T.C.

FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSLİĞİ GÜNCEL KONULAR

2020-2021

Proje Ekip Bilgileri

14542522- Anıl Özçelik **15542515- Şamil Öztoprak**

16542504 - Yakup Samet Koç **16541513-Özge Yeter Şahin**

16541524-Ayşegül Tepecik **16541547 - Beyza Çiftçioğlu**

16541564 -Ayşe Eraslan **175541601-Kutaiba Alaaeddin**

175541007 - Burak Keten **175541024 - Emre Eşkili**

175541035 - Hasan Çahan **175541049-Hakan Yıldız**

İÇİNDEKİLER

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
        4.1.5 Testler
        4.1.6 Performans
    4.2 Veri Tasarımı
        4.2.1 Veri Tanımları
    4.3 Süreç Tasarımı
        4.3.1 Genel Tasarım
        4.3.2 Modüller
    4.4 Ortak Alt Sistemlerin Tasarımı

5. SİSTEM GERÇEKLEŞTİRİMİ

    5.1 Giriş
    5.2 Açıklama Satırları
    5.3 Kod Biçimlemesi
    5.4 Anlamlı İsimlendirme
    5.5 Yapısal Programlama Yapıları
    5.6 Olağandışı Durum Tanımları
    5.7 Kod Gözden Geçirme
    5.8 Gözden Geçirme Sürecenin Düzenlenmesi
    5.9 Gözden Geçirme Sırasında Kullanılacak Sorular
    5.10 Öbek Arayüzü
    5.11 Giriş Açıklamaları
    5.12 Veri Kullanımı
    5.13 Öbeğin Düzenlenişi
    5.14 Sunuş

6. DOĞRULAMA VE GEÇERLEME

    6.1 Giriş

1. GİRİŞ

1.1 Projenin Amacı

Mobil mobil platformlarda bilgi güvenliği için etkili bir güvenlik algoritması oluşturup etkili bir görüntü şifreleme algoritması tasarlarken, mobil cihazınızın RAM verilerini SHA3 işlevi ile şifreleyerek görüntüleri şifreleyip, matematiksel işlevlerle rastgele sayı dizileri oluşturur ve bunları bir web hizmeti olarak sunar.

1.2 Projenin Kapsamı

Sistem ve sistem faydaları çevrimiçi iletişim sırasında paylaşılan verilerin güvenli bir şekilde iletilmesini sağlamak isteyen kullanıcılar için tasarlanır.

2. PROJE PLANI

2.1 Giriş

Uygulama projesinde, belirsiz bir internet ortamında kullanıcıların kendi aralarında iletişim kurması ile sağlanan veri koruma prensibine dayanmaktadır. Bilgiler, gerçek verileri bir saldırgandan korumak için şifrelenir.

2.2 Projenin Plan Kapsamı

Proje beş ana bileşenden oluşmaktadır. Her koşul farklı şekilde oluşturuldu ve bir araya getirildi. IP1 iş paketi, cep telefonu kurulumunu içerir. SHA3 hash algoritması kullanılır. IP2 paket işlevinde, xor işlemi SHA3 ile mevcut veriler ve bir yöntem kombinasyonu ile gerçekleştirilir. IP3 şifreleme sistemi ve ölçüm sistemi inşa edildi. API tasarımı, kullanılan IP4 iş paketi ve şifreleme algoritmaları ile uygulandı. IP5 iş paketinde görüntü şifrelenir, kullanıcıya sağlanır ve sinyal çözülür.

Projenin amacı, güvenli olmayan bir İnternet ortamında kullanıcılar arasında iletişim kurarken sağlanan bilgi koruma temeli üzerine inşa etmektir.

Maliyet Kestirim Planı:

Ölçüm Parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	4	6	24
Kullanıcı Çıktı Sayısı	2	7	14

Ana İşlev Nokta Sayısı			38

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?	**1**
5. Performans kritik mi?	**3**
6. Girdiler, çıktılar, dosyalar ya da sorgular karmaşık mı?	**1**
7. İçsel işlemler karmaşık mı?	**3**
8. Tasarlanacak kod yeniden kullanılabilir mi?	**5**
9. Dönüştürme ve kurulum tasarımda dikkate alınacak mı?	**4**
10. Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak?	**5**
Toplam	**37**

Cevaplar 0 ile 5 arasında puanlandırılır

Bunlar hesaplanıp toplanarak Teknik Karmaşıklık Faktörü (TKF) elde edilir.

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

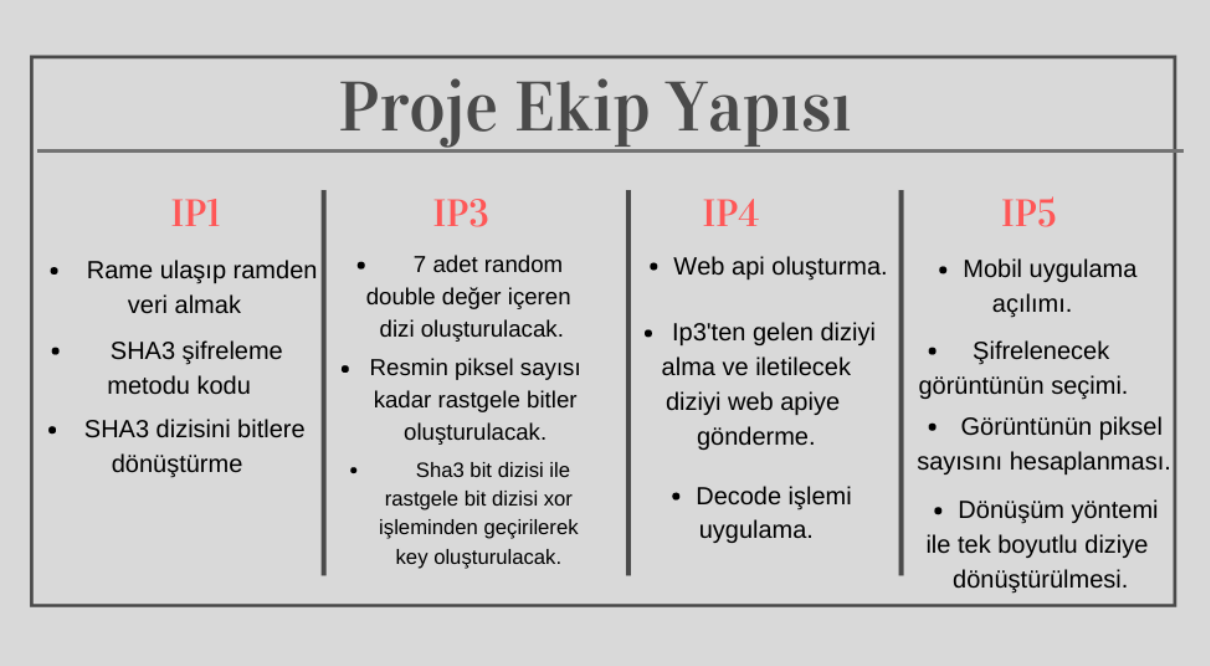
Proje İş-Zaman Planı

İş \ Zaman	1.hafta	2.hafta	3.hafta	4.hafta	5.hafta	6.hafta	7.hafta	8.hafta	9.hafta	10.hafta
Proje Teklifi	X									
Proje Planı		X	X							
Proje Analizi				X	X					
Sistem Çözümleme					X	X				
Kullanıcı Arayüz Tasarımı							X			
Gerçekleştirim							X	X		
Test									X	
Sunum										X

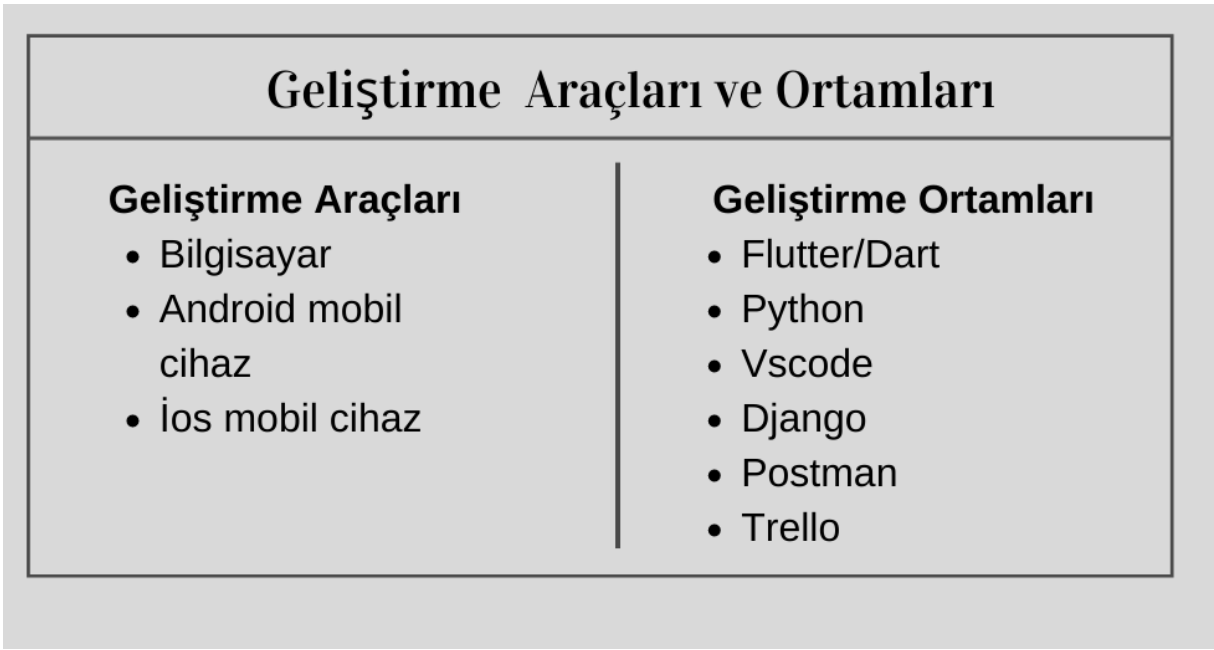
2.4 Proje Ekip Yapısı

Proje Ekip Yapısı

IP1	IP3	IP4	IP5
KUTAİBA ALAAEDDİN	EMRE EŞKİLİ	AYŞE ERASLAN	HAKAN YILDIZ
ANIL ÖZÇELİK	AYŞEGÜL TEPECİK	BEYZA ÇİFTÇİOĞLU	HASAN ÇAHAN
		ÖZGE YETER ŞAHİN	ŞAMİL ÖZTOPRAK
		YAKUP SAMET KOÇ	



2.5 Kullanılan Özel Geliştirme Araçları ve Ortamları



2.6 Proje Standartları Yöntem ve Metodojileri

Aşama	Kullanılan Yöntem/Araçlar	Ne İçin Kullanıldığı	Çıktı
Planlama	- Veri Akış Şemaları, - Süreç Belirtimleri, - Görüşme, - Maliyet Kestirim Yöntemleri - Proje Yönetim Araçları	- Süreç İnceleme - Kaynak Kestirimi - Proje Yönetimi	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 İilişki Şemalarının Veri Tablolarına Dönüştürülmesi	- Başlangıç Tasarım - Ayrıntılı Tasarım - Başlangıç Veri - Tasarımı	Başlangıç Tasarım Raporu
Tasarım	- Yapısal Şemalar - Program Tasarım Dili - Veritabanı Tabloları - Veri Sözlüğü	- Genel Tasarım - Ayrıntılı Tasarım - Veri Tasarımı	Sistem Tasarım Raporu

2.7 Kalite Sağlama Planı

Projedeki kalite sağlama planımız aşağıda yer almaktadır.

1.Ekonomi: Ekonomik olarak programın maliyeti ucuz ve zaman tasarrufu nedeniyle oldukça makul.

2.Tamlık: Projede açıklık olmamalı ve tüm yazılım programı çalışıyor ve tamamlanmış olmalıdır**.**

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

4.Etkinlik: Kullanıcı, sistemin her alanı hakkında tam bilgiye sahip olduğu için sistemi verimli bir şekilde kullanabilir.

5.Bütünlük: Proje bir bütün halinde çalışmaktadır.

6.Güvenilirlik: Yüksek güvenlik önlemleri içerir.

7.Modülerlik: Sistem tek bir parçadan oluşmaktadır.

8.Belgeleme: Bu belgeden de görebileceğiniz gibi, bu belge sistemin tam bir özeti olacak şekilde tasarlanmıştır.

9.Kullanılabilirlik: Karmaşık sistemler, her kullanım düzeyinden insana hitap ettiği için zor renklerden kaçınabilir.

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

11.Değiştirilebilirlik: Kullanıcılar sistemde değişiklik yapabilir.

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

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

14.Sınanabilirlik: Proje, sınanabilir bir yapıya sahiptir.

15.Taşınabilirlik: Sistem herhangi bir mobil cihazda kullanılabildiği için herhangi bir cihazda taşınabilir ve kullanılabilir.

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

2.8 Konfigürasyon Yönetim Planı

Gelecekte sistemin yeni kullanıcı gereksinimlerini karşılayamaması veya sistem yapısının bileşenlerinin güncelliğini yitirmesi durumunda olası bir yapılandırma planı geliştirilmiştir.

2.9 Kaynak Yönetim Planı

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

Kaynak Edinimi:

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

Proje Ekibinin Geliştirilmesi:

- Proje'nin analiz edilip planı yapılması için düzenli olarak her hafta online toplantılar organize edilerek eksik becerilerin geliştirilme süreci gerçekleşmiştir.
- Proje aşamaları kontrol edilerek eksik kalınan noktalarda takım arkadaşlarından destek alıp proje geliştirilme süreci kontrol altına alınmıştır.

- Proje test ve sunum hazırlık aşamasında 2. bir ekip yapısı devreye girerek takım üyesinin kendini geliştirmiş olduğu alanlar incelenerek görev dağılımı yapılmıştır.
- Proje geliştirme sürecini takip etmek üzere trello aracı kullanılarak insan, donanımsal ve yazılımsal kaynaklar kontrol altına alınmıştır.

a. Eğitim Planı

Araçların, programların kullanımında kullanılacak programlama dilleri öğretilmezse bu proje başarıyla tamamlanmayacaktır. Bu nedenle projeye başlamadan önce biraz eğitim gereklidir.

- Fatih Özkaynak - Kriptoloji bilimine giriş
- Python programlama dili
- Django
- Flutter/dart
- Trello
- Canva
- Microsoft Visio
- GitHub kullanım
- Postman Api Development
- An Image Encryption Algortihm Based on Chaotic Selection of Robust Cryptographic Primitives
  Destek alınan araçları kullanmak ve proje geliştirmek için eksik konular giderilmek amaçlı eğitimleri alınmıştır.

b. Test Planı

Testler, uygulamanın sonucuna göre uygulamayı iyileştirmemizde katkı sağlayarak verimli çalışmasına adapte olmuştur. Herhangi bir güncelleme dahilinde uygulamanın sürdürebilirliği kontrol altına alınmıştır.

c. Bakım Planı

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.

3. SİSTEM ÇÖZÜMLEME

3.1 Mevcut Sistem Çözümlenmesi

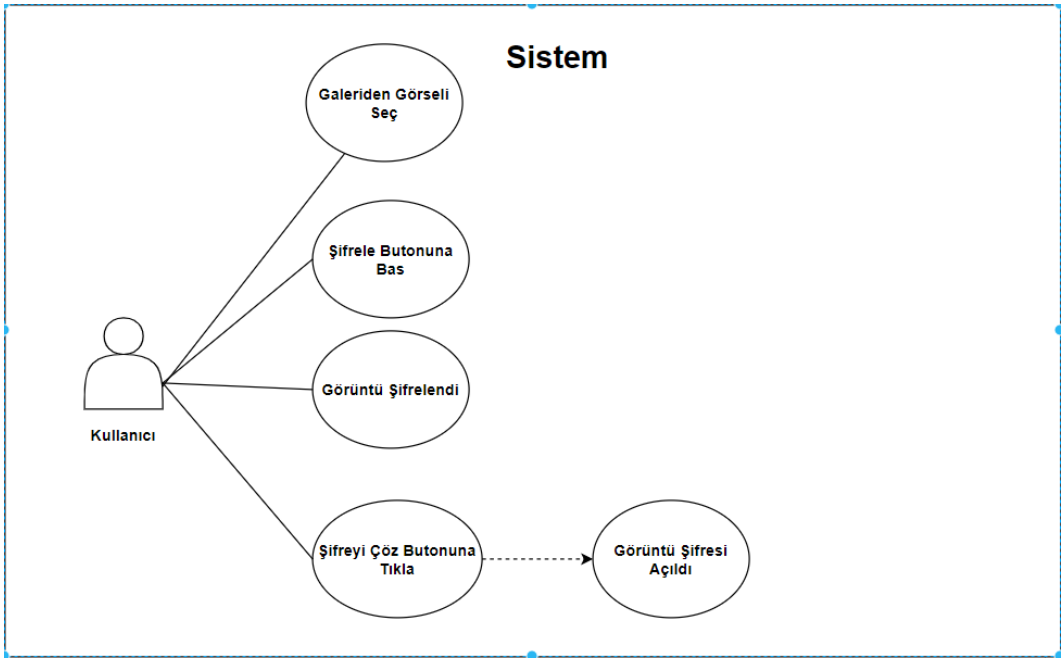
Mevcut sistemin incelenmesi aşamasında gerçekleştirdiğimiz projenin içyapısı kullanıcı ara yüzleriyle ekte eklenmiştir.

3.1.1 Örgüt Yapısı

Örgüt yapısı olarak mobil alanda şifreleme işlem prosedürlerini ortak oluşan bir örgüt yapısı vardır.

Sistemin Örgüt Yapısının Örneklenmesi

3.1.2 İşlevsel Model



Use-Case Diyagramı

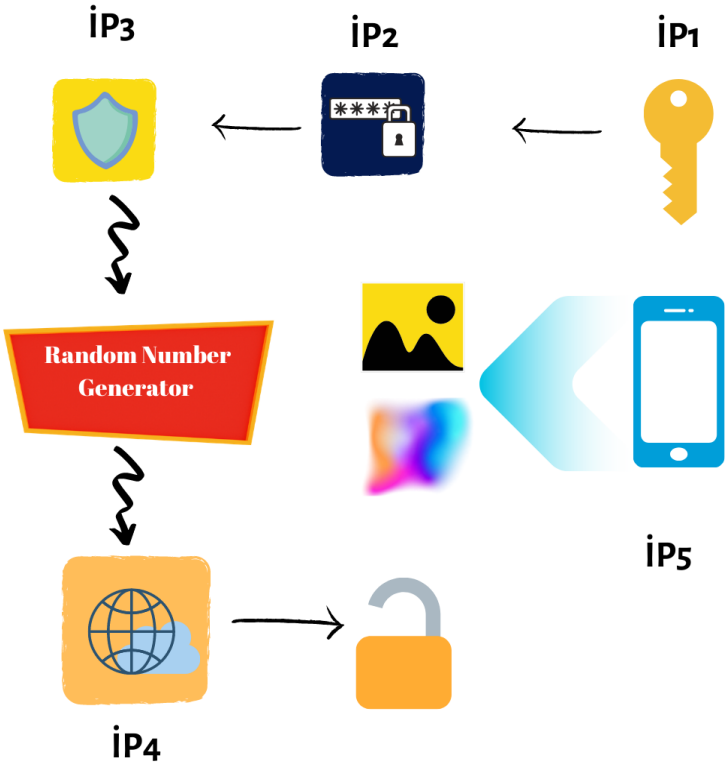
3.1.3 Kullanılan Yazılım/Donanım Kaynakları

DONANIMSAL KAYNAKLAR

- BİLGİSAYAR
- ANDROID MOBİL CİHAZ
- IOS MOBİL CİHAZ

YAZILIMSAL KAYNAKLAR

- FRONT-END: FLUTTER
- BACK-END: DART
- PYTHON
- VSCODE
- DJANGO
- POSTMAN

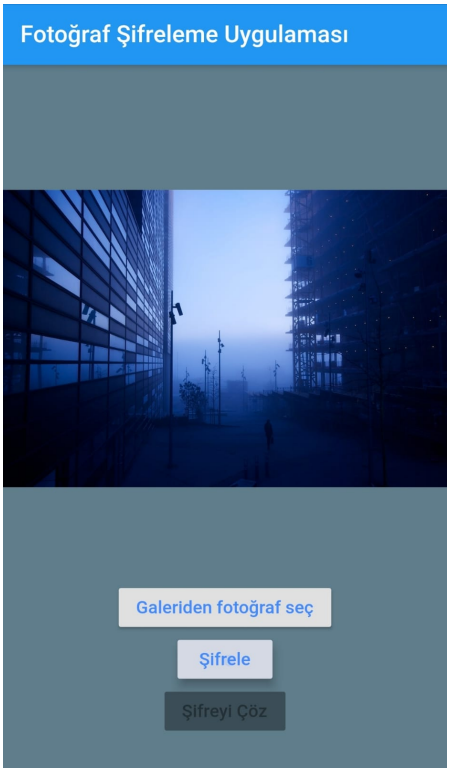


Paketin Genel İşleyişi

3.1.4 Varolan Sistemin Değerlendirilmesi

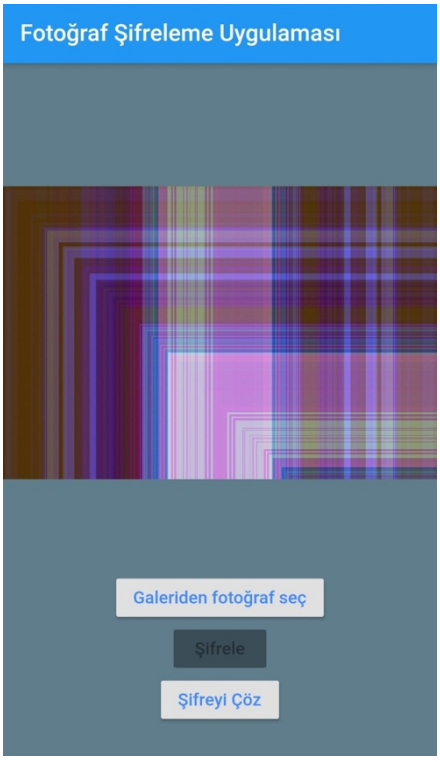
Uygulamamızda seçilen görsel şifrelenebilir, şifrelenmiş görsel tekrardan eski haline getirilebilir bir hal almıştır

Açılış Ekranı



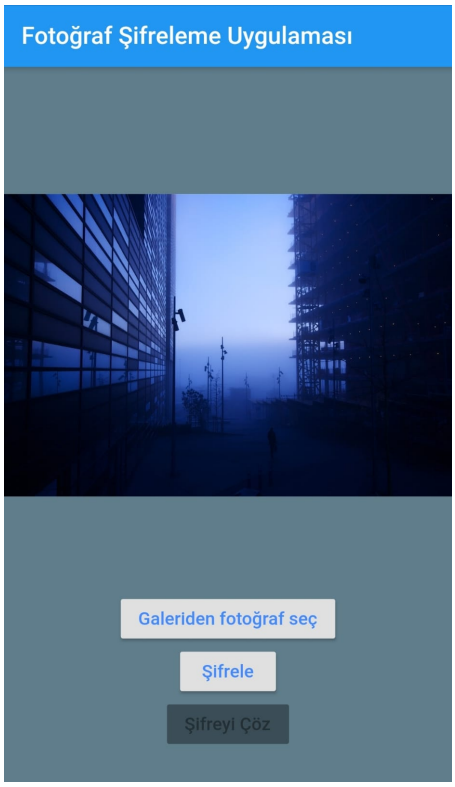
Galeriden Seçilmiş Bir Görsel

Galeriden seçtiğimiz Görseli Şifrele Butonuna basarak Şifreli bir hale getiriyoruz.



Şifrelenmiş Görsel

Şifrelenmiş görseli tekrardan eski haline getirmek için Şifreyi Çöz butonuna basmamız gerekmektedir.



Şifrelenmiş Görselin Eski Haline Getirilmesi

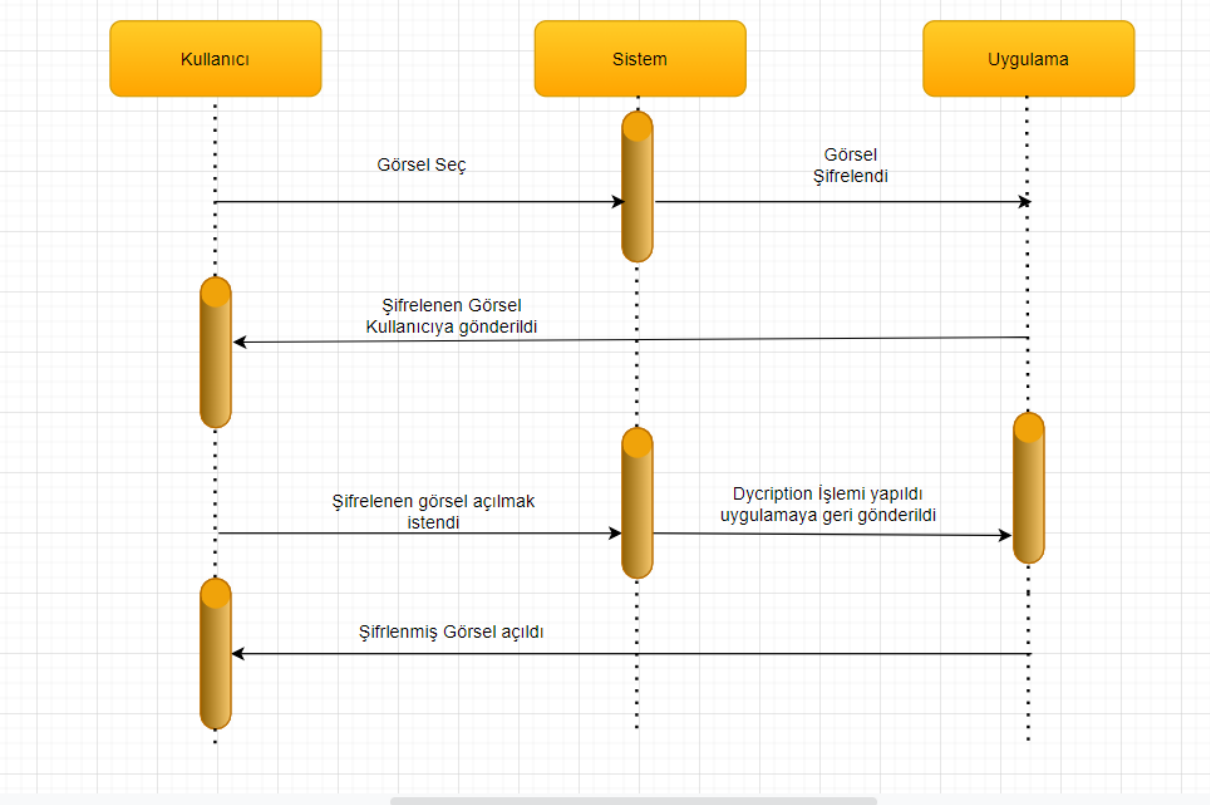
3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

Mevcut sistemler incelendiğinde sonuca giden yolda epeyce eksikler söz konusudur. Yapılan uygulamada diğer sistemlerden farkı Bit sayısı kadar SHA3 kodu oluşturuyor.

3.2.2 Genel Bakış

Genel hatlarıyla sistemi inceleyecek olursak sistemde bir şifreleme söz konusudur. Bir SHA3 şifreleme metodu uygulanarak SHA3 dizisi bitlere dönüştürüldü. 7 adet random double değer içeren dizi oluşturuldu. Resmin piksel sayısı kadar rastgele bitler oluşturularak SHA3 dizisi ile rastgele bit dizisi XOR işleminden geçirilerek bir key oluşturuldu. Oluşturulan Web Api ip'3 den gelen diziyi alıp dcryption işlemi yaptıktan sonra iletimi sağlandı. İP5'de tasarlanan mobil uygulama, şifrelenecek görüntüyü seçimini, görüntünün şifrelenip şifre çözme dönüşümü sağlandı dönüşüm tek boyutlu dizi olarak elde edildi.



İşleyiş Diyagramı

3.2.3 Başarım Gerekleri

Mevcut sistemler incelendiğinde mevcut sistemin eksiklerinden yola çıkarak, sistemin başarımı için

- Sistemin sonuç üretiminde doğrulukları
- Tepki sürelerini en aza indirilmesi
- Hile hata ve yanlışlıkların en aza indirilmesi
- Kullanım kolaylığı
- Anlaşılabilirlik
- Tarafsızlık

Temel gereklilik olarak tespit edilmiştir.

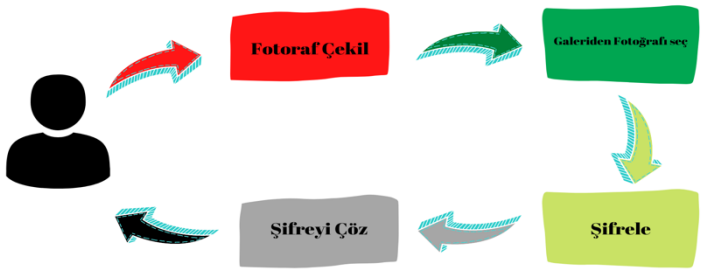
3.3 Kullanıcı Arayüzü

3.3.1 Yazılım Arayüzü

Projenin çalışma esnasında açık verilmemesine önem verildi. Gerekli her türlü değişiklik kodlar üzerinde yapılıp tekrar denendi.

3.3.2 Kullanıcı Arayüzü

Projede kullanıcının arayüzü tasarlanırken herhangi bir şekilde renkler seçilerek tarafsız rahat büyük puntolu yazılı bir arayüz tasarlanacaktır. Çok basit ve kullanışlı bir arayüz tasarlanması amaçlanmaktadır.



Kullanıcı İş Akışı

Kullanıcı Arayüzü

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

Geliştirme sürecinde genel olarak belgelendirilmesi hem ileriye dönük hem de şimdiki geliştirme sürecinde projenin tamamlanma yüzdesini nerede kalınıp nerelerde eksikler olduğunu genel hatlarıyla göstermesi amacıyla yapıldı. Bunun yanı sıra projeye dahil olan ekip arkadaşlarımızın olayın hakimiyeti kavraması açısından bu yönteme başvuruldu.

3.4.2 Eğitim Belgeleri

Mevcut belgemiz bulunmamaktadır.

4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri

4.1.1 Genel Sistem Tanımı

Sistemde bulunan resmin şifrelenip, çözümlenmesi için oluşturulan bu projenin tüm gereksinim ve modellemelerine karar verip projeyi hayata geçirme sürecine adım atılmıştır.

Projeden beklenen gereksinimleri kısaca; sistemde aktif belirleme özelliği bulunan resimlerin şifrelenip, daha sonrasında şifrenin çözümlenmesi için oluşturulan bir kripto alanıdır.

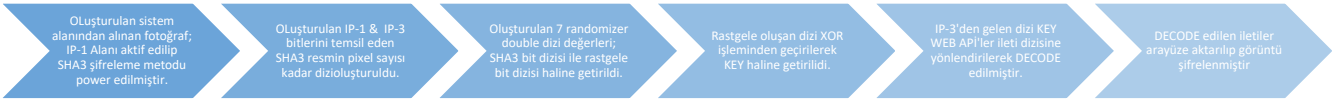
4.1.2 Varsayımlar ve Kısıtlamalar

Projede bulunan varsayımlar;

- Sistemde aktif halde değişen resimlerin IP-1 iş paketine aktarılması.
- IP-1 iş paketinin Flutter'dan RAM'e ulaşabilmesi için SHA3 şifreleme metoduna tanımlanması
- SHA3'ü yansıtan bit dizileri, resmin pixel sayısı kadar oluşturulması.
- 7 Adet randomizer double değerli dizi oluşturulması.
- SHA3'de yeni oluşturulan randomizer dizi değerleri XOR işleminden geçirilerek KEY haline getirilir.

- WEB APİ de oluşturulan KEY verileri iletilecek diziye gönderilerek DECODE işlemi gerçekleştirilir. Bu sayede sistemden seçilen resim şifrelenmiş halde sisteme tekrardan aktarılır.
- Şifrelen resim aynı işlemin DECODE edilmesi ile çözümlenir.

4.1.3 Sistem Mimarisi



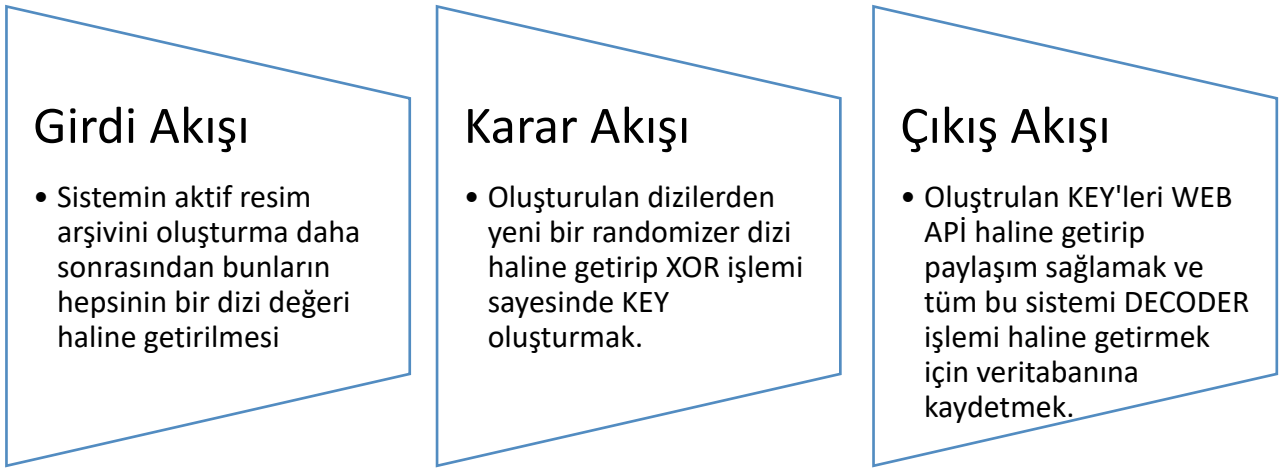
4.1.4 Dış Arabirimler

4.1.4.1 Kullanıcı Arabirimleri

Projede kullanıcı arabirimi olarak;

- Resimlerden oluşan sistemden resim belirleme
- Belirlenen resmin şifreleme butonu ile şifrelenmesi
- Şifrelenen resmin çözümlenmesi butonu ile çözümlenmesi

4.1.4.2 Veri Arabirimleri



4.1.4.3 Diğer Sistemlerle Arabirimler

Kullanacağımız sistemde ilk veri olarak aktardığımız resim değerlerinin bitlerden oluşan dizi hali bütünüyle arabirim verileridir.

4.1.5 Testler

Geliştirilen yazılım projesinde oluşabilecek hatalar çok olası bir durumdur. Bundan dolayı yazılım uygulaması gelişen prototipleri adım adım her 15 günde bir test edilmiştir.

- Sistemin Tasarım Uygulanabilirlik Testi;
  Çözümlemesi yapılan sistem ve mimari bileşenlerinin tasarıma dönüştürülebilirliği ve tasarıma uygunluğu test edilecek.
- Tasarlanan Sistemin Mantıksal İşlev Testi;
  Sistemin çözümlemesinin tasarıma dönüştürülmesiyle mantıksal doğruluğu test edilecektir.
- Kullanıcı Ara Yüzlerinin Kullanılabilirlik Testi;

Sistemin temel işlev fonksiyonlarının ve veri tabanı sistem bilgilerinin kullanıcı ara yüzlerindeki gösterim doğrulukları test edilecektir.

- Tasarım Yapılan Veri tabanının Mantıksal İşlev Testi;

  Çözümlemesi yapılan veri tabanının tasarıma dönüştürülmesiyle mantıksal doğruluğu test edilecektir.

4.1.6 Performans

Geliştirilen yazılım uygulamasının performans değerlendirme ölçütü aşağıdaki kriterlere göre yapılmaktadır.

- Kapsam (%20)
- Zaman (%20)
- Maliyet (%15)
- Kalite (%15)
- Proje Yönetim Metodolojisine Uyum (%15)
- Paydaş / Yönetim Memnuniyeti (%15)

4.2 Veri Tasarımı

Sistemin veri tasarımı yapılırken dikkat edilen noktalar;

- Veri tipleri belirlenip düzenlenmiştir.
- Gerekli tablo sayısı düzenlenmiştir.
- Veri kirliliği olmaması için gereksiz değişken kullanımından kaçınılmıştır.
- Karmaşık görüntü algılama algoritmasına uygun sorgular kullanılmıştır.

4.2.3 Veri Tanımları

Bilgi birikimini düzenleyip katman katman düzenlenmesini sağlayan bir veri saklama yapısıdır. Kullanıcıların rahat veri bulabilmesini ve veri hangi bilgi bölümü ile alakalı ise oraya dağılmadan kolayca katman yapısı ile bulmasını sağlar. Bu sistem sadece kullanıcı kolaylığını sağlamayıp aynı zamanda sistem üreticilerinin de kolay bilgi erişimini ve kolay alt yapı oluşturmalarını sağlar.

Değişken Adı	**Veri Tipi**	**Uzunluk**	**Açıklama**
Ip-1	int	256	**RAM'den çekilen resmin pixellerinin dizi değeri hali**
Ip-3	double	256	**SHA-3 algoritmasının kullanımı için oluşturulan 7 kez randomizer işlem görmüş, Ip-1 in dizi boyutunda double dizi yapısı**
Key	double	256	Ip-1 ve Ip-3 den gelen dizi verilerinin XOR yöntemi ile oluşturduğu yeni yapıdır(şifredir)

4.3 Süreç Tasarımı

4.3.1 Genel tasarım

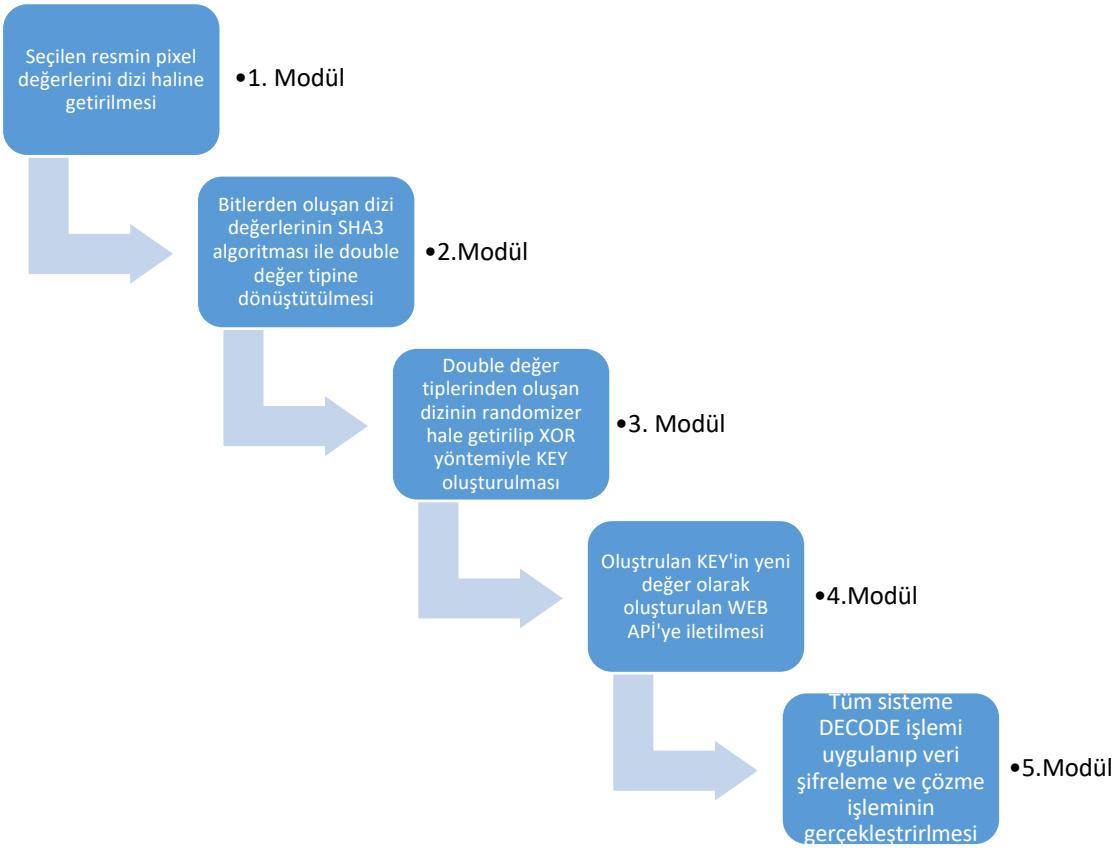
Süreçlerin iç işleyişini daha ayrıntılı ve açık bir şekilde değerlendirebilmek için iyi tasarım kıstaslarının kullanılması gereklidir.

Biz Projemizde;

- IEEE 1016.1-1993,
- IEEE Guide to Software Design Descriptions IEEE 1016-1998,
- IEEE Recommended Practice for Software Design Standartları kullanılmıştır.

4.3.2 Modüller

Modüller, isimleri olan tanımlanmış işlevleri bulunan ve hedef sistemi gerçekleştirmek üzere türleştirilen birimlerdir. Tasarımı daha başarılı olarak yapabilmenin bir yolu da problemi uygun parçalara bölerek yazılımı modüler bir şekilde geliştirmektir. Bu şekilde her bir modül için ayrı ayrı dikkat ve emek sarf edilerek toplamda daha hassas ve doğru çözüm elde edilir. Dolayısıyla aynı büyüklükteki bir problemi ne kadar fazla sayıda modüle ayırırsak toplam karmaşıklık o kadar azalacaktır.



4.3.2.1 Seçilen Resmin Pixel Değerlerini Dizi Haline Getirilmesi

4.2.2.1.1 İşlev

Sistemin ilk verisi olarak da adlandırılabilir. Kaba tabirle şifrelemek istenilen resmin seçilip pixel değerlerinin bitlerden oluşan dizi haline getirilir.

4.3.2.1.2 Kullanıcı Arabirimi

Sistemde kullanıcı arabiriminde görünmemektedir. Bit değerlerinden oluşan bir dizi oluşturulur.

4.3.2.1.3 Modül Tanımı

Sistemden seçilen resmin pixel değerinin bitlerden oluşan diziye dönüştürülüp bir sonraki modülün kullanımına hazır hale getirmek.

4.3.2.1.4 Modül İç Tasarımı

4.3.2.2 SHA3 algoritması ile Dizi Değerlerinin Dönüştürülmesi

4.3.2.2.1 İşlev

Modül1' de oluşturulan dizinin SHA3 algoritması sayesinde bit değer tipinden double değer tipine dönüştürülmesidir.

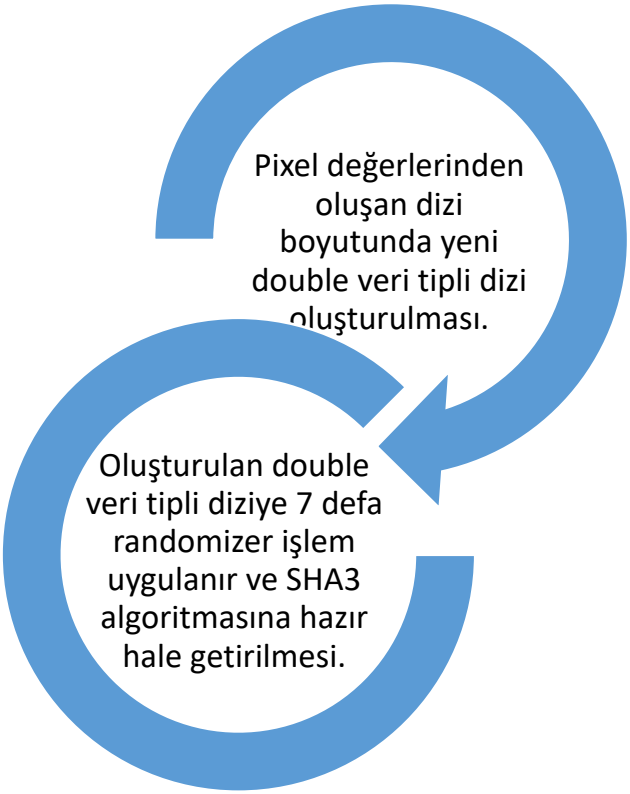
4.3.2.2.2 Kullanıcı Arabirimi

Kullanıcı sistemin bu modülüne de erişememektedir.

4.3.2.2.3 Modül Tanımı

7 defa randomizer işlem uygulanmış; resmin dizi değer boyutu kadar double veriler oluşturulmuş.

4.3.2.2.4 Modül İç Tasarımı



4.3.2.3 KEY Oluşturma Modülü

4.3.2.3.1 İşlev

Şifreleme için XOR yöntemi ile KEY(Anahtar) oluşturulması.

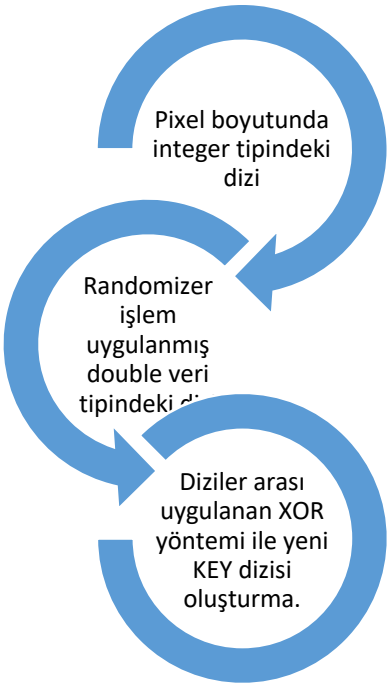
4.3.2.3.2 Kullanıcı Arabirimi

Kullanıcı sistemin bu modülüne de erişememektedir.

4.3.2.3.3 Modül Tanımı

Resimden oluşturulan dizi değeri ile yeni oluşturulan randomizer işlem uygulanan double diziye XOR işlemi uygulanır ve oluşan yeni dizi değeri KEY olarak adlandırılır.

4.3.2.3.4 Modül İç Tasarımı



4.3.2.4 WEB APİ ile İletişim

4.3.2.4.1 İşlev

PrimaryKey olarak gerçekleştirdiğimiz Web Api ye KEY dizisinin iletimini yapmak.

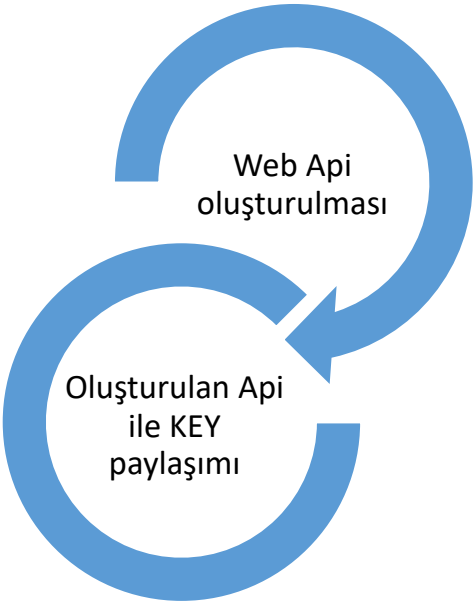
4.3.2.4.2 Kullanıcı Arabirimi

Kullanıcı sistemin bu modülüne de erişememektedir.

4.3.2.4.3 Modül Tanımı

Sistem tarafından daha öncesinde oluşturulmuş ya da yeni oluşturulan ileti merkezi olan Web Api ye şifre çözümleme yapabilmesi için KEY paylaşımı yapılmasıdır.

4.3.2.4.4 Modül İç Tasarımı



4.3.2.5 Decode Modülü

4.3.2.5.1 İşlev

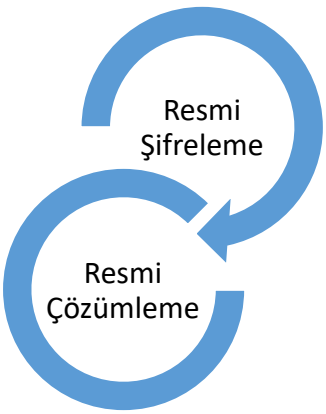
Tüm işlemlerin Api ile paylaşımından sonra sistem döngüsünün oluşabilmesi için Return edebilen yapı modülü

4.3.2.5.2 Kullanıcı Arabirimi

4.3.2.5.3 Modül Tanımı

Sistemden alınan resmin şifrelenip tekrardan çözümlenmesi için DECODE mantığını barındıran modül yapısıdır.

4.3.2.5.4 Modül İç Tasarımı



4.3.4 Entegrasyon ve Test Gereksinimleri

Veri tabanı sistemde ön planda olduğundan dikkat edilmesi gereken ve entegrasyonu dikkatle yapılması gereken bir aşamadır. Veri tabanımız sisteme değişiklik yapılmadan entegre edilecektir. Sisteme veri tabanı eklendikten sonra şu test aşamaları gerçekleştirilecektir.

Test sırasında dikkat edilmesi gereken konulardan bazıları şöyledir;

- Veri tabanı yük altında sistemle uyumlu ve doğru çalışıyor mu?
- Sistemle veri tabanı ile uyumlu mu?
- Veri tabanı ve sistem arasında ki tepki verme süresi hızlı mı?
- Veri tabanının sistemdeki bilgileri okuma doğruluğu yeterli mi?
- Veri tabanı sistem kayıt doğruluğu yeterli mi?

4.4 Ortak Alt Sistemlerin Tasarımı

4.4.1 Ortak Alt Sistemler

Projenin alt sistemleri şu şekildedir;

- Yetkilendirme Alt Sistemi
- Güvenlik alt Sistemi
- Yedekleme Alt Sistemi
- Veri Transferi Alt Sistemi
- Arşiv Alt Sistemi
- Dönüştürme Alt Sistemidir.

4.4.4 Güvenlik Alt sistemi

Güvenlik alt sistemi, bilgi sisteminde yapılan işlerin ve yapan kullanıcıların izlerinin saklanması ve gereken durumlarda sunulması ile ilgilidir. Birçok yazılım geliştirme ortamı ve işletim sistemi, bu amaca yönelik olarak, "**sistem günlüğü**" olanakları sağlamaktadır. Sistem

günlüğü ile sunulanın olanaklar yeterli olmadığı durumlarda ek yazılımlar geliştirilmesi gerekmektedir.

Şifreleme sisteminden oluşan projenin her türlü güvenlik alt sistemi mevcuttur. Ram' den erişilen resim pixelleri dışında sitemin geneli primary key, foreinch key olarak tanımlanmıştır. Ayrıca güvenliğin sağlanması için değişken web api key sistemi kullanılmaktadır.

4.4.5 Veri Dağıtım Alt Sistemi

Veri dağıtımı alt sistemi, verilerin değişik hizmet birimleri arasında güvenli bir biçimde iletilmesi işlemlerini içermektedir. Uygulamada, temel olarak iki veri iletişim türü bulunmaktadır:

- Çevrim-içi Veri İletişimi
- Çevrim-dışı Veri İletişimi

Şifreleme sistemi projemiz de genel işleyişi kurumsal internet ağı üzerinden gerçekleştirilecek biçimde tasarlanmış bu nedenle web api oluşumu dışında sistemde iletim sağlanacak bir ağ alanına taviz verilmemiştir. Çevrim-dışı yapısı da daha öncesinde iletişim gerçekleşen Api ile Key paylaşımı şeklinde olup gizlilik ve güvenliği korunmaktadır.

4.4.6 Yedekleme ve Arşivleme İşlemleri

Sosyal medya istihbarat adlı projesinin olağan dışı durumlara hazırlıklı olmak amacıyla, kullandığı veri tabanını yedekleme ve yedekten geri alma işlevlerine gereksinim duyacağından veriler üzerinde yedekleme yapılmaktadır.

Elde edilen verilerden sık kullanılmayan verileri tespit edilen bilgilerin ayrılması ve gerektiğinde bu bilgilere erişimi arşiv alt sistemi sağlamaktadır.

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1 Giriş

Projemizin Genel Yapısını İnceleyecek Olursak Baştan Sona;

İP1>

- RAM'e Ulaşıp Ramden Veri Almak
- SHA3 ile Şifreleme Metodu
- SHA3 Dizisini Bitlere Dönüştürme

İP3>

- Random Double Değer İçeren Dizi Oluşturma
- Resmin Piksel Sayısı Kadar Rastgele Bit Oluşturma
- SHA3 Bit Dizisi ile Rastgele Bit Dizisi XOR İşleminden Geçirilerek Key Oluşturma

İP4>

- Web Api Oluşturma
- İP3 üzerinden Gelen Diziyi Alma ve İletilecek Diziyi Web Api'ye Göndermek
- DECODE İşlemi Uygulamak

İP5>

- Mobil Uygulama Açılımı
- Şifrelenecek Görüntü Seçimi
- Görüntünün Piksel Sayısının Hesaplanması
- Dönüşüm Yöntemi ile Tek Boyutlu Diziye Dönüştürülmesi

Yazılım Geliştirme Ortamları:

Yazılım geliştirme ortamı, tasarımı sonunda üretilen projenin, bilgisayar ortamında çalıştırabilmesi için gerekli olan:

- Programlama Dili
- SHA3 Şifreleme
- Hazır Program Kitapçıkları

Programlama Dilleri:

Sistemde kullanılan başlıca programlama dilimiz Python olarak belirlenmiştir. SHA3 Şifreleme metodu ile gelen resim piksel halinde şifrelenecektir.

5.3 Açıklama Satırları:

Açıklama satırları karmaşık her satırın sonunda yapıldı. Genel yazılan kod öncesi ve gerekli kod dizelerinde açıklamalar yapıldı.

5.4 Kod Biçimlemesi:

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

5.5 Anlamlı İsimlendirme:

Sistem kodlamasının genel yapısında kullanılan değişkenlere anlamlı isimlendirmelerde bulunduk.

5.5 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.

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.

5.6 Olağandışı Durum Tanımları:

Olağandışı gelişen durumlarda try-catch blokları devreye girecek ve program kırılmadan çalışmasına devam edecek şekilde tasarladık.

5.7 Kod Gözden Geçirme:

Projede aşama tüm kod satırları incelenmeden, göz geçirilmeden işleme alınmadı. Kaynak kod üzerinde gerekli incelemeler yapıldı. Bu sayede daha etkin, az hatalı ve okunabilir bir program ortaya çıktı.

5.8 Gözden Geçirme Sürecinin Düzenlenmesi:

- Hatalar bulundu ve düzeltildi.
- Gerekli incelemeler, çalışmalar ekip halinde yürütüldü ve bakımı gerçekleştirildi.
- Kalite çalışmaları yapıldı. Şifreleme yöntemi uygulandı.

5.9 Gözden Geçirme Sırasında Kullanılacak Sorular:

Programı incelerken, programın her bir öbeği (yordam ya da işlevi) belirlenen sorulara göre yanıtlarını bulduk.

5.10 Öbek Ara yüzü:

Oluşturduğumuz öbekleri test etmek için belli sorular sorduk bu sorular:

• Her öbek tek bir işlevsel amacı yerine getiriyor mu?

• Öbek adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?

• Öbek tek giriş ve tek çıkışlı mı?

• Öbek eğer bir işlev ise, parametrelerinin değerini değiştiriyor mu?

5.11 Giriş Açıklamaları:

Oluşturduğumuz giriş açıklamalarını test etmek için belli sorular sorduk bu sorular:

• 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?

5.12 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?

5.13 Öbeğin Düzenlenişi:

• Modüller birleşimi uyumlumu?

• Modüller arası veri aktarımları sağlanıyor mu?

• Bütün modüller birleştiğinde sistem çalışıyor mu?

5.14 Sunuş:

Artık son kısma gelindiğinde ise şu sorular soruldu:

• Her satır, en fazla bir deyim içeriyor mu?

• Bir deyimin 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?

• Bütün deyimler, belirlenen program stiline uygun olarak yazılmış mı?

• Öbek yapısı içerisinde akıllı "programlama hileleri" kullanılmış mı?

6. Doğrulama ve Geçerleme

6.1 Giriş

Gerçekleştirilen sistemin yazılımının doğrulanması ve geçerlemesi, üretim süreci boyunca gerçekleştirilen bir dizi aşama ve aşamalarda uygulanan birtakım yöntemlerden oluşur.

Söz konusu aşamalar:

- Yazılım belirtimlerinin ve proje yaşam sürecindeki her bir aşama sonunda alınan çıktıların, tamamlanmış, doğru, anlaşılabilir, kullanımı kolay, güvenliği sağlanmış, ve önceki belirtimleri tutarlı olarak betimler durumda olduğunun doğrulanması.
- Proje süresince her bir aşama ürününün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitenin tekrarına sebep olması, teknik yeterliliği sınanırken sınanan ürünün diğer ürünlere olan etkisi
- Projenin bir aşaması süresince geliştirilen anahtar belirtimlerin önceki belirtimlerle karşılaştırılması ve ilişkisi.

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

Şekil 6.1 Doğrulama Geçerleme

Giriş

Sınama planlaması sistemin kendisinden beklenenleri yerine getirip getirmediğini göstermek üzere hazırlanmıştır. İşlenen planlama adımları projede test işlemlerini kapsamaktadır.

Birim Testi: Fonksiyonlar veya kod modülleri (fonksiyonlar, veri yapıları, nesneler vb.) test edilir. Bu test, test uzmanlarınca değil geliştiriciler tarafından yapılır ve program kodunun ayrıntıları ile içsel tasarım biçiminin bilinmesi gerekir.

Entegrasyon Testi: Sistemin bağlı çalıştığı alt sistemler ile entegrasyonu testi yapılır.

Test edilecek ana fonksiyonlar:

Sistemin gereksinimlere göre kendisinden bekleneni yerine getirip getirmemesi test edilmiştir. Proje kapsamında birim ve yazılım testleri gerçekleştirilmiştir. Test edilen modüllerden biri sistemin arayüzü olarak kararlaştırılmıştır.

Hata Raporlama ve Verileri Kaydetme

Test edilen diğer bir modül ise sistemin alt yöneticilerinin giriş bölümüdür.

1. **Derece Hatalar:** Sistemin çalışmasını direkt etkileyen hatalardır.
2. **Derece Hatalar:** Sistemin çalışmasını direkt etkilemeyecek fakat işlevsel bazı özelliklerinin çalışmasını engelleyen hatalardır.
3. **Derece Hatalar:** Sistemin çalışmasını etkilemeyene görsel yönden etkileyebilecek olan hatalardır**.**

Test Sonuç Raporlama

Proje kapsamında yapılacak testlerin sonuçları raporlanarak saklanmıştır. Sisteminin ara yüz bölümüne yapılan testte kullanıcının sisteme erişimi test edilerek hatalar belirlenmeye çalışılmıştır. Sistem ara yüzü anlaşılır olduğundan bir sorun ile karşılaşılmamıştır.