

**FIRAT ÜNİVERSİTESİ**

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği Bölümü

**YMH459 – YMGK   
Proje Uygulaması ve Dokümantasyonu**

**KRİPTOMAT**

**Yazılım Geliştiricileri**

15541055 ATAOL BALKAR

16541016 MUHAMMED DİNÇER

16541402 HALİL İBRAHİM SÜRÜCÜ

16542014 MUHAMMED HAKAN ÇELİK

170541012 HÜSEYİN GÜNEŞ

170541042 FEYZA NUR GÖK

170541073 AYŞE NUR YAMAN

170541307 ŞÜKRÜ FURKAN SELÇUK

180541018 BETÜL YİĞİTTÜRK

180541078 ABDURRAHİM KURT

**Proje Yürütücüsü**

Doç. Dr. Fatih ÖZKAYNAK

**2020 – V9**

İçindekiler

[1. GİRİŞ 3](#_Toc62389643)

[1. PROJENİN AMACI 3](#_Toc62389644)

[2. PROJENİN KAPSAMI 3](#_Toc62389645)

[3. PROJENİN HEDEFLERİ 3](#_Toc62389646)

[4. PROJENİN BAŞARI KRİTERİ 3](#_Toc62389647)

[5. PROJEYE GENEL BAKIŞ 3](#_Toc62389648)

[6. FONKSİYONEL GEREKSİNİMLER 3](#_Toc62389649)

[7. FONKSİYONEL OLMAYAN GEREKSİNİMLER 4](#_Toc62389650)

[8. SİSTEM MODELLERİ 4](#_Toc62389651)

[2. SİSTEM TASARIMI 5](#_Toc62389652)

[1. Genel Tasarım Bilgileri 5](#_Toc62389653)

[1.1 Sistem Mimarisi 5](#_Toc62389654)

[3. Doğrulama ve Geçerleme: 8](#_Toc62389655)

[4. Test Planı 14](#_Toc62389656)

[5. Uygulama 15](#_Toc62389657)

[1. Mobil Uygulama 15](#_Toc62389658)

[2. Sunucu 16](#_Toc62389659)

[3. Örnek Uygulama 17](#_Toc62389660)

6. Grup içi puanlandırma………………………………………………………………………………………28

# GİRİŞ

## PROJENİN AMACI

Sistem RAM ’den aldığı benzersiz veriyle bir görüntüyü şifrelemeyi ve çözmeyi amaçlamaktadır.

## PROJENİN KAPSAMI

Projekapsamında şifrelenen görüntü benzersiz bir anahtarla şifrelenmiştir.

## PROJENİN HEDEFLERİ

* Projemizin öncelikli hedeflerinden birisi görüntüyü güvenlikli bir şekilde şifrelemek ve şifrelenmiş görüntüyü çözebilmektir.
* Şifrelenen ve çözülen görüntünün minimum sürede gerçekleştirilmesi gerekmektedir.
* Kullanıcı sistemi rahat kullanabilmelidir.

## PROJENİN BAŞARI KRİTERİ

* Projede kullanılan resim boyutunun şifreleme için gerekli olan süreyi geciktirmemelidir.

## PROJEYE GENEL BAKIŞ

* Güvenlikli bir şekilde önemli bilgilerini gizlemeyi amaç edinen kurumların tercih edeceği bir uygulamadır.

## FONKSİYONEL GEREKSİNİMLER

* Uygulamayı kullanan kişi Android işletim sistemine sahip bir telefon kullanmalıdır.
* Telefondan uygulama üzerinden resim seçilebiliyor veya resim çekilebiliyor olmalıdır.
* Uygulama üzerinden seçilen resim veya çekilen resim şifrelenmelidir.
* Şifrelenen resim minimum sürede şifrelenmelidir.

## FONKSİYONEL OLMAYAN GEREKSİNİMLER

* **KULLANILABİLİRLİK**

Uygulamada kullanışlı bir arayüz yapılmalıdır.

Butonlar uygun yerlere konulmalıdır.

Fontlar ve renk ayarları dikkatle yapılmalıdır.

* **GÜVENİLİRLİK**

Veri kaybı mümkünse sıfıra indirilmelidir.

Hata yakalama prosedürleri çalıştırılmalı ve yazılımın kesilmesi yerine uygun hata mesajları sunulmalıdır.

* **PERFORMANS**

Uygulama resim boyutu arttıkça şifrelemeyi uzun bekleme sürelerinde yapmamalıdır.

* **DESTEKLENEBİLİRLİK**

Uygulama Android işletim sistemi üzerinde çalışmaktadır.

* **ARAYÜZ**

Veri girişi resim ekle butonuyla olacaktır.

Çıktı olarak şifrelenmiş veri ekranda gözükecektir.

Şifreyi kaldır butonuyla şifre kaldırılacaktır.

## SİSTEM MODELLERİ

* **AKTÖRLER**

KULLANICI: Şifrelenecek resmi seçer. Resim şifrelenir. Şifrelenen resmin üzerinden şifreyi kaldırır.

* **OLAYLAR**

Resim seçilir.

Resim şifrelenir.

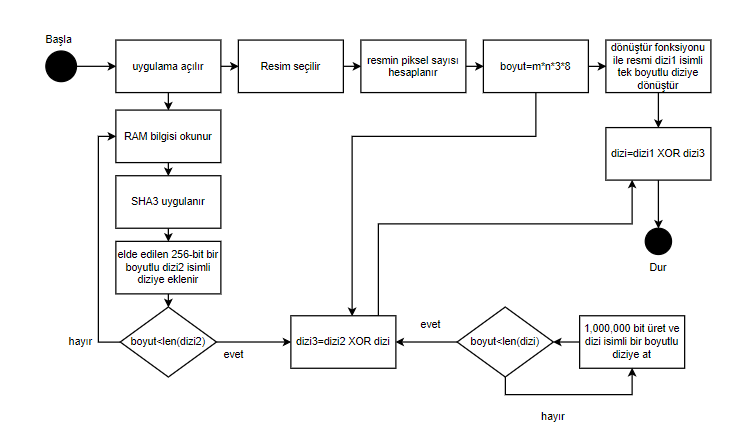
Şifrelenen resmin üzerinden şifre kaldırılır.

# SİSTEM TASARIMI

## 1. Genel Tasarım Bilgileri

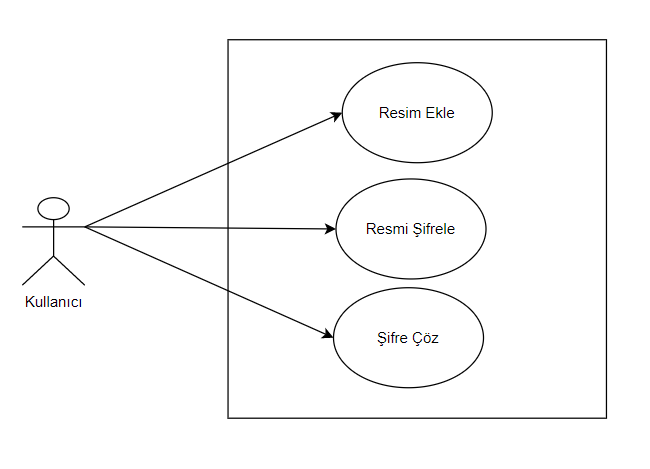
Bu projede kullanıcılar mobil uygulama üzerinden resim şifreleme ve şifre çözme işlemlerini yapılabilmesini sağlamıştır.

## 1.1 Sistem Mimarisi



Şekil Genel Sistem Mimarisi Akış Diyagramı

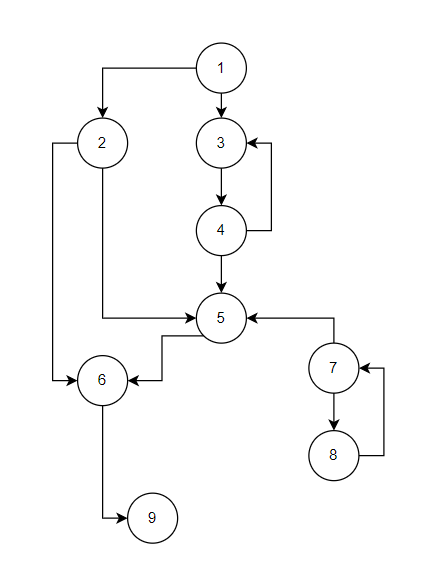
**2. Kullanıcı Use-Case Diyagramı**



Şekil Kullanıcı Use-Case Diyagramı

**3. Program Karmaşıklığı**

3.1 Programın Çizge Biçimine Dönüştürülmesi



3.2 McCabe Karmaşıklık Ölçütü Hesaplama

k = 12 Kenar sayısı

d = 9 Düğüm sayısı

p = 1 Bileşen sayısı

V(G)= k – d + 2p

V(G)=12 - 9 + 2 x 1 = 5

## Doğrulama ve Geçerleme:

**1. Giriş**

Doğrulama, yazılımın yaşam döngüsü boyunca her aşamada bir önceki aşamadaki gereksinimlere uygunluğunu denetleme işlemidir. Geçerleme ise geliştirme işleminin sonunda yazılımın gereksinimlere uygunluğunu, yani kendinden beklenenleri karşılayıp karşılamadığını test etme işlemidir. D&G, yazılımın istenen görevleri doğru şekilde yerine getirip getirmediğini belirlemek, istenmeyen herhangi bir işlem yapmadığından emin olmak ve kalite ve güvenilirliğini ölçmek amacıyla yazılımı kapsamlı bir şekilde test eder.

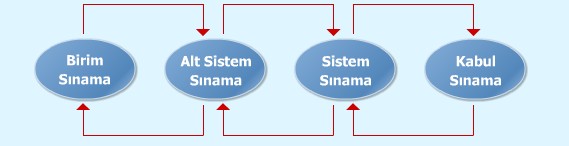


Bizlerde projemizi gerçekleştirirken bu soruları pek çok noktada kendimize sorarak ilerledik.

**2. Sınama Kavramları**

Sınama ve Bütünleştirme işlemlerinin bir strateji içinde gerçekleştirilmesi, planlanması ve tekniklerinin seçilmesi gerekmektedir.

Sınama işlemleri dört ana sınıfta incelenebilir:



**2.1 Birim Sınama**

Bağlı oldukları diğer sistem unsurlarından tümüyle soyutlanmış olarak birimlerin doğru çalışmalarının belirlenmesi amacıyla yapılır.

**2.2 Alt-Sistem Sınama**

Alt-sistemler modüllerin bütünleştirilmeleri ile ortaya çıkarlar. Yine bağımsız olarak sınamaları yapılmalıdır. Bu aşamada en çok hata arayüzlerde bulunmaktadır. Bu yüzden arayüz hatalarına doğru yoğunlaşılmalıdır.

**2.3 Sistem Sınaması**

Üst düzeyde, bileşenlerin sistem ile olan etkileşiminde çıkacak hatalar aranmaktadır. Ayrıca, belirtilen ihtiyaçların doğru yorumlandıkları da sınanmalıdır.

**2.4 Kabul Sınaması**

Çalıştırılmadan önce sistemin son sınamasıdır.

Artık, yapay veriler yerine gerçek veriler kullanılır.

Bu sınama türü alfa sınaması veya beta sınaması olarak ta bilinir.

**3. Doğrulama ve Geçerleme Yaşam Döngüsü**

Gerçekleştirim aşamasına kadar olan süreçlerde doğrulama ve geçerleme işlemlerinin planlaması yapılır.

Planlama genellikle; alt-sistem, bütünleştirme, sistem ve kabul sınamalarının tasarımlarını içerir. Gerçekleştirim aşamasının sonunda ise söz konusu plan uygulanır.

**4. Sınama Yöntemleri**

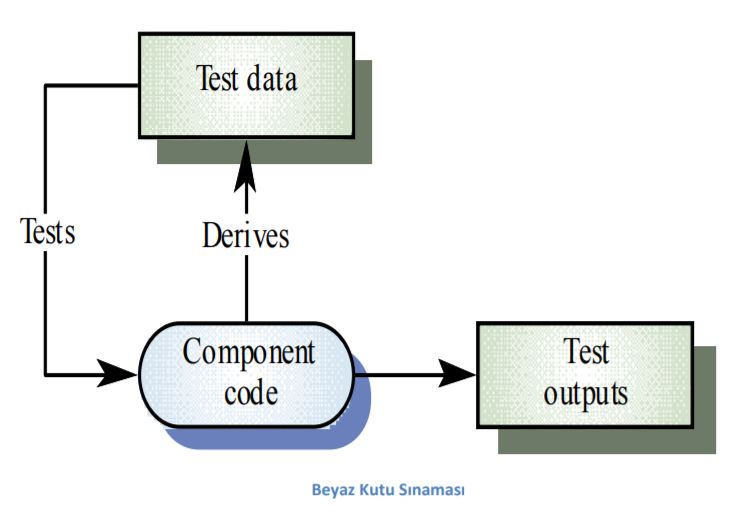
Her yazılım Mühendisliği ürünü iki yoldan sınanır:

**Kara Kutu Sınaması (Black-Box testing ):** Sistemin tümüne yönelik işlevlerin doğru yürütüldüğünün testidir. Sistem şartnamesinin gerekleri incelenir.

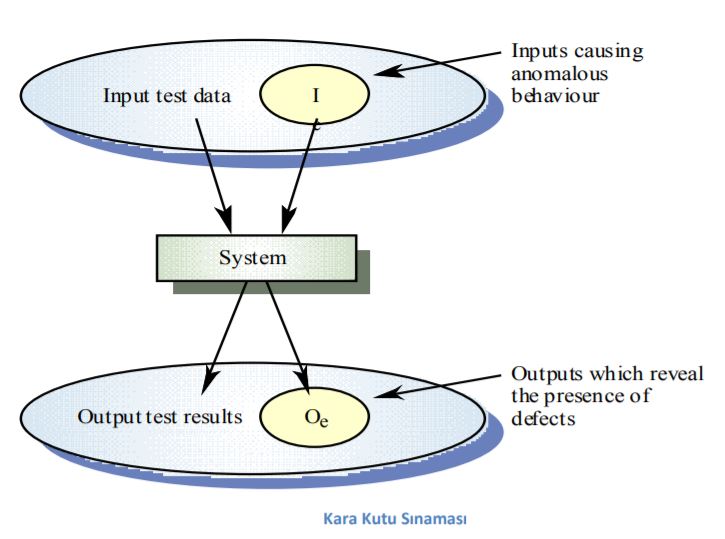
**Beyaz Kutu Sınaması (White Box testing ):** İç işlemlerin belirtimlere uygun olarak yürütüldüğünün bileşenler tabanında sınanmasıdır.

Projemizde her iki sınama yönteminide aktif olarak hem projenin oluşma aşamasında hemde test aşamasında kullandık. Çıkan sonuçlara göre iyileştirmeler ve düzeltmeler yapacağımız noktaları netleştirdik ve bu noktaları hedef alarak ilerlemeye devam ettik. Ayrıntılı olarak bu test aşamalarını inceleyelim;

**4.1 Beyaz Kutu Sınaması**

****

* Bütün bağımsız yolların en az bir kez sınanması gerekir.
* Bütün mantıksal karar noktalarında iki değişik karar için sınamalar yapılır.
* Bütün döngülerin sınır değerlerinde sınanması
* İç veri yapılarının denenmesi

**4.2 Kara Kutu Sınaması**

Ürünlerin test edilmesi sırasında kullanılan en ilkel test metodudur. Bir takım test senaryolarının seçilip, yazılım kodundan bağımsız olarak takip edilmesi temeline dayanmaktadır. Bu yüzden ürünlerin fonksiyonel durumları ve inputlara verdikleri tepkilerin gözlenmesi uygulamada kullanılan kara kutu testlerinin kapsamını oluşturmaktadır. Bu noktada, yazılımın kodunda yapılan herhangi bir değişiklik veya data yapısındaki uyarlamalar kara kutu testleriyle kontrol edilen özellikler değildir.

Test edilecek olan uygulamanın kodu hiç dikkate alınmadan, sadece girdilerin ve çıktıların incelenmesi ile gerçekleştirilen test metodudur. 5 ayrı tekniği bilinir. Denklik sınıfı test tekniği, test verileri gruplanır. Gruplar içinde testler yapılır.

**1**. Uç nokta test tekniği, hataların genelde sınırlarda çıktığı varsayılarak sınır değerlerinde test yapılır.

**2**. Karar tablosu test tekniği, çok fazla test yapılması gereken uygulamalarda verilerin matrix haline getirilerek test edilmesi test edilmesidir.

**3.** Sistem durumu test tekniği, farklı durum geçişleri yer alan sistemlerin testleridir.

**4**. İş senaryosu test tekniği, use case dokümanlarının kullanıldığı test tekniğidir.

**5. Sınama ve Bütünleme Stratejileri**

**5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme**

Yukarıdan-aşağıya bütünleştirmede önce sistemin üst düzeylerinin sınanması ve sonra aşağıya doğru olan düzeylere ilgili modülleri takılarak sınanması söz konusudur.

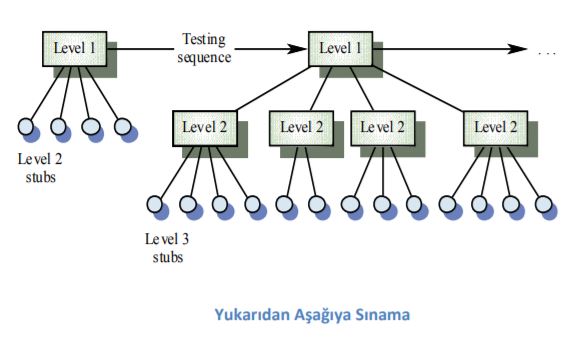
En üst noktadaki bileşen sınandıktan sonra alt düzeye geçilmelidir.

Alt bileşenler henüz hazırlanmamışlardır. Bu sebeple Koçanlar kullanılır. Koçan: Bir alt bileşenin, üst bileşen ile ara yüzünü temin eden, fakat işlevsel olarak hiçbir şey yapmayan çerçeve programlardır.

İki temel yaklaşım vardır:

• Düzey Öncelikli Bütünleştirme: En üst düzeyden başlanır ve aynı düzeydeki birimler bütünleştirilir.

• Derinlik Öncelikli Bütünleştirme: En üst düzeyden başlanır ve her dal soldan sağa olmak üzere ele alınır.

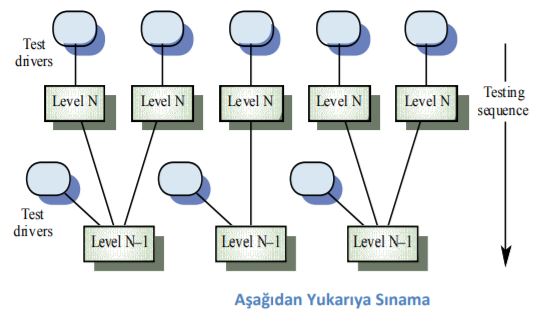


**5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

Önceki yöntemin tersine uygulama yapılır.

Önce en alt düzeydeki işçi birimler sınanır ve bir üst düzey ile sınanması gerektiğinde bu düzey bir sürücü ile temsil edilir.

Bu kez kodlama, bütünleştirme ve sınama, aşağı düzeylerden yukarı düzeylere doğru gelişir.

****

**6. Sınama Planlaması**

Her sınama planı, sınama etkinliklerinin sınırlarını, yaklaşımını, kaynaklarını ve zamanlamasını tanımlar. Plan neyin sınanacağını, neyin sınanmayacağını, sorumlu kişileri ve riskleri göstermektedir. Sınama planları, sınama belirtimlerini içerir.

**7. Sınama Belirtimleri**

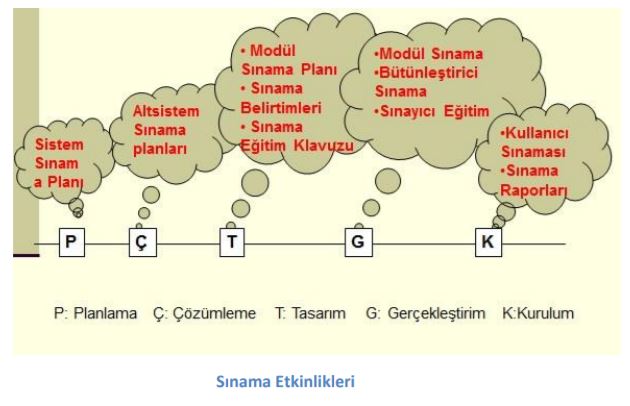
Sınama belirtimleri, bir sınama işleminin nasıl yapılacağına ilişkin ayrıntıları içerir.

Bu ayrıtılar temel olarak: sınanan program modülü ya da modüllerinin adları, sınama türü, stratejisi (beyaz kutu, temel yollar vb.), sınama verileri, sınama senaryoları türündeki bilgileri içerir.

Sınama verilerinin elle hazırlanması çoğu zaman kolay olmayabilir ve zaman alıcı 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 oluşturmasıdır. Sınama işlemi sonrasında bu belirtimlere, sınamayı yapan, sınama tarihi, bulunan hatalar ve açıklamaları türündeki bilgiler eklenerek sınama raporları oluşturulur.

**8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri**

****

# Test Planı

**Yazılım Testi**

Yazılım kodlama aşamasında programcı tarafından oluşabilecek hataları gidermek amacıyla dikkatli bir şekilde yapılır.

**Yeterlilik Testi**

Yazılımın istenilen şekilde yapılıp yapılmadığını kontrol etmek amacıyla yapılır. Yani yazılım isterleri tam olarak karşılıyor mu sorusuna cevap olarak yapılır.

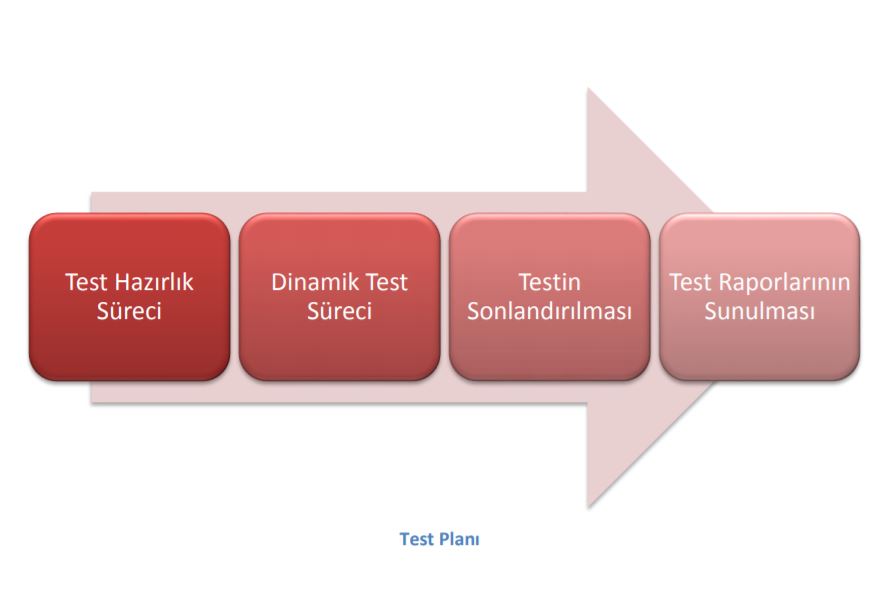
**Sistem Testi**

Yoğun veri akışı altında komple yükleme(load) testleri, normal olmayan koşullarda komple sistemin nasıl davranacağını görmek amacıyla germe(stres) testleri, istemli bir şekilde sistemi çökerterek sistemin nasıl davranacağını tespit etmek amacıyla geri kazanım(recovery) testleri, yazılımın geliştirilmesinde birimde yapay verilerle fabrika kabul testi, sistemin kullanılacağı yerde asıl verilerle kullanım hattı testleri, ve bundan sonra deneme testleri yapılır.

**Test Planı**

Çevik modelde analiz, kodlama, test ve kullanım birbirini izler. Yani her bir prototip kullanıcıya verilmeden her seferinde test edilir.

Bu testleri Yazılım Test Ekibi yapacaktır. Kodlamanın bitmesi beklenmeden test hazırlık sürecine geçilecektir. Projenin test süreci;



# Uygulama

Uygulama iki ayrı ana adımdan meydana gelmektedir. Bunlardan birisi mobil diğeri ise sunucudur.

# Mobil Uygulama

Mobil uygulamamız basit bir arayüze sahiptir. Arayüz farklı cihaz modellerine uyum sağlayacak biçimde tasarlanmıştır.



Şekil 1 Mobil Uygulama Arayüz

Burada en üstte yer almakta olan **Resim Ekle** butonuna basarak **Galeri**’den resim seçerek veya **Kamera** seçeneği ile bir resim seçilebilmektedir. Seçilen resim Resim Ekle butonu üstüne gelecektir. Daha sonrasında uygulamanın altında yer alan **Resmi Şifrele** butonu ile resmi sunucuya şifrelenmesi amacıyla gönderiyoruz.

# Sunucu

Mobilden gelen resim, projedeki **SaveAs** fonksiyonu kullanılarak sunucudaki Uploads klasörüne kaydedilmektedir. Klasöre kayıt edilen resim bitmap nesnesi kullanılarak piksellerine ayrılmaktadır.

Proje içerisinde bit bit işlem yaptığımız için önceden tanımladığımız boyut değişkenine **resim uzunluğu\*resim yüksekliği \* 3 \* 8** formülü kullanılarak bir değişken değeri elde edilmektedir. Bu boyut değeri bizim tek boyutlu dizimizin boyutunu belirlemektedir.

Bu işlemin devamında **Donustur** fonksiyonunu kullanarak resim piksellerinin bir değerlerini fraktal kullanarak diziye atama işlemi gerçekleşmektedir.

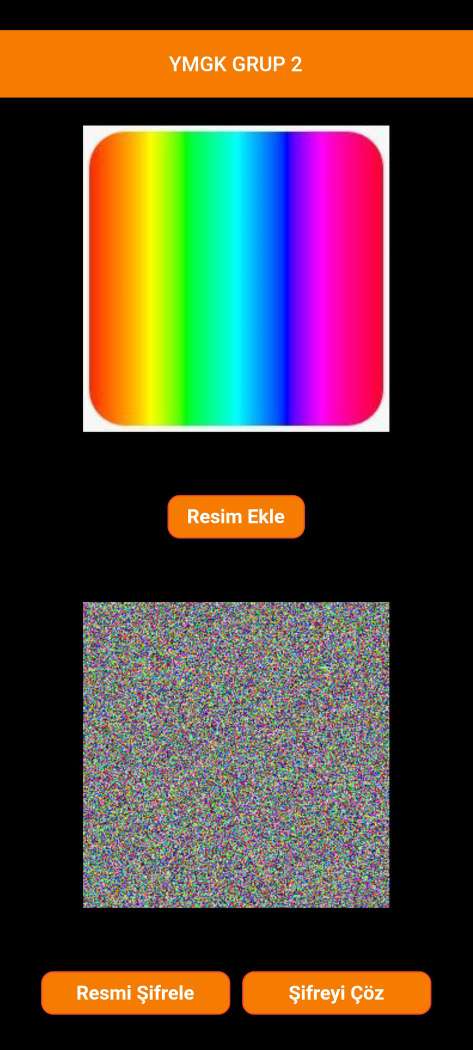
Diziye atama işlemi bittikten sonra **Pointer** kullanılarak Ram’den 64 bitlik veriler çekilmektedir ve bu veriler **SHA3** kullanılarak 256 bitlik değerler üretilmektedir. Projede kullanılan SHA3 fonksiyonu üretilen **ramveri** sınıfındaki **veri** fonksiyonunu kullanarak oluşturulan dizi değerinin boyutu, resim boyutuna eşit olana kadar ramden veri çekip 256 bitlik değerler üretmektedir.

Bu işlemin devamında önceden verilen başlangıç koşulları kullanılarak resim boyutuna eşit olana kadar 1 milyon bit üretildi. Gelen bit dizileri XOR fonksiyonu kullanılarak anahtar oluşturulmaktadır. Mevcutta bulunan resim pikselleri(dizisi) üretilen anahtar ile XOR’lanarak yeni bir bir dizisi oluşturulmaktadır.

Oluşturulan bu bit dizisi byte cinsinde piksel haline dönüştürülüp yeni bir dizi oluşturulmaktadır. Oluşturulan bu son şifreli resim projede uploads klasörüne kayıt edilir ve mobil uygulamaya şifreli resim cevap olarak döndürülür.

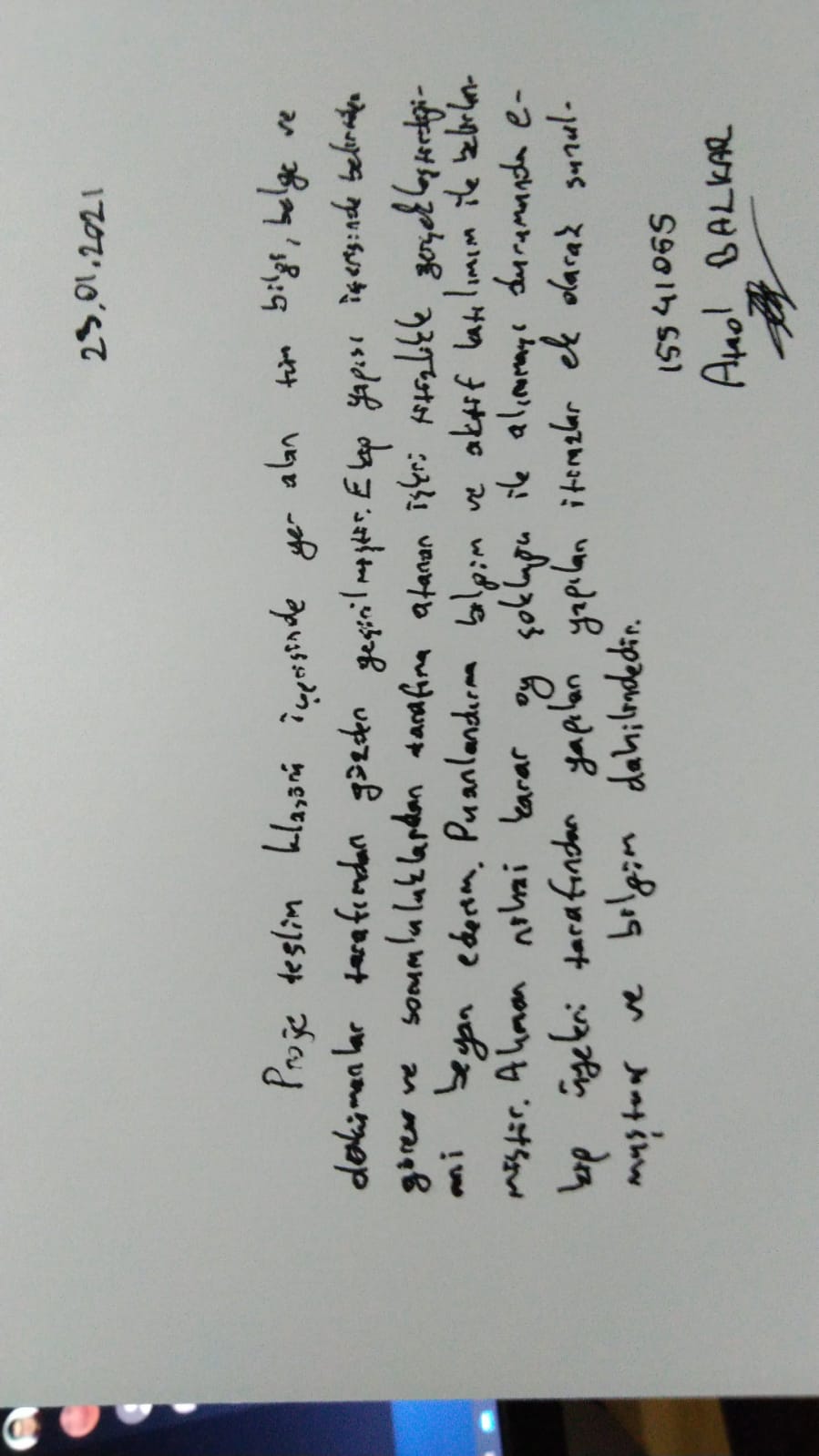
# Örnek Uygulama

**Resim Ekle** butonuna basılarak **Galeri’ye** gidildi ve bir resim seçildi. Sonrasında mobil uygulamamızda arayüze gelen bu resim için sol alt köşede yer alan **Resmi Şifrele** butonuna basılarak sunucu üzerine şifrelenmesi için gönderilmiş oldu. Şifrelenmiş resim Resim Ekle butonunun altına kısa bir süre içerisinde gelmektedir.

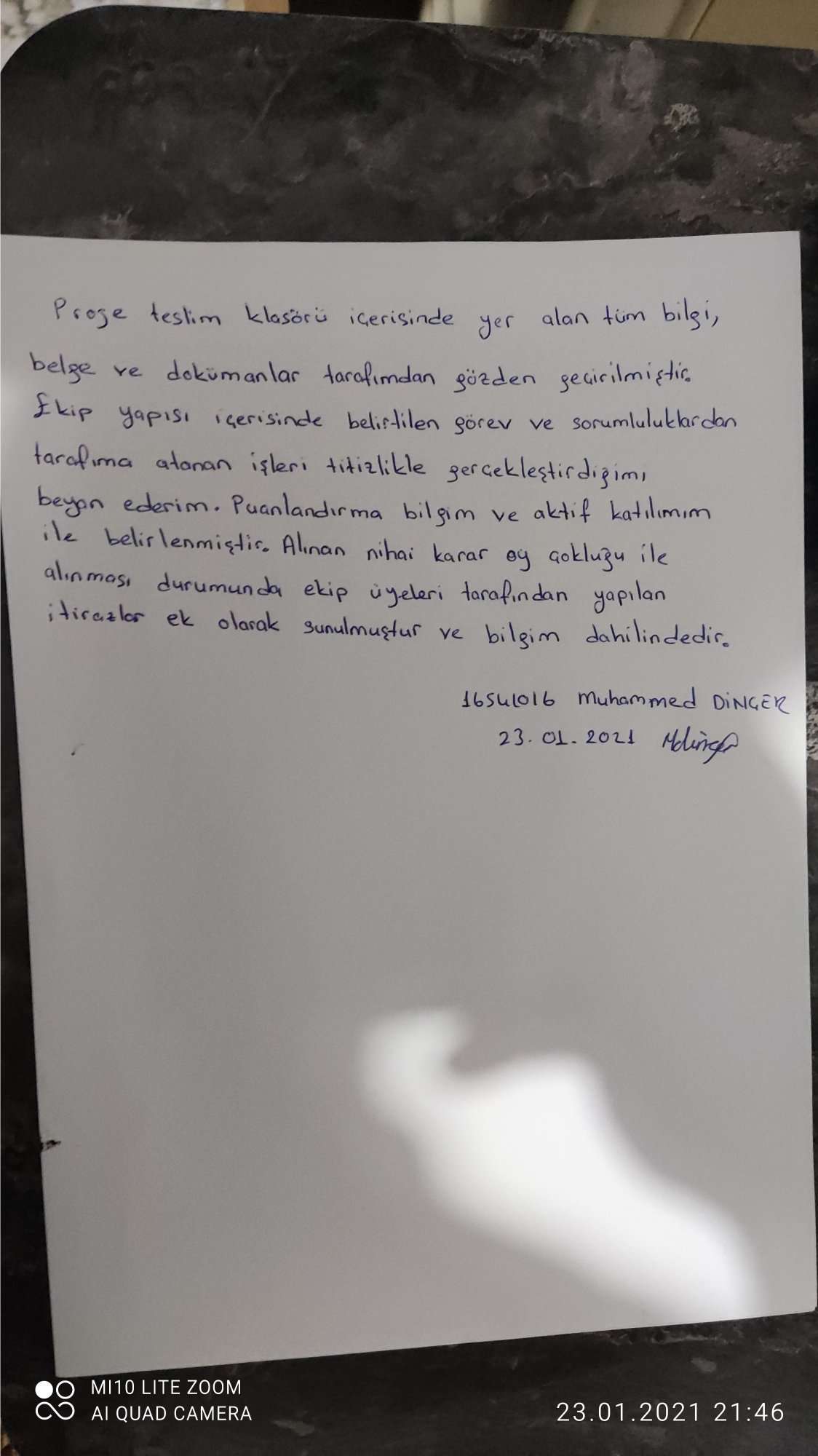


Şekil Örnek Uygulama

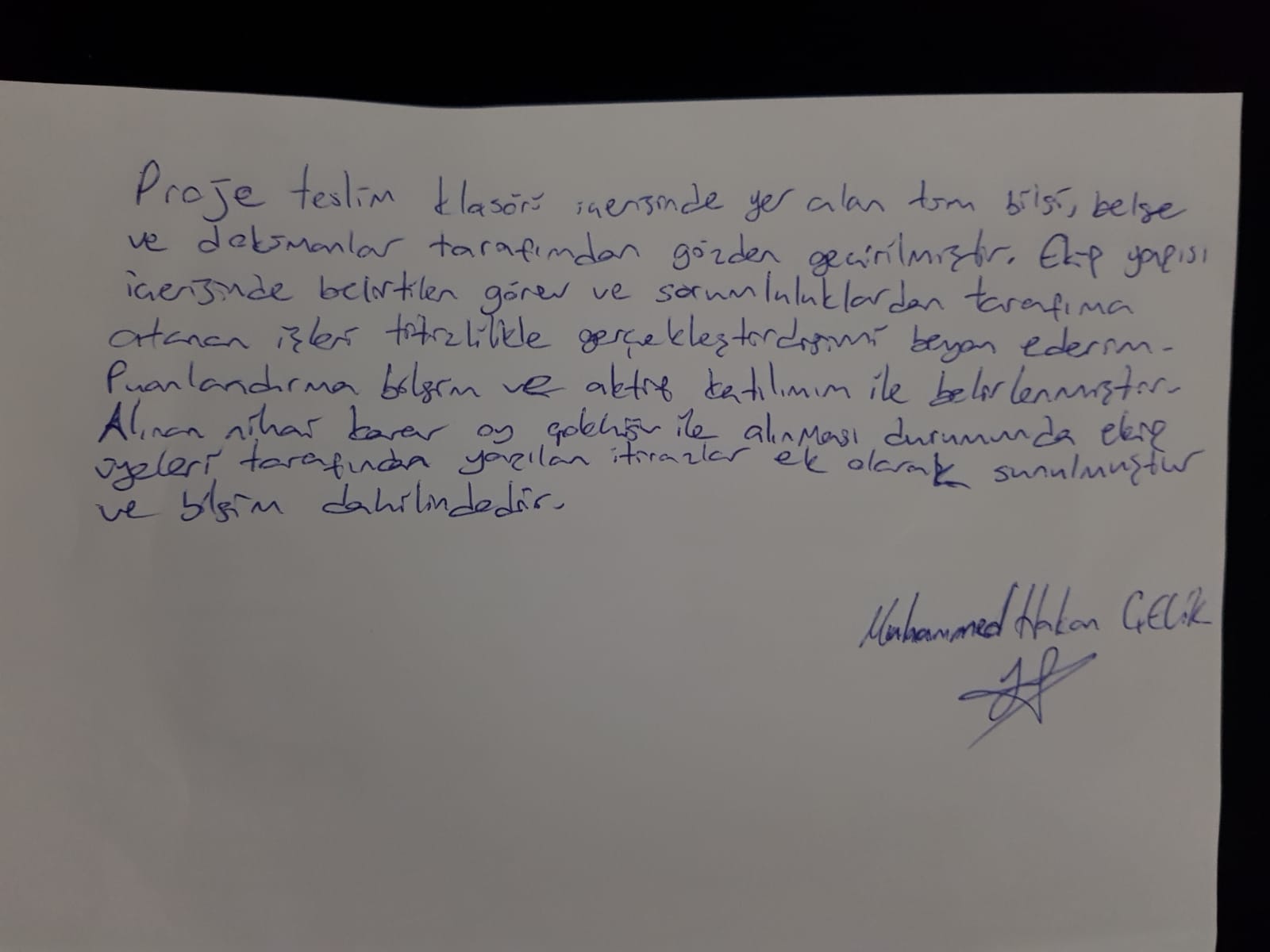
1. **Rıza Metinleri**
2. Ataol BALKAR



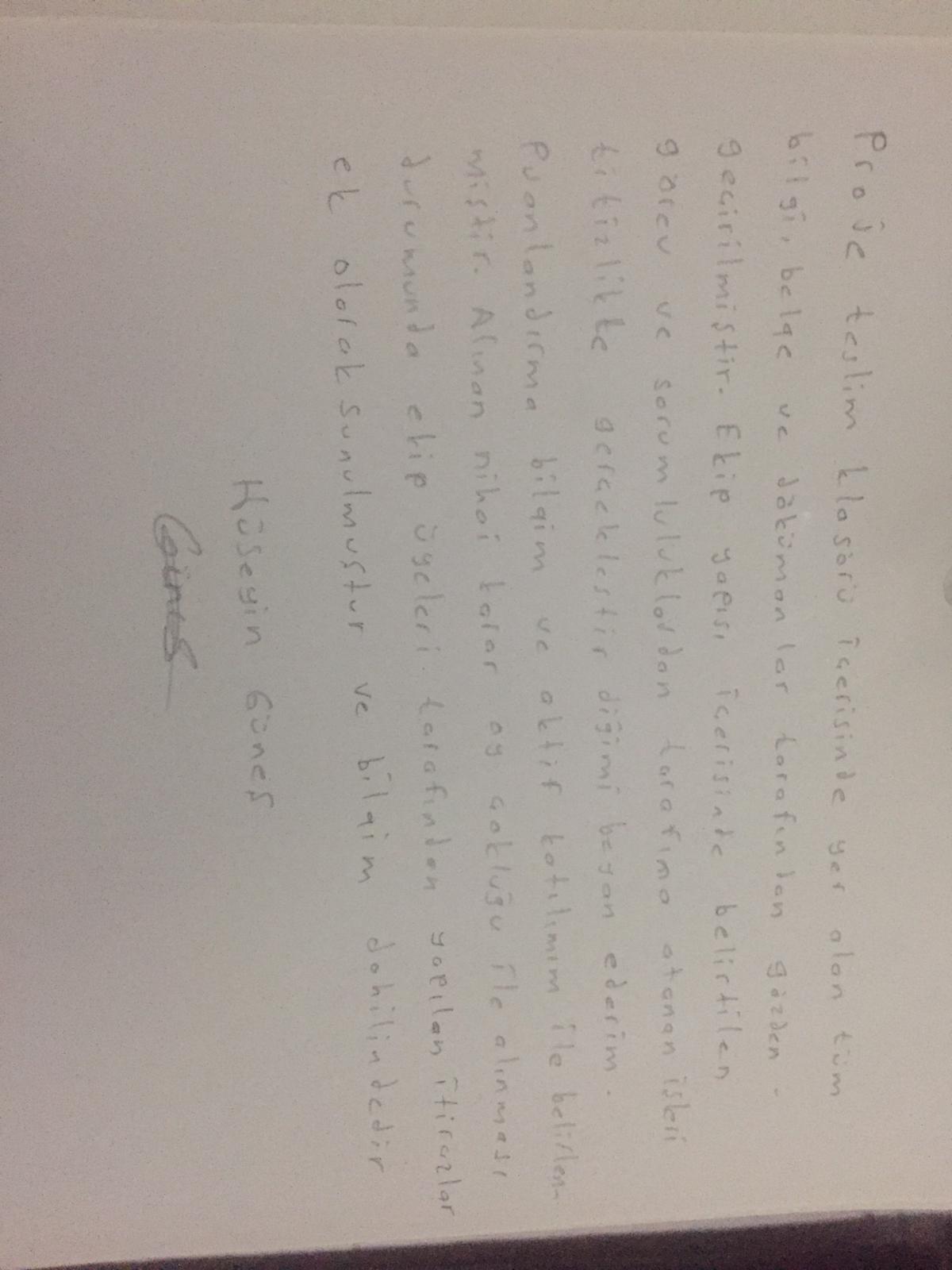
1. Muhammed DİNÇER



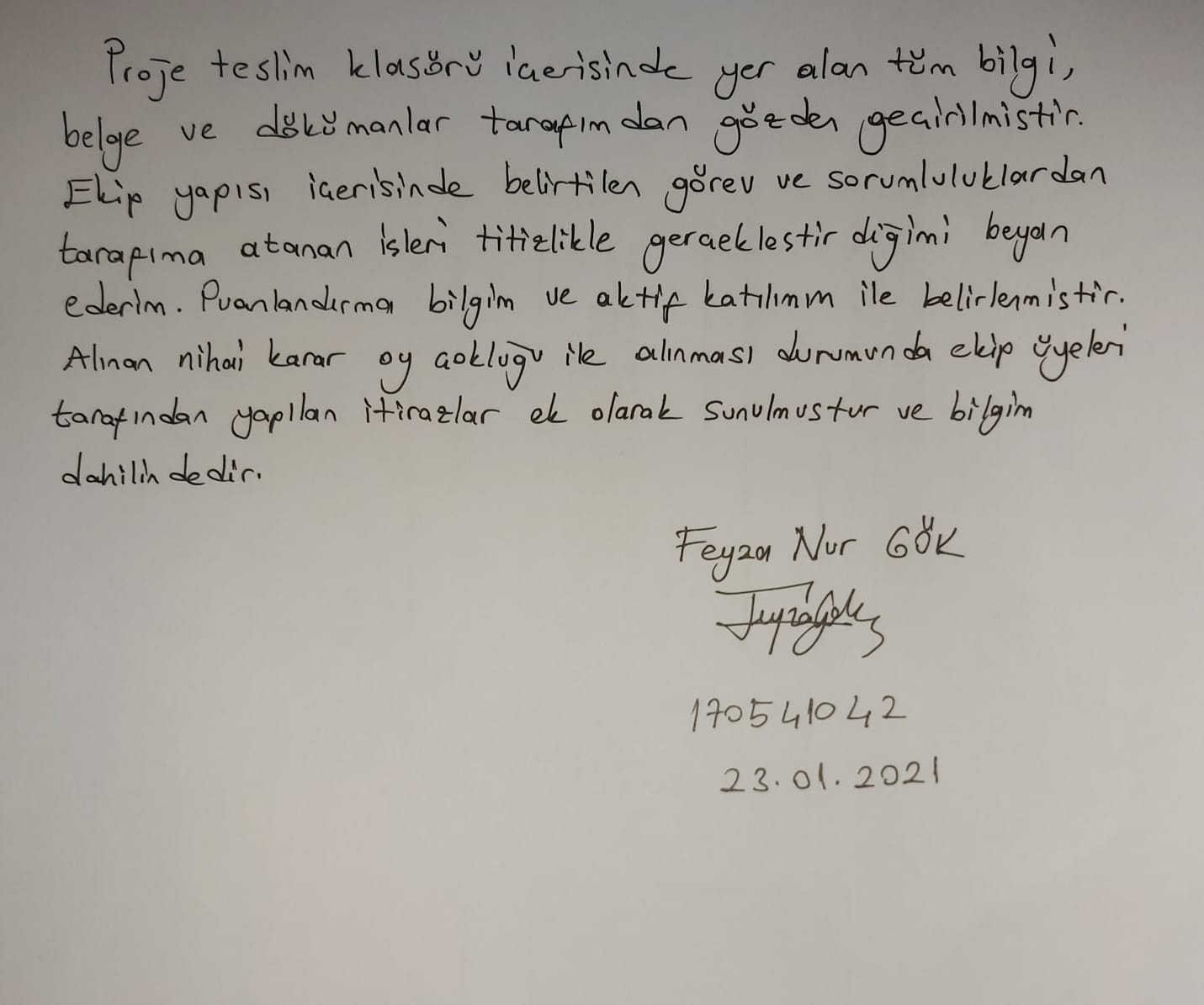
1. Halil İbrahim SÜRÜCÜ
2. Muhammed Hakan ÇELİK



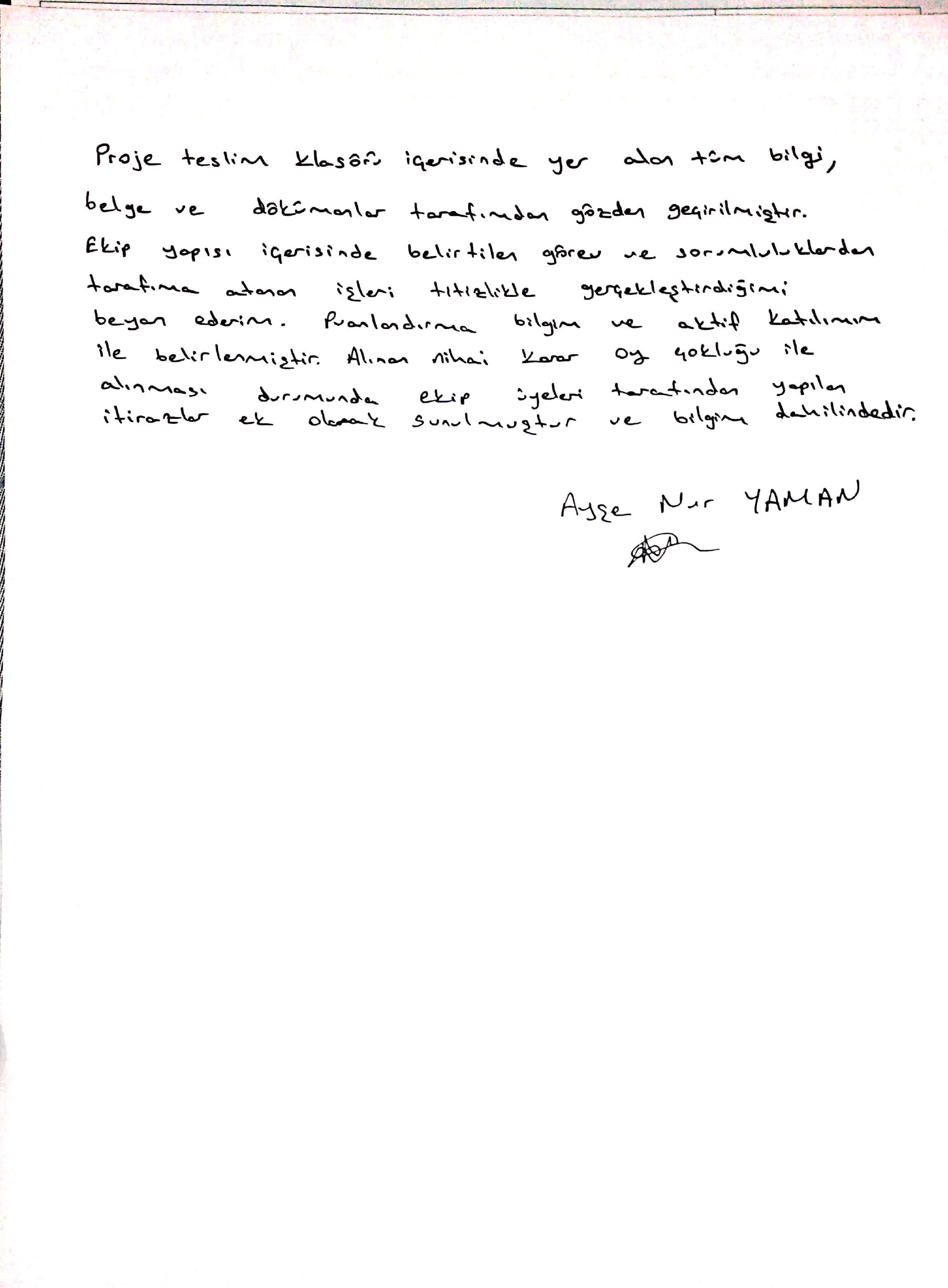
1. Hüseyin GÜNEŞ



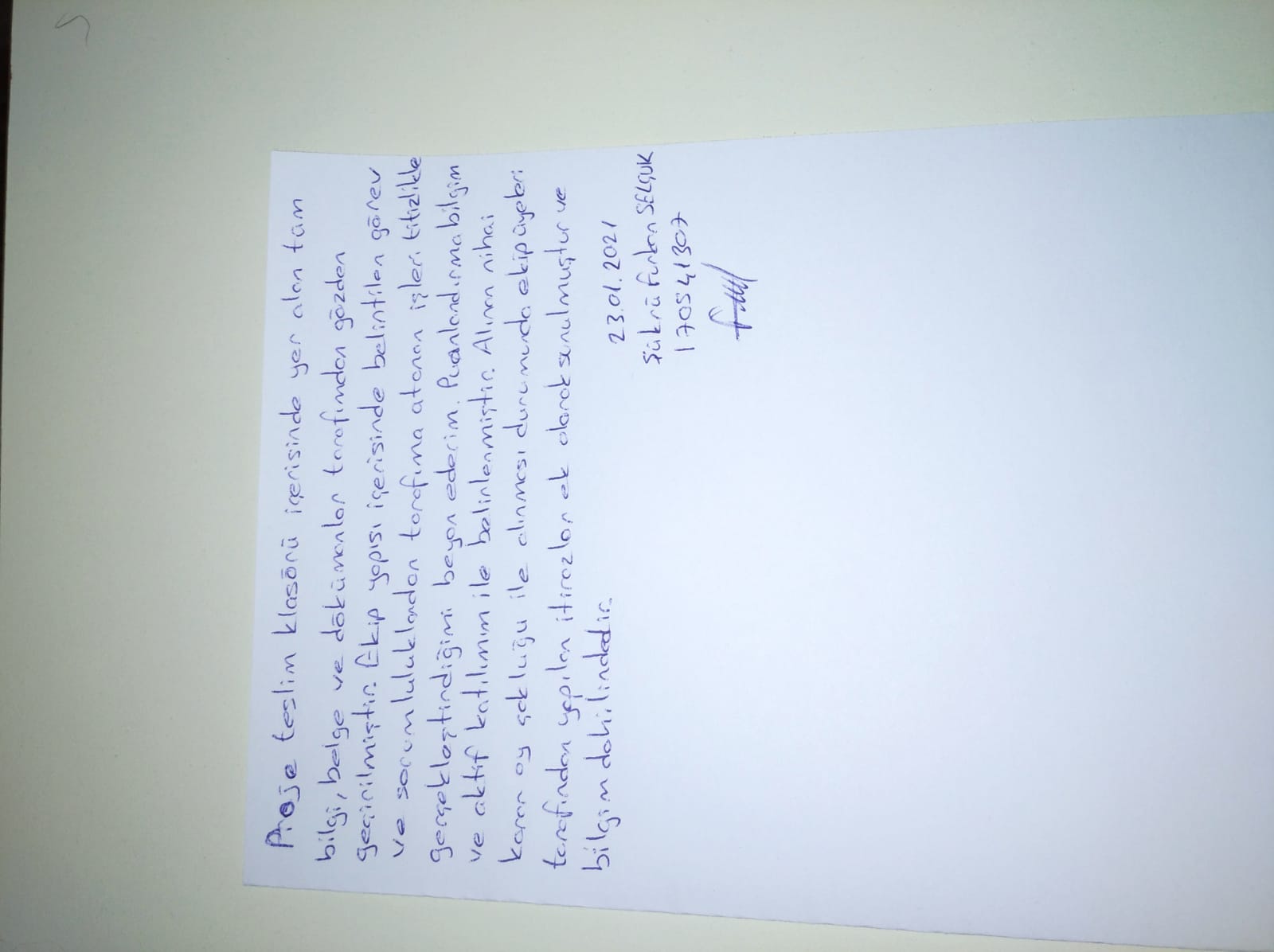
1. Feyzanur GÖK



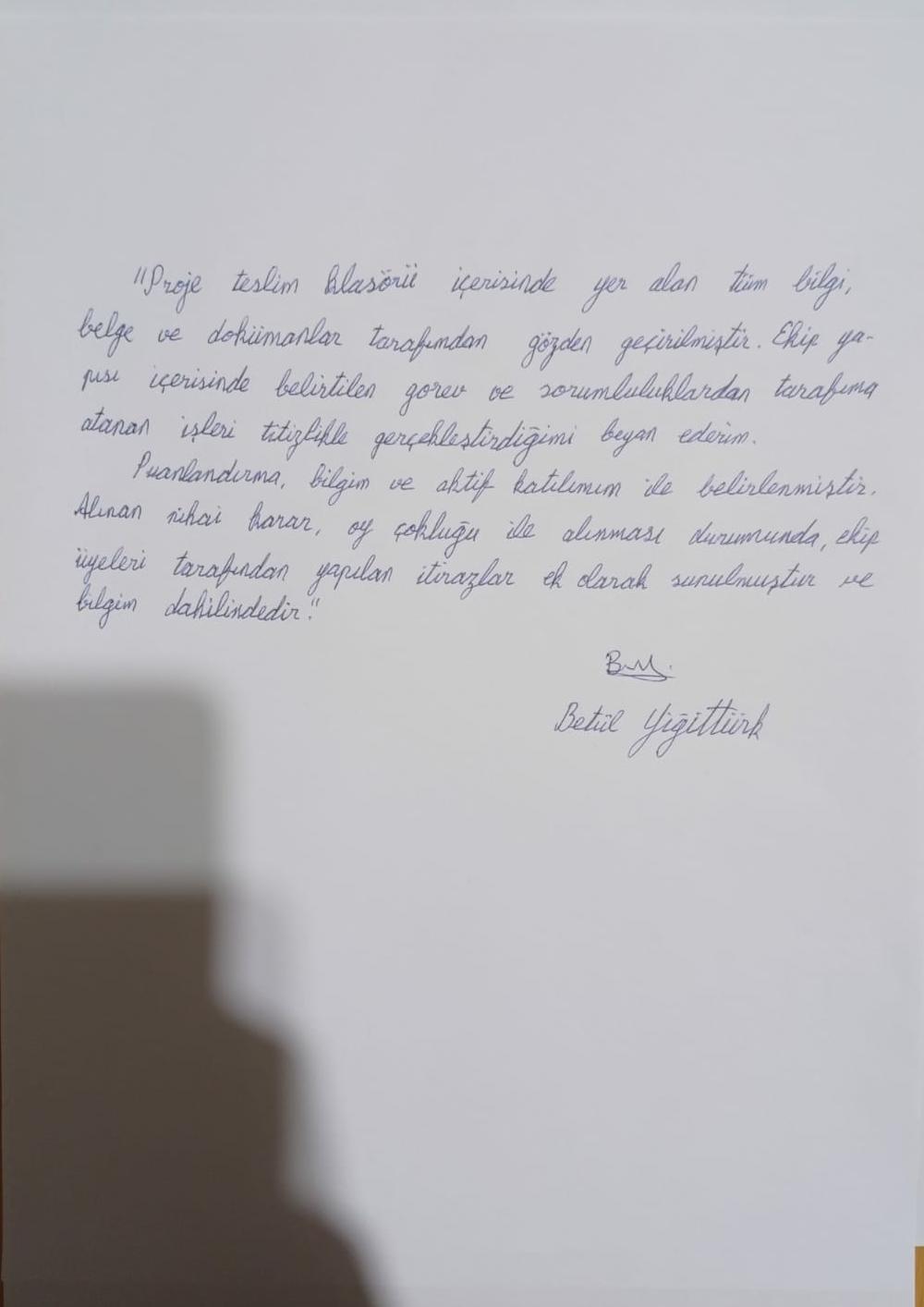
1. Ayşe Nur YAMAN



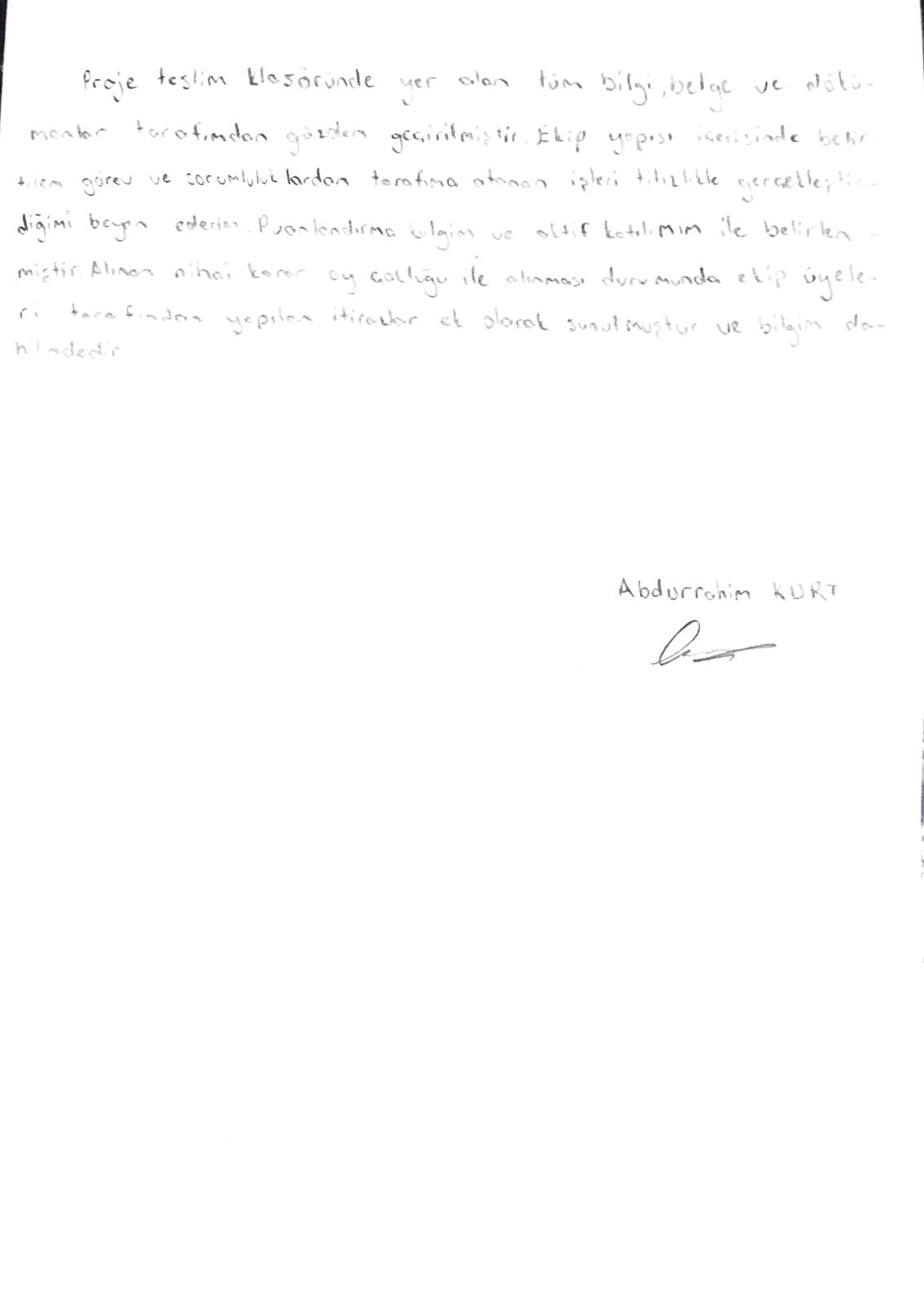
1. Şükrü Furkan SELÇUK



1. Betül YİĞİTTÜRK



1. Abdurrahim KURT



6. Grup İçi Puanlandırma

Grup-2 olarak gerçekleştirdiğimiz bir toplantıda, iş paketleri içerisindeki kişilerin sırayla birbirlerinin performanslarına göre verdikleri puanların ortalaması alınarak bu puanlar belirlenmiştir. Gün sonunda herkes birbiriyle helalleşmiş olup, kimsenin bir gönül kırgınlığı bulunmamaktadır.

15541055 ATAOL BALKAR 🡺 100

16541016 MUHAMMED DİNÇER 🡺 95

16541402 HALİL İBRAHİM SÜRÜCÜ 🡺 75

16542014 MUHAMMED HAKAN ÇELİK 🡺 97

170541012 HÜSEYİN GÜNEŞ 🡺 100

170541042 FEYZA NUR GÖK 🡺 95

170541073 AYŞE NUR YAMAN 🡺 100

170541307 ŞÜKRÜ FURKAN SELÇUK 🡺 95

180541018 BETÜL YİĞİTTÜRK 🡺 100

180541078 ABDURRAHİM KURT 🡺 100