T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSLİĞİ

GÜNCEL KONULAR

YMGK2 - BİLMİYORUM

Proje Ekibi

14545520 Kemal SANLI
14542505 Seda YUMRUTEPE
14545565 Haşim DELİL
15541518 Gül ÖNAL
16542502 Emine SAĞIROĞLU
16541504 Fatih ULUDAĞ
16542510 Turan ÇAYMAZ
16541561 Ali METİN
175541059 Batuhan HARMANŞAH
175541040 Halil İbrahim YANIK
175541028 Furkan ERDOĞAN
175541017 Kübra YILMAZKAR

Ocak 2021

1. GİRİŞ

1.1 Projenin Amacı

1.2 Projenin Kapsamı

2. PROJE PLANI

2.1 Giriş

2.2 Projenin Plan Kapsamı

2.3 Proje Zaman-İş Planı

2.4 Proje Ekip Yapısı

2.5 Önerilen Sistemin Teknik Tanımları

2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları

2.6.1 Kullanılan Programlama Araçları

2.6.2 Prototipleme ve Simülasyon Araçları

2.6.3 Test Araçları

2.6.4 Proje Yönetim Araçları

2.7 Proje Standartları, Yöntem ve Metodolojiler

2.8 Kalite Sağlama Planı

2.9 Konfigürasyon Yönetim Planı

2.10 Kaynak Yönetim Planı

2.11 Eğitim Planı

2.12 Test Planı

2.13 Bakım Planı

2.14 Projede Kullanılan Yazılım/Donanım Araçlar

3. SİSTEM ÇÖZÜMLEME

3.1 Mevcut Sistem İncelemesi

3.1.1 Örgüt Yapısı

3.1.2 İşlevsel Model

3.1.3 Var olan Yazılım/Donanım Kaynakları

3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

3.2.2 İşlevsel Model

3.2.3 Genel Bakış

3.2.4 Bilgi Sistemleri/Nesneler

3.2.5 İşlevlerin Sıradüzeni

3.2.6 Başarım Gerekleri

3.3 Arayüz (Modül) Gerekleri

3.3.1 Yazılım Arayüzü

3.3.2 Kullanıcı Arayüzü

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

3.4.2 Eğitim Belgeleri

3.4.3 Kullanıcı El Kitapları

4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri

4.1.1 Genel Sistem Tanımı

4.1.2 Varsayımlar ve Kısıtlamalar

4.1.3 Sistem Mimarisi

4.1.4 Dış Arabirimler

4.1.4.1 Kullanıcı Arabirimleri

4.1.4.2 Veri Arabirimleri

4.1.5 Testler

4.1.6 Performans

4.2 Süreç Tasarımı

4.2.1 Genel Tasarım

4.2.2 Kullanıcı Profilleri

4.2.3 Entegrasyon ve Test Gereksinimleri

4.3 Ortak Alt Sistemlerin Tasarımı

4.3.1 Güvenlik Alt sistemi

4.3.2 Veri Dağıtım Alt sistemi

4.3.3 Yedekleme ve Arşivleme İşlemleri

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1. Giriş

5.2. Yazılım Geliştirme Ortamları

5.2.1 Programlama Dilleri

5.2.1.1 Hazır Program Kütüphane Dosyaları

5.3. Kodlama Stili

5.3.1 Açıklama Satırları

5.3.2 Kod Biçimlemesi

5.3.3 Anlamlı İsimlendirme

5.3.4 Yapısal Programlama Yapıları

5.4. Olağan Dışı Durum Çözümleme

1. GİRİŞ

1.1 Projenin Amacı

Mobil platformlarda bilgi güvenliğinin sağlanması için etkin bir güvenlik algoritması tasarlamak. Etkili bir görüntü şifreleme algoritmasını tasarlarken mobil cihazın kendi ram verilerini SHA3 fonksiyonundan geçirerek bilinen matematiksel fonksiyonlarla rastgele sayı dizileri üretip bunları web servis hizmeti şeklinde sunarak görüntüleri şifrelemek.

1.2 Projenin Kapsamı

Sistem ve sistemin getirileri, internet ortamında haberleşme esnasında paylaştıkları verilerin güvenli iletilmesini sağlamak isteyen kullanıcılar içindir.

2. PROJE PLANI

2.1 Giriş

Gerçekleştirilen projede, güvensiz olan internet ortamında kullanıcıların birbirleriyle olan haberleşmelerinde iletilen verilerin korunması esasına dayanmaktadır. Bir saldırgandan, gerçek verinin korunması için veri şifrelenmektedir.

2.2 Projenin Plan Kapsamı

Projenin plan kapsamında genel olarak mevcut sistem, sistemin gerekliliği ve bu sistemin güvenilirliğinden yola çıkıldı. Proje beş temel aşamadan oluşmaktadır. Her bir aşama ayrı ayrı gerçekleştirilip birleştirilmiştir. IP1 iş paketinde mobil uygulamanın arayüzü bulunmaktadır. SHA3 hash algoritması kullanılmıştır. IP2 iş paketinde SHA3 ile elde edilen veri ve kaotik sistem metriği ile xor işlemi gerçekleştirilir. IP3 kaotik sistem hesaplamaları ve rastgele sayı üretilir. IP4 iş paketinde Api tasarımı gerçekleştirilmiş ve şifreleme algoritmaları kullanılmıştır. IP5 iş paketinde görüntü şifrelenir, kullanıcıya sunulur ve şifrelenen görüntü tekrar çözümlenmektedir.

Projede amaç, güvensiz olan internet ortamında kullanıcıların birbirleriyle olan haberleşmelerinde iletilen verilerin korunması esasına dayanmaktadır.

Maliyet Kestirim Planı:

Bu bölümde maliyet kestirim planıyla ilgili olan yapılara yönelik durum Tablo 1'de gösterilmiştir.

Tablo 1. *Maliyet Kestirim Planı*

Ölçüm Parametresi	**Sayı**	**Ağırlık**	**Toplam**
Kullanıcı Girdi Sayısı	4	5	20
Kullanıcı Çıktı Sayısı	2	3	6
Ana İşlev Nokta Sayısı			26

İncelenen projeye ait maliyet kestirim planında ölçüm parametreleri göz önünde bulundurularak sayı ve ağırlıklar göz önünde bulundurularak puanlanmıştır.

Bu bölümde teknik karmaşıklık faktörüne yönelik durum Tablo 2'de gösterilmiştir.

Tablo 2. *Teknik Karmaşıklık Faktörü*

Teknik Karmaşıklık Sorusu	**Puan**
1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	**5**
2. Veri iletişimi gerektiriyor mu?	**5**
3. Sistem, çevrim içi veri girişi gerektiriyor mu?	**5**
4. Çevrim içi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu?	**0**
5. Performans kritik mi?	**4**
6. Girdiler, çıktılar, dosyalar ya da sorgular karmaşık mı?	**0**
7. İçsel işlemler karmaşık mı?	**2**
8. Tasarlanacak kod yeniden kullanılabilir mi?	**3**
9. Dönüştürme ve kurulum tasarımda dikkate alınacak mı?	**3**

10. Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak?	**5**
Toplam	**32**

İncelenen projeye ait teknik karmaşıklık faktörü, teknik karmaşıklık soruları 0 ile 5 arasında puanlandırılarak değerlendirilmiştir. Karmaşıklık faktörü puanlama sistemi aşağıdaki gibi belirlenmiştir.

0: Hiçbir Etkisi Yok

1: Çok Az Etkisi Var

2: Etkisi var

3: Ortalama Düzeyde Etkisi Var

4: Önemli Düzeyde Etkisi Var

5: Kesinlikle Etkisi Var

Tablo 1 (Maliyet Kestirim Planı) ve Tablo 2 (Teknik Karmaşıklık Faktörü)'de elde edilen toplam puanlar kullanılarak aşağıda yapılan hesaplamalar ile maliyet kestirimi hesaplanmıştır.

İN=AİN*(0,65*0,01*TKF)

İN= 26*(0,65*0,01*32)

İN= 5.408

Satır Sayısı= İN*30

Satır Sayısı= 1.664*30

Satır Sayısı= 162.24

Üretkenlik = İN / Kişi-Ay

Üretkenlik = 5.408/ 13-4

Üretkenlik = 412

Etkin Maliyet Modeli – COCOMO

Ayrık Proje: a=2,4, b=1,05, c=2,5, d= 0,38

Yarı – Gömülü Projeler İçin: a=3,0, b=1,12, c=2,5, d= 0,35

Gömülü Projeler İçin: a=3,6, b=1,20, c=2,5, d= 0,32

Öncelikle projemizin türünü belirlememiz gerekiyor. Küçük ekip tarafından geliştirildiği için ayrık projeler arasına giriyor.

Aylık Kişi Başı İş Gücü = E = a x (KSS)b

Geliştirme Süresi (Ay) = D = c x (E)d Eleman Sayısı = E / D

Formülde verilen değişkenler şöyle:

KSS = Kod Satır Sayısı manasına gelmektedir ve birimi bin satırdır.

Projenin tahmini kaç bin satırdan oluşacağını belirtmemizi sağlar.

Aylık Kişi Başı İş Gücü = E= 2,4*0,16*1,05=0.4032

Geliştirme Süresi =D= 2,5 x =0.4032*13=13.104

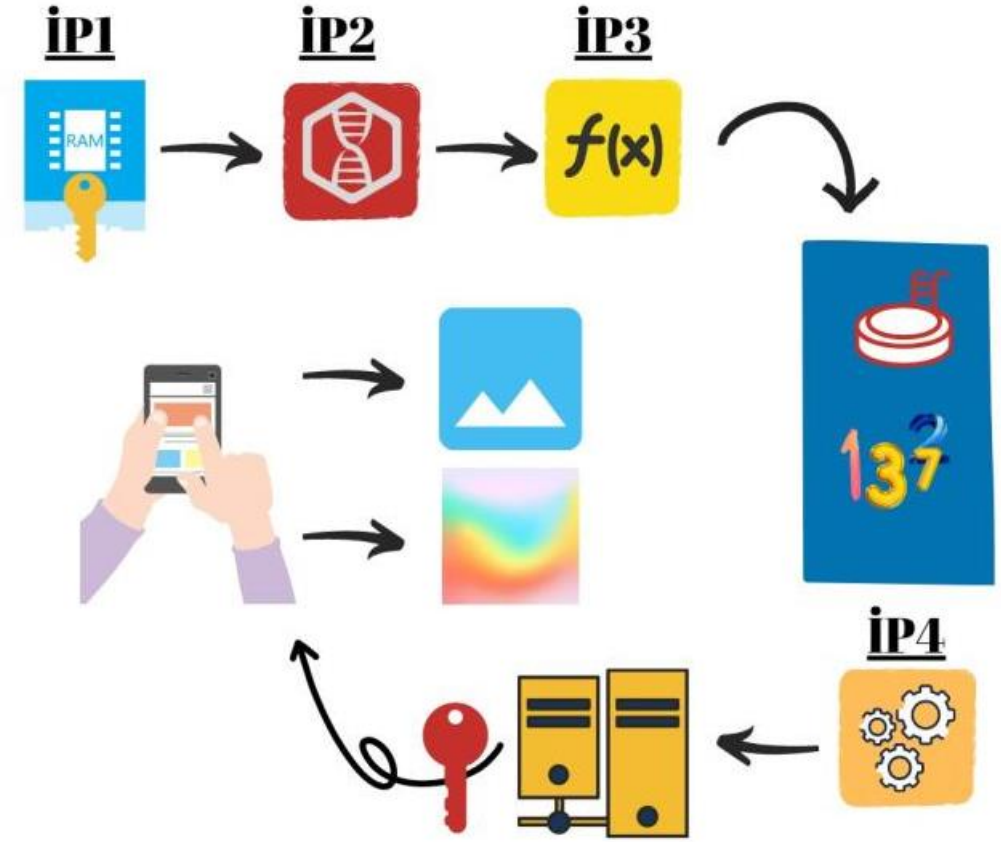
2.3 Proje Zaman-İş Planı

Bu bölümde ekip üyelerinin faaliyetlerine ilişkin zaman ve iş planı yapılmıştır.

Faaliyetler	Sorumlu Kişi	Aylar			
		Ekim	Kasım	Aralık	Ocak
Eğitim	KS, SY,HD,GÖ,FU,ES,TÇ, AM,BH,HİY,FE,KY	X X	X	x	
Ip1 İş Paketi Sistem Gerçekleştirimi	KS		/		
IP2 İş Paketi Sistem Gerçekleştirimi	KS				
IP3 İş Paketi Sistem Gerçekleştirimi	FU,ES,TÇ		X	/	
IP4 İş Paketi Sistem Gerçekleştirimi	KS, FU, BH,HİY		X	X	
IP5 İş Paketi Sistem Gerçekleştirimi	KS, HD, FU, BH		X	/	
Sistem Çözümle(Doküman Ekibi)	SY, GÖ, AM, FE, KY			/	/
Test Gerçekleştirimi	SY, KS				/
KS: Kemal Sanlı, SY: Seda Yumrutepe, HD: Haşim Delil, GÖ: Gül Önal, FU: Fatih Uludağ, ES: Emine Sağıroğlu,					
TÇ: Turan Çaymaz, AM: Ali Metin, BH: Batuhan Harmanşah,HİY: Halil İbrahim Yanık, FE: Furkan Erdoğan,					
KY: Kübra Yılmazkar					

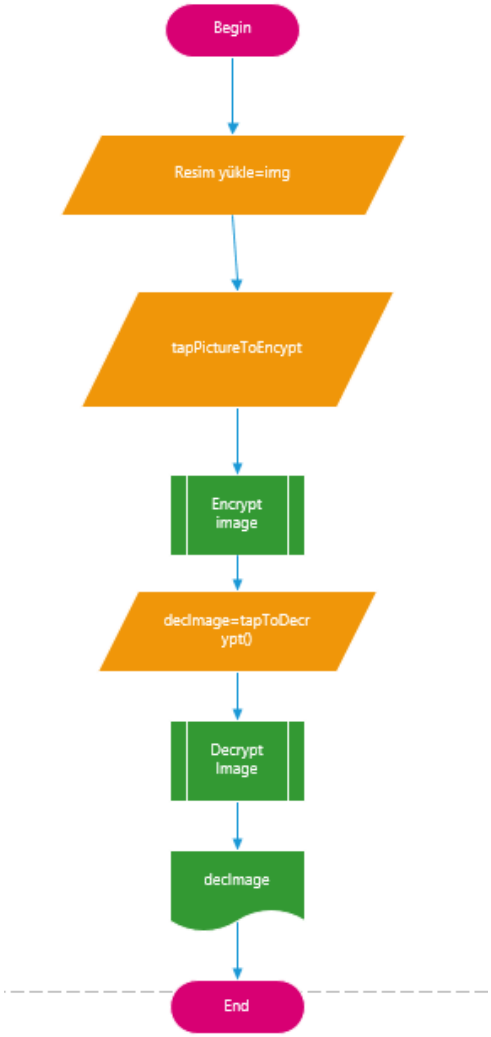
Şekil- 1 (İş - Zaman Tablosu)

Şekil-1’de iş- zaman tablosu verilmiştir. Her bir üyenin her bir faaliyet için harcadığı süre miktarı yer almaktadır. (**“/”** bir haftayı temsil etmektedir.)



Şekil-2 (Proje Süreci)

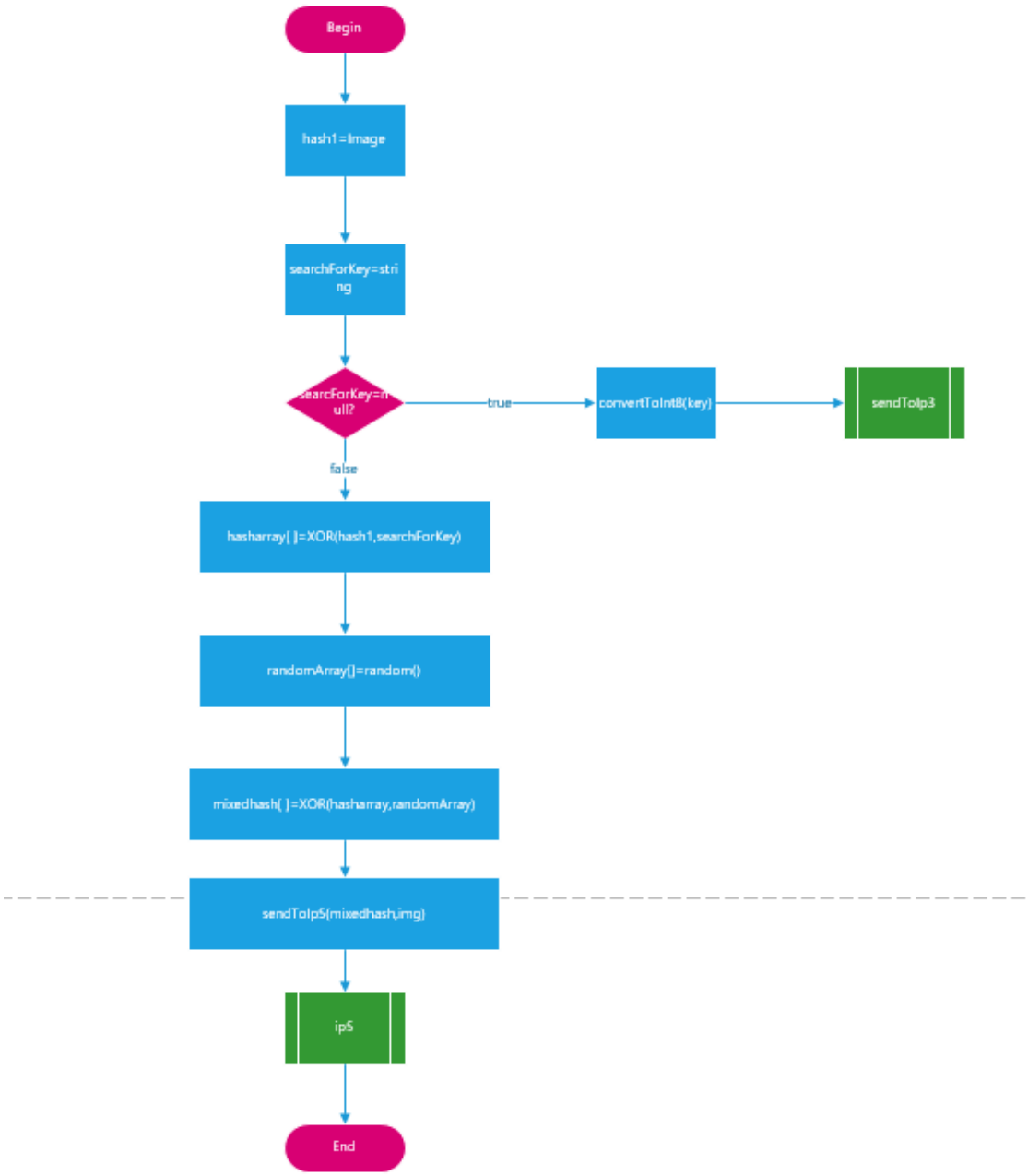
Aşağıda projeye ait iş akış diyagramları yer almaktadır.



Şekil-3 (İş Akış Diyagramı)

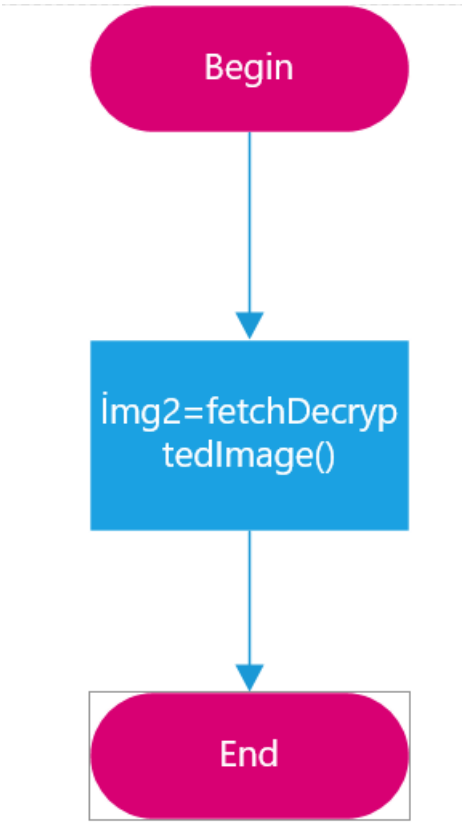
Şekil-3'de projeye ait genel iş akışı yer almaktadır.

Her bir process' e ait daha detaylı iş akış diyagramları Şekil-3, Şekil-4, Şekil-5, Şekil-6 ve Şekil-7'de yer almaktadır.



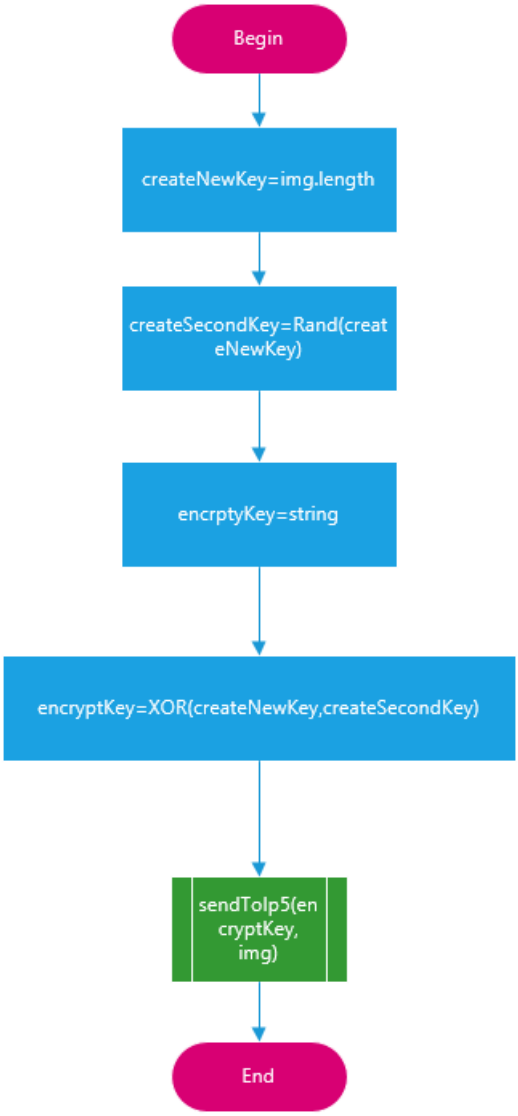
Şekil-4 (İş Akış Diyagramı- Encrypt)

Şekil-4'te görüntünün şifrelenmesine ilişkin iş akış diyagramı yer almaktadır.



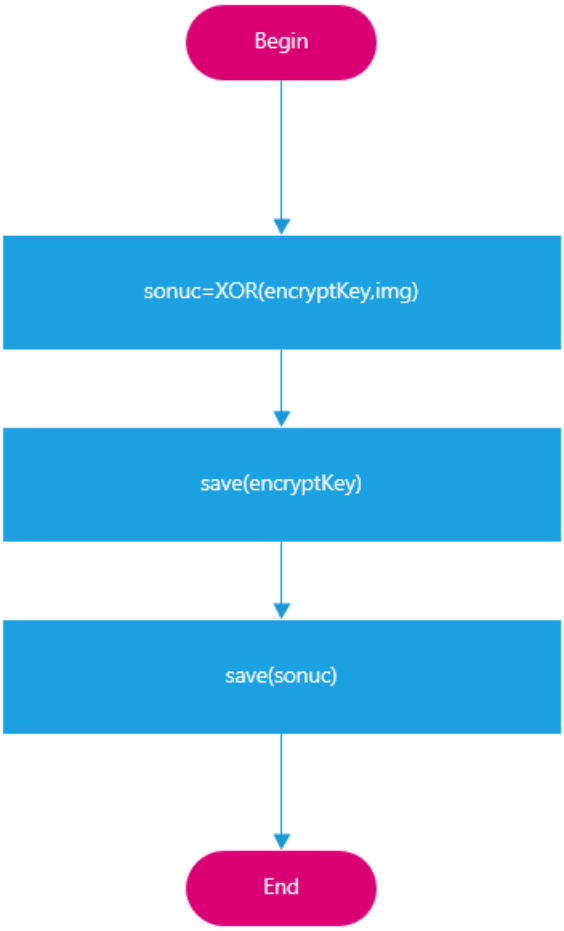
Şekil-5 (İş Akış Diyagramı- Decrypt)

Şekil-5'te görüntüye ait şifrenin açılmasına ait iş akış diyagramı yer almaktadır.



Şekil-6 (İş Akış Diyagramı- IP3)

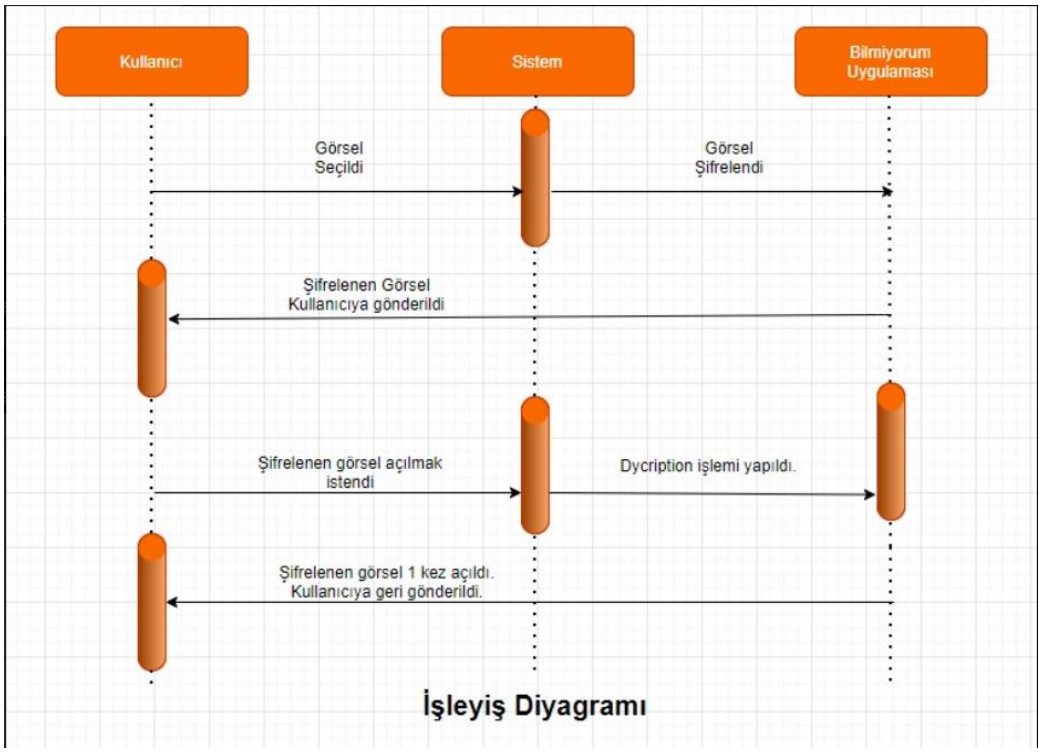
Şekil-6'da iş paketi IP3'e ait iş akış diyagramı yer almaktadır.



Şekil-7 (İş Akış Diyagramı- IP5)

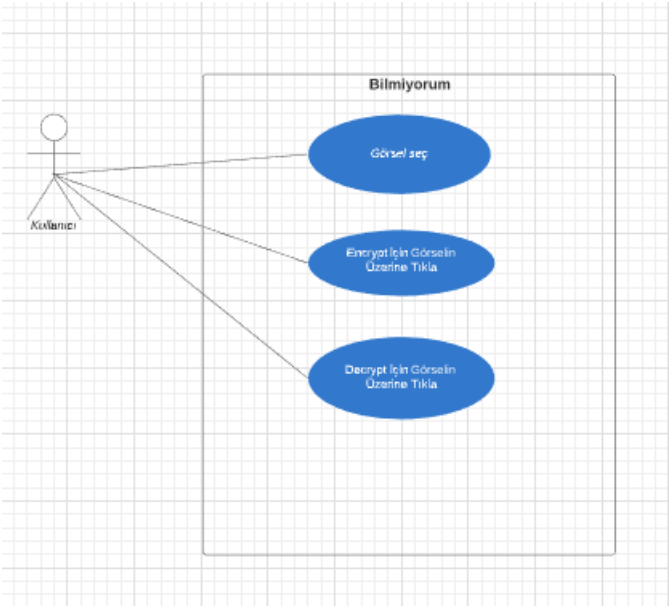
Şekil-7'da iş paketi IP5'e ait iş akış diyagramı yer almaktadır.

Şekil-8'de İşleyiş diyagramı verilmiştir.



Şekil-8 (İşleyiş Diyagramı)

Kullanışlı bir yapıya sahip olan projemize ait Use Case diyagramı aşağıda Şekil-9'da verilmiştir.

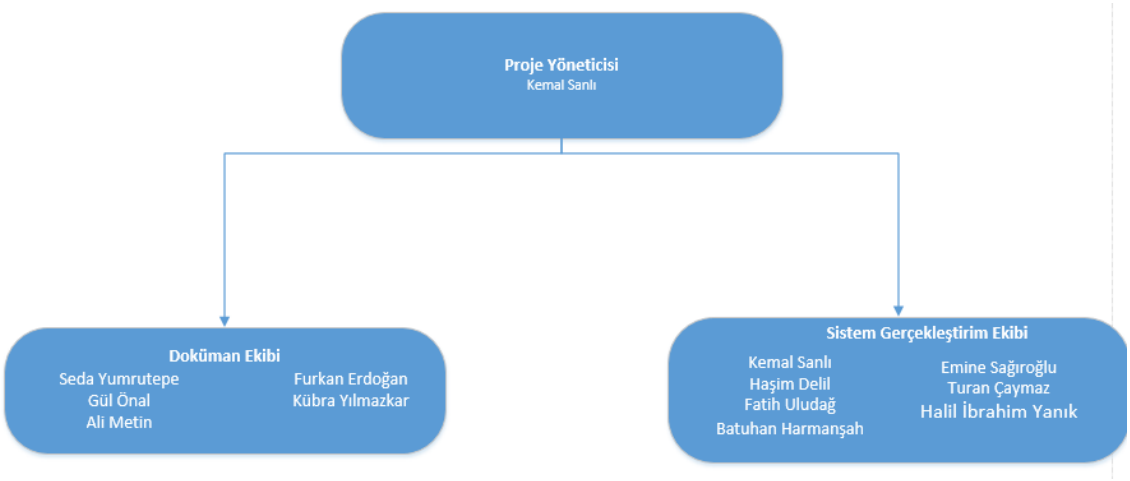


Şekil-9 (Use Case Diyagramı)

Şekil-9'de kullanıcı mobil cihazından bir görsel seçer. Görselin üzerine tıklayınca seçilen görsel şifrelenir. Tekrar görselin üzerine tıklayınca şifrelenen görüntünün şifresi kaldırılır.

2.4 Proje Ekip Yapısı

Proje ekip yapısına ilişkin bilgiler aşağıda yer alan Şekil-10'da detaylı bir şekilde verilmiştir. Ekip üyeleri ve görev aldıkları iş parçacıkları belirtilmiştir.



Şekil-10 (Proje Ekip Yapısı Şeması)

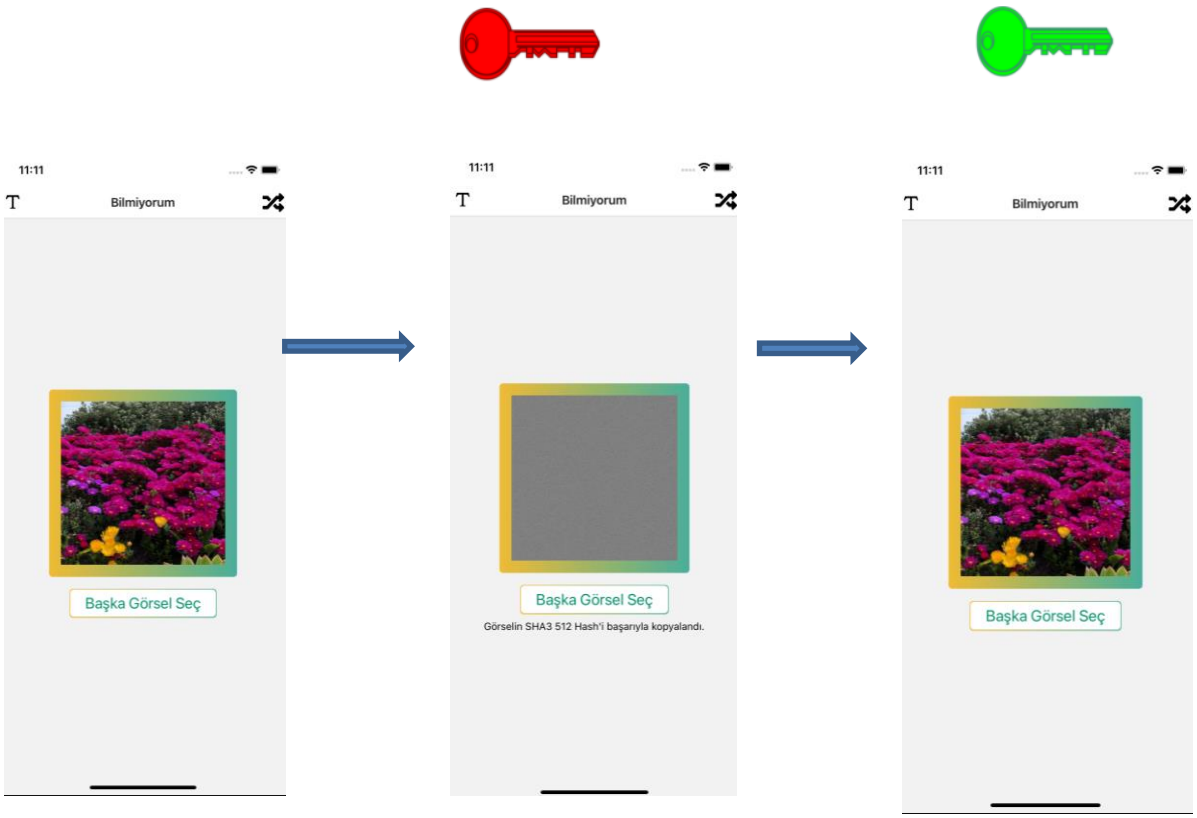
2.5 Önerilen Sistemin Teknik Tanımları

Bu bölümde projemizi gerçekleştirirken kullandığımız teknolojiler yer almaktadır.

Kullanılan Teknolojiler

- Matematiksel şifreleme yöntemleri kullanılır. Okunan veriyi şifrelemek için SHA3 hash algoritması kullanılır.
- Özel anahtar (Sha3 ile elde edilen 256 bit uzunluğundaki veri) Ram 'den okunan veriyi şifrelemek için kullanılır. Sadece kullanıcının kendisinde bulunur. Kullanıldıktan sonra silinir.
- Özel anahtar ile şifrelenmiş görüntü açılır.

Şekil-9'da özel anahtar ile şifrelenen görüntüler verilmiştir.



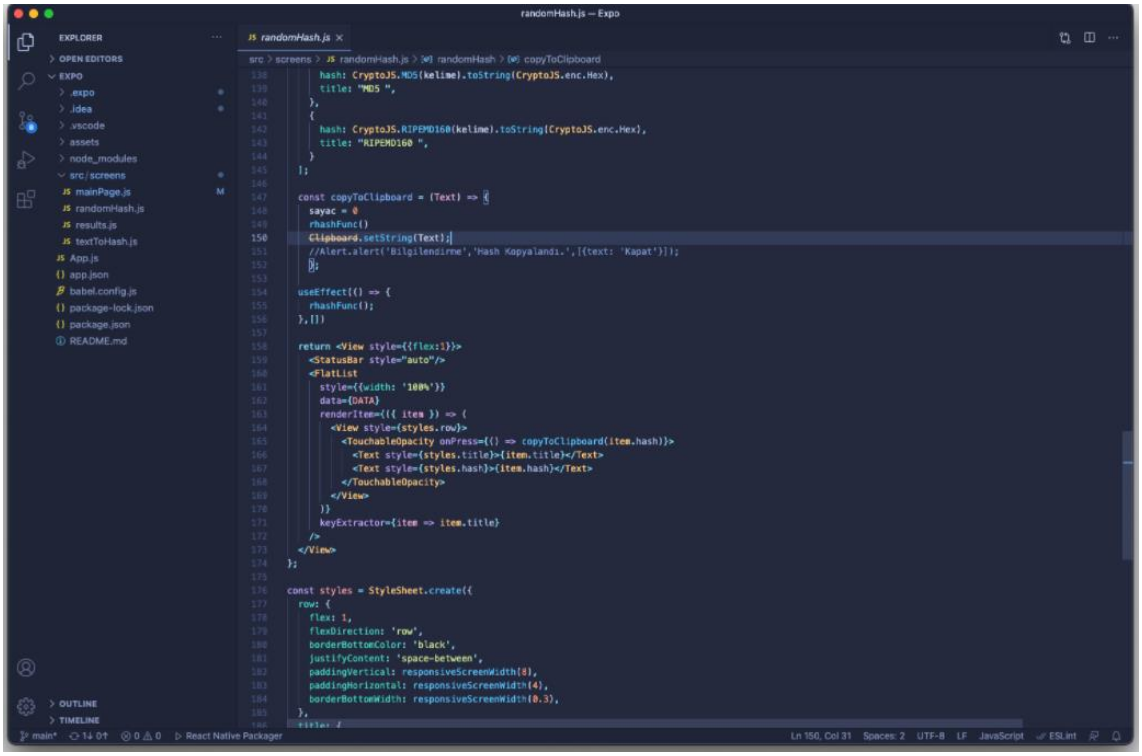
Şekil-11 (Uygulama Şifreleme Örneği)

2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları

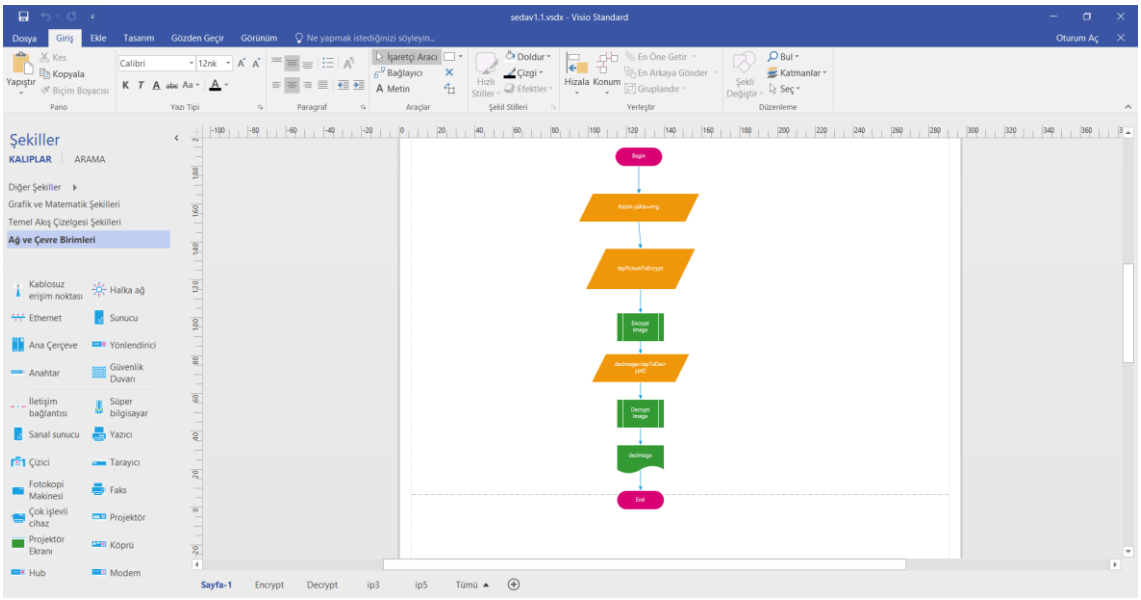
Projede kullanılan özel geliştirme araçları ve ortamları aşağıda yer alan Şekil - 12'deki şemada verilmiştir.

Şekil-12 (Kullanılan Özel Geliştirme Araç & Ortamları)

2.6.1 Kullanılan Programlama Araçları

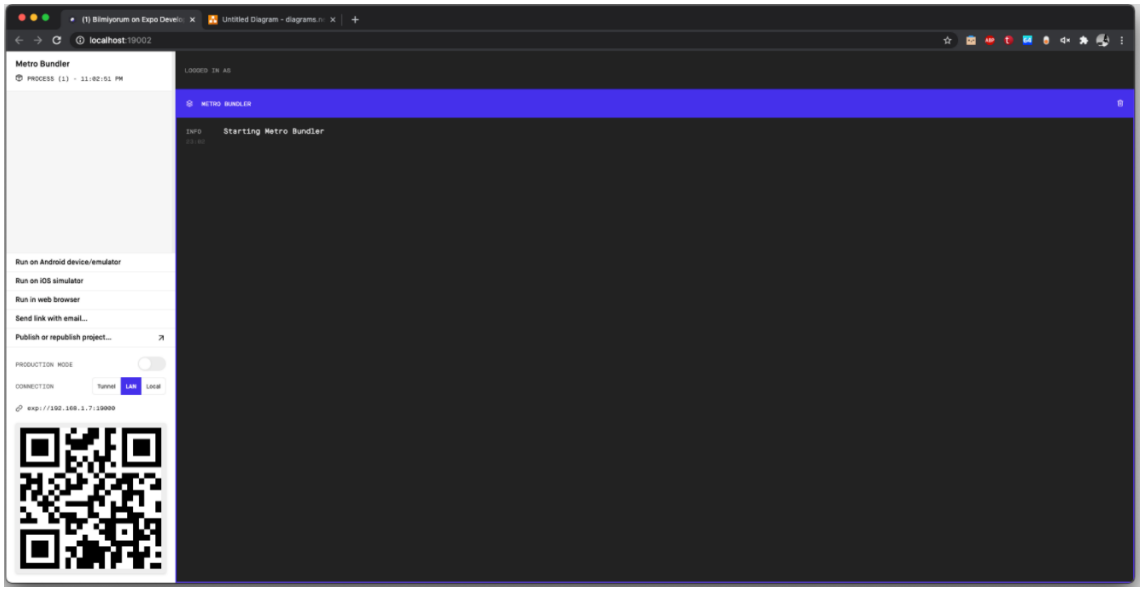


Şekil-13 (Kullanılan Özel Geliştirme Araçları - VS Code)



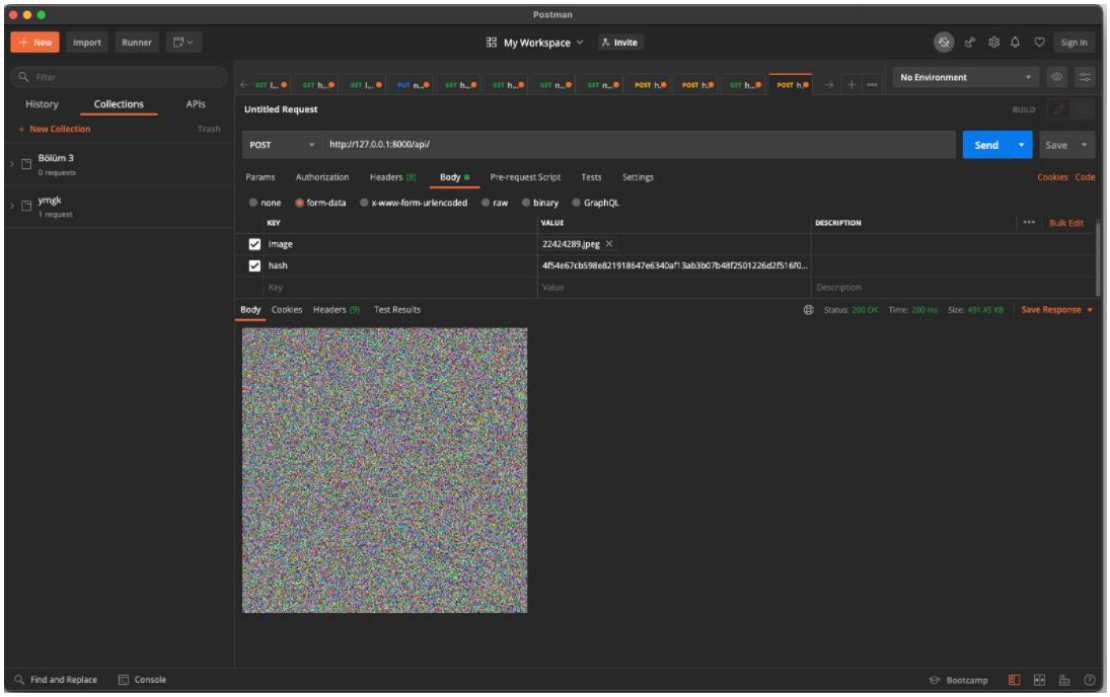
Şekil-14 (Kullanılan Özel Geliştirme Araçları – Microsoft Visio)

2.6.2 Prototipleme ve Simülasyon Araçları



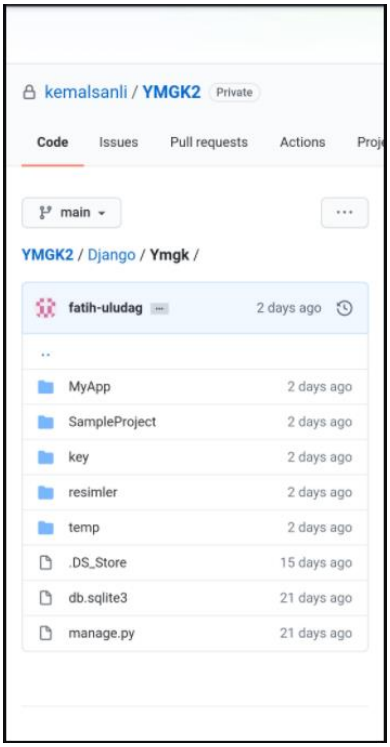
Şekil-15 (Prototipleme ve Simülaasyon Araçları - Expo)

2.6.3 Test Araçları



Şekil-16 (Test Araçları - Postman)

2.6.4 Proje Yönetim Araçları



Şekil-17 (Proje Yönetim Araçları - Github)

2.7 Proje Standartları, Yöntem ve Metodolojiler

Tablo 3. *Proje Standartları Yöntem & Metodolojileri*

Aşama	Kullanılan Yöntem/Araçlar	Ne İçin Kullanıldığı	Çıktı
Planlama	- Veri Akış Şemaları - Süreç Belirtimleri, - Görüşme, - Maliyet Kestirim Yöntemleri - Proje Yönetim Araçları	- Süreç İncelme - Kaynak Kestirim - Proje Yönetim	Proje Planı
Çözümleme	- Süreç Belirtimleri, - Veri Akış Şemaları - Görüşme, - Nesne İlişki Şemaları, - Veri Sözlüğü	- Süreç Çözümleme - Veri Çözümleme	Sistem Çözümleme Raporu
Çözümlemeden Tasarıma Geçiş	- Akışa Dayalı Çözümleme - Süreç Belirtimlerinin Program tasarım Diline dönüştürülmesi - Nesne İlişki Şemalarının Veri Tablolarına Dönüştürülmesi -	- Başlangıç Tasarım - Ayrıntılı Tasarım - Başlangıç Veri Tasarımı	Başlangıç Tasarım Rapooru
Tasarım	Yapısal Şemalar Program Tasarım Dili	- Genel Tasarım - Ayrıntılı Tasarım	Sistem Tasarım

Tablo.3'te projeye projeye ait standartlar, yöntem ve metodolojiler yer almaktadır.

2.8 Kalite Sağlama Planı

Projedeki kalite sağlama planımız aşağıda yer almaktadır.

1.Ekonomi: Ekonomik açıdan yazılımın maliyeti ucuz ve zaman tasarrufundan ötürü gayet uygundur.

2.Tamlık: Projede herhangi bir açık olmamalı ve programda bulunan tüm sistem çalışır ve tamdır.

3.Yeniden Kullanılabilirlik: Proje her koşulda tekrardan düzenlenip kullanılabilir.

4.Etkinlik: Kullanıcı, sistemin her alanına hâkim olduğu için sistemi etkin bir biçimde kullanabilir.

5.Bütünlük: Proje bir bütün halinde çalışmaktadır.

6.Güvenilirlik: Yüksek güvenlik önlemli barındırmaktadır.

7.Modülerlik: Sistem tek bir parçadan oluşmaktadır.

8.Belgeleme: Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olması amacıyla bu doküman oluşturulmuştur.

9.Kullanılabilirlik: Kullanılabilirlik olarak her seviyedeki insana hitap edeceğinden zor renkler karmaşık sistemlerden kaçınılmıştır.

10.Temizlik: Gerçekleştirim aşamasındaki kodlar temiz ve anlaşılır bir şekilde gerçekleştirilmiştir.

11.Değiştirilebilirlik: Kullanıcılar sistemde değişiklik yapabilir.

12.Esneklik: Proje, IOS ve Android platformlarda çalışabildiği için oldukça esnek bir yapıta sahiptir.

13.Genellik: Proje, herkes tarafından kullanılabilir ve geneldir.

14.Sınanabilirlik: Proje, sınanabilir bir yapıya sahiptir.

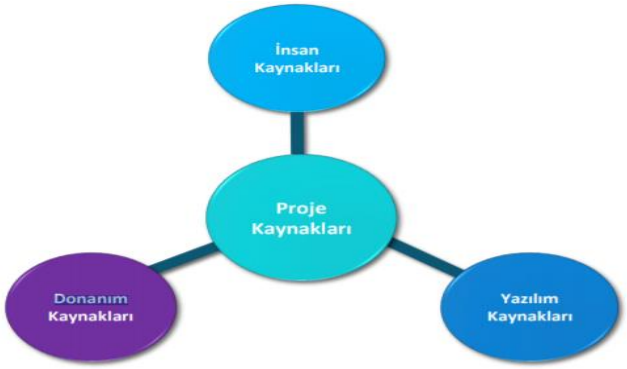
15.Taşınabilirlik: Sistem herhangi bir mobil cihaz üzerinden kullanılacağı için istenilen cihazlarda taşınabilir ve kullanılabilir.

16.Birlikte Çalışılabilirlik: Proje birleşik ve eş zamanlı çalışmaktadır.

2.9 Konfigürasyon Yönetim Planı

Sistemin ilerde kullanıcının yeni istemlerini karşılayamaması veya sistemin yapısındaki bazı bileşenlerin değişmesi sonucu güncelliğini kaybettiğinde olası yapılandırma planı hazırlandı.

2.10 Kaynak Yönetim Planı



Şekil-18 (Proje Kaynakları)

Şekil-18'de görüldüğü gibi proje kaynakları yazılım, donanım ve insan kaynakları olmak üzere 3 başlıkta toplanır.

Yazılım, donanım ve insan kaynakları belirlenip yetkinliklerine göre rol atamaları yapılmıştır.

Kaynak Edinimi:

Fatih Özkaynak - Kriptoloji Bilimine Giriş Dersi, Fatih Özkaynak & Zahir Muhammad Ziad Muhammad - "An Image Encryption Algortihm Based on Chaotic Selection of Robust Cryptographic Primitives".

Proje Ekibinin Geliştirilmesi:

İhtiyaçlar dâhilinde ve mevcut beceriyi geliştirmek amacıyla ekip yetkinlikleri geliştirilmiştir. Ekip etkileşimini arttırmak amacıyla sık sık toplantılar yapılıp beyin fırtınaları yapılmıştır.

Kaynak kontrolü yapılıp fiziksel kaynakların planda yer aldığı gibi atanıp atanmadığı, kaynakların planladığı gibi kullanılıp kullanılmadığı süreç boyunca kontrol edilmiştir.

2.11 Eğitim Planı

Projeden kazanılacak en önemli olaylardan biri de eğitimdir. Kullanılacak programlama dillerinin arayüz, editör ve programların kullanımında hâkim olunamaması halinde bu proje başarıyla neticelendirilemez. Bu nedenle projeye başlamadan önce bir takım eğitimlerin alınması gereklidir. Proje kapsamında alınacak olan eğitimler aşağıda yer almaktadır.

- Fatih Özkaynak - Kriptoloji bilimine giriş eğitimi alınmıştır.
- An Image Encryption Algortihm Based on Chaotic Selection of Robust Cryptographic Primitives adlı makale incelenmiştir.
- Python programlama dili eğitimi alınmıştır.
- React Native eğitimi alınmıştır.
- Microsoft Visio eğitimi alınmıştır.
- GitHub kullanım eğitimi alınmıştır.
- Postman Api Test eğitimi alınmıştır.
- Expo CLI eğitimi alınmıştır.

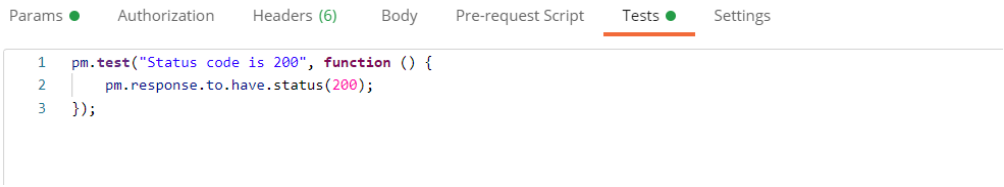
2.12 Test Planı

Postman ile Api Testi

Testler, API'n beklendiği gibi çalıştığından emin olmamıza, hizmetler arasındaki entegrasyonların güvenilir şekilde çalıştığını belirlememize ve yeni geliştirmelerin mevcut herhangi bir işlevi bozmadığını doğrulamamıza olanak tanır. API projemizde bir şeyler ters gittiğinde hata ayıklama sürecine yardımcı olması için test kodu da kullanılabilir.

Her bir isteğe tek tek ya da koleksiyonlara testler eklenebilir.

Aşağıdaki yer alan Şekil-19'da Postman ile gerçekleştirilen api testine ait bir örnek yer almaktadır.



Şekil-19 (Postman Api Test)

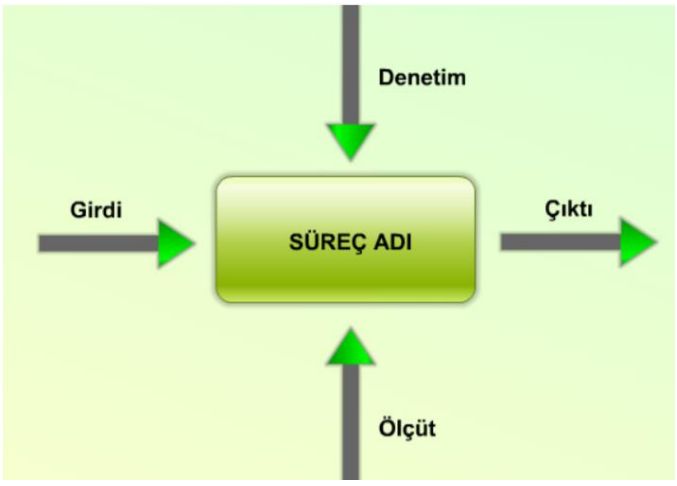
2.13 Bakım Planı

Bakım, işletime alınan yazılımın sağlıklı olarak çalışması ve ayakta kalabilmesi için yapılması gereken çalışmalar bütünü olarak tanımlanır.

Uygulamada çalışan bir yazılımın üç tür bakım gereksinimi bulunmaktadır:

- Düzeltici Bakım
- Uyarlayıcı Bakım
- En İyileyici Bakım

Bakım bölümüne ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı dikkate alınmıştır.



Şekil-20 (IEEE 1219-1998 Bakım Süreç Modeli)

2.14 Projede Kullanılan Yazılım/Donanım Araçlar

Şekil-21 (Kullanılan Yazılımlar ve Kütüphaneler)

Şekil-21'de kullanılan bazı yazılımlar ve kütüphaneler yer almaktadır.

3. SİSTEM ÇÖZÜMLEME

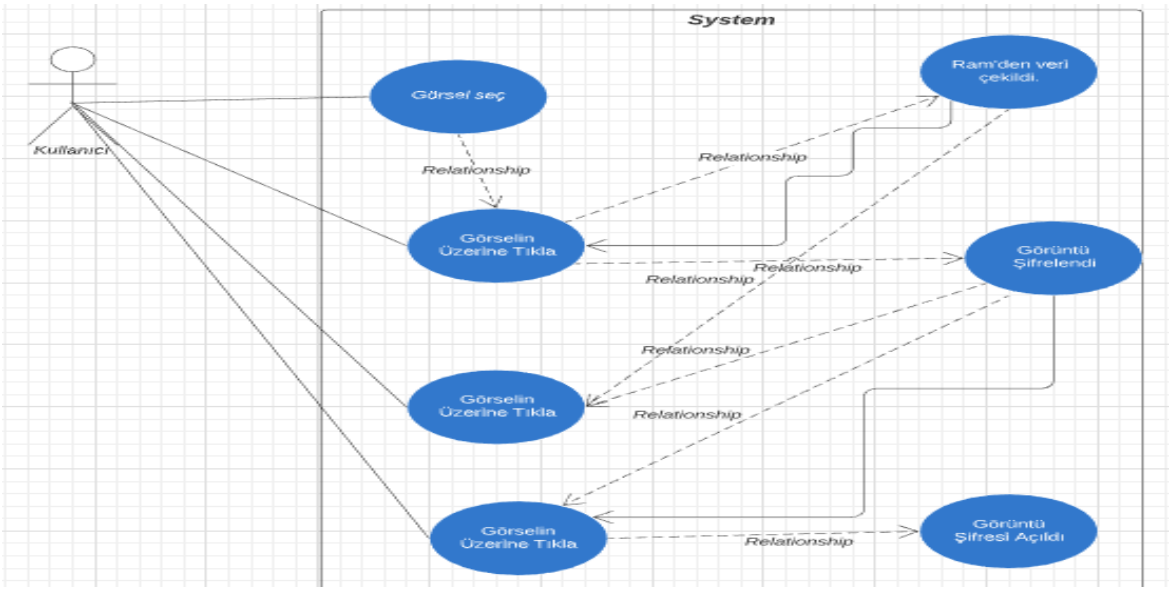
3.1 MEVCUT SİSTEMİN İNCELENMESİ

Genel olarak mevcut sistemin incelenmesi aşamasında oluşturulan projenin içyapısı incelenmiş ve ara yüz ile ekte sunulmuştur.

3.1.1 Örgüt Yapısı:

Örgüt yapısının bir sistem olarak amaçlan çerçevesinde belirlenmesi ve örgüt bileşenlerinin aynı işlevsel birimler olarak değil, örgüt amaçlarına olan katkılan çerçevesinde birbirleri ile ilişkili olarak düzenlenmiştir.

3.1.2 İşlevsel Model:



Şekil-22 (Use Case Diyagramı)

3.1.3 Varolan Yazılım/Donanım Kaynakları:

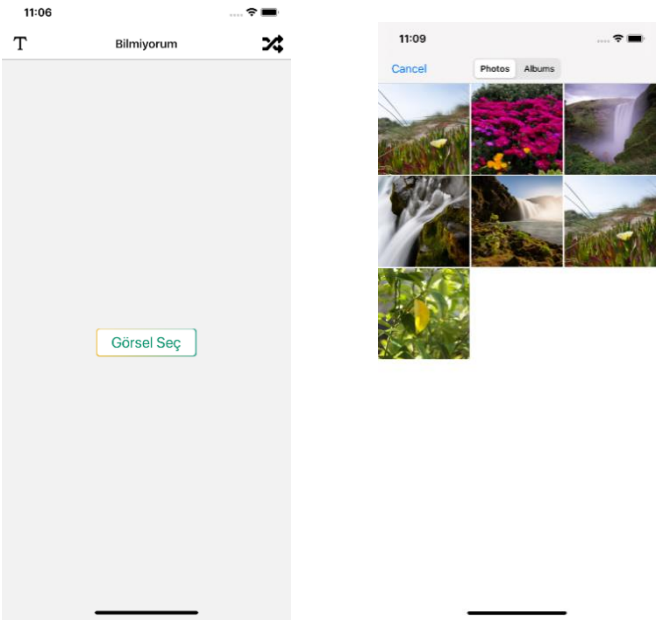
Yazılım Kaynaklar:

- Front – End: React Native
- Back - End: Python
- Back – End: Django

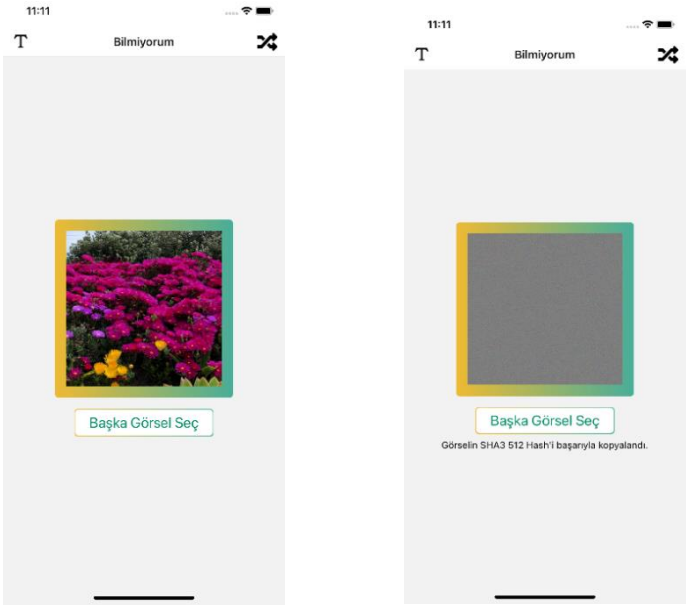
Donanım Kaynaklar:

- Apple
- Asus,
- Lenovo

3.1.4 Varolan Sistemin Değerlendirilmesi:



Şekil-23 (Görsel Seçimi)



Şekil-24 (Seçilen Görselin Şifreli Kopyalanması)

Front-End kısmında React Native ve Back-End kısmında da Python, Django kullanılarak yazılan kodlamalar ile istenilen proje başarılı bir şekilde gerçekleşmektedir. Expo uygulaması ile istenilen görsel seçilmekte olup, seçilen görselin de şifrelenmiş hali görünmektedir.

- SHA3 512
- SHA3 384
- SHA3 256
- SHA3 224

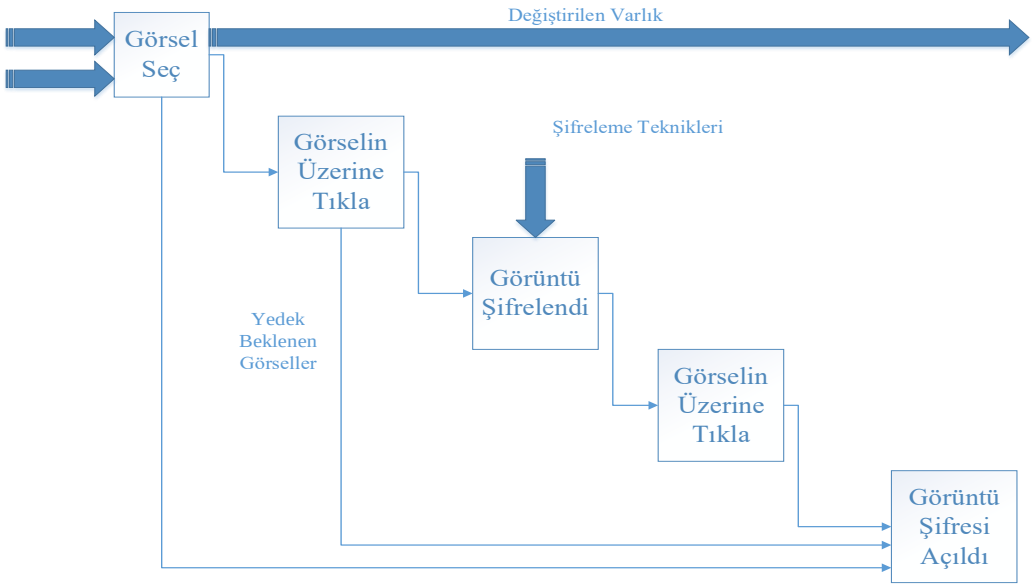
Şifreleme teknikleri kullanılmıştır.

3.2 GEREKSENEN SİSTEMİN MANTIKSAL MODELİ

3.2.1 Giriş

Mevcut sistem ile genel bakış açısıyla bakmak gerekirse; şifreleme teknikleri ile görselin şifrelenip tekrar eski haline getirilmesi söz konusudur.

3.2.2 İşlevsel Model:



Şekil-24 (İşlevsel Model)

3.2.3 Genel Bakış

İşlevsel Modelde de gösterildiği gibi kullanıcının istediği görsel uygulamadan seçilir. Seçilen görselin SHA3 şifreleme tekniği ile istenilen şekilde seçme imkanıyla görsel şifrelenir. Ardından şifrelenen görüntünün üzerine tıklandığı anda görselin tekrar eski haline dönmesiyle çalışma sonlanmış bulunmaktadır.

3.2.4 Bilgi Sistemleri/Nesneler

Kullanıcı

- Görsel seç,
- Görselin üzerine tıkla,
- RAM'den veri çek,
- Görüntü şifrelendi,
- Görselin üzerine tıkla,
- Görüntü şifresi açıldı.

3.2.5 İşlevlerin Sıra Düzeni

Şekil-25 (İşlevlerin Sıra Düzeni)

3.2.6 Başarım Gerekleri

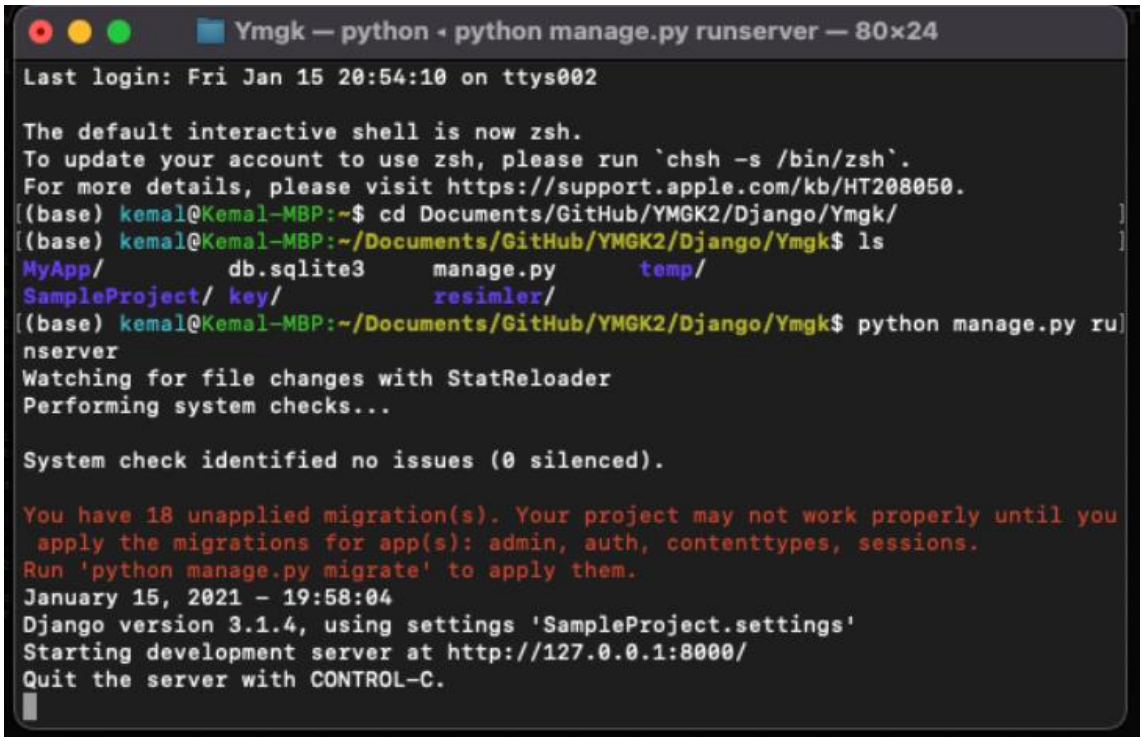
Mevcut sistemler incelendi ve mevcut sistemin eksiklerinden yola çıkılarak, sistemin başarımı için

- Sistemin sonuç üretim doğrulukları
- Tepki sürelerinin en aza indirilmesi
- Mali külfetin azaltılması
- Hile hata ve yanlışlıkların en aza indirilmesi
- Kullanım kolaylılığı
- Anlaşılabilirlik

temel gereklilikler olarak tespit edilmiştir.

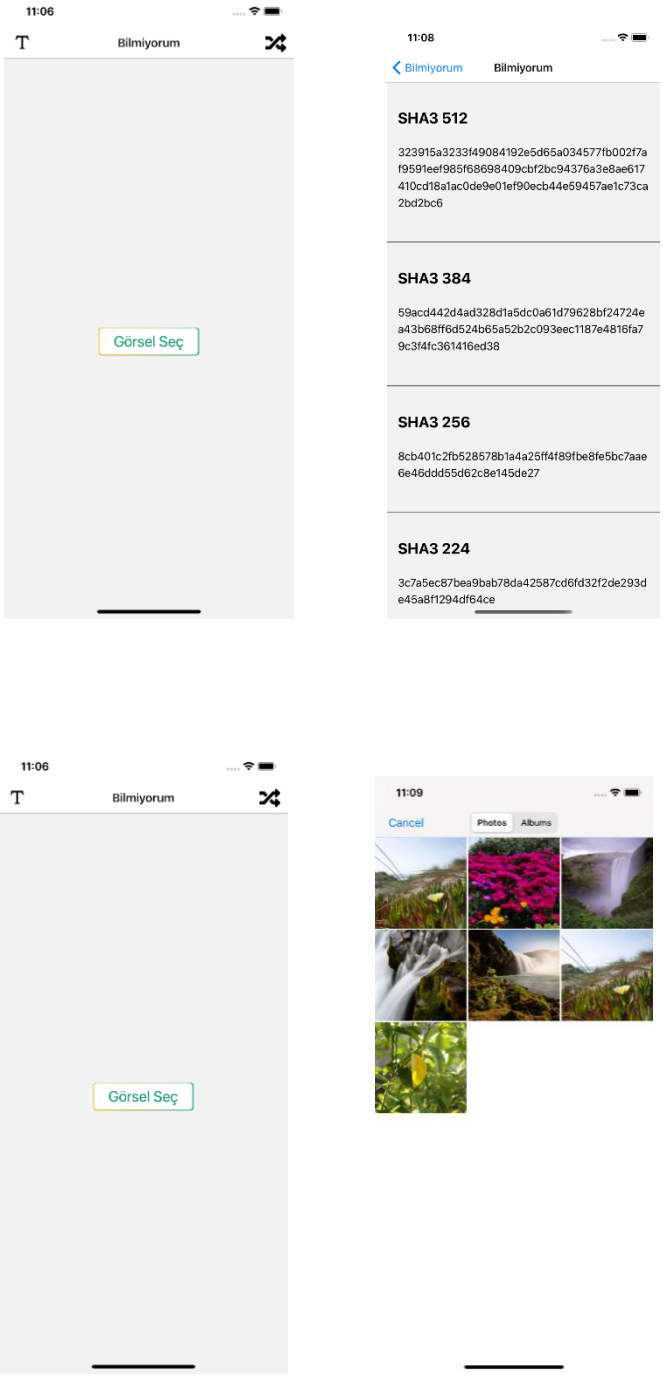
3.3 Arayüz (Modül) Gerekleri

3.3.1 Yazılım Arayüzü

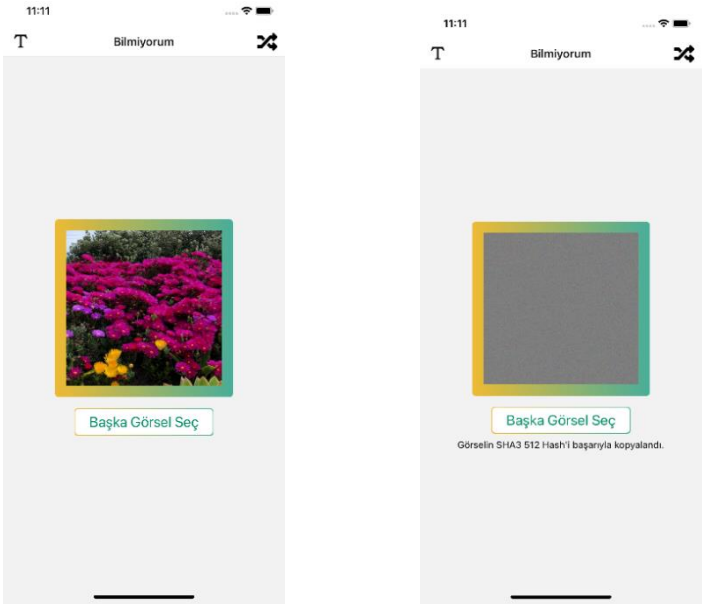


Şekil- 26 (Yazılım Arayüzü)

3.3.2 Kullanıcı Arayüzü



Şekil- 27 (Kullanıcı Arayüzü - I)



Şekil- 28 (Kullanıcı Arayüzü - II)

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

İhtiyaçlar dâhilinde ve mevcut beceriyi geliştirmek amacıyla ekip yetkinlikleri geliştirilmiştir. Ekip etkileşimini arttırmak amacıyla sık sık toplantılar yapılıp beyin fırtınaları yapılmıştır.

Kaynak kontrolü yapılıp fiziksel kaynakların planda yer aldığı gibi atanıp atanmadığı, kaynakların planladığı gibi kullanılıp kullanılmadığı süreç boyunca kontrol edilmiştir. Bunun için herhangi bir belge bulunmamakla birlikte yapılan çalışmalar videoda ve arayüzlerde sunulmuştur.

3.4.2 Eğitim Belgeleri

- Fatih Özkaynak - Kriptoloji bilimine giriş eğitimi alınmıştır.
- An Image Encryption Algortihm Based on Chaotic Selection of Robust Cryptographic Primitives adlı makale incelenmiştir.
- Python programlama dili eğitimi alınmıştır.
- React Native eğitimi alınmıştır.

- Microsoft Visio eğitimi alınmıştır.
- GitHub kullanım eğitimi alınmıştır.
- Postman Api Test eğitimi alınmıştır.
- Expo CLI eğitimi alınmıştır.

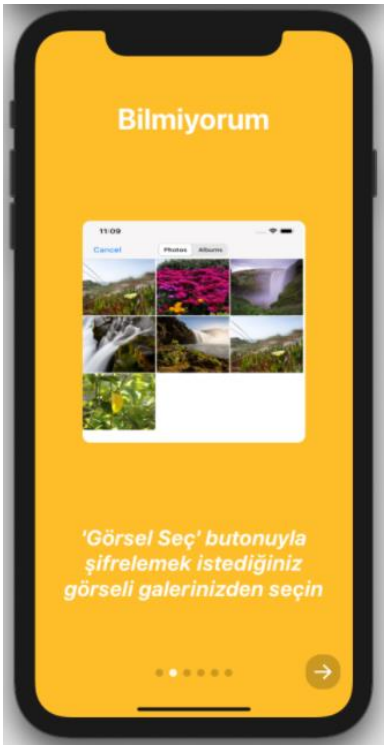
3.4.3 Kullanıcı El Kitapları

Dokümana eklenen ara yüzler kısmında kullanıcı ara yüz bölümünde kullanıcının uygulamayı nasıl kullanacağı belirtilmiştir.

Uygulamamız ilk açıldığında kullanıcıları tanıtım ekranı karşılamaktadır.

Şekil-29, Şekil-30, Şekil-31, Şekil-32, Şekil-33, Şekil-34'te tanıtım ekranları yer almaktadır.

Şekil- 29 (Tanıtım Ekranı - I)



Şekil- 30 (Tanıtım Ekranı - II)

Şekil- 31 (Tanıtım Ekranı - III)

Şekil- 32 (Tanıtım Ekranı - IV)

Şekil- 33 (Tanıtım Ekranı - V)

Şekil- 34 (Tanıtım Ekranı - VI)

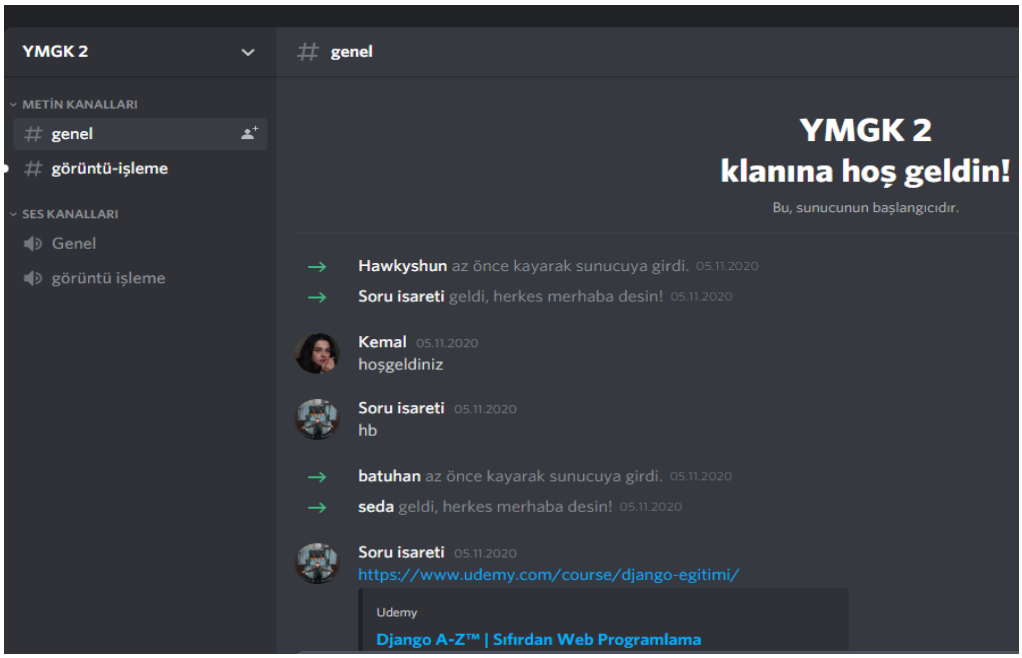
4.1 Genel Tasarım Bilgileri

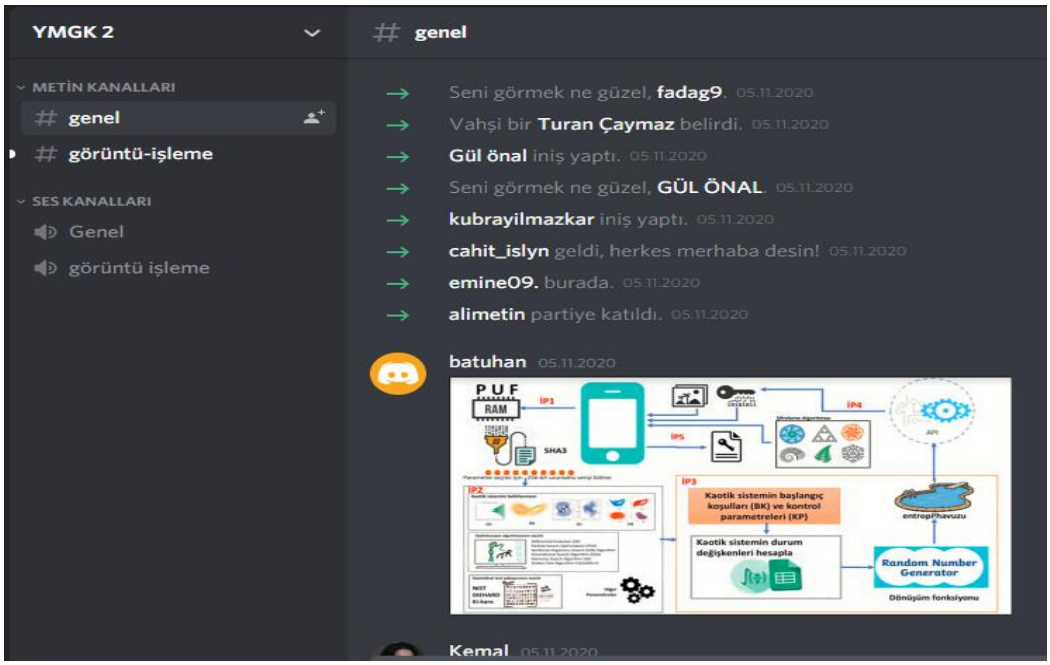
4.1.1 Genel Sistem Tanımı

Şekil-35 (Genel Sistem Tanımı)

GereksinimAnalizi

Projeye başlarken ilk toplantılarımızda kullandığımız soru cümleleri şunlardı: “bize ne lazım?”, “bizden ne yapmamız isteniyor?”,” neler kullana biliriz?” ve daha bir sürü soru zaman geçerken bulunan cevaplar karşımızda yer almaya başladı. Kim hangi dil biliyor ve program hangi dil ile yazılacaktı. Zorlanacağımız yeri düşünüp kimin bilip bilmediğini konuşup ortak karar ile Pyhton dilini kullanma kararı alındı. İş paketleri incelendi. Bize bir platform gerekiyordu bu yüzden React Native kullanmak mantıklı olup her türlü platform için destek sağlanmak istenildi. Projeyi anlamak ve kimin hangi iş paketinde yer alması gerektiği biraz zaman aldı. Bu dağılım da güzel bir şekilde yapıldıktan sonra çalışmalar başladı.





Şekil-36 (Proje Discord Grubu)

İşlevsel Belirtimler

Öncelikle bizden istenenin ne olduğunu anlatayım. Ram den alınan bir resmin (image) şifrelenmesi ve karıncalı bir biçimde iletilmesi iletilen şifreli resmin anahtar yardımıyla tekrardan görüntüsüne kavuşulması istenmekte. Güvenliğin en önemli olduğu zamanlarda

sözleşmeler aracılığıyla gizlilik ihlalleri edildiği için yeni programlara ihtiyaç duyulmaktadır. Bizde yeni bir alternatif ile karşınızda yer almaktayız. Biraz detaylandırmak gerekirse projenin içeriği 5 temel iş paketinden oluşmaktadır. Ilk resim sha3 algoritmasından geçip bir dizi halini alıyor. İkinci iş paketinde bir denklem ile random değerler atanıyor. 3 iş paketinde xor lama işlemi yaparak şifrelenme sağlanıyor. İş paketi 4 ise API kısmı olup paketler arasında bağlanma problemini çözüyor. Son iş paketi ise şifreli veriyi gönderip şifresini çözüyor.

Tasarım

Şablon olarak aşağıdaki süreçler izlenerek tasarım işlemi yapıldı.

Süreç tasarımı: Tüm iş paketleri için uygun yöntem ve tekniklerin belirlenme aşaması.

Arayüz Tasarımı: kullanıcıyla iletişim kurmamızı sağlayacak olan aşamanın yapılması.

Yapısal tasarım: Artık projenin ortaya çıkma oluşum meydana geliş aşaması.

4.1.2 Varsayımlar ve Kısıtlamalar

Varsayılan değerler bulunmakta olup bunlar sırayla yazmaktadır.

- Resim gizlilik için programı kurmak
- Kullanıcı profili tutmamaktayız
- Herhangi bir bilgi veri tabanına işlememekteyiz
- Güvenlik bizim için çok önem arz ettiği için veri tabanı kullanıcı bilgi kaydı bulundurmamaktayız.

4.1.3 Sistem Mimarisi

Sistem mimarisi en basit ve anlaşılır yol olan akış şeması ile yukarda göstermekteyim. Kısaca anlatayım. Yüzeysel olarak program resim şifreleme gönderme ve resmi çözme üzerine temellerden oluşmaktadır. Bu temeller 5 iş paketi şeklinde

bölünmüş olup işlemler sırayla işlenmektedir. İlk iş paketinde SHA3 modelleri işlenmiş olup dizin haline geliyor. İkinci iş paketinde verilmiş olan bir denklemden rast gele sayılar üretip üçüncü iş paketine gönderilmektedir. Üçüncü iş paketi XOR lama yaparak şifrelemekte ve dördüncü iş paketine göndermekte. Dördüncü iş paketi API kısmı olup iş paketleri arasındaki bağlantıları yapmaktadır. Beşinci iş paketimiz ise şifrelenip gönderilen görselin anahtar aracılığı ile eski haline getirmektedir. Sistemimiz bu şekilde ilerlemektedir.

4.1.4 Dış Arabirimler

4.1.4.1 Kullanıcı Arabirimleri

Kullanıcı arabirimleri ilk başında programın giriş ekranı bulunmakta.bu ekranda görsel seçme işlemi bulunmakta daha sonrasında üzerine tıklanarak resmin şifrelenmesi sağlanıyor ve hash kopyalanıyor. Görsel çözüme ulaşıyor galeriye kaydediliyor. Anahtar siliniyor.

4.1.4.2 Veri Arabirimleri

Ara birimler olarak kütüphanelerimiz bulunmakta Django Framework bunlar bir tanesi olup API kısmında kullanılmış bulunmaktadır. React Native kullanılmış olup platform çoğunluğumuzu arttırmış bulunmaktayız.

Şekil-37 (Veri Arabirimleri)

4.1.5 Testler

Genel hatlarıyla testlerimiz iki aşamada gerçekleştirilmesi planlanmaktadır.

Alfa Aşaması: Kullanıcıların sisteme katkıda bulunması ve test etmesi ile yapılmaktadır.

Beta Aşaması: Kullanıcı, geliştirilen sistemi kendi özel cihazında kullanarak bilgilendirme yapacaktır.

4.1.6 Performans

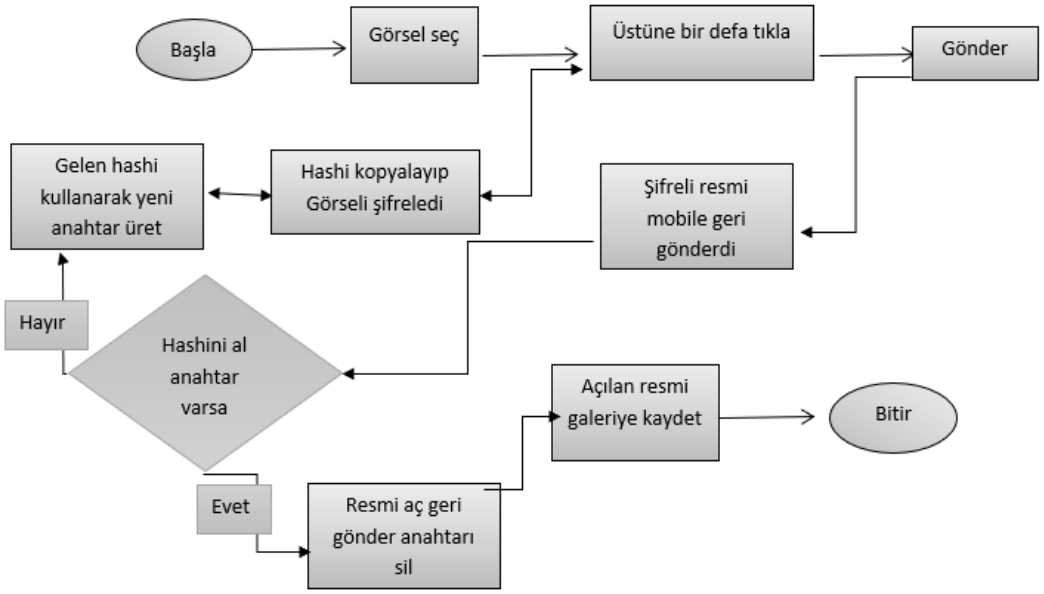
Performansımız çalışma başlatıldığı itibariyle gayet hızlı bir şekilde saniyeler eşliğinde çalışmaktadır. Bizi etkileyecek olan kısım yapılan güncellemeler ve işlenecek olan resmin boyutu dur. Eğer çok çok büyük bir boyuta sahip ise bu gecikme artışta bulunabilir.

4.2 Süreç Tasarımı

Veri, yapı ve arayüz tasarımından sonra yapılır. İdeal şartlarda bütün algoritmik detayın belirtilmesi amaçlanır. Ayrıca süreç belirtiminin tek anlamı olması gerekir ve değişik şahıslar tarafından farklı yorumlanmamalıdır.

4.2.1 Genel Tasarım

Genel tasarım esnasında telefondan bir hash değeri okutuyoruz ve resmi alıp backende gönderiyoruz. Backenden gelen hashi rasgele sayıda sonuna kendisini ekleyerek uzatılmaktadır. Oluşan bu dizinin uzunluğunda yeni bir diziyi oluşturuyoruz. İkisini XORluyoruz oluşan XORlanmış dizinin içerisinden resim boyutunda bir matris oluşturuyoruz. Sonraki işlemde rasgele resim boyutunda bir matris oluşturuyoruz. İkisini xorluyoruz bu bizim resmimizi şifreleyeceğimiz anahtar oluyor, bu anahtar ile resmi xorluyoruz. Anahtarın hashini alıp backende kaydediyoruz. Şifrelenmiş resmi geri döndürmekteyiz. Bu şifrelenmiş resim gelirse önce açıyor sonra anahtarı siliyor ve açılmış resmi geri döndürüyor. Bu işlemler yapılmaktadır.



Şekil-38 (Genel Tasarım)

4.2.2 Kullanıcı Profilleri

Kullanıcıdan beklentimiz sadece mobil uygulamayı kurması ve galeriye erişime izin vermesi başka bilgi almıyoruz ve kullanıcı profili oluşturmuyoruz.

4.2.3 Entegrasyon ve Test Gereksinimleri

Entegrasyon aşamasında gerçekleştireceğimiz tek olay, kullanıcının programı kurduktan sonra galeriye erişim izni vermesidir.

Test gereksinimleri kullanıcı tarafından sağlanmaktadır. Çünkü yapılmış olan program kullanıcının resim göndermek istemesiyle oluşmuş olup test işleminde kullanıcıya ihtiyaç duymaktadır.

4.3 Ortak Alt Sistemlerin Tasarımı

4.3.1 Güvenlik Alt sistemi

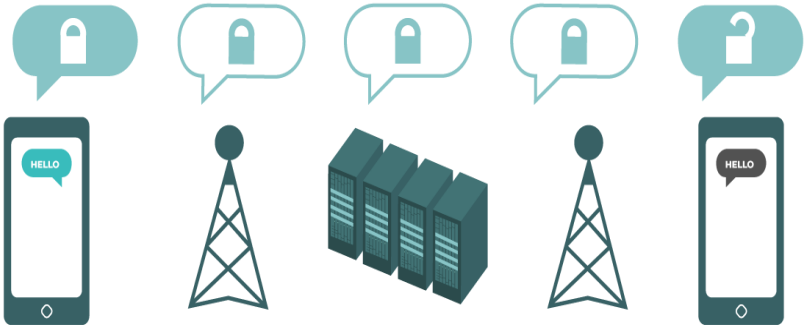
Yazılım sistemlerinin güvenilirliğe ilişkin ölçüleri, kullanıcıların gereksinimlerini karşılayacak şekilde ortaya koymak ve güvenilirliğin hesaplanmasına yönelik verileri

toplamak, istatistiksel tahminde bulunmak, ölçütlerin tespit edilmesi, yazılıma ait mimari özelliklerin belirlenmesi, tasarım, geliştirme ve bunlara yönelik çalışma ortamlarının belirlenmesi ve modellenmesini kapsamaktadır. Bu program güvenlik amaçlı olup şifreleme ile ilgilenmektedir. Bu sebep neticesinde bu konularda daha dikkat edilmesi gereksinimi ortaya çıkmaktadır.

- Model seçme ve düzenlemeye yönelik faaliyetleri esnasında yapılan işlem uygun hedeflerin tespit edilmesi bulunmaktadır.
- Hata ve aksaklıkların analiz edilmesi için gereken uygun verilerin tanımlanmasıdır. Örneğin, problemleri önemine göre sınıflandırma işlemi, hatalar arası ortalama süreyi bulmak, hata nedenlerini araştırmak, hataları bulmaya yönelik test verilerine karar vermek önerilmektedir.
- Belirtilen hedeflere yönelik veriler modellenmiştir.
- Yazılım geliştirme sürecinin modellenmesi, problem ile karşılaşılıp, test sürecine başlamak ve model doğrulama işlemlerini gerçekleştirmek.
- Güvenilirlik tahminde bulunma modelinin seçilmesini sağlamak.
- Güvenilirlik modeli tarafından kullanılacak olan parametrelerini tespit etmek.
- Verilen bir noktayı kullanarak gelecekteki olası problem olacak durumlar hakkında tahmin yapabilmek.
- Tahmin edilen olası problemleri oranları ile gerçekleşen değerleri karşılaştırmak.

4.3.2 Veri Dağıtım Alt sistemi

Bu projede eri dağıtım sistemi bulunmamakta. Veri dağıtım sistemi, güvenliği zedeleyecek olup kullanılmamaktadır. Bizim verimiz tek kişiye iletilmekte ve şifreli vaziyette bulunmak koşuluyla gönderilen kullanıcının anahtarı bilmesi dahilinde görsele erişim sağlanmaktadır.



Şekil-39 (Veri Dağıtım Alt Sistemi)

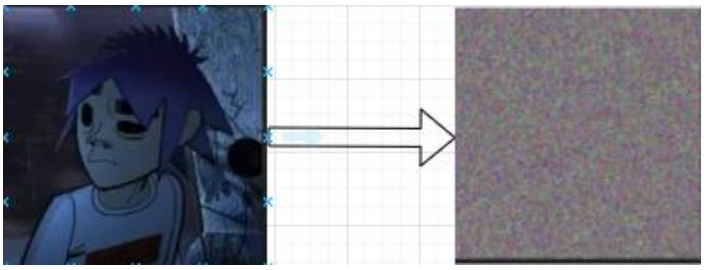
4.3.3 Yedekleme ve Arşivleme İşlemleri

Program işlevinin kullanıcı taraflı kayıt işlemi bulunmadığından dolayı afet gibi felaket durumları sürecinde kullanıcılar tarafından kaybedilecek bir durum görülmemektedir. Bu sebepten dolayı sistemin sadece arka planında yani yazılım kısmında kod kaybı sağlanmaması için beklenmedik durumlara uygun olarak yapılan işlemler başka sistemler tarafından kayıt altına alınıp sürekli zarar gelmemesi için gün gün kayıtlar yapılarak depolanmaktadır.

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1 Giriş

Gerçekleştirim çalışmalarıyla birlikte ele aldığımız görüntünün şifrelenme yoluyla daha karmaşık bir çıktı elde etme hedeflenmiştir. Böylelikle elde olan fiziksel modellerin çalışan yazılım biçimine dönüştürme işlemi yapılacaktır.



GİRİŞ **ÇIKIŞ**

5.2 Yazılım Geliştirme Ortamları

Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

- Programlama Dili
- Hazır Program Kitapçıkları

5.2.1 Programlama dilleri

Sistemde kullanılan başlıca programlama dillerini belirtmiştik ancak burada bir kez daha belirtmekte fayda var. Genellikle görüntü işleme ve kriptoloji alanında baskın olarak kullanılan Python dili ve mobil uygulama için React Native kullanılmıştır.

5.2.2 Hazır Program Kütüphane Dosyaları

Çoğunlukla iş planı 5 (IP 5) de yer alan ve proje planında daha öncede bahsettiğimiz **PILLOW, NUMPY, OPEN CV, HASLİB, OS, RANDOM, TKINTER, SUBPROCESS, SYS** kullanılmıştır.

5.3 Kodlama Stili

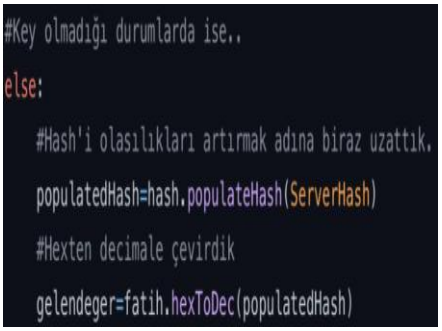
Şekil -40

Şekil-41

5.3.1 Açıklama Satırları

Açıklama satırlarını karmaşık işleri açıklamak ve daha sonra geliştirecek kişilerin daha rahat anlayabilmesi için kullanıldı.

Şekil-42



Şekil-43

5.3.2 Kod Biçimlenmesi

Kod biçimlemesine değinmek gerekirse alt alta oluşan kodlarda indexler kullandık ve iç içe bir biçimde hiyerarşi oluşturduk.

5.3.3 Anlamlı İsimlendirme

Kullanacağımız bir veri tabanı olmuş olsaydı anlamlı isimlendirme yapardık fakat veri tabanı kullanmadığımız için bu konu üzerinde bir yapılanmaya gerek duymadık.

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 yazıcığımıza tek bir döngüyle bu sorunların üstesinden geldik.

5.4 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.

Projemizin herhangi bir internet platformunda bulunmadığından özünde grup üyelerinin katkılarıyla hazırlandığından ötürü kullandığımız verilerin (fotoğraf, görüntü vs.) tamamıyla bizlere ait oluğunu belirtmekteyim. Bununla birlikte kullandığımız sunucuyla da denemeler sonucunda herhangi bir sorun sıkıntı yaşanmamaktadır.

5.4.1 Farklı Olağandışı Durum Çözümleme Yaklaşımları

Bu gibi durumlarda kullandığımız postman aracılığıyla oluşabilecek herhangi bir sorunun sunucu kaynaklı olduğunu belirlemekteyiz. Sorunun giderilmesi için tekrardan farklı bir görüntü üzerinden deneme yapılarak işlemimizi gerçekleştirmekteyiz.

5.5 Kod Gözden Geçirme

Kodlama bölümünde daha öncede belirttiğimiz gibi tamamıyla grup üyelerine ait olduğundan dolayı kullandığımız yazılım aracı yardımıyla tekrar tekrar gözden geçirilmiş olum herhangi bir hata almamaktayız. Bununla birlikte gözlemci kodlama kısmını kendi istekleri için de kullanabilmektedir.

5.5.1 Veri Kullanımı

Proje için kullandığımız görüntü verilerinin farklı fotoğraflar üzerinden deneme yoluyla çözümlenmiştir. Tüm fotoğraflar için giriş kısmında elimizde bulunan görüntüler giriş olarak belirtip çıkış durumunda karıncalanma olarak verilmiştir.

- Tüm görüntüler tek amaçla kullanılmıştır.
- Belirli bellek ataması yapılmıştır
- Görüntüler giriş kısmında anlamlı olup çıkış kısmında bu bütünlüğünü görsel olarak kaybetmiştir.

5.5.2 Sunuş

Açıklama Satırları kısmında belirttiğimiz gibi açıklama satırları kullanılmaya, kodun fazla satıra taşması durumunda kod bütünlüğü bozulmadan alt satıra geçmeye özen gösterildi. Kodlarda karmaşıklığı azaltacak şekilde parantezler kullanıldı. Kodlar arasında anlaşıla bilirliği arttırmak için döngüler ve tanımlamalar arasında boşluklar bırakıldı.

6. DOĞRULAMA VE GEÇERLEME

6.1.Giriş

Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlenmesi, 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.

Doğrulama ve geçerleme işlemleri temel olarak çeşitli düzeylerde sınama, gözden geçirme, denetim ve hata giderme süreçlerinden oluşur.

Şekil-44 (Doğrulama & Gerçekleştirim Şeması)

6.2.Sınama Kavramları

Şekil-45 (Sınama Kavramları)

Şekil-45'te görülen sınama kavramları aşağıda açıklanmıştır.

Birim Sınama: Bağlı oldukları diğer sistem unsurlarından tamamıyla soyutlanmış olarak birimlerin doğru çalışmalarının belirlenmesi amacıyla yapılır.

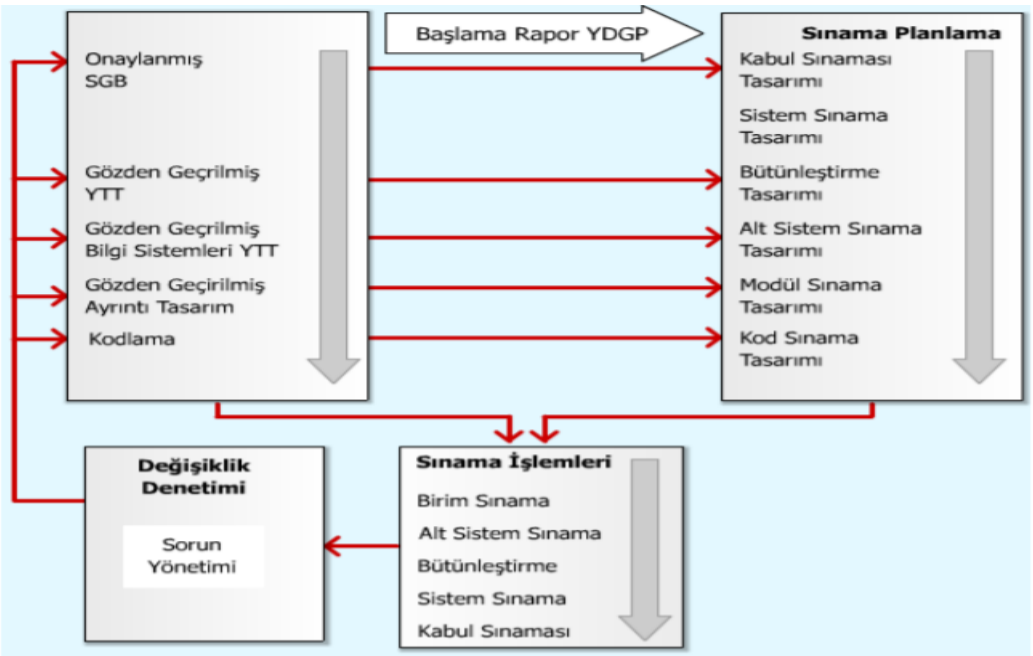
Birimler, ilişki yapıtaşlarının bütünleştirilmesinden oluşurlar.

Alt Sistem Sınama: Modüllerin birleşmesiyle ortaya çıkar. Yine bağımsız olarak sınama yapılmalıdır. En çok hata arayüzlerde bulunmaktadır, arayüz hatlarına yönelik sınamalara yoğunlaşmalıdır.

Sistem Sınama: Sistemin bütün olarak sınanması yapıldı ve programın eksiksiz olduğu onaylandı.

Kabul Sınama: Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha onaylandı.

6.3.Doğrulama ve Geçerleme Yaşam Döngüsü



Şekil-46 (Yaşam Döngüsü)

Şekil-46'ta görülen yaşam döngüsü verilmiştir.

Yazılım belirtimlerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtimleri tutarlı olarak betimler durumda olduğunun doğrulanması işlemidir.

Proje süresince her bir etkinlik ürününün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitenin tekrarına sebep olmaktadır.

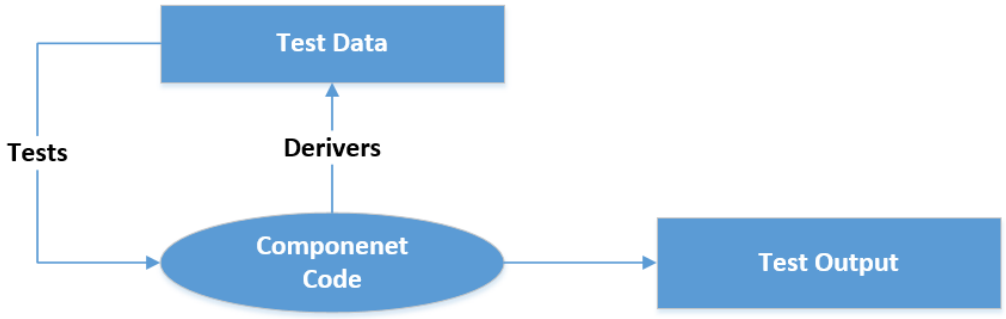
6.4. Sınama Yöntemleri

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 Beyaz Kutu Sınaması

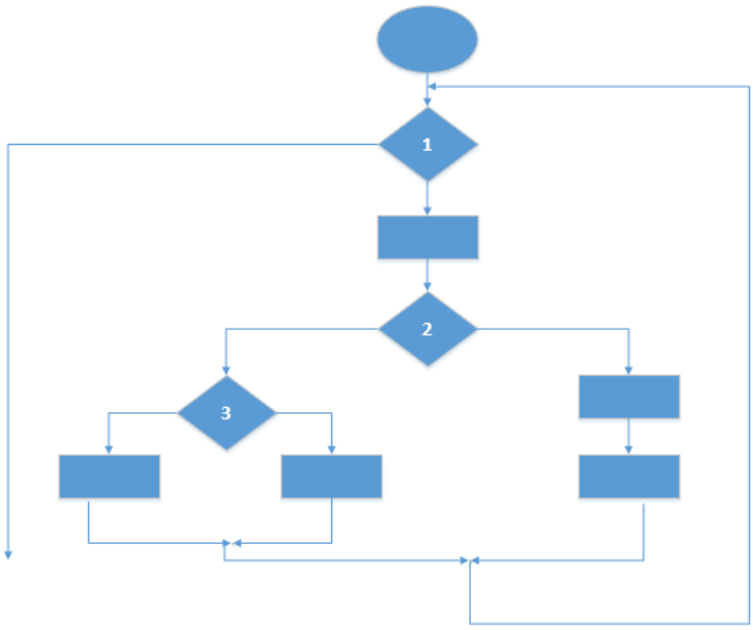
Denetimler arasında:

- Bütün bağımsız yolların en azından bir kere sınanması,
- Bütün mantıksal karar noktalarında iki değişik karar için sınamaların yapılması,
- Bütün döngülerin sınır değerlerinde sınanması,
- İç veri yapılarının denenmesi gerçekleştirildi.

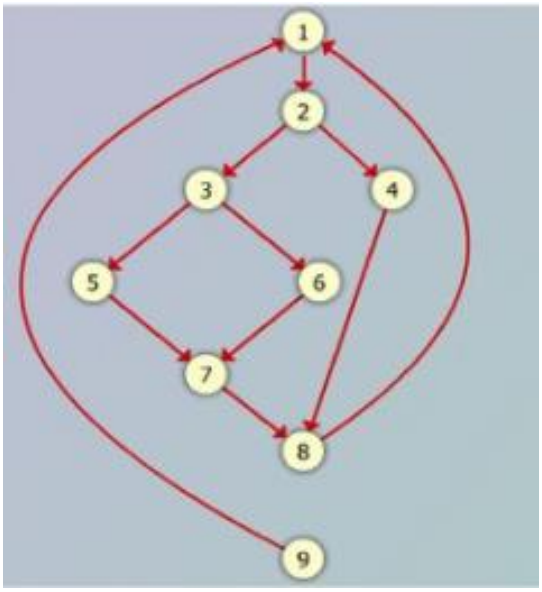


Şekil-47 (Beyaz Kutu Sınaması)

6.4.2 Temel Yollar Sınaması



Şekil-48 (Temel Yollar Sınaması İş Akış Şeması)



Şekil-49 (Temel Yollar Sınaması Grafik Diyagramı)

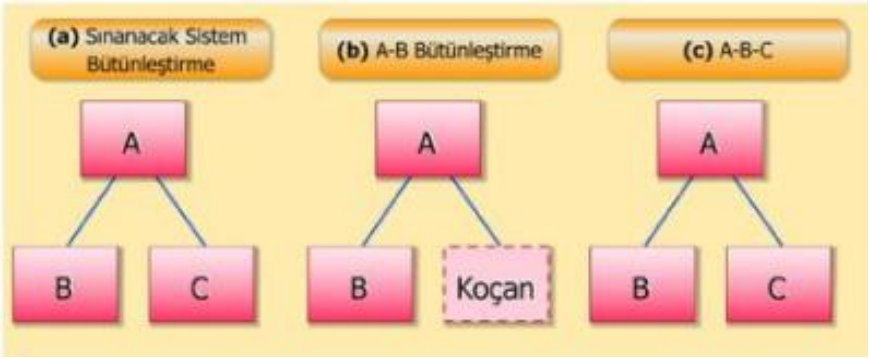
Şekil-43'de görüldüğü gibi sonuçların doğruluğunu ve normal şartların üstünde zorlandığında dayanıklılık sınırını ortaya çıkarır.

6.5.Sınama ve Bütünleştirme Stratejileri

Sınama stratejisi, bütünleştirme stratejisi ile birlikte değerlendirilmektedir. Bazı sınama stratejileri bütünleştirme dışındaki tasaları hedefleyebilir.

6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme

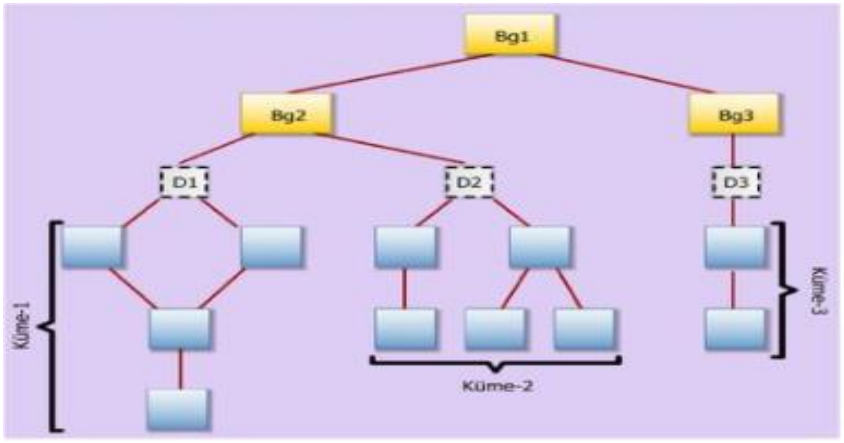
Yukarıdan aşağı bütünleştirmede, önce sistemin en üst düzeylerinin sınanması ve sonra aşağıya doğru olan düzeyleri, ilgili modüllerin takılarak sınanmaları söz konusudur. En üst noktadaki bileşen, bir birim/modül/alt sistem olarak sınandıktan sonra alt düzeye geçilmelidir. Ancak bu en üstteki bileşenin tam olarak sınanması için alttaki bileşenlerle olan bağlantılarının da çalışması gerekir. Genel hatlarıyla özetlemek gerekirse şu mantıkla sitem sınaması yapıldı.



Şekil-50 (Bütünleştirme Sınaması)

6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme

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



Şekil-51 (Aşağıdan Yukarıya Sınama ve Bütünleştirme)

6.6 Sınama Planlaması

Test raporu hazırlanırken şu özellikler mutlaka planda belirtilmelidir;

Test planı kimliği: Test planının adı veya belge numarası

Giriş: Test edilecek yazılımın elemanlarının genel tanıtım özetleri. Ayrıca bu plan kapsamı ve başvurulan belgeler. Kısaltmalar ve terim açıklamaları bu bölümde bildirilmelidir.

Test edilecek sistem: Sistemde bileşenleri sürüm sayıları olarak sıralar ve sistemin özelliklerini bileşenlerini ve nasıl kullanıldıkları açıklanmalıdır. Ayrıca sistemde test edilmeyecek parçalar belirtilmelidir.

Test edilecek ana fonksiyonlar: Sistemin test edilecek ana fonksiyonlarının kısa bir tanıtımı yapılmalıdır.

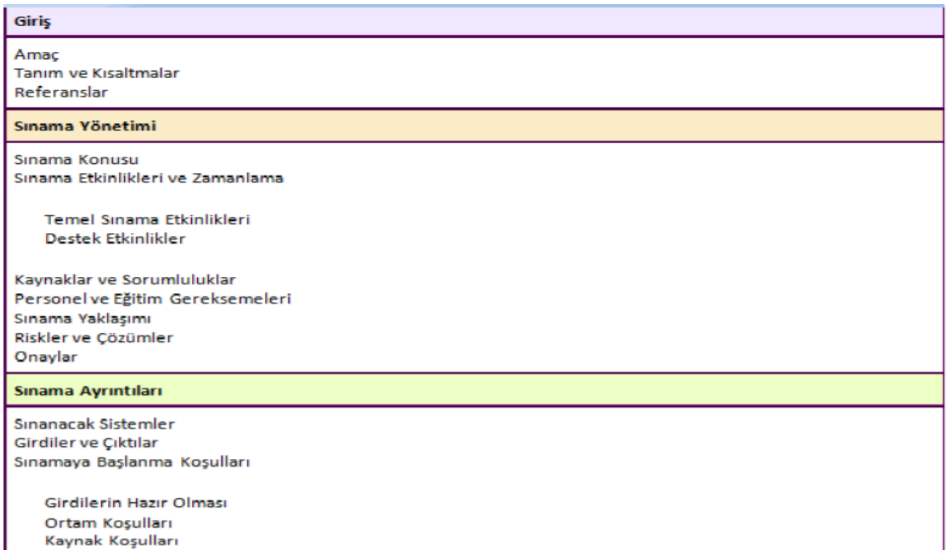
Test edilmeyecek ana fonksiyonlar: Sistemde test edilmeyecek fonksiyonları ve bunların neden test edilmedikleri açıklanacaktır.

Geçti/Kaldı Kriterleri: Bir test sonucunda sistemin geçmiş veya kalmış sayılacağını açıklanmalıdır.

Test dokümanı: Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır.

Sorumluluklar: Hangi kişilerin nelerden sorumlu olduğu ve test takım lideri bilgileri mutlaka raporda belirtilmelidir.

Riskler ve Önlemler: Test planında varsayılan ve olası yüksek riskli durumları belirtir ve bu durumların olması durumunda, etkilerinin en aza indirilebilmesi için alınması gereken önlemleri açıklar.



Şekil-52 (Aşağıdan Yukarıya Sınama ve Bütünleştirme)

6.7 Sınama Belirtimleri

Sınama belirtimleri, bir sınama işleminin nasıl yapılacağına ilişkin ayrıntıları içerir. Sınama raporları, sınama bitiminde imzalanır ve yüklenici ile iş sahibi arasında resmi belge niteliği taşır.

Sınama planları;

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

Şeklindeki bilgileri kapsamaktadır.

Sınama verilerinin elle hazırlanması çoğu zaman kolay değildir ve zaman kaybetmeye neden olabilir. Bu durumda, otomatik sınama verisi üreten programlardan yararlanılabilir. Sınama senaryoları, yeni sınama senaryosu üretebilmeye yardımcı olacak biçimde hazırlanmalıdır. Zira sınama belirtimlerinin hazırlanmasındaki temel maç, etkin sınama yapılması için bir rehber oluşturur.

Sınama işlemi sonrasında bu belirtimlere,

- Sınamayı yapan,
- Sınama tarihi,
- Bulunan hatalar ve açıklamaları

Şeklindeki bilgiler eklenerek sınama raporları oluşturulur.

Sınama raporları, sınama bitiminde imzalanır ve yüklenici ile iş sahibi arasında resmi belge niteliği oluşturur.

6.8 Yaşam Döngüsü Boyunca Sınama Etkinlikleri

Bütün bu etkinlikleri bir hiyerarşi altında incelemek gerekirse:

- Planlama aşamasında genel planlama sınaması gerçekleştirilir. Bu olan tüm planların basit bir ön hazırlığı niteliğindedir.
- Çözümleme aşamasında sınama planı alt sistemler bazında ayrıntılandırılır.
- Tasarım aşamasında sınama plana detaylandırılır ve sınama belirtimleri oluşturulur. Bu oluşumlar daha sonra eğitim ve el kitabında kullanılır.
- Gerçekleştirim aşamasında teknik sınamalar yapılır sınama raporları hazırlanır ve elle tutulur ilk testler yapılır.

Kurulum aşamasında sistemle ilgili son sınamalar yapılır ve sınama raporları hazırlanır.

7. BAKIM

7.1. Giriş

Sistemin tasarımı bittikten sonra artık güncellemelerde sistemin bakıma sokulması gerekir. Bakım bölümüne ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı baz alarak alınmıştır.

7.2. Kurulum

Kurulum sistemi açısından sadece kullanıcının indirip kurması haricinde özel bir şey istenmemektedir. React Native kullanıldığı için her platformda aktif olarak çalışır.

7.3. Yerinde Destek Organizasyonu

Yerinde desteklenecek bağlantı sistemleri nedeniyle yapılacak pek bir şey bulunamıyor. Kullanıcıların indirdikleri platformdan yorum aracılığı ile sorunlarını bildirmeleri dahilinde güncellemeler yapılarak geri dönüşler sağlanacaktır.

7.4.Yazılım Bakımı

Şekil-53 (Yazılım Bakım)

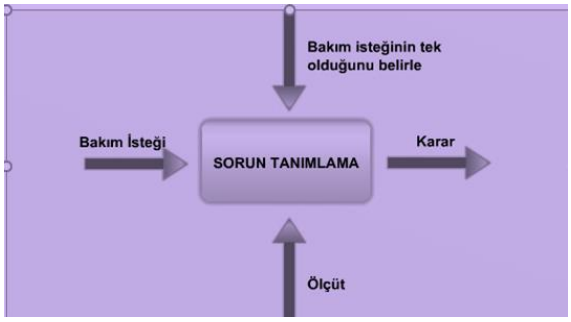
7.4.1 Tanım

Düzeltici bakım	Çalışan bir yazılımda her an bir problemle karşılaşma ihtimali vardır. Düzeltici bakım aslında önleyici bakım yaptırılmamasından dolayı meydana gelen plansız arıza olarak nitelendirilebilir. Düzeltici bakım, arızanın acil durum bakımına dönüşmemesini sağlayan önceki adımdır.
Uyarıcı bakım	Uygulamaya alınmış yazılımlar kurumların, şirketleri veya kişilerin günlük hayattaki işlerini bilgisayar ortamında takip etmesini sağlayan araçlardır. Gün geçtikçe iş süreçlerinde yeni gereksinimler veya var olan gereksinimlerin iptali söz konusu olabilmektedir.
En iyileyici bakım	Uygulama yazılımının performansının zamanla arttırılması gerekebilir. Bu tür bakımlar "en iyileyici bakım" olarak adlandırılır.

Şekil-54 (Tanım)

7.4.2 Bakım Süreç Modeli

Bakım süreç modeli yukardaki yapılan işlemlerin tümünün baştan yapılması demek bunları adım adım bir inceleyelim.

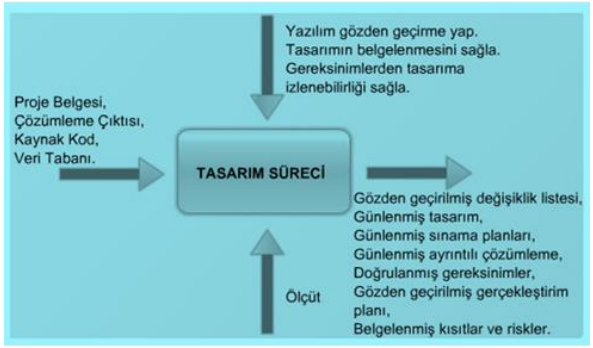


1.sorun tanımlama: Problem ney bakıma neden ihtiyaç duyuluyor bunun sonuçlandığı aşamadır.

Şekil-55 (Bakım Süreç Modeli – Sorun Tanımlama)

2.Çözüm süreci: sorun tanımlama aşamasında ortaya çıkan problemleri ilk olarak kağıt ortamına geçirerek çözümleme aşamadır. Görsel olarak elimizde problemin neden sonuç ilişkisi.

Şekil-56 (Bakım Süreç Modeli – Çözümleme Süreci)

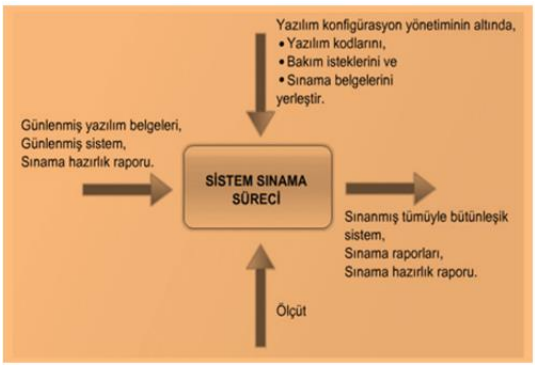


3.Yazılım süreci: yazılım güncelleştirme işleminin olaylar doğrultusunda işleme geçmeye başladığı süreç.

Şekil-57 (Bakım Süreç Modeli – Tasarım Süreci)

4.Gerçekleştirim süreci: Tasarımı yapılan sistemin meydana gelme sürecidir.

Şekil-58 (Bakım Süreç Modeli – Gerçekleştirim Süreci)

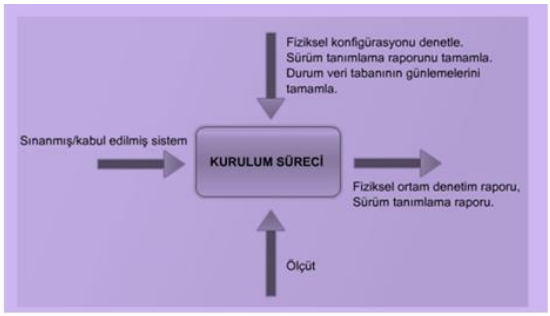


5.Sistem sınama süreci: tekrardan tasarlanan sistemin yapılanlar incelenip sınandığı süreçtir.

Şekil-59 (Bakım Süreç Modeli – Sistem Sınama Süreci)

6.kabul sınaması süreci: Yaparken test etiğimiz programın kullanıcı tarafından test edilme sürecidir.

Şekil-60 (Bakım Süreç Modeli – Kabul Sınaması Süreci)



7.kurulum süreci: Kabul sınamasından geçen projenin tekrardan kullanıcıların hizmetine sunulma aşaması.

Şekil-61 (Bakım Süreç Modeli – Kurulum Süreci)

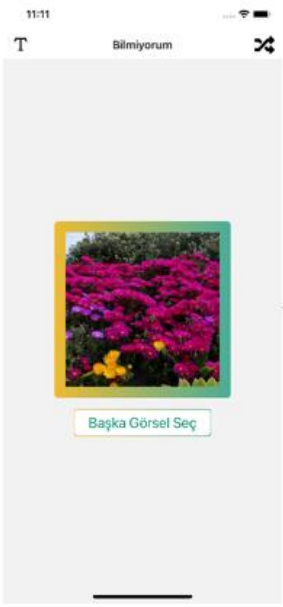
8. SONUÇ

Sonuç olarak sistem hayata geçirildiğinde ele alınan görüntünün mevcut görüntüler üzerinden seçilip Hash fonksiyonuyla şifrelenip yeni görüntüsü elde edilmektedir.

Şekil -1

Şekil -2

Şekil -3



Şekil -4

Şekil -5

Kaynakça

https://github.com/brix/crypto-js
https://github.com/emn178/js-sha3
https://momentjs.com/
https://github.com/react-native-cameraroll/react-native-cameraroll#readme
https://github.com/react-navigation/react-navigation
https://reactnavigation.org/docs/stack-navigator/
https://github.com/brix/crypto-js
https://github.com/expo/expo/tree/master/packages/expo
https://libraries.io/npm/expo-asset
https://libraries.io/npm/expo-cellular
https://github.com/expo/expo/blob/master/changelogVersions.json
https://github.com/expo/expo/issues/10345
https://docs.expo.io/versions/v39.0.0/sdk/linear-gradient/#usage
https://docs.expo.io/versions/latest/sdk/media-library/
https://docs.expo.io/versions/latest/sdk/accelerometer/#usage
https://github.com/expo/expo/issues
https://docs.expo.io/versions/latest/react-native/statusbar/
https://github.com/benjamn/install
https://docs.npmjs.com/
https://tr.reactjs.org/blog/2020/02/26/react-v16.13.0.html
https://reactjs.org/
https://github.com/expo/react-native/archive/sdk-39.0.4.tar.gz",
https://github.com/apollographql/react-apollo#readme
https://github.com/software-mansion/react-native-gesture-handler#readme
https://github.com/software-mansion/react-native-reanimated#readme
https://github.com/react-native-toolkit/react-native-responsive-dimensions#readme
https://github.com/th3rdwave/react-native-safe-area-context#readme
https://github.com/react-navigation/tabs#readme
https://github.com/necolas/react-native-web/issues/1749
https://github.com/react-navigation/react-navigation#readme
https://github.com/react-navigation/stack#readme
https://slideplayer.biz.tr/slide/12212929/
http://web.firat.edu.tr/mbaykara/ytm4.pdf
http://web.firat.edu.tr/mbaykara/ytm4.pdf
https://www.savassakar.com/
https://www.ogznet.com/en-iyi-7-ucretsiz-online-akis-semasi
https://medium.com/kodcular/react-native-nedir-7b333d319597
https://lucid.app/documents#/dashboard
https://www.opusgo.com/Servisler/Yazilim-ve-Bakim-Onarim-Destegi
http://web.firat.edu.tr/mbaykara/ORNEKPROJEDOKUMANI.pdf