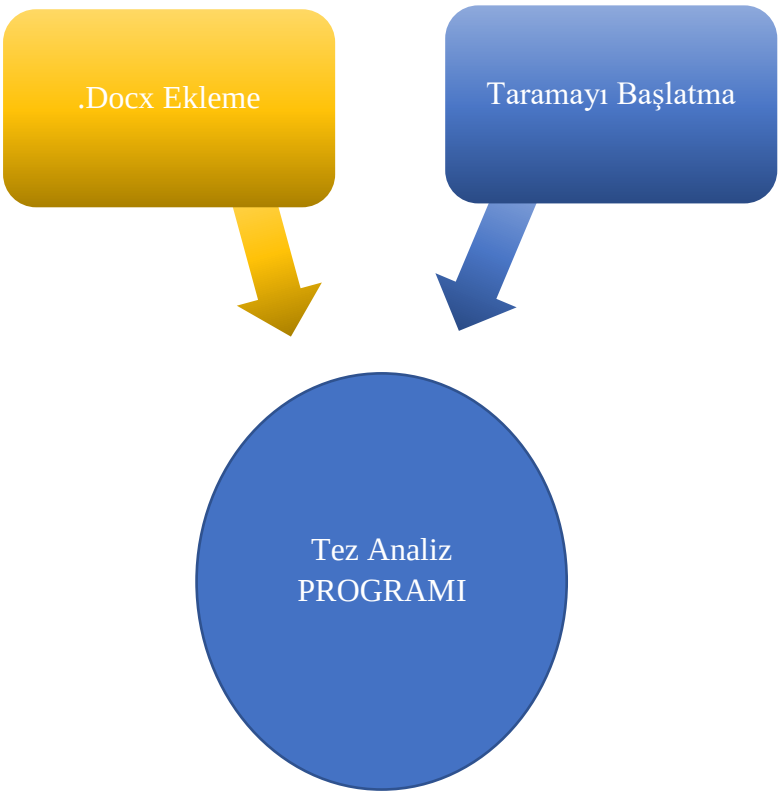
1.2 Projenin Kapsamı

.docx formatının açılabildiği tüm sistemlerde çalıştırılabilmektedir

2. Proje Planı

2.1 Giriş

Bu yazılım, belli kurallara uygun olarak dokümantasyonu inceler ve. Hataları belirtir.



2.1 Projenin Genel Yapısı

2.1 Projenin Plan Kapsamı

Projenin plan kapsamında genel olarak bu programın düzenli bir şekilde veriyi doğru bir şekilde kontrol etmesi esas alınmıştır.

--Tez Analiz NEDEN GEREKLİ ?

Tez Analiz sistemi tez yazarken veya kontrol ederken kaynaklanacak hataları önceden görüp çözümlemek için çok gerekli bir sistemdir

Projenin maliyet kestirim dökümanı

Proje Adı Tez Analiz Programı

Ölçüm Parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	5	5	25
Kullanıcı Çıktı Sayısı	5	6	30
Kullanıcı Sorgu Sayısı	3	6	18
Kütük Sayısı	15	7	105
Kütük Sayısı	14	5	70
Ana İşlev Nokta Sayısı			248

Teknik Karmaşıklık Sorusu	Puan
1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	**5**
2. Veri iletişimi gerekiyor mu?	**5**
3. Dağıtık işlem işlevleri var mı?	**3**
4. Performans kritik mi?	**5**
5. Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak?	**1**

7. Çevrim içi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu?	**0**
8. Ana kütükler çevrim-içi olarak mı günleniyor?	**1**
9. Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı?	**3**
10. İçsel işlemler karmaşık mı?	**2**
11. Tasarlanacak kod, yeniden kullanılabilir mi olacak?	**5**
12. Dönüştürme ve kurulum, tasarımda dikkate alınacak mı?	**1**
13. Sistem birden çok yerde yerleşik farklı kurumlar için mi geliştiriliyor?	**5**
14. Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak?	**1**
TOPLAM	**37**

0: Hiçbir Etkisi Yok
1: Çok Az etkisi var
2: Etkisi Var
3: Ortalama Etkisi Var
4: Önemli Etkisi Var
5: Mutlaka Olmalı, Kaçınılamaz
İN = AİN x (0.65 x 0.01 x TKF)
İN=248*(0,65*0,01*37)
İN=59.644
Satır Sayısı = İN *30

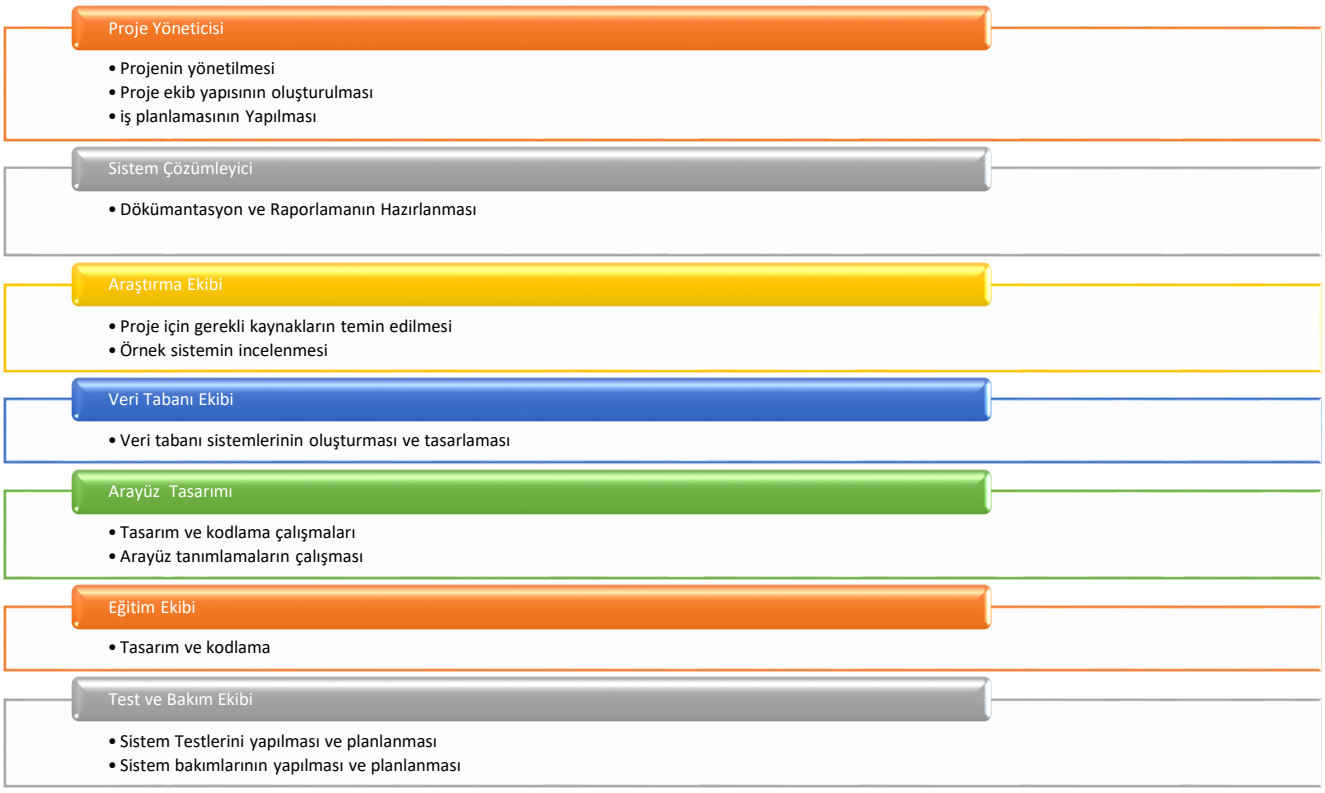
SATIR SAYISI=59.644*30=1789,320SATIR YAKLAŞIK 1800 SATIR
Etkin Maliyet Modeli – COCOMO
Organik proje: a=2,4 , b=1,05 , c=2,5 , d= 0,38
Yarı – Gömülü Projeler İçin: a=3,0 , b=1,12 , c=2,5 , d= 0,35
Gömülü Projeler İçin: a=3,6 , b=1,20 , c=2,5 , d= 0,32
Öncelikle projemizin türünü belirlememiz gerekiyor. Küçük ekip tarafından geliştirildiği için organik projeler arasına giriyor.
Aylık Kişi Başı İş Gücü = E = a x (KSS)b
Geliştirme Süresi (Ay) = D = c x (E)d
Eleman Sayısı = E / D
Formülde verilen değişkenler şöyle:
KSS = Kod Satır Sayısı manasına gelmektedir ve birimi bin satırdır. Projenin tahmini kaç bin satırdan oluşacağını belirtmemizi sağlar.
Aylık Kişi Başı İş Gücü = E= $2,4*1,5^{1,05}$=3.672
Geliştirme Süresi =D= 2,5 x = 10 ay
Tahmini Gerekli Personel Sayısı = 3,18 / 3.67= 1 (yaklaşık)

2.2 Proje Zaman-İş Planı

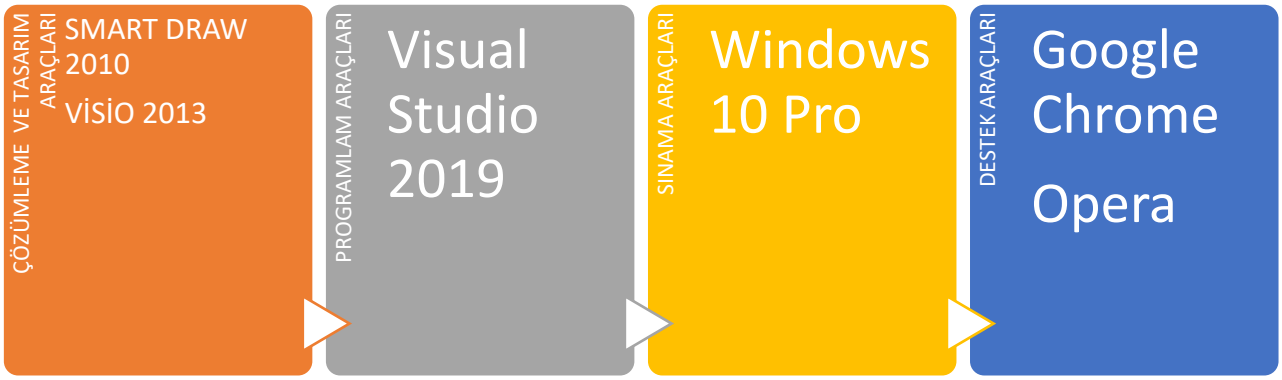
İş-Zaman Çizelgesi			
Zaman **İŞ**	**1.Hafta**	**2.Hafta**	**3.Hafta**
Proje Teklifi	**X**		
Proje Planı	**X**		
Analiz	**X**	**X**	**X**
Sistem Çözümleme	**X**	**X**	**X**
Kullanıcı Ara yüz Tasarımı		**X**	**X**
Gerçekleştirim		**X**	**X**
Test			**X**
Sunum			**X**

2.4Proje Ekip Yapısı


Proje Yönetici
Can Balaman
Sistem Çözümleyici
Can Balaman
Sistem Yöneticisi
Can Balaman
Veri Tabanı
Yöneticisi
Can Balaman
Yazılım
Ekip Lideri
Can Balaman
Sistem Tasarımcısı
Can Balaman
Kalite Tasarım
Can Balaman
Arayüz Tasarımcısı
Can Balaman


Proje Yöneticisi
• Projenin yönetilmesi
• Proje ekib yapısının oluşturulması
• iş planlamasının Yapılması
Sistem Çözümleyici
• Dökümantasyon ve Raporlamanın Hazırlanması
Araştırma Ekibi
• Proje için gerekli kaynakların temin edilmesi
• Örnek sistemin incelenmesi
Veri Tabanı Ekibi
• Veri tabanı sistemlerinin oluşturması ve tasarlaması
Arayüz Tasarımı
• Tasarım ve kodlama çalışmaları
• Arayüz tanımlamaların çalışması
Eğitim Ekibi
• Tasarım ve kodlama
Test ve Bakım Ekibi
• Sistem Testlerini yapılması ve planlanması
• Sistem bakımlarının yapılması ve planlanması

2.5Kullanılan Özel Geliştirme Araçları ve Ortamları



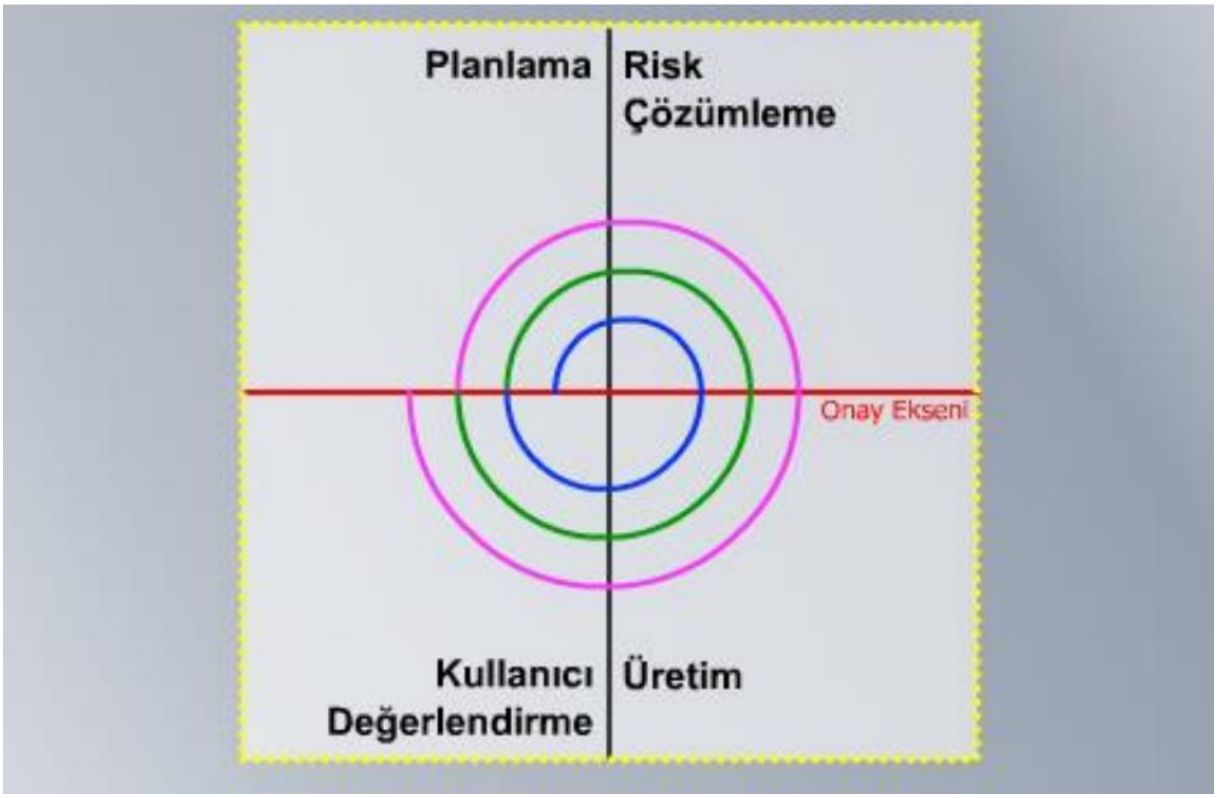
2.6 Proje Standartları, Yöntem ve Metodolojiler

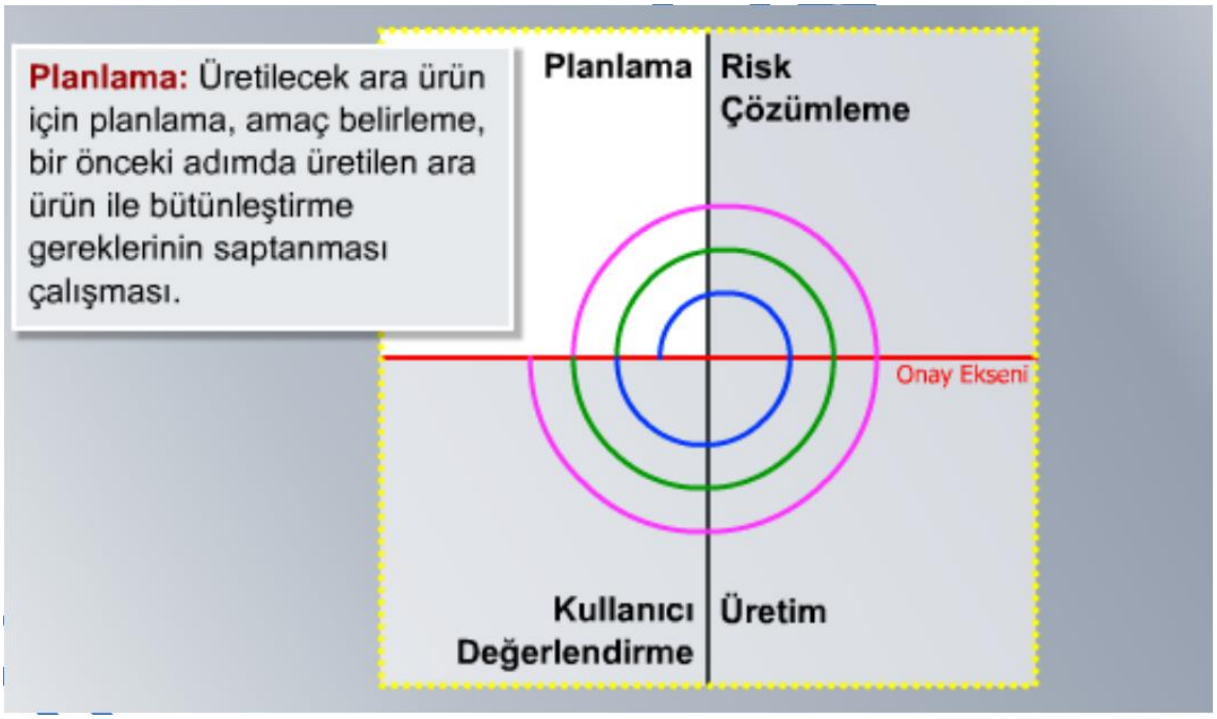
Spiralin başladığı ilk çeyrek içinde ilk isterler toplanır ve buna göre proje planlaması yapılır. İkinci çeyrekte, ilk tanımlanan isterlerse göre risk çözümlemesi yapılır. Üçüncü çeyrekte, risk çözümlemesi sonunda ortaya çıkan isterilerin tanımlanmasındaki belirsizlikleri ortadan kaldırmak için prototipleme yöntemi kullanılır. Gerekirse benzetim(simülasyon) veya diğer modelleme kullanılarak isterilerin daha sağlıklı tanımlanması sağlanır. Dördüncü çeyrekte, müşteri, ortaya çıkan ilk ürünü inceleyerek değerlendirme yapar, önerilerde bulunur. Bu şekilde tanımlanan ilk döngü bir sonraki döngü için bir girdi oluşturur.

Aşama	Kullanılan Yöntem /Araçlar	Ne İçin Kullanıldığı	Çıktı
Planlama	○ Veri Akış Şemaları, ○ Süreç Belirtmeleri ○ Maliyet Kestirim Yöntemleri ○ Proje Yönetim Araçları	○ Süreç İnceleme ○ Kaynak Kestirimi ○ Proje Yönetimi	Proje Planı
Çözümleme	○ Süreç Belirtimleri ○ Veri Akış Şemaları, ○ Görüşme ○ Nesne ilişki Şemaları ○ Veri Sözlüğü	○ Süreç Çözümleme ○ Veri Çözümleme	Sistem Çözümleme Raporu
Çözümlemeden Tasarımı Geçiş	○ Akışa Dayalı Çözümleme, ○ Süreç Belirtimlerinin Program Tasarım Diline Dönüştürülmesi ○ Nesne İlişki Şemalarının Veri Tablolarına Dönüştürülmesi	○ Başlangıç Tasarım ○ Ayrıntılı Tasarım ○ Başlangıç Veri Tasarımı	Başlangıç Tasarım Raporu
Tasarım	○ Yapısal Şermalar ○ Program Tasarım Dili ○ Veritabanı Tabloları	○ Genel Tasarım ○ Ayrıntılı Tasarım ○ Veri Tasarımı	Sistem Tasarım Raporu

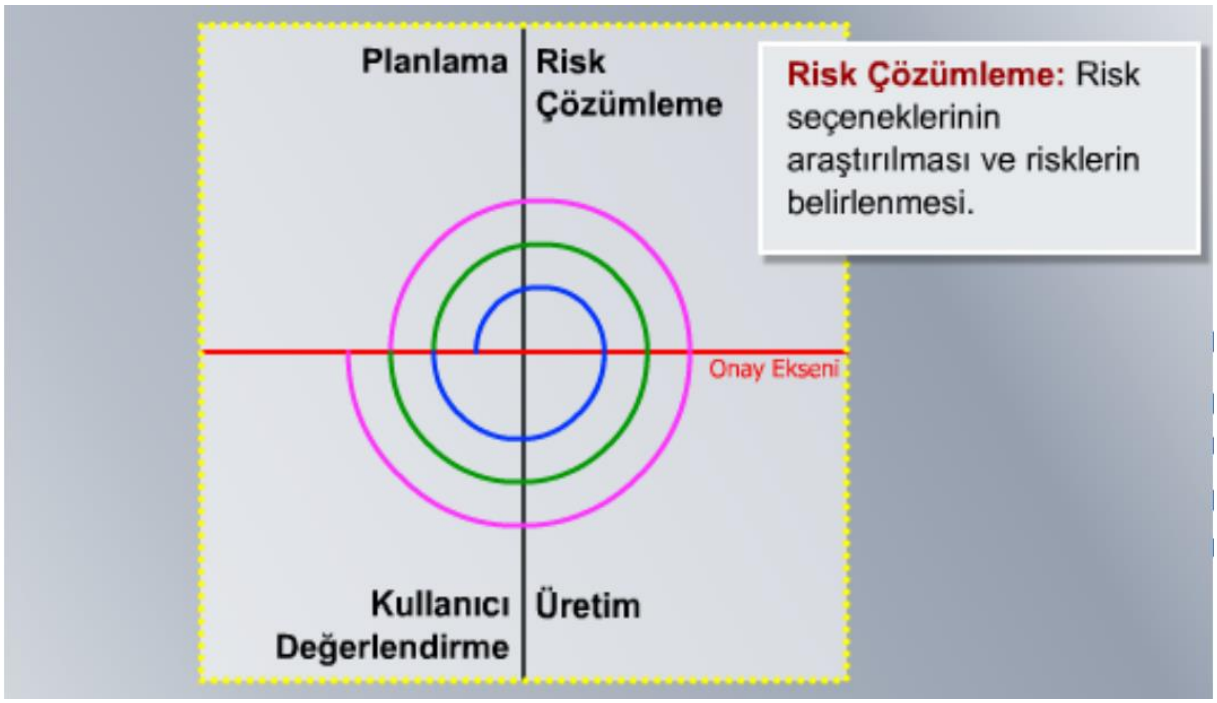
- Veri Sözlüğü

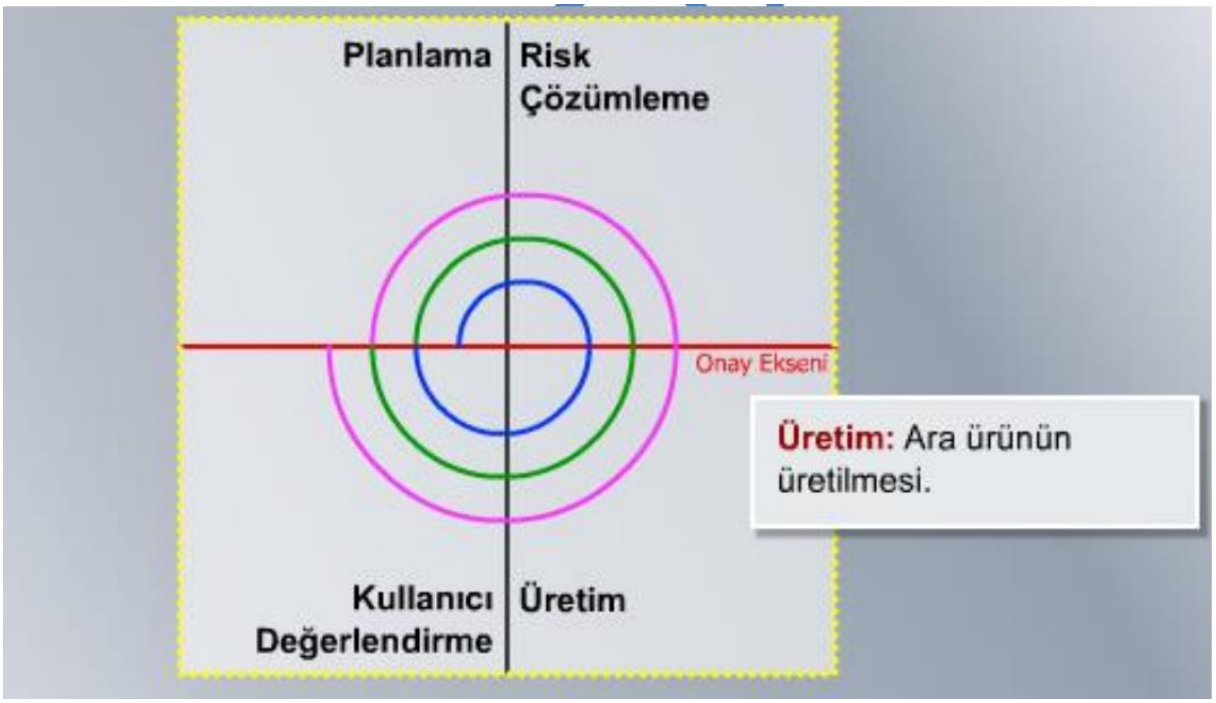
Proje standartları yukarda belirtildiği gibidir. Bunu yanı sıra kullanılan sistem modelinde ise helezonik model kullanılmıştır



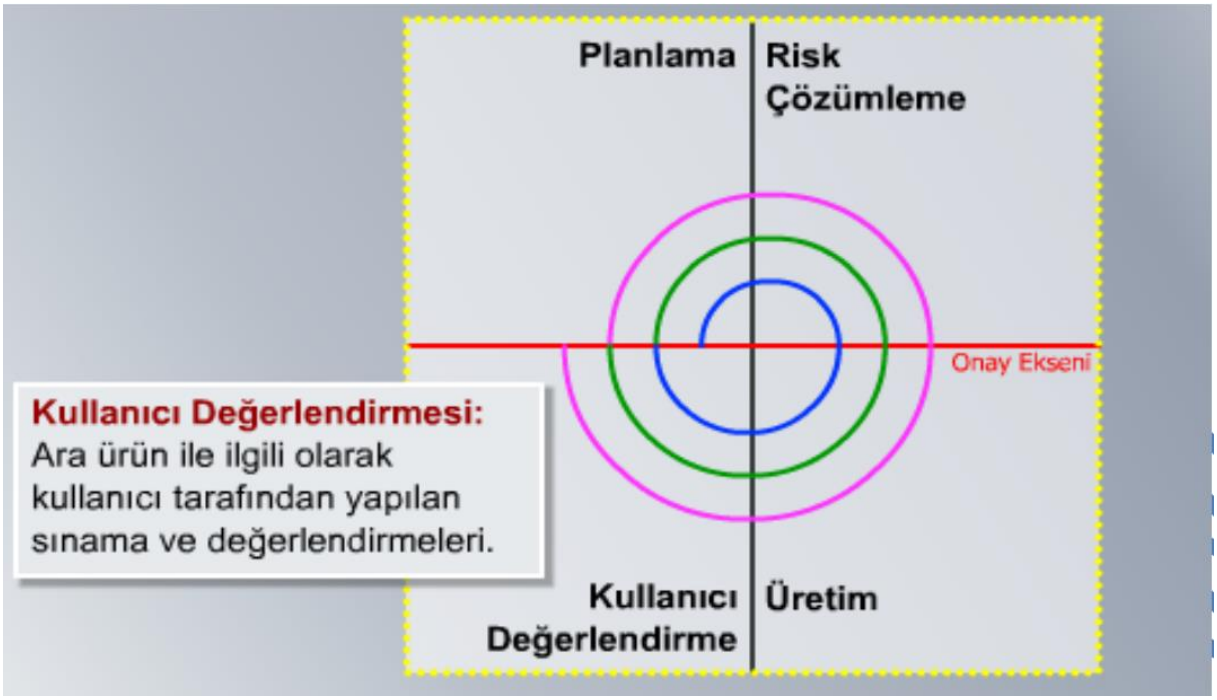


Bu aşamalar yazılım geliştirmede önemli unsurlardır.Bu kullandığımız yazılım geliştirme metodolojilerinden Spiral Model dir.Bunu kullanmamızın amacı döngü şeklinde kontrol edip hataları en aza indirebilmektir.





Üretim aşaması uzun zaman alan bir aşamadır.Tasarıma yer verilir ve projenin büyük çoğunluğu bitmiş olur.Bu aşamada ara ürün üretilir.Proje bu aşamayla epeyce bir yol almış olacaktır.



EKİP YAPISI ZAMAN İŞ PLANI				
	OCAK		**ŞUBAT**	
SİSTEM ÇÖZÜMLEYİCİ	✓	✓		
SİSTEM YÖNETİCİSİ	✓	✓	✓	✓
SİSTEM TASARIMCISI		✓	✓	✓

2.7 Kalite Sağlama Planı

Projedeki kalite sağlama planımız yukardaki tabloda da belli olduğu üzere;

1.Ekonomi: Ekonomik açıdan yazılımın maliyeti her ne kadar ilk seferde pahalı olsa da ileriye dönük düşünüldüğünde ve zaman tasarrufundan ötürü gayet uygundur.

2.Tamlık: Projede herhangi bir açık olmamalı ve programda bulunan tüm butonlar textler vs. çalışır ve tamdır.

3.Yeniden Kullanılabilirlik: Otomasyon her koşulda tekrardan düzenlenip kullanılabilecek.

4.Etkinlik: Kullanıcı sistemin her alanına hakim olduğu için sistemi etkin bir biçimde kullanacak.

5.Bütünlük: Admin1 sistemin tüm kısımlarına hakim olacak ve program bir bütün halinde çalışacaktır.

6.Güvenilirlik: Otomasyon gerekli güvenlik önlemlerinin alınması.

7.Modülerlik: Modülerlik otomasyonun her seviyesindeki kişinin ayrı ayrı sayfalardan söz sahibi olmasını sağlar. Örneğin: Yönetim modülü, Giriş Modülü…

8.Belgeleme: Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olacak bu 23 doküman oluşturulmuştur.

9.Kullanılabilirlik: Kullanılabilirlik olarak her seviyedeki insana hitap edeceğinden zor renkler karmaşık sistemlerden kaçınılmıştır.

10.Temizlik:

11.Değiştirilebilirlik: Otomasyonun veri tabanını erişme yetkisi olan ve sistem hakkında bilgisi olan herkes sistemde değişiklik yapabilecek.

12.Esneklik: Proje farklı platformlarda ve küçük bir kod ekliyeyerek internet üzerinden çalışacağından gayet esnektir.

13.Genellik: Proje her işletmenin kullanılabileceğinden geneldir. Ve Türkiye genelinde kullanılacaktır.

14.Sınanabilirlik: Projedeki pilot bölge uygulaması sınana bilirliğinin göstergesidir.

15.Taşınabilirlik: Sistem işletim sisteminin üzerinde çalıştığından herhangi bir özel cihaz gerektirmez ve istenilen cihazlarda taşınabilir ve kullanılabilir.

16.Birlikte Çalışılabilirlik: Bu projedeki en büyük sıkıntı olacak veri girişi şuanda var olan ve her bireyin bilgilerinin saklayan sistemle birleşik ve eş zamanlı çalışmakta

2.8 Konfigürasyon Yönetim Planı

Sistemin ilerde kullanıcının yeni istemlerini karşılayamaması veya sistemin yapısındaki bazı bileşenlerin değişmesi sonucu güncelliğini kaybettiğinde olası konfigürasyon planı hazırlandı.

- İşletmeye giren yeni personel için sistem personel girişi açması,
- Herhangi bir sebepten dolayı ayrılan personel olması,
- Sistemde herhangi bir istenmeyen durum halinde,

Durumları için konfigürasyon yönetim planı oluşturuldu.

2.9 Kaynak Yönetim Planı

Mevcut bir kaynağımız olmadığından kaynak olarak sadece bu proje dokümantasyonu var.

Kaynak yönetiminde Şu hususlar dikkate alınacaktır;

- Yeterli hata bulunuyor mu?
- Kalite beklendiği gibi mi ?
- Eğer yeterli sayıda hata bulunmuyorsa veya kalite beklenenden daha iyi görünüyorsa, Kalite gerçekten daha iyi olabilir mi ?
- Bu projede yeni bir süreç iyileştirmesi kullanılıyor mu ?
- Proje elamanları yeni bir eğitim aldı mı?
- Yeni bir araç kullanılıyor mu?
- Yeterli vakit harcanıyor mu?
- İnceleyiciler yeterli hazırlık yapıyor mu?
- İnceleme toplantısında çözüm bulmak için zaman kaybediliyor mu ?

2.10 Eğitim Planı

Projeden kazanılacak en önemli olaylardan biride eğitimdir. Kullanılacak dillerin arayüz editör ve programların kullanımında hakim olunamaması halinde bu program başarıyla neticelendirilemez. Bu yüzden projede bazı eğitimler alınması gereklidir. Proje kapsamında alınacak olan eğitimler;

- C# Dil Eğitimi

Gereken Eğitimlerdir

Konu	**Süreç**
Giriş, Açıklamalar ve Ön Bilgi	1.Gün
Temel Kavramlar ve Sistem için gerekli bileşenlerin tanıtılması	2.Gün
Gerekli yöntem ve metodolojilerin incelenmesi	3.Gün
Gerekli Hata çözme yöntemlerinin incelenmesi	4.Gün
Değerlendirme ve Sonu.	5.Gün

Proje teslimi sonrası sistemi kullanacak olan kişilere proje kullanımı olarak 15 dakikalık bir tanıtım yapılacaktır. Bu tanıtımda içeriği şu şekildedir;

- Sisteme genel bakış

2.11 Test Planı

Proje test ekipleri ve görevleri şu şekildedir;

- Admin tüm sayfaların ve tüm yapıları kendi kontrol edecek

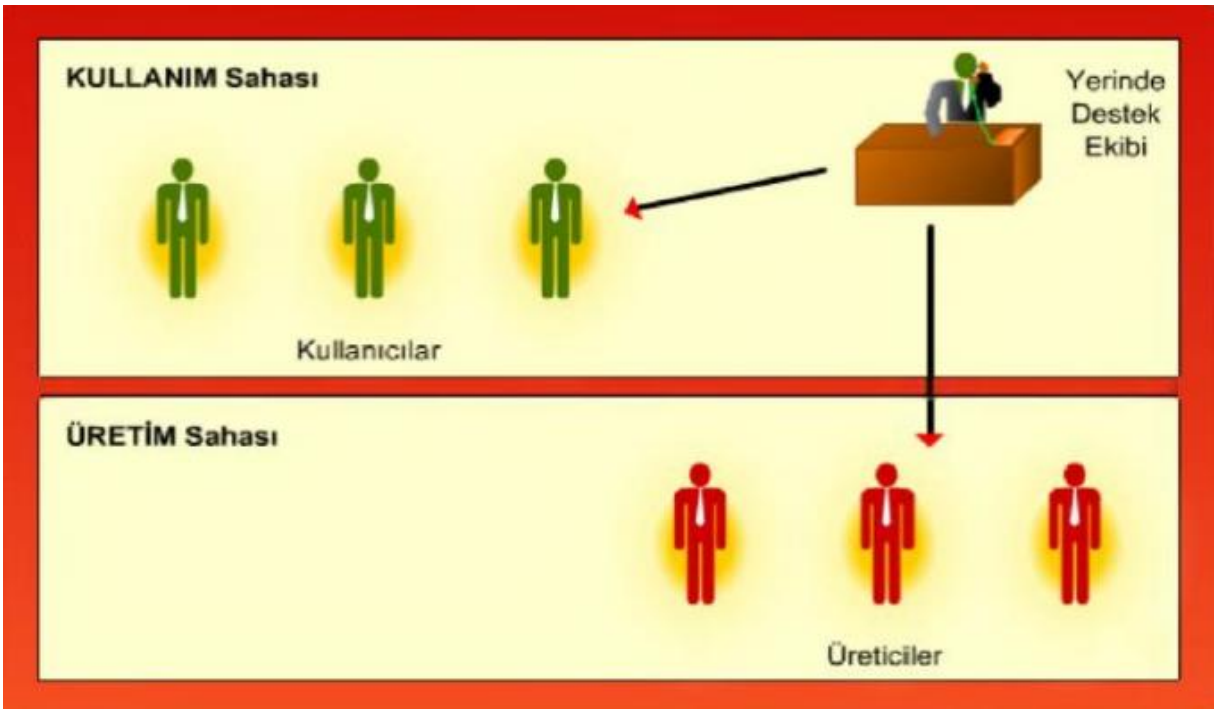
- Bu ekiplerin test aşamasında yapacağı işler aşağıdaki gibidir;

Ön hazırlık olarak küçük bir işletmeden başlamak koşuluyla gerekli hiyerarşi takip edilecek ve gereli testler yapılacaktır bir uygulama gerçekleştirilecek.

2.12 Bakım Planı

Projenin bakım planına gelecek hergün kullanılacak bu sistem tüm değişim ve bazı durumlarda kullanıcı eklenip çıkarılacak tüm bu sistemsel değişiklikler bakım planında

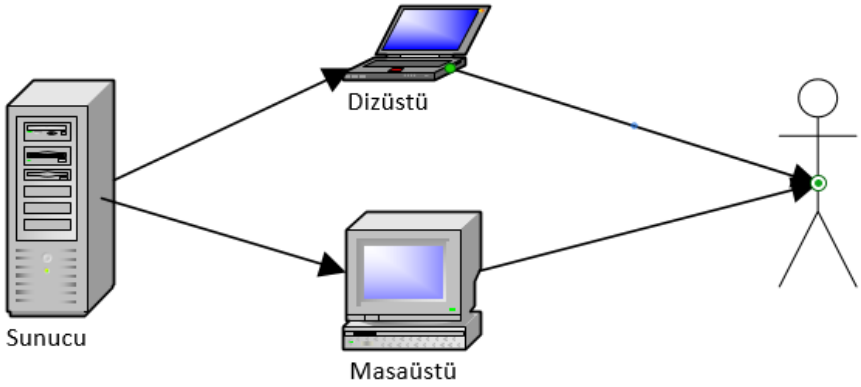
yapılacaktır.



3. SİSTEM ÇÖZÜMLEME

3.1 Mevcut Sistem İncelemesi

Mevcut sistem sunucu mantığı ile çalıştığı için veri tabanını internete bağlı bir sunucuya bağladığımızda ve local olarak kullanabiliriz.

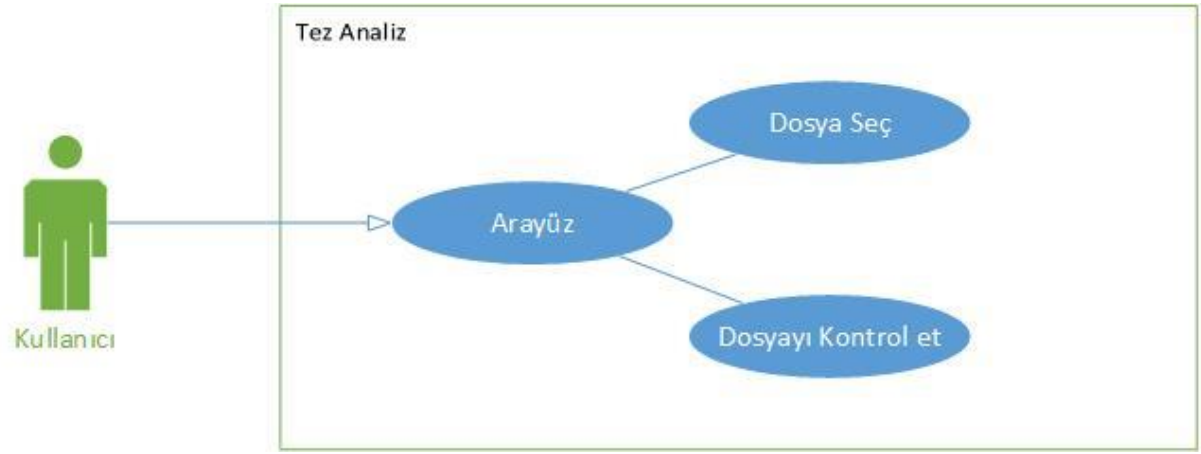


3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

Mevcut sistemler incelendiğinde sonuca giden yolda epeyce bir eksikler ve resmi olmayan durumlar söz konusu

3.2.2 İşlevsel Model



3.2.3 Veri Modeli

Veri tabanı ilişkisel veri modelinde veriler tablolar üzerinden kurulan ilişkiye dayanmaktadır

3.2.4 Başarım Gerekleri

Mevcut sistemler incelendi ve mevcut sistemin eksiklerinden yola çıkılarak, sistemin başarımı için

- Sistemin sonuç üretim doğrulukları
- Tepki sürelerinin en aza indirilmesi
- Mali külfetin azaltılması
- Hile hata ve yanlışlıkların en aza indirilmesi
- Kullanım kolaylılığı
- Anlaşılabilirlik

3.3 Arayüz (Modül) Gerekleri

3.3.1 Yazılım Arayüzü

Projenin çalışması esnasında böyle bir açık verilmemesine özen gösterildi. Gerekli olan her türlü değişiklik seurce kodları üzerinden yapılıp tekrar derlenecek.

3.3.2 Kullanıcı Arayüzü

Projede kullanıcının arayüzü tasarlanırken herhangi bir şekilde renkler seçilerek tarafsız rahat büyük puntolu yazılı bir arayüz tasarlanacaktır

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

Geliştirme sürecinde genel olarak belgelendirilmesi hem ileriye dönük hem de şimdiki geliştirme sürecinde projenin tamamlanma yüzdesini nerede kalınıp nerelerde eksikler olduğunu genel hatlarıyla göstermesi amacıyla yapıldı. Bunun yanı sıra projeye yeni dahil olan personellerin olaya hakimiyeti açısından bu yönteme başvuruldu.

3.4.2 Eğitim Belgeleri

Mevcut bir belgemiz bulunmamaktadır.

3.4.3 Kullanıcı El Kitapları

Bu kısma projenin en son safhasında kullanıcılara verilecek eğitimlerden pilot uygulamalardan yola çıkılarak hazırlanacak.Yani proje sonunda rahat ve kolay kullanımdan dolayı bir eğitim semineri ve bir kullanım kitapcığı hazırlanacaktır.

4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri

4.1.1 Genel Sistem Tanımı

- **Gereksinimler**

Gereksinimler kısmında çok kolay bir tasarım hazırlanacak .Çünkü her ne kadar da Teknoloji Çağı'nda yaşasak da bilgisayarı açıp kapamayı zor yapan insanlar var.

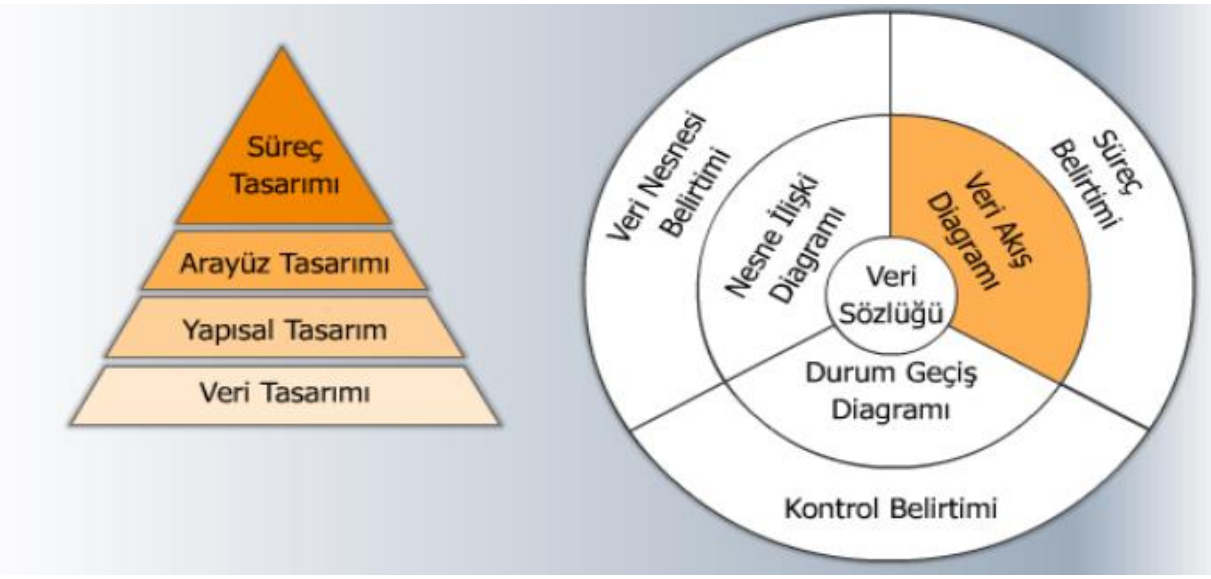
- **İşlevsel Belirtimler**

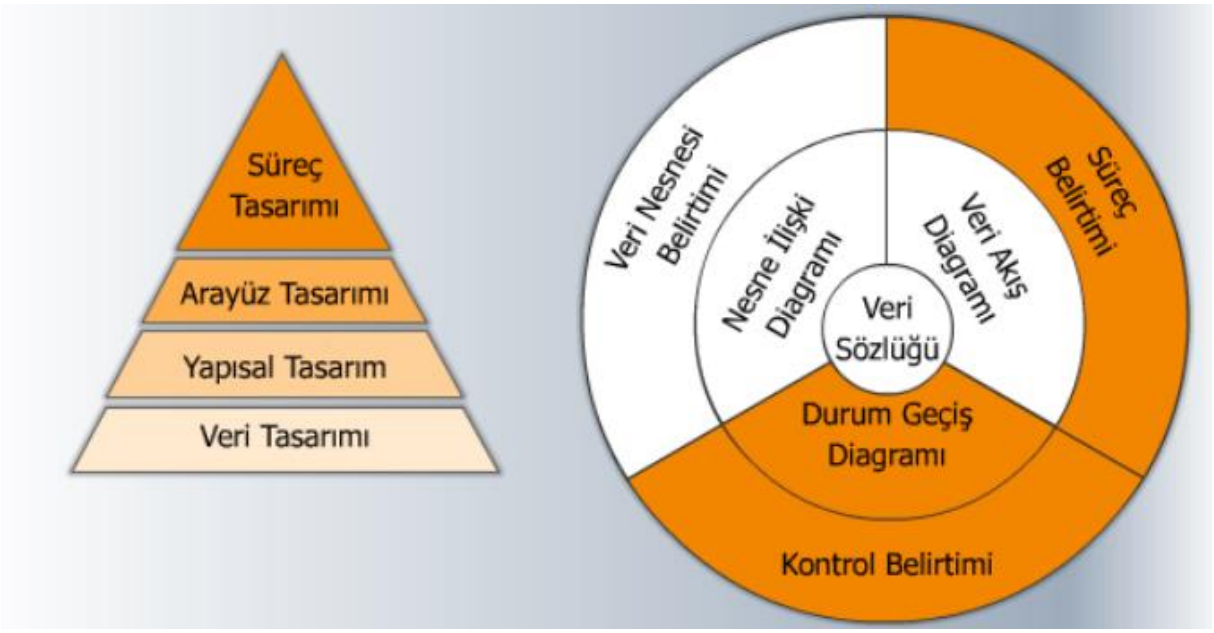
Bu kısımda ilk önce sistemin ne yapacağı sorusuna cevap verelim. Sistem Türkiye de kâğıt üstünde yapılan ve birçok külfeti olan sistemin dijital ortama aktarılması diye özetlenebilir.

- **Tasarım**

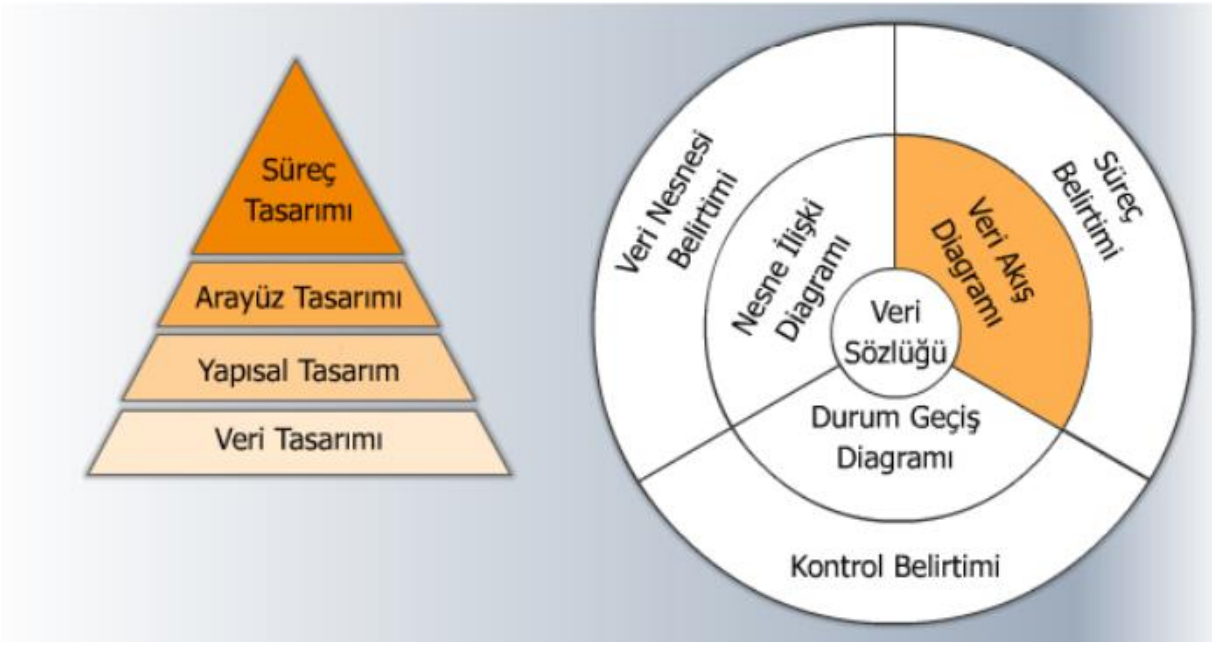
Tasarım aşamasında neler olacağını grafiksel olarak aktarmak isterim.

1. İlk önce bir Süreç tasarımı olacak ve adımlar aşağıdaki gibi olacak.

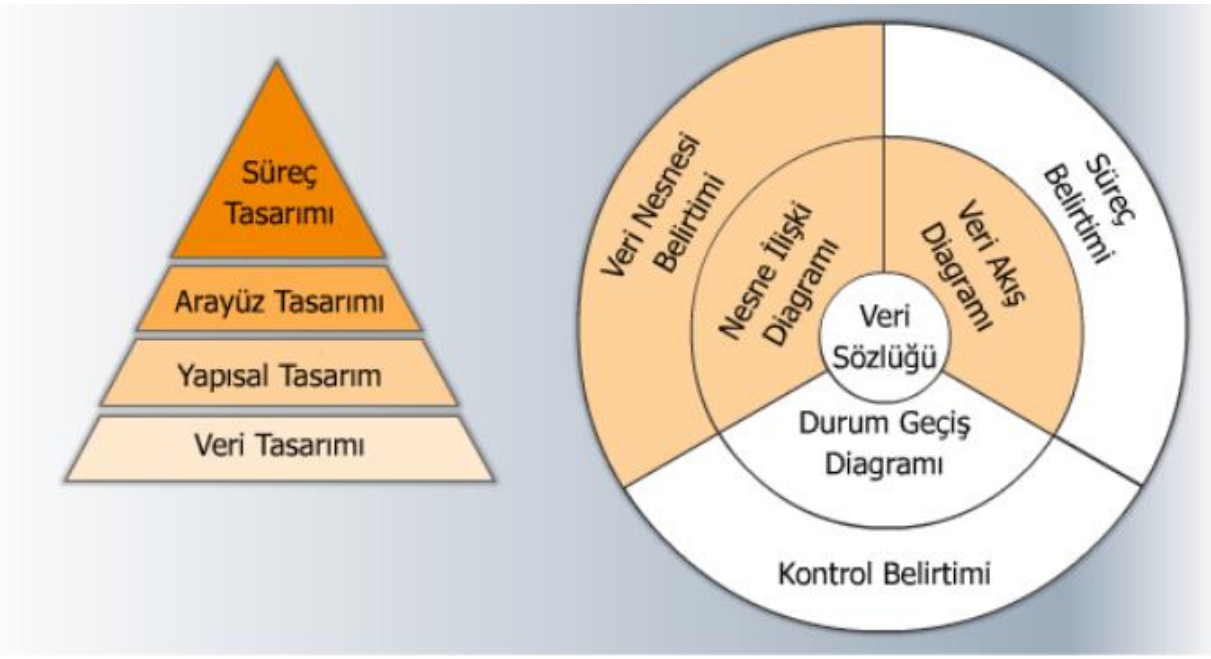




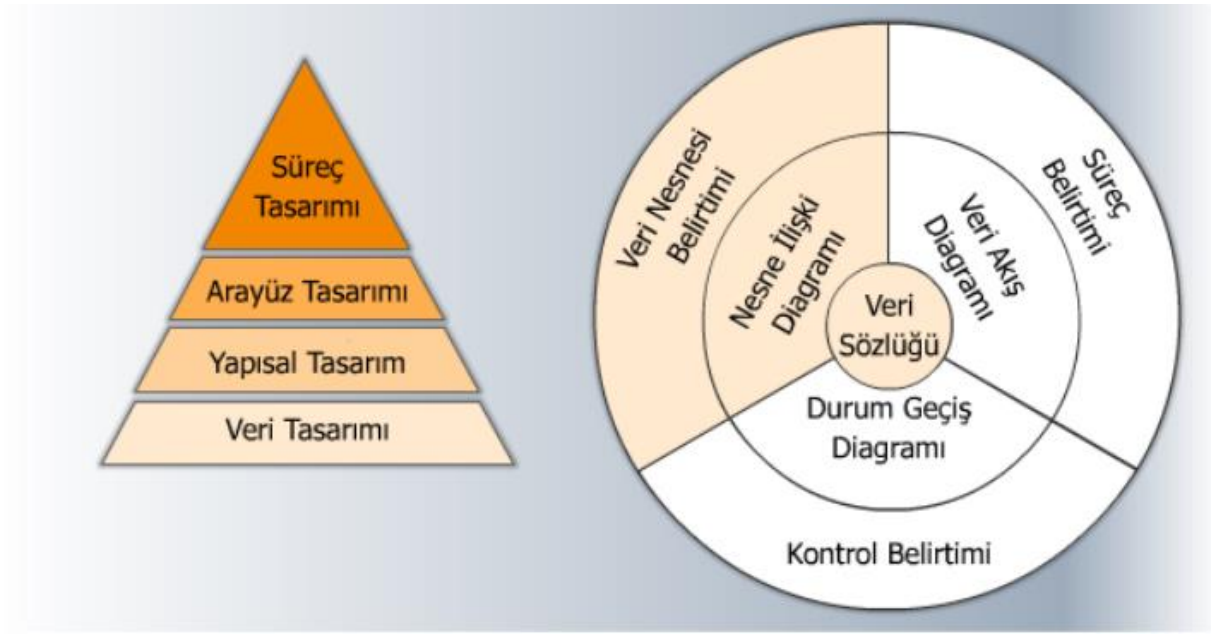
2.Süreç tasarımı bittikten sonra sıra Arayüz tasarımına geldi ve arayüz tasarımı aşağıda göründüğü adımlarla gerçekleştirildi.



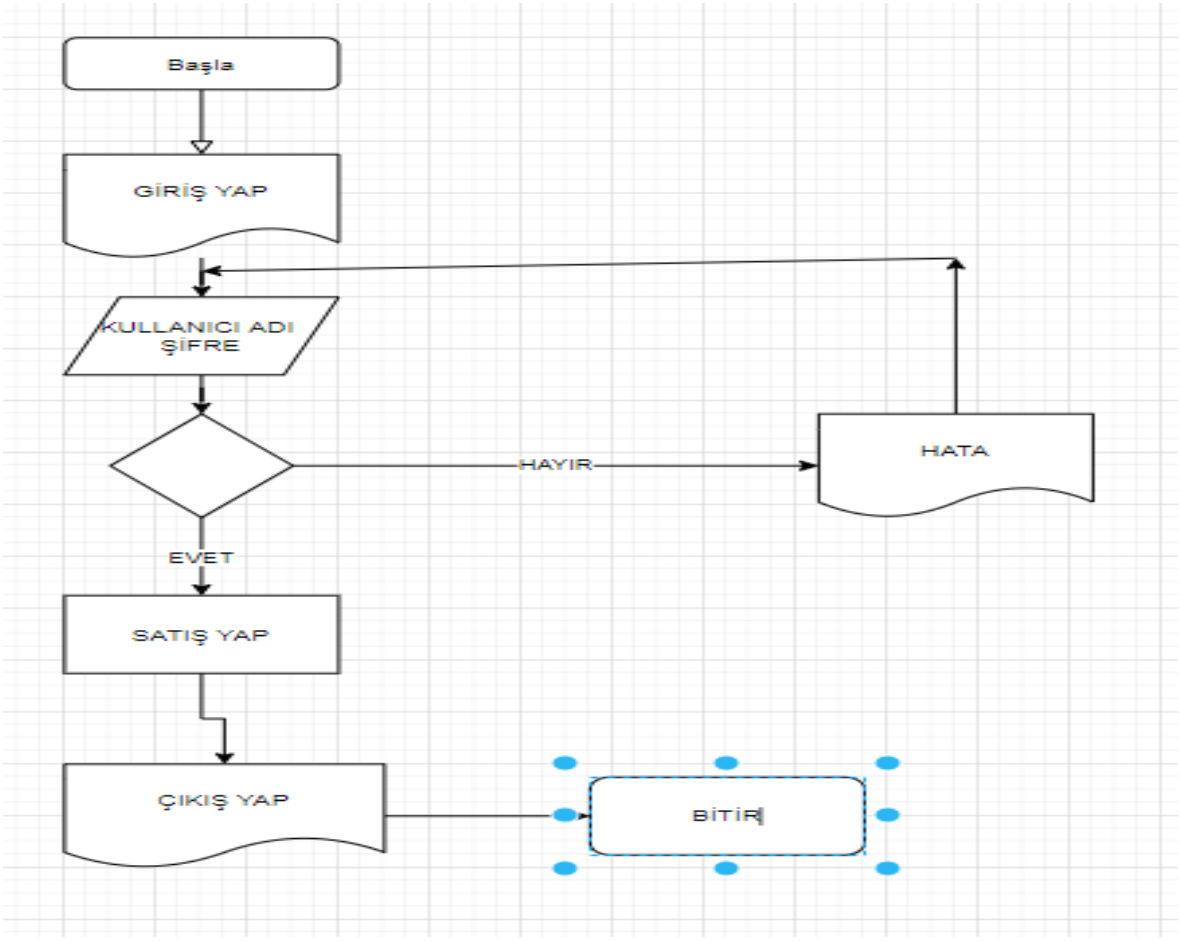
3.Arayüz tasarımını aştıktan sonra sora yapısal tasarıma geldi. Yapısal tasarımda izlenen yollar aşağıdaki gibi oldu.

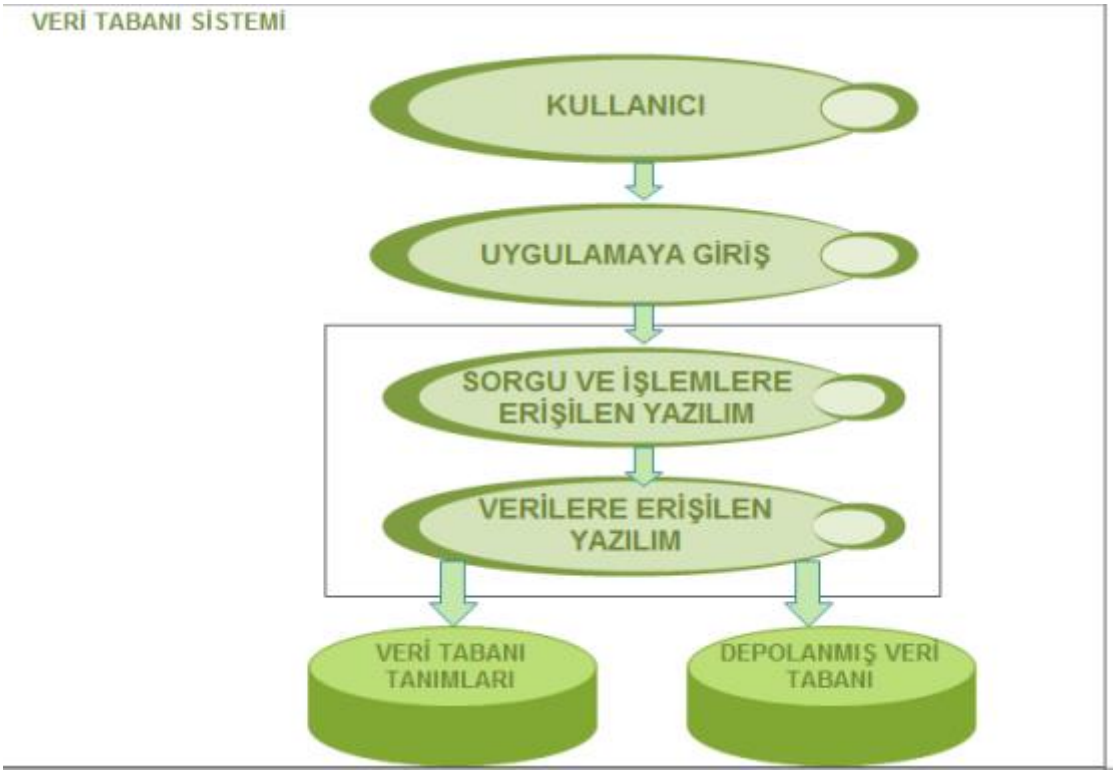


4.Yapısal tasarımı da yapıp Veri tasarımına geçildi ve şu adımlar izlendi.



4.1.2 Sistem Mimarisi





Sistemin mimarisinin akış diyagramı şeklinde verilmesinin temel nedeni sistemin işleyiş mantığının nasıl olduğu ve nasıl bir yol çizileceğinin bilinmesidir .Akış diyagramı sistemin temel mantığı hakkında bize fikir verecektir

4.1.3 Testler

Genel hatlarıyla testlerimiz iki aşamada gerçekleştirilecek. Bilinen adıyla pilot bölge uygulaması yapılacak.

Alfa Aşaması: Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi ile yapılacak.

Beta Aşaması: Kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapılacak.

4.1.4 Performans

Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek Sistemin Tasarıma Uygunluk Performansı; Tasarımı yapılan sistemin stabilitesi ve işleyiş performansı değerlendirilecek. Veri Yapısının Sistemle Performansı; Veri yapısının sistemle stabilitesi ve çalışma zamanındaki uyumluluk düzeyindeki performansı değerlendirilecek.

4.2 Veri Tasarımı

4.2.1 Tablo tanımları

Sistem 9 tablodan oluşmaktadır. Bunun yanı sıra kullanıcının ekstra istekleri doğrultusunda ek tablolara açık bir sistem tasarlanmaktadır.

4.2.2 Veri Tanımlar

Veri tipi olarak integer(int) kullanılmasının amacı sayısal değerleri almaktan ötürüdür. String olarak kullanılan değerler kelime içeren değerleri tutacağından ötürü kullanıldı. Boolean veri tipi işte işin can alıcı noktası burası boolean veri tipi iki faklı değer alır ya truedir ya false dır

4.3 Süreç Tasarımı

4.3.1 Genel Tasarım

Genel olarak tasarımda ilk önce veri tabanı modeli oluşturuldu. Ardından giriş modülü onun ardından yönetici Arayüzü ve en sonda kullanıcı arayüzü oluşturduk.

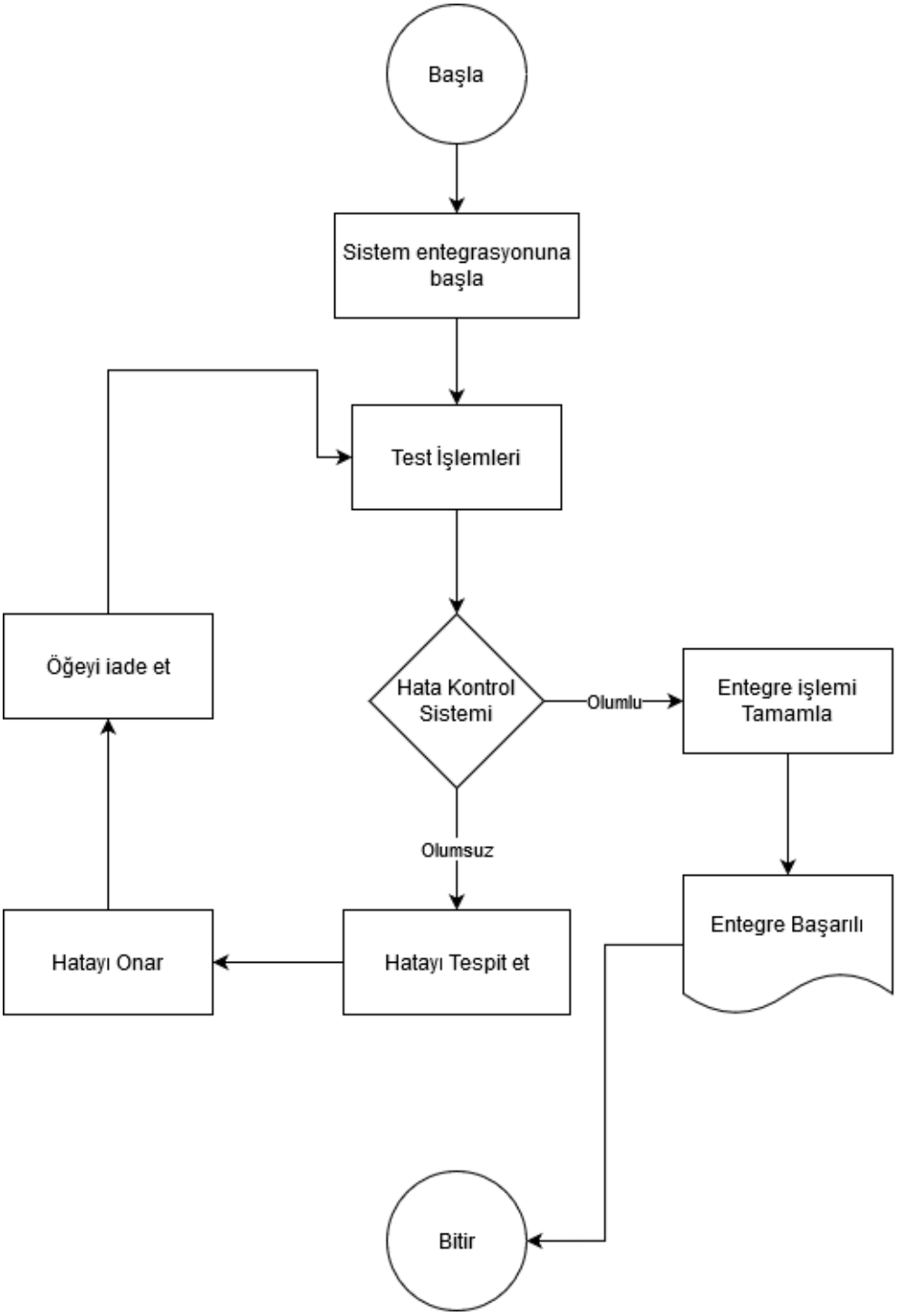
4.3.2 Modüller

4.3.2.1 Giriş Modülü

4.3.2.1.1 İşlev

Kullanıcının sisteme müdahale edecebileceği ekrana erişmesi için aşması gereken bir modüldür.

4.3.2.1.2 Modül iç Tasarımı



Kullanılacak değişkenler:

private Connection con: Veri tabanı için bağlantı nesnesi.

private String vtURL: Veritabanının bulunduğu yol.

private String kullaniciAdi: veritabanına bağlantı için kullanıcı adı.

4.4 Ortak Alt Sistemlerin Tasarımı

4.4.1 Veri Dağıtım Altsistemi

Veri dağıtımından ziyade veri alma üzerine bir sistem kuruldu. Dağıtılan sistemlerde zararlı kişilerin müdahalesi kolay olmaktadır bunun yanı sıra bizim sistemimizde terminaller serverlardan veri çekecek serverlar terminallere veri göndermeyecek.

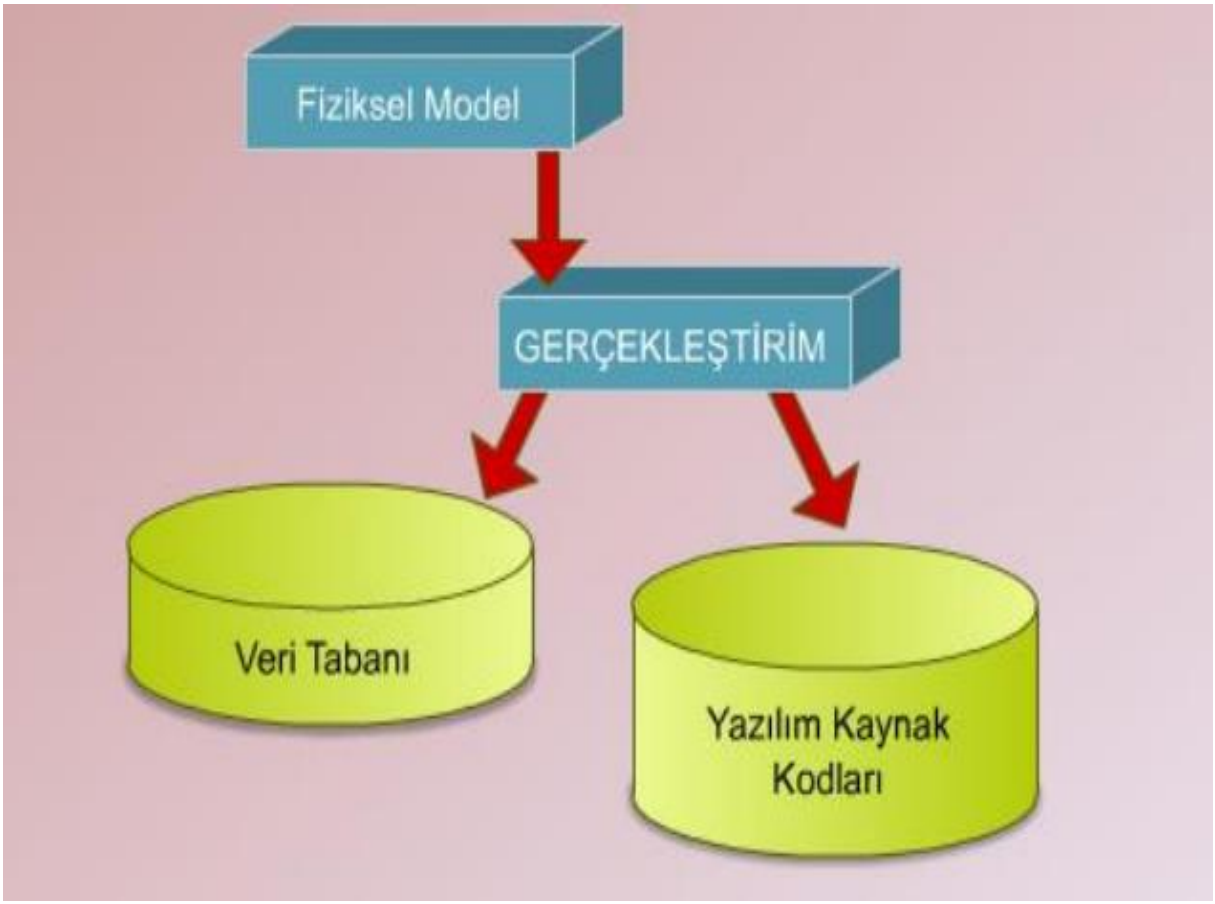
4.4.2 Yedekleme ve Arşivleme İşlemleri

Depolanan verilerin, herhangi bir nedenle zarar görmesi, sistemin çalışma süreçlerinde ciddi zararlar oluşturabilir. Yaşanabilecek bir felaket durumu sonrasında, depolanan verilerin geri yüklenememesi, sistemin sağlandığı kullanıcılara veya kurumlara çok ciddi zararlar verebilir. Bu nedenle sistemin çalışma süreçlerine bağlı olarak, yedekleme sistemleri kurulmalı ve yedekleme işlemleri günlük olarak takip edilmelidir. Yedekleme sistemlerinin kurulumu; yedeklenecek veri miktarı, yedekleme sıklığı, yedeklenen verinin zaman içerisinde değişme oranı ve maksimum veri kaybı gibi parametrelere bağlıdır. Sistemin birden fazla sunucusunun eş zamanlı yedekleme işlemini yapabilmesi, işletim sistemlerinin kayıt dosyalarını tam ve eş zamanlı olarak yedekleyebilmesi ve işletim sistemleri üzerinde çalışan veri tabanı uygulamasının yedeklerini sistem kapatılmadan alabilmesi gerekmektedir.

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1Giriş

Gerçekleştirim çalışması, tasarım sonucu üretilen süreç ve veri tabanının fiziksel yapısını içeren fiziksel modelin bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi çalışmalarını içerir. Yazılımın geliştirilmesi için her şeyden önce belirli bir yazılım geliştirme ortamının seçilmesi gerekmektedir.



5.2 Yazılım Geliştirme Ortamları

geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

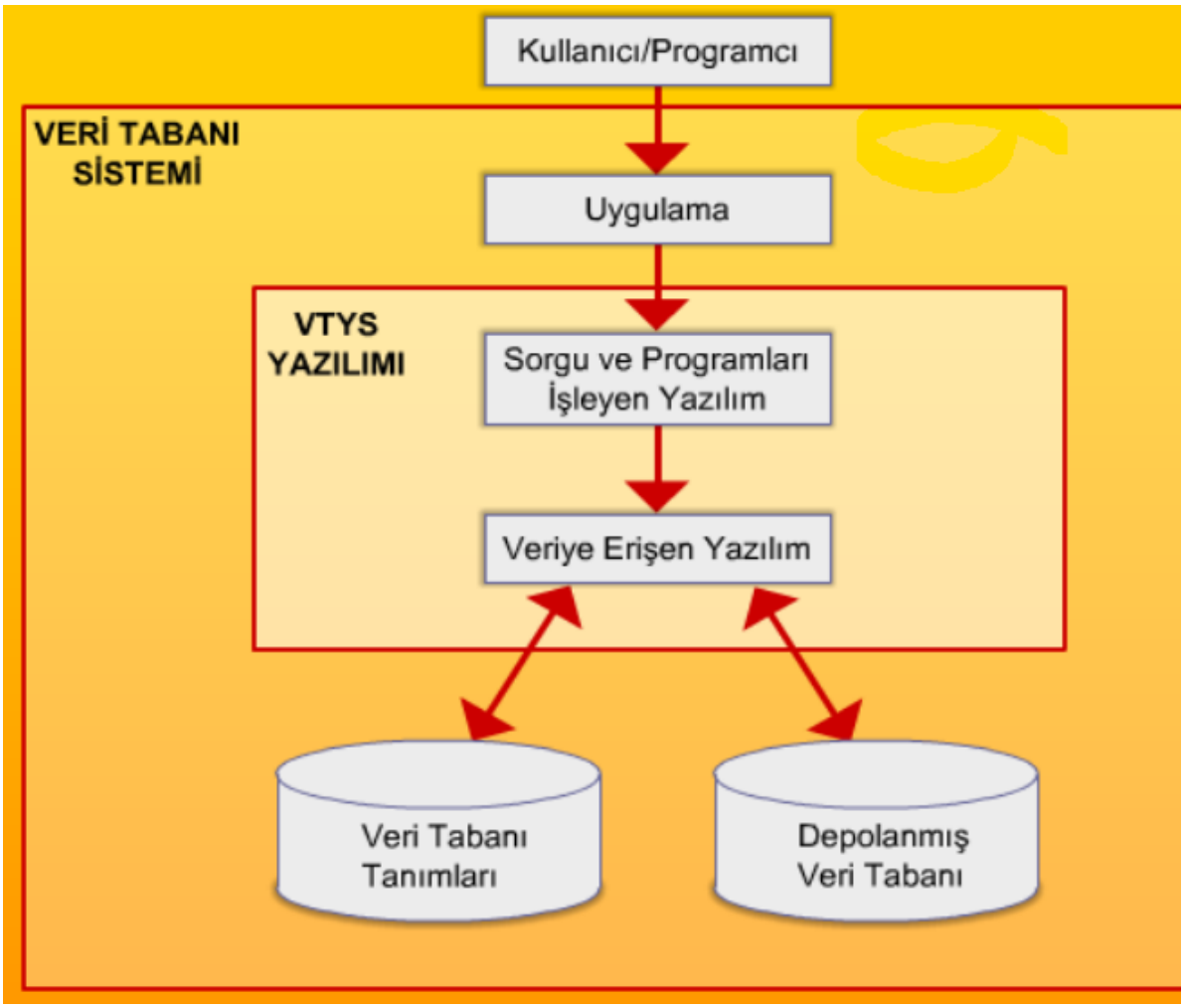
- Programlama Dili
- Veri Tabanı Yönetim Sistemi
- Hazır Program Kitapçıkları

CASE Araçları belirlendi ve yazılım geliştirme ortamı hazırlandı.

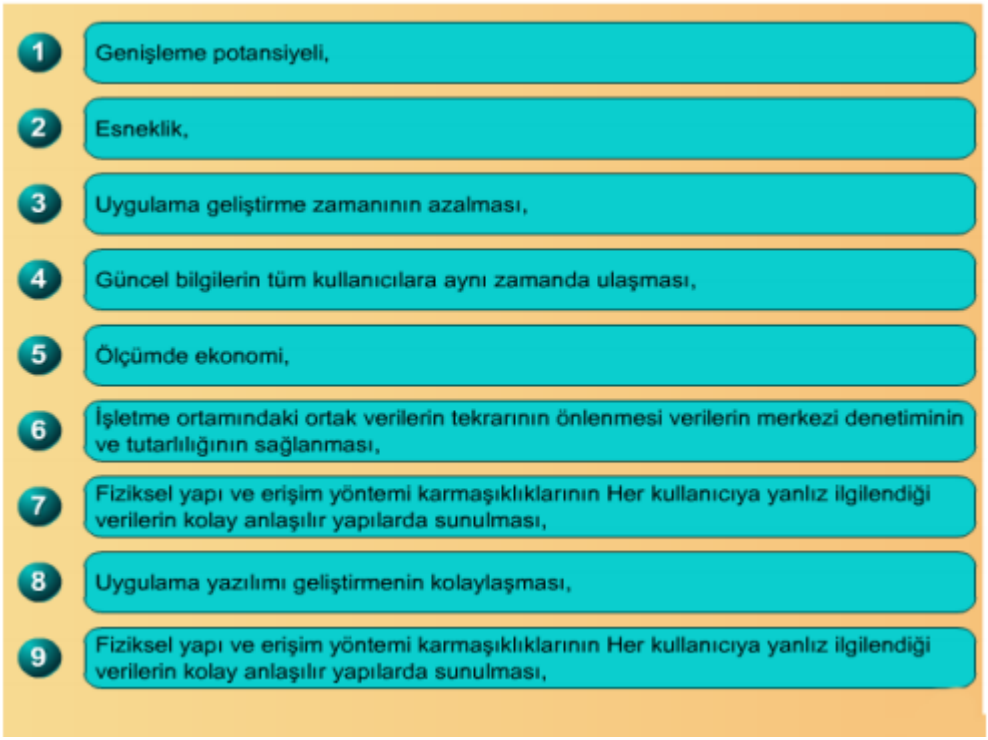
5.2.1 Programlama Dilleri

Kendi içlerinde sınıflandırılan bu diller arasından bizim seçtiğimiz sınıf şüphesiz ki veri işleme yoğunluklu uygulamalarda kullanılacak dillerden olacaktır. Bu yüzden daha çok nesne tabanlı programlama önem verilecek kullandığımız diller arasında JAVA başlıcalarıdır.

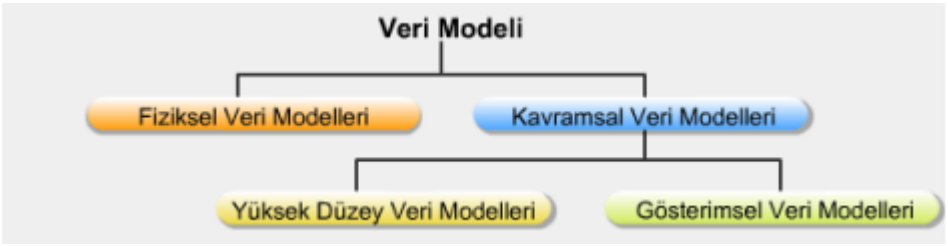
5.2.2 Veri Tabanı Yönetim Sistemleri



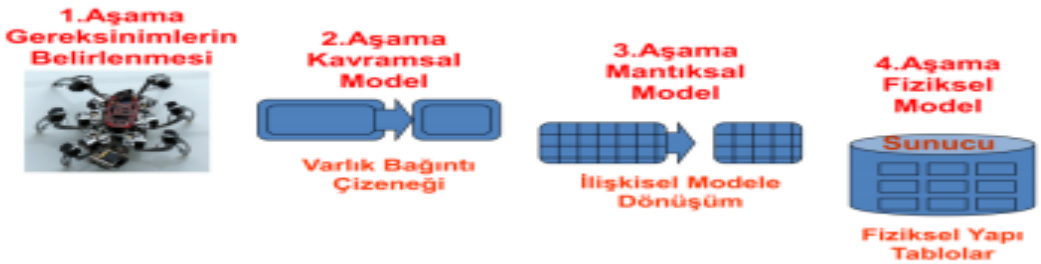
5.2.2.1 VTYS Kullanımının Ek Yararları



5.2.2.2 Veri Modelleri



Fiziksel Veri Modelinde verilerin Mysql de tablolar içindeki alanlarda saklanacağı ve birbirleriyle ilişki içinde olduğunu söyleyebiliriz. Kavramsal Veri Modelini iki ana başlık 64 altında inceleyebiliriz.



Üçlü şema mimarisinde görülen yapıların, kullanıcı gereksinimlerinden yola çıkılarak aşamalı bir şekilde fiziksel olarak gerçekleştirilmesidir.

1. Gereksinimlerin belirlenmesi

- Veri tipleri

- Veri grupları
- Veriler ile ilgili kurallar
- Veriler üzerinde yapılması gereken işlemler

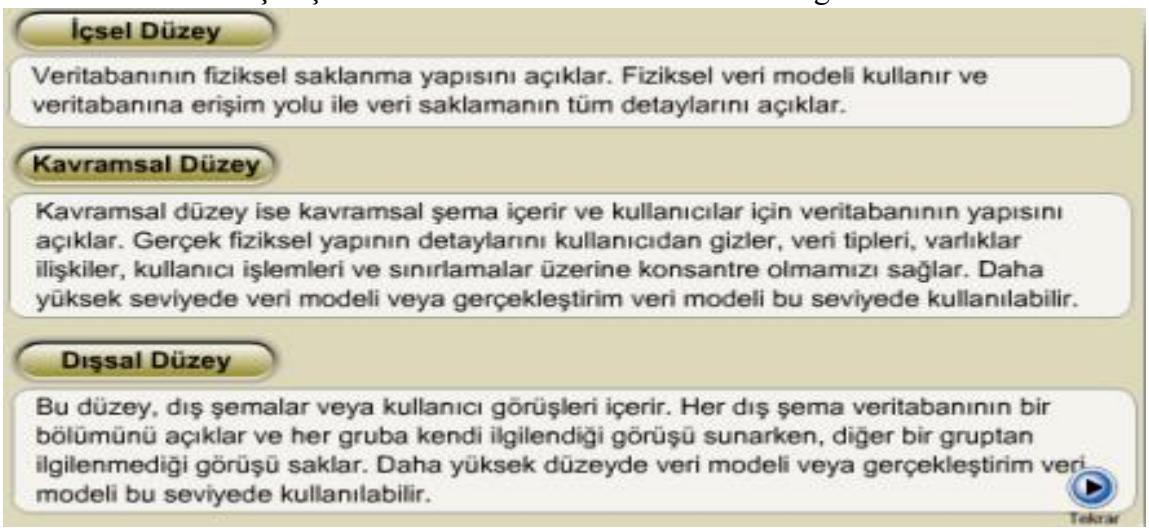
2. Kavramsal Model Kullanıcıdan elde edilecek gereksinimler ile ilgili bir analiz çalışmasının yapılması ve birbiriyle bağlantılı verilerin gruplanarak bir düzenleme içinde modellenmesi gerekmektedir. Bu modeli grafiksel olarak varlık bağıntı seçenekleri ile gösteririz.

3. Mantıksal Model Veri tabanı tasarımlarımızın ilişkisel veritabanı modelinde tablolar ile ifade edilebilmesi için yapılması gereken dönüşümü içerir.

4. Fiziksel Model Fiziksel olarak sistemin kurulması sağlanır. Kullanılacak VTYS ile ilgili ilk temas burada kurulur.

5.2.2.3 VTYS Mimarisi

VTYS mimarisini üç başlık altında ele aldık bunları özetlemek gerekir ise:



5.2.2.5 Veritabanı Dilleri ve Arabirimleri

Sistemimizde veritabanı dili olarak SQL kullanıldı. Henüz prototip aşamasında olan sistemimiz için MYSQL arabirimini yeterli gördük.

Veri Tanımlama Dili (VTD)	Kavramsal şemaları tanımlamak üzere veritabanı yönetici ve tasarımcısı tarafından kullanılır
Saklama Tanımlama Dili (STD)	İçsel şemayı tanımlamak için kullanılır.
Görüş Tanımlama Dili (GTD)	Görüş tanımlama dili kullanıcı görüşlerini tanımlamak ve kavramsal şemaya dönüştürmek amacıyla kullanılır.
Veri İşleme Dili (VİD)	Veri işleme dilidir. Veri tabanı oluşturulduktan sonra Veri tabanına veri eklemek, değiştirmek, silmek veya eklenmiş veriyi getirmek amacıyla kullanılır.

5.2.2.5.1 Veri Tabanı Sistem Ortam Veri tabanı sistem ortamında phpMyAdmin bizim kahrımızı çekti. Tüm Yükleme, yedekleme, performans ölçme, sıralama, veri sıkıştırma, ve benzeri fonksiyonları yerine getirmek amacıyla bu ortamı kullandık.

5.2.2.5.2 VTYS'nin Sınıflandırılması En fazla kullanılan veri modelleri ilişkisel, ağ, hiyerarşik, nesne-yönelimli ve kavramsal modellerdir. Bizim Kullandığımız ise ilişkisel veri modelidir.

5.2.2.5.3 Hazır Program Kütüphane Dosyaları Zaten entegre olduğu için böyle bir şeye ihtiyaç duymadık.

5.2.2.5.4 CASE Araç ve Ortamları Case araçları olarak ise Microsoft un Visio Project ürünlerini kullandık.

5.3Kodlama Stili

Kendimize has kodlama bicini kullandık herhangi bir hazır düzene bağlı kalmadık Bakım programcımıza da aynı stil üzerine eğitim verdik ve sorunları ortadan kaldırdık

5.3.1 Açıklama Satırları

Açıklama satırları karmaşık her satırın sonunda yapıldı. Ve biten her günün sonunda kodun kalındığı yere o günün tarihi atıldı.

5.3.2 Kod Biçimlemesi

Kod biçimlemesine değinmek gerekirse alt alta oluşan kodlarda tabi indexleri kullandık ve iç içe bir biçimde hiyerarşi oluşturduk.

5.3.3 Anlamlı İsimlendirme

Sistem kodlamasının genel yapısında kullanılan değişkenlerin veri tabanında karşılığı varsa önce “tabloadı_islevadı_sayısı” seklinde bir anlamlı isimlendirme yaptık

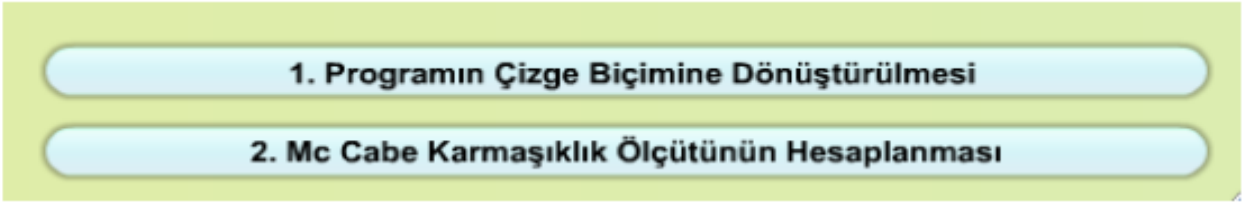
5.3.4 Yapısal Programlama Yapıları

Genel olarak 3 başlıkta incelersek:

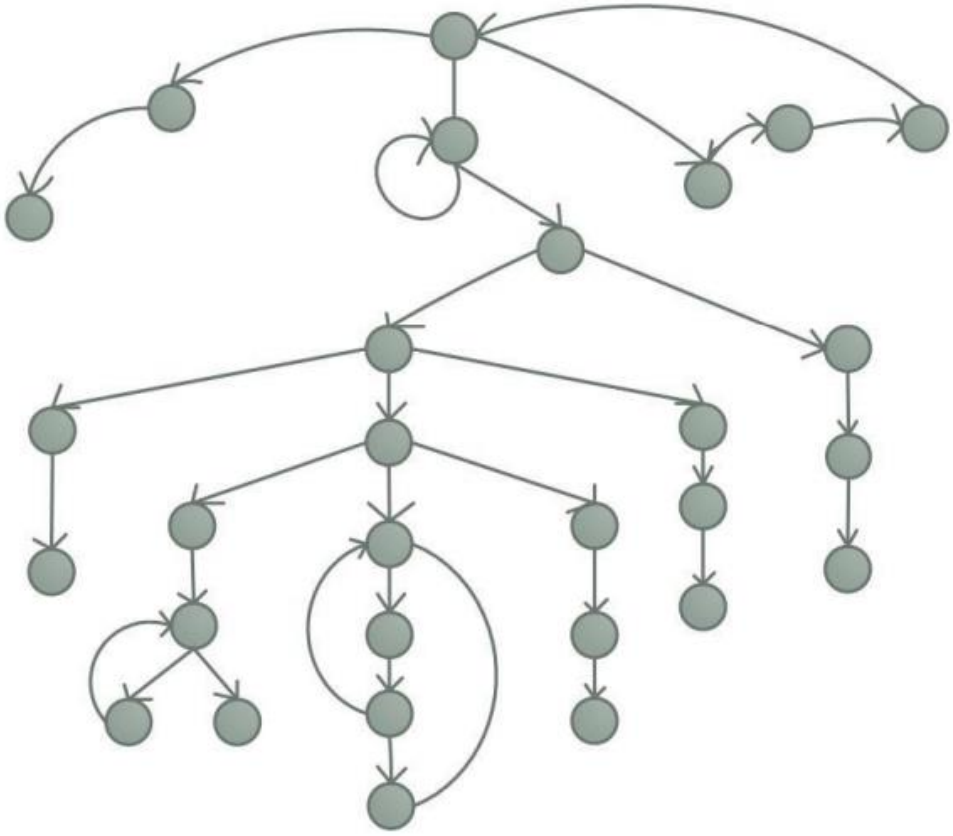
- **Ardışık işlem yapıları**: Bu tür yapılarda genellikle fonksiyon, altprogram ve buna benzer tekrarlı yapıları tek bir seferde çözdük.
- **Koşullu işlem yapıları**: Bu yapıları ise neredeyse programın tamamında kullandık karşılaştırma yapılan her yerde bunlara yer verildi.
- **Döngü yapıları**: Tıpkı ardışık işlemler gibi alt alta birkaç satır yazıcığımıza tek bir döngüyle bu sorunların üstesinden geldik.

5.4 Program Karmaşıklığı

Program karmaşıklığını ölçmek için bir çok teorik model geliştirilmiştir. Bu modellerin en eskisi ve yol göstericisi McCabe karmaşıklık ölçütüdür. Bu bölümde bu ölçüt anlatılmaktadır. Söz konusu ölçüt 1976 yılında McCabe tarafından geliştirilmiştir. Bu konuda geliştirilen diğer ölçütlerin çoğu, bu ölçütten esinlenmiştir. McCabe ölçütü, bir programda kullanılan "koşul" deyimlerinin program karmaşıklığını etkileyen en önemli unsur olduğu esasına dayanır ve iki aşamada uygulanır:



5.4.1 Programın Çizge Biçimine Dönüştürülmesi



5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama

k = 14 Kenar sayısı

d = 9 Düğüm sayısı

p = 1 Bileşen sayısı

V(G)=k-d+2p (Formülüyle bulunur)

V(G) = k – d + 2p

V(G) = 14 – 9+2*1 =7

5.5 Olağan Dışı Durum Çözümleme

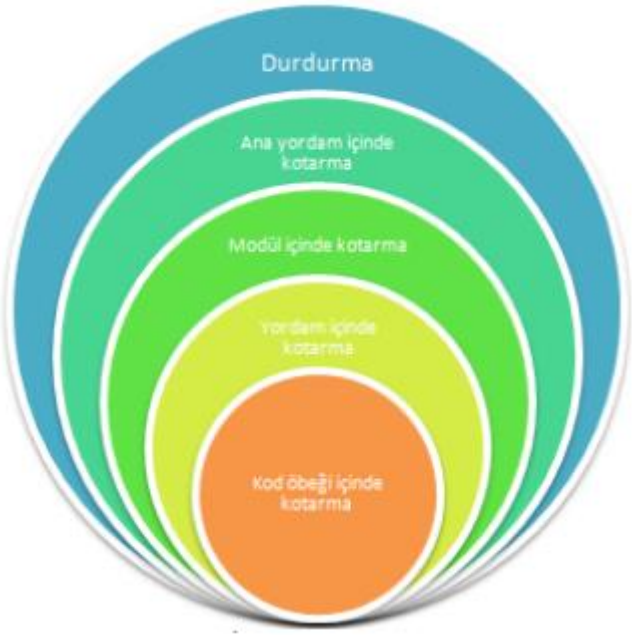
Olağan dışı durum, bir programın çalışmasının, geçersiz ya da yanlış veri oluşumu ya da başka nedenlerle istenmeyen bir biçimde sonlanmasına neden olan durum olarak tanımlanmaktadır

5.5.1 Olağandışı Durum Tanımları

Olağandışı gelişen durumlarda try-catch blokları devreye girecek ve program kırılmadan çalışmasına devam edebilecek şekilde tasarladık.

5.5.2 Farklı Olağandışı Durum Çözümleme

Yaklaşımları Tüm olağan dışı durumlarda program kırılmadan hata mesajlarıyla tekrar başa dönecek şekilde tasarladık.



5.6 Kod Gözden Geçirme

Hiç kimse, önceki sürümlerini gözden geçirmeden ve incelemeden okunabilir bir program yazamaz. Hiçbir yazı editörün onayını almadan basılamayacağı gibi hiçbir program da incelenmeden, gözden geçirilmeden işletime alınmamalıdır. Kod gözden geçirme ile program sınama işlemlerini birbirinden ayırmak gerekir. Program sınama, programın işletimi sırasında ortaya çıkabilecek yanlış ya da hataları yakalamak amacıyla yapılır. Kod gözden geçirme işlemi ise, programın kaynak kodu üzerinde yapılan bir incelemedir. Kod gözden geçirmelerinde program hatalarının %3-5 oranındaki kesimi yakalanabilmektedir. Eğer programı yazan kişi, yazdığı programın hemen sonra bir "kod inceleme" sürecine girdi olacağını bilerek program yazdığında daha etkin, az hatalı ve okunabilir programlar elde edilebilmektedir.

5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi

Gözden geçirme sürecinin temel özellikleri;

- Hataların bulunması, ancak düzeltilmemesi hedeflenir,
- Olabildiğince küçük bir grup tarafından yapılmalıdır. En iyi durum deneyimli bir inceleyici kullanılmasıdır. Birden fazla kişi gerektiğinde, bu kişilerin, ileride program bakımı yapacak ekipten seçilmesinde yarar vardır.
- Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır. biçiminde özetlenebilir. Burada yanıtı aranan temel soru, programın yazıldığı gibi çalışıp çalışmayacağının belirlenmesidir

5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular

Bir program incelenirken, programın her bir öbeği (yordam ya da işlev) aşağıdaki soruların yanıtları aranır. Bu sorulara ek sorular eklenebilir. Bazı soruların yanıtlarının "hayır" olması programın reddedileceği anlamına gelmemelidir.

5.6.2.1 Öbek Arayüzü

Oluşturduğumuz öbekleri test etmek için belli sorular sorduk bu sorular:

- Her öbek tek bir işlevsel amacı yerine getiriyor mu?
- Öbek adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?
- Öbek tek giriş ve tek çıkışlı mı?
- Öbek eğer bir işlev ise, parametrelerinin değerini değiştiriyor mu?

Şeklinde oldu.

5.6.2.2 Giriş Açıklamaları

Oluşturduğumuz giriş açıklamalarını test etmek için belli sorular sorduk bu sorular:

- Öbek, doğru biçimde giriş açıklama satırları içeriyor mu?
- Giriş açıklama satırları, öbeğin amacını açıklıyor mu?
- Giriş açıklama satırları, parametreleri, küresel değişkenleri içeren girdileri ve kütükleri tanıtıyor mu?
- Giriş açıklama satırları, çıktıları (parametre, kütük vb) ve hata iletilerini tanımlıyor mu?
- Giriş açıklama satırları, öbeğin algoritma tanımını içeriyor mu?
- Giriş açıklama satırları, öbekte yapılan değişikliklere ilişkin tanımlamaları içeriyor mu?
- Giriş açıklama satırları, öbekteki olağan dışı durumları tanımlıyor mu?
- Giriş açıklama satırları, Öbeği yazan kişi ve yazıldığı tarih ile ilgili bilgileri içeriyor mu?
- Her paragrafı açıklayan kısa açıklamalar var mı?

Şeklinde oldu.

5.6.2.3 Veri Kullanımı

Oluşturduğumuz veri kullanımlarını test etmek için belli sorular sorduk bu sorular:

- İşlevsel olarak ilintili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?
- Değişken adları,işlevlerini yansıtacak biçimde anlamlı mı?
- Değişkenlerin kullanımları arasındaki uzaklık anlamlı mı?
- Her değişken tek bir amaçla mı kullanılıyor?
- Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?
- Tanımlanan her gösterge değişkeni için bellek ataması yapılmış mı?

Şeklinde oldu.

5.6.2.4 Öbeğin Düzenlenişi

- Modüller birleşimi uyumlumu?
- Modüller arası veri aktarımları sağlanıyor mu?
- Bütün modüller birleştiğinde sistem çalışıyor mu? Gözden geçirme sırasında referans alınacak sorular olacaktır.

5.6.2.5 Sunuş

Artık son kısma gelindiğinde ise şu sorular soruldu:

- Her satır, en fazla bir deyim içeriyor mu?
- Bir deyimin birden fazla satıra taşması durumunda, bölünme anlaşılabilirliği kolaylaştıracak biçimde anlamlı mı?
- Koşullu deyimlerde kullanılan mantıksal işlemler yalın mı?
- Bütün deyimlerde, karmaşıklığı azaltacak şekilde parantezler kullanılmış mı?
- Bütün deyimler, belirlenen program stiline uygun olarak yazılmış mı?
- Öbek yapısı içerisinde akıllı "programlama hileleri" kullanılmış mı?

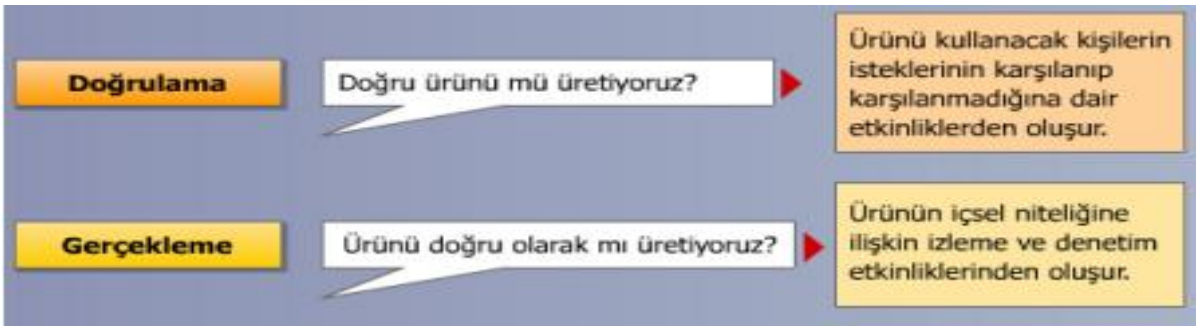
6. DOĞRULAMA VE GEÇERLEME

6.1. Giriş

Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlenmesi, üretim süreci boyunca süren etkinliklerden oluşur. Söz konusu etkinlikler:

- Yazılım belirtimlerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtimleri tutarlı olarak betimler durumda olduğunun doğrulanması
- Proje süresince her bir etkinlik ürününün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitenin tekrarına sebep olması.
- Projenin bir aşaması süresince geliştirilen anahtar belirtimlerin önceki belirtimlerle karşılaştırılması.

Yazılım ürünlerinin tüm uygulanabilir gerekleri sağladığının gerçeklenmesi için sınamaların hazırlanıp yürütülmesi biçiminde özetlenebilir.



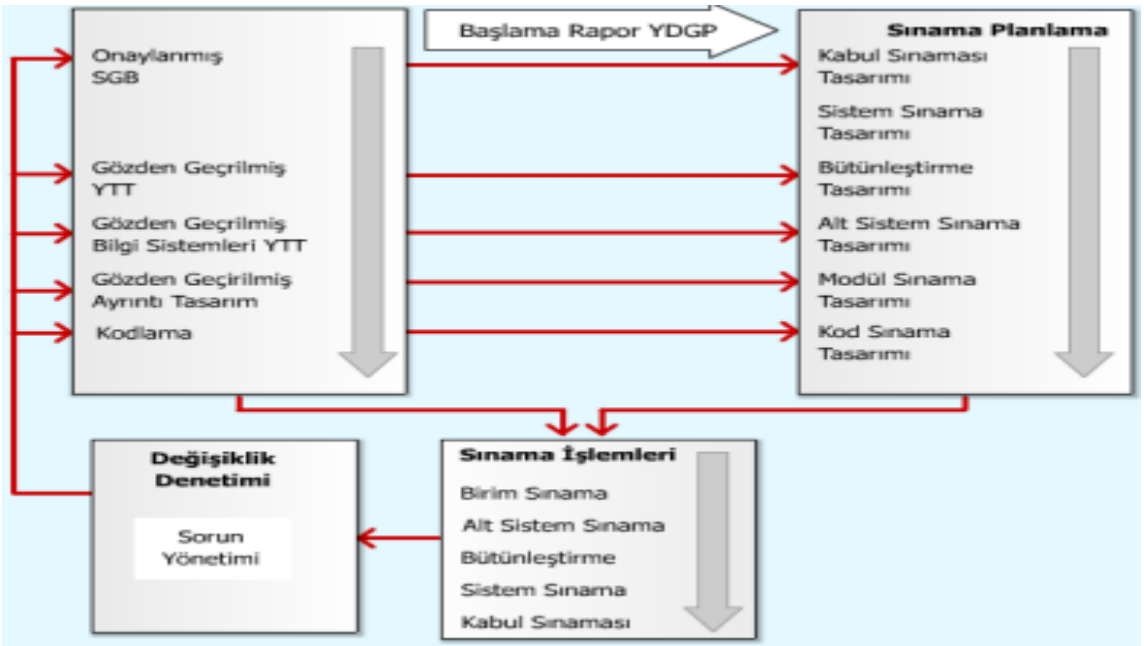
6.2. Sınama Kavramları

Birim Sınama: Sistemin birimleri İşletmeler kendi içlerinde birimleri sınandı ve sonuçları çıkartıldı.

Alt Sistem Sınama: Birimlerin birleşmesiyle modüller oluşturulup bunların kendi içinde sınaması yapıldı. Genel olarak arayüzde ki eksiklikler giderildi.

Sistem Sınama: Sistemin bütün olarak sınanması yapıldı ve programın eksiksiz olduğu onaylandı.

Kabul Sınama: Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha onaylandı



6.3 Sınama Planlaması

Bir tablo ile özetlemek gerekirse şu şekilde özetleyebiliriz.

Test raporu hazırlanırken şu özellikler mutlaka planda belirtilmelidir;

Test planı kimliği: Test planının adı veya belge numarası

Giriş: Test edilecek yazılımın elemanlarının genel tanıtım özetleri. Ayrıca bu plan kapsamı ve başvurulan belgeler. Kısaltmalar ve terim açıklamaları bu bölümde bildirilmelidir.

Test edilecek sistem: Sistemde bileşenleri sürüm sayıları olarak sıralar ve sistemin özelliklerini bileşenlerini ve nasıl kullanıldıkları açıklanmalıdır. Ayrıca sistemde test edilmeyecek parçalar belirtilmelidir.

Test edilecek ana fonksiyonlar: Sistemin test edilecek ana fonksiyonlarının kısa bir tanıtımı yapılmalıdır.

Test edilmeyecek ana fonksiyonlar: Sistemde test edilmeyecek fonksiyonları ve bunların neden test edilmedikleri açıklanacaktır. Geçti/Kaldı Kriterleri: Bir test sonucunda sistemin geçmiş veya kalmış sayılacağını açıklanmalıdır.

Test dokümanı: Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır.

Sorumluluklar: Hangi kişilerin nelerden sorumlu olduğu ve test takım lideri bilgileri mutlaka raporda belirtilmelidir.

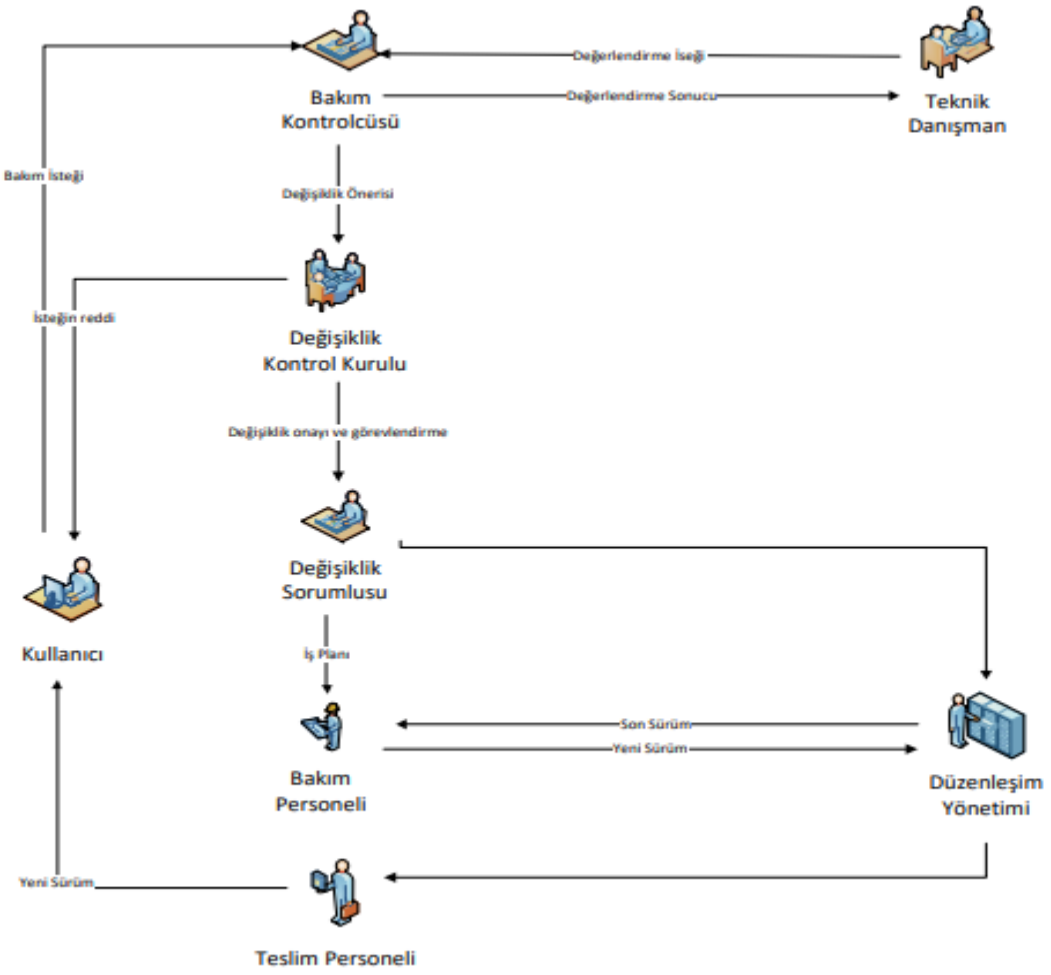
Riskler ve Önlemler: Test planında varsayılan ve olası yüksek riskli durumları belirtir ve bu durumların olması durumunda, etkilerinin en aza indirilebilmesi için alınması gereken

önlemleri açıklar.

7.BAKIM

7.1 Giriş

Sistemin tasarımı bittikten sonra artık seçimden seçime sistemin bakıma sokulması gerekir daha öncede belirttiğimiz gibi sistem hassas ve hata kabul etmeyecek bir sistemden bahsediyoruz. Bakım bölümüne ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı baz olarak alınmıştır



7.2 Kurulum

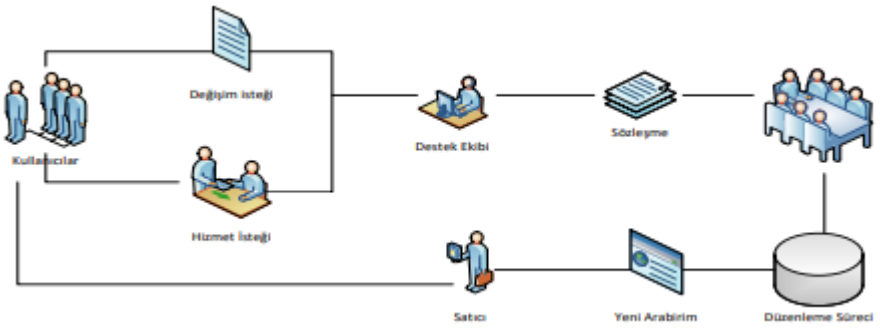
Sistem kurulumuna değinmek gerekirse devlet güvencesinde verilecek olan serverlara yüklenecek olan sistemimizde FTP arayüzü ile dosyaları servera aktaracağız ve internet explorer olan tüm cihazlarda çalışacak.

7.3 Yerinde Destek Organizasyonu

Bu konuyla ilgili pilot bölgede bizzat desteği ben vereceğim bunun yanında sistem canlandırılıp gerçeğe geçirilirse sistem tanımlaması kurulum için bölgelerde bayilik sistemi gibi alt kuruluşlara yetki verilecek eğer profesyonel destek istenirse yol uçak

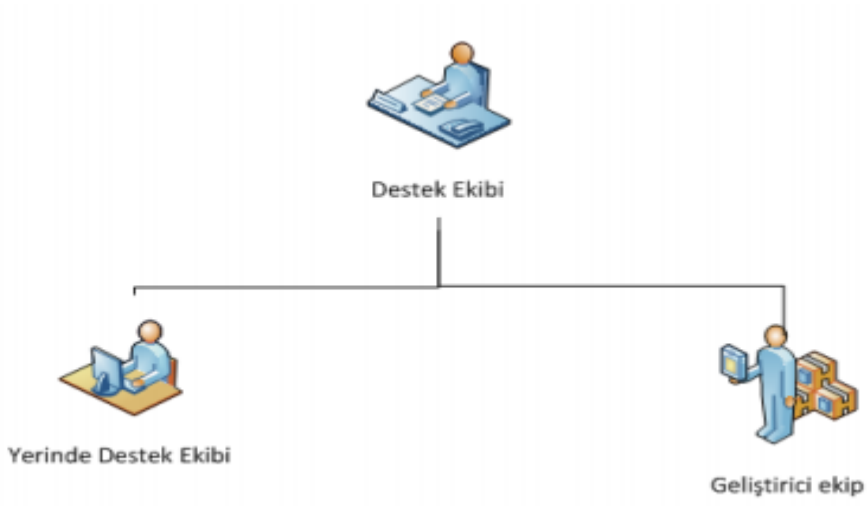
masrafını karşılamak şartıyla bölgeye yetkili gönderilip orada bir organizasyon yapılacaktır.

7.4 Yazılım Bakımı



7.4.1 Bakım Süreç Modeli

Aslına bakmak gerekirse bakım süreç modeli yukardaki yapılan işlemlerin tümünün baştan yapılması demek bunları adım adım bir inceleyelim.



1. Adım: Sorunu Tanımlama Süreci İlk önce bakım ne için yapılıyor sorun ne buna bir bakalım.

2. Adım: Çözümleme Süreci Sorun tanımlamadan çıkan karar doğrultusunda problemi kâğıt üzerinde çözelim.

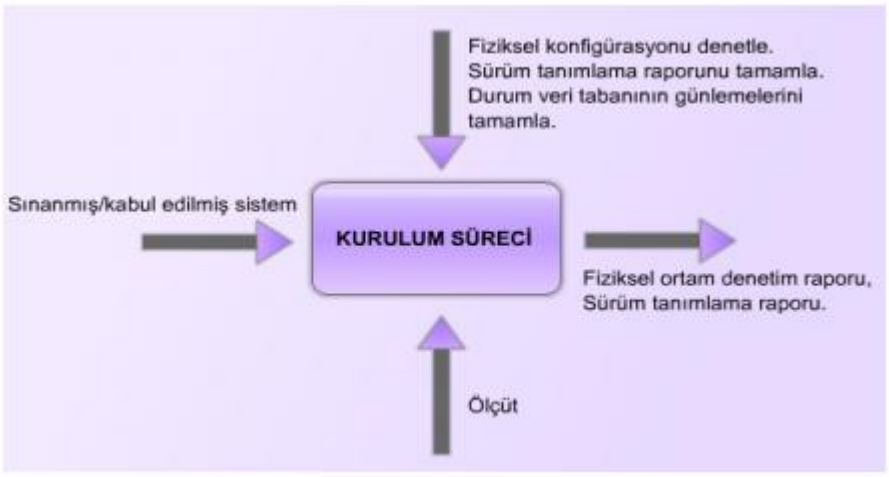
3. Adım: Tasarım Süreci Çözümlenen sistem sonucunda tasarımı güncelleştirmeye geldi sıra.

4. Adım: Gerçekleştirim Süreci Tasarımı yapılan sistemin gerçekleştirmesine sıra geldi.

5.Adım: Sistem Sınama Süreci Artık tekrardan tasarlanan sistemin sınama sürecini tekrar ele almak gerekiyor.

6. Adım: Kabul Sınaması Süreci Kendi içimizde sınadığımız sistemi birde müşteri karşısında sınıyoruz.

7. Adım: Kurulum Süreci Kabul sınamasını geçen sistemimiz artık tekrardan kurulum aşamasına geçiyor.



8.Sonuç

Sonuç olarak sistem hayata geçirildiği zaman neler değişeceği gözler önüne serdik. Bunun yanı sırabasit ama bir o kadarda güvenli olan bu sistemle ek masraflar ortadan kalkacak hataları ortadan kaldırılacak ve kağıt masrafını azaltarak güzel bir satış ve stok programı elde ettik.

9.KAYNAKLAR

İnternet Kaynakları

I. http://muhammetbaykara.com/
II. http://www.canbalaman.com/
III. http://www.kriptarium.com/pd.html

IV.