

T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

Proje Dokümantasyonu

Tez Şablon Kontrolü

Proje Ekibi

Abdullah BEKÇİ

Serdar ARIKAN

OCAK – 2021

1. GİRİŞ	
1.1 Projenin Amacı.....	05
1.2 Projenin Kapsamı.....	05
1.3 Tanımlamalar ve Kısaltmalar.....	05
2. PROJE PLANI	
2.1 Giriş.....	06
2.2 Projenin Plan Kapsamı.....	06
2.3 Proje Zaman-İş Planı.....	07
2.4 Proje Ekip Yapısı.....	07
2.5 Önerilen Sistemin Teknik Tanımları.....	08
2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları.....	08
2.7 Proje Standartları, Yöntem ve Metodolojiler.....	08
2.8 Kalite Sağlama Planı.....	09
2.9 Konfigürasyon Yönetim Planı.....	09
2.10 Kaynak Yönetim Planı.....	09
2.11 Eğitim Planı.....	09
2.12 Test Planı	10
2.13 Bakım Planı.....	10
2.14 Projede Kullanılan Yazılım/Donanım Araçlar.....	10
3. SİSTEM ÇÖZÜMLEME	
3.1 Mevcut Sistem İncelemesi	11
3.1.1 Örgüt Yapısı	11
3.1.2 Varolan Yazılım/Donanım Kaynakları.....	11
3.1.3 Varolan Sistemin Değerlendirilmesi.....	11
3.2 Gereksenen Sistemin Mantıksal Modeli	11
3.2.1 Genel Bakış.....	11
3.2.2 Veri Sözlüğü.....	11
3.2.3 İşlevlerin Sıradüzeni.....	11
3.3 Arayüz (Modül) Gerekleri	12
3.3.1 Yazılım Arayüzü.....	12
3.3.2 Kullanıcı Arayüzü.....	12
3.3.3 Yönetim Arayüzü.....	12
3.4 Belgeleme Gerekleri	12
3.4.1 Geliştirme Sürecinin Belgelenmesi.....	12

3.4.2	Eğitim Belgeleri.....	12
3.4.3	Kullanıcı El Kitapları.....	12
4. SİSTEM TASARIMI		
4.1	Genel Tasarım Bilgileri	13
4.1.1	Varsayımlar ve Kısıtlamalar.....	13
4.1.2	Dış Arabirimler.....	13
4.1.3	Testler.....	13
4.1.4	Performans.....	13
4.2	Süreç Tasarımı.....	13
4.2.1	Modüller.....	13
4.2.1.1	Ana Modül.....	13
4.2.1.1.1	İşlev.....	13
4.2.1.1.2	Modül Tanımı.....	14
4.2.1.1.3	Modül iç Tasarımı.....	14
4.2.2	Kullanıcı Profilleri.....	14
4.2.3	Entegrasyon ve Test Gereksinimleri.....	14
5. SİSTEM GERÇEKLEŞTİRİMİ		
5.1	Giriş.....	15
5.2	Yazılım Geliştirme Ortamları.....	15
5.2.1	Programlama Dilleri.....	15
5.3	Kodlama Stili.....	15
5.3.1	Açıklama Satırları.....	15
5.3.2	Anlamlı İsimlendirme.....	15
5.4	Kod Gözden Kaçırma.....	15
6. DOĞRULAMA ve GEÇERLEME		
6.1	Giriş.....	16
6.2	Sınama Kavramları.....	16
6.3	Sınama Yöntemleri.....	16
6.3.1	Beyaz Kutu Sınaması.....	16
6.3.2	Temel Yollar Sınaması.....	16
6.4	Sınama ve Bütünleştirme Stratejileri.....	17
6.5	Yaşam Döngüsü Boyunca Sınama Etkinlikleri.....	17
7. Bakım		

7.1 Kurulum.....	18
7.2 Yerinde Destek Operasyonu.....	18
7.3 Yazılım Bakımı.....	18

8. Sonuç

9. Kaynakça

1. Giriş

1.1 Projenin Amacı

Üniversitemizde yer alan bölümlerin birçoğunda olduğu gibi Yazılım Mühendisliği bölümünü bitirmek için de bir tez çalışması yapmamız gerekmektedir. Tez çalışmaları yapılırken belli bir şablon üzerine dokümante edilmelidir. Bu proje ile öğrenciler yazdıkları tezin, tez yazım kurallarına uyup uymadığını kontrol edebileceklerdir.

1.2 Projenin Kapsamı

Projenin ilk aşamada kapsamı bölümümüzde tez çalışması yapacak kişilerle sınırlıdır. İlerleyen zamanlarda proje bir web ortamına aktarılabilir ve Türkiye'nin her coğrafyasından tez çalışması yapacak öğrencilere yardımcı olabilir.

1.3 Tanımlamalar ve Kısaltmalar

- TEŞAK: Tez Şablon Kontrolü

2. Proje Planı

2.1 Giriş

Tez çalışması nihai niteleyici çalışmanın türlerinden biri olarak üniversitenin son yılında gerçekleştirilen, eğitim uzmanları, lisans ve yüksek lisans (master) programları kapsamında öğrenim gören öğrencilerin bağımsız yaratıcı çalışmasıdır. Genellikle doğal bilimler, insani, ekonomik ve yaratıcı uzmanlık alanlarında eğitim gören öğrenciler tarafından gerçekleştirilir ve mezunların özel teorik bilgi ve pratik becerilerini sistemleştirmeye, genelleştirmeye ve test etmeyi amaçlar.

Tez yazmanın temel amacı, mezunların teorik bilgilerini ve pratik becerilerini sistematize etmek ve yaygınlaştırmaktır.

2.2 Projenin Plan Kapsamı

Projenin plan kapsamında genel olarak öğrencilerin yaptıkları tez çalışması dahilinde oluşturdukları dokümanı kontrol etmesi gerekliliğinden dolayı oluşacak vakit kaybı yerine doğrudan spesifik olarak nerede hangi hatanın olduğunu söyleyen bir yazılım sayesinde öğrencilerin üzerindeki baskı ve stres azalma eğilimi gösterecektir.

- Böyle Bir Yazılım Neden Gerekli?

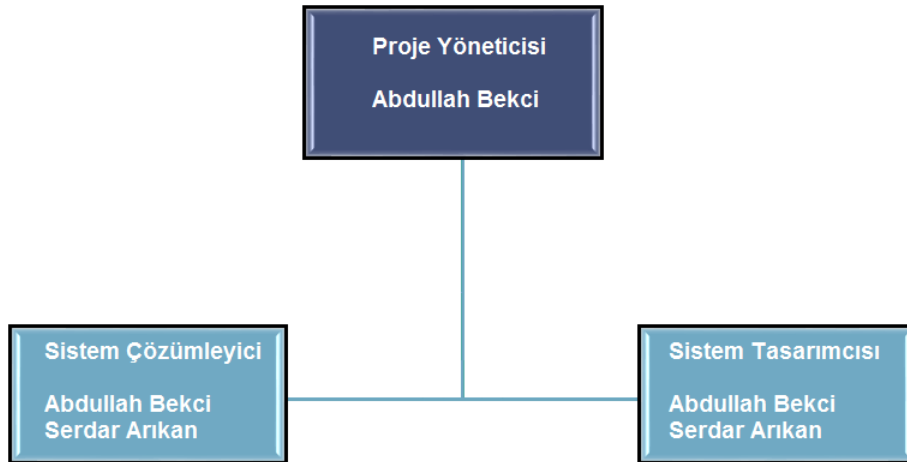
- Her an kullanılması çok kolaydır
- Öğrencilerin oluşturdukları dokümanı rahatlıkla kontrol etmesini sağlar
- Bu sayede öğrenciler yanlış yapma kaygısı gütmenden dokümanlarını yazabilirler
- Kaygı gütmedikleri için çok daha yaratıcı bir çalışma ortaya koyabilirler

Bu sebeple böyle bir yazılım çok faydalı olacaktır.

2.3 Proje Zaman-İş Planı

İş \ Zaman	1. Hafta	2. Hafta	3. Hafta	4. Hafta	5. Hafta	6. Hafta	7. Hafta	8. Hafta	9. Hafta
Ödev verilme tarihi	*								
Proje Planı		*	*						
Analiz			*	*					
Sistem Çözümleme				*	*				
Arayüz Tasarımı					*				
Gerçekleştirim					*	*	*	*	
Test								*	
Sunum									*

2.4 Proje Ekip Yapısı



2.5 Önerilen Sistemin Teknik Tanımları

Dosya Okuma Yöntemi

Bu yöntem sayesinde txt formatında, docx formatında, pdf veya başka formatlardaki içerikler okunabilmekte ve bu dosyalar üzerinde düzenleme yapılabilir.

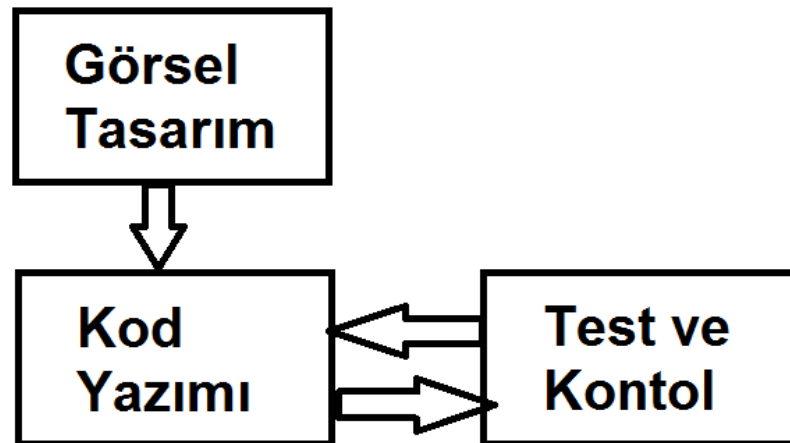
2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları

Çözümleme ve Tasarım Araçları	Programlama Araçları	Sınama Araçları	Destek Araçları
- Microsoft Paint - Microsoft Office Word 2010	- C#	- Microsoft Windows 7 - Microsoft Windows 10	- Google <u>Chrome</u>

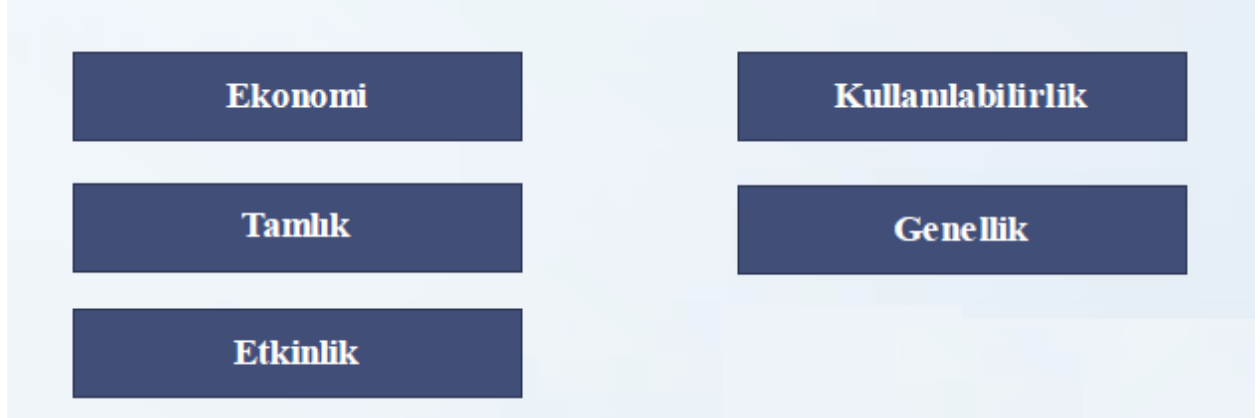
2.7 Proje Standartları Yöntem ve Metodolojiler

Projenin kapsamı oldukça düşük olduğu için spesifik olarak bir süreç modeli seçilmemiş olup, gelişi güzel yöntem dahilinde geliştirme yapılmıştır. Öncelikle problem anlaşılmış ve istekler belirlenmiştir. Belirlenen istekler doğrultusunda C# dili ile yazılımın gerçekleştirilmesine karar verilmiştir.

Öncelikle form ekranında görsel tasarım yapılmış olup, daha sonra bloklar halinde kodlama safhası başlamıştır. Farklı fonksiyon başlıkları altında farklı modüller gibi duran tek bir modül olarak proje hayata geçirilmiştir.



2.8 Kalite Saęlama Planı



Ekonomi: Belgele yönetimi, dosyalama ve yazışmak için kullanılan kaynakların elektronik olarak sağlandığı için maliyet azalmıştır.

Tamlık: Programda kullanılan bütün butonlar ve işlevler aktif durumda.

Etkinlik: Her sistem etkin bir biçimde kullanılır durumda.

Kullanılabilirlik: Kolay kullanımı ile her yaşa hitap eden bir sistem.

Genellik: Proje her üniversitede kullanılabileceğinden geneldir. Ve Türkiye genelinde kullanılacaktır.

2.9 Konfigürasyon Yönetim Planı

Yazılım kullanılmaya başlandıktan sonra görevlerini eksik veya yanlış yapması halinde atılabilecek adımlar düşünüldü.

2.10 Kaynak Yönetim Planı

Kaynak yönetiminde şu hususlar dikkate alınacaktır;

–Kalite beklendiği gibi mi?

–Proje elemanları yeni bir eğitim aldı mı?

–Yeni bir araç kullanılıyor mu?

–Yeterli vakit harcıyor mu?

2.11 Eğitim Planı

Projede kullanılacak programlama dili C# olup, bu dil özelinde dosya okuma ve string kütüphanesinin kullanımına hakim olmak gerekmektedir.

2.12 Test Planı

Proje çok teferruat içermediğinden dolayı test işlemleri performans ve doğruluk açısından test edilmektedir.

2.13 Bakım Planı

Proje bittikten sonra kullanım sırasında oluşabilecek problemleri gidermek veya yazılımın yetersiz geldiği durumlarda eklenecek modüller üzerine çalışma yapılabilecektir.

3. Sistem Çözümleme

3.1 Mevcut Sistem İncelemesi

3.1.1 Örgüt Yapısı

Örgüt yapısında öğrenci ve öğretim görevlisi bulunur. Öğrencinin hazırladığı tez programın verisini oluşturur. Öğretim görevlisi ise programı kullanarak tezin geçerli olup olmadığını kontrol eder.

3.1.2 Varolan Yazılım/Donanım Kaynakları

- C#
- Microsoft Office Word

3.1.3 Varolan Sistemin Değerlendirmesi

Sistem basit bir arayüzden oluşmaktadır. Öğretim görevlisi öğrenciden gelen belgeyi seçerek programa tanıtır ve kontrol et butonuna tıklar. Daha sonra program tezi değerlendirir ve istenilen standartlara uygun olup olmadığına dair bilgi verir.

3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Genel Bakış

Genel hatlarıyla bakacak olursak geliştirdiğimiz uygulama sayesinde uygun standartlarda tez yazmamış öğrenciler kolayca diğerlerinden ayırt edilebilecektir. Bu sayede bunlarla zaman kaybetmeyip daha kaliteli tezlere ağırlık verilebilecektir.

3.2.2 Veri Sözlüğü

Uygulamada kullanılan ana veri öğrencinin göndermiş olduğu tez belgesidir. Bu belge içerisindeki paragraflar cümleler hatta bazen karakterler ayrı ayrı verilere dönüştürülüp işlemler gerçekleştirilmektedir.

3.2.3 İşlemlerin Sıradüzeni

İlk olarak kullanıcı tez belgesini sisteme tanıtır. Daha sonra kontrol et butonuna tıklar. Butona tıklandıktan sonra uygulama arkaplanda tanımlı işlemleri yapar ve tezin belirtilen kurallara uygunluğunu kontrol eder. Bulduğu sonuçları çıktı olarak kullanıcıya sunar.

3.3 Arayüz (Modül) Gereklere

3.3.1 Yazılım Arayüzü

Uygulamada yazılım arayüzü bulunmamaktadır.

3.3.2 Kullanıcı Arayüzü

Kullanıcı uygulamaya girdiğinde karşısına 3 buton ve 1 textbox çıkacak. Butonlar tez belgesini sisteme tanıtmaya, belgeyi kontrol etmeye ve uygulamadan çıkamaya yarayacak. Textbox ise kontrol işlemi sonucunda elde edilen bilgileri kullanıcıya sunacak.

3.3.3 Yönetim Arayüzü

Uygulamamızda herhangi bir yönetici ihtiyacı olmadığı için böyle bir arayüze de ihtiyaç olmayacaktır.

3.4 Belgeleme Gereklere

3.4.1 Geliştirme Sürecinin Belgelendirilmesi

Geliştirme sürecinde projenin hangi aşamada olduğunu anlamak için belgelendirme yapıldı. Bu sayede kalınan yerden tereddütsüz şekilde geliştirmeye devam edildi böylece zaman kaybı yaşanmadı.

3.4.2 Eğitim Belgeleri

Eğitim belgesi bulunmamaktadır.

3.4.3 Kullanıcı El Kitapları

Sizlerin de takdir edeceği şekilde uygulamanın kullanımı basit ve anlaşılırdır. Bundan dolayı herhangi bir el kitabı veya uygulama kılavuzu hazırlanmayacaktır.

4. Sistem Tasarımı

4.1 Genel Tasarım Bilgileri

Tasarım yaparken dikkat ettiğimiz en önemli unsur basit olmasıydı. Kullanıcı tarafından rahat şekilde anlaşılmasını sağlamaya çalıştık. Bundan dolayı karmaşık iç içe sayfalar yerine tek bir sayfa üzerinde tüm işlemleri yaptırmaya çalıştık. Yaptığımız tasarımla da bunu başardığımızı düşünüyoruz.

4.1.1 Varsayımlar ve Kısıtlamalar

Varsayılan değer bulunmamaktadır. Yegane kısıtlamamız ise tezin Microsoft Office Word formatında hazırlanmış olmasıdır.

4.1.2 Dış Arabirimler

Uygulamada farklı arabirimler bulunmamaktadır. Sadece bir ekran bulunmakta ve tüm işlemler orada yapılmaktadır.

4.1.3 Testler

Uygulamayı geliştiriciler olarak her aşamada kendi sistemlerimizde test ettik. Oluşabilecek açıkları dikkat ederek değerlendirdik ve gerektiğinde müdahale ederek en güvenilir ve doğru sonuç verecek şekilde geliştirdik.

4.1.4 Performans

Uygulama gönderilen tez belgesini kontrol ederken tüm karakterleri incelediği için bu biraz zaman almaktadır. Yaklaşık 20-30 saniye civarında sonuç alınmaktadır.

4.2 Süreç Tasarımı

4.2.1 Modüller

4.2.1.1 Ana Modül

4.2.1.1.1 İşlev

Kullanıcının tüm işlemleri yapacağı modüldür.

4.2.1.1.2 Modül Tanımı

Modül gönderilen tez belgesi ve sonuçların gösterildiği bir textbox'tan meydana gelmektedir.

4.2.1.1.3 Modül İç Tasarımı

Modül arkaplanda tüm işlemleri yapmaktadır. Bunu yaparken tanımlanmış olan fonksiyonları kullanmaktadır. Bu fonksiyonların bazıları kelimeSay(), kaynakSay() ve en önemlisi oku() fonksiyonudur.

4.2.2 Kullanıcı Profilleri

Uygulamayı kullanmak için profil oluşturmaya gerek duyulmamaktadır. Bu sayede sadece öğretim görevlilerinin tezi teslim edildikten sonra kontrol etmesi değil aynı zamanda öğrencilerin de tezi göndermeden önce kontrol ederek varsa yanlışlarını düzeltmeleri ve daha sonra sıkıntı yaşamamaları hedeflenmiştir.

4.2.3 Entegrasyon ve Test Gereksinimleri

Uygulamanın kullanılabilmesi için tezin entegre edilmesi gerekmektedir. Bu Aç butonu sayesinde rahatlıkla yapılabilir.

5.Sistem Gerçekleştirimi

5.1 Giriş

Gerçekleştirim aşamasında tasarım aşamasında karar verdiğimiz işlemleri yazılım ortamına aktardık.

5.2 Yazılım Geliştirme Ortamları

Tararım sonucunda oluşan fiziksel modelin yazılı ortamına geçirilmesi için uygun ortamlar kullanılmalıdır. Bizim projemizde normalde kullanılan bir ortam olan veritabanı yönetim sistemi mevcut değil. Bunun sebebi projede veritabanı olmamasıdır.

5.2.1 Programlama Dilleri

Bizim bu projeyi gerçekleştirmek için seçtiğimiz dil C# dilidir. Bu dili kullanmamızın sebebi string sınıfının güçlü özelliklere sahip olmasıdır. Bu sayede Word dökümanı üzerinde daha fazla hakimiyet kurabildik.

5.3 Kodlama Stili

5.3.1 Açıklama Satırlar

Açıklama satırlarını gerekli gördüğümüz, anlaşılması zor olduğunu düşündüğümüz her satırın sonuna ekledik.

5.3.2 Anlamlı İsimlendirme

Programı kodlarken fonksiyonların isimlerinin yaptıkları işlemi barındırmasına dikkat ettik. Aynı şekilde değişkenleri isimlendirirken de kullanım alanlarıyla doğru orantılı isimler vermeye çalıştık.

5.4 Kod Gözden Geçirme

Hazırlanmış bir programın geliştiriciler tarafından gözden geçirilmeden kullanıma sunulması büyük aksaklıklara sebep olabilmektedir. Buna neden olmamak için projeyi geliştirenler olarak kendi yazdığımız kodları diğer arkadaşlara ileterek kontrol etmesini istedik. Bu kontrol sırasında dikkat ettiğimiz en önemli hususlar;

- Kod olması gerektiği gibi çalışıyor mu?
- Kod yazılırken isimlendirme kurallarına uyulmuş mu?
- Anlaşılması zor kısımlarda açıklama satırları kullanılmış mı?

6. Doğrulama ve Geçerleme

6.1 Giriş

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

- Yazılım belirtilmelerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtilmeleri tutarlı olarak betimler durumunda 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 belirtilmelerin önceki belirtilmelerle karşılaştırılması.

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

6.2 Sınama Kavramları

Sistem Sınama: Sistem bütün olarak sınıandı ve sorunsuz çalıştığı onaylandı.

Kabul Sınama: Sistem gerçek veriler ile sınıandı ve tekrar sorunsuz çalıştı.

6.3 Sınama Yöntemleri

6.3.1 Beyaz Kutu Sınaması

- Bütün döngüler alt ve üst sınır değerleriyle sınıandı.
- Bütün fonksiyonların işlevlerini gerektiği gibi yerine getirip getirmediği sınıandı.

6.3.2 Temel Yollar Sınaması

Sistemin tümüne yönelik işlevlerin doğru yürütüldüğünün testidir.

- Eş değerlere bölme
- Uç değerler analizi
- Karar tablosu
- Rastgele test

6.4 Sınama ve Bütünleştirme Stratejileri

Biz ekip olarak sınımamızı yaparken küçük parçalardan başlayıp sistemin tamamına yaymayı hedefledik. Bu amacımız doğrultusunda kodladığımız fonksiyonların hepsini ayrı ayrı sınadık. Daha sonra tüm fonksiyonların sınama işlemleri tamamlandıktan sonra tüm sistemi aynı anda sınavarak bütün olarak nasıl çalıştığını denedik.

6.5 Yaşam Döngüsü Boyunca Sınama Etkinlikleri

Yazılım yaşam döngüsünün her aşamasında sınamaların sürekli ve düzenli şekilde yapılması gerekir. Her aşamada gerçekleştirilen işlemler tekrar tekrar sınama aşamalarından geçirilir. Böylece projenin ileriki zamanlarında sistemde hata olması önceden önlem alınarak engellenmiş ve bu sayede hataların düzeltilmesi işlemleri daha az maliyet ve zamanda çözülmesi sağlanır.

7.Bakım

7.1 Kurulum

Uygulamayı çalıştırmak için herhangi bir kurulum yapmanızı gerektiren bir durum mevcut değildir. Tek yapmanız gereken proje dosyası içerisinde bulunan .exe uzantılı uygulamayı çalıştırıp kullanmaktır.

7.2 Yerde Destek Operasyonu

Kullanıcılar tarafından uygulamayı kullanmada sorun yaşanırsa bizzat geliştiricilere iletişim adreslerinden ulaşmaları halinde her türlü teknik destek sağlanacaktır. Yapılan geri dönüşlerle birlikte uygulamadaki gözden kaçan ayrıntılar da onarılıp mükemmeliyet sağlanacaktır.

7.3 Yazılım Bakımı

Yazılımımızın bakımı kullanıcılar tarafından gönderilen geri bildirimler ve geliştiriciler tarafından farkedilen hatalar doğrultusunda sürekli bir şekilde yapılacaktır. Kullanıcılar karşılaştıkları hataları iletişim adreslerimize göndererek bize yardımcı olup uygulamanın gelişmesine birinci elden katkı sağlayacaklardır.

8.Sonuç

Uygulama kullanıma açıldığı zaman gerek öğretim görevlileri gerekse öğrenciler tarafından sıklıkla kullanılacağını düşünüyoruz. Bunu düşünmemizdeki en önemli etken öğrencilerin hazırladıkları tez dosyasının kontrolünü kolay bir şekilde kendileri yaparak daha sonra ortaya çıkabilecek sorunları önceden düzeltmeleri ve bu sayede istenen özelliklerde tez dosyası göndererek reddedilme ihtimalini en aza indirecekleridir. Öğretim görevlileri tarafından bakarsak uygulama sayesinde sürekli yoğun bir şekilde çalışması gereken öğretim görevlileri bu uygulamayla beraber uzun süreli bir işi kısa zamanda hallederek kendilerine daha fazla zaman kazandırabilecek olmasıdır.

9.Kaynakça

- [1] https://stackoverflow.com/questions/28987095/get-pages-of-word-document?fbclid=IwAR13L9-okbiWOON29AQat2hhMDX0DJh5rA4ixHndJYfOyX5SBoHqHKdR7_c
- [2] <https://stackoverflow.com/questions/16690729/how-to-get-words-from-current-visible-page-of-ms-word-document>
- [3] <https://stackoverflow.com/questions/19035893/finding-second-occurrence-of-a-substring-in-a-string-in-java>
- [4] <https://docs.microsoft.com/en-us/dotnet/api/microsoft.office.interop.word?view=word-pia>
- [5] <https://docs.microsoft.com/tr-tr/dotnet/api/microsoft.office.interop.word.document.range?view=word-pia>