



FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği

YMH319 PROGRAMLAMA DİLLERİ

## **WORD KONTROL PROGRAMI**

**Proje Çalışma Grubu**

16542509 Harun KURT

175541018 Yusuf ÇELİK

## İÇİNDEKİLER

<b>1. TANITIM</b>	
1.1	Projenin Amacı .....5
1.2	Projenin Kapsamı..... 5
<b>2. PROJE PLANI</b>	
2.1	Gantt Diyagramı .....5
2.2	Proje Zaman-İş Planı .....6
2.3	Proje Ekip Yapısı ..... 6
2.4	Maliyet Hesaplamaları ..... 7
2.5	Kullanılan Özel Geliştirme Araçları ve Ortamları..... 11
2.6	Kalite Sağlam Planı ..... 12
2.7	Konfigürasyon Yönetim Planı ..... 13
2.8	Kaynak Yönetim Planı..... 13
2.9	Eğitim Planı ..... 13
2.10	Test Planı ..... 13
2.11	Bakım Planı..... 14
<b>3. ÇÖZÜMLEME</b>	
3.1	Mevcut Sistem Analizi..... 14
3.1.1	Use Case Diyagramı ve Sistemin İşleyiş Senaryosu ..... 14
3.1.2	Sistemin Avantajları ve Dezavantajları ..... 15
3.2	Önerilen Sistemin Mantıksal Modeli ..... 16
3.2.1	İşlevsel Model .....16
3.2.2	Bilgi Sistemleri/Nesneler .....18
3.2.3	Arayüzler ..... 19

## 4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri .....	24
4.1.1 Genel Sistem Tanımı .....	28
4.1.2 Sistem Mimarisi .....	28
4.1.3 Dış Arabimler .....	29
4.1.4 Veri Modeli .....	30
4.1.5 Testler .....	31
4.1.6 Performans .....	31
4.2 Süreç Tasarımı .....	32
4.2.1 Genel Tasarım .....	32
4.2.2 Modüller .....	32
4.2.2.1 İşlemler Modülü .....	32
4.2.2.1.1 İşlev .....	32
4.2.2.1.2 Kullanıcı Arabirimi .....	33
4.2.2.1.3 Modül Tanımı .....	33
4.2.2.1.4 Modül İç Tasarımı .....	33
4.2.2.2 Kullanıcı Modülü .....	34
4.2.3 Kullanıcı Profilleri .....	34
4.2.4 Entegrasyon ve Test Gereksinimleri .....	34
4.3 Ortak Alt Sistemlerin Tasarımı .....	35
4.3.1 Ortak Alt Sistemler .....	35
4.3.2 Modüller Arası Ortak Veriler .....	35
4.3.3 Güvenlik Altsistemi .....	35
4.3.4 Veri Dağıtım Altsistemi .....	35
4.3.5 Yedekleme ve Arşivleme İşlemleri .....	36

<b>5. SİSTEM GERÇEKLEŞTİRİMİ</b>	
5.1 Giriş.....	37
5.2 Yazılım Geliştirme Ortamları.....	37
5.2.1 Programlama Dilleri .....	37
5.2.2 Veri Tabanı Yönetim Sistemleri .....	38
5.2.2.1 Veri Modelleri .....	39
5.2.2.2 Şemalar.....	40
5.2.2.3 Veritabanı Dilleri ve Arabirimleri .....	41
5.2.2.4 Veritabanı Sistem Ortamı.....	41
5.2.2.5 VTYS'nin Sınıflandırılması .....	41
5.2.2.6 CASE Araç ve Ortamları.....	41
5.3 Kodlama Stili.....	41
5.3.1 Açıklama Satırları .....	42
5.3.2 Kod Biçimlemesi .....	42
5.3.3 Anlamlı İsimlendirme .....	42
5.3.4 Yapısal Programlama Yapıları.....	42
5.4 Program Karmaşıklığı .....	42
5.4.1 Programın Çizge Biçimine Dönüştürülmesi .....	42
5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama .....	43
5.5 Olağan Dışı Durum Çözümleme .....	43
5.5.1 Olağandışı Durum Tanımları .....	44
5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları .....	44
5.6 Kod Gözden Geçirme .....	44
5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi .....	44
5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular... ..	44
5.6.2.1 Öbek Arayüzü... ..	45
5.6.2.2 Giriş Açıklamaları .....	45
5.6.2.3 Veri Kullanımı.....	45
5.6.2.4 Öbeğin Düzenlenişi .....	45
5.6.2.5 Sunuş .....	46
<b>6. DOĞRULAMA VE GEÇERLEME</b>	
6.1 Giriş.....	46
6.2 Test Doğrulama Planı.....	46
6.3 Kullanılacak Test Araçları .....	49
<b>7. BAKIM</b>	
7.1 Giriş.....	50
7.2 Kurulum .....	51
7.3 Yerinde Destek Organizasyonu.....	51
7.4 Kurulum Entegrasyon Süreci .....	52
7.4.1 Bakım Süreç Modeli .....	52
<b>8. SONUÇ .....</b>	<b>56</b>
<b>9. KAYNAKLAR .....</b>	<b>57</b>

## 1. GİRİŞ

### 1.1 Projenin Amacı

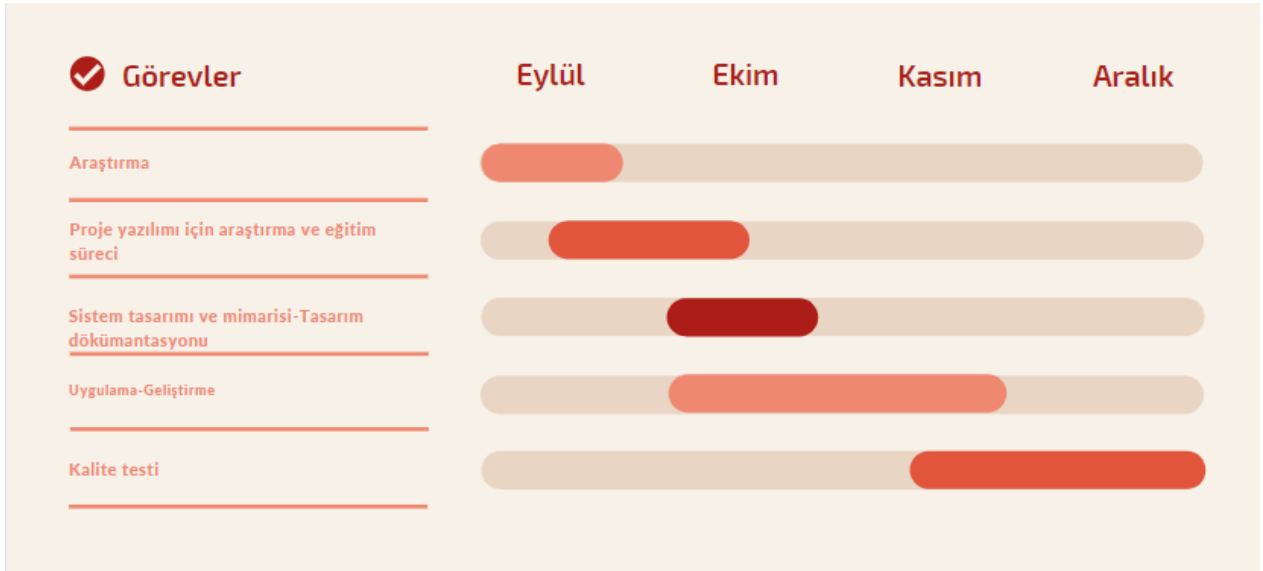
Projemiz Microsoft Word Programının içerisinde yer alan metinlerin analizini yapmayı sağlar

### 1.2 Projenin Kapsamı

Fatura ya da rapor oluşturulan programlarda daha detaylı olmasını sağlar.

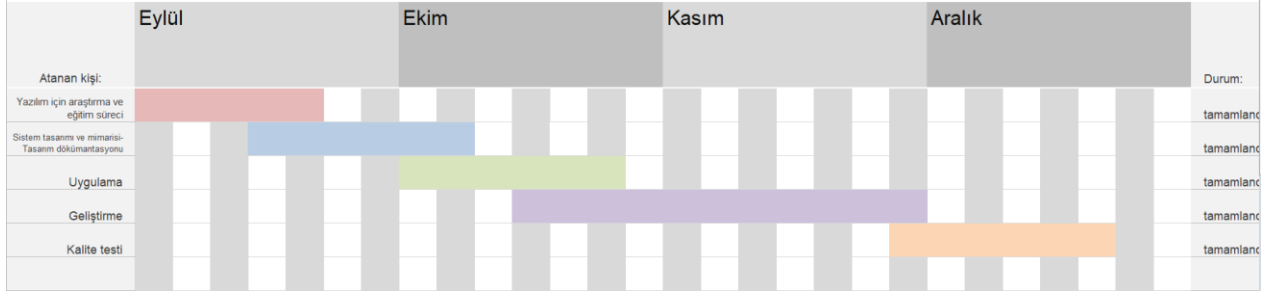
## 2. PROJE PLANI

### 2.1 Gantt Diyagramı



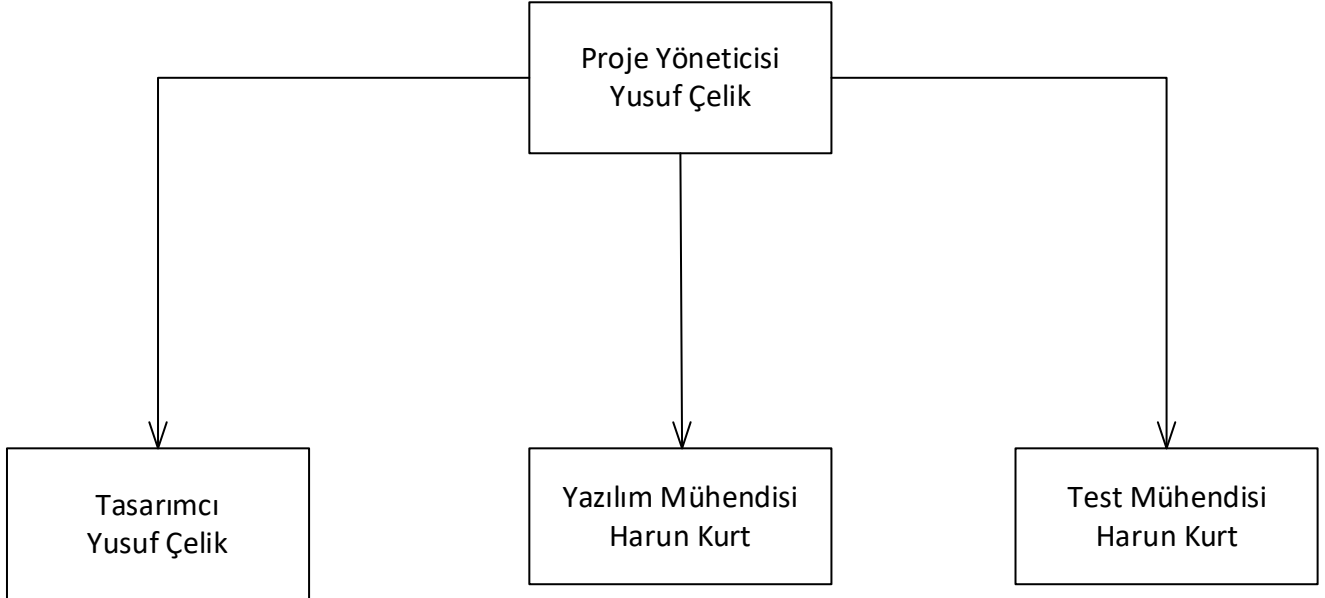
Şekil 2.1 Gantt Diyagramı

## 2.2 Proje Zaman-İş Planı



Şekil 2.2 Proje Zaman-İş Planı

## 2.3 Proje Ekip Yapısı



Şekil 2.3 Proje Ekip Yapısı Şeması

## 2.4 Maliyet Hesaplamaları

Ölçüm parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	1	3	3
Kullanıcı Çıktı Sayısı	2	4	8
Kullanıcı Sorgu Sayısı	3	5	15
Kütük Sayısı	12	10	120
Dışsal Arayüz Sayısı	3	10	30
Ana İşlev Nokta Sayısı			176

Teknik Karmaşıklık Sorusu	Puan
1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	5
2. Veri iletişimi gerekiyor mu?	5
3. Dağıtık işlem işlevleri var mı?	3
4. Performans kritik mi?	5
5. Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak?	1
6. Sistem, çevrimiçi veri girişi gerektiriyor mu?	5
7. Çevrimiçi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu?	4
8. Ana kütükler çevrimiçi olarak mı güncelleniyor?	5
9. Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı?	5
10. İçsel işlemler karmaşık mı?	3
11. Tasarlanacak kod, yeniden kullanılabilir mi olacak?	4
12. Dönüştürme ve kurulum, tasarımda dikkate alınacak mı?	4
13. Sistem birden çok yerde yerleşik farklı kurumlar için mi geliştiriliyor?	5
14. Tasarlanan uygulama kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak?	5
<b>TOPLAM</b>	<b>59</b>

0: Hiçbir Etkisi Yok    1:Çok Az etkisi Var    2:Etkisi Var    3:Ortalama Etkisi Var    4:Önemli Etkisi Var    5:Mutlaka Olmalı, Kaçınılamaz

### Etkin Maliyet Modeli - COCOMO

<b>İŞLEV NOKTASI=</b> $A \cdot N \cdot (0,65 \cdot 0,01 \cdot T_{KF})$	<b>İN=</b> $176 \times (0,65 \times 0,01 \times 59)$
<b>KSS=</b> $300,82 \times 30$ (BİN CİNSİNDEN)	Satır Sayısı= <b>79.6</b> Yaklaşık 80 satır

#### AYRIK PROJE SEÇİMİ

$A=2,4$  , $B=1,05$ ,  $C=2,5$   $D=0,35$

#### AYLIK KİŞİ BAŞI İŞ GÜCÜ

$E=A \times (KSS)^B$

$E=2,4 \times (9,025)^{1.05} = \mathbf{24.17}$

#### GELİŞTİRME SÜRESİ

$D=2,5 \times (24,17)^{0,38} = \mathbf{4 AY}$

#### TAHMİNİ ÇALIŞAN SAYISI

$24.17/4 = \mathbf{6,04}$  (YAKLAŞIK 6 KİŞİ)

Microsoft Office Professional 2019 – 3299,99 ₺ / YIL

## Office Profesyonel 2019

Microsoft Corporation



3.299,99 ₺

Şimdi satın al

Microsoft Visio Professional 2019 – 3899,99 ₺ / YIL





## Visio Professional 2019

Microsoft Corporation

3.899,99 ₺

Şimdi satın al

2 x MSI Prestige

29.809,12 ₺



MSI Prestige 15 A11SCS-035TR Intel Core i7 1185G7 16GB  
512GB SSD GTX 1650Ti Windows 10 Home 15.6" FHD  
Taşınabilir Bilgisayar

MSI

Notebook Alana Görüntülü Akademi Hediye!

%11  
indirim

16.769,00 TL

14.904,56 TL

5,0 ★★★★★

5 Değerlendirme

- 1.341,41 TL x 12 ay'a varan **Taksit** seçenekleri
- 1.401,42 TL x 12 ay'a varan **Alışveriş Kredisi** seçenekleri

Satıcı: **Hepsiburada** | Kampanyaları Gör 2

- 1 +  
Adet

Sepete Ekle

- 21 saat 23 dakika içinde sipariş verirsiniz yarın kargoda

24 saatte  
Kargoda

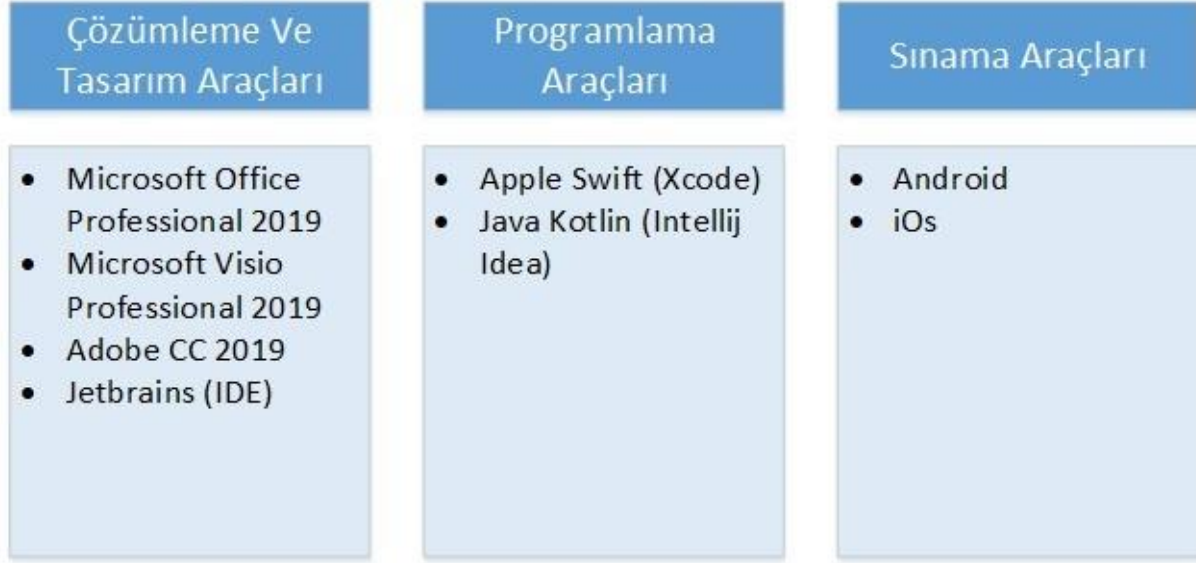
Çoklu kredi  
kartı ile  
ödeme

Alışveriş  
Kredisi  
Seçeneği

Genel Toplam= 229.009,1₺

Mühendis Maaşları (6 kişi) – Aylık  
8000₺  
4 ay toplam=192000₺

## 2.5 Kullanılan Özel Geliştirme Araçları ve Ortamları



Şekil 2.5 Kullanılan Özel Geliştirme Araçları ve Ortamları Şeması

## 2.6 Kalite Sağlama Planı



## Şekil 2.6 Kalite Sağlama Planı

1. **Bütünlük:** Yönetici sitenin tüm işlevlerine hakim olacak ve site bir bütün halinde çalışacaktır.
2. **Güvenilirlik:** Sitede bulunan bütün banka verileri bankaların güncel verileri ile eş zamanlı siteye aktarılacak böylece kullanıcılar güvenli bir şekilde işlemlerini gerçekleştirebilecek.
3. **Belgeleme:** Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olacak bu doküman oluşturulur.
4. **Ekonomi:** Uygulamanın tasarımı aşırı düzeyde bir maliyet göstermeyeceğinden bütçe dostu bir proje olacaktır.
5. **Anlaşılabilirlik:** Siteyi ziyaret eden herhangi bir kullanıcı tarafından kolayca anlaşılabilir olması sağlanacaktır.
6. **Estetiklik:** Site günümüz internet dünyasına en uygun biçimde tasarlanarak yenilikçi ve modern bir görünüme sahip olacaktır.
7. **Etkinlik:** Kullanıcı sitenin her alanına hakim olduğu için siteyi etkin bir biçimde kullanacak.
8. **Tamlık:** Sitede herhangi bir açık olmamalı ve sitede bulunan tüm uygulama işlevleri vs. çalışır ve tamdır.
9. **Esneklik:** Proje internet üzerinden çalışacağından gayet esnektir.
10. **Yeniden Kullanılabilirlik:** Uygulama her koşula rağmen yeniden düzenlenip kullanılabilir bir biçimde tasarlanacaktır.

## 2.7 Kaynak Yönetim Planı

Kişisel verilerin korunum kanununa göre uyarlanılmıştır.

**Kaynak yönetiminde şu hususlar dikkate alınacaktır;**

- Kalite beklenildiği gibi mi?
- Yeterli hata bulunuyor mu?
- Eğer yeterli sayıda hata bulunmuyorsa veya kalite beklenenden daha iyi görünüyorsa kalite gerçekten daha iyi olabilir.
- Proje elemanları yeni bir eğitim aldı mı?
- Yeni bir araç kullanılıyor mu?
- Yeterli vakit harcanıyor mu?
- İnceleme toplantılarında yeterli sürede vakit harcanıyor mu?

## 2.8 Eğitim Planı

Proje kapsamında alınması gereken eğitimler

- PYTHON 3,19 Eğitimi
- GİTHUB Kullanım Eğitimi.

Gerekten eğitimlerdir.

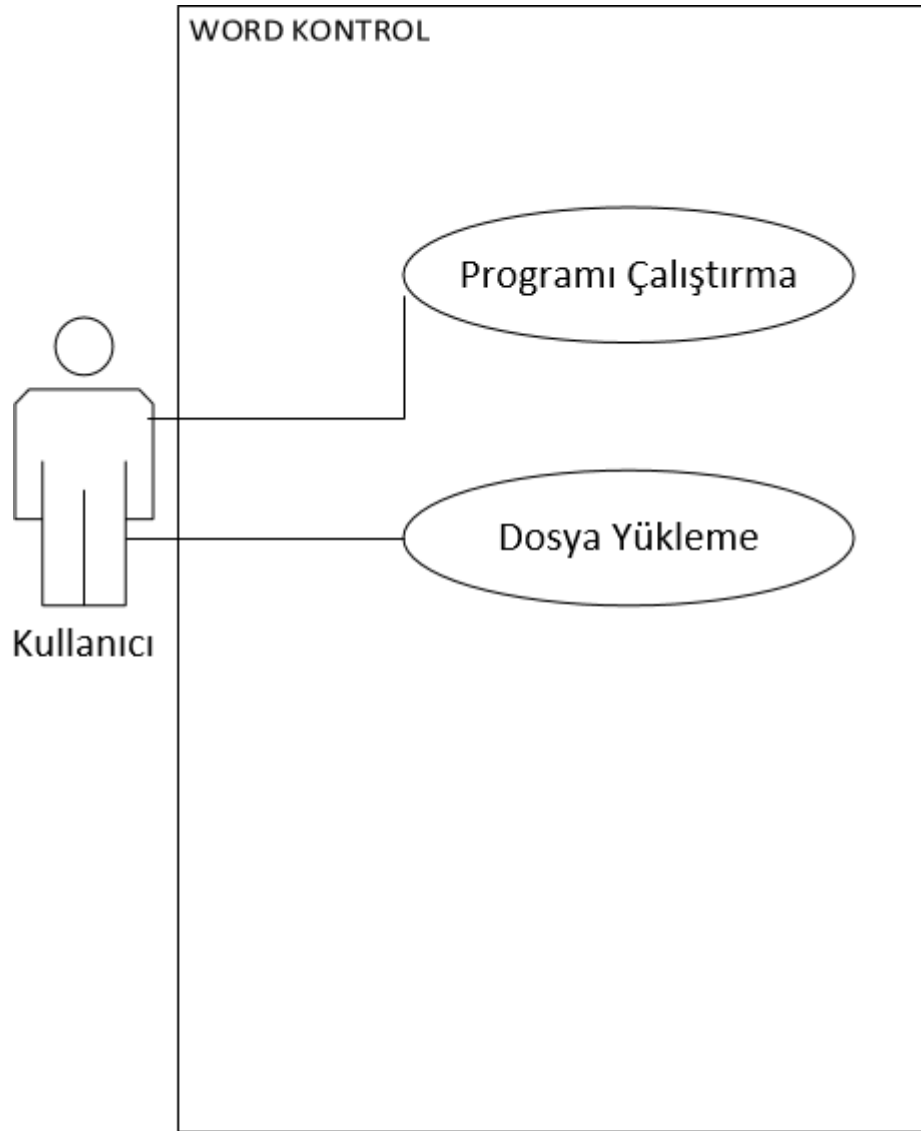
## **2.9 Test Planı**

Windows ve Mac işletim sistemleri üzerine optimize edilmiş uygulamamız, gerekli test ekipleri tarafından test edildikten sonra her türlü sorun çözülmüş olup, güncellemelerle 1.0 sürüme çıkış yapmıştır.

## **2.10 Bakım Planı**

Projenin bakım planında kullanıcılardan gelecek olan talepleri müşteri hizmetleri aracılığıyla veya uygulama üzerinden geri bildirim yapılması durumunda değerlendirilecektir.

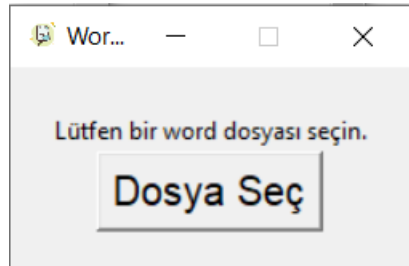
## **2.11 Use Case**



### 3.1 Mevcut Sistem Analizi

Fırat Üniversitesi Tez yazım kuralları çerçevesinde yazılan tezin doğrulunu kontrol eden bir program tasarladık.

#### 3.1.1 Arayüzler



Giriş ekranımızda kullanıcıya ayrıntılı verilerini istediğimiz Word dosyasını seçmesini istiyoruz.

```
*****  
*****  
*****  
İlk aşama Tamamlandı..  
İşlem Word dosyasına eklendi...  
İkinci aşama tamamlandı..  
Word Raporu Oluşturuldu..  
_
```

Kullanıcı Word dosyasını seçtikten sonra program çalıştırılır.

Programın adım adım yaptığı bölümler kullanıcıya gösterilir.

# WORD KONTROL RAPORU

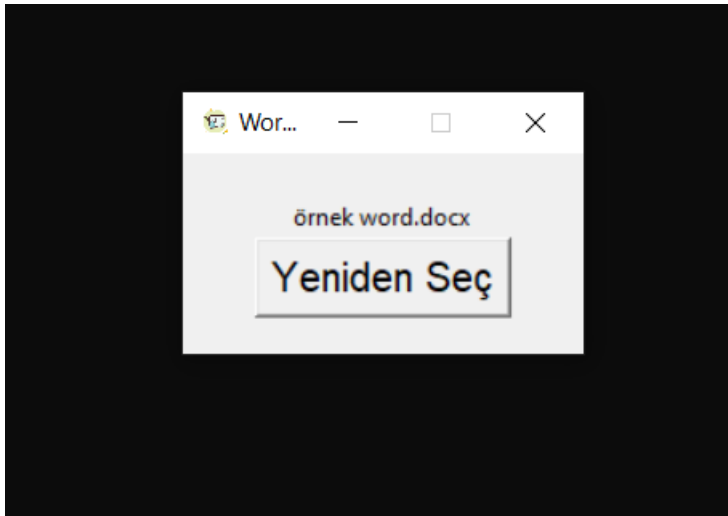
---

1. Cift tirnak arasinda kullanilan kelime adedi>50 degil
2. Giris bölümünün son bölümünde tezin organizasyonu ve kapsamına yer verilmemis

16542509-Harun Kurt

175541018-Yusuf Çelik

Programımız tarafından oluşan detaylı çıktımız bu şekildedir.



Program çıktı oluşturulduktan sonra kullanıcıya tekrar kullanmak istediği sorulur.



## 4. SİSTEM TASARIMI

### 4.1 Genel Tasarım Bilgiler

#### 4.1.1 Genel Sistem Tanımı

Kullanıcıdan alınan Word belgesi programımız tarafından incelendikten sonra işlem 1 uygulanıp RAPOR.docx adlı dosyaya kaydedilir. Daha sonra ise işlem 2 uygulanıp aynı dosyaya kaydedilir.

#### 4.1.2 Testler

Uygulama Github'tan ücretsiz sunulacağı için testler yazılım ekibimizde beta(tam) sürümü Github'tan açılacaktır.



Şekil 4.11 Uygulama Test Aşamaları

**Alfa Aşaması:** Uygulamanın geliştirildiği ortamda kullanıcıların katkıda bulunması uygulamayı test etmesi ile yapılacaktır.

**Beta Aşaması:** Uygulama Github'tan rahatlıkla erişilebilecektir.

#### 4.1.3 Performans

Uygulamanın performansını etkileyen faktörlerin test verileri değerlendirilecek

Uygulamanın tasarımına uygunluk performansı

Tasarımı yapılan sistemin stabilitesi ve işleyiş performansı değerlendirilecek

Veri yapısının sistemle performansı

## **4.2 Süreç Tasarımı**

### **4.2.1 Genel Tasarım**

Seçilen Word belgesinin detaylı raporu ve rapor oluşturma adımları verilir.

### **4.2.2 Modüller**

#### **4.2.2.1 İşlemler Modülü**

##### **4.2.2.1.1 İşlev**

Kullanıcıların uygulamada yapabileceği tüm işlemlere ulaşması için gerekli olan bir modüldür.

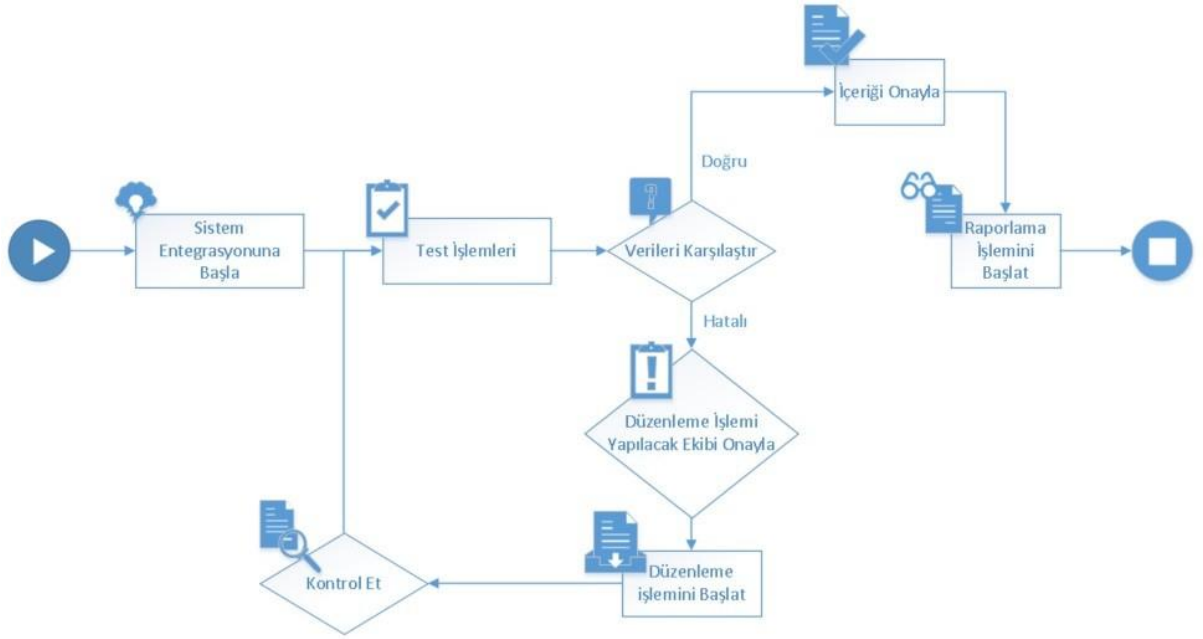
##### **4.2.2.1.2 Kullanıcı Arabirimi**

#### **4.2.2.2 Kullanıcı Modülü**

Kullanıcı Word belgesinin boyutu fark etmeksizin programımızdan detaylı rapor elde edebilir.

### 4.2.3 Entegrasyon ve Test Gereksinimleri

Sistemimizin daha öncede belirttiğimiz gibi devletin mevcut veri tabanlarıyla entegrasyon halinde olması gerekir.



Şekil 4.15 Entegrasyon Şeması

### 4.2.4 Güvenlik Alt Sistemi

Yazılım sistemlerinin güvenilirliğe ilişkin nicelikleri, kullanıcıların gereksinimlerini karşılayacak şekilde ortaya koymak ve güvenilirliğin hesaplanmasına yönelik verileri toplama, istatistiksel tahminleme, ölçütlerin tespiti, yazılıma ait mimari özelliklerin belirlenmesi, tasarım, geliştirme ve bunlara yönelik çalışma ortamının belirlenmesi ve modellenmesini kapsamaktadır.

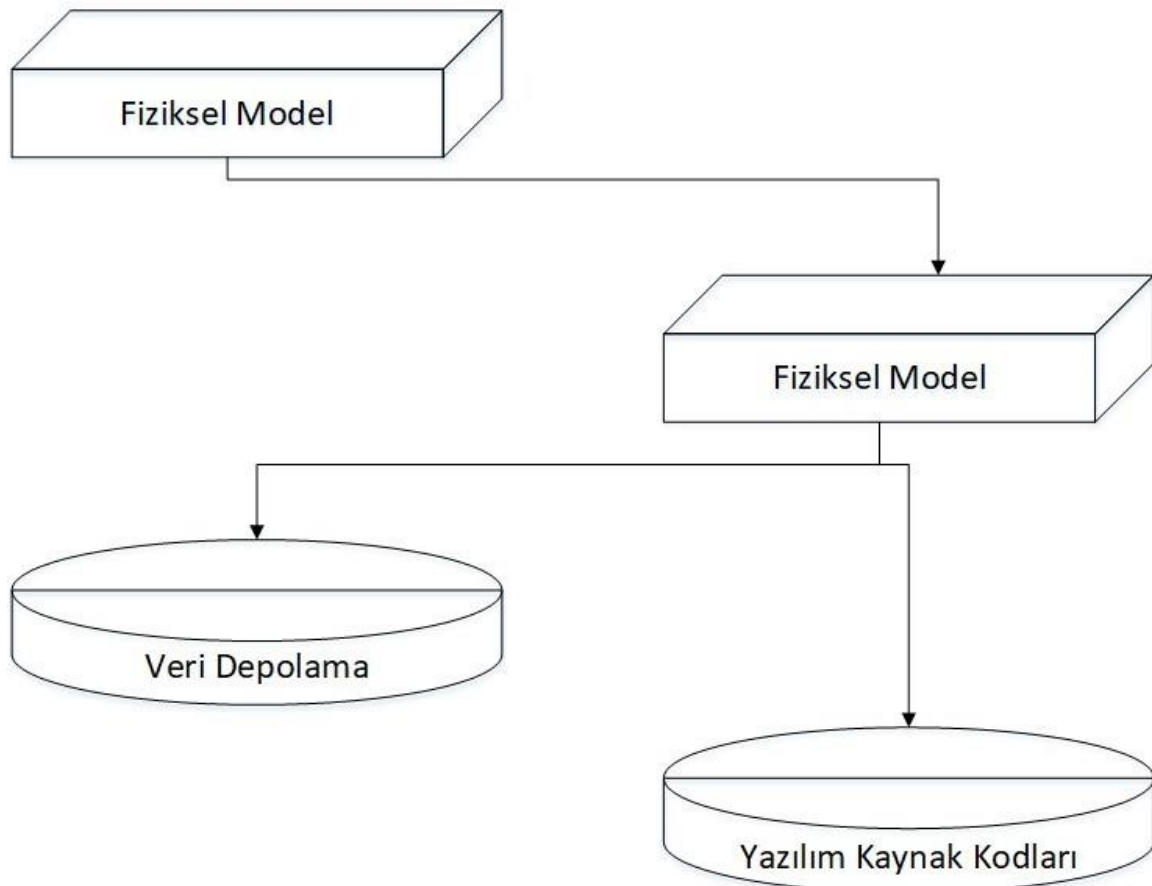
- Model seçme ve düzenlemeye yönelik faaliyetlerin temelinde uygun hedeflerin tespit edilmesi bulunmaktadır.
- Hata ve aksaklıkların analiz edilmesi için uygun verilerin tanımlanması gerekmektedir.
- Belirtilen hedeflere yönelik veriler modellenir.
- Geçmişe yönelik verilerin zaman bilgilerini de içerecek şekilde elde edilerek yazılım geliştirme sürecine dâhil etmek.

- Yazılım geliştirme sürecinin modellenmesi, hata ile karşılaşılıp, test sürecine başlamak ve model doğrulama işlemlerine gerçekleştirmek.
- Güvenilirlik tahminleme modelinin seçilmesini sağlamak.
- Güvenilirlik modeli tarafından kullanılacak olan parametrelerini tespit etmek.
- Verilen bir noktayı kullanarak gelecekteki olası hatalar hakkında tahmin yapmak.
- Tahmin edilen hata ve arıza oranları ile gerçekleşen değerleri kıyaslamak.

## 5. SİSTEM GERÇEKLEŞTİRİMİ

### 5.1 Giriş

Tasarlanan sistemin diğer sistemlerde çalışması için farklı işletim sistemlerine uyarlanması gerekmektedir. Bu uyarlamada her sistemin aynı veritabanlarına bağlanması ve aynı sorguları çalıştırması gerekir.



## Şekil 5.1 Sistem Gerçekleştirim Modeli

### 5.2 Yazılım Geliştirme Ortamları

#### 5.2.1 Programlama Dilleri

- Python 3  
Programı tamamen Python platformunda yazıldı.
- Apple Swift(XCode)

İkinci bir işletim sistemi olan İOS, Android kullanmayan bazı kullanıcılar için tercih edilmektedir. Bu işletim sisteminde sınırlı kodlama programları olduğundan bu işletim sistemine en duyarlı olan Xcode kullanılmıştır.

#### 5.2.1.1 CASE Araç ve Ortamları

Case araçları olarak ise Visio Project kullandık.

### 5.3 Kodlama Stili

Kodlama stili olarak yazılım dünyası standartlarında kodlama kurallarına uygun şekilde kodlamalar yapıldı.

#### 5.3.1 Açıklama Satırları

Her karmaşık kod satırında kodun ne işe yaradığını belirten kısa cümlelerle açıklama satırları eklendi.

#### 5.3.2 Kod Biçimlemesi

Nesne tabanlı programlama yöntemi kullanıldı.

#### 5.3.3 Anlamlı İsimlendirme

Projenin devamı esnasında oluşabilecek personel kaybında bir başkasının görevinin en kolay biçimde devam ettirilebilmesi için değişkenler anlamlı bir şekilde isimlendirildi.

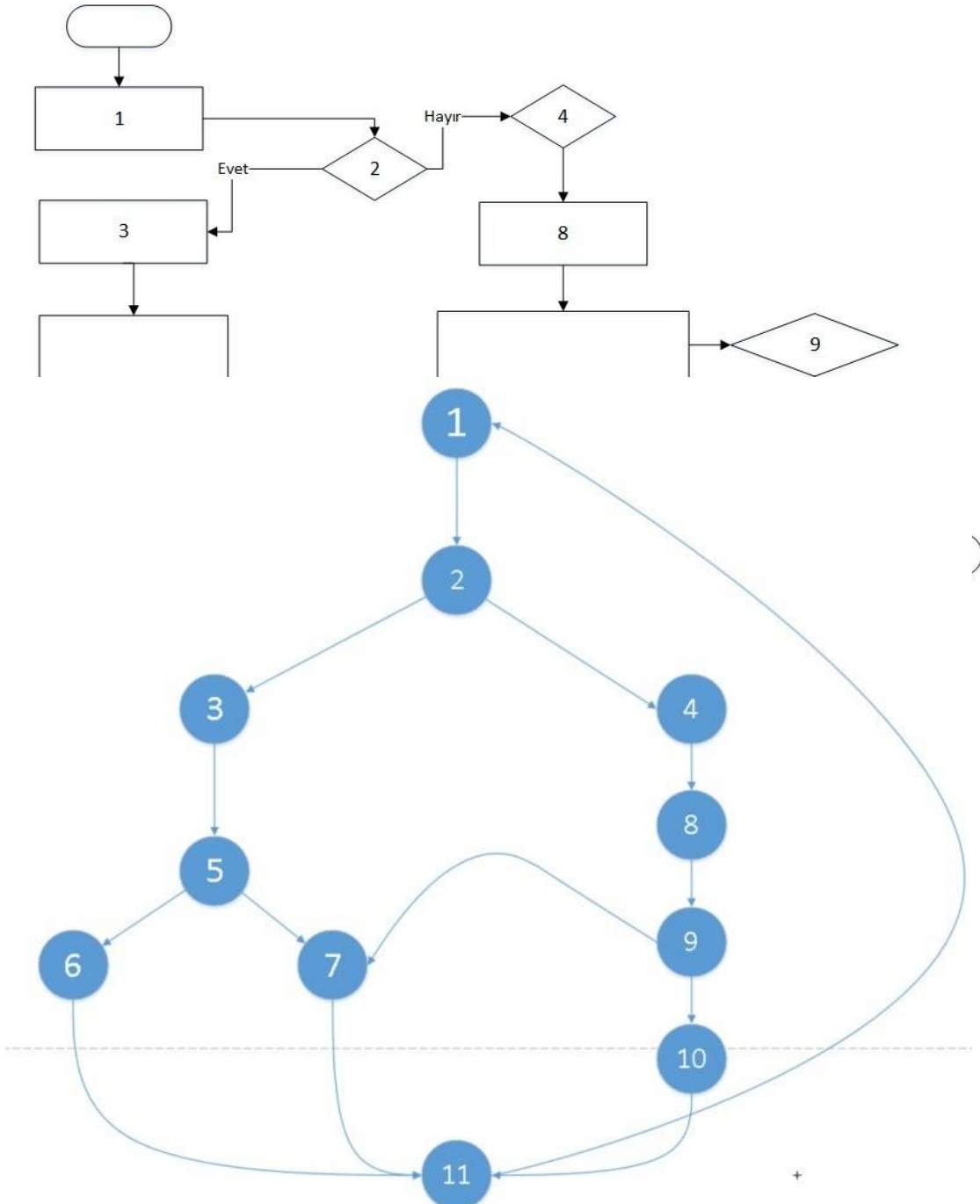
### 5.3.4 Yapısal Programlama Yapıları

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 yazıcımıza tek bir döngüyle bu sorunların üstesinden geldik.

## 5.4 Program Karmaşıklığı

### 5.4.1 Programın Çizge Biçimine Dönüştürülmesi



## 5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama

**(k-d+2p)**

k : Kenar Sayısı	14
d : Düğüm Sayısı	11
p : Bileşen Sayısı	1

$$14 - 11 - 2 \times 1 = 5$$

Fazla riskli olmayan basit modül.

Şekil 5.7 Uygulamanın Çizgi Hali

## 5.5 Olağan Dışı Durum Çözümleme

Karşılaşılması beklenilmeyen durumlar karşımıza olağan dışı durum olarak çıkacaktır.

### 5.5.1 Olağandışı Durum Tanımları

- Word dışı belgenin programa verilmesi.

### 5.5.2 Farklı Olağan Dışı Durum Çözümleme Yaklaşımları

## 5.6 Kod Gözden Geçirme

Öncelikli amacımız kod kalitesini arttırmak. Yazdığımız kodu bizden farklı kişilerin inceleyip yorumlaması kod kalitesi açısından çok faydalı olacaktır. Code Review(Kod Gözden geçirme) yaptığımızda projenin her alanında yetkinliğimiz artar ve bir yerde bug çıktığında bug fix yaparken zorlanmayız. Yazılımcıların yetkinliğini de olumlu yönde etkilemektedir. Bir projedeki yazılımcıların hepsi aynı yetkinlikte değildir kimisi daha deneyimli kimisi çok daha az deneyimlidir. Code Review(Kod Gözden geçirme) yaptığımızda bilgi birikimimizi diğer arkadaşlarımızda aktararak herkesin yetkinliğini arttırmış oluruz.

### 5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi

Gözden geçirme sürecinin temel özellikleri:

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

### 5.6.2 Gözden Geçirme Sürecinde Kullanılacak Sorular

Bir program gözden geçirilirken programın her bir işlevi için bu soruların yanıtları aranır.

#### 5.6.2.1 Öbek Arayüzü

- Uygulamanın her bir öbeği istenilen hizmeti yerine getirebiliyor mu?
- Uygulamanın her bir öbeği sade ve anlaşılabilir bir durumda mı?
- Uygulamanın girdi ve çıktı uyuyor mu?

#### 5.6.2.2 Giriş Açıklamaları

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

şeklinde oldu.

#### 5.6.2.3 Veri Kullanımı

- Oluşturduğumuz veri kullanımlarını test etmek için belli sorular sorduk bu sorular:
- İşlevsel olarak ilintili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?



- Değişken adları,işlevlerini yansıtacak biçimde anlamlı mı?
- Değişkenlerin kullanımları arasındaki uzaklık anlamlı mı?
- Her değişken tek bir amaçla mı kullanılıyor?
- Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?
- Tanımlanan her gösterge değişkeni için bellek ataması yapılmış mı?

şeklinde oldu.

#### 5.6.2.4 Öbeğin Düzenlenişi

- Modüller birleşimi uyumlumu?
- Modüller arası veri aktarımları sağlanıyor mu?
- Bütün modüller birleştiğinde sistem çalışıyor mu?

Gözden geçirme sırasında referans alınacak sorular olacaktır.

#### 5.6.2.5 Sunuş

Artık son kısma gelindiğinde ise şu sorular soruldu:

- Her satır, en fazla bir deyim içeriyor mu?
- Bir deyim 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?

## 6. DOĞRULAMA VE GEÇERLEME

### 6.1 Giriş

Geliştirilen bu uygulama belirli aralıklarla sistemin en başından en sonuna kontrol edilmektedir. Girdi ve çıktı sayılarının bir önceki aya göre aynı sonuçları vermesi ya da geliştirilen yeni girdi ve çıktıların doğru şekilde çalışması kontrol edilmektedir.

Geliştirilen sistem müşteri tarafından istenen sistem midir? Sorusuna doğru yanıt almak gerekmektedir.

Ürünün içsel niteliğine ilişkin izleme ve denetim etkinliklerinden oluşur. Geçerleme Ürünü doğru olarak mı üretiyoruz? Doğrulama ve geçerleme işlemleri temel olarak çeşitli düzeylerde sına, gözden geçirme, denetim ve hata giderme süreçlerinden oluşur.

**Doğrulama:** Bir sistemin ya da bir bileşenin başlangıç aşamasında dayatılan şartları, ürünün verilen geliştirme aşaması emniyetini saptamak için değerlendirilmesidir. Örneğin; Yazılım tasarımı önce belirlenen gereksinimleri uygulamak açısından yeterli mi? Kod tamamıyla ve doğru bir şekilde tasarımı kodlayabiliyor mu?

**Geçerleme:** Bir sistemin ya da bileşenin geliştirme aşamasının sonunda ya da süresince belirlenen şartları yerine getirip getirmediğini saptamak için değerlendirilmesidir. Diğer bir deyişle; Kodlanan sistem belirlenen kullanıcı gereksinimlerini karşılıyor mu?

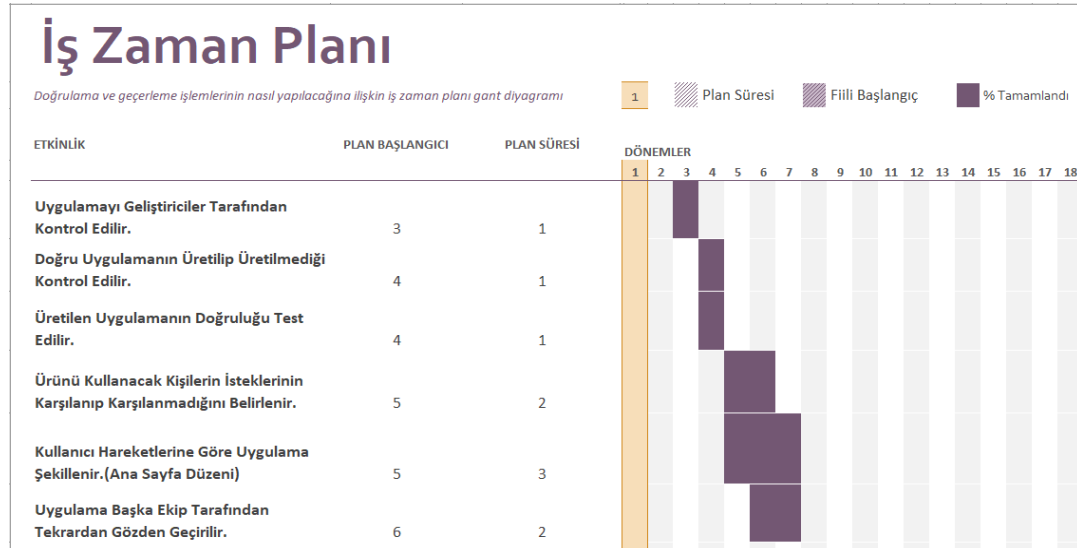
Tüm Doğrulama ve Gerçekleştirme işlemleri bir başka şirket tarafından yapılır. Kurum içi başka bir bölüm/grup tarafından da yapılabilir.

### Avantajları

- Konusunun uzmanları tarafından yapılır.
- Kurum içi siyasetten etkilenmez.
- Hatalar daha erken çözümlenir.

### Dezavantajları

- Daha pahalıdır.
- Daha sıkı koordinasyon gerektirir.
- Sistemi anlama aşaması vardır.
- Proje 3. Kişiler tarafından bilinmektedir.
- Sonuçlarda çelişkiye düşülebilir.



Şekil 6.1 Doğrulama ve Geçerleme İş-Zaman Planı

## 6.2 Test Doğrulama Planı

### Test İçin Kullanılan Yöntemler

Uygulamaya çalıştığınız herhangi bir yazılım testi aslında beş boyut içermektedir. Bunlar:

**Testi Yapanlar: Yazılımı testi işlemini kim yapıyor?**

- Uygulama geliştiriciler tarafından test edilmektedir. Geliştiriciler tarafından test edildikten sonra belirli kullanıcılar tarafından sistem kapsamlı bir teste sokulur. Burada kullanıcılar tüm fonksiyonları denerler ve hataları tekrardan geliştiricilere bildirirler.

**Potansiyel Problemler:** Neden yazılım testi uygulanıyor (Hangi risklere bakılıyor)?

Tablolarda belirtilen fonksiyonların her biri için hizmeti yerine getirip getirilmediği kontrol edilir.

Ortaya çıkabilecek sorunları daha önceden belirleyip kullanıcılara bunu yansıtmadan hatanın düzeltilmesi amaçlanmaktadır.

**Doğrulama:** Yazılım testinin geçtiği ya da kaldığı nasıl belirlenecek?

Belirli kullanıcıların sistemi kullandıktan sonra geri bildirim olarak sistemdeki sorunları belirtirse bu uygulama tekrardan geliştiriciler tarafından düzeltilecektir. Geliştiriciler uygulamayı test ederken herhangi bir problemle karşılaşmaz ise bu uygulama artık testin geçtiğini gösterir. Aksi halde geliştiriciler herhangi bir problemle karşılaşırsalar uygulama yazılım testinden kalır ve tekrardan hataların giderilmesi amaçlanır.

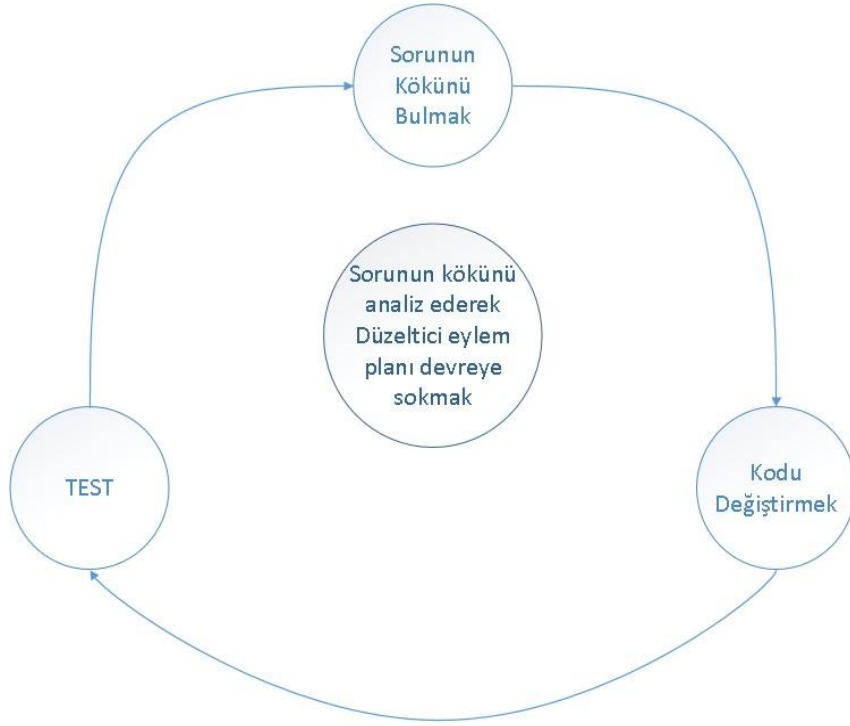
### 6.3 Kullanılacak Test Araçları

- Python IDE
- Xcode

## 7. BAKIM

### 7.1 Giriş

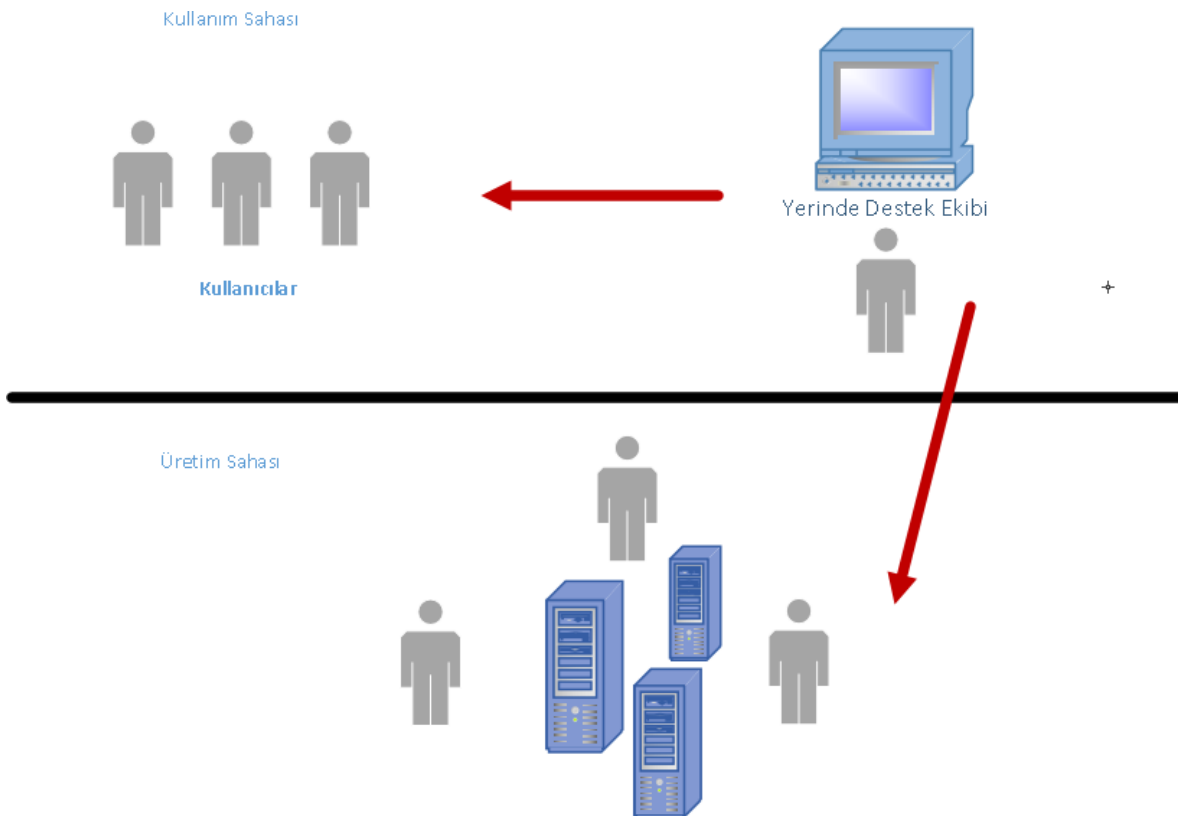
Sınama işlemleri bitirilen yazılımın kullanıcı alanına yüklenmesi ve uygulamanın başlatılması gerekmektedir. Yazılım kullanıma geçtikten sonra, yaşam döngüsünün en önemli ve hiç bitmeyecek aşaması olan "bakım" aşaması başlar. Bankacılık Denetleme ve Düzenleme Kuruluna uygun bir şekilde sürekli değişen kurallar çerçevesinde yapılacak hassas ve hata kabul etmeyecek bir sistemi düzenlemek ve bakımını gerçekleştirmek için yapacağımız açıklamalar da IEEE 1219-1998 standardı baz alınmıştır.



Şekil 7.1 Bakım Planı

## 7.2 Kurulum

Sistem kurulumu Github'tan ulaşabilir MAC/WINDOWS işletim sistemlerinde rahatlıkla kullanılabilir.

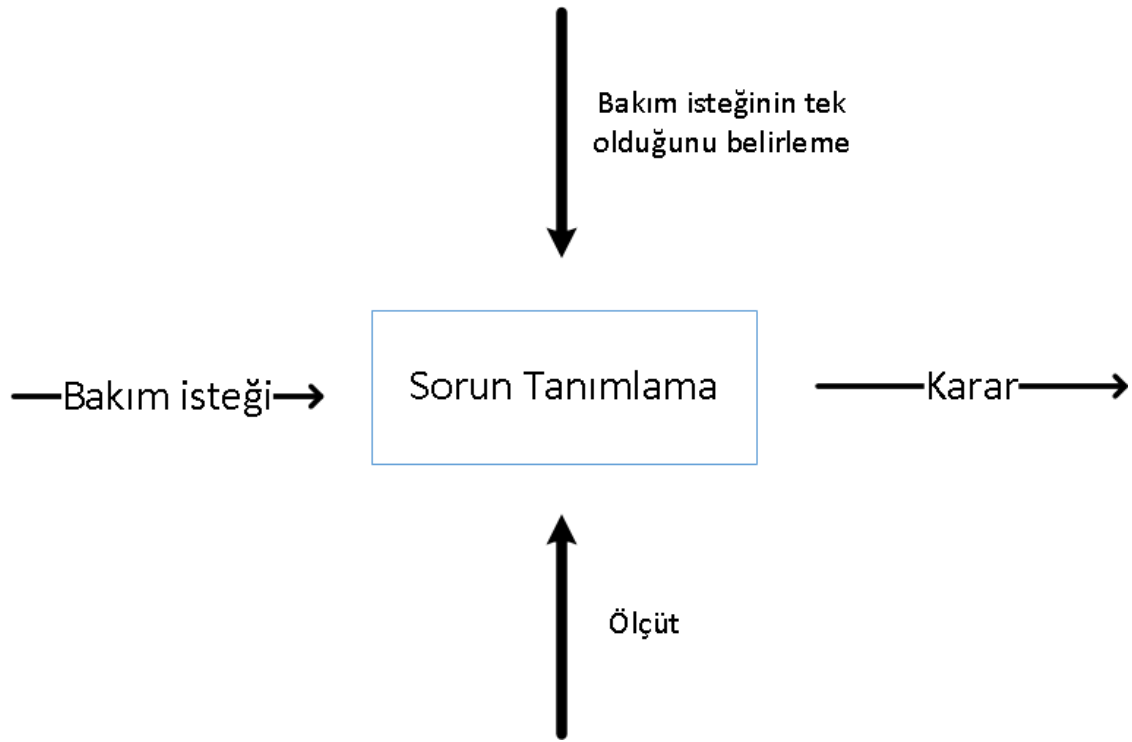


Şekil 7.2 Bakım Aşaması

### 7.3 Kurulum Entegrasyon Süreci

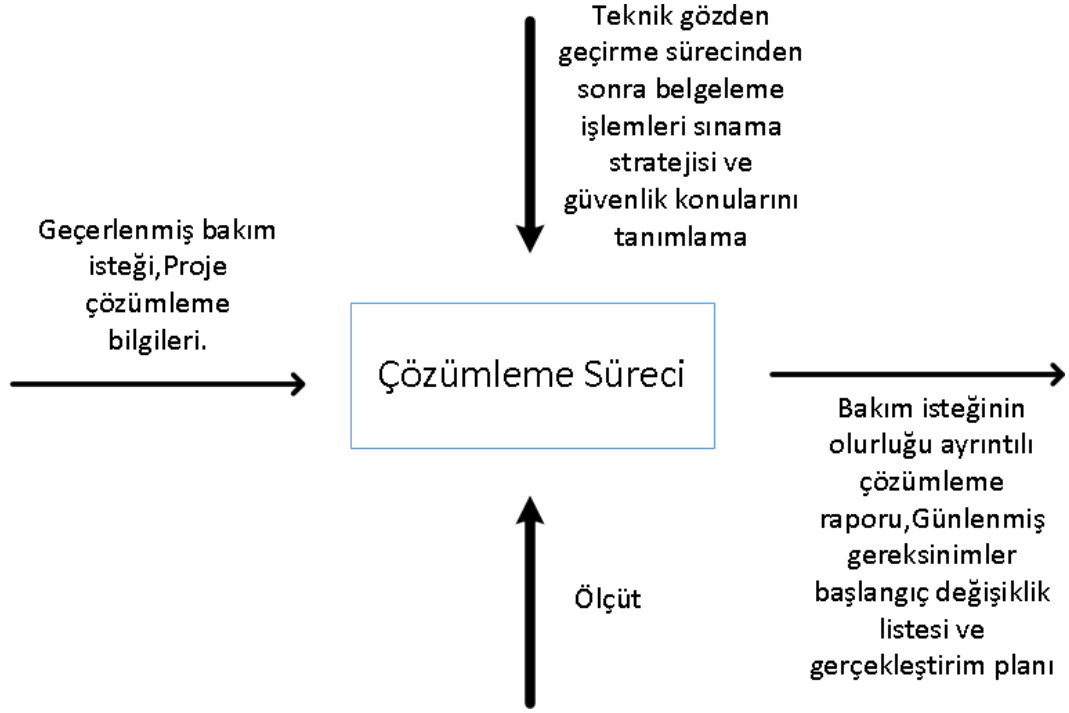
#### 7.3.1 Bakım Süreç Modeli

##### 1. Adım: Sorunu Tanımlama Süreci



Şekil 7.3 Sorun Tanımlama

## 2. Adım: Çözümleme Süreci



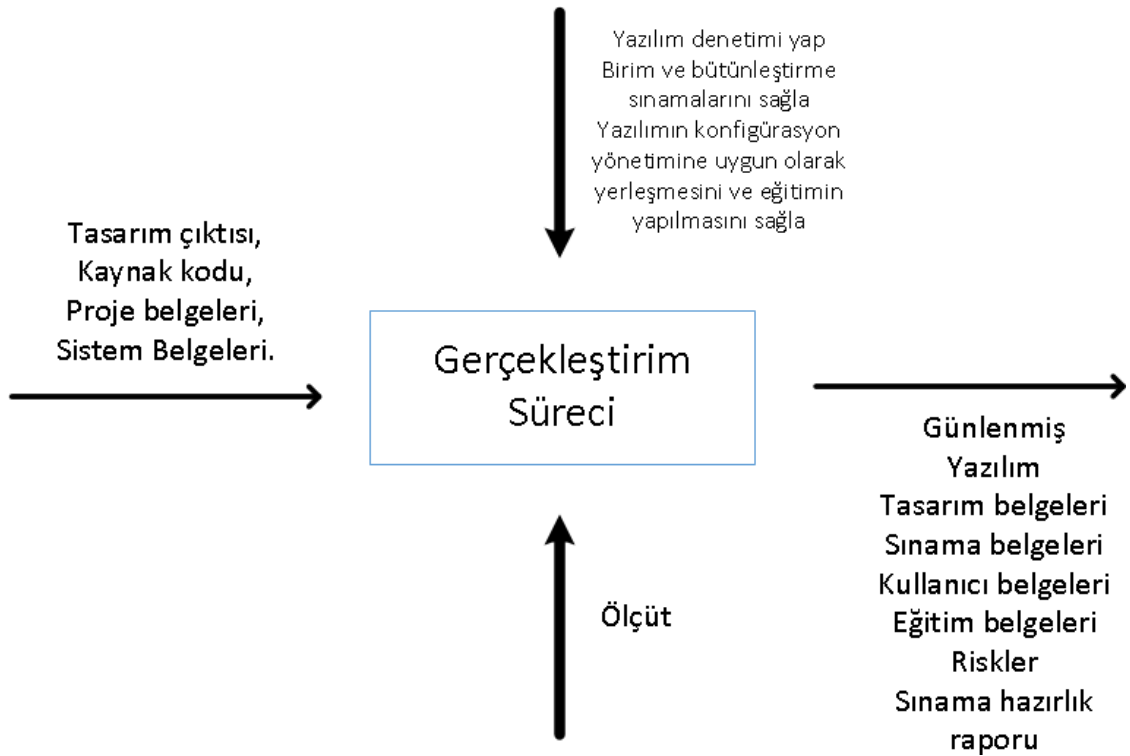
Şekil 7.4 Çözümleme Süreci

## 3. Adım: Tasarım Süreci



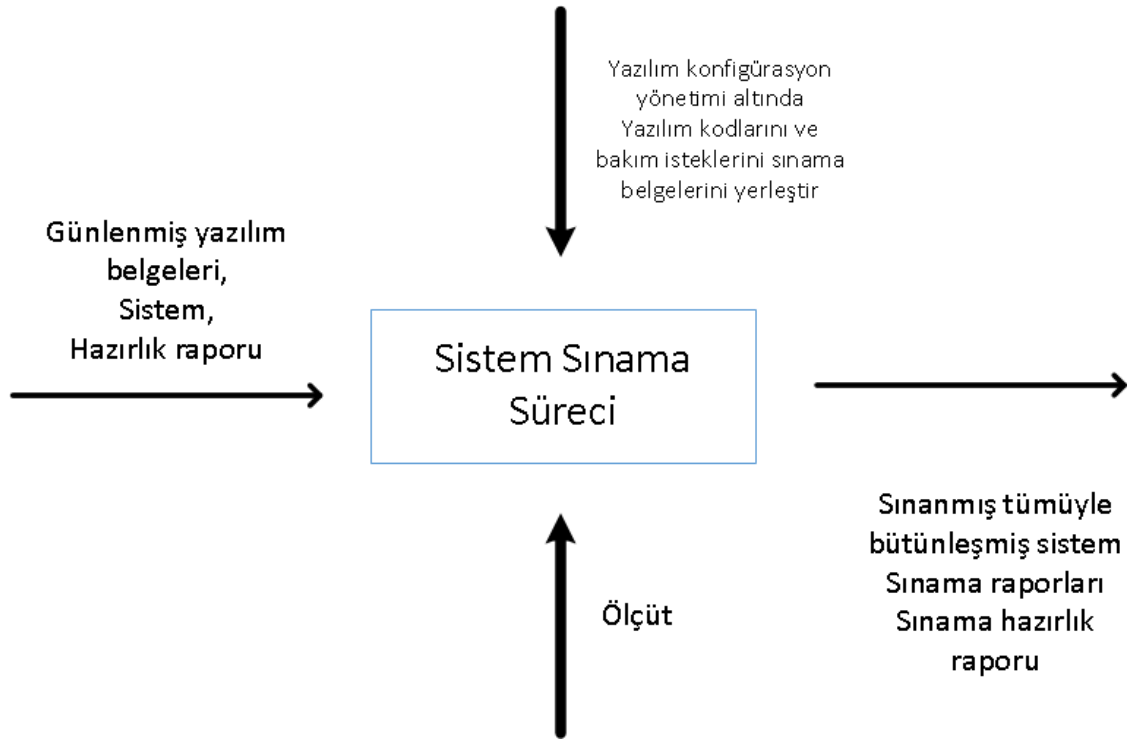
Şekil 7.5 Tasarım Süreci

#### 4. Adım: Gerçekleştirim Süreci



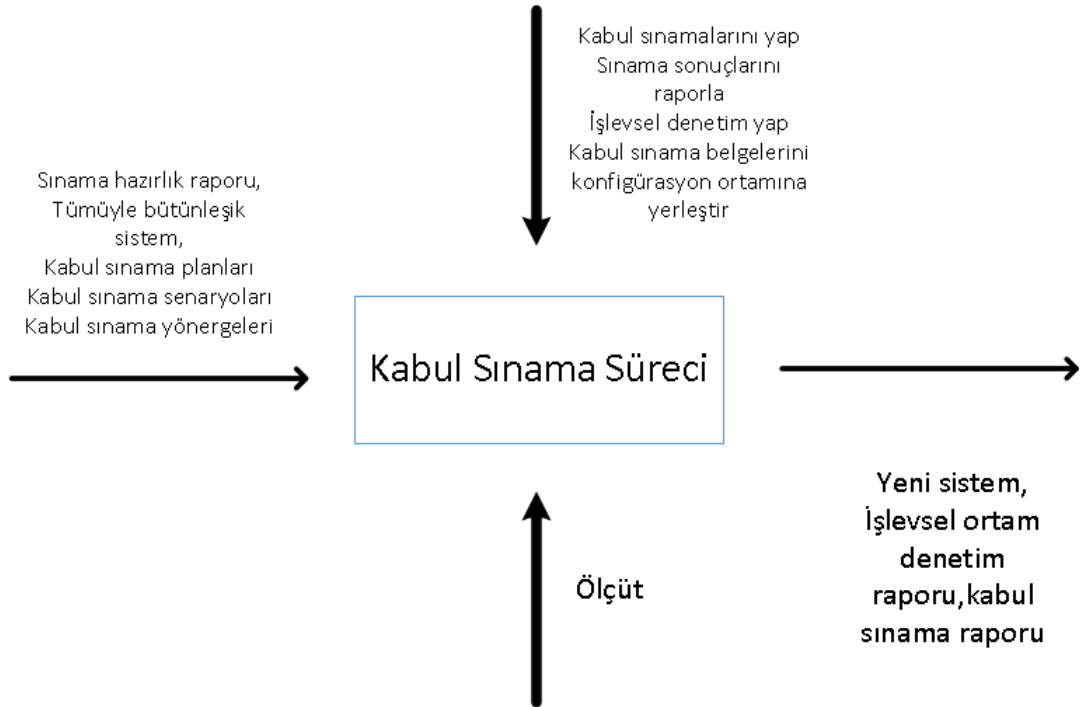
Şekil 7.6 Gerçekleştirim Süreci

## 5. Adım: Sistem Sınama Süreci



Şekil 7.7 Sistem Sınama Süreci

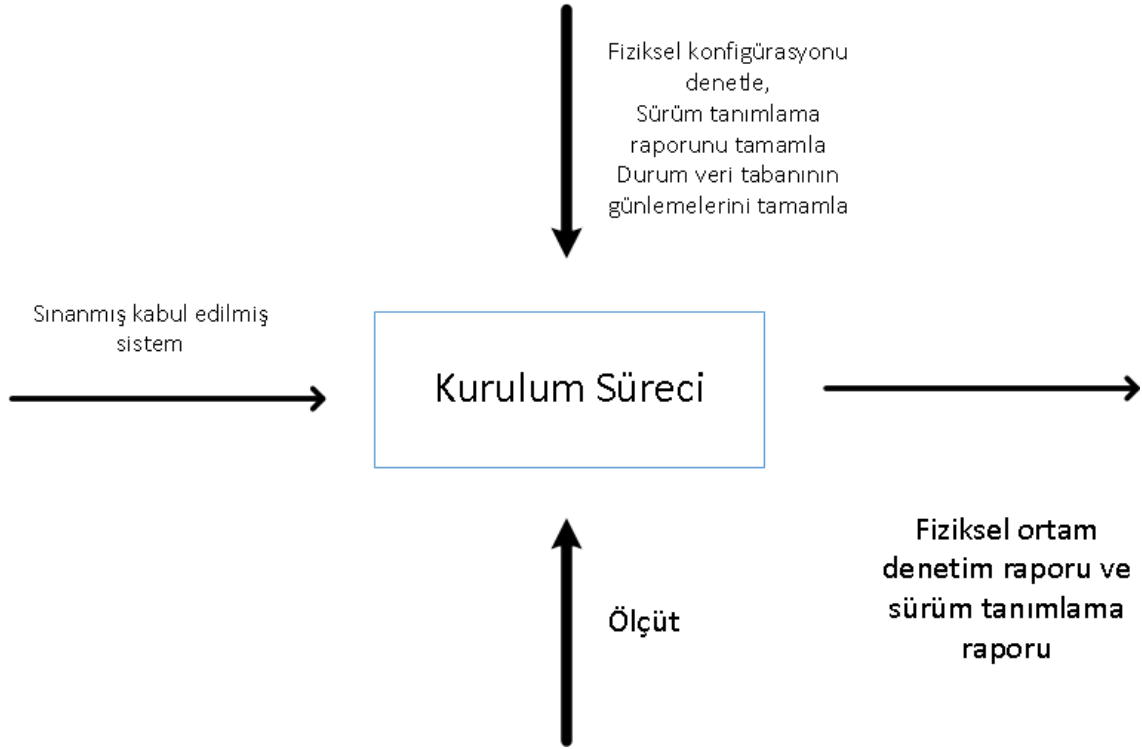
## 6. Adım: Kabul Sınaması Süreci



Şekil 7.8 Kabul Sınama Süreci



## 7. Adım: Kurulum Süreci



Şekil 7.9 Kurulum Süreci

## 8. SONUÇ

Kullanıcı programa girdi olarak verdiği Word belgesinin ayrıntılı ve detaylı şekilde incelenmiş raporunu yine Word belgesi olarak elde eder.

## 9. KAYNAKLAR

### İnternet Kaynakları

<http://muhammetbaykara.com/wp-content/uploads/2017/02/2.pdf> (2.Bölüm – Genel)

<https://docs.microsoft.com/tr-tr/dynamics365/unified-operations/supply-chain/production-control/visual-scheduling-production> (2. Bölüm – Gantt Diyagramı)

<https://templates.office.com/tr-TR/Proje-zaman-%C3%A7izelgesi-TM00000005> (2. Bölüm – Zaman İş Planı)

<http://muhammetbaykara.com/wp-content/uploads/2017/03/YMH-114-HAFTA-4.pdf> (2. Bölüm – Maliyet Hesaplama)

<http://www.yazilimcildunyasi.com/2017/01/cocomo-constructive-costing-model.html> (2. Bölüm – COCOMO Modeli)

[http://muhammetbaykara.com/wp-content/uploads/2017/03/YMT114-5.Hafta\\_.pdf](http://muhammetbaykara.com/wp-content/uploads/2017/03/YMT114-5.Hafta_.pdf) (3.Bölüm – Genel)

[http://muhammetbaykara.com/wp-content/uploads/2017/03/YMT114-5.Hafta\\_.pdf](http://muhammetbaykara.com/wp-content/uploads/2017/03/YMT114-5.Hafta_.pdf) (4.Bölüm – Genel)

<http://muhammetbaykara.com/wp-content/uploads/2017/03/YMT114-6.Hafta-Tasar%C4%B1m.pdf> (4. Bölüm – Genel)

<https://medium.com/@halilozel1903/kotlin-nedir-43e312d2dca6> (5.Bölüm – Programlama Dilleri)

<https://tr.wikipedia.org/wiki/Xcode> (5. Bölüm – Programlama Dilleri)

<https://www.featuredcustomers.com/vendor/oracle-exadata/customers/ziraat-bankasi> (5. Bölüm – Veri Tabanı Dilleri ve Arabirimleri)

<http://muhammetbaykara.com/wp-content/uploads/2017/04/YMT114-7.Hafta-Ger%C3%A7ekle%C5%9Ftirim.pdf> (5.Bölüm – McCabe)

<http://muhammetbaykara.com/wp-content/uploads/2017/04/YMT114-8.Hafta-Yaz%C4%B1l%C4%B1m-Do%C4%9Frulama-ve-Ge%C3%A7erleme.pdf> (6 – Doğrulama Ve Geçerleme)

<http://muhammetbaykara.com/wp-content/uploads/2017/09/1-Yaz%C4%B1l%C4%B1m-M%C3%BChendisli%C4%9Finde-Kalite.pdf> (6. Bölüm – Yazılım Mühendisliğinde Kalite)

<https://systemanalyze.wordpress.com/mimari-tasarim/> (6.Bölüm – Mimari Tasarım)

<https://sherpa.blog/sozluk/arayuz-arabirim-nedir> (6.Bölüm - Arayüz Arabirimleri)

<https://docs.microsoft.com/tr-tr/dotnet/framework/winforms/interfaces-related-to-data-binding> (6. Bölüm - Arayüz)

<https://prezi.com/eyytplk0uh3m/arabirimler/> (6.Bölüm – Arabirimler)

<https://www.btgrubu.com/urunler/yazilim-test-araclari/> (6.Bölüm – Yazılım Test Araçları)

<http://celenkdeniz.weebly.com/nedir-bu-selenium-web-driver/selenium-webdriver-nedir> (6. Bölüm – Selenium webdriver)

<http://testkalite.com/katalon-studio-test-otomasyon-araci/> (6. Bölüm – Katolon Studio)

<http://muhammetbaykara.com/wp-content/uploads/2017/04/YMT114-9.Hafta-Bak%C4%B1m.pdf> (7. Bölüm – Genel)

<https://www.mera.com/services/maintenance> (7. Bölüm – 7.1)