

**FIRAT ÜNİVERSİTESİ**

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği Bölümü

**YMH319 - Programlama Dilleri Proje Uygulaması ve Dokümantasyonu**

**TEZ KONTROL SİSTEMİ**

185541075 Umut Kutlu

**Şubat – 2021**

|  |
| --- |
| **1. GİRİŞ** |

* 1. **Projenin Amacı**

Tez yazmak zahmet bir iş olduğu kadar şablonlara uygun ve hatasız bir şekilde yazılıp yazılmadığının kontrol edilmesi de zahmetli bir iştir. Bu projemizde yazılan bir tezin belli kurallara uyup uymadığını kontrol edilmesi ve tespit edilen hataların gösterilmesi amacıyla bir program geliştirilmesi amaçlanmaktadır.

* 1. **Projenin Kapsamı**

Programı tez yazan ya da kontrol eden herkes kullanılabilir. Örnek kapsama alanı:

- Kurumlar ve kuruluşlar,

- Öğretim üyeleri

- Öğrenciler,

Olarak sıralanabilir.



Şekil 1 tez yazmak

|  |
| --- |
| **2. PROJE PLANI** |

**2.1 Giriş**

Program tez kurallarının kontrol edilmesini, mevcut hataların gösterilmesini ve tezin belli şablonlara uyup uymadığını kontrol edilmesini sağlar. Giriş olarak programa bir Word dosyası verilir. Program sayfaları, şekilleri, paragrafları, içindeki her şeyi tarayarak belli şablonlara uyup uymadığını hızlı bir şekilde yapar ve kullanıcıya şayet bir hata varsa bunu gösterir.

**2.2 Projenin Plan Kapsamı**

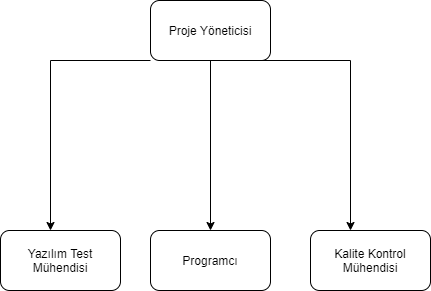
Bu yazılımın planı; yazılımın oluşturulması için gereken insan kaynaklarını, donanım ve yazılım kaynaklarını, yazılımın maliyet hesaplamalarını ve önerilen sistemin gerektirdiği fonksiyonları kapsamaktadır.

**2.3 Yazılım Zaman-İş Planı**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ZAMAN  İŞ | Proje Başlangıç  Tarihi:01/11/2020 | | |  |  |  |  |  | HAFTALAR | | |  |
| 1 | 2 | 3 | | 4 | 5 | | 6 | | 7 | 8 | |
| İhtiyaç /  Gereksinim  Geliştirme |  |  |  | |  |  | |  | |  |  | |
| Sistem  Çözümleme |  |  |  | |  |  | |  | |  |  | |
| Kullanıcı  Arayüz  Tasarımı |  |  |  | |  |  | |  | |  |  | |
| Gerçekleştirim |  |  |  | |  |  | |  | |  |  | |
| Sistem  Doğrulanması |  |  |  | |  |  | |  | |  |  | |

**Şekil 2.1 Yazılım Zaman-İş Planı**

**2.4 Yazılım Ekip Yapısı**



**Şekil 2.4.1. Ekip Yapısı Şeması**

**2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları**

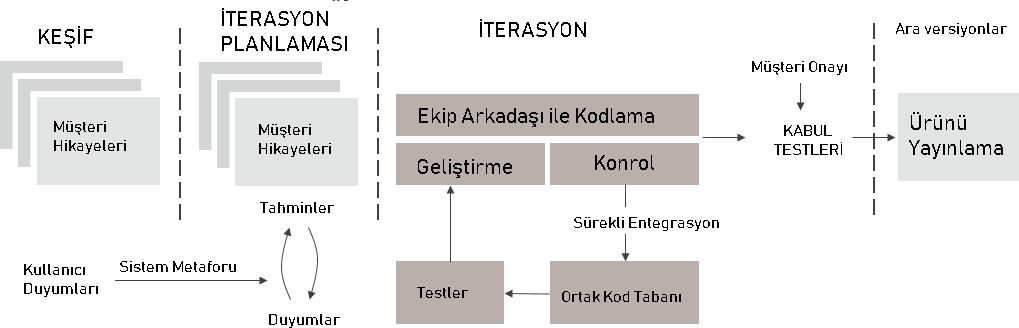
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ÇÖZÜMLEME TASARIM** | **PROGRAMLAMA** | **SINAMA** | **DESTEK** |
| **ORTAMLAR** | Windows 10 | Python 3.8 | Windows 10 |  |
| **ARAÇLAR** | Microsoft Word Draw.io | Visual Studio Code |  |  |

**2.7 Proje Standartları, Yöntem ve Metodolojiler**

Projenin 8 Hafta gibi bir sürede tamamlanması hedeflenmiştir. Gereksinimlerin çıkarılması ile birlikte sırasıyla sistemin çözümlenmesi, kullanıcı ara yüzünün tasarlanması, gerçekleştirim ve en son olarak sistemin doğrulanması hedeflenmektedir.

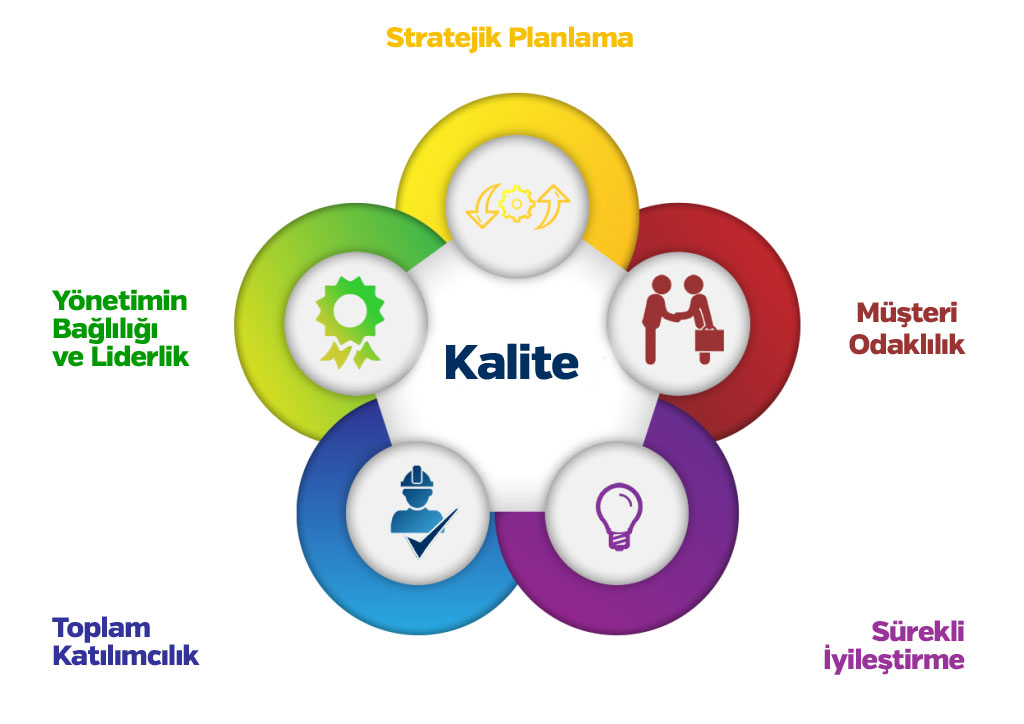
Bir Agile uygulaması olarak SCRUM, her ne kadar popülaritesi olan bir metodoloji olsa da sertifikalı bir SCRUM Master’ın varlığını gerektirdiğinden projemizde uygulanamayacaktır.

Projede Extreme programlama metodolojisi kullanıldı. Kullanılmasının sebebi basitlik, cesaret, geri dönüş…



Şekil 2.4.2 Extreme Programlama Metodolojisi Akışı

**2.8 Kalite Sağlama Planı**

****

**Şekil 2.4.3 Kalite Sağlama Planı**

1. **Tamlık:** Yazılımdaki fonksiyonların, butonların düzgün bir şekilde çalışmasıdır.
2. **Yeniden Kullanılabilirlik:** Yazılımın gerektiğinde tekrar düzenlenip kullanılabilmesidir.
3. **Güvenilirlik:** Yazılımdaki gerekli güvenlik önlemlerinin alınmasıdır.
4. **Modülerlik:** Yazılımın küçük parçalardan oluşması ve her bir parçanın bütün yapıyı bozmadan değiştirilebilmesinin sağlanmasıdır.
5. **Genellik:** Yazılımın herkes tarafından kullanılabilmesidir.
6. **Taşınabilirlik:** Yazılımın internete bağlı masaüstü cihazlarda taşınmasıdır.
7. **Kullanılabilirlik:** Yazılımın her insan tarafından kullanılabilmesidir.

**2.9 Konfigürasyon Yönetim Planı**

Programın ilerde ek özellikler katılabilmesi ve yeni isteklerin eklenerek güncellenmesi için konfigürasyon planı hazırlanmıştır.

Konfigürasyon planının uygulanmasını gerektiren bazı durumlar aşağıdaki gibidir:

* yeni hataların kontrol edilmesi için ek özellikler eklenmek istenmesi,
* Kullanılan yazılım veya donanımın güncelliğini yitirmesi,
* Sistemde herhangi bir istenmeyen durum hali

**2.10 Kaynak Yönetim Planı**

Sistemde donanımsal olarak sadece bir bilgisayar kullanılması yeterlidir bu kullanıcı tarafından karşılanacaktır. Ek olarak kullanılan program olarak Visual Studio Code ücretsiz bir şekilde kullanılmaktadır.

**2.11 Eğitim Planı**

Sistem eğitimi için doküman, kullanım kılavuzu, videolar vb. şeyler kullanılarak sistemi kullanacak kişilerin eğitim verilmesi hedeflenmektedir.

**2.12 Test Planı**

Scrum ekiplerindeki test mühendislerinin görevleri şunlardır:

* Projede oluşturulan modüllerin proje gereksinimlerini karşılamasını sağlamak,
* Sistemdeki hız sorunlarının çözümü,
* Sistemin optimizasyonu ve buna bağlı olarak performans artışını sağlamak.

**2.13 Bakım Planı**

Kullanıcılardan gelen geri dönüşlere göre 6 ayda bir sistem bakıma alınarak ek özellikler eklenmesi ve mevcut hataların giderilmesi hedeflenmektedir.

|  |
| --- |
| **3. SİSTEM ÇÖZÜMLEME** |

**3.1 Mevcut Sistem İncelemesi**

Yapılan araştırmalar neticesinde daha önceden mevcut bir sisteme rastlanmamıştır.

**3.1.1 İşlevsel Model**

****

**Şekil 3.1.1 Sistemin işleyişi**

Sistem gereksinimleri için müşterilerden gelen örnek senaryolar şunlardır:

1- Ben bir kullanıcı olarak, sayfa kenar boşluklarının tez kurallarına uygun bir şekilde yazılıp yazılmadığını kontrol etmek istiyorum.

2- Ben bir kullanıcı olarak, tez içerisinde paragrafların paragraf başı yapılarak ve büyük harf ile başlanıp başlanmadığını kontrol etmek istiyorum

3- Ben bir kullanıcı olarak, tez içerisinde tez başlığının belirlenen kurallara göre yazılıp yazılmadığını kontrol etmek istiyorum

4- Ben bir kullanıcı olarak, tez içerisinde isimlerin(Ad SOYAD) formatında yazılıp yazılmadığını kontrol etmek istiyorum

5- Ben bir kullanıcı olarak, tez içerisinde tarih bilgilerinin doğru bir formata yazılıp yazılmadığını kontrol etmek istiyorum

6- Ben bir kullanıcı olarak, tez içerisinde yazım düzenini kontrol etmek istiyorum

7- Ben bir kullanıcı olarak, tez içerisinde sayfaların sırayla numaralandırılıp numaralandırılmadığını kontrol etmek istiyorum

8- Ben bir kullanıcı olarak, tez içerisinde başlıkların ve alt başlıkların uygun formatta yazılıp yazılmadığını kontrol etmek istiyorum

9- Ben bir kullanıcı olarak, tez içerisinde içindekiler kısmında verilen sayfa numaralarının tez sayfaları ile karşılaştırarak var olup olmadığını kontrol etmek istiyorum.

10- Ben bir kullanıcı olarak, tez içerisinde satır aralıkları ve girintileri kontrol etmek istiyorum.

11- Ben bir kullanıcı olarak, tez içerisinde listeler kısmında kısaltmaların alfabetik sırayla yazılıp yazılmadığını kontrol etmek istiyorum.

12- Ben bir kullanıcı olarak, tez içerisinde numaraya dayalı atıf kullanılmış ise; kaynaklar metin içinde veriliş sırasına uygun sıralanıp sıralanmadığını kontrol etmek istiyorum.

13- Ben bir kullanıcı olarak, tez içerisinde tez sayfalarında kazıntı, silinti, karalama veya bir şekilde kapatılmış bölge bulunup bulunmadığını kontrol etmek istiyorum.

14- Ben bir kullanıcı olarak, tez içerisinde farklı ortamlarda alınan şekillere referans verilip verilmediğini kontrol etmek istiyorum.

15- Ben bir kullanıcı olarak, tez içerisinde tez içi kaynak gösterme yöntemine uygun tipte kaynak listeleme yapılıp yapılmadığını kontrol etmek istiyorum

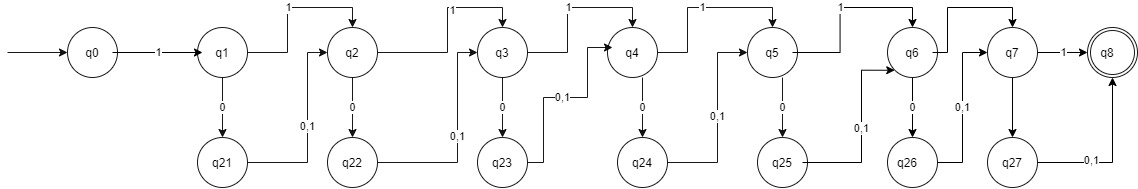
16- Ben bir kullanıcı olarak, tez içerisinde atıf sistemi isme dayalı olan kaynaklar bölümünün alfabetik sıraya göre yazılıp yazılmadığını kontrol etmek istiyorum

17- Ben bir kullanıcı olarak, tez içerisinde yazıların büyüklük olarak 10 punto ve 1,25 Aralık ayarlı yazılıp yazılmadığını kontrol etmek istiyorum

**3.1.2 Veri Modeli**

Projede veri modelinin gösterilmesi için DFA modelleri oluşturulmuştur.

* **ÖN SAYFALAR**



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q8,q21,q22,q23,q24,q25,q26,q27}

∑ = {0, 1}

q0 = {q0}

F = {q8}

q0 = Start state

q1 =Sayfa Kenar boşlukları

q2 = antet

q3  = Tez Başlığı

q4  = İsimler

q5 = Anabilim Dalı

q6 = Tarihler, Ay ve Yıl bilgileri

q7 = Yazım düzeni, sayfa numaraları gizli mi?

q8 = final state

q21 sayfa kenar boşluk hatası

q22  Antetler hatası

q23  Tez Başlığı hatası

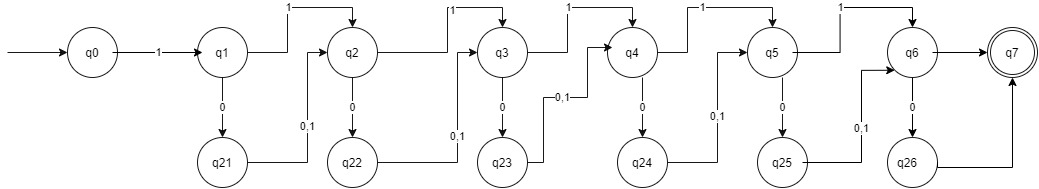
q24  İsimler hatası

q25  Anabilim Dalı hatası

q26  Tarihler, Ay ve Yıl bilgileri hatası

q27 Yazım düzeni, sayfa numaraları gizleme hatası

* ÖNSÖZ



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q21,q22,q23,q24,q25,q26}

∑ = {0, 1}

q0 = {q0}

F = {q7}

q0 Start state

q1 Kenar Boşlukları

q2 İlk paragraf içerikleri

q3  Satır aralıkları

q4  Yazar ismi, şehir ve tarih

q5 Sayfa Numarası (sayfa altına ortalı –iv (veya v))

q6 Protokol numaralı destek bildirimleri

q7 final state

q21 Kenar Boşlukları boşluk hatası

q22  İlk paragraf içerikleri hatası

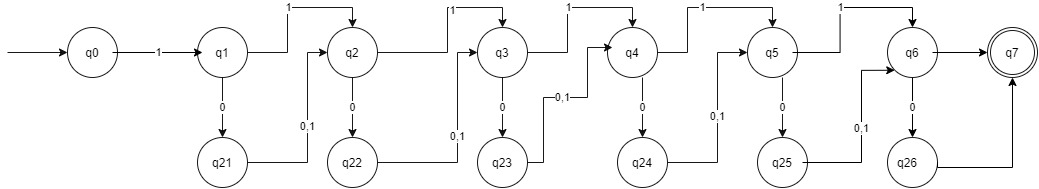
q23  Satır aralıkları hatası

q24  Yazar ismi, şehir ve tarih hatası

q25  Sayfa Numarası (sayfa altına ortalı –iv (veya v)) hatası

q26  Protokol numaralı destek bildirimleri hatası

* İÇİNDEKİLER



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q21,q22,q23,q24,q25,q26}

∑ = {0, 1}

q0 = {q0}

F = {q7}

q0 Start state

q1 Sayfa Kenar boşlukları

q2 Başlık yazısı (Küçük büyük seçili, satır alt çizgisi ve koyu olmalı) ve sonraki boşluk

q3  Satır aralıkları ve Girintiler

q4  İçindekiler kısmında verilen sayfa numaralarının tez sayfaları ile karşılaştırılması

q5  Ana Bölüm başlıkları (küçük büyük seçili, koyu, liste numaraları, 11 punto)

q6 Ekler ve Özgeçmiş, yazım düzeni, liste numarasız, Özgeçmiş sayfa numarası bulunmaz

q7 final state

q21 sayfa kenar boşluk hatası

q22  Başlık yazısı hatası

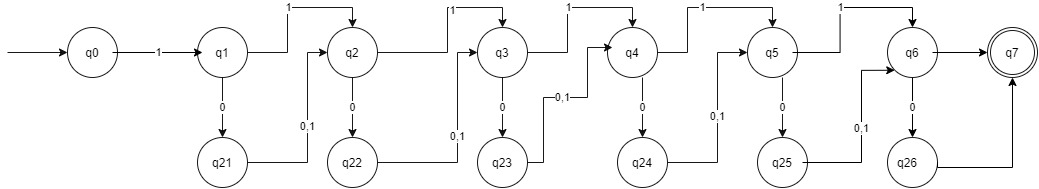
q23  Satır aralıkları ve Girinti hatası

q24  İçindekiler kısmında verilen sayfa numaralarının tez sayfaları ile karşılaştırılması hatası

q25  Ana Bölüm başlıkları hatası

q26  Ekler ve Özgeçmiş, yazım düzeni, liste numarasız, Özgeçmiş sayfa numarası bulunma hatası

* Türkçe ÖZET ve ABSTRACT (İngilizce Özet)



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q21,q22,q23,q24,q25,q26}

∑ = {0, 1}

q0 = {q0}

F = {q7}

q0 Start state

q1 Sayfa Kenar boşlukları

q2 Başlık yazısı (Küçük büyük seçili, satır alt çizgisi ve koyu olmalı) ve sonraki boşluk

q3  Tez başlığı, Yazar Adı SOYADI, Üniversite, Enstitü, Anabilim Dalı, Tarih, Sayfalar

q4  Paragraf düzeni (Özette 1 cm sol girintili, 1,25 aralık ayarlı 10 punto ile yazılır)

q5 Anahtar kelimeler (en az 3, en fazla 6 adet olmalı)

q6 Özet ve Abstract sayfası bir sayfa ile sınırlı olmalı

q7 final state

q21 sayfa kenar boşluk hatası

q22  Başlık yazısı hatası

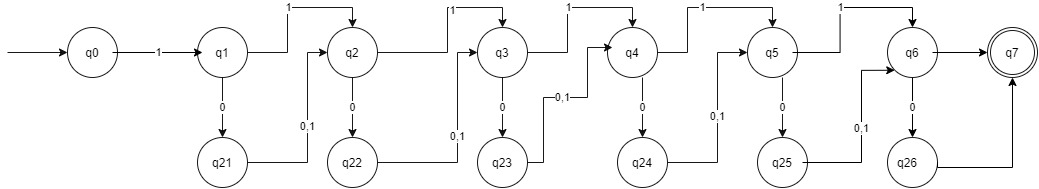
q23  Tez Başlığı hatası

q24  Paragraf düzeni hatası

q25  Anahtar kelimeler hatası

q26  Özet ve Abstract sayfası bir sayfa ile sınır aşma hatası

* LİSTELER (Şekiller, Tablolar, Ekler, Simgeler ve Kısaltmalar)



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q21,q22,q23,q24,q25,q26}

∑ = {0, 1}

q0 = {q0}

F = {q7}

q0 Start state

q1 Sayfa Kenar boşlukları

q2 Başlık yazıları (küçük büyük seçili, koyu ve ortalı), sonraki boşluk

q3  Satır aralığı ve yazı büyüklüğü, satır alt çizgisi kullanımı

q4  Yazım düzeni (ilk cümle kuralı, kaynaklar gizli, girintiler, koyu normal yazım düzeni)

q5 Sayfa bilgilerinin doğruluğu

q6 Kısaltmalarda alfabetik sıralama

q8 final state

q21 sayfa kenar boşluk hatası

q22  Başlık yazıları hatası

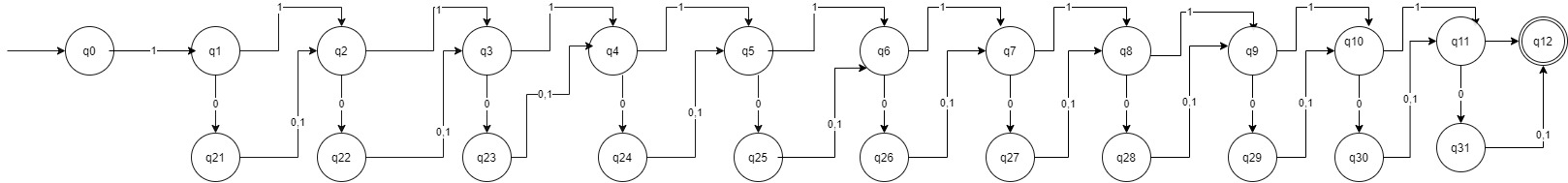
q23  Satır aralığı ve yazı büyüklüğü hatası

q24  Yazım düzeni hatası

q25  Sayfa bilgilerinin doğruluğu hatası

q26  Kısaltmalarda alfabetik sıralamama hatası

* GİRİŞ ve TEZ ANA METNİ



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q21,q22,q23,q24,q25,q26,q27,q28,q29,q30,q31}

∑ = {0, 1}

q0 = {q0}

F = {q12}

q0 Start state

q1 Sayfa Kenar boşlukları

q2 Ana bölüm Başlıkları (Tamamı 16 punto küçük büyük seçili ve koyu) sonra iki boşluk

q3  En fazla 3. Derece alt başlık olmalı

q4  Alt başlıklar koyu ve 12 punto ile sola yaslı olmalı

q5 Yazı büyüklüğü ve satır aralığı, paragraf girintileri

q6 Ana bölüm ilk sayfalarında sayfa numaraları gizli olmalı

q7 Sayfa numaraları sayfa altında ortalı, sadece rakamlar olmalı, alttan 1,5 cm yukarıda bulunmalı

q8 Numaraya dayalı atıf sistemi kullanılmış ise; kaynaklar metin içinde veriliş sırasına uygun sıralanmalı

q9 Tez sayfalarında kazıntı, silinti, karalama veya bir şekilde kapatılmış bölge bulunmamalı

q10 Farklı ortamlardan alınan şekillerin tamamına referans verilmiş mi? Farklı ortamlardan alınan şekillerin tamamına referans verilmiş mi?

q11

q12 final state

q21 sayfa kenar boşluk hatası

q22  Ana bölüm Başlıkları hatası

q23  Alt başlık aşma hatası

q24  Alt başlıklar koyu ve 12 punto ile sola yaslı olmama hatası

q25  Yazı büyüklüğü ve satır aralığı, paragraf girintileri hatası

q26  Ana bölüm ilk sayfalarında sayfa numaraları gizli olmama hatası

q27 Sayfa numaraları düzen hatası

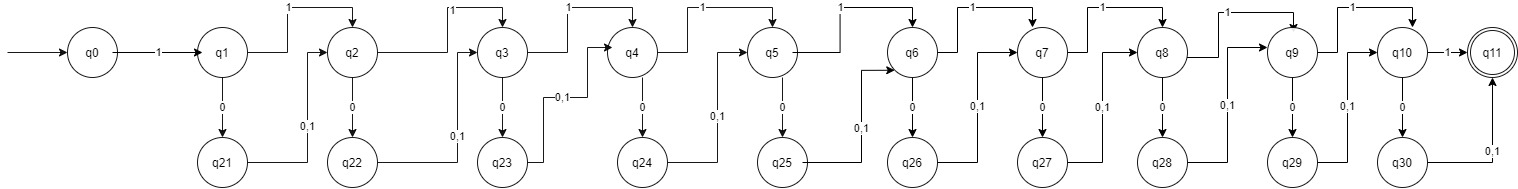
q28 kaynaklar uygun sıralanmama hatası

q29 sayfada kazıntı, silinti, karalama veya bir şekilde kapatılmış bölge hatası

q30 Farklı ortamlardan alınan şekillerin tamamına referans verilmeme hatası

q31 Kaynaklar Bölümünde listelenen tüm kaynaklara metin içinde atıfta bulunulmama hatası

* ŞEKİL, TABLO ve DENKLEM YERLEŞİMLERİ



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q21,q22,q23,q24,q25,q26,q27,q28,q29,q30}

∑ = {0, 1}

q0 = {q0}

F = {q11}

q0 Start state

q1 Numaraları her bölüm içinde yeniden başlatılmalı, sıralı olmalı

q2 Şekil yazıları şekil altında, Tablo yazıları tablo üstünde, bir boşluklu yazılmalı

q3  Bir satırdan uzun olan açıklama yazıları tek satır aralıkla yazılmalı, ikinci satırda girintili olmalı iki yana yaslı olmalı

q4  Tek satırlık açıklama yazıları metin alanına yatayda ortalanmalı

q5 Şekiller ve tablolar sayfa kenar boşluklarına taşmamalı

q6 Tüm şekil, tablo ve denklemlere metin içinde numaraları ile birlikte değinilmeli

q7 Alıntı olan tüm şekil ve tablolara kaynak gösterilmeli

q8 3. Derece numaralama yapılmamalı

q9 Denklemler paragraf girintisi ile başlamalı

q10 Denklem numaraları metin sağ kenarı ile hizalanmalı, parantez içinde verilmeli

q11 final state

q21 Numaraları her bölüm içinde yeniden başlatılmalı, sıralı olmama hatası

q22  Şekil yazıları şekil altında, Tablo yazıları tablo üstünde, bir boşluklu yazılmama hatası

q23  Satır girinti hatası

q24  Tek satırlık açıklama yazıları metin alanına yatayda ortalanmama hatası

q25  Şekiller ve tablolar sayfa kenar boşluklarına taşma hatası

q26  Tüm şekil, tablo ve denklemlere metin içinde numaraları ile birlikte değinilmeme hatası

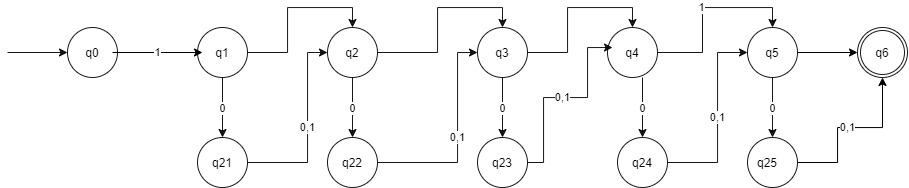
q27 Alıntı olan tüm şekil ve tablolara kaynak gösterilmeme hatası

q28 3. Derece numaralama hatası

q29 Denklemler paragraf girintisi ile başlamama hatası

q30 Denklem numaraları metin sağ kenarı ile hizalanmama, parantez içinde verilmeme hatası

* KAYNAKLAR



Q = {q0, q1, q2,q3,q4,q5,q6,q21,q22,q23,q24,q25}

∑ = {0, 1}

q0 = {q0}

F = {q6}

q0 Start state

q1 Başlık yazısı ve metinle arasındaki mesafe

q2 Yazı büyüklüğü 10 punto ve 1,25 aralık ayarlı yazılmalı

q3  İki kaynak arasında 6 nk boşluk bulunmalı

q4  Tez içi kaynak gösterme yöntemine uygun tipte kaynak listeleme yapılmalı

q5 Atıf sistemi isme dayalı ise; Kaynaklar Bölümünde alfabetik sıralama yapılmış olmalı

q6 final state

q21 Başlık yazısı ve metinle arasındaki mesafe hatası

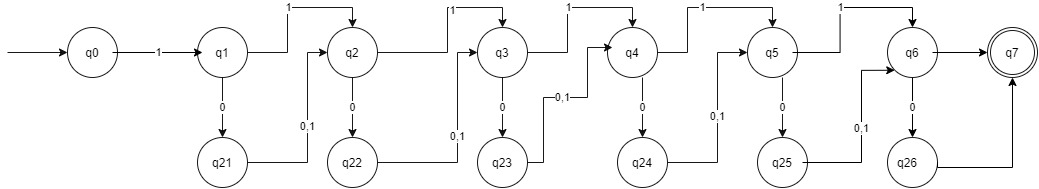
q22  Yazı büyüklüğü 10 punto ve 1,25 aralık ayarlı yazılmama hatası

q23  İki kaynak arasında 6 nk boşluk bulunmama hatası

q24  Tez içi kaynak gösterme yöntemine uygun tipte kaynak listeleme yapılmama hatası

q25  Atıf sistemi isme dayalı ise; Kaynaklar Bölümünde alfabetik sıralama yapılmış olmama hatası

* SON SAYFALAR (EKLER ve ÖZGEÇMİŞ)



Q = {q0, q1, q2,q3,q4,q5,q6,q7,q21,q22,q23,q24,q25,q26}

∑ = {0, 1}

q0 = {q0}

F = {q7}

q0 Start state

q1 Başlık yazısı ve metinle arasındaki mesafe

q2 Yazı büyüklüğü 10 punto ve 1,25 aralık ayarlı yazılmalı

q3  Her bir Ek numaralı bir başlığa sahip olmalı

q4  Özgeçmiş Örneğe uygun olmalı

q5 Özgeçmişte akademik bilgiler bulunmalı

q6 Ekler normal sayfa numaralarına sahip olmalı, Özgeçmiş sayfası numarasın olmalı

q7 final state

q21 Başlık yazısı ve metinle arasındaki mesafe hatası

q22  Yazı büyüklüğü 10 punto ve 1,25 aralık ayarlı yazılmama hatası

q23  Her bir Ek numaralı bir başlığa sahip olmama hatası

q24  Özgeçmiş Örneğe uygun olmama hatası

q25  Özgeçmişte akademik bilgiler bulunmama hatası

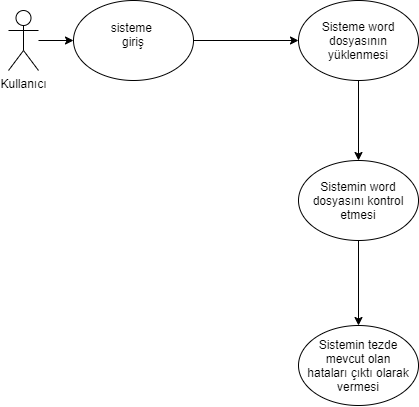
q26  Ekler normal sayfa numaralarına sahip olmalı, Özgeçmiş sayfası numarasın olmama hatası

**3.2 Gereksenen Sistemin Mantıksal Modeli**

**3.2.1 Giriş**

Kullanıcı programa bir Word dosyasını giriş olarak verdiğinde çıktı olarak tezde mevcut olan hataları vermesi beklenmektedir. Sistemin bu şekilde oluşturulması gerektiği ortaya konulmuştur.

**3.2.2 İşlevsel Model**

****

**Şekil 3.2.2 Sistemin Temel UML Diyagramı**

**3.3 Arayüz(Modül) Gerekleri**

**3.3.1 Yazılım Arayüzü**

Proje, çalışma süresince değişime gerek duyulmayacak şekilde tasarlanmıştır. Bu yüzden yazılım arayüzü kullanılmamıştır. Bir hata meydana gelmesi halinde projenin kaynak kodlarında değişiklik yapılarak hata giderilecektir.

**3.3.1 Kullanıcı Arayüzü:**

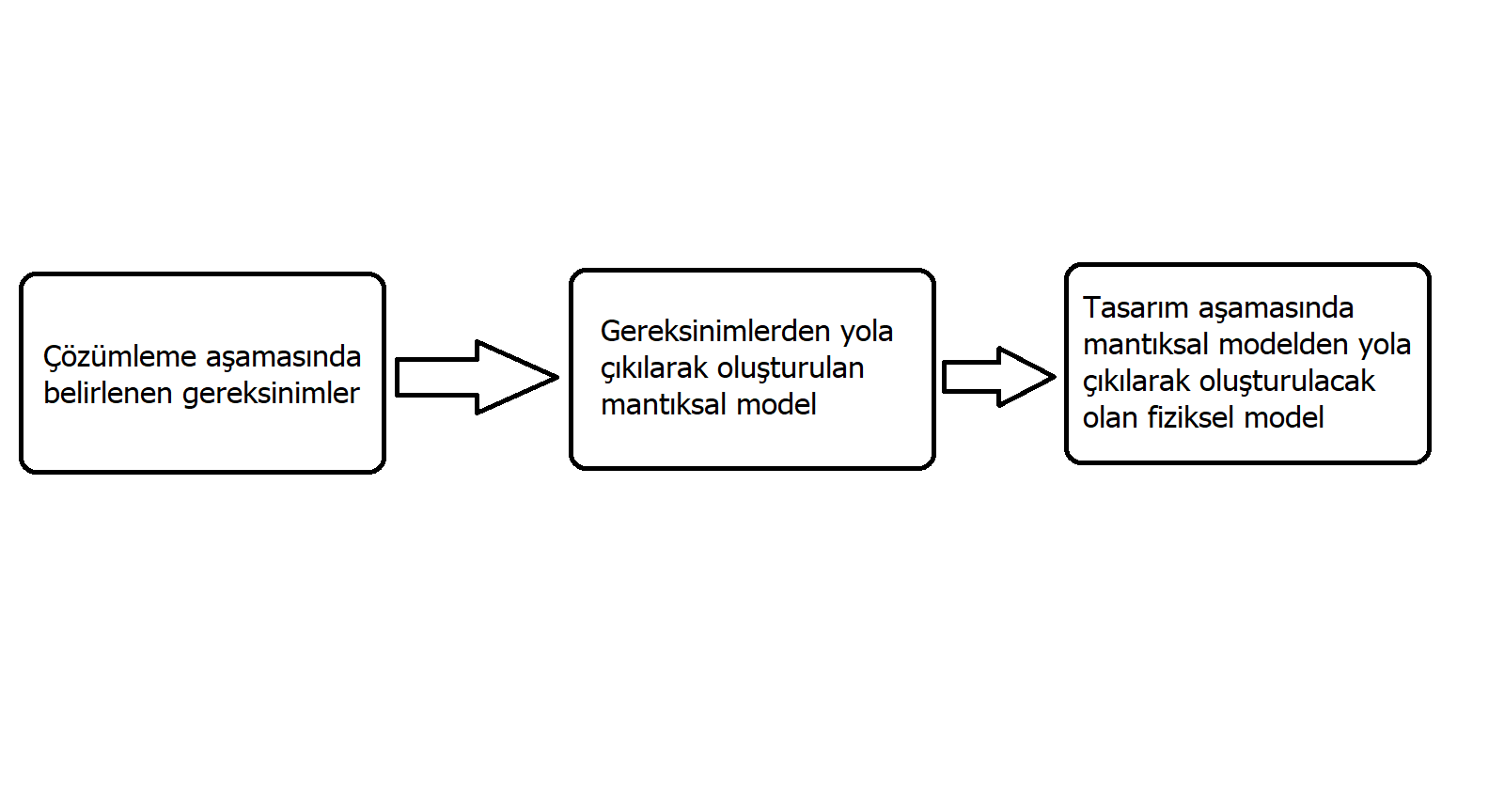
Projede kullanıcı için ek olarak bir arayüze gerek duyulmacağı düşünülmüştür. Proje terminal ekranında basit komutlar ile giriş yapılması ve gerekli işlemlerin yapılması hedeflenmiştir.

**Şekil 3.3.1.1 Terminal Ekranı ve Kullanıcıya Hataların getirilmesi**

|  |
| --- |
| **4. SİSTEM TASARIMI** |

**4.1 Genel Tasarım Bilgileri**

**4.1.1 Genel Sistem Tanımı**

****

**Şekil 4.1.1.1 Genel Sistem Tanımı**

* Gereksinimler

Örnek tezler incelenerek kullanıcılardan gelen User Story’ler aracılığıyla gereksinimler belirlenmiştir.

* İşlevsel Belirtimler

Bu kısımda sistemin işlevleri belirtilecektir. Bu sistem nasıl çalışacak, ne yapacak gibi sorulara cevap verilecektir.

Sisteme kullanıcı giriş yapıp Word dosyasını sisteme giriş olarak vermesi gerekiyor.

Sistem Word dosyasını tarar içinde, şablonlarla karşılaştırır mevcut hatalar(kenar boşluklarının karşılaştırılması, satır arası boşlukların kontrolü vb.) varsa bunları çıktı olarak vermesi beklenmektedir.

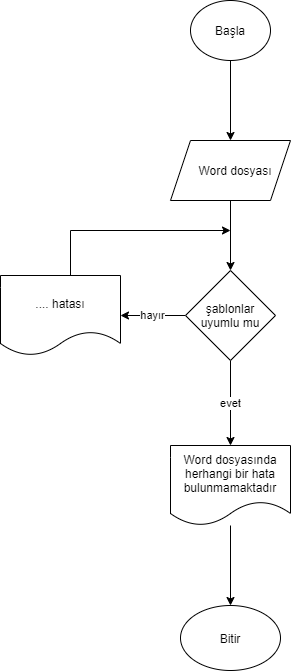
* Tasarım

Tasarım aşamasında olacaklar grafiksel olarak belirtilmiştir.

**4.1.2 Varsayımlar ve Kısıtlamalar**

* Sistemde herhangi bir kısıtlama bulunmamaktadır. Kullanıcının Sisteme Word dosyasını giriş olarak vermesi yeterlidir.

**4.1.3 Sistem Mimarisi**

****

**Şekil 4.1.3.1 Sistem Akış Diyagramı**

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 fikir verecektir.

**4.1.4 Testler**

Sistem testleri çeşitli üniversitelere ücretsiz şekilde dağıtılıp bunlardan geri dönüş yapılması hedeflenmektedir.

**Alfa Aşaması:** Sistem öncelik üniversitelerin belli bölümlerinde kullanılacaktır. Buralardan gelen geri dönüşler ile sistem hatalarının giderilmesi ve sistem testinin yapılması sağlanacaktır.

**Beta Aşaması**: Kullanıcılar, geliştirilen sistemi pilot bölgelerde, yetkili personel eşliğinde yapacaktır.

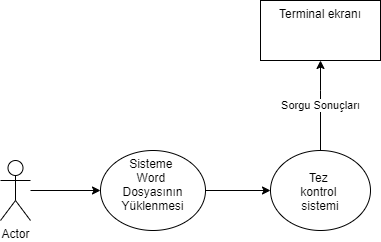
**4.1.5 Performans**

Test aşaması ile müşterilerden gelen geri dönüşler ile sistemin performansının değerlendirilmesi, gerekli değişikliklerin yapılması ve sistemin daha uyumlu bir hale getirilmesi hedeflenmektedir.

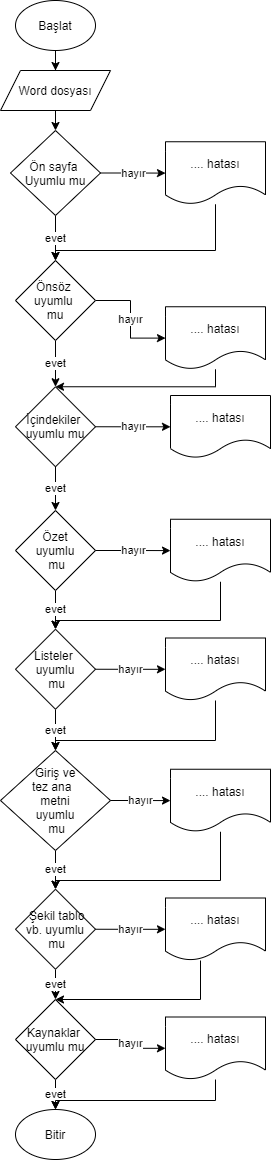
**4.2 Süreç Tasarımı**

**4.2.1 Genel Tasarım**

Genel olarak tasarımda ilk önce veri tabanı modeli oluşturulmuştur. Sonra sisteme giriş modülü, daha sonra yönetici modülleri, daha sonra kullanıcı arayüzü ve en sonunda hesap ve para transfer modülleri oluşturulmuştur.



**Şekil 4.2.1.1 Sistem Süreç Genel Tasarımı**



**Şekil 4.2.1.2 Modüller**

|  |
| --- |
|  |
| **5. SİSTEM GERÇEKLEŞTİRİMİ** |

**5.1 Giriş**

Gerçekleştirim bölümü, tasarım kısmında oluşturulan fiziksel modelin programlama dilleri ve araçları kullanılarak gerçeğe dönüştürülmesini içerir. Fiziksel modelin gerçekleştirimi için de en başta bu süreçlerin takibine yardımcı olacak bir yazılım geliştirme ortamı gerekmektedir.

C:\Users\Maruf\Downloads\dfgdgdgd.jpg

**Şekil 5.1.1. Sistem Gerçekleştirim Modeli**

**5.2 Yazılım Geliştirme Ortamları**

Yazılım geliştirme ortamı, tasarım kısmında oluşturulan fiziksel modelin gerçeğe dönüştürülmesinde kullanılacak olan:

* Programlama Dilleri
* Tasarım Araçları
* Veri Tabanı Yönetim Sistemi (VTYS) gibi araçların olduğu bir ortamdır.

**5.2.1 Programlama Dilleri**

Bu sistemde kullanılacak programlama dili python’dır. Hem nesne tabanlı oluşu, hem fonksiyonel programlama uyumluluğu hem de sistemin ilk sürümü olacağı için python programlama dili seçilmiştir.



**Şekil 5.2.1.1. Python** **Programlama Dili**

**5.2.3 Kodlama Stili**

Sistemin kodlama işleminin daha kolay ve anlaşılabilir olması için sisteme özel belirli bir kodlama stili olması gerekmektedir. Kodlama stili 4 aşamadan oluşur.

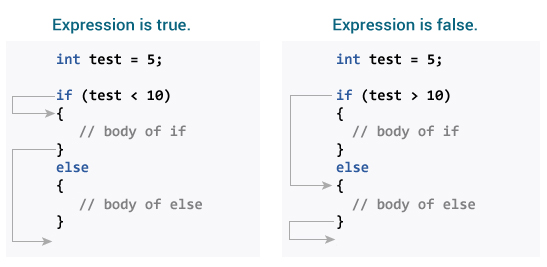
**5.3.1 Açıklama Satırları**

Program kod satırlarının arasına kodlarda düzenleme yapacak yetkili personellerin kullanması adına açıklama (yorum) satırları yerleştirilmiştir.

**5.3.2 Kod Biçimlemesi**

Programın okunabilirliğini artırmak amacıyla oluşturulmuş biçimlemedir.

Örnek:



Şekil 5.3.2.1 Python if else

**5.3.3 Anlamlı İsimlendirme**

Program kodlarındaki değişken isimleri Camel Case (degiskenAdi) tipinde ve değişkenin yaptığı işi açıklayacak biçimde verilmiştir.

**5.3.4 Yapısal Programlama Yapıları**

Yapısal programlama, büyük çaptaki projelerin daha küçük parçalara ayrılarak kodlanması ve sorunlarının bu sayede daha hızlı giderilmesi yöntemidir. Bu yöntem, üç tür yapılanma kullanır:

* Sıralı (Ardışık) Yürütme
* Mantıksal (Şartlı) Yürütme
* Döngü

**5.3 Program Karmaşıklığı**

Program karmaşıklığını ölçmek için birçok teorik model geliştirilmiştir. Bu modellerin en eskisi ve yol göstericisi McCabe karmaşıklık ölçütüdür. Söz konusu ölçüt 1976 yılında geliştirilmiştir. Bu konuda geliştirilen diğer ölçütlerin çoğu, bu ölçütten esinlenmiştir. Bir program bir ana program ve onunla ilgili birden fazla alt programdan oluşur. Gerek ana program gerekse alt programların tümü, McCabe karmaşıklık ölçütünün hesaplanmasından önce çizge biçimine dönüştürülmelidir. McCabe ölçütü, bir programda kullanılan koşul deyimlerinin program karmaşıklığını etkileyen en önemli unsur olduğu esasına dayanır ve iki aşamada uygulanır.

**5.3.1 McCabe Karmaşıklık Ölçütü Hesaplama**

Ana program ve alt programlar çizge biçimine dönüştürüldükten sonra programın McCabe karmaşıklık ölçütü (V(G)) aşağıdaki biçimde hesaplanır.

**Şekil 5.3.1.1 McCabe Karmaşıklık Ölçütü**

k: kenar sayısı

d: düğüm sayısı

p:bileşen sayısı

Bu formüle göre: k: 10, d: 9, p: 1 V(G) = 10 – 9 + 2 = 3 olarak bulunur.

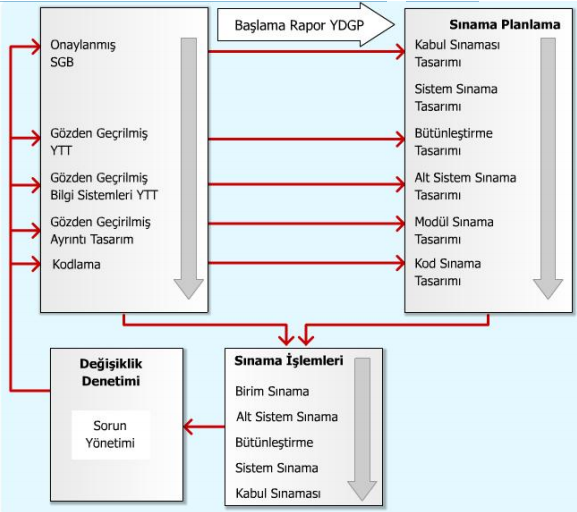
**6. DOĞRULAMA VE GEÇERLEME**

6.1. Giriş Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlemesi, 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 gerçeklemesi için sınamaların hazırlanıp yürütülmesi biçiminde özetlenebilir.

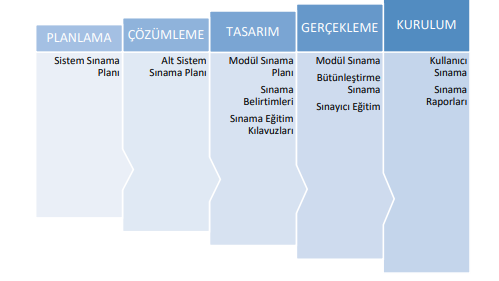
Şekil 6.1 Doğrulama ve Geçerleme

6.2. Doğrulama ve Geçerleme Yaşam Döngüsü



Şekil 6.2 Doğrulama ve Geçerleme Yaşam Döngüsü

6.3. Yaşam Döngüsü Boyunca Sınama Etkinlikleri



Şekil 6.3. Yaşam Döngüsü Boyunca Sınama Etkinlikleri

|  |
| --- |
| **7. BAKIM** |

**7.1 Giriş**

Sistemin tasarımı ve gerçekleştirimi bittikten sonra sistemdeki hataların gözden geçirilip düzeltilmesi için sistemin test mühendisleri tarafından bakıma alınması gerekmektedir. Daha önce de belirtildiği gibi güvenilir olması gereken ve en ufak bir hatada verilerin kaybolabileceği bir sistemden bahsedilmektedir. Bakım bölümüne ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı temel alınmıştır.



**Şekil 7.4.1 Yazılım Bakım Süreci**

**7.4 Yazılım Bakımı**

Bir yazılımın bakımı, yazılımın geliştirilmesi tamamlandıktan sonraki süreçte, yazılımın ömrünü sağlıklı bir şekilde sürdürmesini sağlayan yaklaşımlar bütünüdür. Yazılım bakımı yazılım geliştirme sürecinin (Software Development Process) son parçasını oluşturur. Müşteri gereksinimlerine göre oluşturulan yazılım çıktısı, değişen teknolojik araçlara ve müşteri gereksinimlerine göre oluşturulan yazılım çıktısı, değişen teknolojik araçlara ve müşteri gereksinimlerine göre oluşturulan sürekli bakımın yapılması, denetlenmesi gerekir. Nasıl bir arabanın yıllık bakımı yapılıyorsa, yazılım ürünlerinin de belirli sürümler dâhilinde geliştirilmesi, iyileştirilmesi ve korunması yapılmalıdır.

**7.4 Bakım Süreç Modeli**

IEEE 1219-1998 Yazılım Bakım Standardı esaslarına göre sağlanacaktır.

|  |
| --- |
| **8.** SONUÇ |

Sonuç olarak bir tez dosyasının basit bir şekilde kontrollünü yapan tez analiz sistemi adında bir program geliştirildi. Program aynı lokasyonda ki bir doc veya pdf dosyasını alarak kontrol etmektedir. Şablonlara uyup uymadığı, tez hakkında bilgileri çekme vb. özellikleri sağlamaktadır.

|  |
| --- |
| **9. KAYNAKLAR** |

1. <http://muhammetbaykara.com/2017/04/05/ymh-114-ornek-taslak-proje/>
2. <http://bilgisayarkavramlari.com/2008/11/11/belirli-sonlu-otomat-deterministic-finite-automat/>
3. <https://python-istihza.yazbel.com/>
4. <https://python-docx.readthedocs.io/en/latest/>
5. <https://www.horato.com/tr/post/yazilim-mimarisi-software-architecture-60>
6. <https://medium.com/architectural-patterns/pipe-and-filters-mimarisi-fee576ef6fb0>