



FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği Bölümü

YM319 – Programlama Dilleri
Proje Uygulaması ve Dokümantasyonu

TEZ KONTROL SİSTEMİ

TKS

Geliştiren

170541046 GÜNDÜZ-B
Mehmet Emin ULUSOY
180541608 GÜNDÜZ-B
Baraa KALAAJİ
180541026 GÜNDÜZ-B
Cemal TİRYAKİ
180541080 GÜNDÜZ-B
İsmail ONUR

Proje Yürütücüsü

Doç. Dr. Fatih ÖZKAYNAK

İçindekiler

1. GİRİŞ.....	4
1.1. Projenin Amacı	4
1.2 Projenin Kapsamı.....	4
1.3 Referanslar	4
2. PROJE PLANI.....	5
2.1 GİRİŞ.....	5
2.2 Proje İş-Zaman Planı	6
2.3 Proje Ekip Yapısı	7
2.4 İş Tanımları.....	7
2.5 Önerilen Sistemin Teknik Tanımları	7
2.7 Kullanılan Özel Geliştirme Araçları ve Ortamları.....	8
2.8 Eğitim Planı	8
2.9 Proje Standartları, Yöntem ve Metodolojiler	9
3. SİSTEM ÇÖZÜMLEMESİ.....	11
3.1 Sistemin Use-Case Diyagramı:	11
3.2 DFA Diyagramı:	12
3.4 Proje UML Diyagramı:	13
3.4 Gereksinimler	14
3.4.1 Gereksinim Verisi Toplama	14
3.4.1.1 Sorma Yöntemi	14
3.4.1.2 Anket.....	15
3.4.2 KULLANICI ARAYÜZ PROTOTİPİ	16
3.4.3 İşlevsel Gereksinimler.....	18
3.4.4 Performans Gereksinimleri.....	19
3.4.5 Fiziksel Çevre.....	19
3.4.6 Arayüzler (g8)	19
3.4.7 İşlevsellik(g9)	19
3.4.8 Veri(g7).....	19
3.4.9 Kaynaklar.....	19
3.4.10 Güvenlik(g6)	20
3.4.11 Kalite Güvencesi	20
3.5 Kalite Özellikleri	21
3.5.1 Güvenilirlik (“Reliability”)	21
3.5.2 Kullanılabilirlik (“Availability”)	21
3.5.3 Güvenlik	21
3.5.4 Taşınabilirlik (“Portability”).....	21

3.5.5 Kullanılabilirlik (“Usability”)	21
3.6 GEREKSİNİMLERİN ÖNCELİĞİ ve GEREKSİNİM TABLOSU:	22
4. SİSTEM TASARIMI	23
4.1 Kullanıcı Arabirimleri	24
4.2 Testler	24
4.2.1 Alfa Aşaması:	24
4.2.2 Beta Aşaması:	24
4.3 Performans.....	24
5. GERÇEKLEŞTİRME.....	25
5.1 Yazılım Geliştirme Ortamları.....	25
5.2 Programlama Dilleri.....	25
5.3 Olağan Dışı Durum Çözümleme	25
5.4 Olağandışı Durum Tanımları	25
6.DOĞRULAMA GEÇERLEME	40
6.1 Alt Sistem Sınama:	41
6.2 Sistem Sınama:	41
6.3 Kabul Sınama:	41
6.4 Sınama Yöntemleri	41
6.4.1 Kara Kutu Testi	41
6.4.2 Beyaz Kutu Testi	42
7.BAKIM	42
7.1 KURULUM	42
8.SONUÇ	43

1. GİRİŞ

1.1. Projenin Amacı

Günümüzde birçok konu hakkında hazırlanılmış tez vardır. Ülkemizde şuan yok.gov.tr adresine kayıtlı 594.578 adet tez vardır. Fırat Üniversitesinin ise toplamda 9071 adet kayıtlı tezi bulunmaktadır. (*) ve 2019 yılında tez indirilme sayısı **16.364.817**'dir. bunlar azımsanacak sayılar değildir. Buradan anlaşıldığı gibi tez yazımı ve tezlere olan ilgi gayet büyüktür. Ama tez yazımı çok zorlu bir süreç olduğu göz ardı edilemeyecek bir gerçektir. Tez yazılırken belli standartlar kullanılır ve bu standartların dışına çıkılamaz. Ve bu yüzden gözle görülemeyecek hatalara saatlerce vakit harcamamak bizim için çok önemlidir. Çünkü her geçen zaman bizim çalışma motivasyonumuza ya da disiplinimize etki eder. Bizim projemiz ise burada gözle görülemeyecek hataları ve eksiklikleri sizlerin saatlerce aramanızdan kurtarıp bu işlemi dakikalar içerisinde tespit edip size bildirir. Ve sizin bu hataları önlemenize olanak sağlar. Bu sayede odaklanmanız gereken tek şey tez konusu olur. Ve bu sayede çok daha iyi ve içeriği güzel tezler ortaya çıkabilir.

[https://acikerisim.yok.gov.tr/Sayfalar/Haberler/yok-ulusal-tez-merkezi-istatistikleri.aspx\(*\)](https://acikerisim.yok.gov.tr/Sayfalar/Haberler/yok-ulusal-tez-merkezi-istatistikleri.aspx(*))

1.2 Projenin Kapsamı

Proje kapsamı Fırat Üniversitesi'dir. Projemiz **Fırat Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kılavuzunda ki kurallar** kapsamında yazılan tezler için geliştirilmiştir.

1.3 Referanslar

Bu belge şablonu için Yazılım Mühendisliğinin Temelleri dersi ve Programlama Dilleri dersi için verilen şablonlar kullanılmıştır.

2. PROJE PLANI

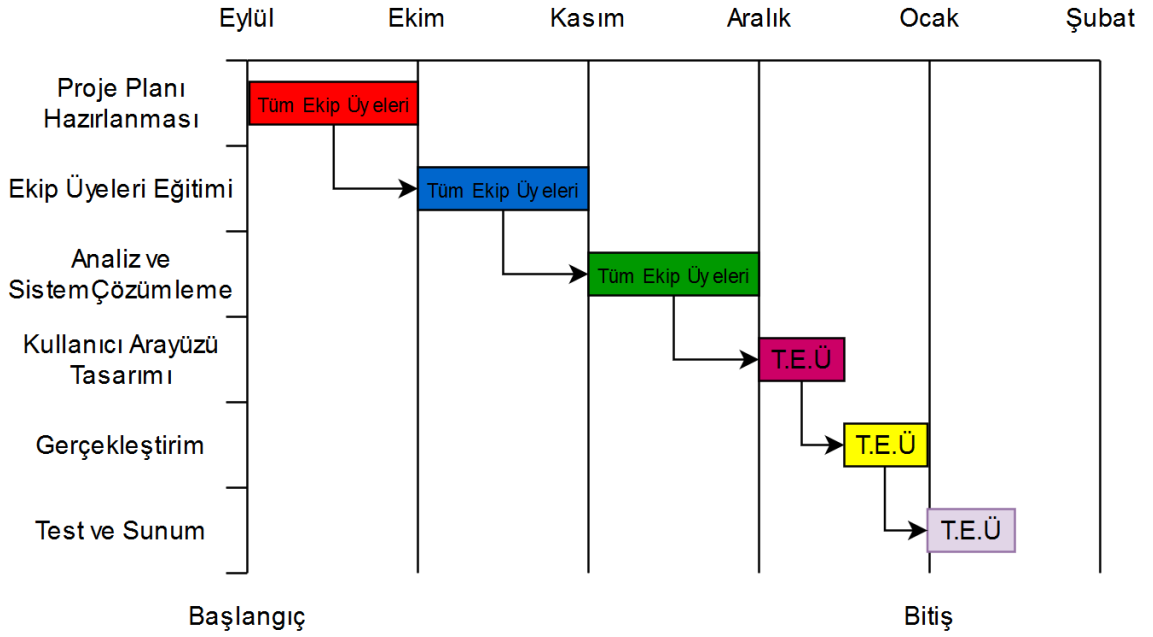
2.1 GİRİŞ

Uygulamaya öncelikle işlenilecek pdf yolu girilir. Eğer girilen pdf yolunda ki belge tezin kontrol edilmesi için yeterli kriterlere (BEYAN, ÖNSÖZ, İÇİNDEKİLER, KAYNAKLAR vb.) sahipse program bir sonraki aşama olan kontrol aşamasına geçer. Burada tez içerisinde ki her bölüm için verilen kurallar kapsamında titizlikle kontroller gerçekleşir. Proje için 4 kişilik bir ekip kurulmuştur. Her ekip üyesi farklı bir modül üzerinde geliştirme yapmaktadır.

2.2 Proje İş-Zaman Planı

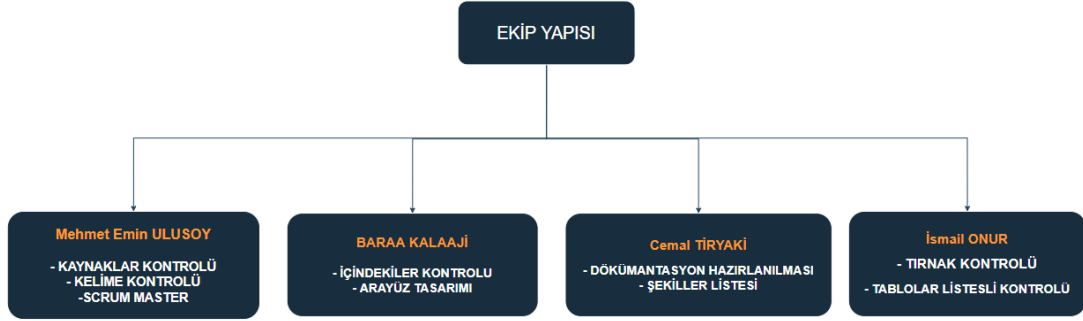


Gantt Diyagramı:



2.3 Proje Ekip Yapısı

Ekibimizin 4 kişi olmasına karar verilmiştir. Bu üyeler sırasıyla proje yürütücüsü, grafik tasarımcısı, ürün yönetimi ve 1 tane programcı (**Windows**) olarak seçilmiştir.



2.4 İş Tanımları

Dökümantasyon Hazırlanması: Burada ki ekip üyesi ürünün iyi bir şekilde ifade edilebilmesi için dökümantasyon ve ürün eğitimi veren kişidir.

Scrum Master: Scrum ekip yapısının oluşturulmasında, iş planlamasında ve scrum takım teorisinin uygulanmasında çevik yaklaşım kurallarına uyarak kararları alan kişidir.

Arayüz Tasarımı: Uygulama içi grafikler, ikonlar ve logolardan sorumlu programcılarla uyumlu bir şekilde çalışacak kişidir.

Modül Tasarımları: Yukardaki tanımların dışında kalan kısımlar, her ekip üyesi için tabloda belirtilmiştir. Her ekip üyesi kendi kontrolünü yapacağı kısımdan sorumludur.

2.5 Önerilen Sistemin Teknik Tanımları

Önerilen sistemimiz, minimum 1GB ram ve Windows 10 işletim sistemine sahip her cihazda sorunsuz bir şekilde çalışmaktadır.

2.7 Kullanılan Özel Geliştirme Araçları ve Ortamları

Uygulamamız Masaütü uygulama olarak tasarlamıştır.

- C#
- VISUAL STUDIO 2019
- ISHARPTXT
- REGEX

2.8 Eğitim Planı

Projeden kazanılacak en önemli olaylardan biride eğitimidir. 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;

- Takım için Agile Metodoloji ve Scrum Eğitimi
- C# Eğitimi
- C# GUI Eğitimi
- Visual Studio Code 2019 Eğitimi
- Regex (Regular Expression) Eğitimi

Gereken eğitimlerdir.

VERİLECEK EĞİTİMLER	Başlama Tarihi	Bitiş Tarihi
C# Eğitimi	1 Mayıs 2021	16 Mayıs 2021
C# GUI Eğitimi	1 Mayıs 2021	16 Mayıs 2021
Visual Studio Code 2019 Eğitimi	1 Mayıs 2021	1 Haziran 2021
Regex (Regular Expression) Eğitimi	1 Mayıs 2021	1 Haziran 2021
Agile ve Scrum Eğitimi	1 Mayıs 2020	15 Mayıs 2020

2.9 Proje Standartları, Yöntem ve Metodolojiler

Bu proje kapsamında Agile Metodoloji ve Scrum kullanılacaktır.

Agile Metodoloji: Yazılım sistemlerini etkili ve verimli bir şekilde modellemeye ve dokümantasyonunu yapmaya yönelik pratiğe dayalı yöntemlere denir.

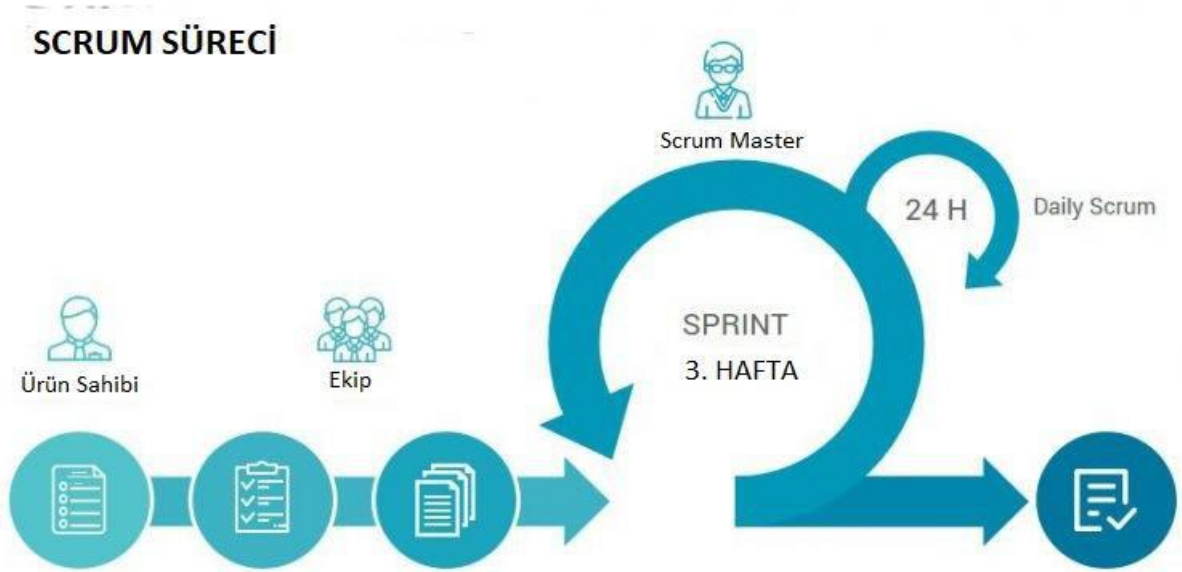
- ▣ Bireyler ve etkileşimi, süreç ve araca tercih etmek.
- ▣ Çalışan bir yazılımı, detaylı belgelendirmeye tercih etmek.
- ▣ Müşteri ile işbirliğini, sözleşmedeki kesin kurallara tercih etmek.
- ▣ Değişikliklere uyum sağlayabilmeyi, belirli bir plana tercih etmek.

Scrum: Düzenli geri bildirim ve planlamalarla hedefe ulaşmayı sağlar. Bu anlamda ihtiyaca yönelik ve esnek bir yapısı vardır. Müşteri ihtiyacına göre şekillendiği için müşterinin geri bildirimine göre yapılanmayı sağlar. Şeffaflık ön plandadır.

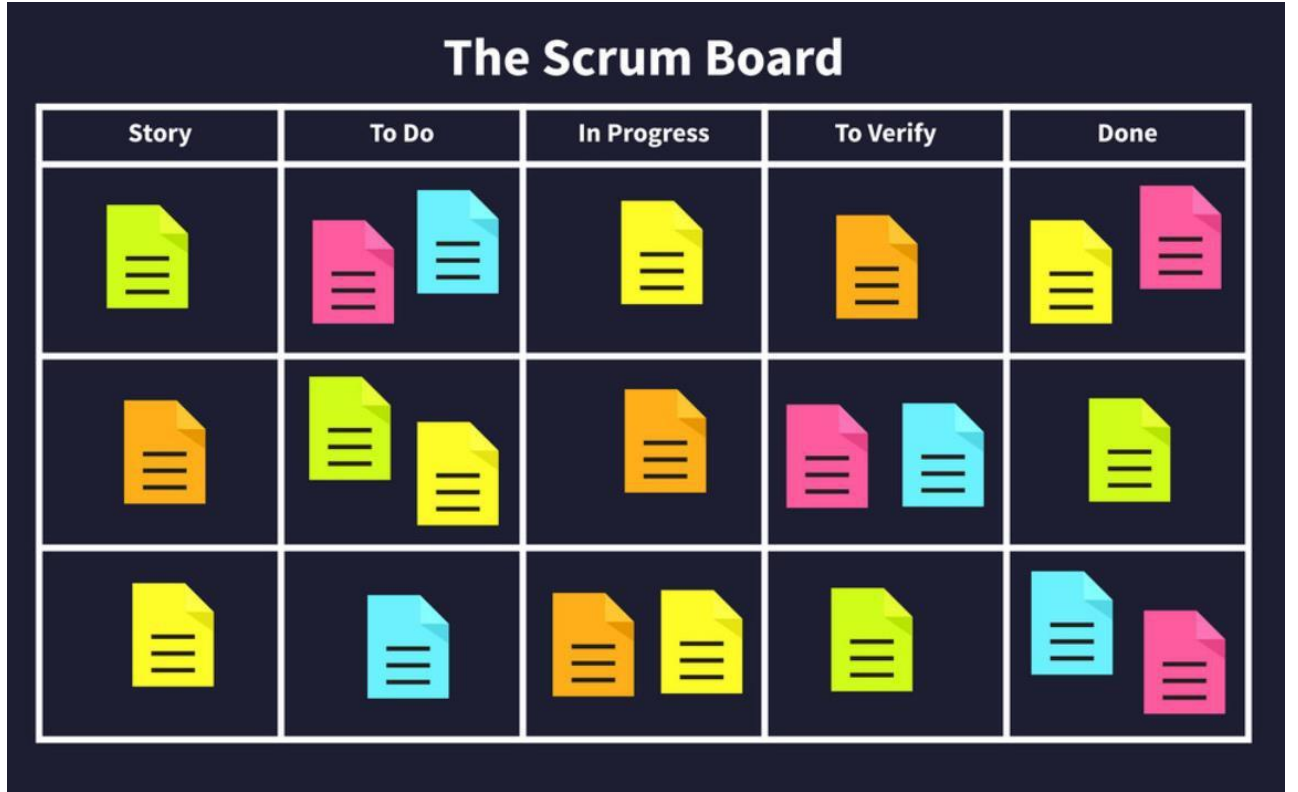
Scrum kapsamında ekip içi ve proje içi toplantılar yapılmaktadır. Ekip içi yapılan toplantılar Daily Scrum olarak adlandırılır ve 15 dakikadır. Bu toplantılar günlük olarak yapılır, Scrum Master tarafından 24 saat için yapılacaklar ekibe iletilir ve plan yapılır.

Projenin daha rahat takip edilmesi için kullanılacak Scrum Task Board ektedir.*

Sprintler 3 haftada olacak şekilde planlanmıştır. Sprint esnasında Ürün sahibi, Scrum master, Ekip, İş analisti ve İş sahibi katılmalıdır.



*Scrum Board:



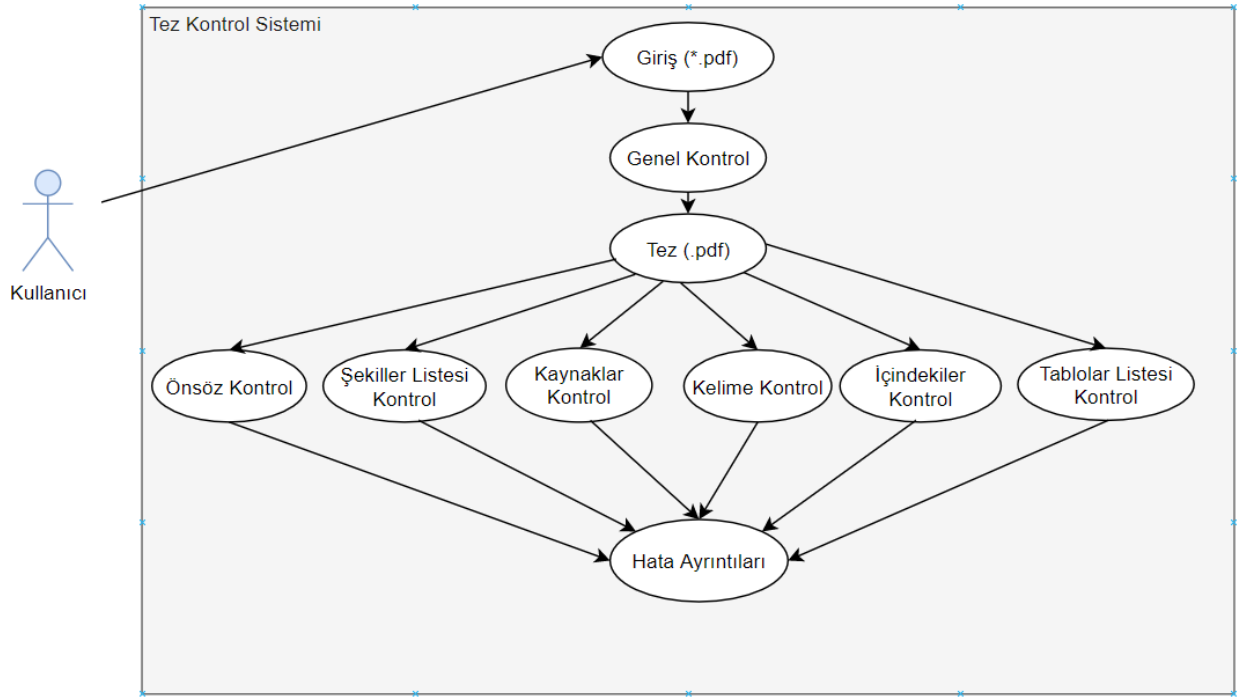
****To Do:**Yapılacaklar – **In Progress:**Şuan Yapılanlar – **To Verify:**Doğrulama-Test – **Done:**Yapıldı
Scrum Master ve ekip tarafından To Do listesi belirlenecek, her ekip üyesi yapacağı işi listeden seçip uygun yere yapıştıracak, iş sonuna kadar süreç panodan takip edilebilecek.

Proje standartları aşağıdaki gibidir.

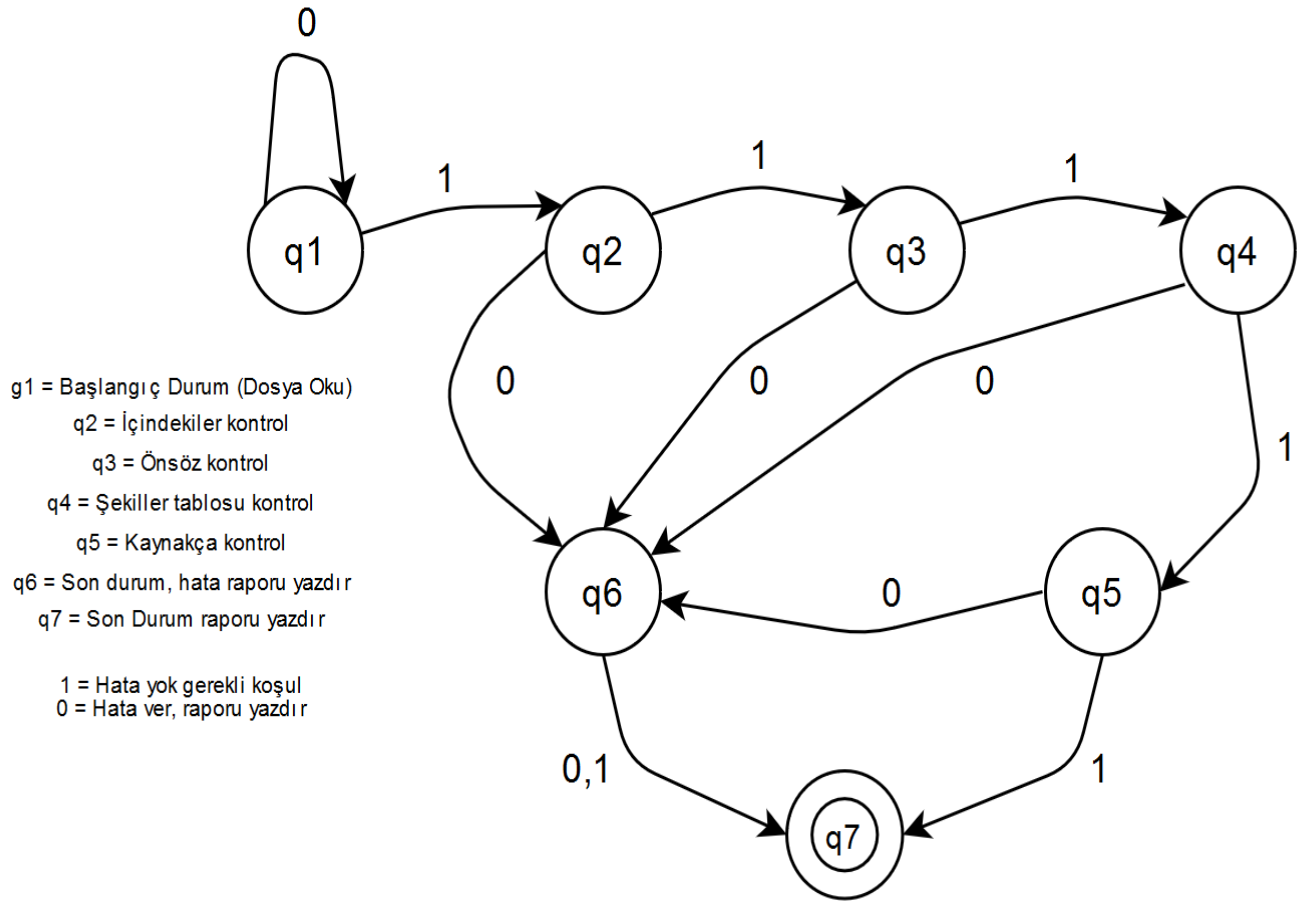
Aşama	Kullanılan Yöntem/Araçlar	Ne İçin Kullanıldığı	Çıktı
Planlama	<ul style="list-style-type: none">- Veri Akış Şemaları,- Süreç Belirtilimleri,- Görüşme,- Maliyet Kestirim Yöntemleri- Proje Yönetim Araçları	<ul style="list-style-type: none">- Süreç İnceleme- Kaynak Kestirimi- Proje Yönetimi	Proje Planı
Çözümleme	<ul style="list-style-type: none">- Süreç Belirtilimleri,- Veri Akış Şemaları,- Görüşme,- Nesne İlişki Şemaları,- Veri Sözlüğü	<ul style="list-style-type: none">- Süreç Çözümleme- Veri Çözümleme	Sistem Çözümleme Raporu
Çözümlemeden Tasarıma Geçiş	<ul style="list-style-type: none">- Akışa Dayalı Çözümleme,- Süreç Belirtilimlerinin Program Tasarım Diline Dönüştürülmesi- Nesne İlişki Şemalarının Veri Tablolarına Dönüştürülmesi	<ul style="list-style-type: none">- Başlangıç Tasarım- Ayrıntılı Tasarım- Başlangıç Veri Tasarımı	Başlangıç Tasarım Raporu
Tasarım	<ul style="list-style-type: none">- Yapısal Şemalar- Program Tasarım Dili- Veritabanı Tabloları- Veri Sözlüğü	<ul style="list-style-type: none">- Genel Tasarım- Ayrıntılı Tasarım- Veri Tasarımı	Sistem Tasarım Raporu

3. SİSTEM ÇÖZÜMLEMESİ

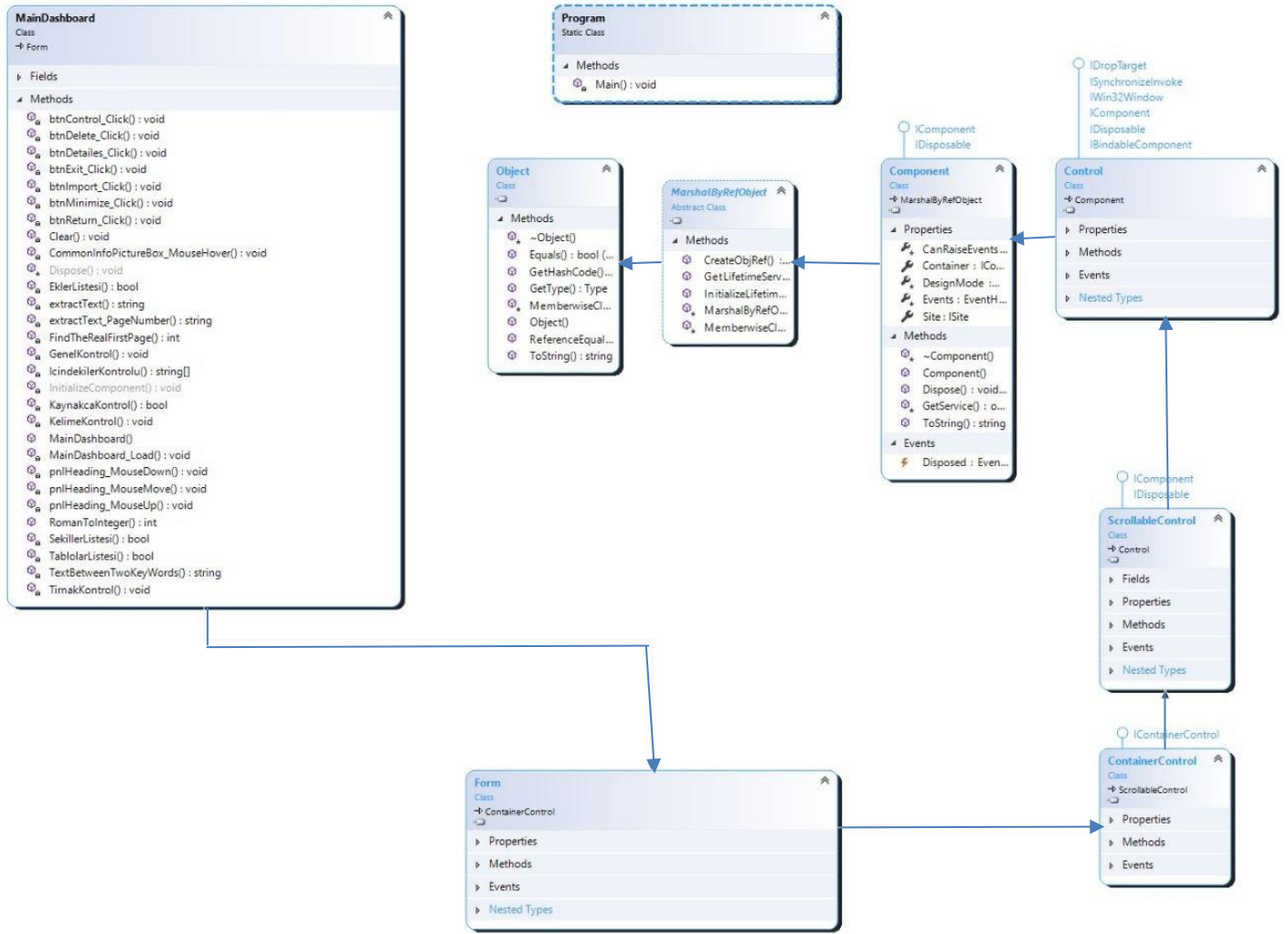
3.1 Sistemin Use-Case Diyagramı:



3.2 DFA Diyagramı:



3.4 Proje UML Diyagramı:



3.4 Gereksinimler

3.4.1 Gereksinim Verisi Toplama

Sorma yöntemi ile sistemin teknik gereksinimleri belirlenmiştir. Bunun yanında uygulamayı kullanabilecek kitleye anket uygulanmıştır. Bu süreçte ekibimiz tarafından hazırlanan sorular ve anket aşağıdadır;

3.4.1.1 Sorma Yöntemi

Donanım:

1. Mevcut donanım dışında yeni donanım alınmalı mıdır?
2. Mevcut donanım potansiyeli karşılayabilir mi?
3. Bilgisayarın güvenliğini için lisanslar mevcut mu?
4. Mevcut sistem için yazılım geliştirme ekibinin işletim sistemleri neler?

Yazılım:

1. Kullanılacak programlama dilleri neler?
2. Daha önce bu dillerle ilgili tecrübeniz var mı?
3. Kullanılacak İDE'ler ile ilgili bilgi sahibi misiniz?
4. Mevcut uygulama dışında hangi uygulamalar kullanımda?
5. Mevcut uygulama başka bir sistemin bileşeni midir?

Arayüz:

1. Ekran tasarımının nasıl olmasını istersiniz?
2. Uygulamada hangi renklerin kullanılmasını istersiniz?

3.4.1.2 Anket

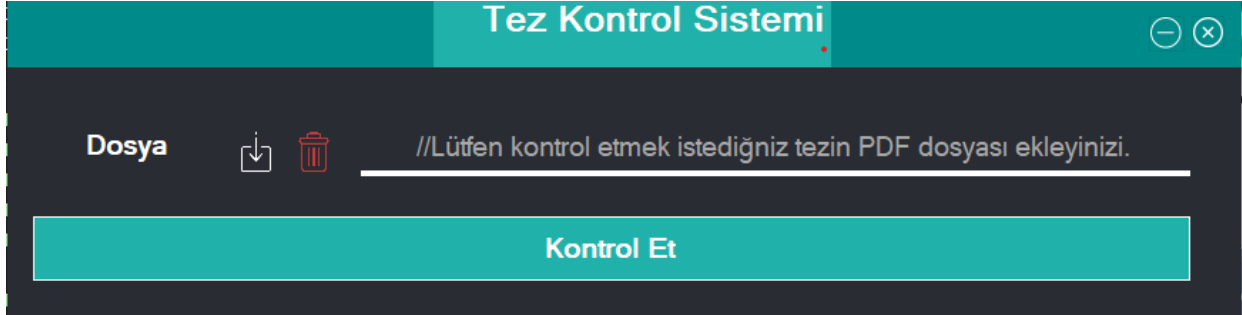
Anket olarak aşağıdaki anket kullanılacaktır. Evet/Hayır dışında verilen cevaplar son sütuna kaydedilecek.

SORULAR	EVET	HAYIR	CEVAPLAR
1.Daha önce herhangi bir Tez Kontrol Sistemi kullandınız mı?			
2.Tez hazırlama aşamasında çok fazla yazım hatası yapar mısınız?			
3.Tez hazırlarken kaynaklar kısmında kaynakları belirtmeyi unuttuğunuz oldu mu?			
4.Tez hazırlarken içindekiler kısmında eksiklikleriniz oldu mu?			
5.Tez kontrolünü gerçekleştiren bir programınız olsa kullanır mısınız ?			
6.Üniversitelerin Tez Kontrol Sistemine ihtiyacı bulunmakta mıdır?			
7.Üniversitenizde bu tarz bir uygulama kullanır mısınız?			
8.Tez Kontrol Sistemini güvenli buluyor musunuz ?			
9.Tez Kontrol Sistemi işinize yarayabilir mi?			
10.Tez Kontrol Sistemini siz yapsaydınız ne gibi eklentiler kullanırdınız?			
11.Tez Kontrol Sisteminde bulunmasını istediğiniz özellikler nelerdir?			
12.Çalışma durumunuz nedir? (Öğrenci-Akademisyen)			

*Gereksinim verisi toplamaları ile işlevsel ve yardımcı gereksinimler toplanacaktır.

3.4.2 KULLANICI ARAYÜZ PROTOTİPİ

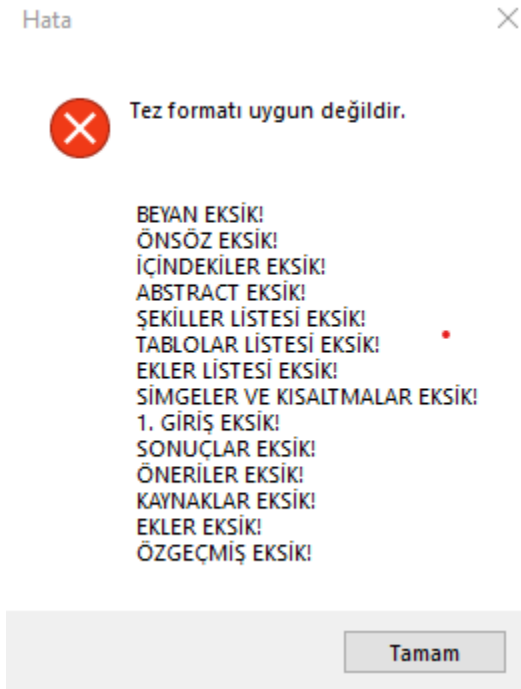
1. Açılış Sayfası



Uygulamanın Anasayfası yukarıdadır.

Anasayfa: Döküman belgesinin seçildiği ve kontrol et butonu ile programın başlatıldığı sayfa.

2. İlk Kontrol



Hata Sayfası : Eğer yüklenen pdf dosyası tez olmak için gerekli yeterlilikleri sağlamıyorsa kullanıcı uyarılıyor.

3. Anasayfa



Ansayfa: Burada kontrol edilen tezin çıktısı ekrana basit ve genel bir şekilde sunuluyor. Tekrardan başka bir pdf verilerek çalıştırılabilir. İlgili içeriklerde hata varsa bölüm yanlarına yazılıyor. Tooltipler kullanılmıştır, içerik üzerine gelinerek gerekli ayrıntılarda görülebilir. Sol tarafta 'Hataların Dağılımı' şeklinde bir grafikte yer almaktadır.

4. Ayrıntı Sayfası

Tez Kontrol Sistemi	
Hatanın ayrıntıları	Hatanın kaynağı
Blok atıf sayısı 3'ten fazla: [60-64]	Kaynaklar
Blok atıf sayısı 3'ten fazla: [9-12]	Kaynaklar
' ANABİLİM DALI ' KELİMESİ YANLIŞ YAZILMIŞ!	Kelimeler
' Anabilim Dalı ' KELİMESİ YANLIŞ YAZILMIŞ!	Kelimeler
' OYBİRLİĞİ ' KELİMESİ YANLIŞ YAZILMIŞ!	Kelimeler
' tekrardan ' KELİMESİ YANLIŞ YAZILMIŞ!	Kelimeler
Geri Dön	

Burada Hata Sayfasında ekrana verilen değerlerin ayrıntılı çıktısı gösterilmektedir. Yine bu sayfadan “Geri Dön” butonu aracılığıyla hata sayfasına geri dönülebilir.

3.4.3 İşlevsel Gereksinimler

İşlevsel gereksinimler, gerçekleştirilmesi gereken gerekli görev, eylem veya etkinliği belirleyerek neler yapılması gerektiğini açıklar. Fonksiyonel ihtiyaç analizi için üst düzey fonksiyonlar olarak kullanılacaktır.

****GEREKSİNİM YANINDA GEREKSİNİM KODU YAZMAKTADIR.**

- ✓ Kullanıcı belgeyi “.pdf” uzantılı olarak girmeli. –g1
- ✓ Kullanıcı yalnızca Fırat Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına göre yazılmış tezi girmelidir. –g2
- ✓ Kullanıcı tez başlıklarını yazım klavuzuna göre girmelidir. –g3
- ✓ Kullanıcı yazılımı Windows 8 ve üzeri işletim sisteminde kullanmalı. –g4
- ✓ Kullanıcı yazılımı ilk örnek tez ile test etmeli.. –g5

3.4.4 Performans Gereksinimleri

Varolan sistemi kullanabilmek için gerekli olan Windows işletim sistemi sürümü minimum Windows 8 olmalıdır. Sistem aynı anda bir tezi işleyebilir. Ortalama Kaynak kullanımı; RAM:70 MB kullanılan ram yapılan işlemlere göre değişkenlik gösterebilir.

3.4.5 Fiziksel Çevre

Projemiz Windows tabanlı işletim sistemine sahip bilgisayarlarda geliştirilecektir. Kullanım aygıtları ise bu işletim sistemine ait bilgisayarlar ve tabletlerdir. Kullanıcılar bu cihazlara uygulamaları yükleyerek kolayca kullanabilir.

3.4.6 Arayüzler (g8)

Sisteme gelen girdiler aynı anda tek bir cihazdan gelebilir, çıktılar da yine aynı anda tek bir cihaz için gönderilebilir. Bunlar göz önüne alınarak uygulama gerekli durumlara cevap verebilecek şekilde geliştirilecektir. Uygulama arayüz tasarımı renkleri anket sonucunda müşterinin istediği renk ve iş analisti tavsiyesine göre yapılır. Uygulam arayüzü sade ve anlaşılırdır.

3.4.7 İşlevsellik(g9)

Sistem aynı tipteki kullanıcılar için ortak bir ağ oluşturacaktır. Uygulamamız offline bir uygulama olduğundan sürekli açık kalmasına ya da herhangi bir internet bağlantısına gerek duymayan bir yazılımdır. Gerekli güncellemeler uygulama dağıtıcısı tarafından yapılır. Uygulamayı kullananlara güncelleme bildirimi mail ve sms üzerinde yapılır ve güncellemeler gönderilir.

3.4.8 Veri(g7)

Kullanıcı verileri yazılımımızda asla tutulmaz. Bunun yanında kullanıcıların paylaştıkları belgelerin (docx,pdf) verileri kendi sistemlerinde bulunan donanımlarda tutulur. Bu veriler sorumluluğu tamamıyla kullanıcıya aittir. Yazılımımızda asla bir kişisel veri bulunmamaktadır.

3.4.9 Kaynaklar

Uygulama Masaüstü ortamı için tasarlanmıştır. Windows işletim sistemi için tasarlanmıştır. Gerektiği takdirde proje planlamasındaki personel sayısı artırılacaktır. Proje zaman çizelgesi projenin planlama aşamasında belirtilmiştir.

3.4.10 Güvenlik(g6)

Yazılımımızda bir girdi ekranı olmadığı ve sadece PDF belgelerine izin verildiği için sistem güvenliğini ihlal edecek herhangi bir fonksiyon bulunmamaktadır. Yazılımımızı çalıştırmak için Admin ya da Root yetkileri gerekmemektedir. Yazılımımız herhangi bir hafıza birimine çalışma esnasında kullanmaz. Program ve döküman dosyası geçici hafıza birimi olan RAM tutulur. Ve program kapandığında ya da durdurulduğunda silinir.

3.4.11 Kalite Güvencesi

Sistemdeki hatalar geliştiriciler tarafından güncellemeler ile giderilecektir. Bu güncellemeler sadece hata gidermek için değil uygulamaya yeni özelliklerde eklemek için de olacaktır. Bunun yanında kullanıcılar tarafından geribildirimler alınacaktır.

3.5 Kalite Özellikleri

3.5.1 Güvenilirlik (“Reliability”)

Sistemde tüm olası hatalara karşılık try-catch blokları kullanılmıştır. Geliştirme esnasında test işlemi yapılmıştır.

3.5.2 Kullanılabilirlik (“Availability”)

Uygulama dağıtıcısına ulaşarak yazılımı edinebilirsiniz. Uygulama indirilerek kullanılabılır. Uygulama arayüzü yukarıda verilmiştir. Basit anlaşılır bir arayüze sahip olduğundan rahatlıkla kullanılabılır.

3.5.3 Güvenlik

Uygulamamız herhangi bir kişisel veri tutmadığı için ya da herhangi bir donanıma direk yetkisi olmadığı için güvenlik gerekmemektedir. Bu aşamada tüm sorumluluk kullanıcıya aittir.

3.5.4 Taşınabilirlik (“Portability”)

Uygulama Windows Tabanlı işletim sistemlerde çalışacak şekilde tasarlanmıştır. Linux sistemlerde (Ubuntu 20.04 LTS & Arch Linux 5.14.2)Wine programı aracılığıyla uygulamamız yine çalışabilmektedir.

3.5.5 Kullanılabilirlik (“Usability”)

Windows sisteme sahip herkes uygulamayı indirip kullanabilir. Uygulamada anlamadıkları noktalar uygulama dağıtıcısının programın açılışına koyduğu mail ile iletişime geçebilirler.

3.6 GEREKSİNİMLERİN ÖNCELİĞİ ve GEREKSİNİM TABLOSU:

Gereksinim Id	Gereksinim Açıklama	İşlevsellik(Çekirdek/Yardımcı)	Mantıksal Grup	Paydaşlar	Girdiler (E/H/tanımsız)			Çıktılar(E/H/tanımsız)		Proses Adımları tanımlanmış mı?	Fizibilite (E/H)			Öncelik
					Tanımlı mı?	Biçimler/ ÖrneklerMevcut Belgeler	Veri Doğrulama Tanımlı mı?	Tanımlı mı?	Biçimler/Örnek belgelir tanımlı mı?		Teknik	Finansal	Zaman Çizelgesi	
g1		Ç	Veri	K	E	E	E	E	E	E	E	H	H	1
g2		Ç	Veri	K	E	E	E	E	E	E	E	H	H	2
g3		Y		K	E	E	E	E	E	E	E	H	H	8
g4		Ç		K	E	E	E	E	E	E	E	H	H	7
g5		Y		K	E	E	E	E	E	E	E	H	H	9
g6		Ç	Güv.	K-E	E	E	E	E	E	H	E	H	H	3
g7		Y	Veri	K	E	E	E	E	E	E	E	E	H	4
g8		Y	Aray	K-E	E	E	E	E	E	E	E	E		6
g9		Y	İşlev.	K-E	E	E	E	E	E	E	E	E	E	5

****GEREKSİNİM AÇIKLAMALARI ÜST KISIMDA YAPILMIŞTIR.**

****PAYDAŞLAR BÖLÜMÜ İÇİN K:KULLANICI,E:EKİP**

****MANTIKSAL GRUP:GÜV:GÜVENİLİK-ARAY:ARAYÜZ**

4. SİSTEM TASARIMI

Öncelikle sistemi bir açıklayalım. Sistemimiz yalnızca tez hazırlama durumda olan kullanıcıların etkileşime geçeceği bir kontrol uygulamasıdır. Kullanıcılar tek tip olup girdi olarak sadece belge yolu belirteceklerdir. Bunun dışında ise çıktı olarak okunabilir bir ekran gösterilmektedir. Ve programdan çıkmadan başka bir belgeyi analiz etmeye olanak sağlanılmıştır. Uygulamaya öncelikle işlenilecek pdf yolu girilir. Eğer girilen pdf yolunda ki belge tezin kontrol edilmesi için yeterli kriterlere (BEYAN, ÖNSÖZ, İÇİNDEKİLER, KAYNAKLAR vb.) sahipse program bir sonraki aşama olan kontrol aşamasına geçer. Burada tez içerisinde ki her bölüm için verilen kurallar kapsamında titizlikle kontroller gerçekleştirir.

Genel Kontrol; Bu kısım tez ilk seçildiğinde otomatik gerçekleştiren tez içerisinde İÇİNDEKİLER – ÖNSÖZ – ÖZET – BEYAN – ŞEKİLLER LİSTESİ – TABLOLAR LİSTESİ – EKLER LİSTESİ – KAYNAKLAR – SİMGELER VE KISALTMALAR bu başlıkların tamamı kontrol edilir. Eksik kısımlar için uyarı oluşturulur.

İçindekiler Kontrolü; Bu kısımda iligli başlığın belirtilen sayfa numarası içerisinde olup olmadığı kontrol edilir.(romen rakamları dahildir)

Şekiller Listesi Kontrolü; Bu kısımda şekiller listesinde belirtilen şekillerin, şekiller listesi hariç tez içerisinde kullanılıp kullanılmadığı kontrol edilir.

Tablolar Listesi Kontrolü; Bu kısımda tablolar listesinde belirtilen tabloların, tablolar listesi hariç tez içerisinde kullanılıp kullanılmadığı kontrol edilir.

EKLER Listesi Kontrolü; Bu kısımda ekler listesinde belirtilen eklerin, ekler listesi hariç tez içerisinde kullanılıp kullanılmadığı kontrol edilir.

Kaynaklar Kontrolü; Kaynaklar kısmında belirtilen kaynakların, tez içerisindeki kontrolleri yapılır. Burada blok referanslar(xx-xx), çoklu referanslar(xx, xx, xx) şeklinde tamamı kontrol edilir. Blok atıf için 3'ten fazla katnak varsa uyarı verilir, çoklu referansta 5'ten fazla kaynak varsa uyarı verilir.

Kelime Kontrolü; Türkçe'de en sık yanlış yazılan kelimelerden özel olarak oluşturulmuş bir wordlist bağlı olarak tez içerisindeki kelimelerin yazımı kontrol edilir. Yanlış yazan kelimeler raporlanır.

Tırnak Kontrolü; Burada tez içerisinde kullanılmış ("") tırnak sayısı belirtilir. Olası hatalara karşı önlem içindir.

Tüm kontroller için try-catch blokları yazılmış program hatalarından kaçınılmıştır. Program içerisinde Regular Expression(REGEX)'ler sıklıkla kullanılmıştır.

Bu uygulama Fırat Üniversitesi'nde uygulanacak, bu yüzden **Fırat Üniversitesi Fen Bilimleri Enstitüsü'ne ait Tez Yazım Kılavuzunda** ki kurallara göre hazırlanmıştır.

4.1 Kullanıcı Arabirimleri

Kullanıcıdan beklenen ve yazılımımız için dışardan girilecek veri yoludur. Örnek bir veri yolu "C:\Users\testUser\Documents\README.pdf". Bunun dışında kullanıcı ile program arasında bir etkileşim bulunmamaktadır. Kullanıcı girişi bu arabirimde amaçlanmıştır. Kullanıcı bilgilerini girerek bu sisteme giriş yapmış olacaktır.

4.2 Testler

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

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

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

4.3 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.

5. GERÇEKLEŞTİRME

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 gerekir.

5.1 Yazılım Geliştirme Ortamları

Yazılım geliştirme ortamı, projenin bilgisayar ortamında yapımını kapsar. Proje Windows 10 işletim sisteminde geliştirilecektir.

5.2 Programlama Dilleri

Programlama dili olarak C# kullanılacaktır. Bunun sebebi diğer dillerde .pdf uzantılı dosyaların saf veri haline çevrilmesi sürecinde hatalar ya da eksik çıktılar almamızdan ve C# programlama dilinin gerek hızlı derlenmesi gerek masaüstü program geliştirme gücünden dolayı seçtik. Ve seçimimizin güçlü bir seçenek olduğu gayet açıktır.

5.3 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. Bu aşamada yapılabilecek en iyi durum öncelikle programın son sürümü kullanıldığından emin olunduktan sonra mail ile uygulama dağıtıcısıyla iletişimi geçmektir.

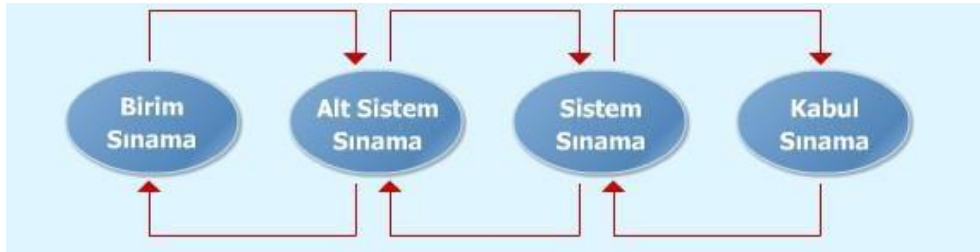
5.4 Olağandışı Durum Tanımları

Olağandışı gelişen durumlarda try-catch blokları devreye girecek ve program kırılmadan çalışmasına devam edebilecek şekilde tasarladık. Bu sayede alınan hatalarda açılan dosyaya zarar verilmeden güvenli bir şekilde kapatılıp olası bir veri kaybını önleyebiliyoruz. Olası bir olağan dışı durum ise verilen pdf'lerin içerisine makro olarak verilen zararlı yazılımlardır. Yine bu durum için aldığımız önlem ise dosyayı **sadece okuma** yetkisi ile açıyoruz. Bu sayede yeni çıkacak birçok zero day zafiyetinde korunmuş oluyoruz.

6.DOĞRULAMA GEÇERLEME

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

Yazılım belirtilerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtileri 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. Yazılım ürünlerinin tüm uygulanabilir gerekleri sağladığının gerçekleşmesi için sınamaların hazırlanıp yürütülmesi biçiminde özetlenebilir.



6.1 Alt Sistem Sınama: Birimlerin birleşmesiyle modüller oluşturulup bunların kendi içinde sinaması

yapıldı. Genel olarak arayüzde ki eksiklikler giderildi. Dosya okuma fonksiyonu kontrol edildikten sonra ayrıştırma modulu ve regex arama sorguları girildi. Ve program fonksiyonlara ayrılarak kontrol edildi.

6.2 Sistem Sınama: Sistemin bütün olarak sınanması yapıldı ve programın eksiksiz olduğu onaylandı. Program .pdf uzantılı dosyaları okuyabiliyor. Try-Catch blokları sorunsuz çalışıyor.

6.3 Kabul Sınama: Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha onaylandı. Fırat Üniversitesi Fen Bilimleri Enstitüsüne ait Örnek_TEZ.pdf uzantılı belge taratıldı ve program bir kez daha onaylandı.

6.4 Sınama Yöntemleri

Sınama işlemi, geliştirmeyi izleyen bir düzeltme görevi olmak ile sınırlı değildir. Bir "sonra" operasyonu olmaktan çok, geliştirme öncesinde planlanan ve tasarımı yapılması gereken bir çaba türüdür.

6.4.1 Kara Kutu Testi

Kara kutu test tekniği, test edilen yazılımın davranışının uygulamanın kod yapısına ve kodun nasıl işlediğine bakılmaksızın uygulanır. Bu test tekniğinde tamamen uygulamanın girdileri ile çıktılarına yoğunlaşırlar ve uygulamanın çıktıları girdilerden nasıl elde ettiğine ilişkin durumlar göz ardı edilir. Kara kutu test tekniğinde uygulama bizim için bir kara kutudur. Uygulamanın içerisini yani yapısını, mimarisini ve işleri nasıl ele aldığı ile ilgilenmeyiz. Bu test tekniği tamamen özellikler ve gereksinimler üzerine inşa edilir. Bu test tekniğinde kara kutu olan uygulamamızın gereksinimleri ve özellikleri bir şekilde yerine getirmesi beklenir. Yapılan kara kutu testinde uygulamamızda herhangi bir hata çıkmamıştır.



6.4.2 Beyaz Kutu Testi

Bu test tekniğinin en genel tarifi kodun kendisinin test edilmesidir. Test eden kişi uygulamaya ait girdi-çıkıtı, kod tasarımı ve kodun doğru çalıştığını kontrol eder. Bu test tekniğini yapmak için kodlama bilgisinin iyi seviyede olması gerekir. Çünkü test arayüz seviyesinde değil, doğrudan doğruya kod seviyesinde yapılacaktır. Yapılan beyaz kutu testinde kodda bir hataya rastlanmamıştır.



7.BAKIM

Sistem bakımında uygulamada belirlenen **eksiklikler** veya **eklenecek** yeni özellikler uygulamaya gelecek güncellemelerle eklenecektir. Bakım bölümüne ilişkin yapılan açıklamalarda **IEEE 1219-1998** standardı baz olarak alınmıştır. **Yerinde Ekibi** adında oluşturulacak bir ekip yazılımda meydana gelebilecek olası hatalara karşı hazır bulunacaktır.

7.1 KURULUM

Kurulum .msi uzantılı bir dosya olarak sadece Windows İşletim sistemine ait bir bilgisayara kurabilmektedir. Burada kurulum ekranında dikkat edilmesi gerekenler dosyanın kurulacağı klasör. Ve daha sonra kurulum için izin verildikten sonra kurulum güvenli bir şekilde tamamlanacaktır.

8.SONUÇ

Projemizin planlaması ile iş-zaman planı, personel sayısı, kullanılacak metodolojiler, maliyet kestirimi, yazılım-donanım-insan kaynakları belirlenmiştir. Çözümleme aşamasında projenin gereksinimleri toplamış, user-storyler belirlenmiş, gerekli diyagramlar çizilmiş, proje kalite özellikleri açıklanmıştır. Gereksinim verilerini toplamak için anket ve sorma yöntemi uygun görülmüş, belirlenen gereksimlere göre Kullanıcı Arayüz Prototipi tasarlanmış ve gereksinimler önceliklendirilerek tablosu çizilmiştir. Sonraki aşamalar da yapılacaklar için bir taslak oluşturulmuştur. Bunların yanında, tezlerin hazırlama sürecinde sadece konuya odaklanıp kuralların kontrolünü TKS (Tez Kontrol Sistemine) yaptırmak. Hem zaman hem de motivasyon kaybını büyük bir oranda engellemektedir. Bu sayede daha detaylı ve daha güzel içeriklerin ortaya çıkacağı şüphesiz bir gerçektir. Günümüzde buna benzer uygulamalar bulunmakta ama bunun kadar detaylı bir uygulama bulunmamaktadır. Bu yüzden gerekli her geri dönüş bizim ve uygulamamızın gelişimi açısından çok önemlidir.

