T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YMH319-PROGRAMLAMA DİLLERİ DERSİ

PROJE DOKÜMANTASYONU

TEZ KONTROL SİSTEMİ

GELİŞTİREN

190541065- AHMET ÇEKİN

OCAK-2021

ÖNSÖZ

Bu projeyi gerçekleştirirken yardımlarını esirgemeyen başta hocam Doç. Dr. Fatih ÖZKAYNAK' a, ailem ve arkadaşlarıma yardımlarından ve desteklerinden dolayı teşekkürü borç bilirim.

Ahmet ÇEKİN

İÇİNDEKİLER

5. SİSTEM GERÇEKLEŞTİRİMİ

6. DOĞRULAMA VE GEÇERLEME

1.GİRİŞ

1.1 Projenin Amacı

Projemiz son sınıf öğrencilerinin hazırlamış olduğu bitirme tezlerini kontrol etmek üzere yapılmak istenen bir projedir. Günümüz teknolojisinin ilerlemesi insan gücünün yerini makinelere bırakmaktadır. Bu makinelerden bizim için neredeyse her alanda bizlere yardımcı olan bilgisayarlar bulunmaktadır. Bizler birer bilgisayar bilimcisi olarak teknolojiden maksimum düzeyde faydalanmak zorundayız. Bu sebeple bakıldığında tez kontrol aşamasının oldukça zahmetli ve uzun bir iş olduğunu söyleyebiliriz. Sürecin bu denli uzun ve zahmetli olması olası insan hatalarının da artmasına neden olacağı kaçınılmazdır. Hayata geçirmek istediğimiz bu projenin ortaya çıkan bu sorunları tamamıyla ortadan kaldıracağını ve tez kontrol aşamasını daha kolay hale getireceğine inanıyoruz.

Projemiz uzun dokümanları dakikalar ile ifade edilebilecek bir süre içinde kontrol edip tez kurallarına uygun olup olmadığını ve nerelerde, hangi kurallara uyulmadığını bizlere iletecek bir şekilde oluşturulacaktır. Bu kurallar önceden belirlenmiş dokümandan alınarak sistem için anlamsal olarak değil daha çok fiziksel anlamda entegre edilecektir.

Girdi olarak sisteme tezler (word) dosyası eklenecek ve çıktı olarak sisteme entegre edilen kurallar kapsamında tezin uygunluğu test edilecektir.

Bu sistem tez kontrolü yapan hocalarımıza zamandan ve olası hatalardan kazanç sağlayacaktır. Bunun yanı sıra tezi azan öğrenciler için teslim etmeden önce kontrol edebilme imkanı verecektir.

1.2 Projenin Kapsamı

Tez kontrol sistemi çok kullanıcı tarafından kullanılabilecek, kullanımı oldukça basit, arayüzleri sade ve anlaşılabilir bir şekilde tasarlanacak. Kapsam olarak bu tezlerin yazımı ve kontrolü aşamasında var olan tüm kullanıcıları kapsayacaktır. Daha çok eğitim alanını kapsadığı için üniversite gibi lisans ve lisansüstü derslerin doküman hazırlama durumunda kontrolü sağlanacaktır. Geliştirmeye açık kod kaynağına sahip olacağı için birçok alanda sürümü geliştirebilecektir. Web tabanlı veya mobil olarak da geliştirebilmenin önü açıktır. Projemiz gerçekleştirilip bakım aşamasında yapılacak testler ile oluşturulacak dokümanda ve anketlerden yola çıkılarak sistem için bu aşamalar göz önünde bulundurulacaktır.

1.3 Tanımlamalar ve Kısaltmalar

FA'ya göre M = (Q, Σ, δ, q_0 , F) ile tanımlanır

1- Sistemdeki Durumların Kümesi (Q);

- Tez Kapak Sayfası
- Önsöz Metni
- İçindekiler Tablosu
- Özet Metni
- Şekiller Listesi
- Tablolar Listesi
- Ekler Listesi
- Simgeler ve Kısaltmalar Listesi
- Giriş
- Kaynaklar

2- Sistemdeki Sembollerin Kümesi (Σ);

Sistemde girdimiz olan Word veya PDF dosyasını incelerken anlam bilgisine şu aşamada bakmadığımız için sembollerimiz sadece dokümanın fiziksel bakımdan belirlenen yazım kurallarından oluşan sembollerden oluşacaktır.

3- Geçiş Durumları

q_0 ile belirtilen durum sistem için durumlar kümesinde gelecek her bir durumda başlangıç durumu (initial state) olarak kabul edilecektir.

▬▬▬ Şekli ile sistemin oluşturulan diyagramdan kabul gördüğünü ifade eder.

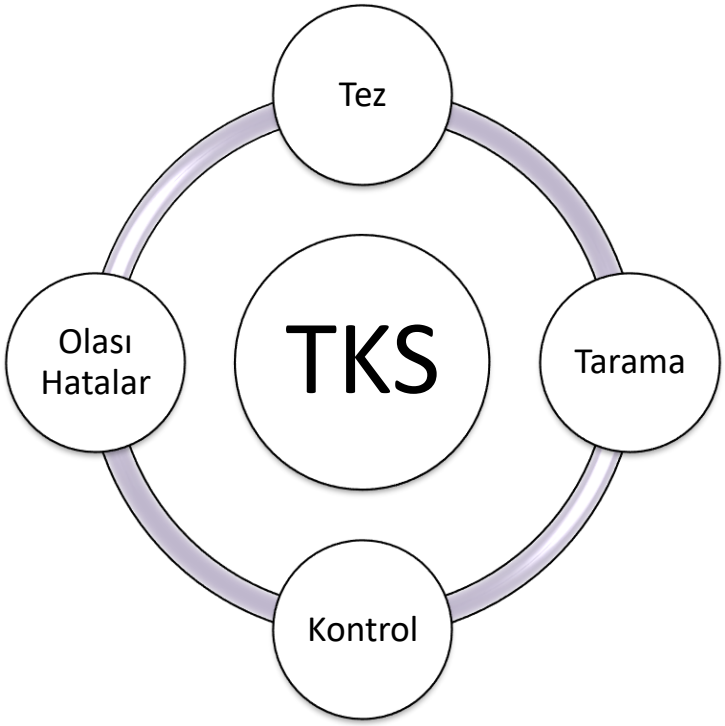
4- TKS: Tez Kontrol Sistemi

5- Final state olarak belirlenen durum sistemin kontrol edilen durumun en son adımı olarak belirtilecektir. Bu aşamada her bir adımda alınan kontrol hataları trap olarak belirlenen hata durumuna geçiş yapacaktır.

2. PROJE PLANI

2.1 Giriş

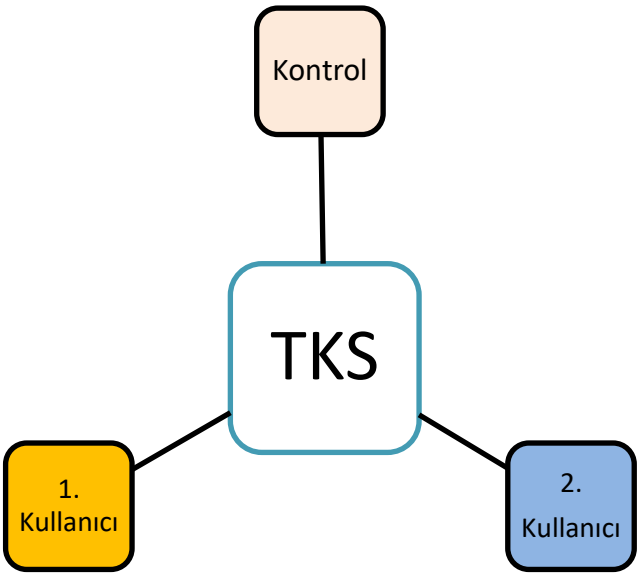
Tez Kontrol Sistemi tezin oluşturulması gereken kurallarını barındırır. Bu kuralları her bölüm için ayrı ayrı ele alır. Yaptığı taramada kurallar dışında herhangi bir durum ile karşılaşırsa hata aldığı bölüm için hata mesajını yazdıracaktır. Tez Kontrol Sistemi için temel yaklaşım aşağıdaki Şekil 2.1' de belirtilmiştir. Lütfen inceleyiniz.



Şekil 2.1. TZK'nin Temel Yapısı

2.2 Projenin Plan Kapsamı

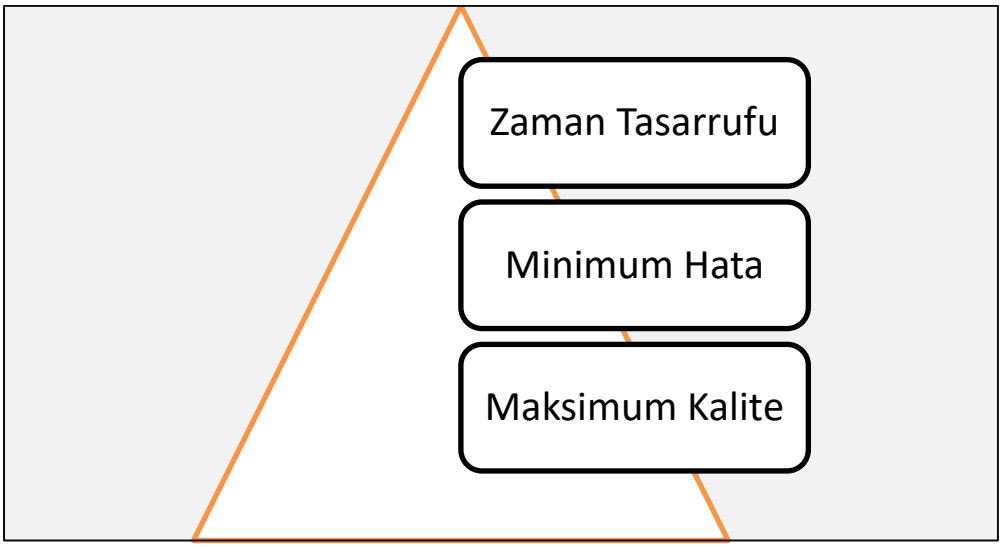
Bu proje, tamamen tez kontrolü sırasında olası tüm insan kaynaklı hataların minimize edilmesi ve zamandan tasarruf etmek adına tasarlanacaktır. Günümüz gelişen ve gelişmekte olan teknolojinin faydalarını kullanmak ve bir bilgisayar bilimcisinin insanların hayatını kolaylaştıracak bir amaçla bir mühendis yaklaşımında bulunmaktır. Hem hocalarımızın hem de tez hazırlayan öğrencilerin kullanabileceği genel aynı zamanda kullanışlı bir sistem oluşturmak temel hedeflerdendir.



Şekil 2.2. Kullanıcı İlişkileri

TKS Neden Gerekli?

- Gelişen teknolojiden olabildiğince yararlanmak.
- Oluşabilecek insan kaynaklı hataları yok denebilecek seviyeye indirmek.
- Kontrol sırasındaki harcanan zamanı minimize etmek.
- Tez yazım kurallarına bağlı kalarak oluşturulacak çalışmalar için maksimum kaliteyi sağlamak.



Şekil 2.3. TKS'nin avantajları

Tez Kontrol Sistemi Nasıl Çalışır?

TKS bir adet arayüzden oluşacaktır. Bu arayüz içerisinde kontrol edilmek istenen tez dokümanı Word dosyası yani .docx formatında olmalıdır. Arayüzde kontrol edilecek tezin bilgisayarda bulunduğu konum eklenecektir. Bu yolun arayüze eklenmesiyle kontrol et butonu aktif hale gelecek ve dosyanın bulunduğu konumu tarayacaktır. Dosya belirtilen konumda bulunmazsa veya dosyanın biçimi .docx biçiminde değilse hata mesajı ayrı bir panelden kullanıcıya sunulacaktır. Eklenen dosya yolu doğru ise ve kontrol edilecek tez dosyası .docx formatında ise sistem kontrole başlayacaktır. Sistemin arka tarafında taranacak tezin boyutuna göre tarama işleminin boyutu farklı sürelerde farklı değerler gösterebilir. Oluşan hatalar aynı arayüz içerisinde ayrı bir box içinde kullanıcıya satır bazlı sunulacaktır.

Maliyet Kestirim Dokümanı

Tez Kontrol Sistemi

Ölçüm Parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	1	4	4
Kullanıcı Çıktı Sayısı	10	3	30
Kullanıcı Sorgu Sayısı	1	6	6
Kütük Sayısı	1	5	5
Dışsal Arayüz Sayısı	1	3	3
Ana İşlev Nokta Sayısı			66

Teknik Karmaşıklık Tablosu	Puan
1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	1
2. Veri iletişimi gerektiriyor mu?	5
3. Dağıtık işlem işlemleri var mı?	2
4. Performans kritik mi?	8

5. Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak?	4
6. Sistem çevrimiçi veri girişi gerektiriyor mu?	1
7. Çevrimiçi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu?	1
8. Ana kütükler çevrimiçi olarak mı güncelleniyor?	1
9. Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı?	5
10.İçsel işlemler karmaşık mı?	2
11.Tasarlanacak kod, yeniden kullanılabilir mi olacak?	4
12.Dönüştürme ve kurulum, tasarımda dikkate alınacak mı?	5
13. Sistem birden çok yerde yerleşik farklı kurumlar için mi geliştiriliyor?	2
14. Tasarlanan uygulama kolay kullanılabilir ve değiştirilebilir mi?	5
TOPLAM	46

0: Hiçbir etkisi yok
1: Çok az etkisi var
2: Etkisi var
3: Ortalama etkisi var
4: Önemli etkisi var
5: Mutlak olmalı, kaçınılmaz

İN = AİN * (0,65 * 0,01 * TKF)

İN = 48 * (0,65 * 0,01 * 46)

İN = 14,352

Satır Sayısı = İN * 30

Satır Sayısı = 14,352 * 30 = 430,56 Yaklaşık 430 Satır

Etkin Maliyet Modeli – COCOMO

Organik Projeler İçin: a=2,4 , b=1,05 , c=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 Tez Kontrol Sisteminin türünü belirlememiz gerekiyor. Bu projemiz küçük ekip tarafından gerçekleştirildiği için organik proje grubuna dahil olmaktadır.

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*0,81,05=1,89

Geliştirme Süresi =D= 2,5 x = 4 ay

Tahmini Gerekli Personel Sayısı = 3,18 / 1,89= 2 (yaklaşık)

2.3 Proje Zaman-İş Plan

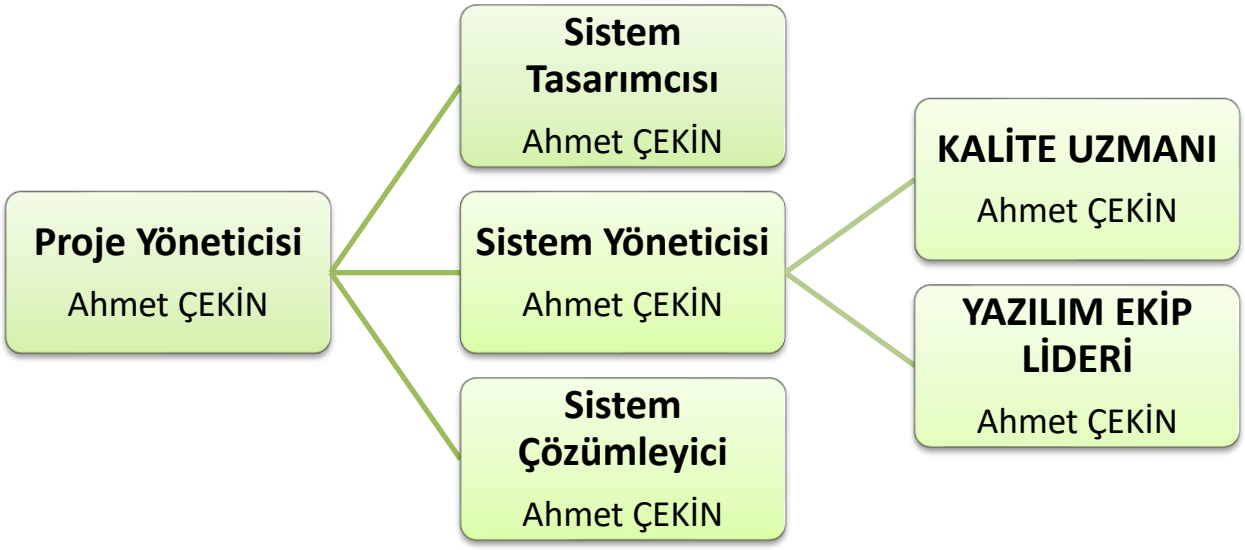
İş-Zaman Çizelgesi

zaman / İş	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta	10. Hafta
Proje Teklifi	√									
Proje Planı		∞	√							
Analiz			∞	√						
Sistem Çözümleme				∞	∞	√				
Kullanıcı Arayüz Tasarımı						∞	√			
Gerçekleşti-rim							∞	∞	√	
Test									√	
Sunum										√

Şekil 2.4. İş-Zaman Çizelgesi

- '∞' işareti belirtilen zaman aralığında belirlenen işe başlandığını temsil etmektedir.
- '√' işareti belirtilen zaman aralığında belirlenen işin bittiğini temsil etmektedir

2.4 Proje Ekip Yapısı



Şekil 2.6 Proje Ekip Şeması

Proje Yöneticisi	• Projenin yönetimi • Proje ekip yapısının oluşturulması • İş planının yapılması
Sistem Çözümleyici	• Dokümantasyon ve raporların hazırlanması
Araştırma Ekibi	• Proje için gerekli kaynak temini Örnek sistemlerin incelenmesi
Web Tasarımı	• Arayüz ve grafiksel tasarımların yapılması
Programcı	• Tasarım ve kodlama
Eğitim Ekibi	• Programcı eğitimi • Kullanıcı eğitimi
Test ve Bakım Ekibi	• Sistem testlerinin yapılması • Sistem bakım planının yapılması

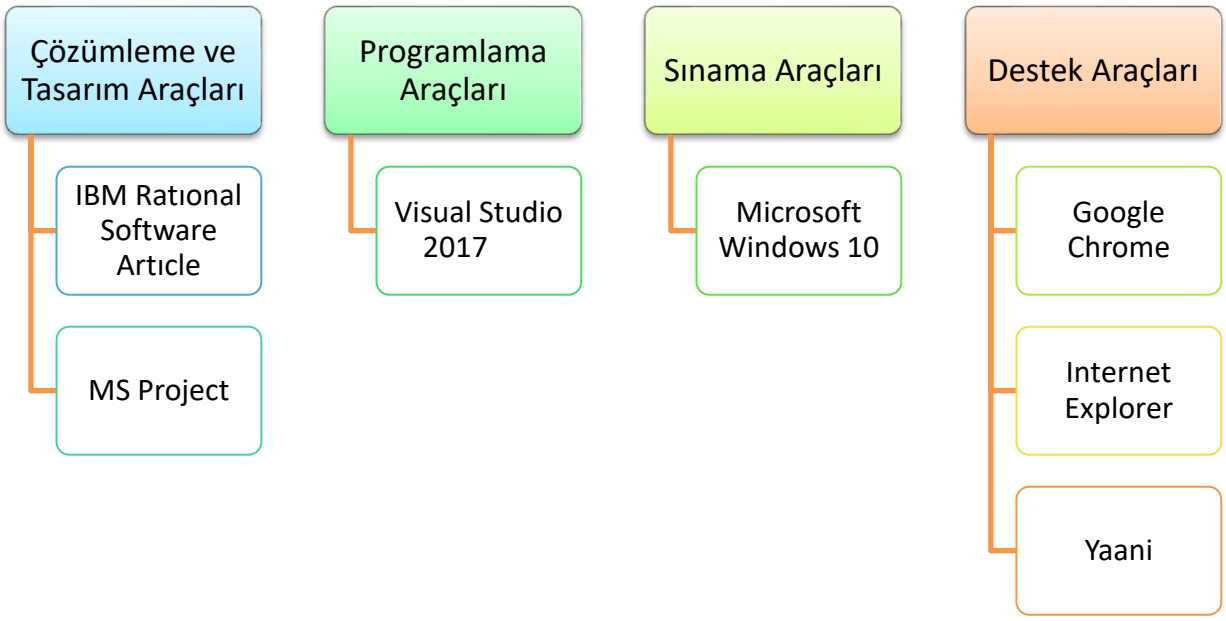
Şekil 2.7 Proje Ekip Görevleri

2.5 Önerilen Sistemin Teknik Tanımları

Kullanılan teknolojiler

- Sistemimiz C# programlama dili kullanılarak hayata geçirilecektir. C# programlama dilinin güçlü ve dinamik olması durumundan dolayı seçimimizde etkili olmuştur. Bunun yanı sıra projenin güçlü arayüze sahip olabilmesi ve C# dilinin bize sunduğu birçok özellikten faydalanabilmek için geliştirme aracı olarak Microsoft Visual Studio kullanılacaktır.

2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları



Şekil 2.8 Geliştirme Ortamları ve Araçları

2.7. Proje Standartları, Yöntem ve Metodolojiler

Tez Kontrol Sisteminin proje aşamaları Şekil 2.9'da gösterildiği gibidir. Bunun yanı sıra kullanılan sistem modelinde ise Çağlayan (Şelale) Model kullanılmıştır.

İyi tanımlama işlemi için bu metodolojinin kullanılması kaçınılmazdır. Süreç içerisindeki tekrarlamalar, projede bulunan aşamalar arasındaki geçişlerle sağlanması gerekli olup, bu modelin daha yararlı olacağı saptanmıştır. Bu modelde proje yöneticilerinin iş dağılımı yapmasının kolay olması, kalite gereksiniminin yüksek olması bu proje için önemlidir.

Aşama	Kullanılan Sistem ve Araçlar	Ne için Kullanıldığı	Çıktı
Planlama	-Veri akış şemaları -Maliyet kestirim yöntemleri -Proje Yönetimi	-Kaynak kestirimi -Süreç Kestirimi	-Proje Planı
Çözümleme	-Veri Sözlüğü -Nesne İlişki şemaları -Veri Akış Şemaları	-Veri Çözümleme -Süreç Çözümleme	-Sistem Çözümleme Raporu
Çözümlemeden Tasarıma Geçiş	-Programın tasarım koduna dönüştürülmesi - Nesne İlişki tablolarının Veri Tablolarına dönüştürülmesi	-Başlangıç ve Veri Tasarımı	-Başlangıç Tasarım Raporu
Tasarım	-Yapısal Şemalar -Program Tasarım Dili -Veri Sözlüğü	-Genel ve Ayrıntılı Tasarım -Veri Tasarımı	-Sistem Tasarım Raporu

Şekil 2.9 Yöntem ve Metodolojiler

Ekip Yapısı Zaman-İş Planı						
Aylar / Ekip	Ekim		Kasım		Aralık	
Sistem Çözümleyici	√	√				
Sistem Yöneticisi	√	√	√	√	√	√
Sistem Tasarımcısı			√	√	√	√

Şekil 2.10 Ekip Yapısı Zaman Planı

2.8 Kalite Sağlama Planı

Şekil 2.11 Kalite Sağlama Planı

2.10 Kaynak Yönetim Planı

Hastane Ziyaretçi Takip Sistemi projesinde kaynak olarak sadece mevcut doküman kullanılmıştır. Onun dışında hali hazırda yeni ve var olan kaynak kullanılmamıştır.
Kaynak yönetimi konusunda proje üyelerinin dikkat etmesi gereken hususlar bulunmaktadır. Bunlar genelde soru şeklinde olup üyelerin rahatlıkla cevap vermesi gereken özelliklerdir.

- Kalite beklendiği gibi mi?
- Proje elemanları yeni bir eğitim aldı mı?
- Yeterli hata bulunuyor mu?
- Yeterli vakit harcanıyor mu?
- İnceleyiciler yeterli hazırlık yapıyor mu?
- Sistem analizindeki sonuçlar istenilen doğrultuda mı?

2.11 Eğitim Planı

Bu sistemin geliştiricilerinden başlamak üzere kullanıcılara kadar olan insan kaynakları hususunda bir eğitim verilmesi gereklidir. Bu kapsamda öncelikle geliştiricilerin alacağı eğitimler belirlenmiş olup bu eğitimleri en ince detayına kadar bilmeleri gerekli bir özelliktir. Çünkü kullanıcıların oluşabilecek anlık değişim veya sorunlarına anında karşılık verebilme ve sorunu çözebilme özellikleri geliştiricilerden beklenen özelliklerdir. Bunun yarı sıra kullanıcıların da sistemi iyi tanımlı ve kullanış bakımında rahat ve daimi olarak işe yarayacak özelliklerini saptamak için alması gereken eğitimler vardır.

Geliştirici Eğitimleri;

- C# eğitimi
- Microsoft Word eğitimi

Kullanıcı Eğitimleri;

Kullanıcıların beklentisi projenin iyi tanımlı, kullanışlı ve hata içermeden kendilerine teslim edilmesidir. Bu süreçte kullanıcılar projenin yapımında dahil edilmesinden mümkün olduğunca kaçınılmalıdır. Kolay anlaşılacak ve sistemin kullanılmasını açık bir dille anlatan bir sunum ile kullanıcı eğitimi verilmesi yeterlidir.

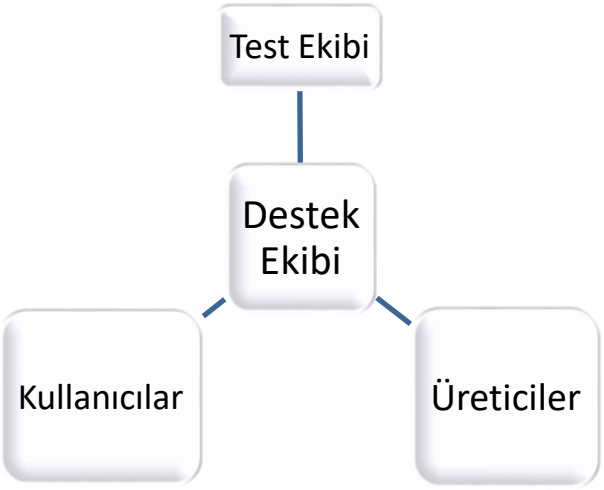
Tamamlana eğitimlerin devamında kullanıcı yorumu sistem üreticisi için önemlidir. Bu sebeple yapılan sistemi kullanan bir yetkili tarafından kullanıcılara bir seminer verilmelidir. Bu seminerin ardından sistemin işleyişi hakkında kullanıcılardan alınacak yorumlar dikkate alınmalıdır. Aynı zamanda yapılan sistem ile ilgili kullanıcılara anket yapılmalı bu ankette ise kullanıcının her türlü düşüncesi tahmin edilmeli ve o yöne dayalı sorular sorulmalıdır. Hesaplanacak anket puanında ise ortak düşüncelere ulaşılmalı ve bu ortak düşünceden çıkartılacak sonuç, yapılmış olan sistemin bakım aşamasında uygulamaya dökülmelidir.

2.12 Test Planı

Sistemin test ekipleri ile yapılması gereken birkaç adım bulunmaktadır. Öncelikle sistemin uygulamaya koyulabilmesi için, faklı bölümleri olan orta veya ileri büyüklükte bir hastane seçilecektir. Farklı yaş aralıklarında sistemin en çokta kullanılabilirlik özelliğinin test edilebilmesi için kullanıcılar belirlenecektir. Sistemin içindeki tüm kullanıcılar tek tek sistemdeki görevlerini yerine getirecek ve test ekipleri tarafından raporlanacaklardır.

2.13 Bakım Planı

Sistemin bakım planına bakacak olursak, verilen kullanıcı eğitiminden sonra yapılacak olan anket ve alınan yorumlar üzere bir bakım hareketi belirlenecektir. Bakımın yapılması gereken alan belirlenecek ve sistem içindeki ilgili birimlerce sorun ve istek giderilecektir.



Şekil 2.12 Üretim ve Kullanım Sahası

Normal üretim sahasının aksine, yaptığımız bu sistemin üretim sahasında bir de test ekibi olacaktır. Bu durum kullanıcı ile üretici arasındaki etkileşim ve ilişkiyi arttıracak aynı zamanda projenin müşteriden gelecek olan yönünün üretici tarafından iyi anlaşılır olmasını sağlayacaktır.

3. SİSTEM ÇÖZÜMLEME

3.1. Mevcut Sistemin İncelenmesi

Mevcut sistem için halihazırda bir sistem incelenmemiştir. Fakat gerçekleştirim aşamasında olan bir yazılım şirketin geliştirdiği buna benzer bir sistem için bilgi alınmıştır. Yapılan sistem de Tez Kontrol Sistemine benzemekte fakat daha çok tezin anlam bilgisinin kontrolünü yapmaktadır. Yazılan metinlerin diğer metinlerle olan benzerliği kontrol edilmektedir. Çıktı olarak belirli bir benzerlik oranı vermektedir. Bu sayede belirli bir oranın üstünde olan tezler özgün değildir ifadesi kullanılır ve kabul edilmez. Girdileri ve amaçları da geliştirdiğimiz sisteme ne kadar benzese de birebir olmadığı için mevcut sistem için inceleme alanımız kısıtlıdır.

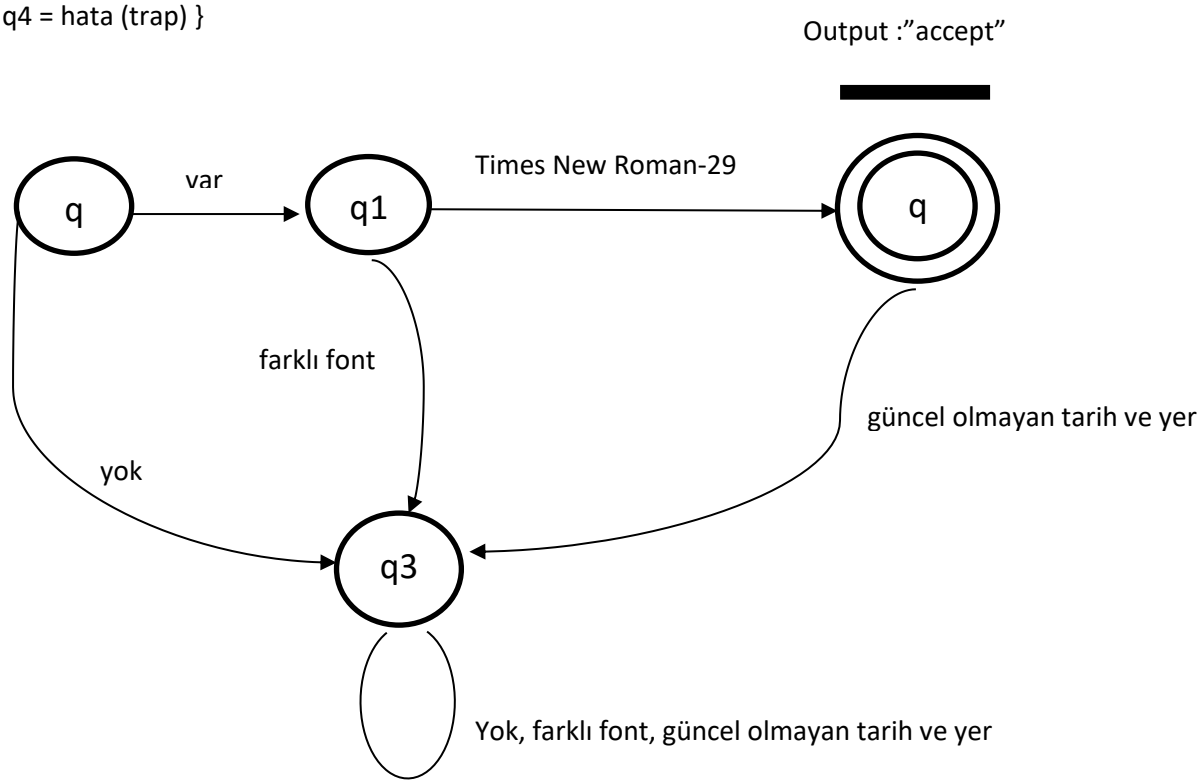
3.1.2. İşlevsel Model

Sistemdeki Durumların Kümesi (Q);

- Tez Kapak Sayfası
- Önsöz Metni
- İçindekiler Tablosu
- Özet Metni
- Şekiller Listesi
- Tablolar Listesi
- Ekler Listesi
- Simgeler ve Kısaltmalar Listesi
- Giriş
- Kaynaklar

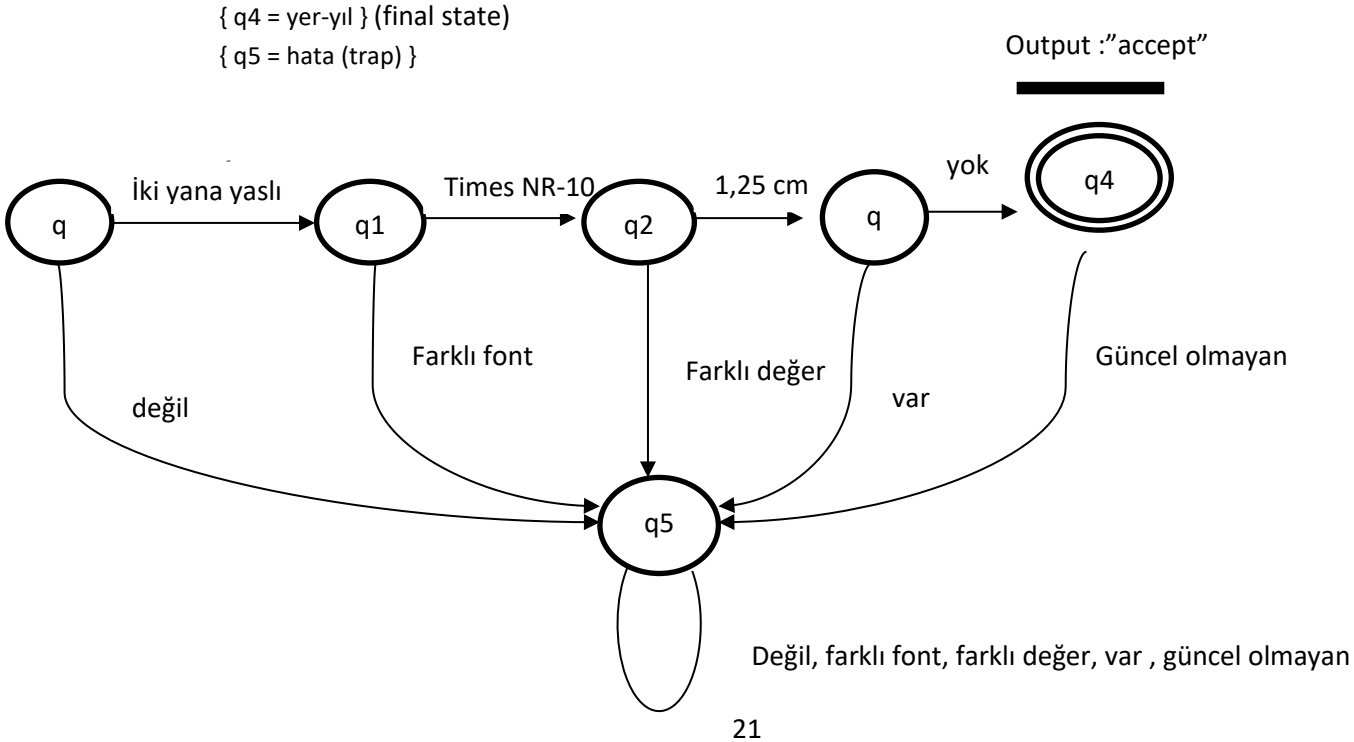
a- Q (Tez Kapak Sayfası)

{ q0 = üniversite logosu } (initial state)
{ q1 = ana başlık fontu }
{ q2 = tarih ve yer ismi } (final state)
{ q4 = hata (trap) }



b- Q (Önsöz Metni)

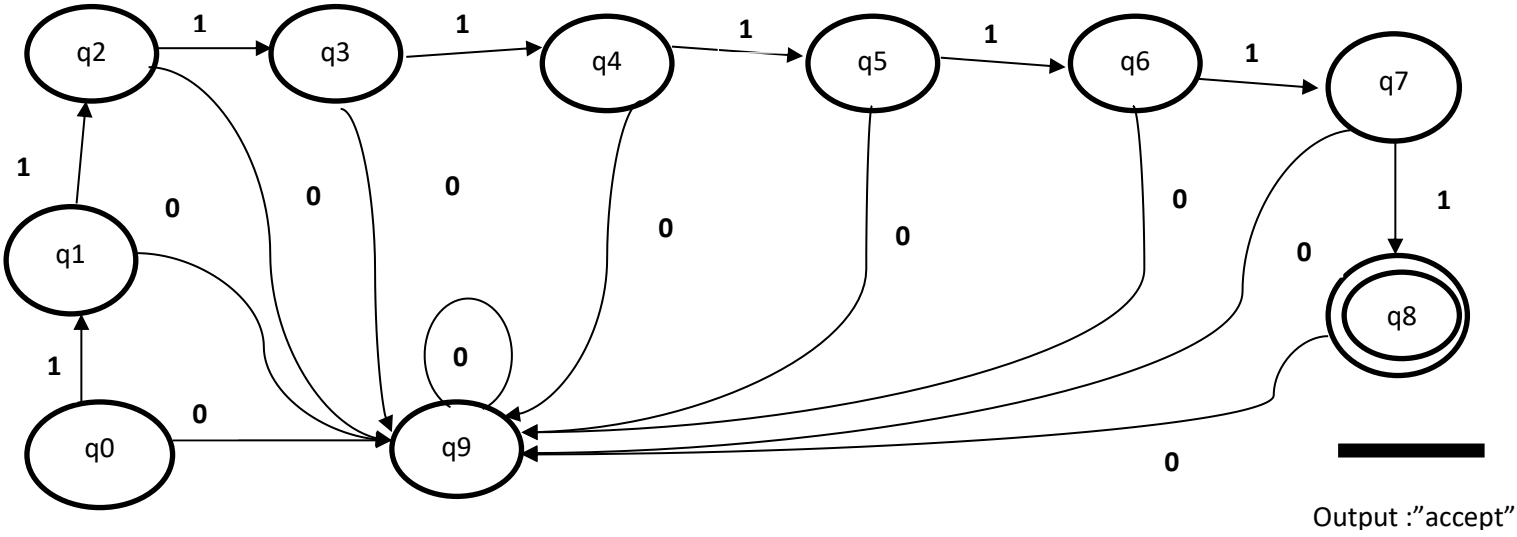
{ q0 = iki yana yaslı } (initial state)
{ q1 = yazı fontu }
{ q2= satır boşluğu }
{ q3 = teşekkür ifadesi }
{ q4 = yer-yıl } (final state)
{ q5 = hata (trap) }



c- Q (İçindekiler Tablosu)

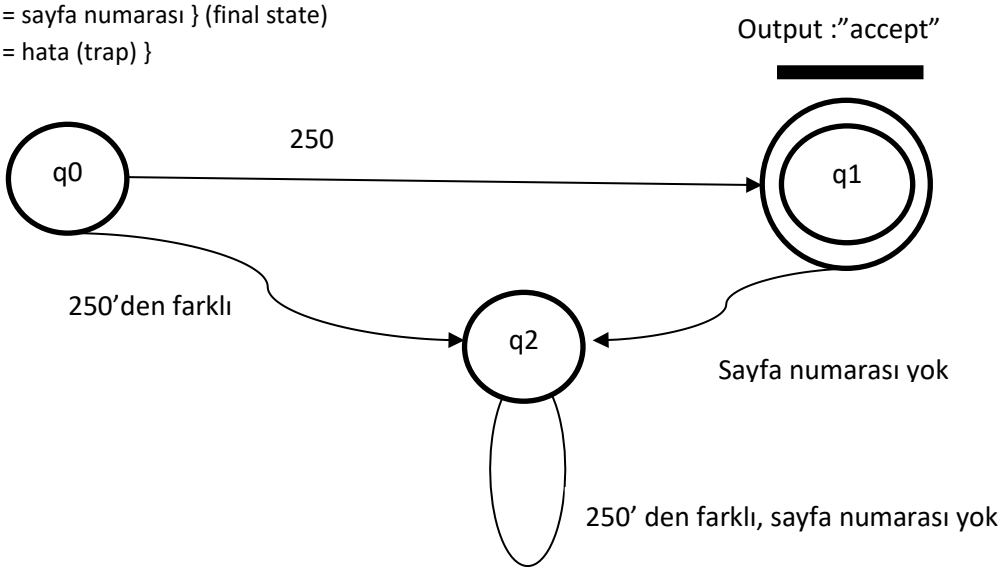
{ q0 = ön bölümler kısmında roman rakamları } (initial state)
{ q1 = zorunlu başlıklar }
{ q2 = başlıklar metin içerisindeki başlıklara uygun }
{ q3 = sayfa numarası verilmesi }
{ q4 = sayfa numarası metin içerisinde tutarlı }
{ q5 = başlıklarda liste numarası }
{ q6 = özgeçmiş varsa sayfa numarası }
{ q7 = ekler bölümü varsa ekler listesi }
{ q8 = ana bölüm başlık fontu } (final state)
{ q9 = hata (trap) }

// şekil karmaşıklığının artmaması için olması gereken simgeler 1, zıttı olanlar 0 ile ifade edilmiştir.



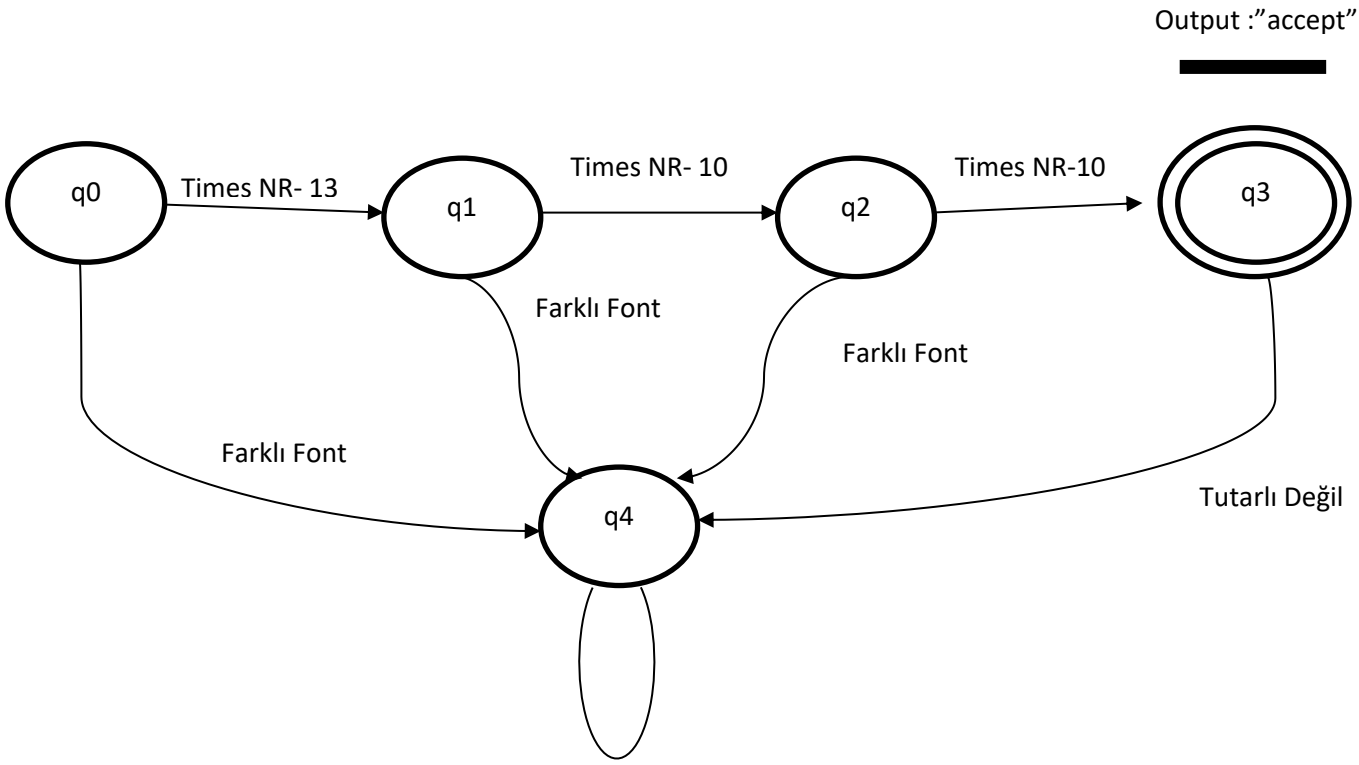
d- Q (Özet Metni)

{ q0 = kelime sayısı } (initial state)
{ q1 = sayfa numarası } (final state)
{ q2 = hata (trap) }



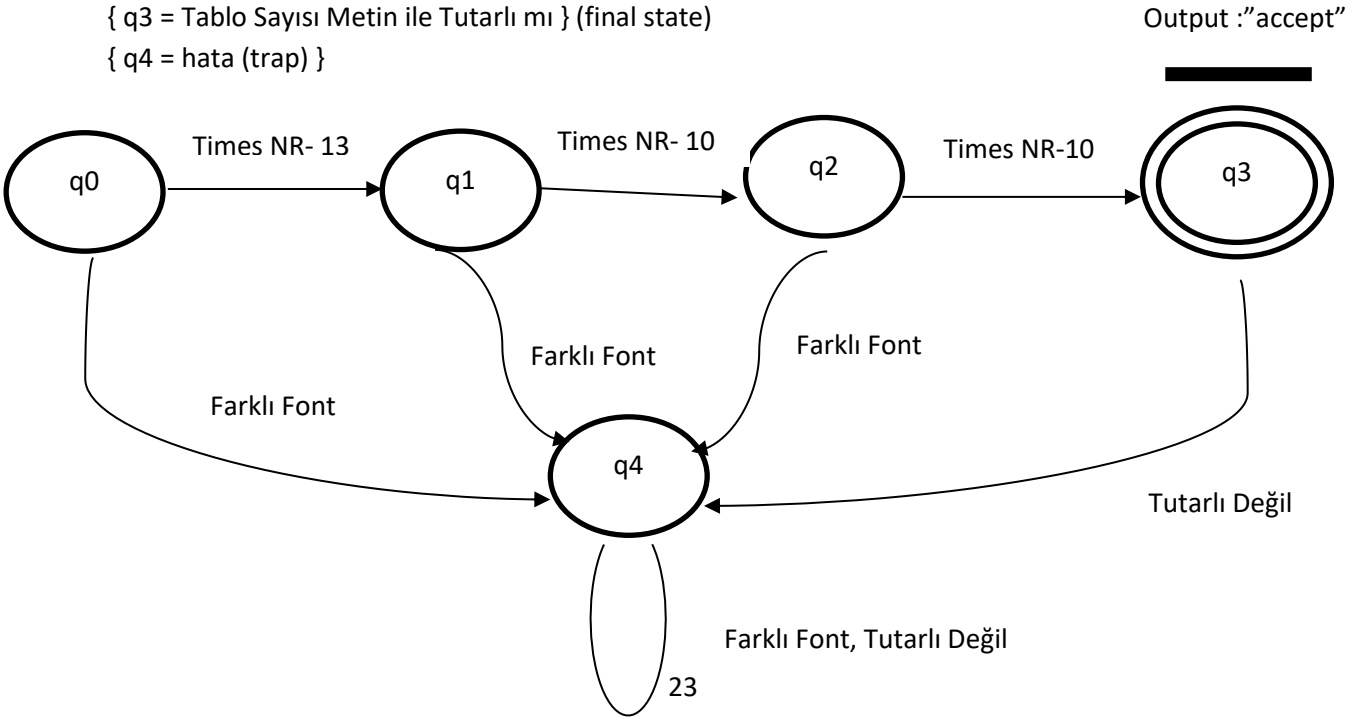
e- Q (Şekiller Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Şekil Sayısı Metin ile Tutarlı mı } (final state)
{ q4 = hata (trap) }



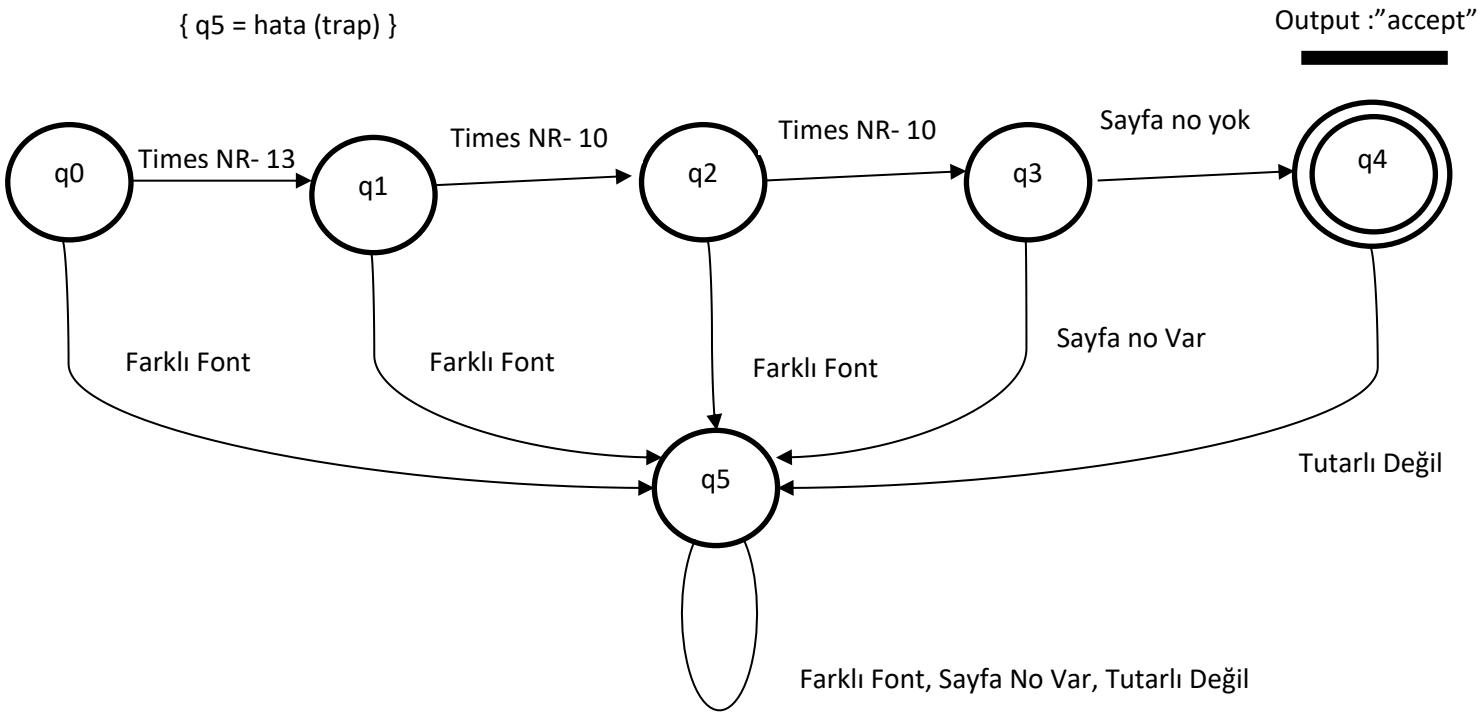
f- Q (Tablolar Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Tablo Sayısı Metin ile Tutarlı mı } (final state)
{ q4 = hata (trap) }



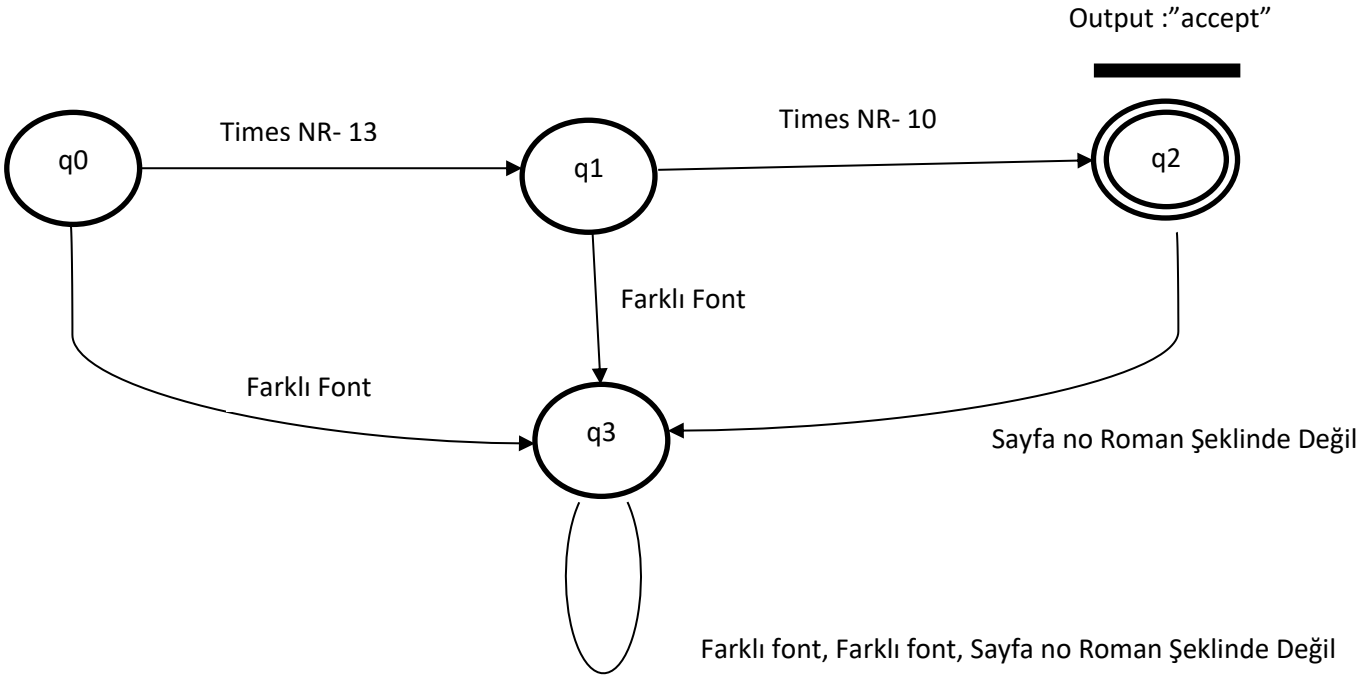
g- Q (Ekler Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Sayfa Numarasının Eklenmesi }
{ q4 = Ek Sayısı Verilen Eklerle Tutarlı Mı } (final state)
{ q5 = hata (trap) }



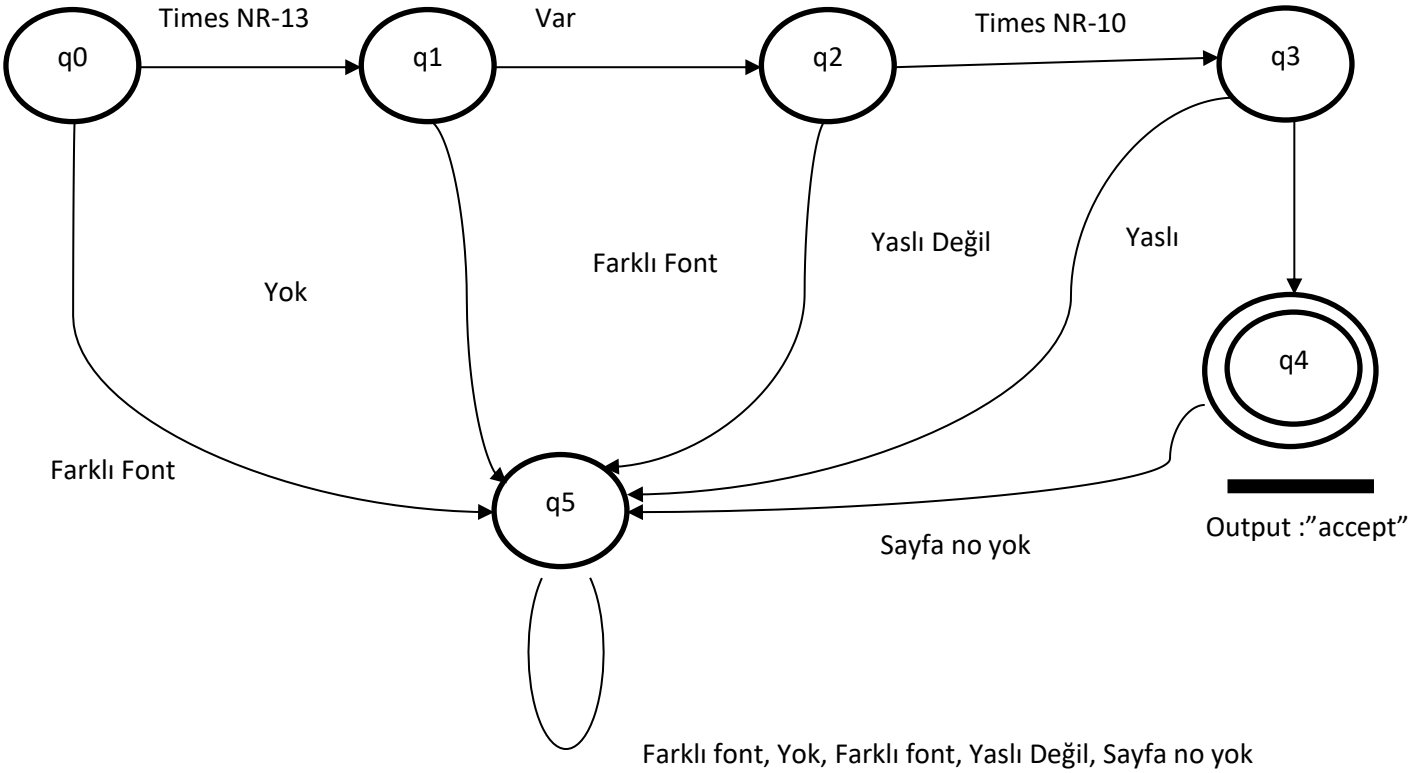
h- Q (Simgeler ve Kısaltmalar Listesi)

{ q0 = Ana Başlık Fontu } (initial state)
{ q1 = Ara Başlık Fontu }
{ q2 = Sayfa Numarası Roman Şeklinde mi }
{ q3 = hata (trap) } (final state)



i- Q (Giriş Metni)

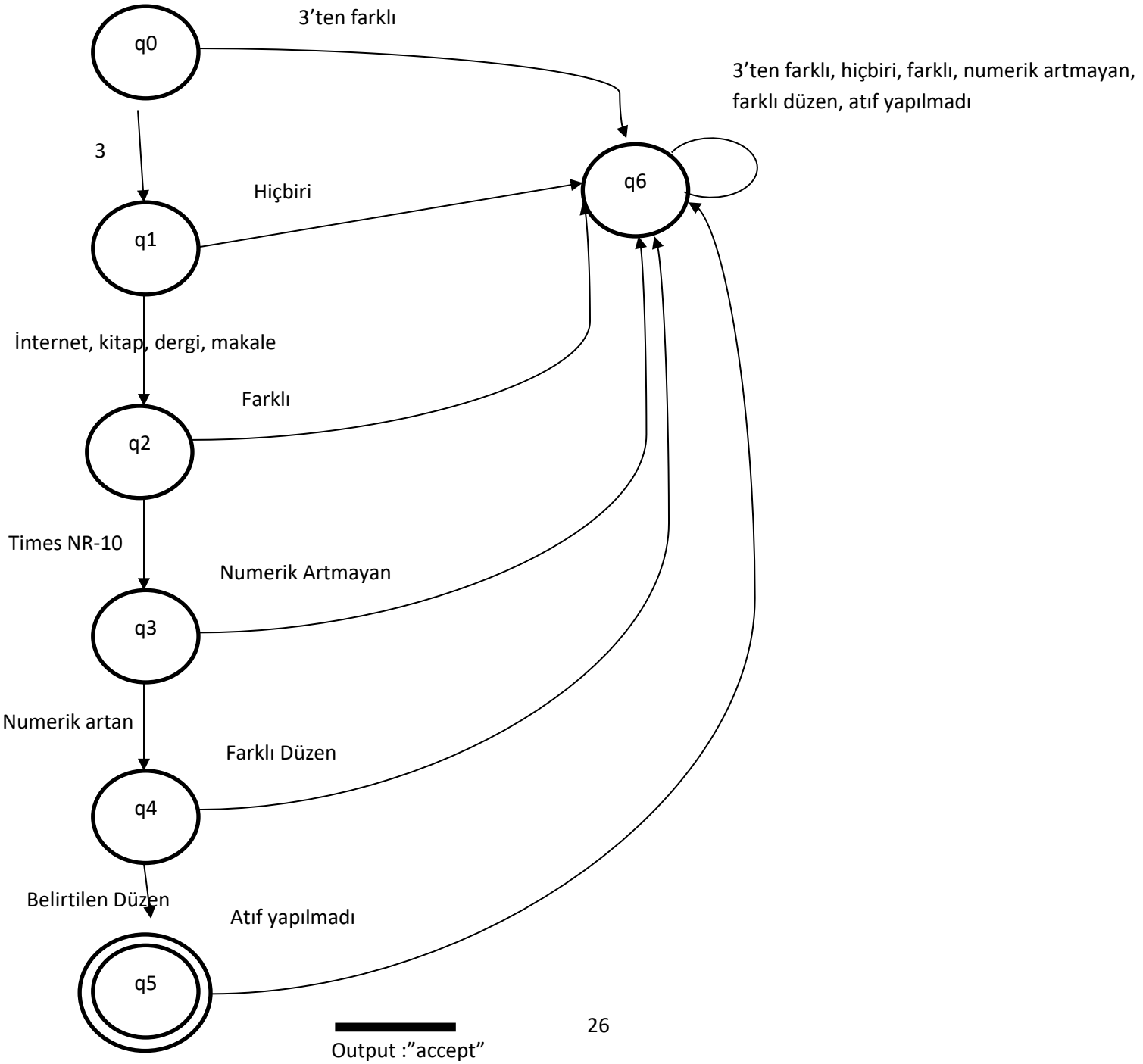
{ q0 = Ana Başlık Fontu } (initial state)
{ q1 = Zorunlu Başlıklar }
{ q2 = Zorunlu Başlıklar Fontu }
{ q3 = İki Yana Yaslı }
{ q4 = Sayfa Numarası } (final state)
{ q5 = hata (trap) }



j- Q (Kaynaklar)

//Bu kümede kaynakların en fazla üç adet olmasını istemiştik. Bu sebepten dolayı üçten fazla kaynak belirtilmişse hata alınması gerekmektedir.

{ q0 = kaynak sayısı} (initial state)
{ q1 = kaynak türü}
{ q2 = yazı fontu }
{ q3 = kaynak sırası }
{ q4 = isim- soyisim- tarih- eser adı- yer }
{ q5 = atıf yapıldı mı } (final state)
{ q6 = hata (trap) }



Output :"accept"

3.1.3. Veri Modeli

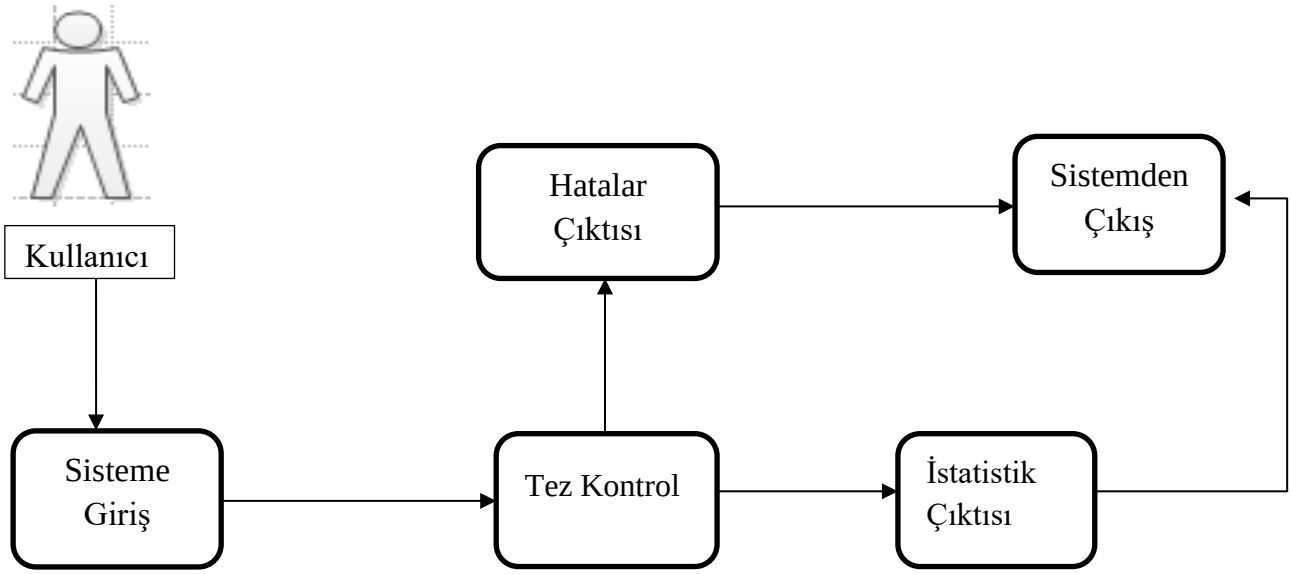
Şekil 3.1 Veri Modeli

3.2. Gereksenen Sistemin Mantıksal Modeli

3.2.1. Giriş

Mevcut olan sistemin geliştirmekte olduğumuz sistemle farklılıklar gösterdiğine daha önceden değinmiştik. Mevcut sistemin yeterli olmadığı ve tez aşamasında metinlerin fiziksel olarak da kontrol edilmesinin bir süreç olduğu kaçınılmazdır. Şimdi bu sistemin işlevsel modelini inceleyelim.

3.2.2. İşlevsel Model



Şekil 3.2 Basit İşlevsel Model

Durum ismi	tez dosyası ile sisteme giriş
Aktör Başlatılıyor	Kullanıcı
Aktör Katılımı	Kullanıcı
Giriş Durumu	Word dosyası ile sisteme giriş yapıldı
Olayların Çıkışı	Tez verilerine göre sistem yönlendirmesi yapıldı
Durum Çıkışı	Hatalar ve İstatiksel Veriler çıktı olarak verildi

3.2.3 Genel Bakış

Genel anlamıyla Şekil 3.2' de de görüldüğü gibi sistemin kolay bir işlevsel modeli bulunmaktadır. Sistem içerisinde tek kullanıcı bulunmaktadır. Sistemin girdi olarak bir Word dosyasına ihtiyacı vardır. Sisteme tanımlı kurallar çerçevesinde kontrol sağlanacak ve sistemdeki kullanıcıya sistem içerisine girdi olarak verilen Word dosyası hakkında bilgiler verilecektir.

3.2.4 Bilgi Sistemleri/Nesneler

Kullanıcı: Sistem içerisindeki tüm yetkilere sahip olan kişidir. Sistemin girdisinin sağlanması, sistemin çıktısı olan hatalara doğrudan erişen kişidir.

Girdi (Tez): Sistemin en önemli unsurudur ve çalışabilmesi için gereklidir. Word türünde olmalıdır.

Çıktı: Rapor niteliğindedir. Sistemin gerekli ve olması gereken özelliklerinden biridir.

3.2.6. Veri Sözlüğü

FA'ya göre $M = (Q, \Sigma, \delta, q_0, F)$ ile tanımlanır.

Sistemdeki Durumların Kümesi (Q);

- Tez Kapak Sayfası
- Önsöz Metni
- İçindekiler Tablosu
- Özet Metni
- Şekiller Listesi
- Tablolar Listesi
- Ekler Listesi
- Simgeler ve Kısaltmalar Listesi
- Giriş
- Kaynaklar

Sistemdeki Sembollerin Kümesi (Σ);

Sistemde girdimiz olan Word veya PDF dosyasını incelerken anlam bilgisine şu aşamada bakmadığımız için sembollerimiz sadece dokümanın fiziksel bakımdan belirlenen yazım kurallarından oluşan sembollerden oluşacaktır.

Q (Tez Kapak Sayfası)

{ q0 = üniversite logosu } (initial state)
{ q1 = ana başlık fontu }
{ q2 = tarih ve yer ismi } (final state)
{ q4 = hata (trap) }

Q (Önsöz Metni)

{ q0 = iki yana yaslı } (initial state)
{ q1 = yazı fontu }
{ q2= satır boşluğu }
{ q3 = teşekkür ifadesi }
{ q4 = yer-yıl } (final state)
{ q5 = hata (trap) }

Q (İçindekiler Tablosu)

{ q0 = ön bölümler kısmında roman rakamları } (initial state)
{ q1 = zorunlu başlıklar }
{ q2 = başlıklar metin içerisindeki başlıklara uygun }
{ q3 = sayfa numarası verilmesi }
{ q4 = sayfa numarası metin içerisinde tutarlı }
{ q5 = başlıklarda liste numarası }
{ q6 = özgeçmiş varsa sayfa numarası }
{ q7 = ekler bölümü varsa ekler listesi }
{ q8 = ana bölüm başlık fontu } (final state)
{ q9 = hata (trap) }

Q (Özet Metni)

{ q0 = kelime sayısı } (initial state)
{ q1 = sayfa numarası } (final state)
{ q2 = hata (trap) }

Q (Şekiller Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Şekil Sayısı Metin ile Tutarlı mı } (final state)
{ q4 = hata (trap) }

Q (Tablolar Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Tablo Sayısı Metin ile Tutarlı mı } (final state)
{ q4 = hata (trap) }

Q (Ekler Listesi)

{ q0 = Başlık Fontu } (initial state)
{ q1 = Alt Başlık Fontu }
{ q2 = Yazı Fontu }
{ q3 = Sayfa Numarasının Eklenmesi }
{ q4 = Ek Sayısı Verilen Eklerle Tutarlı Mı } (final state)
{ q5 = hata (trap) }

Q (Simgeler ve Kısaltmalar Listesi)

{ q0 = Ana Başlık Fontu } (initial state)
{ q1 = Ara Başlık Fontu }
{ q2 = Sayfa Numarası Roman Şeklinde mi }
{ q3 = hata (trap) } (final state

Q (Giriş Metni)

{ q0 = Ana Başlık Fontu } (initial state)
{ q1 = Zorunlu Başlıklar }
{ q2 = Zorunlu Başlıklar Fontu }
{ q3 = İki Yana Yaslı }
{ q4 = Sayfa Numarası } (final state)
{ q5 = hata (trap) }

Q (Kaynaklar)

//Bu kümede kaynakların en fazla üç adet olmasını istemiştik. Bu sebepten dolayı üçten fazla kaynak belirtilmişse hata alınması gerekmektedir.

{ q0 = kaynak sayısı} (initial state)
{ q1 = kaynak türü}
{ q2 = yazı fontu }
{ q3 = kaynak sırası }
{ q4 = isim- soyisim- tarih- eser adı- yer }
{ q5 = atıf yapıldı mı } (final state)
{ q6 = hata (trap) }

3.2.7 İşlevlerin Sıradüzeni

Kullanıcı sisteme erişmek için herhangi bir form doldurması kayıt işlemleri gerekmektedir. Sistemi çalıştırdıktan sonra yapması gereken form üzerindeki ilgili alanlara taratmak istediği tez dosyasının bilgisayar içerisindeki yolunu yazmaktır. Dosyanın boyutuna göre tarama işlemi devam edecek ve aynı form içerisinde kurallara uymayan kısım ve bulunduğu konum ekrana yazdırılacaktır.

3.2.8 Başarım Gerekleri

Mevcut sistemin incelenmesi sonucu; sistemin eksiklerinden yola çıkılarak, sistemin başarımı için;

- Sorgu işleminde bekleme süresinin en aza indirilmesi
- Kullanım kolaylığı

- Güvenilir
- Anlaşılabilir
- Sistemin sonuç doğrulukları

Temel gereklilik olarak tespit edilmiştir.

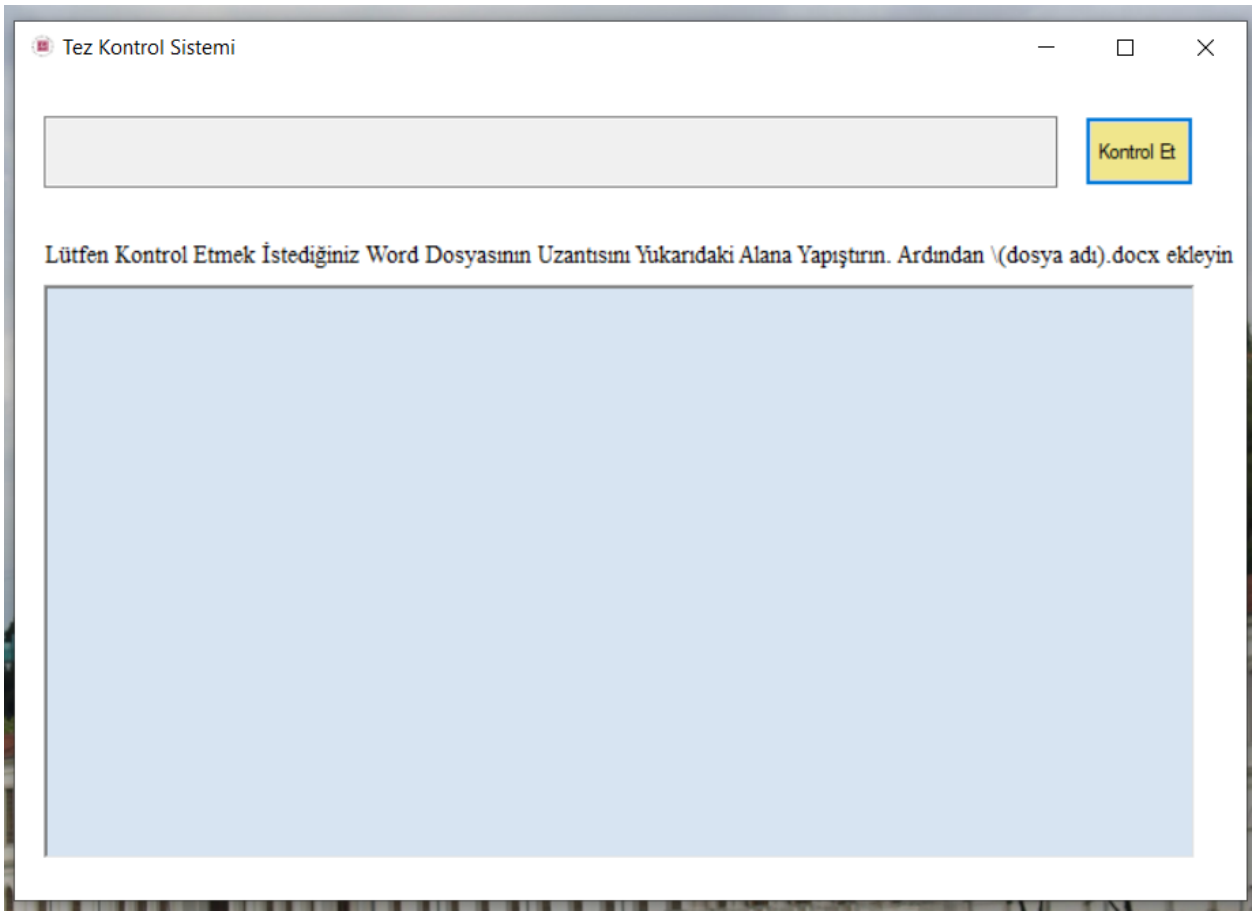
3.3. Arayüz (Modül) Gerekleri

3.3.1. Yazılım Arayüzü

Sistemin herhangi bir açıklığı ve sorunu bulunmaması için yazılım kısmına ayrıca özen gösterilmiştir. Gerekli olan veriler ileriki kısımlarda verilecektir.

3.3.2. Kullanıcı Arayüzü

Proje tasarlanırken tamamen kullanıcının kolaylıkla kullanabileceği bir sistem düşünülmüştür. Kullanıcı Arayüzü kısmında, renk uyumu, renk kolaylığı, kolay okunabilecek yazı stilleri, büyük yazı puntoları, zenginleştirilmiş grafiksel arayüzler kullanılarak, kullanıcının çokça düşünüldüğü bir arayüz tasarlanması hedeflenmektedir.



Şekil 3.3 Kullanıcı Arayüzü

3.3.4. Yönetimsel Arayüz

Sistemimiz tek arayüzden oluşmaktadır. Sistemi çalıştıran tüm kullanıcılar yönetici sıfatı taşır ve sistemin bu arayüzü yönetimsel arayüz olarak da anlam kazanabilir. İlerleyen safhalarda sistem geliştirilmeye açık olduğunda farklı arayüzler eklenebilir.

3.5 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 eğitim belgemiz bulunmaktadır.

3.4.3 Kullanıcı El Kitapları

Bu kısma ise projemizin en sonunda değineceğiz.

4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri

4.1.1 Genel Sistem Tanımı

Şekil 4.1 Genel Sistem Tanımı

✓ **Gereksinimler**

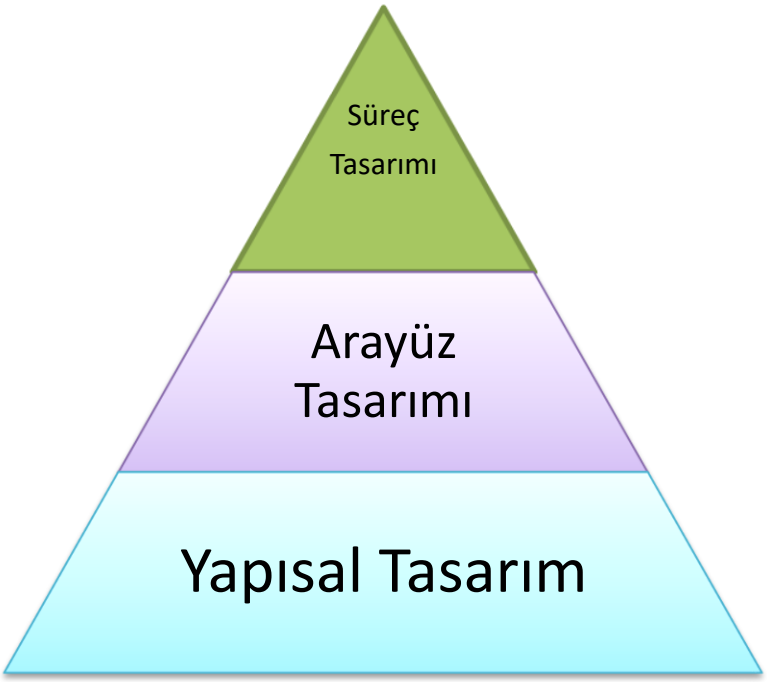
Belirtilen sistemin gereksinim raporunun hazırlanması için belirtilen şehirde- ki bu şehir Elazığ olarak belirlenmiştir- ve belirtilen üniversitede- Fırat Üniversitesi- yapılması gerekmektedir. Sisteme kullanıcı olarak tezi kontrol etmek ve onay vermekle sorumlu yöneticiler ile aynı zamanda tezleri hazırlayan öğrenciler olmak üzere iki farklı kullanıcı grubuna anket yapılması doğru bulunmaktadır.

✓ **İşlevsel Belirtimler**

Bu kısımda sistemin neler yapacağına değinmekte fayda var. Yapılacak olan sistemimiz, günümüzde ucu bucağı olmayan bilişim ve internet sektöründe gelişmiş yapıların hızından ve minimum hata payından faydalanmak üzere kurulu bir sistemdir. İnsanların daha kullanışlı ve zaman tasarrufu gerektiren sistemlerde daha kolay hareket edecekleri göz önüne alınmıştır. Bu sebeple sistemimiz tamamen kullanıcı ihtiyaç ve davranışlarına göre şekillenip, isteklere daha hızlı cevap verecektir.

✓ **Tasarım**

Tasarım aşamasında bir pramit model süreci takip edilecektir. Bütün katmanlar birbiriyle ilişkili olmakla beraber birbirinin tamamlamaktadır. Birbirini takip eden bu süreçler tasarım kısmını oluşturacaktır.



Yandaki Piramit modelde görüldüğü gibi sitemim tasarım aşaması birbirinin takip eden adımlar ve basamaklardan oluşmaktadır. Tüm basamakların tasarımının gerçekleştirilmesi ortaya Sistem Tasarımı kavramının tamamlandığını gösterecektir. Bu kısım büyük derecede önem arz etmektedir.

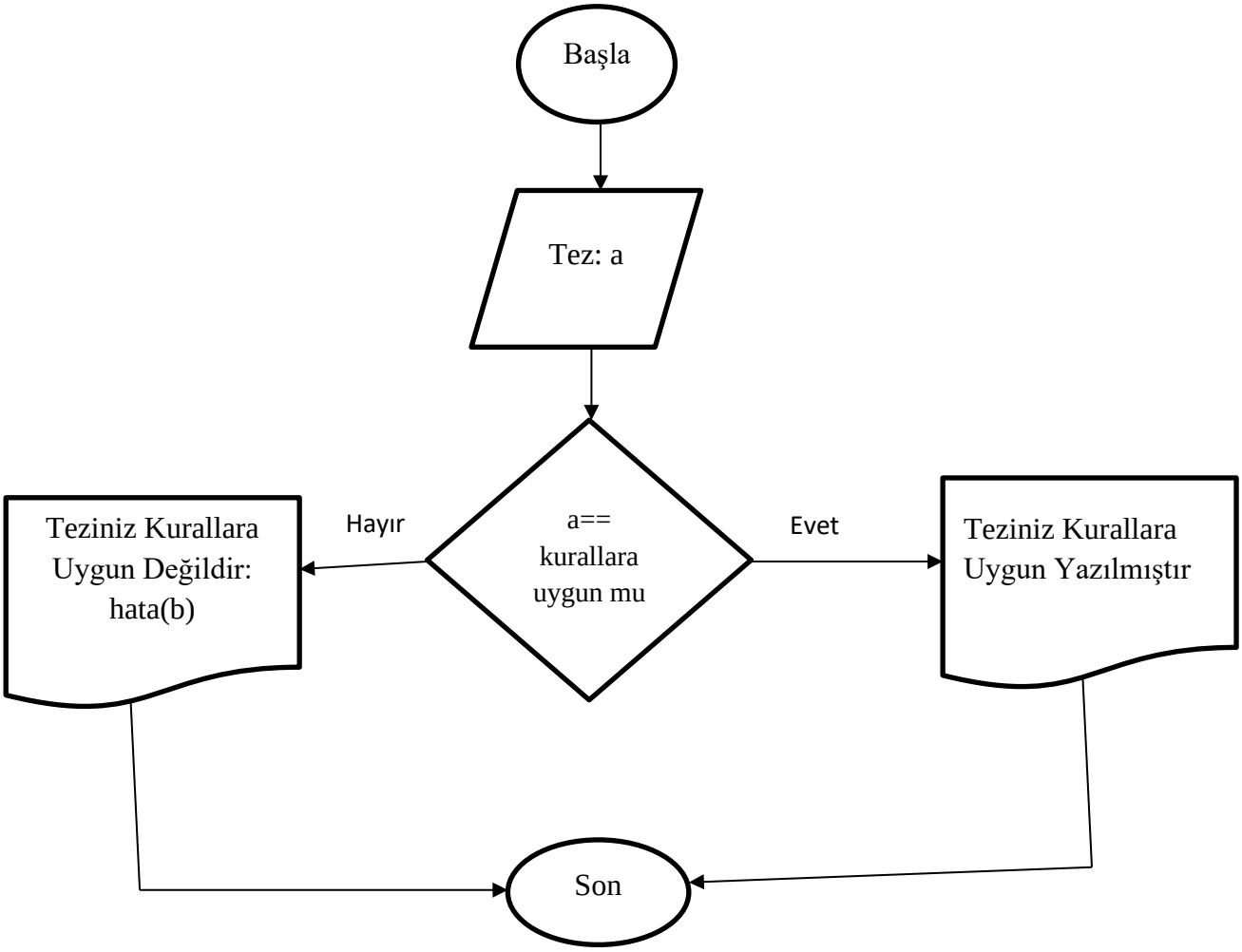
Şekil 4.2 Tasarım Aşamaları

4.1.2 Varsayımlar ve Kısıtlamalar

Sistemde varsayılan değerler bulunmamaktadır. Bunun yanı sıra kısıtlamalar belirli verilerin yerine getirilmemesi durumunda uygulanmaktadır. Var olan sistemi kullanabilmek için gerekli sistem eğitimini almak gerekli temel özelliklerden biridir.

4.1.3 Sistem Mimarisi

Hayatımızın her aşamasında kullandığımız, bizi çözümlere ulaştıran kavramlardan biri de algoritmalardır. Algoritmalar hayatımızın her aşamasında bizlere yardımcı olmaktadır. Sistemimizde ise sistemin mimarisi olarak verilecek olan diyagram, algoritma mantığını baz alarak akış şemaları olarak verilecektir. Bunun nedeni sistemin işleyişinin mantığını bir mühendislik disiplini içerisinde, her sorunu belirli aşamalardan geçirerek kesin sonuçlara ulaştıran, iyi tanımlı, sistem fikrini belirleyip, anlayabilmektir.



Şekil 4.3 Genel Sistem Mimarisi

4.1.4 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.

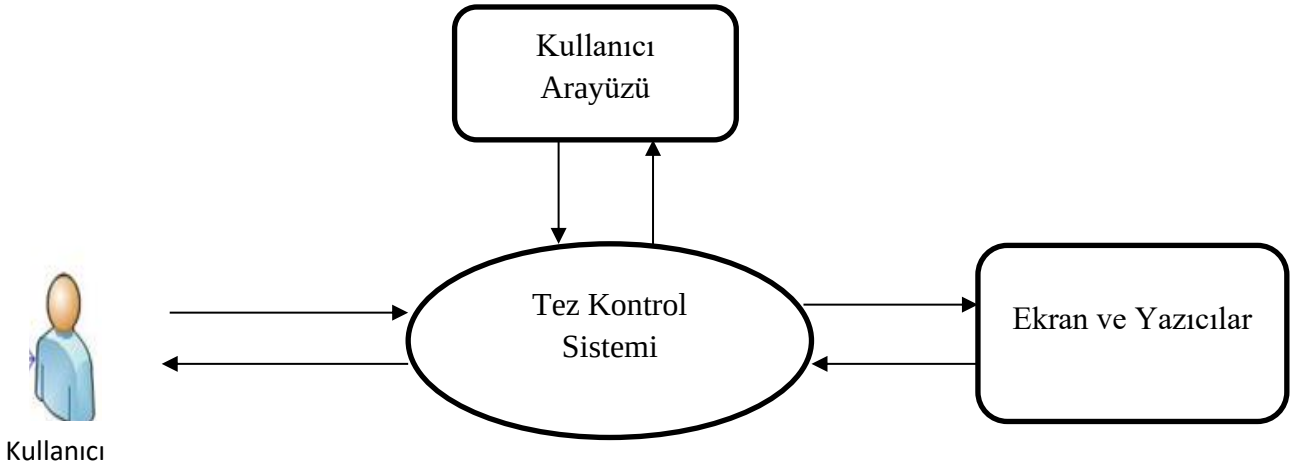
4.1.5 Performans

Genel olarak sistemin performans kriterini, sistem içerisinde yapılacak olan tüm testlerden yola çıkarak belirleyeceğiz. Bunun yanı sıra kullanıcıların sistemi test etmelerinden sonraki aşamada bizlere bildirecekleri geri dönüş ile sistem performansında artırma planlanmakta.

4.2. Süreç Tasarımı

4.2.1. Genel Tasarım

Genel olarak sistemimiz tek arayüzden oluştuğu için giriş modülü ve kullanıcı modülü aynı arayüzdür. Bu sebeple buna bağlı arayüzün işleyişi aşağıda verilmiştir.



Şekil 4.4. Kullanıcı Sayfası Modülü

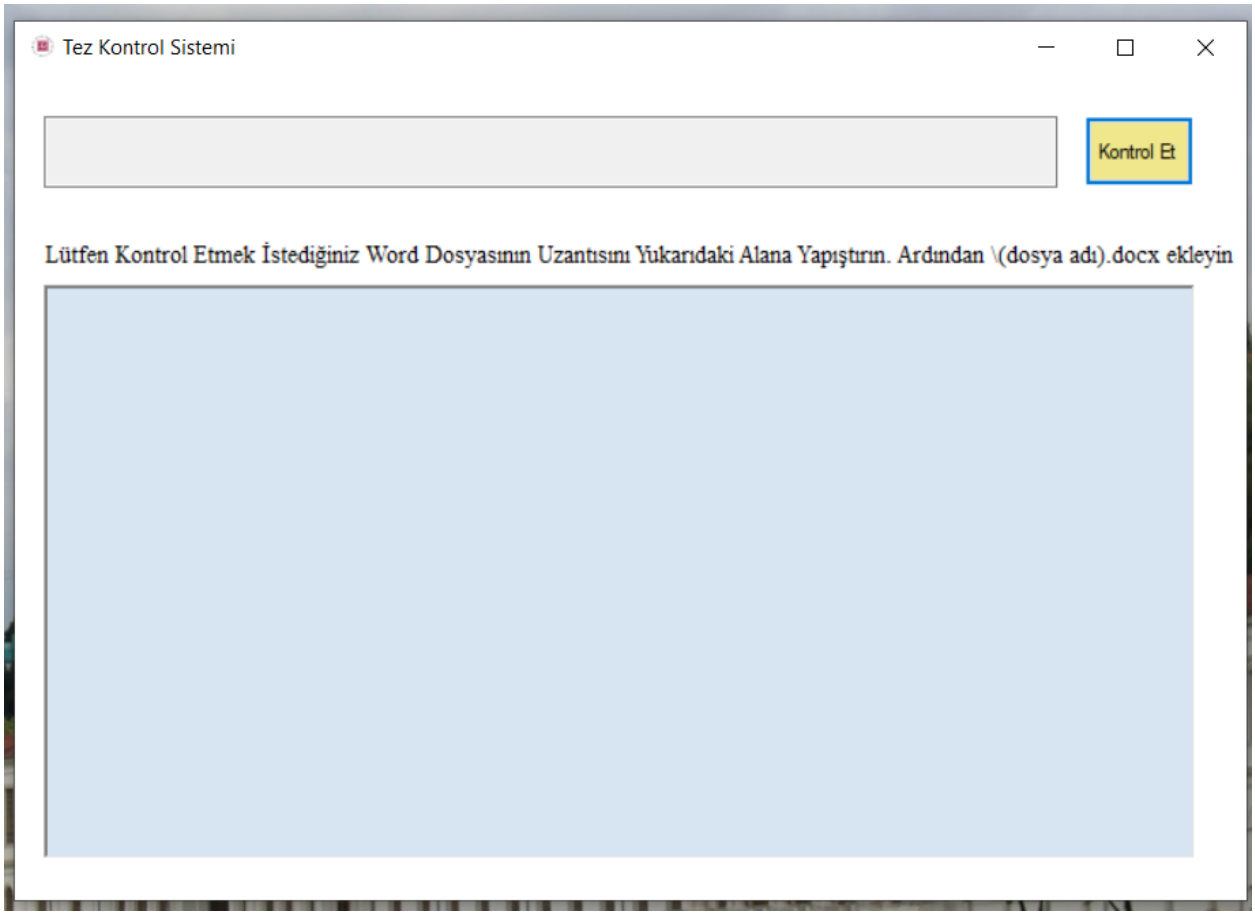
4.2.2. Modüller

4.2.2.1. Giriş Modülü

4.2.2.1.1. İşlev

Kullanıcının sisteme girişi ve sistemin çalışması için gerekli girdiyi vereceği bir modüldür.

4.2.2.1.2. Kullanıcı Arabirimi

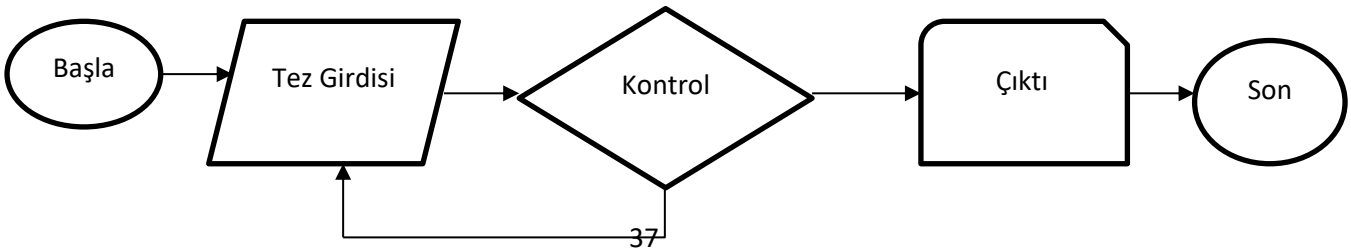


Şekil 4.5. Kullanıcı Arabirimi

4.2.2.1.3. Modül Tanımı

Bu modül sistemin ana unsurudur. Kullanıcı sistem üzerindeki tüm etkinliklerini bu modülde gerçekleştirecektir.

4.2.2.1.4. Modül İç Tasarımı



Metot Adı	Tanım
Giriş()	Sisteme iletilecek tez konum yolu
KontrolEt()	Tezin kurallar kapsamında kontrolü
Çıkış()	Sistemden çıkış

Şekil 4.6. Metotlar ve Tanımları

4.2.2.2. Yönetici Modülü

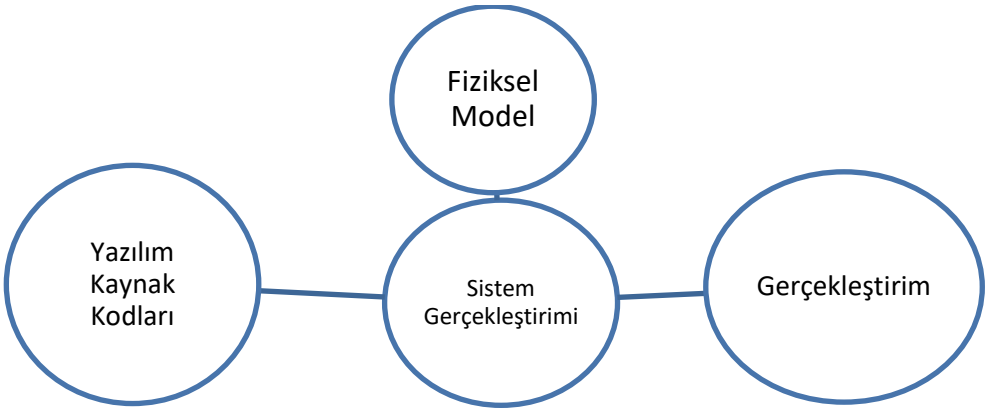
Daha önceden de belirtildiği üzere şu aşamada sistem içerisinde tek arayüz kullanılmaktadır. Bu sebeple yönetici olarak ayrıca bir arayüz modülü tasarlanmamıştır. İlerleyen süreçte sistemin geliştirilmeye açık olduğunu belirtmiştik. İlerleyen aşamalarda yönetici arayüzü tasarlayabilmek mümkündür.

4.3. Ortak Alt Sistemlerin Tasarımı

Sistemimize ait alt sistemler bulunmamaktadır. Aynı zamanda sistemimiz arşivleme ve yedekleme sistemleri barındırmamaktadır. İlerleyen süreçlerde sistemde kontrol edilen tez veya kullanıcı bilgilerinin yedeklenmesi gerekebilecek düzeye gelebilir. Bu başlığa ilerleyen süreçlerde değinilecektir.

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1. Giriş



Gerçekleştirim çalışması, tasarım sonucu üretilen sürecin 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

Yazılım geliştirme ortamı, tasarım aşamasında yazılımın ortaya çıkartılacağı analizler sonucu gerekli olan;

- ✓ Programlama Dili (C# - Microsoft Visual Studio)
- ✓ Hazır Program Kitapçıkları

Araçları belirlendi ve gerekli olan yazılım geliştirme ortamı hazırlandı.

5.2.1 Programlama Dilleri

Hasta Ziyaretçi Takip Sistemi'nde daha öncede belirttiğimiz gibi güçlü ve dinamik platformu üzerine kurulu bir dil olan C# programlama dilini kullanmaktayız. C# bize bu sistem içerisinde kaliteli kullanıcı yüzleri oluşturmamızı sağlayacaktır

5.2.2 Veri Tabanı Yönetim Sistemi

Sistemimiz veritabanı çalışması içermemektedir.

5.3 Kodlama Stili

Kendimize has kodlama biçimi 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 yazacağımıza tek bir döngüyle bu sorunların üstesinden geldik.

5.4 Olağandışı Durum Çözümleme

Olağandışı durum tabiri, istenmeyen durumun yanı sıra beklenilmeyen istenmeyen durum olarak tanımlanır. Bu tür sistemler tüm durumların senaryosunu, kolaylıkla karşılık verilebilmesi için önceden belirlemiş olması gerekmektedir.

5.4.1 Olağandışı Durum Tanımları

Sistemin işleyişi genel olarak kod satırlarıyla gerçekleştirildiğinden dolayı oluşabilecek herhangi bir olağandışı duruma karşı try-catch blokları devreye girecektir. Sistemde bulunan dört arayüzde de sıkça kullanılmıştır.

5.5 Kod Gözden Geçirme

Bir gazete yazısı düşünelim. O yazının gazeteye basılabilmesi için önce gazete editörleri tarafından kontrol edilmesi gerekir. Çünkü hatasız bir ürün ortaya çıkartılmak istenmektedir. Sistemimizde de az ve düzenli kod yazma mantığı örnek model olarak benimsenmiştir. Daha çok göze de hitap eden bir sistem yazılımı oluşturuldu. Biz yazılımcılar kodlarımızı yazarken, ileride tekrar kontrol etmemiz gerektiğinin ve hatalara karşı duyarlı olunması gerektiğinin farkında olarak kod yazmaktayız. Bu benimsenen fikirle sistemde oluşabilecek hataların oranı %3-4’lere kadar düştüğü görülmektedir.

5.5.1 Gözden Geçirme Sürecinin Düzenlenmesi

Bu süreç düzenlenirken asıl hedef sistem içerisindeki kod hatalarını çözmek değildir. Sorunlar tespit edilir. Gruplarda test aşamasında bu sorunları çözecek küçük çaplı grup oluşturulur. Amaç sistemin hedeflenen doğrultu ve beklenen hedefteki amacına ulaşıp ulaşmadığını kontrol etmektir.

5.5.2 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.

5.5.2.1 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?

5.5.2.2 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?

5.5.2.3 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?

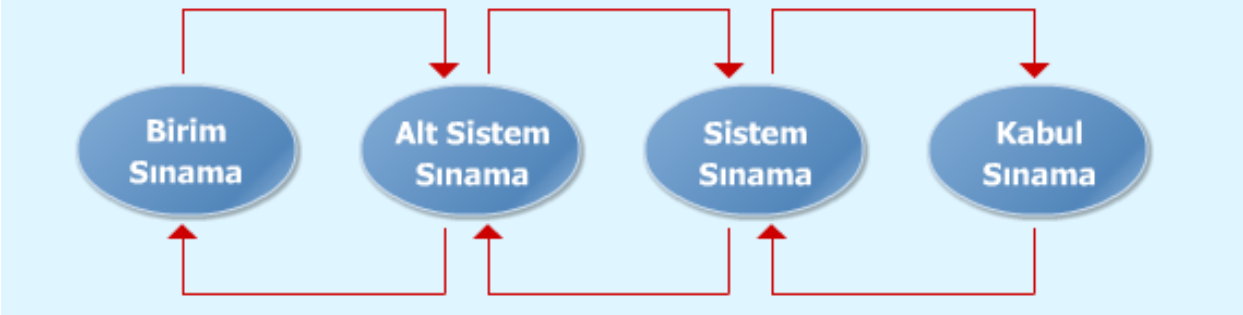
6.DOĞRULAMA VE GEÇERLEME

6.1 Giriş

Geliştirilen bu sistem için, doğrulama ve geçerleme aşamasında hedeflenen tutarlılık kavramına ulaşmaktır. Planlama aşamasından başlayıp sistemin teslimine kadar olan kısımda hedeflenen amaca ve istenilen projeye ulaşılmasını kontrolü doğrulama ve geçerleme safhasıdır. Burada gerekli belgeler, amaçlar, kapsamlar, test ve ekip durumu kontrol edilir. Projenin geliştirilirken var olan sistemin geliştirilme yöntemleri de incelenip karşılaştırma yapılır. Doğrulama ve geçerleme kavramları arasında farklılıklar vardır. Daha iyi anlaşılabilmesi için aşağıdaki şema verilmiştir.

Şekil 6.1 Doğrulama ve Geçerleme

6.2 Sınama Kavramları



Birim Sınama: Sistem içerisinde yer alan Kullanıcı Girişi-Tez Kontrol Girişi, Parolamı Unuttum Girişi- Tez Kontrol Girişi, Üye Ol Girişi- Tez Kontrol Girişi birimleri sınandı ve sonuçları çıkarıldı.

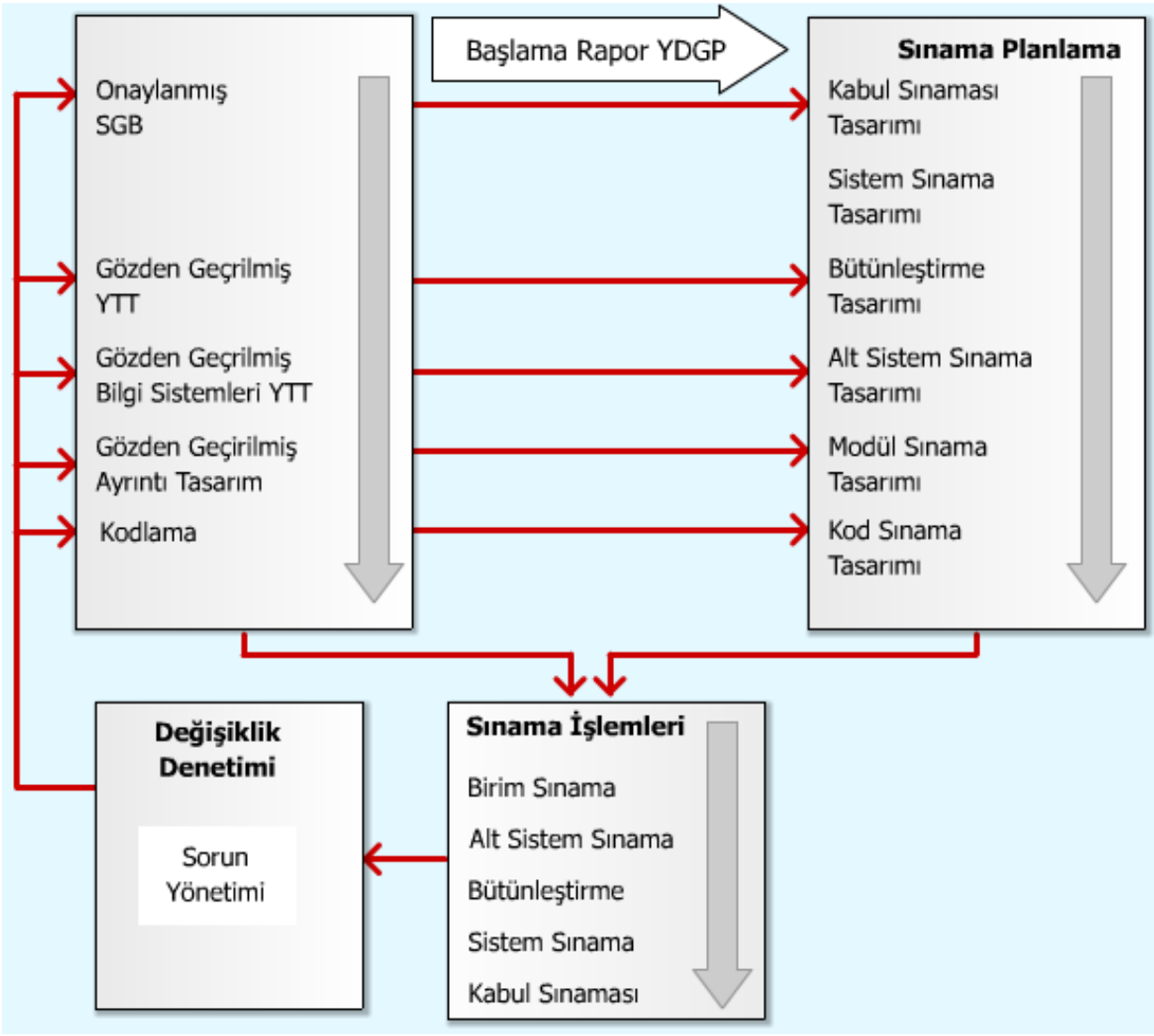
Alt Sistem Sınaması: Sistemde bulunan dört farklı arayüz modülü kendi arasında sınandı. Genel olarak arayüzlerdeki eksiklikler giderildi.

Sistem Sınaması: Sistemin arayüzleri ile tüm kullanıcı özellikleri birleştirilerek kendi arasında bir bütün olarak sınandı. Programın eksiksiz olduğu onaylandı.

Kabul Sınaması: Sistem prototipten çıkartılıp gerçek verilerle sınandı. Sorunsuz olduğu onaylandı.

6.3 Doğrulama ve Geçerleme Yaşam Döngüsü

Sınama işlemlerimiz, sınanma planı ve sınanmış işlemlerin birleşimi ile oluşturulmuş bir bütündür. Sistem için ortaya çıkabilecek değişimler ve sonradan eklenecek belirtimler için sorun yönetimi devreye giriyor. Burada sistem sınaması aşamalarında yapılacak tüm sınama modülleri öncesinde, sırasında ve sonrasında sistemin değişimlerine hızlı bir şekilde çözüm üretilecektir. Bu yaklaşım modeli bir yaşam döngüsü olarak nitelendirilebilir. Temel hedefimiz sistemin doğrulama ve geçerleme işlevlerinin doğru ve eksiksiz bir şekilde yerine getirildiğinin kontrolüdür. Aşağıda Şekil 6.2 de sistemin doğrulama ve geçerleme yaşam döngüsünün detaylı bir şekilde gösterildiği ve işlemlerin sıra düzeninin belirtildiği şeklimiz mevcuttur. Lütfen inceleyeniz.



Şekil 6.2 Yaşam Döngüsü

6.4 Sınama Yöntemleri

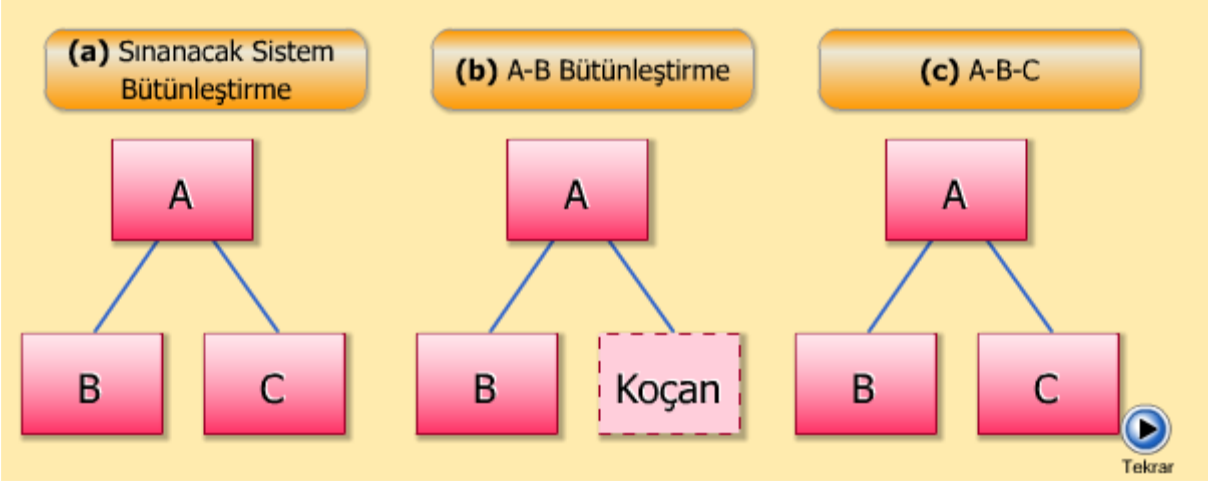
Terimsel olarak baktığımızda sınama kavramı geliştirdiğimiz sistem için düzeltme, tamamlama olarak görülmemelidir. Sistemimiz için sınama kavramı bir sonraki adımı tahmin etme, o adıma göre davranma olarak tanımlanmaktadır. Sistemimiz için Beyaz Kutu Sınaması ve Temel Yollar Sınaması yapılmıştır. Bu durumda elde edilmiş birkaç sonuç ortaya çıkmıştır. Yapılan sınamalar ile hem sistemin parça şeklinde sınanması hem de bütünsel sınanması değerlendirilmiştir.

6.5 Sınama ve Bütünleme Stratejileri

Genel olarak sınama kavramını düşündüğümüzde bir sistem içerisinde sistemin bir bütün olarak sınanması durumu ortaya çıkıyor. Fakat bazı durumlarda sistem için ayrı bölgelerde ayrı işlevlere sahip sınamalar yapılması zorunluluğu vardır. Bu duruma bakarsak Aşağıdan Yukarı ve Yukarıdan Aşağı Sınamaları gerekli görülmektedir.

6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme

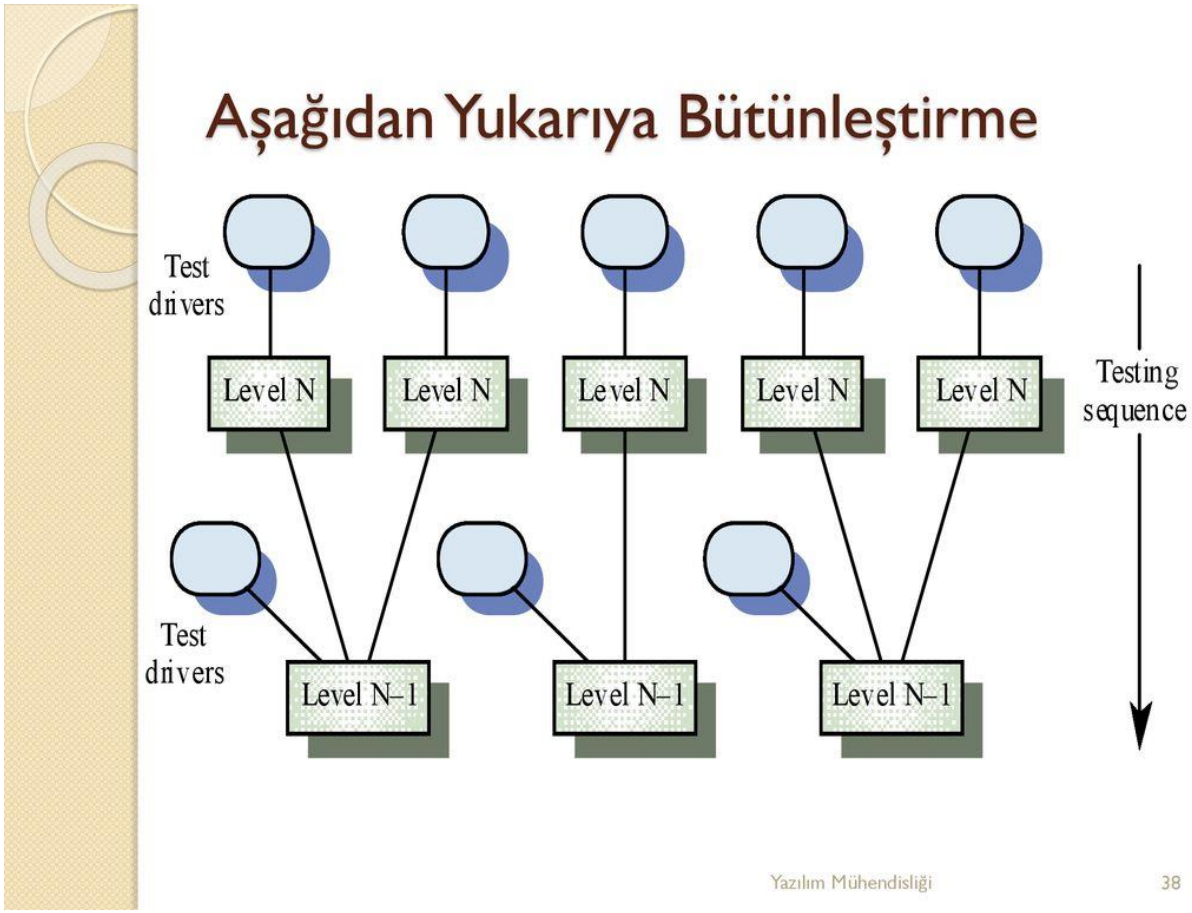
Verilen sınama metodunda, sistem içerisinde en üst seviyeden başlanılarak yani sistemin bütününden en küçük birimine doğru sınama yapılır. En alt birime ilerleyene kadar sistem içerisinde tüm metot, birim ara yüz sınamaları yapılmaktadır. Aşağıda verilen şemada daha açık bir şekilde görmektesiniz.



Şekil 6.6 Yukarından Aşağı Sınama

6.5.2 Aşağıdan Yukarı Sınama ve Bütünleştirme

Aşağıdan yukarı bütünleştirmede ise, önceki yöntemin tersine uygulama yapılır. Önce en alt düzeydeki işçi birimleri sınanır ve bir üstteki birimle sınama edilmesi gerektiğinde bu üst bileşen, bir '**sürücü**' ile temsil edilir. Yine amaç, çalışmasa bile arayüz oluşturacak ve alt bileşenin sınanmasını sağlayacak bir birim edinmektir.



Şekil 6.7 Aşağıdan Yukarı Sınama

6.6 Sınama Planlaması

Sınama Planlaması içerisinde giriş, test edilecek sistem, test edilecek ana fonksiyonlar, test edilmeyecek ana fonksiyonlar, geçti/kaldı kriterleri, test dokümanı gibi test aşamaları başlıkları altında rapor hazırlandı. Aşama aşama başlıklarda belirlendiği gibi gerçekleştirilerek sistemin sınama planlaması hazırlandı. Aşağıdaki tabloda sınama planlaması daha açık ifade edilmiştir

Giriş

Amaç
Tanım ve Kısaltmalar
Referanslar

Sınama Yönetimi

Sınama Konusu
Sınama Etkinlikleri ve Zamanlama

- Temel Sınama Etkinlikleri
- Destek Etkinlikler

Kaynaklar ve Sorumluluklar
Personel ve Eğitim Gereksemeleri
Sınama Yaklaşımı
Riskler ve Çözümler
Onaylar

Sınama Ayrıntıları

Sınanacak Sistemler
Girdiler ve Çıktılar
Sınamaya Başlanma Koşulları

- Girdilerin Hazır Olması
- Ortam Koşulları
- Kaynak Koşulları

6.7 Sınama Belirtimleri

Bu ayrıtılar temel olarak:

- sınanan program modülü ya da modüllerinin adları,
- sınama türü, stratejisi (beyaz kutu, temel yollar vb.),
- sınama verileri,
- sınama senaryoları

türündeki bilgileri içerir.

7. BAKIM

7.1 Giriş

Tez Kontrol Sistemi'nin tasarım ve gerçekleştirim aşaması bittikten sonra bakım aşamasına alınmıştır. Daha önce de sistemin güvenlik özelliğinin ön plana çıktığını belirtmiştik. Bu sebeple sistemin hata durumu ile karşılaşması mutlak bir suretle istenmemektedir. Bu sebeple bakım aşamasının tüm durum kriterleri değerlendirilerek yapılmaktadır. Bu aşamada IEEE 1219-1998 standardı baz alınmıştır.

7.2 Kurulum

Sistem kurulumu için kullanılacak bilgisayarda sistemin form uygulaması son sürümünün bulunması gereklidir. Yazılım ve donanım kaynaklarından ötürü sistemin bir mobil veya web sürümü henüz geliştirilmemiştir. Kodların herkese ulaşabilmesi için açık kaynak kod olarak ilgili platformlarda paylaşılmıştır. Gerek sistem içerinde gerek açık kaynak kod dosyalarında sistemin iç işleyişi ve kurulum ile ilgili bilgiler yer almaktadır.

7.3 Yerinde Destek Optimizasyonu

Bu aşamada sistem kurulumu yapıldıktan sonra sistemin işleyişi ile ilgili rehber ve denetmen sistem tasarımcısı olarak bana düşüyor. Eğer sistemin kurulumu tam anlamıyla istenirse, gerekli ihtiyaçlar karşılandıktan sonra sistem içerisinde ilgili bir yetkili gönderilebilir.

7.4 Yazılım Bakımı

Kullanıcılardan alınan geri dönüşler üzerine, iki farklı geri dönüş aşaması değerlendirilecektir. Bu aşamada geri dönüş olarak Değişim İsteği ve Hizmet İsteği olarak bildirilecektir. Oluşturulmuş hali hazırda bulunan destek ekibi geri dönüşü alacaktır. Yapılan sözleşmeler baz alınarak sistemin tüm üreticileri tarafından istişarede bulunacaktır. Bu istişare sonrasında izlenecek yollar belirlenecek ve alınan geri dönüş üzerine düzenleme sürecine girilecektir. Düzenleme süreci sonrasında kullanıcının verdiği geri dönüş çözüldükten sonra, yeni arabirim oluşturulacaktır. Oluşturulan yeni arabirim satıcıya teslim edilecektir. Satıcı teslim aldığı yeni ara yüzü kullanıcılara tekrardan sunacaktır

8.SONUÇ

Tez Kontrol Sistemini hayata geçirmiş bulunmaktayız. Belirttiğimiz ve planladığımız adımları bir bilgisayar bilimcisi olarak gerçekleştirdik. Sistematik problem çözme yaklaşımı ile günümüz gelişen teknolojisinin yararlarını kullanarak, hayatımızda ve eğitim alanında gerekli olan bir sistem geliştirmiş olduk. Bu sistem ile hem lisans eğitimi alan son sınıf öğrencileri hem de gerekli tezin kurallarını kontrol edecek olan akademisyen hocalarımızın zamandan kazanç ve iş gücünü azaltma etkinliklerini yerine getirmiş olduk. Yani bu sistem ile hem daha hızlı hem de daha kaliteli içeriklerin ortaya çıkartılması daha kolay bir hal almıştır. Bu sistemi geliştirirken desteklerini ve yardımları esirgemeyen hocalarıma ve arkadaşlarıma teşekkür ederim.

9. KAYNAKLAR

[1]. Bülent Çobanoğlu - 'Java İle Programlama ve Veri Yapıları'- Pusula Yayıncılık - Mayıs 2017.

[2]. Muhammet Baykara – 'Yazılım Sınama Teknikleri'- pdf- page 6.6- Fırat Üniversitesi- Teknoloji Fakültesi Yazılım Mühendisliği.

[3]. Erkan Tanyıldızı – 'Nesne Tabanlı Programlama'- pdf – Fırat Üniversitesi – Teknoloji Fakültesi Yazılım Mühendisliği.

[4]. Resul Daş – 'Yazılım Tasarım ve Mimarisi' – pdf – Fırat Üniversitesi – Teknoloji Fakültesi Yazılım Mühendisliği.

İnternet Kaynakları

1- http://web.firat.edu.tr/mbaykara/ytm4.pdf
2- https://blog.metu.edu.tr/e102792/2018/03/13/yazilim-dogrulama-ve-gecerleme-sureci-ve-test-asamalari/
3- http://www.kriptarium.com/algoritma.html#
4- http://www.kriptarium.com/pd.html#
5- https://www.youtube.com/channel/UCUkxvNiWsRdLIWeIaPW8hlA (Cem Baydoğan)
6- https://www.youtube.com/channel/UCtOX6QsKtHYMYh7uD1kaBcw
7- https://github.com/